Sparsifying instance segmentation models for efficient vision-based industrial recycling

Melanie Neubauer[1] (✉), Ozan Özdenizci[2], Justus Piater[3], and Elmar Rueckert[1]

[1] Chair of Cyber-Physical-Systems, Technical University of Leoben, Austria
[2] Institute of Machine Learning and Neural Computation, Graz University of Technology, Austria
[3] Department of Computer Science, University of Innsbruck, Austria
{melanie.neubauer,rueckert}@unileoben.ac.at,
oezdenizci@tugraz.at,justus.piater@uibk.ac.at

Abstract. Recycling is essential to the circular economy. However, efficient material sorting, particularly in steel scrap recycling, remains challenging due to material diversity and contamination. Visual computing via deep learning offers a significant promise in automation, with models such as YOLO and Mask R-CNN excelling in object detection and segmentation. However, high computational requirements often limit industrial deployment, which necessitates more efficient algorithmic solutions targeted for such applied machine learning problems. We introduce a novel approach to prune large image segmentation models based on *instance-based importance scores (IBIS)*, specifically tailored to the problem of instance segmentation for automated steel scrap recycling. Our method identifies and prunes low priority parameters by leveraging parameter importance scores estimated by considering the presence of recyclable instances to be segmented in the frames. Moreover, we utilize a novel custom dataset constructed for the instance segmentation task during copper and steel scrap recycling, which involves recyclable objects of different sizes with various levels of difficulty. Our evaluations demonstrate promising computational efficiency gains without significant performance drops, while also enabling powerful out-of-distribution generalization, a game-changing capability. Finally, we discuss the potential of our work for real-world industrial applications, enabling resource-efficient deep learning deployment in large-scale automated sorting systems.

Keywords: instance segmentation · steel scrap recycling · neural network pruning · sparsity · out-of-distribution generalization.

1 Introduction

As the European Union (EU) advances toward its goal of becoming a sustainable, climate-neutral economy by 2050 under the European Green Deal [5], pressure is mounting on energy-intensive industries like steel manufacturing to adopt more environment friendly practices. Steel production remains one of the most

carbon-intensive industrial processes globally, contributing to 5.7% of total EU emissions [33]. Recycling plays a pivotal role in reducing the environmental impact of such industries, yet effective material sorting especially in complex environments like steel scrap recycling remains a challenge. Traditional sorting relies heavily on manual labor, which is slow, costly, and error-prone, making it unsuitable for large-scale, real-time operations. To address these challenges, automated solutions based on machine learning and computer vision have become increasingly important [6,32]. These technologies offer significant improvements in sorting efficiency and scalability, especially for the diverse and often contaminated materials found in recycling streams. Real-time image segmentation and object detection models are at the core of these systems, enabling faster and more accurate classification of materials. However, effective deployment of these models in real-world industrial settings often require significant computational efficiency improvements in terms of memory and energy requirements.

In the field of recycling, deep learning based models, particularly in image segmentation and object detection, have proven effective in automating material classification [7]. Deep neural network architectures such as YOLO (You Only Look Once) [29] and Mask R-CNN [13] are commonly used for real-time detection due to their inference speed and accuracy, making them suitable for various recycling applications, including waste sorting and steel scrap classification. These models have shown promise in differentiating between different types of materials, such as plastics, metals, and paper [3], thereby enabling automated separation. Among these, YOLO variants are often regarded as state-of-the-art for real-time applications in recycling, due to their strong downstream task performances. However, these models also pose significant computational challenges, particularly in industrial settings where real-time processing often requires handling 50-100 frames per second, each involving 100-10.000 objects. Furthermore, high memory consumption and increased inference times limits their deployment on edge devices or in resource-constrained environments. To address these issues, efficient solutions must balance high performance with reduced computational and memory demands for large-scale automated recycling.

Model sparsification based on weight pruning is a widely-studied approach for reducing the memory usage and computational load of deep neural networks [15]. Pruning involves removing less important parameters from a pre-trained model, leading to smaller, more efficient networks with minimal sacrifice in accuracy [11]. Particularly in semantic segmentation tasks, pruning techniques have been successfully applied to U-Net type models to reduce model size, computational complexity, and memory requirements while maintaining high performance [21]. We present a novel neural network pruning criterion that utilizes **instance-based importance scores (IBIS)**, to prune YOLO-based steel scrap industrial recycling segmentation models. Our method harnesses parameter gradients from an instance-based strategy, allowing us to identify and remove less critical parameters while preserving essential features needed for accurate scrap classification. We conduct extensive experiments on a novel dataset specifically designed for

industrial copper and steel scrap recycling applications, evaluating feasibility of our method in real-world industrial settings. Our contributions are as follows:

- We present a model pruning criterion that utilizes *instance-based importance scores (IBIS)* to prune YOLO-based steel scrap segmentation models, and significantly reduce model size while maintaining high performance.
- We introduce a novel dataset constructed for instance segmentation during copper and steel scrap recycling in a real-world industrial setting, which involves recyclable objects of different sizes with hierarchical task difficulty.
- We empirically show that our approach enhances computational efficiency with up to 95% reduction in model size, while maintaining high performance. Moreover, we demonstrate strong out-of-distribution generalization capabilities of our approach, with enhanced robustness to different scrap material sizes observed for the first time during inference.

2 Related Work

We present an overview of key advancements in real-time image segmentation, object detection, and neural network pruning, with a specific emphasis on their applications in the recycling industry and steel scrap classification.

2.1 Real-Time Image Segmentation & Object Detection

Real-time image segmentation and object detection are crucial tasks in computer vision, with applications ranging from autonomous driving to medical imaging and industrial automation like in the recycling industry. In the context of recycling, these techniques enable efficient material classification and sorting. Traditional object detection methods, such as R-CNN [9] and Fast R-CNN [8], achieved strong accuracy but were computationally expensive, limiting their real-time applicability. The introduction of Mask R-CNN [13] improved instance segmentation by generating precise pixel-wise object masks. However, these models remained slow and required significant computational resources.

To address the speed limitations of these models, the YOLO (You Only Look Once) [29] series was developed, significantly improving detection efficiency while maintaining high accuracy. Recent versions, such as YOLOv11 [16], further enhance performance by integrating object segmentation, detection, and classification into a single framework, making it suitable for recycling applications. Despite these advancements, deploying these models in industrial settings remains challenging due to resource constraints and real-time processing demands.

2.2 Visual Computing in Steel Scrap Recycling

The use of deep learning for scrap material classification has gained traction in recent years. Previous studies [34] have demonstrated the effectiveness of convolutional neural networks (CNNs) in intelligent waste recognition, leading to improvements in classification, sorting, and recycling efficiency. Some approaches,

such as the system introduced in [35], leverage machine vision for steel scrap quality inspection, while others, like ConvoWaste [26], apply image processing techniques to classify various waste types. However, several challenges hinder the development of robust machine learning solutions for steel scrap classification. Unlike well-researched categories such as vehicles or human faces, steel scrap remains an underexplored domain with limited publicly available datasets [30]. Existing datasets focus on landfill waste or non-shredded scrap [2], making them less suitable for training deep learning models tailored to shredded steel scrap [27].

Additionally, variations in shredded scrap output across different industrial shredders introduce further complexities [1]. The configuration, blade design, and operational parameters of each shredder influence the final output, making it difficult to create standardized datasets that generalize across different recycling plants. These factors highlight the need for adaptable and efficient machine learning models capable of handling diverse scrap materials.

2.3 Neural Network Pruning

Pruning neural networks is a widely explored technique for reducing model complexity while preserving performance [11]. Seminal works, such as Optimal Brain Damage [19] and Optimal Brain Surgeon [12], introduced structured pruning strategies by identifying and removing less critical weights. State-of-the-art methods, such as global magnitude-based pruning (GMP) [11], single-shot network pruning (SNIP) [20], pruning considering pre-training (PCPT) [18] and Taylor expansion criterion based pruning [24], refine this concept by estimating the *importance* of individual weights or filters and eliminating redundant components in a single-shot. Recent works extended these with novel criteria on how importance scores are derived [4,10,28,31].

In segmentation tasks, pruning has been successfully applied to models like U-Net [21] to achieve significant reductions in model size and floating point operations (FLOPs) while maintaining accuracy. Filter pruning techniques in CNN-based segmentation models have shown promising results in reducing computational complexity without degrading segmentation performance. In [22] they focus on dynamic pruning in region-merging-based segmentation, significantly reducing computational complexity while maintaining segmentation accuracy, enabling large-scale applications in remote sensing. Another work [14] introduces context-aware pruning for deep neural networks, leveraging inter-channel dependencies to sparsify models while preserving performance, demonstrating effectiveness across various segmentation architectures. Despite these advancements, existing pruning methods often overlook task-specific importance measures, particularly in industrial recycling applications. We introduce a novel pruning concept based on instance-based importance scores, tailored to instance segmentation models (e.g., YOLO [16]) for steel scrap classification.

Table 1: Detailed breakdown of the steel and copper scrap material allocation in the prepared datasets, including the material sizes, weights and quantities.

	Material Size	Weight [kg]	Quantity	Material Allocation		
				Dataset 1	Dataset 2	Dataset 3
Steel Scrap	Large	9	45	✓	✓	✓
	Medium-Large	20	95	✓	✓	✓
	Medium	9	72		✓	✓
	Small-Medium	6	80			✓
	Small	2.5	104			✓
Copper Scrap	Large	15	17	✓	✓	✓
	Medium	8.5	47		✓	✓
	Small	1	37			✓
	Total	**46.5**	**497**			

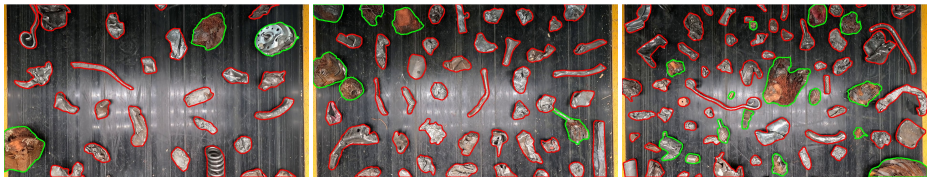
3 Industrial Steel Scrap Recycling Dataset

We focus on the recycling process of steel scrap derived from end-of-life vehicles and electrical appliances, which are shredded to produce the E40 scrap fraction [25]. Despite initial pre-sorting using magnetic separators, the resulting material still contains a significant proportion of unwanted contaminants, like copper objects. We address real time detection and segmentation of these undesirable particles with deep learning based instance segmentation models.

Due to the absence of publicly available labeled segmentation datasets for steel scrap, we developed a custom dataset tailored to our application. Specifically, our dataset encompasses a diverse range of objects varying in size, with a particular focus on components containing copper, such as cables and embedded wiring. The collected material is classified into two primary categories: steel scrap (considered acceptable) and copper scrap (considered contaminants). Steel scrap consists of particles that are entirely free from copper inclusions, while copper scrap includes both visibly contaminated pieces and those with concealed copper elements. The dataset composition and material quantities used in our simulations are detailed in Table 1. To maintain a hierarchically structured dataset and facilitate effective model training, the scrap particles were carefully sorted by size, ensuring balanced representation and manageable data complexity.

3.1 Data Recording

We collected three datasets by parsing video recordings of a conveyor belt from top view, where materials were manually positioned to ensure controlled conditions. Each dataset was recorded over five iterations, with slight adjustments to the positioning of the particles in each run. To enhance variability and generalization, we maintained a consistent distance between particles, while capturing



(a) Sample from Dataset 1 (b) Sample from Dataset 2 (c) Sample from Dataset 3

Fig. 1: Materials represented in the datasets vary in terms of object size, quantity, and the spacing between them. Objects segmented in red belong to the steel scrap objects category, while those segmented in green represent copper scrap objects.

them from different angles using a GoPro Hero 11 camera configured with optimized settings. The camera recorded at 100 frames per second, ensuring smooth and detailed footage. The conveyor belt operated at a speed of approximately 0.4 to 0.5 m/sec during recordings. A shutter speed of 1/100 was selected to balance exposure and motion clarity, while the sharpness setting was adjusted to high to enhance the visibility of fine details. These settings were chosen to maximize image quality and facilitate accurate segmentation and classification.

We extracted individual frames from these videos to create our datasets. To ensure sufficient variation between frames while avoiding excessive redundancy, we selected every second frame from the recordings. The full dataset was then labeled by human experts using an automated segmentation tool [27] to ensure accuracy and reliability. The complexity of the datasets varies based on factors such as the number, size, and spatial distribution of the particles. Our datasets (shown in Figure 1) were designed to support both model training and evaluation. Dataset 1 comprises 5,694 images, which were split into three subsets: 67% (3,831 images) for training, 16% (907 images) for validation, and 17% (956 images) for testing. In contrast, Dataset 2 (1,187 images) and Dataset 3 (1,599 images) were exclusively designated as test sets to assess generalization capabilities. This partitioning strategy ensures rigorous evaluations, with Dataset 1 serving as both training/validation and testing, while Datasets 2 and 3 provide insights into the model's performance on unseen data.

Table 1 provides an overview of the dataset composition and material distribution. Dataset 1 primarily features larger steel and copper scrap particles, with additional variations introduced in subsequent datasets. The datasets also differ in object density per frame, reflecting increasing complexity. Dataset 1 contains approximately 30-40 objects per frame, ensuring a structured yet moderately challenging environment for initial training and validation. Dataset 2 increases the density to 40-50 objects per frame, introducing greater variation in object positioning while maintaining a manageable level of overlap. Dataset 3 presents the highest complexity, with up to 70 objects per frame, better representing real-world industrial scenarios with a diverse mix of particle sizes and spacing.

(a) Example input frame x with various instances to be segmented, used to compute $\nabla_{w_j}\mathcal{L}(f(x,\boldsymbol{w}),y)$ for each parameter.

(b) Example image $\tilde{x}$ without instances to be segmented, used to compute $\nabla_{w_j}\mathcal{L}(f(\tilde{x},\boldsymbol{w}),\mathbf{0})$ for each parameter.

Fig. 2: Example image frames from our training set $\mathcal{D}_{\text{train}}$, which is used to estimate the importance scores necessary to prune our neural network models.

4 Pruning Segmentation Models via Instance-based Importance Scores (IBIS)

We propose a novel pruning method, instance-based importance scores (IBIS), that leverages gradient information from multiple sources to evaluate the significance of model weights. We calculate averaged gradients across standard training images (as in Figure 2a), while also incorporating gradients obtained from images without any objects or labels (Figure 2b). By analyzing the difference between these two gradient distributions, we derive an importance score that effectively identifies less critical weights. This strategy offers a more refined assessment of weight importance, leading to an efficient and effective pruning process.

Importance Score Calculation: Given a model $f(x;\boldsymbol{w})$ and a training set $\mathcal{D}_{\text{train}}$, we denote the training set as consisting of pairs $\{(x_i,y_i)\}_{i=1}^{N}$, where x_i is the input image, y_i is the corresponding label, and N is the total number of training samples. In neural network pruning, we aim to optimize a global binary pruning mask $\boldsymbol{m}\in\{0,1\}^n$, with n being the number of parameters. We obtain a new model $f(x;\boldsymbol{m}\odot\boldsymbol{w}')$ with $||\boldsymbol{m}\odot\boldsymbol{w}'||_0 \leq (1-\mathcal{R})\cdot n$, where the symbol $\odot$ denotes element-wise multiplication and $\mathcal{R}\in[0,1)$ represents the pruning ratio (e.g., when $\mathcal{R}=0.85$, our model sparsity is 85%). The finetuning process updates the model parameters from $\boldsymbol{w}$ to $\boldsymbol{w}'$. We define a non-binary, continuous-valued parameter importance vector s and determine the binary mask values as: $m_j = \mathbb{1}\left[s_j-\tilde{s}_\gamma\right],\ \forall\, j\in\{1,...n\}$, where $\gamma=(1-\mathcal{R})\cdot n$ and $\tilde{s}=\text{SortDescending}(s)$, so that $\tilde{s}_\gamma$ is the γ-largest element in s, and $\mathbb{1}[.]$ denotes the indicator function. For a selected model instance j, an importance score s_j^{IBIS} is computed as follows,

$$s_j^{\text{IBIS}} = |w_j|\cdot\left[1-\exp\left(-c\right)\right], \quad \text{where} \tag{1}$$

$$c = \left|\mathbb{E}_{(x,y)\sim\mathcal{D}_{\text{train}}}\left[\nabla_{w_j}\mathcal{L}(f(x;\boldsymbol{w}),y)\right] - \nabla_{w_j}\mathcal{L}(f(\tilde{x};\boldsymbol{w}),\mathbf{0})\right|. \tag{2}$$

is the variable which denotes the absolute difference between the two gradients. Here, $\tilde{x}$ represents a background image (shown in Figure 2b), and the absence of segmentation labels is denoted by $\mathbf{0}$.

Design Intuition: Here, the term $1-\exp(-c)$ controls how much of the weight magnitude $|w_j|$ contributes to the parameter importance score, with $c \in [0, \infty)$ acting as a tuning parameter that determines the degree of attenuation. This tuning parameter enforces the importance scores to be bounded within $[0, |w_j|)$, where larger values of c (indicating greater differences between the two gradients) result in a stronger influence of parameter magnitudes on the score s_j^{IBIS}. To the contrary, for small values of c (i.e., very little difference between the gradient terms regardless of the presence of instance to be segmented), the parameter importance score would approach zero. This formulation differs from Global Magnitude Pruning (GMP), which ranks weights solely by their absolute magnitudes, pruning the smallest ones without considering gradient-based significance. Unlike SNIP, which multiplies weights by their gradients without distinguishing whether the gradients are influenced by objects in the image, our approach leverages c, capturing the difference between the standard gradient and the gradient from an image without objects (i.e., without labels). This results in a more refined pruning criterion that prioritizes weights based on their relevance to object regions rather than treating all gradients uniformly.

In our implementation of IBIS, we apply the pruning ratio $\mathcal{R}$ globally to the model, thus the per-layer unstructured sparsity rates can vary. It is important to note that our pruning method has only a single hyperparameter, the pruning ratio $\mathcal{R}$, which simplifies the utility of the method.

5 Experimental Setup

We evaluate the effectiveness of our pruning criterion by comparing it to state-of-the-art importance score estimation techniques. For each method, we prune the same pre-trained baseline YOLO11n-seg model [16] and finetune for the same duration, then assess performance on an unseen test set.

5.1 Baseline Pruning Methods

Global Magnitude Pruning (GMP) [11] is a common pruning technique. Its main goal is to shrink the model by eliminating less critical parameters, like weights, while maintaining its performance. MP calculates the importance of each parameter by evaluating its magnitude, and it prunes the parameters with the smallest values, which are considered less influential to the model's overall performance. The importance score for GMP pruning is computed as:

$$s_j^{\text{GMP}} = |w_j|\,, \tag{3}$$

where, in contrast to IBIS, only the absolute magnitudes of the weights are considered, ignoring the gradient information.

Single-shot Network Pruning (SNIP) [20] is a technique that simplifies the process by identifying and eliminating less important connections in a single step. SNIP assesses the significance of each weight by calculating the gradient of the loss with respect to the weight. This score assesses the importance of each weight by computing the product of its magnitude and the expected gradient of the loss and prunes those that have the least impact on the overall network performance, via the score:

$$s_j^{\text{SNIP}} = \left| w_j \cdot \mathbb{E}_{(x,y)\sim\mathcal{D}_{\text{train}}} \left[\nabla_{w_j} \mathcal{L}(f(x; \boldsymbol{w}), y) \right] \right|. \tag{4}$$

Here, differently than the GMP importance score, both the weights and the gradients are utilized for calculating the score.

Pruning Considering Pre-Training (PCPT) [18] is a pruning method that distinguishes between two types of parameters: (a) stable, large-value parameters that change little during fine-tuning, and (b) unstable, small-value parameters that change chaotically. Parameters of type (a) are pruned based on their magnitude, while type (b) parameters are pruned using the SNIP method, which captures changes due to downstream task optimization. The score defined as:

$$s_j^{\text{PCPT}} = \left| w_j \cdot \mathbb{E}_{(x,y)\sim\mathcal{D}_{\text{train}}} \left[\nabla_{w_j} \mathcal{L}(f(x; \boldsymbol{w}), y) \right] \right| + \alpha \cdot w_j^2, \tag{5}$$

depicts that PCPT extends the original SNIP importance score by introducing a parameter α, which is multiplied by the square of the weights.

5.2 Model Training Configurations

For the initial training, we used the following key configurations. The model architecture was based on the *yolo11n-seg.yaml* file from the Ultralytics [17] library. The training process was set to run for 20 epochs with a batch size of 32 and an image size of 640x640 pixels. A single NVIDIA GeForce RTX 4090 GPU was utilized for simulations. The optimizer used was Adam, with a learning rate of 0.01, and weight decay applied at 0.0005 to reduce the risk of overfitting. The training included a warmup period of 3 epochs, where the learning rate increased gradually, and the momentum started at 0.8. To optimize training efficiency, automatic mixed precision [23] was enabled, allowing for faster computations while maintaining model accuracy. The training also involved validation, using a separate validation set from our dataset. These configurations were selected to balance model performance and training efficiency. For the pruning baseline method PCPT, we selected $\alpha = 0.001$ through cross-validation.

Fine-tuning the model weights was conducted over 10 epochs, with early stopping was activated 2 epochs of no improvement. During this phase, we used a learning rate of 0.001 as opposed to the initial training configuration. Both the training and validation sets were utilized for fine-tuning, while the test set was reserved for evaluation at the end of the process to assess the final model performance. In our experiments, the sparsity ratio for $\mathcal{R}$ concerns the convolutional layers. The total network sparsity can slightly vary in our simulations.

Table 2: Comparison of object segmentation performance mAP50-95 (%) across different sparsity levels with various pruning criteria. Accuracies are averaged over 3 random seeds. Values in parentheses indicate standard deviations.

	#params	GFLOPs	Segmentation mAP50-95 (%)				
			Random	GMP [11]	SNIP [20]	PCPT [18]	**IBIS (Ours)**
Dense	2 835 153	10.36	76.80 (0.0)				
Sparse (25%)	2 104 061	9.68	13.87 (14.3)	74.50 (0.6)	74.57 (0.4)	74.60 (0.4)	**74.60** (0.7)
Sparse (50%)	1 402 138	9.08	15.60 (14.3)	**74.77** (0.4)	74.23 (1.4)	74.50 (0.4)	74.43 (0.5)
Sparse (75%)	699 097	6.65	0.43 (0.4)	75.03 (0.9)	74.97 (0.6)	75.37 (1.0)	**76.10** (1.0)
Sparse (85%)	421 458	4.76	11.83 (20.2)	74.9 (1.4)	76.07 (2.0)	76.63 (1.7)	**76.73** (1.9)
Sparse (90%)	281 036	3.62	0.47 (0.4)	75.13 (1.5)	74.50 (1.5)	74.07 (1.2)	**77.43** (1.4)
Sparse (95%)	140 457	2.37	0.03 (0.1)	75.97 (0.7)	73.63 (0.8)	73.77 (1.0)	**76.23** (1.4)

5.3 Evaluation Metrics

We assess performance and efficiency in the context of segmentation and bounding box tasks mainly based on the mAP50-95 (Mean Average Precision at Intersection over Union (IoU) thresholds from 50% to 90%) metrics [17].

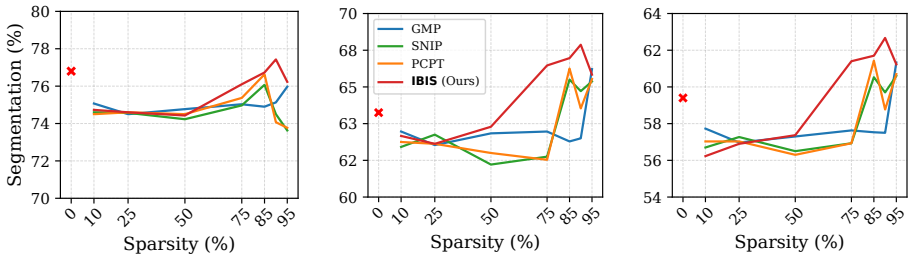
Segmentation mAP50-95: This metric evaluates segmentation quality across IoU thresholds (50-95%). The model optimizes segmentation performance using multiple losses, including Mask Loss for mask alignment, binary cross-entropy loss for pixel-wise classification, and dice loss for improved mask overlap. Additional consistency losses may enhance smoothness in predictions [16].

Bounding Box mAP50-95: This metric evaluates object detection performance across IoU thresholds from 50% to 95%. The model optimizes detection using multiple losses, including classification loss, Bounding Box Loss, IoU Loss and objectness loss, which collectively enhance detection accuracy [16].

Efficiency Metrics: In addition to accuracy, the number of non-zero parameters (#params) is tracked to assess the model's size and complexity. This metric provides insight into storage and computational requirements, which is crucial for deploying the model on resource-constrained devices. We also estimate the number of FLOPs (floating point operations) to evaluate the computational complexity of the model during forward pass operations at inference time.

6 Experimental Results

In this section, we present the results of applying our pruning method. After pruning, all models undergo the same fine-tuning phase, during which they are trained on a combined training and validation set to recover any performance



(a) Testing on Dataset 1 (b) Testing on Dataset 2 (c) Testing on Dataset 3

Fig. 3: Segmentation mAP50-95 (%) across different sparsity levels and pruning methods. Average over three seeds with increasing complexity of the validation set. The red cross marks the evaluation result of the dense model.

loss introduced by the pruning process. Subsequently, we evaluated the models on the test set to assess their generalization performance.

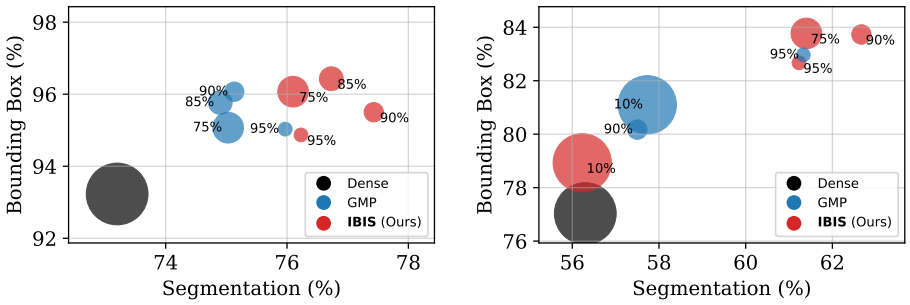
We compare our method to existing approaches, followed by an evaluation of its performance on out-of-distribution segmentation tasks. Next, we investigate the trade-off between model performance and size. Finally, we conduct an ablation study to evaluate the effectiveness of different instance-based methods.

6.1 Comparisons with Existing Methods

Table 2 presents a comparison of segmentation performance (mAP50-90) across various sparsity levels and pruning methods, including GMP, SNIP, and our proposed approach, IBIS. The table illustrates the reduction in model parameters and computational cost (GFLOPs) as the sparsity level increases, with results are averaged over three random seeds. At 0% sparsity (dense model), the model achieves a high mAP50-90 of 76.80% (first row in Table 2). As the sparsity increases from 25% to 95%, the parameter count decreases significantly, and the performance varies across different pruning methods. Notably, our method, IBIS, consistently outperforms the other methods for sparsity levels greater than or equal to 75%, albeit by a small margin. While no clear second-best method emerges, IBIS remains the top choice overall. Furthermore, the table compares the GFLOPs for each configuration, showing the reduction in computational cost as the model sparsity increases. Despite the decrease in parameters, IBIS maintains high accuracy, demonstrating superior performance under high sparsity.

6.2 Impact of Pruning on Out-of-Distribution Segmentation

Figure 3 illustrates the impact of different pruning methods on segmentation performance across varying sparsity levels. The results, averaged over three seeds (denoted by the standard deviation in Table 2), are evaluated on three increasingly complex test sets. The x-axis represents the sparsity percentage, while the



(a) Results of testing on Dataset 1 (b) Results of testing on Dataset 3

Fig. 4: Trade-off comparison of different models' performance and size, averaged over three seeds. The marker sizes indicate the number of non-zero parameters, providing a visual representation of model complexity.

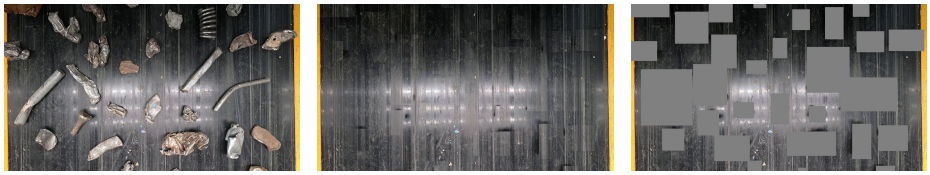
y-axis denotes segmentation mAP50-90. The comparison includes global magnitude pruning (GMP), SNIP, and our proposed IBIS method. The findings demonstrate that our approach (red lines) consistently outperforms other pruning techniques at higher sparsity levels, even on more complex datasets. Notably, while other methods experience a decline in segmentation accuracy as sparsity increases, our IBIS-based pruning retains or even enhances performance, indicating better robustness to out-of-distribution variations in the dataset.

The red cross ('x') in each plot marks the dense model's result, serving as a reference for evaluating pruning strategies. While pruning typically reduces accuracy; however, our IBIS method maintains—or even surpasses—the performance of the dense model at certain sparsity levels. As shown in Figure 3, performance improves in some cases, likely due to the effects of fine-tuning.

6.3 Trade-Off Between Model Performance and Size

Figure 4 presents a comparative analysis of segmentation and bounding box mAP50-95 metrics for different pruning strategies. Mainly, our results on the bounding box metrics follow a similar trend as in the segmentation metric results from our previous discussions (see red circles appearing towards the top right of the plots in Figure 4). Additionally, we test out-of-distribution generalization capabilities on Dataset 3, since it has the highest task difficulty. The x-axis represents segmentation accuracy, while the y-axis denotes bounding box accuracy. We compare the performance of the dense YOLOv11 model (black) against pruned variants using GMP (blue) and our IBIS method (red).

Figure 4(a) shows that on Dataset 1, IBIS-pruned models achieve a favorable balance between accuracy and sparsity. On Dataset 3, see Figure 4(b), it is shown that features a more challenging distribution of objects, the advantage of IBIS pruning becomes even more apparent. While GMP pruning significantly reduces segmentation accuracy at higher sparsity levels, it shows an accuracy



(a) Example input x (b) Patterned Masking (c) Gray Box Masking

Fig. 5: Comparison of masking strategies used in gradient estimation ablations.

Table 3: Results for segmentation and bounding-box performance under different masking scenarios. Accuracies are averaged over 3 random seeds.

	Segmentation mAP50-95 (%)			Bounding-Box mAP50-95 (%)		
	IBIS	Patterned Masking	Gray Box Masking	IBIS	Patterned Masking	Gray Box Masking
Sparse (25%)	**74.60**	74.17	74.50	95.80	95.77	**95.83**
Sparse (50%)	74.43	74.23	**74.77**	95.70	**96.00**	95.93
Sparse (75%)	**76.10**	75.03	75.03	**96.07**	95.27	95.37
Sparse (85%)	**76.73**	75.30	75.13	**96.43**	96.07	95.93
Sparse (90%)	**77.43**	75.37	75.33	95.50	**96.00**	95.67
Sparse (95%)	**76.23**	75.97	76.00	94.87	**95.13**	95.10

increase again at 95% sparsity. In contrast, our approach maintains competitive performance with a significantly reduced parameter count. These results highlight IBIS's in optimizing neural networks for real-world recycling applications while ensuring a favorable trade-off between model size and predictive accuracy.

6.4 Ablation Study with Masking-based Score Estimates

We analyze the impact of different masking strategies during pruning on segmentation and bounding-box performance across various sparsity levels. Table 3 presents the results for segmentation (mAP50-95) and bounding-box (mAP50-95) under three scenarios: **IBIS** (Figure 2), **Patterned Masking** (Figure 5b), and **Gray Box Masking** (Figure 5c). IBIS computes importance scores using full-background frames, a lightweight and effective approach requiring only a background-only image. If such a frame is unavailable, alternative masking strategies like Patterned and Gray Box Masking could also be considered.

In the Patterned Masking strategy, objects in the images are masked using background patches. Instead of masking objects with a uniform color, we mask them with patches extracted from the background of the original frame. This technique preserves natural scene structure, helping the model learn in a more contextually realistic manner, but introduces inconsistencies at object boundaries. In contrast, Gray Box Masking fully masks objects with solid gray boxes, forcing the model to rely solely on the surrounding context for importance esti-

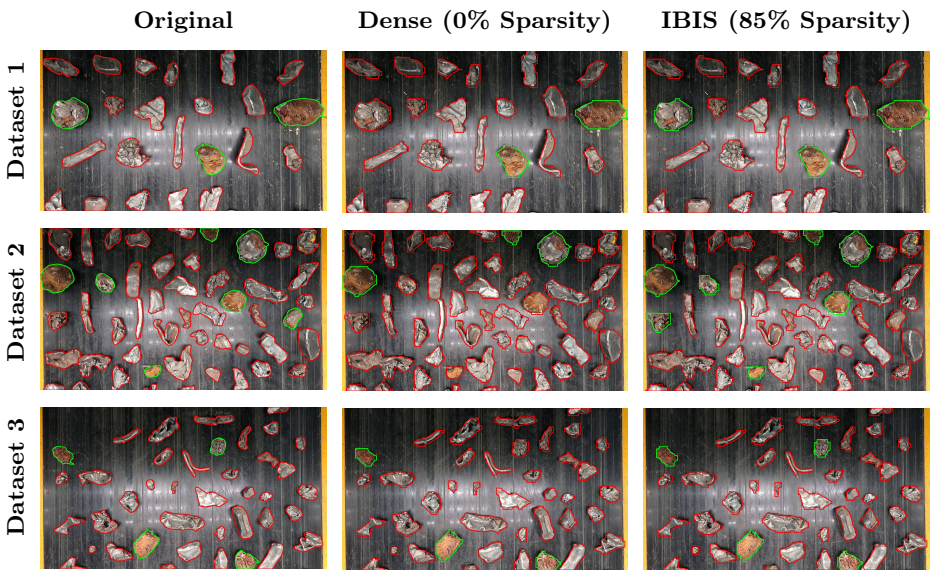


Fig. 6: Comparison of segmentation results on three different test sets (Dataset 1, Dataset 2, and Dataset 3). Each column corresponds to a different model: (1) Original, (2) Dense, and (3) IBIS. The leftmost images in each row show the original ground truth labels, while the middle and right images show the segmentation results from the Dense model and IBIS method, respectively.

mation. Importance scores in these settings are computed as:

$$\bar{s}_j^{\mathrm{OCC}} = |w_j| \cdot [1 - \exp(-\bar{c})], \quad \text{where} \tag{6}$$

$$\bar{c} = \left| \mathbb{E}_{(x,y)\sim\mathcal{D}_{\text{train}}}[\nabla_{w_j}\mathcal{L}(f(x;\boldsymbol{w}),y)] - \mathbb{E}_{(x,y)\sim\mathcal{D}_{\text{train}}}[\nabla_{w_j}\mathcal{L}(f(\bar{x};\boldsymbol{w}),\bar{y})] \right|. \tag{7}$$

In IBIS, c (Eq. 2) uses the gradient from a single background image $\tilde{x}$, while this method uses $\bar{c}$, the averaged gradient from masked images $\bar{x}$. Table 3 shows that all masking strategies perform similarly at lower sparsity levels. However, as sparsity increases, differences emerge. For segmentation, Patterned Masking tends to underperform at higher sparsity levels (90-95%), suggesting that inconsistencies in background patch placement negatively impact model generalization. Meanwhile, Gray Box Masking remains competitive across all sparsity levels, achieving comparable or superior results to IBIS in some cases. For bounding-box detection, all methods perform closely at moderate sparsity levels, but at extreme sparsity (90-95%), Patterned Masking shows a slight advantage over IBIS, while Gray Box Masking remains stable. These findings highlight the importance of masking strategy selection in pruning. When full-background frames are unavailable, Patterned and Gray Box Masking are viable alternatives, with the latter offering more consistent performance under extreme sparsity.

7 Conclusion

We introduce IBIS, an instance-based importance scoring method for pruning segmentation models used in an industrial recycling setting. Unlike traditional pruning approaches like GMP and SNIP, IBIS leverages gradient-based importance scores, prioritizing weights based on their relevance to object regions rather than treating all gradients uniformly. This approach preserves critical features and maintains high segmentation accuracy, particularly at high sparsity levels.

Our results demonstrate that IBIS not only reduces model size and inference costs but also enhances generalization, even in out-of-distribution scenarios. This makes it a viable solution for resource-constrained applications such as real-time recycling systems. Furthermore, we show that masking strategy selection influences pruning effectiveness. When full-background frames are unavailable, Gray Box Masking offers a slight advantage at extreme sparsity, providing a robust alternative. As shown in Figure 6, IBIS, with 85% sparsity, achieves better performance than the dense model (0% sparsity), demonstrating its practical effectiveness and robustness in real-world applications. To support future research and industrial deployment, we plan to publicly release the full annotated dataset used in this study. Overall, IBIS balances efficiency and accuracy, making it a promising approach for scalable industrial deep learning applications with lightweight neural network models.

Acknowledgments. The project "KIRAMET KI based Recycling Metalcompound-Waste" (Project number FO999899661) is funded by the Austrian Research Promotion Agency (FFG) and the Federal Ministry for Climate Action, Environment, Energy, Mobility, Innovation, and Technology. Video and photo material of steel waste were recorded at the Digital Waste Research Lab of the Chair of Waste Processing Technology and Waste Management, TU Leoben. Scholz Austria GmbH contributed as a partner for scrap test specimens and a research collaborator.

References

1. Aboussouan, L., et al.: Steel scrap fragmentation by shredders. Powder Technology **105**(1-3), 288–294 (1999)
2. Bashkirova, D., et al.: Zerowaste dataset: Towards deformable object segmentation in cluttered scenes. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 21147–21157 (2022)
3. Bircanoğlu, C., et al.: Recyclenet: Intelligent waste sorting using deep neural networks. In: Innovations in intelligent systems and applications (INISTA). pp. 1–7. IEEE (2018)
4. Chen, T., et al.: Only train once: A one-shot neural network training and pruning framework. Advances in Neural Information Processing Systems **34**, 19637–19651 (2021)
5. European Commission: Communication from the commission to the european parliament, the european council, the council, the european economic and social committee, and the committee of the regions: The european green deal (2019)

6. Fournier-Viger, P., et al.: Machine learning for intelligent industrial design. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases. pp. 158–172 (2021)
7. Gedam, P.B., et al.: A systematic review: Development of AI based computer vision scrap sorting system for metal scrap. In: International Conference on Multi-Agent Systems for Collaborative Intelligence (ICMSCI). pp. 876–881. IEEE (2025)
8. Girshick, R.: Fast R-CNN. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 1440–1448 (2015)
9. Girshick, R., et al.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 580–587 (2014)
10. Gritsch, J.V., et al.: Preserving real-world robustness of neural networks under sparsity constraints. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases. pp. 337–354 (2024)
11. Han, S., et al.: Learning both weights and connections for efficient neural network. Advances in Neural Information Processing Systems **28** (2015)
12. Hassibi, B., et al.: Optimal brain surgeon and general network pruning. In: IEEE International Conference on Neural Networks. pp. 293–299 (1993)
13. He, K., et al.: Mask R-CNN. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 2961–2969 (2017)
14. He, W., et al.: Cap: Context-aware pruning for semantic segmentation. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 960–969 (2021)
15. He, Y., Xiao, L.: Structured pruning for deep convolutional neural networks: A survey. IEEE Transactions on Pattern Analysis and Machine Intelligence **46**(5), 2900–2919 (2023)
16. Jocher, G., Qiu, J.: Ultralytics yolo11 (2024), https://github.com/ultralytics/ultralytics
17. Jocher, G., et al.: Ultralytics YOLO (2023), https://ultralytics.com
18. Kohama, H., et al.: Single-shot pruning for pre-trained models: Rethinking the importance of magnitude pruning. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 1433–1442 (2023)
19. LeCun, Y., et al.: Optimal brain damage. Advances in Neural Information Processing Systems **2** (1989)
20. Lee, N., et al.: Snip: Single-shot network pruning based on connection sensitivity. arXiv preprint arXiv:1810.02340 (2018)
21. López-González, C.I., et al.: Filter pruning for convolutional neural networks in semantic image segmentation. Neural Networks **169**, 713–732 (2024)
22. Lv, X., et al.: Pruning for image segmentation: Improving computational efficiency for large-scale remote sensing applications. ISPRS Journal of Photogrammetry and Remote Sensing **202**, 13–29 (2023)
23. Micikevicius, P., et al.: Mixed precision training. arXiv preprint arXiv:1710.03740 (2017)
24. Molchanov, P., et al.: Pruning convolutional neural networks for resource efficient inference. arXiv preprint arXiv:1611.06440 (2016)
25. Muchová, L., et al.: End-of-waste criteria for iron and steel scrap: technical proposals. Joint Research Centre–Institute for Prospective Technological Studies. Luxembourg: Publications Office of the European Union (2010)
26. Nafiz, M.S., et al.: Convowaste: An automatic waste segregation machine using deep learning. In: 3rd International Conference on Robotics, Electrical and Signal Processing Techniques (ICREST). pp. 181–186. IEEE (2023)

27. Neubauer, M., Rückert, E.: Semi-autonomous fast object segmentation and tracking tool for industrial applications. In: 21st International Conference on Ubiquitous Robots (UR). pp. 77–83. IEEE (2024)
28. Quétu, V., et al.: The simpler the better: An entropy-based importance metric to reduce neural networks' depth. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases. pp. 92–108 (2024)
29. Redmon, J., et al.: You only look once: Unified, real-time object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 779–788 (2016)
30. Schäfer, M., et al.: DOES-A multimodal dataset for supervised and unsupervised analysis of steel scrap. Scientific Data **10**(1), 780 (2023)
31. Tanaka, H., et al.: Pruning neural networks without any data by iteratively conserving synaptic flow. Advances in Neural Information Processing Systems **33**, 6377–6389 (2020)
32. Teixeira, S., et al.: Improving smart waste collection using AutoML. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases. pp. 283–298 (2021)
33. Watch, C.M.: Decarbonising steel: A guide to the steel sector's decarbonisation (2022), https://carbonmarketwatch.org/wp-content/uploads/2022/03/CMW_Decarbonising-Steel_v02.pdf
34. Wu, T.W., et al.: Applications of convolutional neural networks for intelligent waste identification and recycling: A review. Resources, Conservation and Recycling **190**, 106813 (2023)
35. Xu, W., et al.: Classification and rating of steel scrap using deep learning. Engineering Applications of Artificial Intelligence **123**, 106241 (2023)