Materials Technology Reports 2026, 4(1), 4151.
doi: https://doi.org/10.59400/mtr4151

Article

Automated quality control of recycled aggregates via deep learning: A unified framework for instance segmentation and mass estimation

Jérôme Lux[1,2*], Pierre-Yves Mahieux[1,2], Philippe Turcry[1,2]

[1] Laboratoire des Sciences de l'Ingénieur pour l'Environnement (LaSIE), UMR CNRS 7356, La Rochelle University, 17000 La Rochelle, France
[2] Department of Civil Engineering and Sustainable Construction, IUT La Rochelle, La Rochelle University, 17000 La Rochelle, France
*** Corresponding author:** Jérôme Lux, jerome.lux@univ-lr.fr

CITATION

Lux J, Mahieux PY, Turcry P. Automated quality control of recycled aggregates via deep learning: A unified framework for instance segmentation and mass estimation. Materials Technology Reports. 2026; 4(1): 4151. https://doi.org/10.59400/mtr4151

ARTICLE INFO

Received: 1 March 2026
Revised: 28 April 2026
Accepted: 8 May 2026
Available online: 14 May 2026

Abstract: The large-scale use of recycled aggregates (RA) in high-grade construction applications is currently hindered by the high variability of their physical properties. Current quality control relies on manual sorting, which is labor-intensive and limits scalability. This study presents RAMSES (Recycled Aggregates Mass estimation and Segmentation), an automated framework based on deep learning, designed to bridge the gap between high-speed production and rigorous material characterization. A central contribution of this work is the introduction of a large-scale, publicly available dataset comprising 90,000 labeled and batch-weighed aggregate instances. This extensive dataset supports a strong statistical robustness across diverse RA compositions and serves as a benchmark for automated waste characterization. Using this dataset, RAMSES performs simultaneous instance segmentation and direct mass estimation from 2D images. By integrating a dual-branch architecture, the model effectively decouples morphological features from instance-dependent density factors. The framework achieves high precision in particle identification (mean Average Precision mAP@[0.5:0.95] = 0.84, mAP@0.5 = 0.91) and a 0.3% relative error in total mass prediction, which meets industrial requirements for batch monitoring. By providing a scalable alternative to manual inspection, this approach improves the consistency of RA-based concrete mixes, directly supporting the transition to a circular construction economy.

Keywords: circular economy; construction and demolition waste; recycled aggregates; convolutional neural networks; instance segmentation; classification; mass prediction

1. Introduction

The recycling of construction and demolition waste (CDW) plays a crucial role in promoting sustainable construction practices and advancing the circular economy. Among CDW by-products, recycled aggregates (RA) are widely reused in road applications but their heterogeneous composition and variable properties often limit their valorization, especially in high-value applications like concrete [1, 2]. Current quality control relies largely on manual sorting (European Standard EN 933-11 [3]), which is labor-intensive, time-consuming, and prone to operator subjectivity. To enable large-scale deployment of RA in the construction sector, there is a growing need for automated, accurate, and real-time characterization methods.

Computer vision and deep learning are powerful tools for object detection, classification, and segmentation in various industrial contexts. In particular, convolutional neural networks (CNNs) have achieved state-of-the-art performance

in instance segmentation tasks, either through region-based methods such as Mask R-CNN [4] or anchor-free models, such as SOLOv2 [5], YOLACT [6], or the famous YOLO family of architectures [7]. More recently, hybrid approaches combining CNNs and Transformers have further improved accuracy and efficiency in image segmentation tasks [8–11].

Numerous recent studies have leveraged deep neural network models to enhance the efficiency and automation of CDW management [12]. These efforts primarily focus on applications such as on-site material handling, including the detection, classification, and sorting of materials for recycling. For this purpose, various deep learning models, particularly the YOLO family, have been successfully trained to segment and classify CDW in real time [13–15]. For example, Zhou et al. used a modified YOLOv5 model on a dataset of 3,046 images (4 different RA classes), obtaining a high mean Average Precision at IoU threshold of 0.5 (mAP@0.5) of 0.948 for bounding box detection. Similar to Zhou et al. [15], Demetriou et al. [13] trained a YOLOv8x model on a dataset of 3,129 images containing 16k objects across 10 CDW classes, achieving a mAP@[0.5:0.95] of 0.936 for bounding boxes and 0.863 for masks.

Complementing these approaches based on neural networks, Serranti et al. [16] recently demonstrated an alternative sensor-based procedure using Short-Wave Infrared Hyperspectral Imaging (SWIR–HSI). Unlike deep learning models that rely on morphological and textural features, this technique identifies materials based on their specific chemical bonds and reflectance spectra. By implementing Partial Least-Squares Discriminant Analysis, the authors successfully automated the recognition of both recyclable constituents (e.g., concrete, natural stones, bituminous materials) and contaminants (e.g., glass, wood, gypsum) as defined by European standards. Their models achieved high classification accuracy, with precision and recall values exceeding 0.9 for all material categories, offering a robust solution for industrial quality control that is independent of particle shape.

Despite these advances, only limited research has focused on extending these methods to predict physical properties of objects, such as mass, from 2D images. This remains an important limitation, as the quality of aggregates is defined by their bulk composition (e.g., Standard EN 933-11). In the studies of Hamdan et al. and Miura et al. [17,18], a regression head is attached to the top of a pre-trained CNN backbone in order to predict the mass of objects on a conveyor belt. In Dohmen et al.'s study [19], the authors applied Mask R-CNN followed by a CNN regressor to predict the body mass of heifers from RGB images. In these works, there is only one type of material on each frame, making it possible to compute a global mass per frame without object localization and classification. In Standley et al.'s study [20], a more sophisticated architecture, "image2mass", is proposed to estimate the mass of a single object. The proposed model is based on a pre-trained Xception backbone and two interconnected modules which predict the density and the volume of the object. The model is trained on a dataset made of about 150 k images collected from the Amazon Marketplace Web Service Application Programming Interface (API). This work highlights the benefits of using two complementary modules for density and shape estimation.

In our previous work [21], we introduced the Recycled Aggregates

Characterization Network (RACNET), a neural network designed to classify individual RA and estimate their mass using only 2D images [21].

In RACNET, the mass prediction was computed as the product of the binary mask area and two learned factors: one linked to the aggregate class, extracted from a ResNet backbone [22], and another capturing additional geometric information from early feature maps. This approach proved effective, achieving a mean absolute percentage error (MAPE) below 3% across 11 classes.

However, RACNET had important limitations. It required pre-segmented images of single aggregates, obtained via a separate lightweight instance segmentation step, which hindered real-time applicability. Moreover, inputs had to be resized to a fixed dimension, leading to loss of resolution for larger aggregates and a noticeable decline in both classification and mass prediction performance.

To address these limitations, we propose RAMSES (Recycled Aggregate MasS Estimation and Segmentation), a new deep learning architecture built upon SOLOv2 [5]. RAMSES unifies instance detection, classification, segmentation, and mass estimation within a single framework, enabling direct processing of high-resolution images with multiple aggregates. By introducing two dedicated branches for instance-dependent and geometry-dependent factors, the model leverages complementary information to produce accurate per-instance mass predictions. This approach has the potential to replace manual sorting and traditional granulometric methods, offering both scalability and precision in RA characterization.

2. Materials and methods

2.1. Model architecture

Following the results in Standley et al.'s work [20] and our previous work [21], we assume that the mass of an object is the product of a factor related to the instance, denoted δ and a local factor denoted τ accounting for the geometry of the aggregate, such that the mass of an aggregate i can be written as:

$$m_i = \delta_i \sum\nolimits_{x \in M_i} a_p \times \tau(x) = \delta_i a_p \mathrm{T}_i \tag{1}$$

Where $\tau(x)$ is a local factor predicted by the network at the spatial location x, T_i is the sum of $\tau(x)$ for all pixels in the binary mask M_i of the instance i, δ_i is the instance-wise mass factor associated with the instance i and a_p is the area of one pixel (cm^2). To be consistent, the product $\delta_i \times \mathrm{T}_i$ is expressed in $\mathrm{g} \cdot \mathrm{cm}^{-2}$, which can be seen as the product between a density and a thickness.

It is important to note that, although mass is known, the volume and density of individual aggregates remain unavailable, as our input is limited to 2D images. Consequently, there are no ground truth targets for δ and τ. δ can be interpreted as an instance-specific global factor, while τ provides the network with additional flexibility to account for local morphological or geometrical features. In **Appendix C**, we show that this two-factor model outperforms a one-factor model, where the mass is predicted using only the product between the mask area and the δ factor. Although the improvement is modest (a 2-percentage-point reduction in MAPE), it is consistent

across all classes.

In recent years, there has been a significant amount of research on instance segmentation and classification [23]. Numerous approaches and architectures have been developed to address this problem using either a convolutional neural network [4, 7, 24, 25], Vision Transformers [8–10], or both [11]. Recently, some authors proposed fully convolutional and anchor-free solutions to the instance segmentation problem [5, 6, 26]. Among these models, the SOLOv2 architecture is a very efficient and simple pipeline featuring dynamic mask kernels, where the convolutional kernels are conditioned by the input image. SOLOv2 is based on a convolutional network backbone and a Feature Pyramid Network (FPN) for multiscale object detection. It features two shared heads: one head is responsible for the classification (like in Fully Convolutional One-Stage Object Detection (FCOS) network [27]), and the other is responsible for predicting the mask convolution kernels at different locations. Furthermore, all FPN levels are aggregated into one high resolution unified mask feature representation of spatial shape $H_{mask} \times W_{mask}$. To generate the instance masks, a convolution is performed between the mask feature map and the N_{pos} predicted kernels corresponding to positive detections in the class head. A sigmoid activation is applied to the results of shape $N_{pos} \times H_{mask} \times W_{mask}$, as each slice corresponds to a single predicted instance. A final Non Maximum Suppression (NMS) algorithm is then used to filter the detected masks and to get the final N_{pred} predictions.

To integrate the mass prediction in the SOLOv2 architecture, we first add a new shared head (instance-wise mass factor branch) to predict the instance factor at each grid location in all FPN levels. The tensor δ of shape N_{pos} is formed by taking only the values at the grid locations corresponding to positive detections in the class head. We also add a new branch after the merging of all FPN levels, in parallel to the unified mask representation. This branch is responsible for predicting the factor $\tau(x)$ at each location, which gives a tensor of shape $1 \times H_{mask} \times W_{mask}$. To compute the mass of each instance, each mask (after the sigmoid activation) is first multiplied by the τ factor feature map, resulting in a tensor of shape $N_{pos} \times H_m \times W_m$. This tensor is then summed along the spatial axis, which gives a tensor T of shape N_{pos}. Finally, the mass of each instance is computed as the element-wise product between the δ and the T tensors. The overall architecture is illustrated in **Figure 1**. Additional details on the implementation are provided in **Appendix A**.

2.2. Mass target values

The input images can have different spatial resolutions, which means that an aggregate can have a different mask area depending on the resolution. While δ is independent of the resolution, the term $T = \sum_{x \in M} \tau(x)$ depends on the number of pixels in the mask. However, as the network is resolution-agnostic, the predicted values of $\tau(x)$ for a given object should be invariant to the image resolution. The target values for mass prediction, noted m^*, must therefore take into account the resolution and scales

with the mask area. Here, we propose to define the target values as:

$$m^* = R^2 m \tag{2}$$

with R the resolution in pixel per cm, m the mass in g and m^* in g·pixel·cm^{-2}. Multiplying both sides of Equation (1) by R^2, we have:

$$m_i^* = \delta_i \sum\nolimits_{x \in M_i} \underbrace{a_p R^2}_{=1} \tau_i(x) = \delta_i \sum\nolimits_{x \in M_i} \tau_i(x) \tag{3}$$

where $a_p R^2 = 1$ pixel.

We can estimate the average value for $\tau_i(x)$ for each instance as:

$$\overline{\tau_i(x)} = \frac{\sum_{x \in M_i} \tau_i(x)}{M_i} \tag{4}$$

with M_i the area of the binary mask in pixels. Then, we have the following estimation for the product between the instance mass factor and the average geometrical factor:

$$\delta_i \overline{\tau_i(x)} = \frac{m_i^*}{M_i} = \frac{m_i}{A_i} \tag{5}$$

with A_i the area of the mask in cm^2.

The distribution of $\delta_i \overline{\tau_i(x)}$ in our dataset is presented in **Figure 2**. The mean is 0.69, and the standard deviation is 0.4; most of the values are below 2. Scaling the mass targets by R^2 appears therefore sufficient to ensure that the local τ values are well distributed and need no additional normalization.

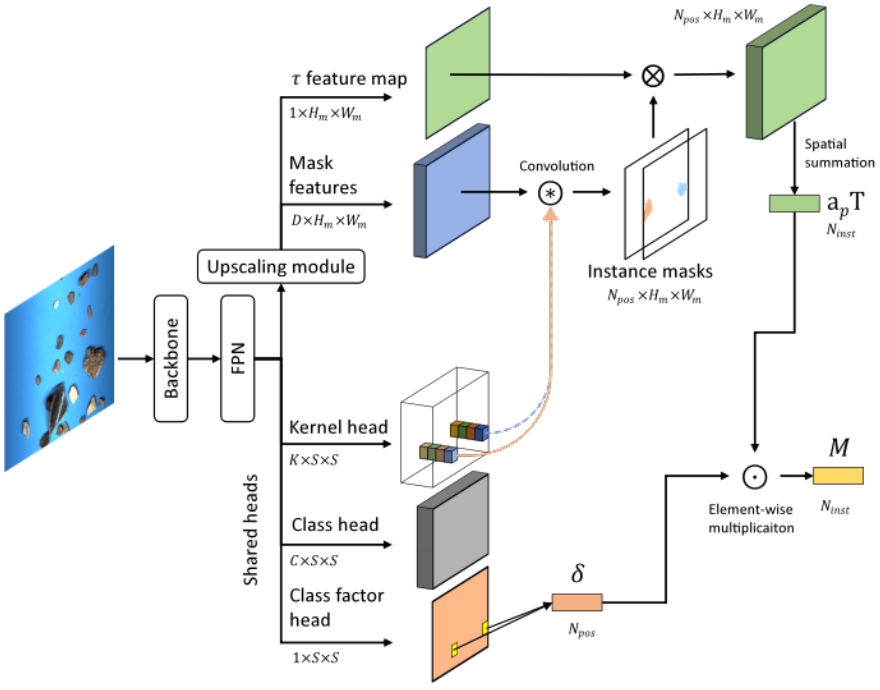


Figure 1. Model overview.

Note: Two new branches have been added to the SOLOv2 architecture. A new shared head predicts the instance mass factors at each location, i.e., for each detected object in the class head at different scales. Another branch predicts the τ feature map, which gives the values $a_p T_i$ for each instance i. The final mass tensor is computed as the element-wise product between the instance mass factor tensor δ and the geometrical factor tensor $a_p T$.

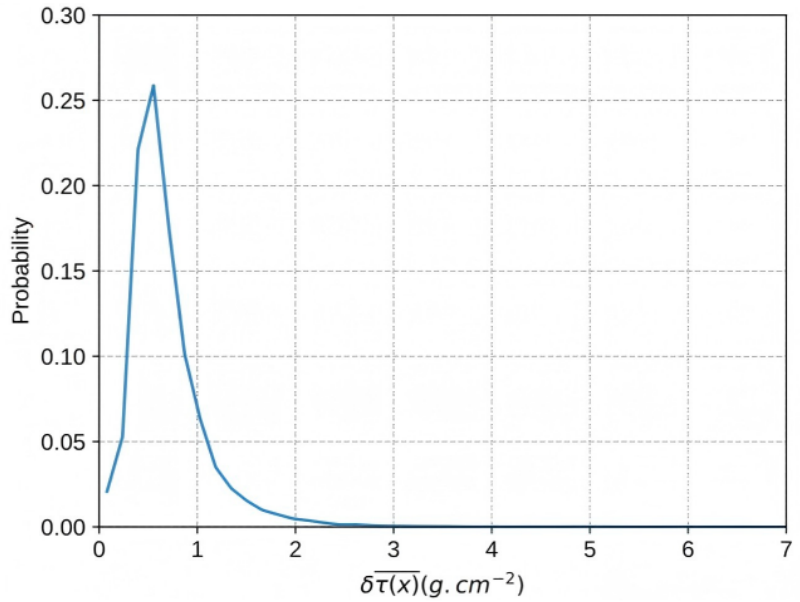


Figure 2. Distribution of $\delta\overline{\tau(x)} = \frac{m}{A}$ values for our current dataset.

2.3. Recycled aggregate dataset

To train our neural networks, we constructed a dataset currently containing nearly 90 k instances of individual segmented recycled aggregates in 2,683 pictures (available at https://doi.org/10.57745/KC4EA2). The aggregates were sorted manually into 18 classes, partly based on the EN 933-11 standard. The standard defines 6 classes of RA, namely concrete grains (Rc), natural stones (Ru), ceramics (Rb), bituminous grains (Ra), glass (Rg) and others (X). However, some classes are highly heterogenous, like the Ru or Rb classes, which contain elements with considerable differences in aspect and properties. We have therefore proposed a finer classification in order to define more homogeneous classes based on the EN 933-11 naming convention. See **Table 1** for details of the classes used in the dataset. Note that in some images we have placed a 5-euro cent coin in order to determine the resolution. The network is trained to detect the Coin class, but it is not included in the table.

Table 1. Class definition and number of elements per class in the current recycled aggregate dataset.

Class ID	Number of instances	Class description
Pl	55	Plaster
Ra	7,309	Bituminous grains
Rb01	7,795	terracotta material, like clay bricks or roof tiles
Rb02	3,742	Ceramic tiles, earthenware tiles, etc.
Rc	26,910	Concrete grains
Rcu01	547	Lime mortar
Rg	145	Glass
Ru01	14,245	White stones such as limestone
Ru02	10,614	Grey stones such as basalt & grainy stones of similar color
Ru04	5,683	Siliceous and rather angular stones
Ru05	5,993	Rounded alluvial stones
Ru06	2,891	Slate
X01	1,797	Wood

Table 1. *Cont.*

Class ID	Number of instances	Class description
X02	586	Plastics
X03	769	Steel
X04	133	Paper and cardboard
SHELLS	48	Shells
UNKNOWN	28	Do not belong to other categories
Total	**89,600**	

Our dataset has the particularity of including images with a high number of aggregates (up to around 600), which helps to represent the most realistic and varied industrial situations.

The photo acquisition protocol was described in two previous articles [21, 28]. About a third of the aggregates were photographed on a copy stand using a Fujifilm X-T20 camera of 24 million pixels resolution (6,000 × 4,000 pixels) along with two lenses, a Fujinon XF60 mm F2.4 R Macro (for 4/10 fraction, resolution 23 pixels/mm) and a Fujinon XF35 mm F2 R WR (for 10/31.5 fraction, resolution 12 pixels/mm). The other aggregates were captured on a conveyor belt using a JAI SW-8000Q-10GE-F line camera (8,192 pixels, resolution of 28.7 pixels/mm). The images obtained have a size of 4,096 × 8,192 and are then cropped to 4,096 × 6,144. See **Figure 3** for an overview of the device and **Figure 4** for an overview of the images obtained on a stand and a conveyor belt.

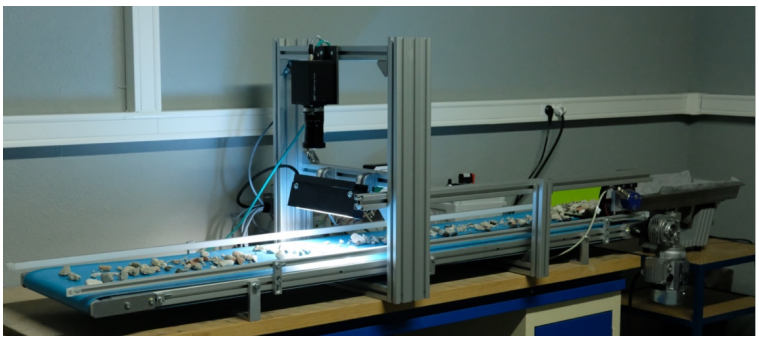
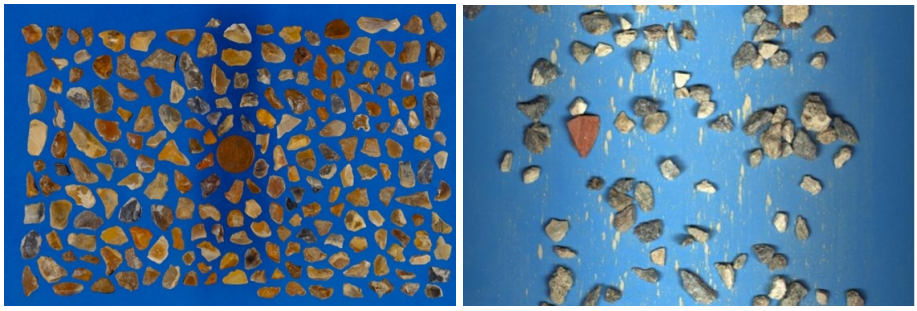


Figure 3. Automatic aggregates characterization device. A vibratory feeder (right) distributes aggregates on the conveyor belt. A line camera captures images of the aggregates.



(a) **(b)**

Figure 4. Pictures from our RA dataset: **(a)** Photo taken on a stand; **(b)** Photo taken on a conveyor belt.

Most of the aggregates were weighed in small homogeneous batches so that their

individual mass could be calculated using a form factor approach (see Lux et al.'s article [21] for details on the method and error assessment), and approximately 1,500 aggregates were individually weighed to validate the approach. The dataset includes some photos containing aggregates from a single class, and others with a mixture of aggregates of different classes. Note that it is much more difficult to classify an aggregate using only a picture than by being able to physically handle it, even for a well-trained operator. The mass of the aggregates is therefore not estimated for images containing a mixture of classes.

When we started creating the dataset, the images were annotated manually or using a standard image analysis method. The annotation was then carried out using our own trained models (mostly SOLOv2) and/or using the Segment Anything Model [29] in the CVAT software [30].

As shown in **Table 1**, the dataset is very unbalanced, as some elements are very uncommon in the RA available on recycling platforms. We propose a few solutions in the next section to create more balanced sets to train the network. Note that compared to our previous work [21], the Ru02 (grey stones) and Ru03 (grainy stones) classes were merged because it is often very difficult to distinguish between them. It contains mostly basalt, diorite and granite aggregates. The distribution of the number of aggregates per image is presented in **Figure 5**. We found that about 75% of the images contain fewer than 50 aggregates, and approximately 232 images (8.6%) contain more than 200 aggregates.

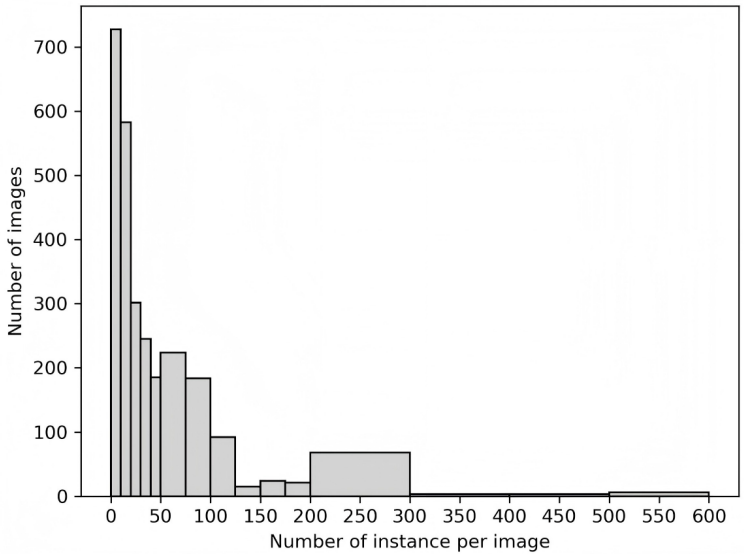


Figure 5. Distribution of the number of objects per image.

2.4. Training procedure

The network was trained in 3 main steps. First, we trained the SOLOv2 heads and the FPN with a frozen pre-trained ConvNextv2_femto backbone [31]. Then, we trained the entire instance segmentation model (excluding the mass prediction heads), using a very small learning rate for the backbone. This was necessary because our dataset is highly specific, so adapting the backbone to our particular problem was essential. Finally, we trained only the mass prediction heads.

We used a Focal Loss to train the detection/classification head [32] and a DICE

loss to train the mask segmentation head. As the mass target values m^* are very spread out, we use the MAPE loss function to train the new mass prediction heads.

Regarding the mass prediction, it is calculated as follows during training. First, each slice of the ground truth (GT) masks is multiplied element-wise by the predicted τ feature map. Then, the result is summed along the spatial axis, giving the tensor T of size $N_{GT\ instances}$. The δ tensor is constructed by selecting the values of the instance mass factor mass head at positive locations in the GT class targets (flattened, for all FPN levels). Its shape is $N_{pos} > N_{GT\ instances}$, since each GT instance may be associated with several positive locations. Each value of the δ tensor is then multiplied with the T value of its corresponding mask to compute the final normalized mass tensor of size N_{pos}.

The initial learning rates for the different components of the network are shown in **Table 2**. The learning rate is halved when the training loss does not decrease for 3 successive epochs. The training is stopped when the validation/test loss does not decrease for 10 successive epochs. We then use the model with the lower validation loss. Note that we did not use a separate test and validation sets, as the scarcity of images in some classes makes a three-way split impractical, but there is no interaction between the validation/test loss and the training dynamics.

Table 2. Initial learning rate for each component of the network.

Component	Initial learning rate	Stage
Backbone	10^{-5}	2
FPN and classification head	$10^{-3}/10^{-5}$	1/2
Kernel and mask heads	$10^{-5}/5 \times 10^{-6}$	1/2
Mass prediction heads	10^{-4}	3

We used an AdamW optimizer [33] with a weight decay of 5×10^{-3} for all stages.

2.5. Training sets

As seen in Section 2.3, some classes (namely Pl, X04, SHELLS and UNKNOWN) contain very few elements and were not used for training. We also added the "Coin" class, so that the number of classes in the training set is 15. To increase the number of elements in the remaining under-represented classes, we generated synthetic images using existing segmented instances. We generated images containing approximately 100 instances of the most under-represented classes. To do so, we first took 177 photos of the conveyor belt without aggregates in order to have a collection of background images. Then, existing instances were randomly rotated and resized and put onto a randomly chosen background image. We allow an overlap of at most 10% of the instance area. We also add transparency and randomly oriented shadows at the border of each instance. **Figure 6** shows an example of generated images. 200 synthetic images were generated for a total of 18 k instances. Obviously, this synthetic dataset does not include mass data, because the instances are randomly resized and have various resolutions.

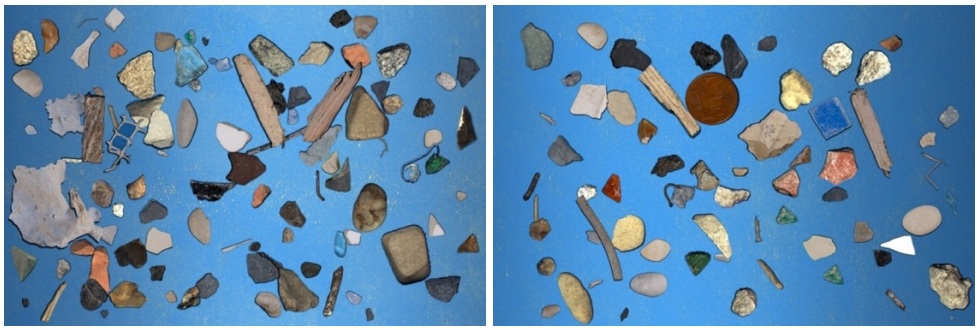


Figure 6. Example of synthetic images generated using existing instances.

Another approach to reducing class imbalance is to perform oversampling. When generating the training dataset, we allow an image to be used 2 times when needed (not applied to synthetic images).

Using these two approaches, we generated a training set containing approximately 145 k instances in 15 classes, as shown in **Table 3**. The dataset consists of 3,472 images, 2,194 of which are unique. The full training set is much more balanced than the dataset, although some classes are still under-represented (see **Table 3**).

Table 3. Number of instances per class in the two training sets.

Class	Full training set (stage 1 & 2)	Mass training set (stage 3)
Coin	668	549
Ra	14,416	13,539
Rb01	15,145	14,270
Rb02	7,952	6,348
Rc	20,749	19,749
Rcu01	*1,825*	*1,056*
Rg	*1,934*	*271*
Ru01	20,814	19,843
Ru02	21,474	14,246
Ru04	11,821	10,193
Ru05	12,557	10,876
Ru06	6,237	4,734
X01	3,804	2,506
X02	*2,765*	*1,148*
X03	*2,851*	*1,492*
Total	145,012	120,820

Note: Classes in italic are used during training only.

Note that even with oversampling and synthetic images, some classes do not contain enough elements to avoid overfitting (in italic in **Table 3**). We kept them for training, but we did not use them in the test set to evaluate the network's performances.

During training, we also used various online augmentations to limit overfitting (see **Table 4**). We found that the most effective technique was CutMix. This technique was, however, not used during stage 3, which focused on mass prediction training. This is because CutMix results in incomplete instances, and it's not straightforward to recompute the mass for a partial instance.

Table 4. Augmentations used during the different stages of the training.

Augmentation	Probability	Parameters	Stage
Gaussian Noise	33%	$\mu = 0$ $\sigma = 0.05$	All
Random Brightness	33%	±5%	All
Random Rotation	33%	[−60°; 60°]	All
CutMix	50%	1 to 3 patches. Size between 10% and 25% of the image size.	1 & 2

For stage 3, we used a smaller dataset, as we dropped synthetic images and images containing instances without mass data (see **Table 3**). This dataset contains 2,897 images, 1,750 of which are unique.

2.6. Test set

The test set consists of 5,000 aggregates (approximately 500 instances per class) across 258 unique images that were not used during training. This set only contains instances with mass data and no synthetic image. As we have discarded the most under-represented classes, the test set contains only 10 classes, as detailed in **Table 5**.

Table 5. Number of instances in each class in the test dataset.

Class	Test set
Ra	500
Rb01	500
Rb02	499
Rc	500
Ru01	500
Ru02	500
Ru04	500
Ru05	500
Ru06	499
X01	502
Total	**5,000**

3. Results and discussion

3.1. Evaluation metrics

To assess the overall performance of our approach, we evaluate both the quality of the instance segmentation (classification and segmentation) and the error in mass prediction.

For the first task, we use the Average Precision (AP) metric [34]. AP is computed for each class at different intersection-over-union (IoU) thresholds ranging from 0.5 to 0.95 with increments of 0.05. Here, the IoU is computed using the masks of the instances and not their bounding boxes. The mean AP (mAP_t) is computed by averaging AP over all classes for a given threshold t and the mAP@[0.5:0.95] is the average of mAP_t for all IoU thresholds.

The mass prediction is assessed at two levels:

1. **Instance-level**: we compute the Mean Absolute Percentage Error (MAPE) for positive detections, defined as predicted instances with an IoU > t and a matching class relative to the ground truth. Each ground truth instance is assigned to the corresponding prediction with the highest class score. To further assess the robustness of our model against significant failures, we also report the percentage of instances with an Absolute Percentage Error (APE) exceeding 50%. While these metrics provide insight into individual detection quality, they do not quantify the cumulative mass error per class, which is critical for practical applications.
2. **Sample-level:** we calculate the Relative Error (RE) between the total ground truth mass (m_{GT}) and the sum of predicted mass per class (m_{PRED}) after NMS: $RE = \frac{m_{PRED} - m_{GT}}{m_{GT}}$. This global metric considers all predicted instances, thereby mirroring real-world applications where ground truth data is unavailable and compensation effects between individual errors may occur.

3.2. Performance assessment

The evaluation is conducted using the following fixed parameters, determined through a sensitivity analysis (see **Appendix B**) over 48 combinations of class score, mask segmentation, and NMS thresholds:

- Mask head segmentation threshold (after sigmoid activation): 0.6.
- Minimum score for positive detection: 0.5.
- Mask NMS threshold: 0.35.

Figure 7 illustrate the mAP_t for four distinct input sizes: 1,024 × 1,536, 1,280 × 1,920, 2,048 × 3,072, and 3,072 × 4,608, plotted against each IoU threshold t. These results indicates that 2,048 × 3,072 provides the best performance among the evaluated resolutions. While the 1,024 × 1,536 resolution performs substantially worse, both 1,280 × 1,920 and 3,072 × 4,608 yield comparable mAP scores. Furthermore, the highest resolution does not provide measurable performance gains while increasing memory usage and inference time. A large input resolution is required to properly process the smallest aggregates (on the order of 4 mm).

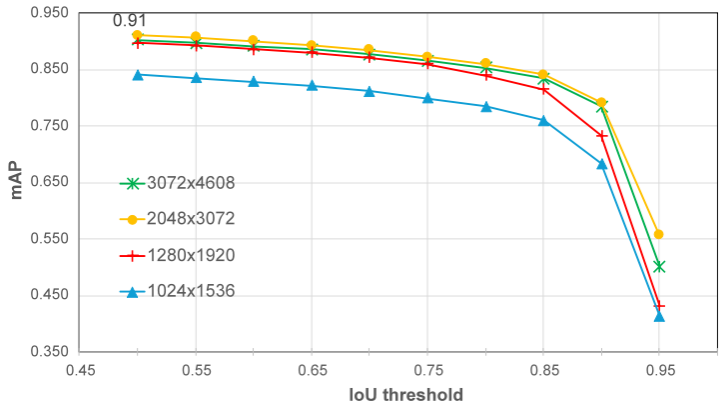


Figure 7. mAP for IoU thresholds in [0.5, 0.95] for different input sizes.

The mAP@[0.5:0.95] for the 2,048 × 3,072 resolution is of 0.84 which indicating that the SOLOv2 detector performs competitively in this setting. The mAP@[0.5:0.95]

is close to those reported in Demetriou et al.'s work [13] for models with a similar number of parameters: 0.81 for YOLOv8s (11.2 M parameters) and 0.849 for YOLOv8m (25.9 M), for 10 classes and image size 640 × 640. Note that in our case, several classes (especially the Ru series) exhibit strong visual similarity, which may increase the difficulty of the classification task.

To further analyze the model's performance, we show the AP and MAPE results for each class in **Table 6**. The last column presents the percentage of correctly detected aggregates with an absolute percentage error (APE) greater than 50%. These metrics were calculated using an IoU threshold of 0.5, which corresponds to the highest mAP score (0.91). The final row, labeled "ALL," reports the mean AP and mass MAPE values across all classes. **Table 7** shows the predicted values for the total mass of each class, as well as the relative error with the GT mass. As the network was trained on 15 classes, but the test set contains only 10 classes, **Table 7** also reports false positive detections, i.e., predictions of classes that are not present in the test set. These predictions represent less than 0.2% of the total ground truth mass.

Table 6. AP, Mass Mean absolute Error (MAPE) and percentage of outliers (APE > 50%) on the test dataset for an IoU threshold = 0.5.

Class	AP@0.5	Mass MAPE @0.5 (%)	% of outliers: Aggregates with a mass APE > 50%
Ra	0.81	34.7%	24.83%
Rc	0.93	7.9%	0.83%
Rb01	0.92	7.7%	1.46%
Rb02	0.89	9.5%	1.28%
Ru01	0.97	5.7%	0.41%
Ru02	0.95	6.4%	0.61%
Ru04	0.95	7.7%	0.83%
Ru05	0.93	6.6%	0.63%
Ru06	0.94	8.9%	2.29%
X01	0.87	27.3%	12.85%
All	**0.91**	**12.0%**	**4.48%**

Table 7. Ground truth and predicted mass, as well as relative error (%).

Class	GT mass (g)	Predicted (PRED) mass (g)	Total mass relative error (RE) (%)
Ra	350.2	286.4	−18.21%
Rc	362.8	329.5	−9.17%
Rb01	305.7	313.0	2.37%
Rb02	3,201.1	3,207.2	0.19%
Ru01	785.1	791.9	0.87%
Ru02	542.9	528.2	−2.70%
Ru04	429.2	416.9	−2.86%
Ru05	522.4	504.0	−3.51%
Ru06	194.7	195.4	0.33%
X01	369.2	456.4	23.60%
X02	0.0	11.3	
X03	0.0	0.1	
Rg	0.0	2.0	
All	**7,063.3**	**7,042.2**	**−0.30%**

Average Precision values are relatively consistent across classes, except for the Ra and X01 classes, which have lower scores. It is expected for the X01 class (wood), since it contains fewer unique instances than the other classes, and the aggregates have a much greater variability in shape and size than those of the other classes. The network sometime encounters difficulties in detecting thin, elongated and tortuous objects. Wood aggregates are often over-segmented, which leads to an overestimation of the global predicted mass. The lower AP for the Ra class is mostly due to a higher number of false positives. The network occasionally has trouble identifying individual bituminous aggregates when they are touching or overlapping. The dark color of these aggregates often renders their edges indistinct, complicating detection.

The Mass MAPE is comprised between 6% and 35% and can be rather high for certain classes (Ra and X01). For most classes, however, the percentage of aggregates having an APE greater than 50% remains extremely low (<2.3%), indicating that the segmentation and mass estimation are robust for a vast majority of instances. In contrast, the Ra and X01 classes show significantly higher outlier rates, with 25% and 12.9% of aggregates exceeding the 50% APE threshold, respectively. This is primarily a consequence of the lower AP scores for these two classes, leading to more substantial relative errors on their total mass prediction (23.6% for wood and −18.2% for bituminous aggregates).

The error on the total mass remains much smaller than the MAPE due to two main factors. First, the MAPE is an arithmetic mean, so an error in estimating the mass of a light aggregate is given the same weight as an error in estimating the mass of a heavy aggregate. Second, there is a compensating effect between the overestimated and underestimated masses that reduces the overall error. It is encouraging to note that the relative error between the total mass and the total ground truth mass is as low as 0.3%. The predictions are remarkably close to the ground truth for certain classes, notably all Ru types, with errors falling between 0.87% and −3.5%.

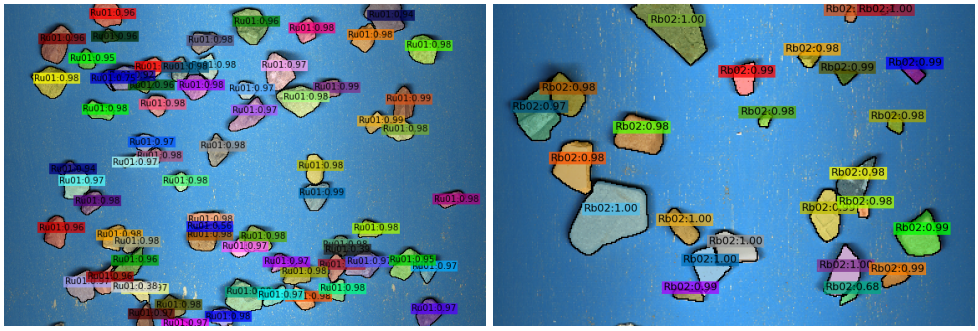
Figure 8 shows two examples of failure cases, where mass estimation is inaccurate. As illustrated in **Figure 8a**, successful detection and segmentation do not guarantee precise mass estimation; in this instance, the prediction shows a substantial discrepancy, with a relative error of 153%, despite a low absolute error.

(a)
GT mass = 0.96 g
PRED mass = 2.42 g
Mass RE = 153%

(b)
GT mass = 24.6 g
PRED mass = 45.3 g
Mass RE = 84%

Figure 8. Failure cases with corresponding mass error.

In contrast, **Figure 9** displays two images where both detections and mass predictions were accurate.



(a)
GT mass = 53.33 g
PRED mass = 53.2 g
Mass RE = −0.24%

(b)
GT mass = 61.3 g
PRED mass = 63.1 g
Mass RE = 2.9%

Figure 9. Accurate detection, segmentation and mass prediction.

4. Conclusion

In this work, we introduced RAMSES, a new deep learning architecture that simultaneously performs detection, classification, instance segmentation, and mass estimation of recycled aggregates (RA) from high-resolution 2D images. Building upon the SOLOv2 framework, the model integrates two additional branches to disentangle class-dependent and geometry-dependent factors for mass prediction. Our results demonstrate that RAMSES achieves high performance in both instance segmentation (mAP@0.5 ≈ 0.91) and mass estimation, with a mean absolute percentage error of 12% at the aggregate level and a relative error of 0.3% on the total predicted mass across classes. These results highlight the potential of combining object recognition and physical property estimation within a unified framework.

However, some limitations remain. The model exhibits signs of overfitting due to the imbalance of certain classes in the training set, and performance degrades for underrepresented or highly variable categories such as wood. The prediction of individual aggregate mass is also sensitive to resolution and image quality. While synthetic data augmentation has proven useful, it does not fully compensate for the scarcity of real annotated instances in rare classes. Furthermore, estimating target masses via form factors instead of direct measurement introduces noise, which limits the precision of individual mass estimation. Therefore, our method is better suited for bulk batch assessment than individual grain-size distribution analysis. To address this, future work will focus on two strategies: increasing the number of individually weighed aggregates to reduce target mass uncertainty and developing alternative mass estimation methodologies.

Increasing the amount and diversity of training data would very likely reduce overfitting and improve prediction accuracy, particularly for underrepresented classes of aggregates. In addition, incorporating complementary spectral information (e.g., near-infrared imaging) could provide more discriminative features, helping to better separate materials with similar appearance in the visible spectrum while preserving the simplicity of a 2D acquisition setup.

As demonstrated in our sensitivity study, the results are robust across a reasonable range of range of hyperparameters, including classification, segmentation and NMS thresholds. Nevertheless, adopting a query-based Transformer architecture for instance detection and segmentation, such as Mask2Former [9], could potentially further enhance this robustness.

Finally, we plan to optimize the inference speed to further improve the system's industrial applicability. While the current, unoptimized implementation already enables the characterization of several kilograms of material within a few minutes, representing a substantial reduction in processing time compared to manual sorting, there remains significant potential for further acceleration. This could allow RAMSES to better meet the throughput requirements of continuous industrial monitoring.

Beyond research, the proposed approach has promising applications in the automatic characterization of recycled aggregates, offering a viable alternative to manual sorting and conventional granulometric analysis. By providing per-aggregate classification and mass estimation, RAMSES can contribute to quality control in recycling platforms and support the broader adoption of recycled aggregates in construction, in line with circular economy objectives.

Author contributions: Conceptualization, JL; methodology, JL; software, JL; validation, JL, PT and PYM; formal analysis, JL; investigation, JL; resources, JL; data curation, JL; writing—original draft preparation, JL; writing—review and editing, JL, PT and PYM; visualization, JL; supervision, JL; project administration, JL; funding acquisition, JL, PYM and PT. All authors have read and agreed to the published version of the manuscript.

Funding: This work was partly supported by Aquitaine Science Transfert.

Institutional review board statement: Not applicable.

Informed consent statement: Not applicable.

Data availability statement: The data that support the findings of this study are openly available at: https://doi.org/10.57745/KC4EA2. The code as well as the model's weights used in this paper are available at: https://github.com/jerome-lux/RAMSES2.

Acknowledgement: We wish to thank Spie Batignolles Malet for the technical assistance and for providing the necessary materials. We also acknowledge Dr. JD Lau Hiu Hoong for his major contribution to the dataset construction.

Conflict of interest: The authors declare no conflict of interest.

AI use statement: During the preparation of this work, the authors used Gemini for French-to-English translation and linguistic refinement.

References

1. de Larrard F, Colina H (editors). Concrete Recycling: Research and Practice. CRC Press; 2019. doi: 10.1201/9781351052825
2. Wang B, Yan L, Fu Q, et al. A Comprehensive Review on Recycled Aggregate and Recycled Aggregate Concrete.

Resources, Conservation and Recycling. 2021; 171: 105565. doi: 10.1016/j.resconrec.2021.105565
3. EN 933-11:2009. Tests for Geometrical Properties of Aggregates—Part 11: Classification Test for the Constituents of Coarse Recycled Aggregate. 2009.
4. He K, Gkioxari G, Dollár P, et al. Mask R-CNN. In: Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV); 22–29 October 2017; Venice, Italy. pp. 2980–2988. doi: 10.1109/ICCV.2017.322
5. Wang X, Zhang R, Kong T, et al. SOLOv2: Dynamic and Fast Instance Segmentation. In: Larochelle H, Ranzato M, Hadsell R, et al. (editors). NIPS '20: Proceedings of the 34th International Conference on Advances in Neural Information Processing Systems, Proceedings of the 34th Conference on Neural Information Processing Systems (NeurIPS 2020); 6–12 December 2020; Vancouver, BC, Canada. Curran Associates, Inc.; 2020. pp. 17721–17732.
6. Bolya D, Zhou C, Xiao F, et al. YOLACT++: Better Real-time Instance Segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2020; 42(2): 1108–1121. doi: 10.1109/TPAMI.2020.3014297
7. Terven J, Córdova-Esparza DM, Romero-González JA. A Comprehensive Review of YOLO Architectures in Computer Vision: From YOLOv1 to YOLOv8 and YOLO-NAS. Machine Learning and Knowledge Extraction. 2023; 5(4): 1680–1716. doi: 10.3390/make5040083
8. Li F, Zhang H, Xu H, et al. Mask DINO: Towards A Unified Transformer-based Framework for Object Detection and Segmentation. In: Proceedinhs of the 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). 17–24 June 2023; Vancouver, BC, Canada. pp. 3041–3050. doi: 10.1109/CVPR52729.2023.00297
9. Cheng B, Misra I, Schwing AG, et al. Masked-attention Mask Transformer for Universal Image Segmentation. In: Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 18–24 June 2022; New Orleans, LA, USA. pp. 1280–1289. doi: 10.1109/CVPR52688.2022.00135
10. Carion N, Massa F, Synnaeve G, et al. End-to-End Object Detection with Transformers. In: Vedaldi A, Bischof H, Brox T, et al. (editors). Computer Vision—ECCV 2020, Proceedings of the European Conference on Computer Vision; 23–28 August 2020; Glasgow, UK. Springer International Publishing; 2020. 12346, pp. 213–229. doi: 10.1007/978-3-030-58452-8_13
11. Guo R, Niu D, Qu L, et al. SOTR: Segmenting Objects with Transformers. In: Proceedings of the 2021 IEEE/CVF International Conference on Computer Vision (ICCV); 10–17 October 2021; Montreal, QC, Canada. pp. 7137–7146. doi: 10.1109/ICCV48922.2021.00707
12. Gao Y, Wang J, Xu X. Machine Learning in Construction and Demolition Waste Management: Progress, Challenges, and Future Directions. Automation in Construction. 2024; 162: 105380. doi: 10.1016/j.autcon.2024.105380
13. Demetriou D, Mavromatidis P, Petrou MF, et al. CODD: A Benchmark Dataset for the Automated Sorting of Construction and Demolition Waste. Waste Management. 2024; 178: 35–45. doi: 10.1016/j.wasman.2024.02.017
14. Demetriou D, Mavromatidis P, Robert PM, et al. Real-time Construction Demolition Waste Detection Using State-of-the-art Deep Learning Methods; Single-stage vs Two-stage Detectors. Waste Management. 2023; 167: 194–203. doi: 10.1016/j.wasman.2023.05.039
15. Zhou Q, Liu H, Qiu Y, et al. Object Detection for Construction Waste Based on an Improved YOLOv5 Model. Sustainability. 2023; 15: 681. doi: 10.3390/su15010681
16. Serranti S, Palmieri R, Bonifazi G, et al. An Automated Classification of Recycled Aggregates for the Evaluation of Product Standard Compliance. Sustainability. 2023; 15: 2009. doi: 10.3390/su152015009
17. Hamdan M, Rover D, Darr M, et al. Mass Estimation from Images Using Deep Neural Network and Sparse Ground Truth. In: Proceedings of the 2019 18th IEEE International Conference on Machine Learning and Applications (ICMLA); 16–19 December 2019; Boca Raton, FL, USA. pp. 1987–1992. doi: 10.1109/ICMLA.2019.00318
18. Miura Y, Sawamura Y, Shinomiya Y, et al. Vegetable Mass Estimation Based on Monocular Camera Using Convolutional Neural Network. In: Proceedings of the 2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC); 11–14 October 2020; Toronto, ON, Canada. pp. 2106–2112. doi: 10.1109/SMC42975.2020.9282930
19. Dohmen R, Catal C, Liu Q. Image-based Body Mass Prediction of Heifers Using Deep Neural Networks. Biosystems Engineering. 2021; 204: 283–293. doi: 10.1016/j.biosystemseng.2021.02.001
20. Standley T, Sener O, Chen D, et al. image2mass: Estimating the Mass of an Object from Its Image. In: Levine S, Vanhoucke V, Goldberg K, et al. (editors). Proceedings of the 1st Annual Conference on Robot Learning, Proceedings of the 1st Conference on Robot Learning (CoRL 2017); 13–15 November 2017; Mountain View, CA, USA. PMLR; 2017. 78, pp. 324–333.
21. Lux J, Lau Hiu Hoong JD, Mahieux PY, et al. Classification and Estimation of the Mass Composition of Recycled Aggregates by Deep Neural Networks. Computers in Industry. 2023; 148: 103889. doi:

10.1016/j.compind.2023.103889
22. He K, Zhang X, Ren S, et al. Identity Mappings in Deep Residual Networks. In: Leibe B, Matas J, Sebe N, et al. (editors). Computer Vision—ECCV 2016, Proceedings of the 14th European Conference, October 11–14, 2016; Amsterdam, The Netherlands. Springer International Publishing; 2016. 9905, pp. 630–645.
23. Gu W, Bai S, Kong L. A Review on 2D Instance Segmentation Based on Deep Neural Networks. Image and Vision Computing. 2022; 120: 104401. doi: 10.1016/j.imavis.2022.104401
24. Lee Y, Park J. CenterMask: Real-Time Anchor-Free Instance Segmentation. In: Proceedings of the 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 13–19 June 2022; Seattle, WA, USA. pp. 13903–13912. doi: 10.1109/CVPR42600.2020.01392
25. Chen H, Sun K, Tian Z, et al. BlendMask: Top-Down Meets Bottom-Up for Instance Segmentation. arXiv preprint. 2020. doi: 10.48550/arXiv.2001.00309
26. Tian Z, Shen C, Chen H. Conditional Convolutions for Instance Segmentation. In: Vedaldi A, Bischof H, Brox T, et al. (editors). Computer Vision—ECCV 2020, Proceedings of the European Conference on Computer Vision; 23–28 August 2020; Glasgow, UK. Springer International Publishing; 2020. pp. 282–298.
27. Tian Z, Shen C, Chen H, et al. FCOS: Fully Convolutional One-Stage Object Detection. In: Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV); 27 October 2019–2 November 2019; Seoul, South Korea. pp. 9626–9635. doi: 10.1109/ICCV.2019.00972
28. Lau Hiu Hoong JD, Lux J, Mahieux PY, et al. Determination of the Composition of Recycled Aggregates Using a Deep Learning-based Image Analysis. Automation in Construction. 2020; 116: 103204. doi: 10.1016/j.autcon.2020.103204
29. Kirillov A, Mintun E, Ravi N, et al. Segment Anything. In: Proceedings of the 2023 IEEE/CVF International Conference on Computer Vision (ICCV); 1–6 October 2023; Paris, France. pp. 3992–4003. doi: 10.1109/ICCV51070.2023.00371
30. Cvat.ai/cvat. Available online: https://github.com/cvat-ai/cvat (accessed on 11 February 2026).
31. Woo S, Debnath S, Hu R, et al. ConvNeXt V2: Co-designing and Scaling ConvNets with Masked Autoencoders. In: Proceedings of the 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR); 17–24 June 2023; Vancouver, BC, Canada. pp. 16133–16142. doi: 10.1109/CVPR52729.2023.01548
32. Lin TY, Goyal P, Girshick R, et al. Focal Loss for Dense Object Detection. In: Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV); 22–29 October 2017; Venice, Italy. pp. 2999–3007. doi: 10.1109/ICCV.2017.324
33. Loshchilov I, Hutter F. Decoupled Weight Decay Regularization. arXiv preprint. 2019. doi: 10.48550/arXiv.1711.05101
34. Padilla R, Passos, WL, Dias, TL, et al. A Comparative Analysis of Object Detection Metrics With a Companion Open-Source Toolkit. Electronics. 10(3): 279. doi: 10.3390/electronics10030279
35. Lin TY, Dollar P, Girshick R, et al. Feature Pyramid Networks for Object Detection. In: Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR); 21–26 July 2017; pp. 936–944. doi: 10.1109/CVPR.2017.106

Appendix A. Implementation details

The backbone is a ConvNextv2_femto backbone (see https://github.com/facebookresearch/ConvNeXt-V2). We chose this lightweight model because other models with more parameters, while performing better on the training set, led to greater overfitting.

The FPN [35] and SOLOv2 architecture [5] are the same as in the paper (with the exception of the number of filters used to upscale the FPN level to create the Unified Mask Representation, which is reduced to 128). The final segmentation mask feature map has, however, 256 filters.

The new instance mass factor shared head has only two convolutional layer with 256 filters, as more did not seem to increase the performance (4 layers decreased the performance, both in the two-factor and one-factor models). The τ factor feature map is obtained from the Unified Mask Representation after 4 convolutions with 128 filters.

The model has 15.3 M Parameters, which is quite light for an instance segmentation model.

The grid number and scale ranges used in the SOLOv2 module are given in **Table A1** for the model with input size

of 2,048 × 3,072. These values were chosen to guarantee that the smallest instances in each level are represented by at least one grid cell within their reduced target box (by a factor of 0.75).

Table A1. Grid numbers and instance scale ranges (in pixels of the input image) for each level of the FPN.

FPN level	Grid number	Min size (pixels)	Max size (pixels)
P3	144	-	128
P4	72	64	256
P5	36	128	512
P6	18	256	-

Appendix B. Sensitivity study

In this section, we evaluate the model's sensitivity to hyperparameter selection by plotting boxplots for the mAP@0.5, the average per-instance MAPE, and the total predicted mass absolute percentage error.

The analysis assesses a total of 48 parameter combinations, defined by the following hyperparameter ranges:

- Class Score threshold: {0.25, 0.5, 0.75}.
- Mask threshold: {0.5, 0.6, 0.7, 0.75}.
- NMS threshold: {0.25, 0.3, 0.35, 0.4}.

The resulting boxplots in **Figure A1** reveal a low sensitivity to these variations, demonstrating the model's inherent robustness within the tested range. **Table A2** summarizes the mean and standard deviation for these metrics.

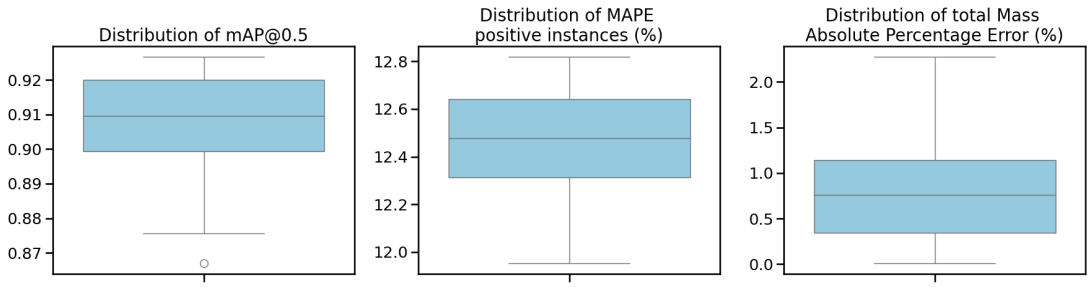


Figure A1. Distribution of performance metrics (mAP, MAPE, mass error) over 48 hyperparameter settings.

Table A2. Mean and standard deviation values for the mAP, MAPE and Total Mass Relative error obtained during the sensitivity study.

Metric	Mean	Standard deviation
mAP@0.5	0.908	0.014
MAPE (positive instances) (%)	12.5	0.23
Total Mass APE (%)	0.8	0.5

Based on this analysis, we identified the optimal configuration as a class score threshold of 0.5, a mask threshold of 0.6, and an NMS threshold of 0.35.

Appendix C. Ablation study

A distinctive feature of the proposed architecture is that mass is calculated as the product of an instance-specific factor derived from the shared heads and locally defined factors computed from the high-resolution representation, which aggregates feature maps across multiple scales. This provides the model with greater flexibility to account for high-resolution geometric features. To demonstrate the efficacy of this approach, we compared it against a mass prediction performed solely at the shared head level. In this one-factor model, the predicted factor δ is multiplied by the area

of the mask of the corresponding instance to get the final normalized mass (this is equivalent to a constant τ factor). Directly predicting normalized mass (without multiplying by the mask area) is impractical, as values span several orders of magnitude, ranging from 1.9×10^{-3} to 51×10^{3} with a mean of 152. **Table A3** compares the aggregate-level performance. The modulation provided by the variable τ factor yields a systematic improvement in mass prediction across all classes, with an average MAPE reduction of 2% points. This performance gain is more pronounced in the validation MAPE monitored during training, where the MAPE decreased by 3 pp (from 18.7% to 15.7%). Although this improvement is modest, it is consistent across all categories, with specific classes exhibiting performance gains of several percentage points; we consider this a non-negligible contribution to the model's overall robustness.

Table A3. Comparison between the two-factor and one-factor models for an IoU threshold of 0.5.

	Mass MAPE @0.5 (%)		**Diff (pp)**
Class	**Two-factor model**	**One-factor model**	
Ra	34.7%	35.7%	1.06
Rc	7.9%	9.0%	1.16
Rb01	7.7%	9.0%	1.38
Rb02	9.5%	10.6%	1.14
Ru01	5.7%	7.7%	2.01
Ru02	6.4%	8.5%	2.05
Ru04	7.7%	9.1%	1.37
Ru05	6.6%	7.5%	0.89
Ru06	8.9%	10.7%	1.80
X01	27.3%	34.3%	6.99
All	12.0%	13.9%	1.97