Double Transfer Learning to Detect Lithium-Ion Batteries on X-Ray Images

David Rohrschneider, Nermeen Abou Baker(✉), and Uwe Handmann

Computer Science Institute, Ruhr West University of Applied Sciences,
Luetzowstrasse 5, 46236 Bottrop, Germany
Nermeen.Baker@hs-ruhrwest.de

Abstract. With the soaring popularity of electronic gadgets, Lithium-Ion Batteries (LIB) have witnessed a remarkable surge. The inspiration behind this study arises from the urgent need to automate the identification of batteries in diverse contexts, such as electronic waste recycling facilities or security screening at airports. Ultimately, it strives to minimize health hazards associated with battery recycling by enabling more accurate sorting with minimal human involvement. In this paper, we applied double transfer learning to eight cutting-edge object detectors, unlocking the potential of X-Ray images in recognizing and categorizing electronic mobile devices (EMD) along with their embedded Lithium-Ion batteries (LIB).

Keywords: Benchmark · Object Detection · Transfer Learning · Double transfer Learning · X-Ray Images · Lithium-Ion Batteries

1 Introduction

1.1 Problem Statement

The presence of such substances is noticeable in X-Ray images by darker color at the location of the battery. Based on this fact, there are a couple of applications that can benefit from detecting and sorting EMD and LIB. The usage of EMD has increased significantly in recent years, and it can be concluded that the number of disposals of these devices is also increasing. A study by [1] reported that, before sorting and recycling batteries, each electronic device must first be classified as to whether it contains batteries or not. In this process, there is a proportion of devices where removing batteries is more difficult because they have been glued or welded together. This results in considerable manual time and expenses, and can also pose an increased health risk to personnel, e.g. due to damaged LIBs. Recent research has already inspected that it is possible to classify electronic devices by model on RGB images, capturing each device from a top view and using a convolutional neural network for classification [2].

D. Rohrschneider and N.A. Baker—Equal contribution.

I. Rojas et al. (Eds.): IWANN 2023, LNCS 14134, pp. 175–188, 2023.
https://doi.org/10.1007/978-3-031-43085-5_14

Recognizing the model series can deliver exact information on the number, type, and location of batteries inside the device, but it requires the neural network to already know all possible device models found in the recycling facility, which is why direct inspection of EMD internals using X-Ray images may be more applicable. Another application is the security inspection of passenger baggage of LIB to prevent potential risks of heating and burning during the flight [3].

1.2 Research Gap

Some of the main challenges that cause low recycling rates of Lithium are barriers for sorting, disassembly, and pre-treatment steps, evoked by diversity and non-standardization of LIBs [4]. Being able to detect and classify EMD and LIB on X-Ray images using artificial intelligence, the electronic waste could be processed more efficiently to improve the economical viability of recycling electronic waste. To ensure the passengers' safety within an aircraft for example in the USA, the Federal Aviation Administration states that EMD should be kept in carry-on baggage and, if stored in checked baggage, should be packed to protect them from any outer damage [5]. Utilizing state-of-the-art (SOTA) object detection methods in this use case could enhance the speed and quality of baggage inspection. To our knowledge, none of the previous works tested You Only Look Once (YOLO)v7 [6], YOLOv8 [7] or vision transformer models in detecting EMD and LIB on X-Ray images.

Section 2 investigates related work and the SOTA of deep learning approaches to object detection. Section 3 describes the datasets preparation and model implementation in detail. Then, in Sect. 4 model performance is compared, and the impact of employing double transfer learning is analyzed. Finally, the experimental results are discussed, and future prospects are suggested.

2 State of the Art

2.1 Related Work

In addition to our previous work, which will be described in Sect. 2.2, two other publications have addressed the detection of EMD or batteries on X-Ray images.

The work by [8] introduced the HiXray dataset, which will also be briefly presented in Sect. 2.3, along with the Lateral Inhibition Module. With the module being detached from deep learning models and backbone structures, it can be integrated into existing architectures to enhance a model's performance. The HiXray dataset was tested using three different types of deep learning models: Single Stage Detector (SSD), Fully convolutional one-stage object detection (Fcos), and YOLOv5s [9]. The results show that the suggested module improves the mean Average Precision (mAP) of each of the existing models by an average of 1.5% and achieves the best MAP of 96.8% when combined with YOLOv5.

The authors of [1] deal with the detection and classification of batteries in the context of automating electronic waste recycling. For the experiment, 532 objects of electronic waste were imaged with a Computed Tomography scanner to visualize the batteries within, regardless of external dirt or damage to the object. Each of the 943 batteries was manually labeled, each assigned to one of six battery types, including prismatic and pouch lithium-ion batteries among others. The YOLOv2 model with a LightNet backbone was used for the detection and evaluated by the precision, recall, and the F1-score. The precision of just classifying whether or not an electrical device can be seen on an X-Ray image is 89% and the precision for detecting batteries within the image is 82% for the pouch LIB class and 75% for the cylindrical LIB class.

2.2 Motivation

In our previous work [10], we tested the detection of EMD and LIB using YOLOv5m [9] and the HiXray dataset. Multiple transfers of weights are utilized to detect the EMD and LIB, achieving a precision of 0.94 for detecting EMD and 0.935 for detecting LIB. It is also noted that the performance significantly improves when applying a second knowledge transfer for LIB detection in X-Ray images. Building upon this finding and utilizing the same database, this study incorporates the transfer of weights from the EMD detection task to the LIB detection task. However, the previous work was limited to YOLOv5m only and therefore was bound to the performance of a 1-Stage detector model, which was designed for real-time object detection tasks [9].

2.3 Public X-Ray Datasets

X-Ray image datasets related to object detection include, SIXray [11], OPIXray [12], PIDray [13], HiXray [8], and GDXray Baggage [14]. Solving our problem requires a dataset with appropriate object classes to detect EMD and LIB in X-Ray images. Furthermore, it should have a sufficient number of samples since the batteries contained in EMD appear smaller in relation to surrounding objects and are more difficult to identify. To present an overview of recent X-Ray datasets for object recognition, Table 1 compares their numbers of images and classes" for improved phrasing and readability.

2.4 Methods of Object Recognition

In the past, object recognition models were traditionally categorized into two types: two-stage and one-stage models. In 2020, a new technique called "End-to-End Object Detection with Transformers (DETR)" was introduced [15]. By utilizing the transformers' self-attention mechanism, these models can put objects in a global context of the image and create relationships between them, which helps in finding the final bounding box and classification decisions. Recent SOTA examples of object detection models employing the transformer architecture are the Swin transformer [16] or DINO models [17].

This work tested 8 object detection models, as shown in Table 2.

Table 1. Public X-Ray datasets (MD classes are written in bold)

Dataset	Number of images	Classes
SIXray	1,059,231	Gun, Knife, Wrench, Pliers, Scissors, Hammer
OPIXray	8,885	Folding Knife, Straight Knife, Scissors, Utility Knife, Multi-tool Knife
PIDray	47,677	Gun, Bullet, Knife, Wrench, Pliers, **Powerbank**, Baton, Lighter, Sprayer, Hammer, Scissors, Handcuffs
HiXray	45,364	**Portable Charger 1**, **Portable Charger 2**, **Mobile Phone**, **Laptop**, **Tablet**, Cosmetic, Water, Nonmetallic Lighter
GDXray Baggage	8,150	Handgun, Razor Blade, Shuriken, Pen Case, Clip, Spring, Door Key, Knife

Table 2. Comparison of models with pre-trained weights

Technique	2-Stage	1-Stage					Transformer	
Model	Faster R-CNN [18]	SSD Lite [19]	YOLO v5m	YOLO v7 W6	YOLO v8	Efficient Det D1 [20]	Casc. Mask R-CNN	DINO 4scale
Backbone	Res Net50	Mobile Net v2	YOLO v5	E-ELAN	Darknet -53	Efficient Net-b1 BiFPN	Swin-S	Swin-L
#M Parameters	29.162	4.475	21.2	70.4	25.9	6.6	107	218
COCO mAP	29.3	29.1	45.4	54.9	50.2	38.4	51.9	58.0

3 Materials and Methods

3.1 Selection of Models and Dataset

Models. To compare the impact of transfer learning on the detection of LIB on X-Ray images among the three object detection strategies presented in Sect. 2.4, at least one SOTA model was selected for each approach for the experiment. Considering the limited computing time and capacity available in practical scenarios, the models are selected to achieve a trade-off between computational cost and prediction accuracy. To perform double transfer learning, it is necessary to have access to pre-trained weights from a large and high-quality dataset that is publicly available.

Dataset. To choose a suitable database for this benchmark, the datasets compared in Table 1 were evaluated based on the number of samples and the EMD classes present. The HiXray dataset was selected among them, because it consists of more than 45,000 images in total and 5 different classes of EMD, while the other X-Ray datasets have, at most, one EMD class. It is not publicly available but can be obtained for academic purposes upon request. The set comes with images saved in JPG format and has an average resolution of 1,200 × 900 px, but they vary from image to image. The images are divided into two sets, with 36,295 assigned to the training set and 9,069 to the test set, resulting in a ratio of approximately 4:1. The bounding boxes for each image were manually annotated for the eight different object classes. The descriptions of these bounding boxes are provided in separate text files that correspond to the respective images. On average, there are 2.27 classes per image, indicating that multiple objects can be assigned to the eight object classes [8].

3.2 Dataset Preparation

The HiXray dataset consists of five EMD classes, namely portable charger1, portable charger2, mobile phone, laptop, and tablet, "as well as three additional classes: water, cosmetic, and non-metallic lighter". Since this work focuses on the detection of EMD and batteries, retaining these three classes might adversely affect the results, thus, they are initially excluded from the label files. For the first dataset shown in Fig. 1(a), 12,000 annotated samples and 2, 000 unannotated samples are used for the training split, while 3,000 annotated samples and 500 unannotated samples are used for the testing split, all derived from the original HiXray dataset. Noting that the "Mobile Phone" class has the biggest number of occurrences, which poses a challenge in achieving a balanced distribution.

The second dataset was manually derived as a subset of the remaining images for the battery contained in the EMD. As previously indicated in our previous work [10], the three different LIB classes, namely 'Prismatic LIB,' 'Cylindrical LIB,' and 'Pouch LIB,' were found across different EMD classes, regardless of

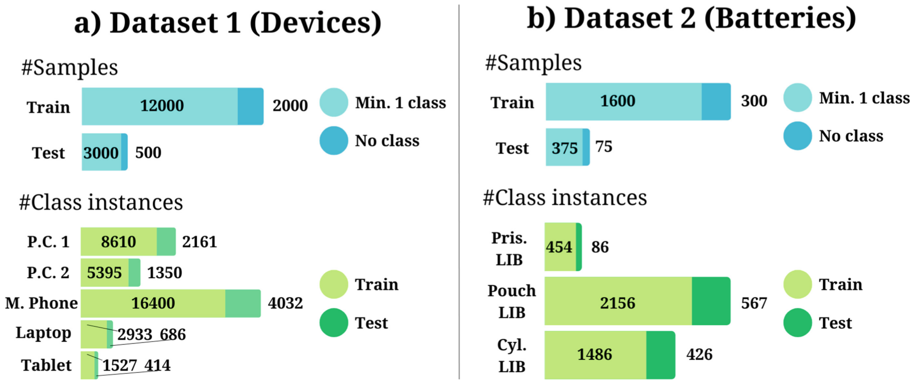

Fig. 1. Distribution of samples and class instances per dataset

their brand or model series. The ‘Pouch LIB’ appears in ‘Mobile Phone’, ‘Laptop’ and ‘Tablet’, ‘Prismatic LIB’ could only be found inside ‘Portable charger1’ and ‘Cylindrical LIB’ was found to appear in ‘Portable charger2’, as well as in some instances of the ‘Laptop’ class. As a result, 300 samples containing only the class ‘Portable Charger2’ and three times 300 samples containing only the classes ‘Mobile Phone’, ‘Laptop’ and ‘Tablet’ are created, along with 400 samples containing only the class ‘Portable Charger1’ since the prismatic LIB appears only once per instance. Another 300 samples without any annotations are added too. The testing set is created likewise, making up a total of 1,900 training and 450 testing samples, which then are manually annotated by drawing a bounding box around each battery instance and discarding the EMD classes. To increase the number of training samples, two augmentations are applied per image, including horizontal flipping and cropping with a zoom rate of 0–50%. The entire process is implemented using Roboflow [21] and the results in the train-test-split are visualized in Fig. 1(b).

3.3 Setting up the Environment

For the implementation, Google Colab Pro+ is used. To train and evaluate the chosen models with pre-trained weights and the custom dataset, we utilized the official code repositories published by the authors. These repositories offer prepared scripts, which are used to perform the training using equal hyperparameters and the evaluation with equal metrics described in the following sections.

YOLO. As stated in Table 2, three models from the YOLO family were tested in this study: YOLOv5m [9], YOLOv7-W6 [6] and YOLOv8m [7]. Since the dataset format is the same for all three models, our two datasets were exported only once using RoboFlow [21].

TensorFlow Object Detection API. To prepare Faster R-CNN [18], SSD Lite [19] and EfficientDet D1 [20] for the training and evaluation in Google Colab Pro+, the TensorFlow Object Detection API [22] is used. The original repository provides a wide range of object detection models and pre-trained weight checkpoint files trained on the COCO 2017 dataset. To simplify the use of the framework, the dataset needs to be converted to the TF-Record format. This format is compatible with the RoboFlow [21], making the conversion process seamless and efficient.

SWIN. With the release of Swin Transformer [16], the Microsoft research team made the source code available, including training scripts and pre-trained weights from the ImageNet-1K dataset [23]. The toolbox requires the dataset to be in COCO-JSON format, which is also supported by RoboFlow.

DINO. The researchers of [17] also published their code based on PyTorch via GitHub along with the weights checkpoint obtained from training on the MS COCO dataset. The pre-trained weights file for the Swin-L backbone needs to be downloaded separately from the official Swin transformer repository, which is also used above. DINO also excepts the dataset to be in COCO-JSON format, which had already been exported before.

Hyperparameters. The entire experiment was conducted using one Google Colab Pro+ notebook per framework. This setup ensures that all tests are conducted on a Tesla T4 GPU with 15 GB of RAM on two Intel(R) Xeon(R) CPUs @ 2.30 GHz sharing 52 GB of RAM. To ensure a fair comparison of the tested object detection models, the hyperparameters are fixed before starting the experiments and then passed to each configuration file. The input image size is set to 640x640 Pixels, the optimizer used was the Stochastic Gradient Descent (SGD), the initial learning rate was set to 0.01 with cosine decay, the confidence threshold during training to 0.001, and Intersection over Union (IoU)-threshold was set to 0.7 for the anchor-based models. Momentum is used with a β value of 0.937, along with the three warm-up epochs for learning rate and momentum term. The number of training epochs is fixed at 20 for fine-tuning on the first and 30 for fine-tuning on the second dataset while early stopping was used to ensure generalization. Considering the varying sizes of models and datasets, as well as the restricted resources in Google Colab Pro+, the batch size for each training was determined based on the capacity of GPU memory. Details on these will be described in Sect. 4.3.

4 Evaluation

4.1 Evaluation Metrics

The performance evaluation of each object detection model was conducted using the original repositories and the COCO API package [24]. The COCO API provides access to commonly used metrics, including mAP@0.5 and mAP@0.5:0.95 per category. The numbers following the @-symbol denote the IoU threshold used to calculate the mAP. The notation 0.5:0.95 indicates that the average mAP is calculated for each IoU threshold between 0.5 and 0.95, with a step size of 0.05. Therefore, the mAP@0.5:0.95 is a more critical metric as it measures the accuracy of predicted bounding box coordinates in relation to the ground truth, often resulting in lower values compared to mAP@0.5. The early stopping method was utilized, and the epoch number at which the highest evaluation

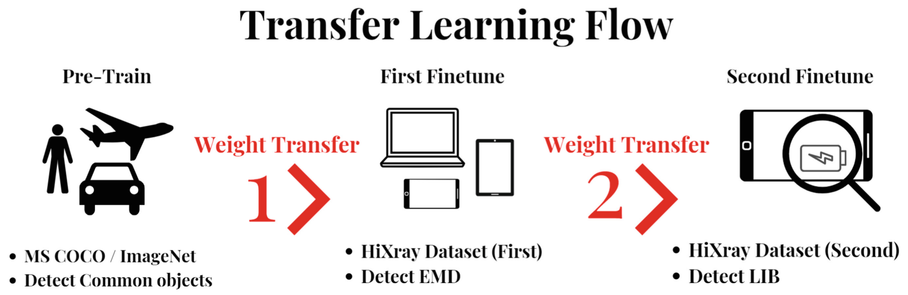

Fig. 2. Visual summary of the finetuning process applied to each model. The two weight transfers are colored red. (Color figure online)

score was achieved, was recorded. Additionally, the inference time was measured in milliseconds, and the mean epoch time was measured in hours. These metrics were compared by considering the batch sizes used for each training.

4.2 Evaluation Strategy

After setting up all the frameworks and datasets, the pre-trained models are fine-tuned on the first dataset for 20 epochs. At this point, the first knowledge transfer from the general task of detecting common objects within the MS COCO dataset to the specific task of detecting EMD on X-Ray images is performed, as can also be seen in Fig. 2.1. The COCO evaluation metrics are calculated on the testing split after each training epoch and displayed in the console. Additionally, a checkpoint file is created after each epoch, and the early stopping method is employed to select the checkpoint state with the highest average mAP across all classes for the second training, detecting three classes of batteries on X-Ray images. For this, the second dataset, which was introduced in Sect. 3.2, is uploaded into the notebooks. Next, the training is started using the same hyperparameters, the best checkpoint file from the previous training, and setting the epochs to 30. By doing so, the second knowledge transfer takes place by using the weights from the task of detecting EMD on X-Ray images and finetuning each model to predict bounding boxes for the batteries inside of the EMD. This transfer of weights is pointed out in Fig. 2.2. Once again, the evaluation metrics are computed after each epoch and the training is stopped when there are no significant improvements in the average mAP across all classes. Following each training iteration in the experiment, an additional evaluation is conducted on the testing set using a batch size of 1 to obtain mAP values for a valid model comparison. Furthermore, the final training speed is calculated by dividing the elapsed time taken to reach the best checkpoint by the number of epochs. Finally, if the framework does not display the inference time per image, an individual image inference is performed to measure the time in milliseconds per image.

4.3 Benchmark Results

Detection of EMD. Table 3 displays the evaluation results for the detection of EMD on X-Ray images.

Table 3. Results from the first transfer: Detecting EMD on X-Ray images

Model	mAP@ 0.5	mAP@ 0.5:0.95	Batch size	Best epoch	Ø Epoch time	Inference time
YOLOv5m	0.968	**0.605**	32	20	0.13 h	17 ms
YOLOv7-W6	0.969	0.598	16	20	**0.078 h**	**13.6 ms**
YOLOv8m	0.973	0.585	32	20	0.28 h	15 ms
SSD Lite	0.941	0.471	16	20	0.135 h	35.1 ms
EfficientDet D1	0.939	0.425	8	20	0.407 h	68.7 ms
Faster R-CNN	0.96	0.59	4	20	0.262 h	110.6 ms
Cas. Mask R-CNN (Swin-S)	0.959	0.581	4	20	2.33 h	57 ms
DINO (Swin-L)	**0.976**	0.562	2	**8**	3.73 h	700 ms

As shown in Table 3, the models achieved a minimum mAP@0.5 of 0.939 when predicting EMD on X-Ray images using their pre-trained weights. The DINO model with a Swin-L backbone performs the best out of the eight models, with a value of 0.976, and thus having an average miss-prediction rate of 2.4%. DINO achieves its best evaluation results at epoch 8 of 20 and attains the highest mAP@0.5 among all the experiments. On the other hand side, DINO can only be trained with a batch size of 2, resulting in the longest average epoch time of 3.73 h and the slowest inference speed of 700 ms per image. This happens due to its big number of parameters, which were compared in Table 2. Similarly, the Cascade Mask R-CNN model with Swin-S transformer backbone has the second-longest average epoch time of 2.33 hours but is more than 12 times faster than DINO when it comes to a single image inference. In contrast to this, the 2-Stage detector Faster R-CNN with a ResNet50 backbone shows up nearly 10 times smaller average epoch time and higher mAP scores, but an inference speed almost twice as long. Among the 1-Stage detectors, the YOLO models, despite having fewer parameters, can compete with the larger vision transformer models in terms of mAP results. Each of the three tested YOLO models achieved higher mAP@0.5:0.95, could be trained with larger batch size, and exhibited faster epoch and inference times compared to the transformer models. In particular, the recently published model YOLOv8m could achieve a mAP@0.5 of 0.973, exhibiting only 0.31% lower accuracy and more than 46 times faster when performing a single image inference compared to the DINO vision transformer. Furthermore, the YOLO models surpass the other 1-Stage detectors (SSD-Lite and EfficientDet-D1) in terms of mAP@0.5, mAP@0.5:0.95, and inference time.

Detection of LIB. Table 4 shows the evaluation results for the detection of LIB on X-Ray images.

Table 4. Results from the second transfer: Detecting Batteries on X-Ray images

Model	mAP@ 0.5	mAP@ 0.5:0.95	Batch size	Best epoch	Ø Epoch time	Inference time
YOLOv5m	0.928	0.746	32	30	**0.021** h	18 ms
YOLOv7-W6	0.932	0.731	32	30	0.033 h	**13.5** ms
YOLOv8m	0.94	**0.753**	32	24	0.06 h	14.8 ms
SSD Lite	0.827	0.516	16	30	0.074 h	30 ms
EfficientDet D1	0.86	0.547	8	30	0.207 h	50 ms
Faster R-CNN	0.863	0.551	4	30	0.198 h	109 ms
Cas. Mask R-CNN (Swin-S)	0.944	0.72	8	20	0.245 h	67 ms
DINO (Swin-L)	**0.947**	0.727	2	**3**	2.82 h	660 ms

Using the weights from the first training to train the models on detecting the three LIB classes in X-Ray images, the results are noted in Table 4. It is worth mentioning that the second dataset contains fewer samples than the first, as previously shown in Fig. 1. Therefore, the batch size could be raised in the cases of YOLOv7-W6 and Cascade Mask R-CNN with Swin-S backbone. A similar pattern to the results in Table 3 can be observed in the current table. Starting with the DINO model, it is able to achieve the highest mAP@0.5 score with 0.947 at the early epoch of 3, which is also the highest value among all models. Following closely, the second vision transformer model, Cascade Mask R-CNN with Swin-S backbone, achieves a mAP@0.5 value of 0.944 at the 20th epoch. Interestingly, its average epoch time is less than one-tenth of DINO's average epoch time, likely due to the fact that the batch size in the second training is twice as large as in the first training. The 2-Stage model Faster R-CNN remains the second slowest model when it comes to the inference per image speed and is, along with SSD-Lite and EfficientDet-D1, almost 10% less precise than the YOLO and vision transformer models in terms of both mAP categories. Among the YOLO models, YOLOv8m achieves the highest mAP@0.5 of 0.94 and outperforms all other models with a mAP@0.5:0.95 of 0.753. On the other hand, it has the slowest average epoch time compared to YOLOv7-W6 and YOLOv5m in both experiments but is still faster than the remaining five models. Moreover, the mAP@0.5 values, particularly for SSD-Lite, EfficientDet-D1, and Faster R-CNN, are lower than in the first training with the task of detecting EMD on X-Ray images and the mAP@0.5:0.95 values are higher. One possible reason for this is the presence of smaller and more occluded object instances, such as the cylindrical LIBs. Figure 3 emphasizes this aspect in a side-by-side comparison of three original images from the dataset.

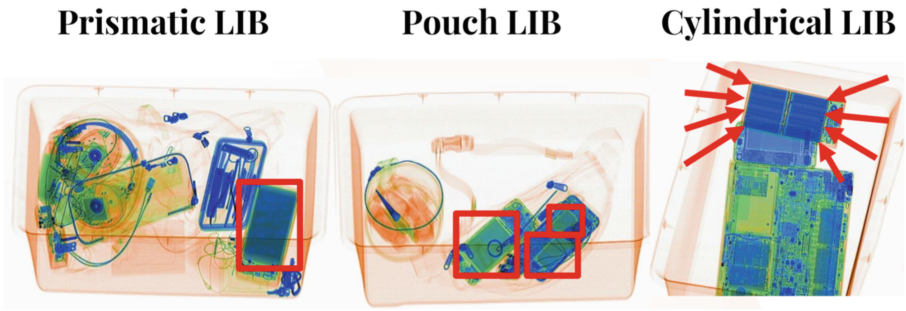

Fig. 3. Side-by-side comparison of the three different battery instances (Marked by red rectangles and arrows). (Color figure online)

To further investigate this phenomenon, Table 5 provides a closer examination of the mAP values, along with the average image proportion of a bounding box per category.

It is noticeable, that the cylindrical LIB is not predicted as accurately as the other classes in terms of mAP@0.5:0.95 for all models. Table 5 provides additional details on why the three models mentioned earlier achieve significantly lower mAP results. SSD Lite, EfficientDet-D1, and Faster R-CNN achieve a maximum score of 0.753 for mAP@0.5 and 0.361 for mAP@0.5:0.95. Since the cylindrical LIB is approximately one-third the size of the other two classes, meaning that

Table 5. Comparison of bounding box proportion and mAP per class

	Pouch LIB		Prismatic LIB		Cylindrical LIB	
Øbox/image	1.69%		1.68%		0.57%	
	mAP@ 0.5	mAP@ 0.5:0.95	mAP@ 0.5	mAP@ 0.5:0.95	mAP@ 0.5	mAP@ 0.5:0.95
YOLOv5m	0.966	**0.816**	0.901	0.799	0.917	0.624
YOLOv7-W6	**0.975**	0.809	0.894	0.785	0.926	0.6
YOLOv8m	0.962	0.806	0.922	0.814	**0.936**	**0.638**
SSD Lite	0.91	0.595	0.904	0.622	0.667	0.33
EfficientDet D1	0.929	0.642	0.948	0.696	0.703	0.303
Faster R-CNN	0.923	0.627	0.912	0.665	0.753	0.361
Cas. Mask R-CNN (Swin-S)	0.964	0.767	0.935	0.793	0.934	0.601
DINO (Swin-L)	0.958	0.78	**0.969**	**0.815**	0.913	0.586
Average	0.948	0.73	0.923	0.749	0.844	0.505

a single instance's bounding box makes up only 0.57% of the image it is found in, there could be a close relationship to the models detecting them with less precision. However, the other models can overcome this obstacle, with YOLOv8m reaching a maximum score of 0.936 mAP@0.5.

4.4 Double Transfer Learning

To address the question of how well the double transfer learning applies to the task of detecting EMD and LIB on X-Ray images, the loss function trend of all the 8 individual trainings is visualized in Fig. 4. The loss values are normalized to a range between 0 and 1 for each model using Min-Max-Normalization, which involves taking the minimum and maximum values from the combined set of EMD and LIB training losses. The loss graphs displayed in blue correspond to the first training to detect EMD, while the red graphs represent the second training to detect LIB. This benchmark evaluates a total of eight different models, with each model having one blue and one red graph, resulting in a total of 16 loss graphs displayed in Fig. 4.

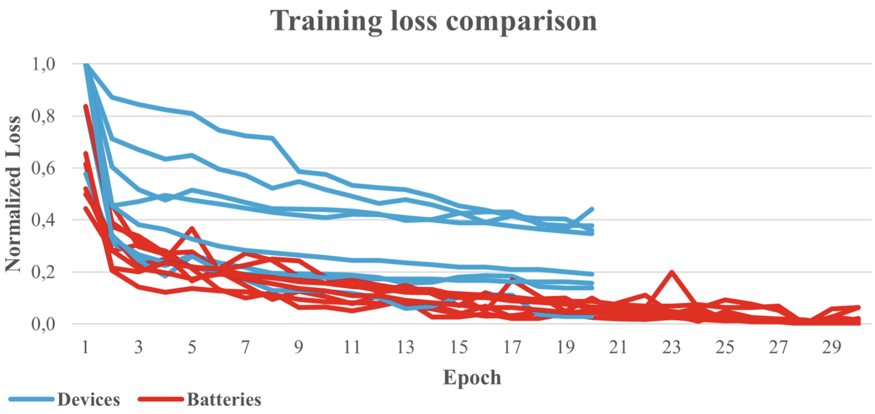

Fig. 4. Comparison of normalized training loss between detecting devices and batteries (Color figure online)

The graphic demonstrates that during the first training, the majority of loss values are higher compared to the second training after transferring weights for the second time. Additionally, it can be observed that in most of the red graphs, the initial loss value in epoch 1 is lower. This suggests that reusing weights from the first training on EMD leads to lower losses and, consequently, higher prediction accuracy values at the early stage of the second training. This gradual introduction allows the model to learn to detect more specific and smaller object categories, such as the LIB within the EMD. This is particularly useful for vision transformer models, as they typically require a substantial amount of data to achieve performance comparable to convolutional neural networks [25]. Transfer learning, or in this case, double transfer learning, can help to reduce the amount of data needed for object recognition from a new context.

5 Summary and Outlook

In summary, it is possible to outperform the SOTA in detecting EMD and LIB on X-Ray images using pre-trained models, regardless of whether a 1-Stage, 2-Stage, or vision transformer model is employed. The results in Sect. 4.3 demonstrate that, with the utilization of double transfer learning, a mAP@0.5 value of 0.947 after the third training epoch in the case of DINO could be achieved. This can be a significant advantage if the training process of a complex model is computationally expensive and there are limited resources available for the task of detecting batteries whether in a recycling facility or at an airport security control. The use of multiple weight transfer, particularly in the case of vision transformers such as Swin or DINO, shows promise as a method to address the challenge of requiring extensive data for training these types of models. Furthermore, it has become evident, that the YOLO models, especially YOLOv8m, can achieve nearly equal performance with a tenth of the number of parameters compared to DINO, resulting in a more than 45 times faster single image inference speed. Although our previous work [10] demonstrated the positive impact of double weight transfer on LIB detection in X-Ray images, certain models, notably SSD-Lite, EfficientDet-D1, and Faster R-CNN, faced challenges in detecting smaller cylindrical LIB instances. This consideration is crucial in tasks such as automated battery sorting or security inspections. On one hand, it could be a limitation that there is no guarantee that every EMD or LIB is correctly identified on an X-Ray image. On the other hand, relying solely on manual inspection by humans can also result in errors and incorrect decisions. Therefore, it becomes crucial to establish precise criteria in the future for determining the required level of accuracy for neural networks.

For future work in this field, there are several avenues to explore for improving performance. Firstly, utilizing the entire HiXray dataset, as well as considering other versions of the specific model series, can provide more comprehensive and diverse training data, potentially leading to enhanced performance. Additionally, upgrading to better hardware resources can also contribute to more efficient training and inference processes. Furthermore, a new dataset with real X-Ray images of electronic waste can be created, which can be beneficial for the context of automating battery sorting in recycling facilities. Object recognition using machine learning models could also act as a part of an EMD or LIB detection pipeline, in which multiple sensors like serial number detectors or tools for weight and chemical composition analysis are combined to improve performance and stability. Lastly, the classification of smartphone model series using deep learning was also covered in previous research [2], and finding an appropriate combination of these techniques could result in more efficient solutions, as well as a lower health risk for the human being.

References

1. Sterkens, W., Diaz-Romero, D., Goedemé, T., Dewulf, W., Peeters, J.R.: Detection and recognition of batteries on x-ray images of waste electrical and electronic equipment using deep learning. Resour. Conserv. Recycl. **168**, 105246 (2021)
2. Abou Baker, N., Stehr, J., Handmann, U.: Transfer learning approach towards a smarter recycling. In: Pimenidis, E., Angelov, P., Jayne, C., Papaleonidas, A., Aydin, M. (eds.) ICANN 2022. LNCS, vol. 13529, pp. 685–696. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-15919-0_57
3. Dangerous goods. https://www.easa.europa.eu/en/domains/passengers/dangerous-goods. Accessed 10 Feb 2023
4. Ma, X., Azhari, L., Wang, Y.: Li-ion battery recycling challenges. Chem **7**(11), 2843–2847 (2021)
5. Lithium batteries in baggage. https://www.faa.gov/newsroom/lithium-batteries-baggage. Accessed 22 July 2022
6. Wang, C.-Y., Bochkovskiy, A., Liao, H.-Y.M.: YOLOv7: trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. arXiv preprint arXiv:2207.02696 (2022)
7. Jocher, G., Chaurasia, A., Qiu, J.: YOLO by Ultralytics (2023). https://ultralytics.com/. Accessed 06 Feb 2023
8. Tao, R., et al.: Towards real-world x-ray security inspection: a high-quality benchmark and lateral inhibition module for prohibited items detection. In: 2021 IEEE/CVF International Conference on Computer Vision (ICCV), pp. 10903–10912 (2021)
9. YOLOv5 and Vision AI. https://ultralytics.com/. Accessed 22 July 2022
10. Abou Baker, N., Rohrschneider, D., Handmann, U.: Battery detection of xray images using transfer learning. In: The 30th European Symposium on Artificial Neural Networks (ESANN 2022), (Bruges, Belgium), pp. 241–246 (2022)
11. Miao, C., et al.: Sixray: a large-scale security inspection x-ray benchmark for prohibited item discovery in overlapping images. in: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2114–2123 (2019)
12. Wei, Y., Tao, R., Wu, Z., Ma, Y., Zhang, L., Liu, X.: Occluded prohibited items detection: an x-ray security inspection benchmark and de-occlusion attention module. CoRR, abs/2004.08656 (2020)
13. Wang, B., Zhang, L., Wen, L., Liu, X., Wu, Y.: Towards real-world prohibited item detection: a large-scale x-ray benchmark. In: 2021 IEEE/CVF International Conference on Computer Vision (ICCV), pp. 5392–5401 (2021)
14. Mery, D., et al.: Gdxray: the database of x-ray images for nondestructive testing. J. Nondestr. Eval. **34**, 1–12 (2015)

15. Carion, N., Massa, F., Synnaeve, G., Usunier, N., Kirillov, A., Zagoruyko, S.: End-to-end object detection with transformers. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12346, pp. 213–229. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58452-8_13
16. Liu, Z., et al.: Swin transformer: hierarchical vision transformer using shifted windows. arXiv preprint arXiv:2103.14030 (2021)
17. Zhang, H., et al.: Dino: Detr with improved denoising anchor boxes for end-to-end object detection (2022)
18. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: towards real-time object detection with region proposal networks. In: Advances in Neural Information Processing Systems, vol. 28 (2015)
19. Liu, W., et al.: SSD: single shot multibox detector. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9905, pp. 21–37. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46448-0_2
20. Tan, M., Pang, R., Le, Q.V.: Efficientdet: scalable and efficient object detection. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 10778–10787 (2020)
21. RoboFlow. https://roboflow.com/. Accessed 22 July 2022
22. Huang, J., et al.: Speed/accuracy trade-offs for modern convolutional object detectors. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 7310–7311 (2017)
23. Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., Fei-Fei, L.: Imagenet: a large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition, pp. 248–255 (2009)
24. Lin, T.-Y., et al.: Microsoft COCO: common objects in context. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8693, pp. 740–755. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10602-1_48
25. Beal, J., Kim, E., Tzeng, E., Park, D.H., Zhai, A., Kislyuk, D.: Toward transformer-based object detection. CoRR, vol. abs/2012.09958 (2020)