Clean Technologies and Recycling

Clean Technologies and Recycling, 6 (1): 106–121.
DOI: 10.3934/ctr.2026005
Received: 09 October 2025
Revised: 18 December 2025
Accepted: 15 January 2026
Published: 03 February 2026

https://www.aimspress.com/journal/ctr

Research article

Exploring deep learning for sorting shredded end-of-life wind turbine blades

Christian Linder *, Fabian Rechsteiner and Samuel Eiler

Fraunhofer Institute for Casting, Composite and Processing Technology IGCV, Am Technologiezentrum 2, 86159 Augsburg, Germany

* **Correspondence:** Email: christian.linder@igcv.fraunhofer.de.

Abstract: The increasing popularity of wind energy has led to an extensive growth in wind turbine installations. This poses challenges when these wind turbine blades reach the end of their operational lifespan. Thus, suitable recycling techniques are required to extract the different materials from the turbine blades. Therefore, we explored the use of deep learning-based object detection for sorting shredded End-of-Life wind turbine blades into individual material classes. To reach this goal, a custom dataset combining images from shredded wind turbine blades, with a synthetic data generation approach, was employed. Three popular object detection architectures (SSD, YOLO, and Faster R-CNN) were implemented and tested. The impact of different popular backbone networks and Feature Pyramid Networks on accuracy and speed was analyzed. Our results showed that the SSD model with a ResNet18 backbone and a version of Feature Pyramid Network performed well in terms of accuracy and speed. Moreover, synthetic data generation proved useful but showed a performance decline when transitioning to real-world pictures, suggesting the need for combining synthetic and real-world data. This study highlights the potential of deep learning in recycling and sorting wind turbine blades, encouraging further research in this field.

Keywords: End-of-life wind turbine blades; sorting; deep learning; object detection; synthetic data

1. Introduction

Sustainable End-of-life (EoL) management of wind turbines is essential for minimizing environmental impact. While most parts of a wind turbine plant are recycled, the EoL of wind turbine blades (WTB) remains a challenge [1]. WTBs are built using a number of different materials, including glass and carbon fiber reinforced polymers (GFRP and CFRP, respectively), metals, wood, foams, and coating in order to meet the properties needed for the strong forces in high wind phases. The composite material structure provides the needed strength to meet the increasingly longer and heavier blades [2,3]. This material composition leads to fundamental challenges in the decommissioning and recycling procedure of wind turbine plants. Therefore, WTBs often end up in landfills or waste incineration plants [4]. Landfill bans for WTBs from wind associations like Wind Europe in 2021 [5] are just one attempt by the industry to decrease the amount of WTBs going to landfills. To reduce the WTBs in landfills, suitable recycling processes need to be in place. For an adequate recycling of the different material classes, they must be sorted into the corresponding material classes. Recycling processes rely on low impurities to produce material of high quality. Some composite recycling processes cannot work with low metal quantities and vice versa [6]. However, there are limited technologies available to sort shredded WTB waste. One potential technology is AI by using deep learning-based object detection algorithms.

AI technologies are being deployed in different aspects of the wind industry or other sustainability technologies, as seen in [7] for fault diagnosis in wind turbines, in [8] for offshore support structures, in [9] for specific parts of the wind turbine like the gear box, and in [10] for resource allocation and optimization in large-scale networks. In this paper, we will further develop AI technology by focusing on the development of deep learning-based object detection algorithms to identify the material classes of shredded EoL WTBs and categorize them into the classes GFRP, CFRP, metal, wood, chalk, and textile when shown on the same picture. Another focus is the generation of synthetic data to overcome the difficulties associated with labeling the materials of shredded EoL WTBs and the need for large and diverse data sets.

1.1. Industrial sorting process

The proposed object detection approach is suitable for an industrial sorting process, where a camera is placed above a conveyor belt to capture the materials stream, which is then processed using the developed sorting algorithm. The material is distributed by a vibration machine at the beginning of the conveyor belt to assure no overlapping of material pieces [11]. At the end of the conveyor belt, air impulses from several high-pressure valves in a row separate the materials into baskets, according to their material class, utilizing the information from the sorting algorithm and conveyor belt speed. A systematic setup can be taken from Figure 1. See [11] for more details.

1.2. Related work

Object detection methodologies are generally categorized into two types: One-stage and two-stage models. Two-stage models, such as Region-Based CNNs (R-CNNs) like Faster R-CNN [12] and Mask R-CNN [13], operate in two distinct phases [12–15]. The initial phase generates potential object regions, followed by a second phase where these regions are classified using a separate neural network. In contrast, one-stage models, including variations of YOLO [16] and SSD [17], integrate region proposals and classification tasks into a single neural network, which results in superior processing speeds compared to their two-stage counterparts. Advancements in the field of object detection in the last decades have been predominantly driven by innovations in Convolutional Neural Networks (CNNs) [18–20]. Despite the efficiency of CNNs, which have long served as the backbone for these detection systems, they are facing competition from transformers [21]. Recent developments have shown that transformers can achieve higher accuracy than CNNs, although CNNs remain popular due to their relatively simple training procedures and more compact model sizes [22,23]. Additionally, anchor-free models, such as the anchor-free versions of YOLO [24] and DETR [25], have emerged, demonstrating promising results by eliminating the need for predefined anchor boxes and simplifying the detection pipeline.

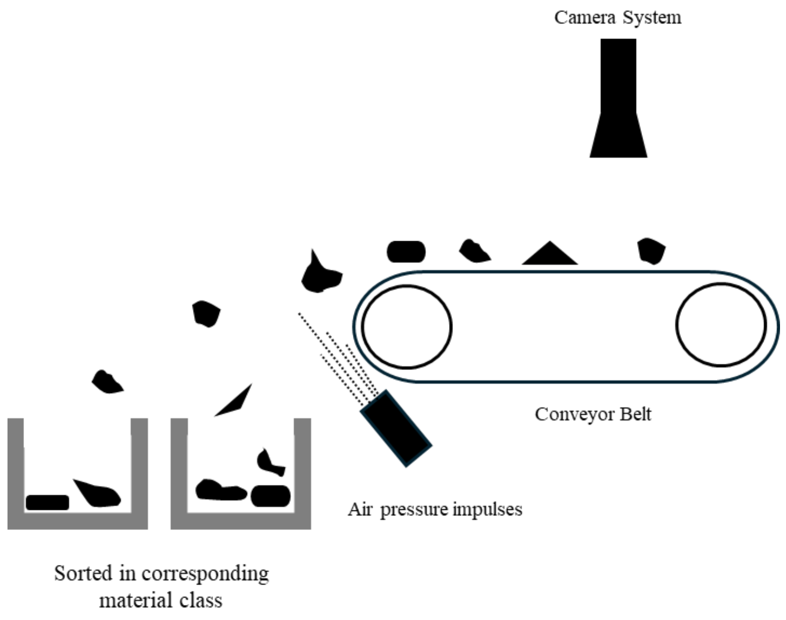

Figure 1. Industrial sorting setup using a camera system and air pressure impulses [11].

In the context of waste sorting, the application of deep learning-based object detection algorithms has demonstrated significant promise. Recent advancements include the development of specialized models and datasets tailored for the complexities of waste sorting. For instance, the OATCR [26] project utilized YOLOv4 and Mask R-CNN with the TACO [27] dataset to achieve high detection accuracy for trash classification, emphasizing real-time processing capabilities. Additionally, the research by D. Melinte et al. [28] highlighted the effectiveness of SSD with MobileNetv2 for faster and more accurate waste classification compared to Faster R-CNN. Notably, the work by Koskinopoulou et al. [29] introduced a robotic waste sorting system leveraging Mask R-CNN trained on synthetically generated data, showcasing practical deployment in industrial settings.

Furthermore, W. Mao et al. [30] addressed the challenge of single-object datasets by employing the TWRD dataset for real-world waste detection, demonstrating the need for context-specific datasets to enhance model performance in operational environments. These advancements illustrate the ongoing evolution of object detection technologies in addressing the pressing issue of waste management.

Hence, in this work, we investigate the utilization of three object detection algorithms, namely YOLO [16], Faster RCNN [12], and SSD [17], to sort shredded WTB pieces and use synthetically generated data in the process. The evaluation is conducted using common metrics like mean average precision (mAP), speed, and recycling-related metrics such as purity and recovery rate.

2. Methodology

In the following section, we propose an approach to detect and classify shredded WTB materials. To effectively recycle these materials, accurate sorting is essential. Given the complexity and variety of the shredded components, we use synthetic data to enhance the dataset diversity and reduce the labor-intensive task of manual labeling. By incorporating synthetic data, our aim is to cover a wide range of material variations and improve the robustness of the detection system. Various object detection methods are compared to determine the most suitable technique for accurate detection and classification of the materials.

2.1. Generation of synthetic data

For this application, synthetic data is generated through a multi-step process that is visualized in Figure 2. First, objects derived from shredded WTBs are manually sorted according to their respective material classes. Manual sorting has the advantage that each object can be examined from multiple angles, simplifying the process as it is difficult to differentiate each material from a single image. Following the sorting phase, objects of the same material class are grouped together and positioned on a black background so that the objects do not overlap. Thereafter, images are taken with a resolution of 1920x1080 pixels. An example for each material class is visualized in Figure 3. Classical image processing techniques are then used to segment and extract the individual objects within the captured images. Specifically, the captured images are converted to grayscale, smoothed with a median blur, and binarized by thresholding to produce a mask with background pixels set to zero and object pixels to one. Contours are extracted from this mask and analyzed using connected-component analysis to identify individual objects in the binary mask.

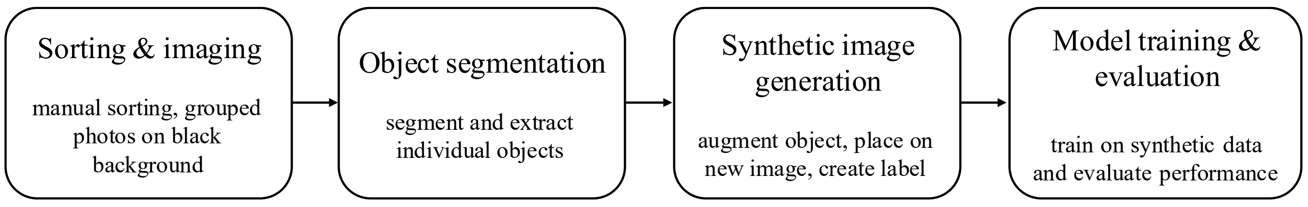
Sorting & imaging
manual sorting, grouped photos on black background
Object segmentation
segment and extract individual objects
Synthetic image generation
augment object, place on new image, create label
Model training & evaluation
train on synthetic data and evaluate performance

Figure 2. Workflow of synthetic data generation and training.

Similar to the methodology described in [29], the segmented objects can then be combined using transformation functions on each individual object and placing them onto a new image with a predefined background to create an image with multiple objects of different material classes. The background can be selected based on real-world conditions, which, in our case, is a black background resembling a conveyor belt. The transformation functions implemented include resizing, rotation, vertical and horizontal flipping, brightness, contrast, hue, saturation, noise, and blur. The parameter range for each transformation during training is described in Table 1. In addition, the objects are placed so that there is no overlap, and a pre-defined number of objects are placed within the image. Data augmentation is also applied to the image using random noise and a Gaussian blur effect. An example of a synthetically generated image is visualized in Figure 4. The synthetic data generation method offers significant advantages by enabling the number of objects per image, the class distribution, and the object size distribution to be modified and training to be focused on specific parameters. In addition, the process is easily transferable to new scenarios, as the resolution and background of the generated image can be changed. The method also enables large amounts of data to be generated from a few real images by placing objects in different positions and supports object-level data augmentation, which is typically not achievable with standard techniques that modify the image.

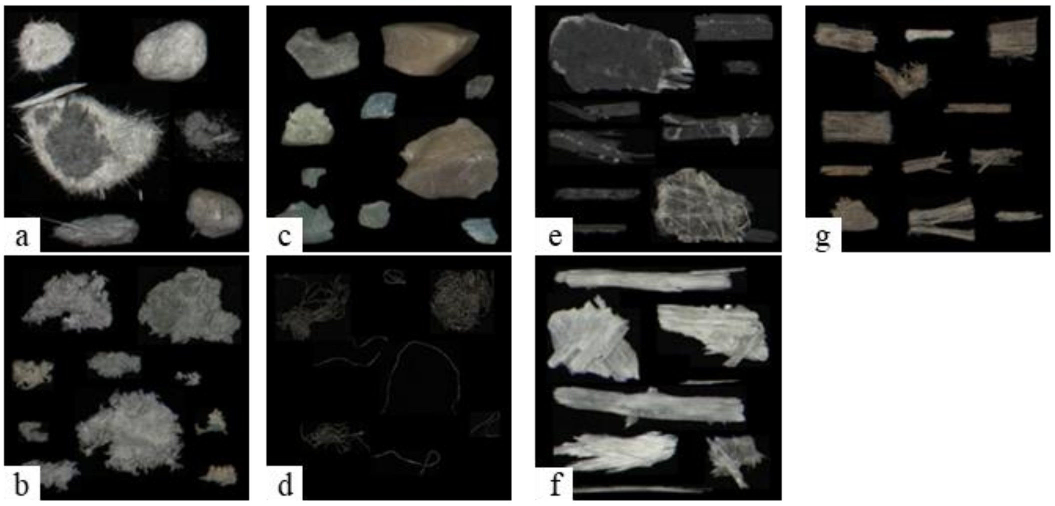

Figure 3. Examples of grouped material images for each material class with a) chalk, b) textile, c) plastic, d) metal e) CFRP, f) GFRP, and g) wood.

Table 1. Parameter range of each augmentation method used for the synthetic dataset generation.

Data augmentation	
Method (Object level)	**Parameter range**
Rotation	0-360°
Flip	H, V
Resize	-6, 20%
Brightness	-10, +10
Contrast	-7, +7
Hue	-15, +15°
Saturation	-7, +7%
Method (Image Level)	
Brightness	-10, +10
Contrast	-7, +7
Hue	-15, +15°
Saturation	-7, +7%
Blur	0.01,1
Noise	0.75

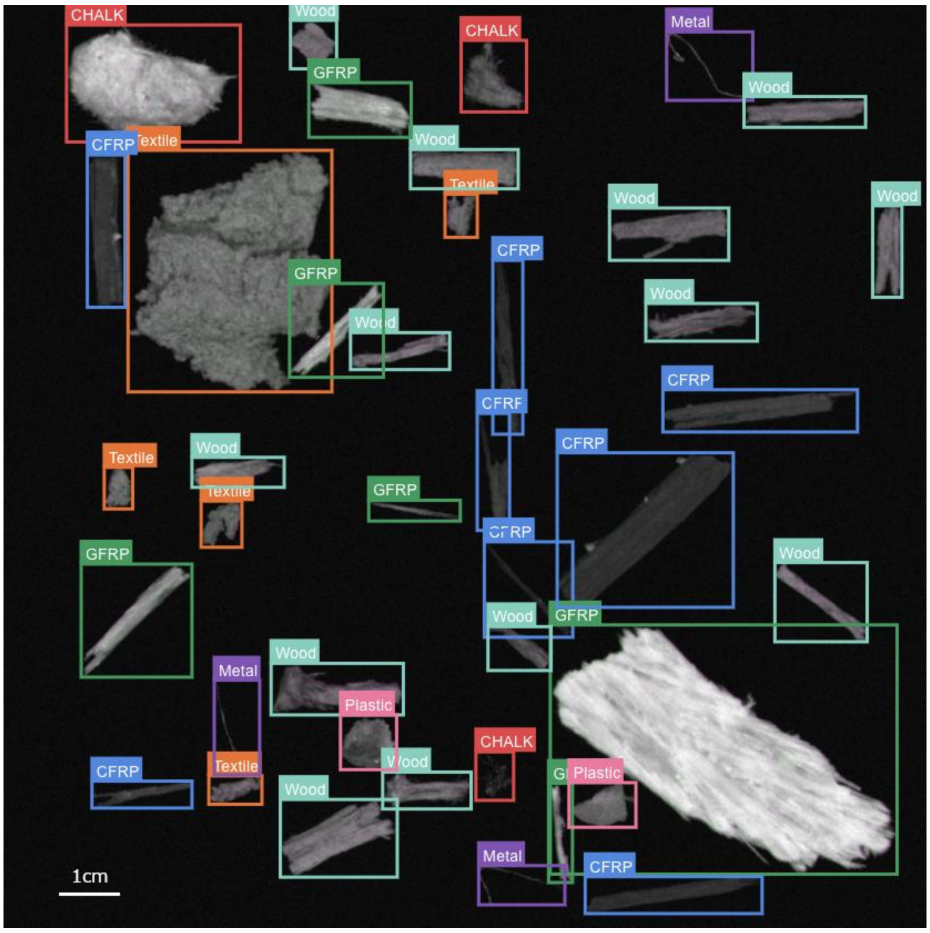

Figure 4. Example image from the synthetically generated dataset.

2.2. *Algorithms*

In this study, multiple object detection models from two-stage and single-stage categories, specifically Faster R-CNN, YOLO, and SSD, are evaluated. These models are selected based on a literature review that identified them as among the most widely used and well-established object detection models. The models are fine-tuned on a synthetically generated dataset and combined with different backbones. The chosen backbones (see Table 2) span lightweight mobile architectures, standard residual networks, efficiency-focused CNNs, and modern Transformer-based models. By covering more traditional CNN-based designs and recent Transformer-based vision models, our selection reflects several generations of architecture development and a broad range of model capacities and computational costs. As a result, our selection enables a direct comparison across a wide variety of detector-backbone combinations and enables us to identify the architectures that achieve the best result. The models' performance is compared in terms of Mean Average Precision (mAP) at different Intersection over Union (IoU) thresholds, and latency is measured as frames per second (FPS). We also calculate precision, recall, purity, and recovery rate. Let TP, FP, and FN denote the numbers of true positives, false positives, and false negatives, respectively. Recall is defined as

$$recall = \frac{TP}{TP + FN}, \tag{1}$$

and precision as

$$precision = \frac{TP}{TP + FP}. \quad (2)$$

We calculate recovery rate analogously to recall, using the same formula,

$$recovery\,rate = \frac{TP}{TP + FN}, \quad (3)$$

and purity analogously to precision,

$$purity = \frac{TP}{TP + FP}. \quad (4)$$

2.2.1. Multi-Scale Feature Extraction

Regardless of the backbone architecture, feature maps are generated from the input image at three scales:

Scale 1: Downscaling factor of 1/16
Scale 2: Downscaling factor of 1/32
Scale 3: Downscaling factor of 1/64,
(obtained by using an additional convolutional block)

These multi-scale feature maps are then fed into the neck of the models, where two versions of Feature Pyramid Networks (FPNs) are tested to facilitate top-down and bottom-up information flow. The aggregated feature maps generated by the FPNs are used as input for the object detection heads of the model architectures. Figure 5 shows the process.

2.2.2. Model Description

Faster R-CNN [12]: Adapted with improvements from Mask R-CNN [15], such as ROI-Align and feature aggregation from the FPN. This model uses a Region Proposal Network (RPN) to generate proposals for possible object locations, followed by classification and refinement of these proposals.

Single Shot MultiBox Detector (SSD) [17]: Modified based on the original SSD model with influences from RetinaNet [31] and EfficientDet [32]. It features two small prediction heads for class and box regression, enabling high-speed object detection directly from the dense feature maps.

You Only Look Once (YOLO): A variant of YOLOv5 [33] with a decoupled prediction head inspired by YOLOX [24] and YOLOv6 [34]. It features separate heads for class, box, and objectness prediction, enabling faster and more accurate detection of multiple objects in a single image.

Each model is fine-tuned on a synthetically generated dataset, which includes diverse augmentations of single and multiple object images. The models are then tested on synthetic and real-world images to provide a comprehensive comparison of their accuracy and latency.

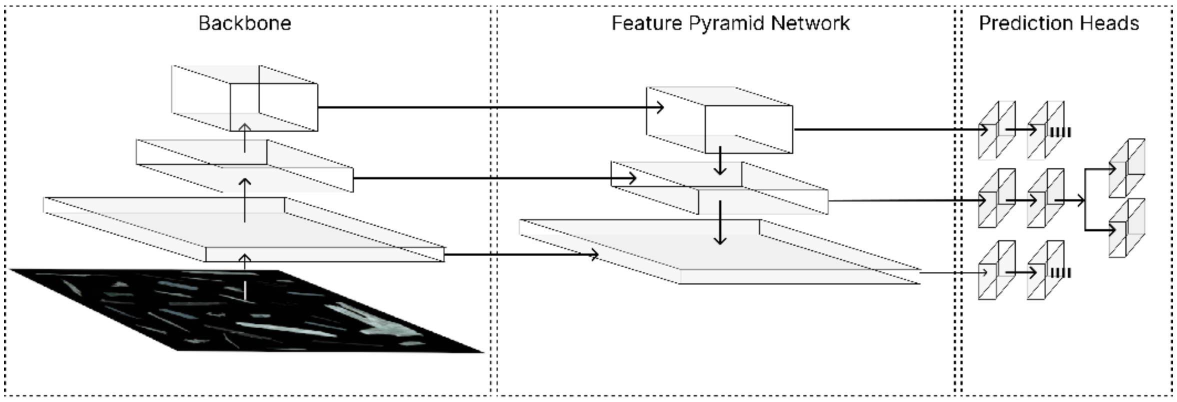

Figure 5. General model overview for all experiments using object detection architectures.

2.3. Experimental setup

The training setup is designed to maximize efficiency and performance. The models are trained on a NVIDIA GeForce RTX 2080 Ti GPU for 120 epoch. All models use the Adam optimizer [35] with a weight decay set to 5×10^{-6}. The learning rate for each model is estimated using the WandB sweep function [36] and scaled to the batch size by formula $LR=LR_0\times BS/8$, where the batch size is limited by GPU memory. Table 2 provides a detailed list of the learning rates and batch sizes for each model configuration. Furthermore, a cyclic learning rate approach is adopted, where the learning rate initially increases and then gradually decreases over time. This technique aids in escaping local minima during the training process. In addition, transfer learning based on pre-trained ImageNet1k backbones is used as it boosts the model's performance. All layers are fine-tuned during training.

Table 2. Learning rate and batch size configuration during training for each backbone and object detection architecture.

	YOLO		SSD		RCNN	
Backbone	**LR0**	**BS**	**LR0**	**BS**	**LR0**	**BS**
MobileNetv2	1e-6	16	1e-5	16	1e-6	8
MobileNetv3s	1e-6	32	1e-5	32	1e-6	16
MobileNetv3l	1e-6	16	1e-5	16	1e-6	10
EfficientNet b0	1e-6	16	1e-6	16	1e-6	12
EfficientNet b1	1e-6	10	1e-6	10	1e-6	8
EfficientNet b2	1e-6	10	1e-6	10	1e-6	8
EfficientNetv2	1e-6	8	1e-6	8	1e-6	8
ResNet18	1e-6	16	1e-5	16	1e-6	8
ResNet34	1e-5	12	1e-5	16	1e-6	8
ResNet50	1e-6	8	1e-5	8	1e-6	4
ResNet101	1e-7	4	1e-5	4	1e-6	2
ResNeXt50	1e-6	8	1e-5	8	1e-6	4
Convnext	1e-7	8	1e-6	8	1e-6	4
Swin t	1e-7	8	1e-6	8	1e-6	4

The images taken from the materials are from a batch of shredded wind turbine blades provided by a wind turbine recycling facility called Eurecum GmbH & Co. KG from Lutherstadt Eisleben in Germany. The material mix includes seven classes. A total of 3096 objects form the base of the synthetic dataset. The distribution across the seven classes is as follows: CFRP (573), Chalk (273), GFRP (537), Metal (243), Plastic (459), Textile (489), and Wood (558). A total of 80 sets of 1024 images each (81,920 images in total) with a resolution of 832x832 pixels are generated for training purposes. Additionally, 256 images are created for testing and another 256 for validation. For testing and validation, the parameter range for the transformation functions during data creation is reduced to half the values in Table 2. Alongside these synthetic images, a small subset of 16 real, non-synthetic data is used to test the robustness of our models. The dataset exhibits moderate class imbalance. During the generation of synthetic images, we preserve this distribution and do not use additional resampling or class weighting strategies. This choice is made to reflect the expected real-world frequencies of object classes. However, the class distribution in our dataset can be adjusted through the synthetic data generation pipeline by oversampling the less frequent categories, which we plan to explore in future work.

3. Results and discussion

3.1. Model

The evaluation of the models is carried out using the mAP metric at different IoU thresholds, as well as inference speed measured in FPS. Table 3 presents the results for mAP@[0.5:0.95] (full range), and specific thresholds mAP.5 (IoU = 0.5) and mAP.75 (IoU = 0.75), along with the inference speeds for different configurations. For simplicity, only the better-performing version of the Feature Pyramid Network (FPN) or the revised BiFPN (rBiFPN) is presented in the table.

3.1.1. Impact of Backbone Architectures and Model Components

The SSD model with the ResNet101 backbone and standard FPN achieve the highest overall mAP of 0.910, also leading in mean average recall (mAR100) for 100 boxes. For specific thresholds, such as mAP.5 and mAP.75, the SSD model with an EfficientNetv2s backbone and FPN shows the best performance, with scores of 0.962 and 0.956, respectively. This highlights the significant impact that backbone selection has on detection accuracy, especially in scenarios that require precise localization.

However, in terms of speed, the SSD model equipped with a ResNet18 backbone and rBiFPN emerges as the fastest, achieving over 200 fps. This model processes images three times faster than the top-performing SSDs with ResNet101 and EfficientNetv2s backbones, with only minor reductions of 3.2 points in mAP, 2.0 points in mAP.75, 1.9 points in mAP.5, and 2.7 points in mAR100. This demonstrates a compelling trade-off between accuracy and speed, making the ResNet18 SSD model highly suitable for real-time applications.

When comparing CNN-based backbones with the Transformer-based Swin Transformer model, CNN-based architectures consistently outperform the latter in terms of accuracy and speed for this application. Although the Swin Transformer has potential, it may require further optimization of hyperparameters and training strategies to match the efficiency of CNN-based models. Similarly, smaller models like MobileNet are found to be highly sensitive to hyperparameter configurations, impacting their reliability in this setup.

During training, we also monitor the training and validation loss for all compared models over the epochs. After an initial decrease, training and validation loss generally plateau, and no pronounced divergence between them is observed, indicating that the models do not strongly overfit to the training data.

Table 3. Mean average precision over different Intersection over Unions and inference speeds of all tested backbone and object detection architecture.

	YOLO				SSD				RCNN			
Backbone	**mAP**	**mAP$_{75}$**	**mAP$_{50}$**	**fps**	**mAP**	**mAP$_{75}$**	**mAP$_{50}$**	**fps**	**mAP**	**mAP$_{75}$**	**mAP$_{50}$**	**fps**
Convnext_t	0.708	0.861	0.951	77	0.884	0.940	0.950	83	0.759	0.857	0.894	54
Eff.Netv2s	0.753	0.908	0.949	58	0.909	0.956	0.962	63	0.793	0.887	0.912	49
Eff.Net b0	0.773	0.876	0.930	101	0.868	0.939	0.948	116	0.811	0.888	0.912	78
Eff.Net b1	0.785	0.912	0.945	50	0.883	0.940	0.948	55	0.768	0.884	0.924	44
Eff.Net b2	0.748	0.899	0.944	67	0.894	0.949	0.956	72	0.842	0.911	0.924	42
Mob.Netv2	0.775	0.908	0.939	88	0.870	0.939	0.947	97	0.838	0.906	0.920	65
Mob.Netv3l	0.765	0.899	0.936	98	0.854	0.931	0.940	112	0.750	0.862	0.902	76
Mob.Netv3s	0.716	0.854	0.917	81	0.773	0.882	0.912	93	0.714	0.831	0.882	64
ResNet101	0.822	0.917	0.936	57	0.910	0.947	0.950	63	0.896	0.933	0.946	30
ResNet18	0.840	0.930	0.943	152	0.878	0.936	0.943	203	0.815	0.896	0.912	94
ResNet34	0.711	0.841	0.911	88	0.880	0.935	0.944	14	0.869	0.925	0.939	75
ResNet50	0.853	0.933	0.947	87	0.908	0.948	0.952	87	0.891	0.935	0.946	53
Resnext50	0.856	0.938	0.950	80	0.908	0.948	0.953	80	0.870	0.921	0.931	44
Swin_t	0.571	0.674	0.888	58	0.847	0.929	0.938	45	0.607	0.701	0.826	44

3.1.2. Comparative Analysis of Architectures

- **Faster R-CNN:** Delivers the highest mAP scores with ResNet backbones but suffer from slower inference speeds, making it less suitable for real-time applications.
- **YOLO:** Provides competitive detection speed and accuracy but shows reduced bounding box regression performance at higher IoU thresholds.
- **SSD:** Outperforms the other models in terms of detection accuracy and speed, particularly with EfficientNet backbones, making it the ideal choice for real-time sorting tasks.

3.2. Detailed Analysis of the ResNet18 Model

A detailed analysis is conducted on a single model to evaluate the performance characteristics. The SSD ResNet18 with rBiFPN is chosen for in-depth evaluation because it demonstrates the best trade-off between high inference speed and competitive accuracy, as shown in Figure 6, with a mAP of 0.878 (mAP.75: 0.936 and mAP.50: 0.943) and FPS of 203. This makes it particularly suitable for real-time applications, where rapid decision-making is crucial.

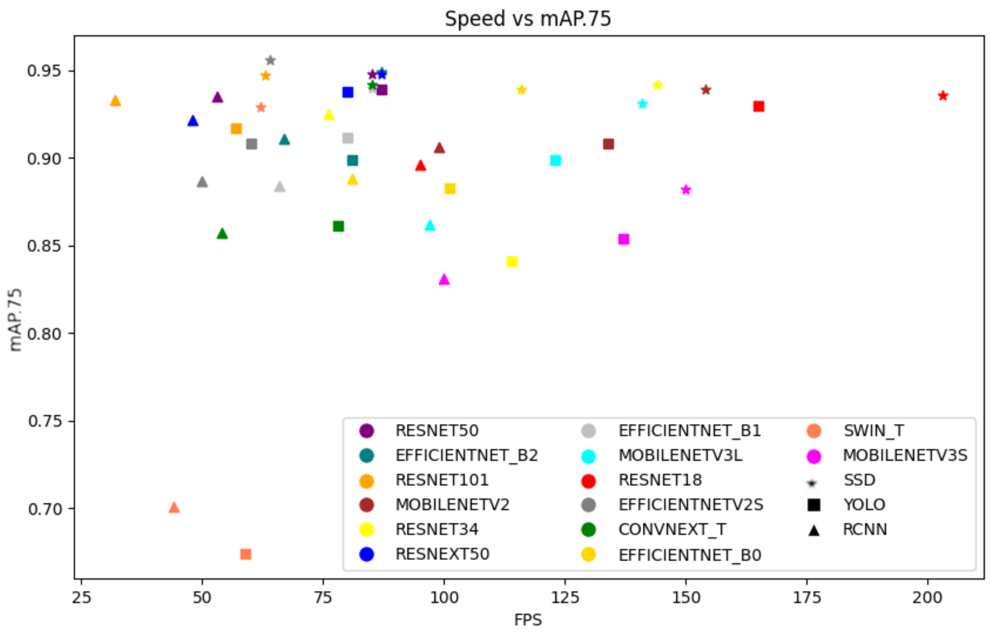

Figure 6. Speed vs mAP with an IoU threshold of 0.75.

3.2.1. Ablation Study on Anchor Boxes and Aspect Ratios

The ResNet18 SSD model with rBiFPN achieves an overall mAP of 0.878, showing the best performance on medium-sized objects (areas between 50^2 and 100^2 pixels), with a mAP of 0.905. Its performance for large objects (area $>100^2$ pixels) is slightly lower, with a mAP of 0.874, and the lowest performance is observed for small objects (area $<50^2$ pixels) with a mAP of 0.804. The lower performance for small objects is likely due to the limited resolution and the inherent difficulty of reliably classifying very small components, which can be challenging even for humans. We plan to address this issue in the future using higher-resolution input images to better preserve the details of small objects.

During experimentation, varying the size of anchor boxes reveals that larger anchor boxes improve detection on medium and large objects but decreases accuracy on small objects (see Table 4). Conversely, reducing the base anchor sizes significantly degrades performance across all object scales. Additional tests with different numbers of anchor box aspect ratios indicate that increasing the number of aspect ratios reduces overall performance, while decreasing them improves mAP.75 but leads to lower scores in mAP and mAP.5.

Table 4. Anchor box ablation study with the SSD ResNet18 model.

Modification	mAP	mAP_{50}	mAP_{75}	mAP_{small}	mAP_{medium}	mAP_{large}
Base Model SSD ResNet18 rBiFPN	**0.878**	0.943	**0.936**	**0.804**	0.905	0.874
More Ratios (0.3, 0.6, 1.0, 1.4, 1.7)	0.828	0.911	0.898	0.725	0.871	0.816
Less Ratios (0.8, 1.2)	0.876	**0.945**	0.935	0.791	0.895	0.896
Smaller Boxes (Scales 40, 90, 160)	0.802	0.901	0.883	0.732	0.808	0.805
Larger Boxes (Scales 60, 120, 220)	**0.878**	0.939	0.933	0.767	**0.924**	0.901

These observations suggest that the SSD model's performance is sensitive to anchor box configurations, and careful tuning of anchor sizes and aspect ratios is crucial to balancing detection accuracy across object scales. This ablation study provides only a preliminary insight into these effects. Future work should involve a more systematic optimization of anchor dimensions and aspect ratios for this application.

3.2.2. Inter-Class Performance Evaluation

The inter-class mAP distribution is relatively consistent, with the best detection accuracy achieved for the *Wood* class, and the lowest performance observed for the *CFRP* class. Detailed scores for each class are shown in Table 5, indicating robust performance across most material types, with slight variations in precision for specific classes.

Table 5. Inter-class performance evaluation with mAP with the SSD ResNet18 model.

	CFRP	**GFRP**	**Chalk**	**Plastic**	**Metal**	**Wood**	**Textile**
mAP	0.838	0.851	0.900	0.882	0.870	0.913	0.893

These results indicate that while the detection accuracy is generally strong across the board, specific material types like CFRP could benefit from further refinement in the detection algorithm or the dataset.

3.2.3. Analysis of Detection Errors

In waste sorting, false negatives are often less critical compared to false positives. In our evaluation, the model accurately detects 8,364 of 8,870 objects, resulting in a detection accuracy of 94.3%. However, 506 instances (5.71%) are either missed or misclassified. By analyzing the confusion matrix, we derive purity (analogous to precision) and recovery rates (analogous to recall), achieving a purity rate of 97.0% and a recovery rate of 94.6%, highlighting the model's strong performance in minimizing contamination rates.

Upon closer analysis of the misclassified objects, it becomes evident that certain objects are consistently misclassified due to their visual similarity with other factions, disproportionately affecting the overall performance. For instance, a single object in the Chalk class accounts for 75% of the false positives in that category. Furthermore, the dataset includes some images that are either incorrectly labeled or visually ambiguous, which negatively impacts the model’s training and evaluation processes. Table 6 shows the results for each class.

Table 6. Purity and Recovery Rates of the different material classes with the SSD ResNet18 model.

	CFRP	**GFRP**	**Chalk**	**Plastic**	**Metal**	**Wood**	**Textile**
Purity	0.969	0.990	0.904	0.956	0.998	0.999	0.971
Recovery Rate	0.946	0.931	0.968	0.915	0.961	0.970	0.926

3.3. Performance Difference Between Synthetic and Real Images

A noticeable performance gap emerges when evaluating models trained on synthetically generated data against real-world images. The mAP scores exhibit a decrease usually in the range of 10 to 20 points depending on the model and backbone. Although the decline in performance is somewhat less severe when considering specific thresholds like mAP@0.5 and mAP@0.75, the reduction remains significant, highlighting the challenges models face when transitioning from synthetic to real-world scenarios. Comparative results are shown in Table 7.

Interestingly, certain models demonstrate better resilience to this domain shift. For instance, the YOLO model with a ResNet50 backbone and FPN exhibits only a marginal drop of around 5 points in mAP when moving from synthetic to real-world images, maintaining a competitive mAP@0.5 of 0.921 for real test data. This suggests that some architectures are inherently more adaptable to the variations found in real-world images, likely due to their robust feature extraction capabilities and the effectiveness of their training strategies.

This indicates that with the right combination of architecture and feature extraction techniques, the impact of the domain shift can be minimized, maintaining competitive detection accuracy in real-world applications.

Table 7. Analysis of the domain shift from synthetic to real image.

	mAP synth.	mAP real	$mAP_{.75}$ synth.	$mAP_{.75}$ real	$mAP_{.5}$ synth.	$mAP_{.5}$ real
YOLO ResNet50	**0.853**	0.805	**0.933**	**0.872**	0.947	0.921
SSD ResNet18	0.878	0.710	0.936	0.790	0.943	0.812

These findings emphasize the importance of fine-tuning models with real-world data to achieve optimal accuracy in practical applications. While synthetic data is highly effective for initial training and rapid prototyping, its limitations become apparent when transitioning to real-world conditions. The significant domain shift suggests that adopting a mixed training strategy, leveraging synthetic and real-world datasets, could significantly enhance the models' robustness, helping bridge the performance gap when deployed in real-world scenarios.

4. Conclusion and outlook

In this paper, we investigate the creation of a synthetic data set of images containing shredded WTBs and the identification of these objects with suitable algorithms. The results indicate that the SSD model achieves the best trade-off between speed and accuracy. However, other models like YOLO and Faster-RCNN provide competitive results, but with a lower inference speed. It is shown that the backbone greatly influences the accuracy and speed of the model. Moreover, ResNets, EfficientNets, and MobileNets provide a suitable feature extraction for this case. Additionally, a synthetic data generation approach is tested to generate training data without manually preparing thousands of images. The results show that it produces a high variation of different images while keeping the time required for data preparation and labeling low. The method also simplifies the transfer of object detection algorithms to changed conditions and data set variations because every object can be augmented individually. However, the performance of the model decreases significantly when a model trained on synthetic images is tested on real images. Therefore, in future research, researchers should investigate the combination of synthetic and real-world data in the training phase to overcome the challenges posed by the domain shift between training and deployment environments. Furthermore, to reduce misclassification of very small objects, researchers could investigate training with higher-resolution images and address class imbalance by modifying the synthetic data generation to oversample less frequent categories. Finally, this study provides only an initial insight into the influence of anchor box configurations, and future work should more systematically optimize anchor sizes and aspect ratios to better balance detection performance across object scales.

However, the results and approach presented are a substantial step toward a more efficient way of recycling EoL WTBs by increasing the purity of each material and, consequently, the quality of the recycled product. Through the separation of each material into the corresponding class, the material can be fed to a suitable recycling process, resulting in the closing of material cycles. Through this, the recycling of WTBs no longer poses an environmental challenge, but instead becomes a stepping-stone for future applications that may leverage these recovered materials in increasingly innovative ways.

Use of AI tools declaration

The authors declare they have not used Artificial Intelligence (AI) tools in the creation of this article.

Acknowledgments

The work described in this paper has been carried out within the EoLO-HUBs project (https://www.eolo-hubs.eu/) co-funded by the European Union (Agreement No. 01096425 - EoLO-HUBs - HORIZON-CL5-2022-D3-01). Views and opinions expressed are however those of the authors only and do not necessarily reflect those of the European Union or CINEA. Neither the European Union nor the granting authority can be held responsible for them.

Conflict of interest

All authors declare no conflicts of interest in this paper.

References

1. Andersen PD, Bonou A, Beauson J, et al. (2014) Recycling of wind turbines. *DTU Int Energy Rep* 2014: 91–97.
2. Veers PS, Ashwill TD, Sutherland HJ, et al. (2003) Trends in the design, manufacture and evaluation of wind turbine blades. *Wind Energy* 6: 245–259. https://doi.org/10.1002/we.90
3. Brondsted P, Njissen R, Goutianos S (2023) *Advances in wind turbine blade design and materials*. Woodhead Publishing.
4. Hao S, Kuah AT, Rudd CD, et al. (2020) A circular economy approach to green energy: Wind turbine, waste, and material recovery. *Sci Total Environ* 702: 135054. https://doi.org/10.1016/j.scitotenv.2019.135054
5. WindEurope (2024) How to build a circular economy for wind turbine blades through policy and partnerships. Available from: https://windeurope.org/newsroom/press-releases/wind-industry-calls-for-europe-wide-ban-on-landfilling-turbine-blades/
6. David E, Kopac J (2015) Use of separation and impurity removal methods to improve Aluminium

waste recycling process. *Mater Today Proceed* 2: 5071–5079. https://doi.org/10.1016/j.matpr.2015.10.098

7. Guo H, Guo X, Zhang X, et al. (2025) Fault diagnosis of wind turbine based on dual-channel feature aggregation network with attentional mechanism. *Eng Appl Artif Intell* 161: 112291. https://doi.org/10.1016/j.engappai.2025.112291
8. Meng D, Yang H, Yang S, et al. (2024) Kriging-assisted hybrid reliability design and optimization of offshore wind turbine support structure based on a portfolio allocation strategy. *Ocean Eng* 295: 116842. https://doi.org/10.1016/j.oceaneng.2024.116842
9. Yang S, Chen Y (2025) Modelling and analysis of offshore wind turbine gearbox under multi-field coupling. *IJOSM* 2: 52–66. https://doi.org/10.1504/IJOSM.2025.146804
10. Xu S, Abbas Z, Zhang X-G, et al. (2025) Joint optimization of utility and privacy for space-air-ground integrated network task offloading: A Co-D3QN approach. *IEEE Trans Veh Technol*: 1–16. https://doi.org/10.1109/TVT.2025.3623376
11. Gundupalli SP, Hait S, Thakur A (2017) A review on automated sorting of source-separated municipal solid waste for recycling. *Waste Manag* 60: 56–74. https://doi.org/10.1016/j.wasman.2016.09.015
12. Ren S, He K, Girshick R, et al. (2017) Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE Trans Pattern Anal Mach Intell* 39: 1137–1149. https://doi.org/10.1109/TPAMI.2016.2577031
13. He K, Gkioxari G, Dollár P, et al. (2017) Mask R-CNN. In: *Proceedings of the IEEE international conference on computer vision*: 2961–2969. https://doi.org/10.1109/ICCV.2017.322
14. He K, Zhang X, Ren S, et al. (2015) Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE Trans Pattern Anal Mach Intell* 37: 1904–1916. https://doi.org/10.1109/TPAMI.2015.2389824
15. Girshick R (2015) Fast R-CNN. In: *Proceedings of the IEEE international conference on computer vision*: 1440–1448. https://doi.org/10.1109/ICCV.2015.169
16. Redmon J (2016) You only look once: Unified, real-time object detection. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*: 779–788. https://doi.org/10.1109/CVPR.2016.91
17. Liu W, Anguelov D, Erhan D, et al. (2016) SSD: Single Shot MultiBox Detector. *Computer Vision – ECCV 2016* 9905: 21–37. https://doi.org/10.1007/978-3-319-46448-0_2
18. Voulodimos A, Doulamis N, Doulamis A, et al. (2018) Deep learning for computer vision: A brief review. *Comput Intell Neurosci* 2018: 1–13. https://doi.org/10.1155/2018/7068349
19. Chai J, Zeng H, Li A, et al. (2021) Deep learning in computer vision: A critical review of emerging techniques and application scenarios. *Mach Learn Appl* 6: 100134. https://doi.org/10.1016/j.mlwa.2021.100134
20. Zou Z, Chen K, Shi Z, et al. (2023) Object detection in 20 years: A survey. In: *Proceedings of the IEEE* 111: 257–276. https://doi.org/10.1109/JPROC.2023.3238524
21. Vaswani A, Shazeer N, Parmar N, et al. (2017) Attention is all you need. arXiv preprint arXiv:1706.03762.
22. Khan S, Naseer M, Hayat M, et al. (2022) Transformers in vision: A survey. *ACM Comput Surveys (CSUR)* 54: 1–41. https://doi.org/10.1145/3505244
23. Lin T-Y, Maire M, Belongie S, et al. (2014) Microsoft COCO: Common objects in context. *Computer vision - ECCV 2014* 8693: 740–755. https://doi.org/10.1007/978-3-319-10602-1_48

24. Z. Ge, S. Liu, F. Wang, et al. (2021) Yolox: Exceeding yolo series in 2021. arXiv preprint arXiv:2107.08430.
25. Carion N, Massa F, Synnaeve G, et al. (2020) End-to-end object detection with transformers. *Computer Vision – ECCV 2020* 12346: 213–229. https://doi.org/10.1007/978-3-030-58452-8_13
26. Kulshreshtha M, Chandra SS, Randhawa P, et al. (2021) OATCR: Outdoor autonomous trash-collecting robot design using YOLOv4-tiny. *Electronics* 10: 2292. https://doi.org/10.3390/electronics10182292
27. Proença PF, Simões P (2020) TACO: Trash annotations in context for litter detection. arXiv preprint arXiv:2003.06975.
28. Melinte DO, Travediu A-M, Dumitriu DN (2020) Deep convolutional neural networks object detector for real-time waste identification. *Appl Sci* 10: 7301. https://doi.org/10.3390/app10207301
29. Koskinopoulou M, Raptopoulos F, Papadopoulos G, et al. (2021) Robotic waste sorting technology: Toward a vision-based categorization system for the industrial robotic separation of recyclable waste. *IEEE Robot Automat Mag* 28: 50–60. https://doi.org/10.1109/MRA.2021.3066040
30. Mao W-L, Chen W-C, Fathurrahman HIK, et al. (2022) Deep learning networks for real-time regional domestic waste detection. *J Clean Product* 344: 131096. https://doi.org/10.1016/j.jclepro.2022.131096
31. Lin T-Y, Goyal P, Girshick R et al. (2017). Focal loss for dense object detection. *Proceedings of the IEEE international conference on computer vision*: 2980-2988
32. Tan M, Pang R, Le QV (2020) EfficientDet: Scalable and efficient object detection. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*: 10781–10790. https://doi.org/10.1109/CVPR42600.2020.01079
33. G. Jocher, A. Chaurasia, A. Stoken, et al. (2022) ultralytics/yolov5: v7. 0-yolov5 sota realtime instance segmentation. https://doi.org/10.5281/zenodo.7347926
34. Li C, Li L, Jiang H, et al. (2022) YOLOv6: A single-stage object detection framework for industrial applications. arXiv preprint arXiv:2209.02976.
35. Kingma DP, Ba J (2014) Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.
36. Biewald L (2020) Experiment tracking with weights and biases. https://www.wandb.com/