

Article

Comparison of Machine Learning Approaches for Robust and Timely Detection of PPE in Construction Sites

Roxana Azizi, Maria Koskinopoulou *  and Yvan Petillot 

School of Engineering and Physical Sciences, Heriot-Watt University, Edinburgh EH14 4AS, UK;
ra2064@hw.ac.uk (R.A.); y.r.petillot@hw.ac.uk (Y.P.)

* Correspondence: m.koskinopoulou@hw.ac.uk

Abstract: Globally, workplace safety is a critical concern, and statistics highlight the widespread impact of occupational hazards. According to the International Labour Organization (ILO), an estimated 2.78 million work-related fatalities occur worldwide each year, with an additional 374 million non-fatal workplace injuries and illnesses. These incidents result in significant economic and social costs, emphasizing the urgent need for effective safety measures across industries. The construction sector in particular faces substantial challenges, contributing a notable share to these statistics due to the nature of its operations. As technology, including machine vision algorithms and robotics, continues to advance, there is a growing opportunity to enhance global workplace safety standards and mitigate the human toll of occupational hazards on a broader scale. This paper explores the development and evaluation of two distinct algorithms designed for the accurate detection of safety equipment on construction sites. The first algorithm leverages the Faster R-CNN architecture, employing ResNet-50 as its backbone for robust object detection. Subsequently, the results obtained from Faster R-CNN are compared with those of the second algorithm, Few-Shot Object Detection (FsDet). The selection of FsDet is motivated by its efficiency in addressing the time-intensive process of compiling datasets for network training in object recognition. The research methodology involves training and fine-tuning both algorithms to assess their performance in safety equipment detection. Comparative analysis aims to evaluate the effectiveness of novel training methods employed in the development of these machine vision algorithms.



Citation: Azizi, R.; Koskinopoulou, M.; Petillot, Y. Comparison of Machine Learning Approaches for Robust and Timely Detection of PPE in Construction Sites. *Robotics* **2024**, *13*, 31. <https://doi.org/10.3390/robotics13020031>

Academic Editor: Fernando Torres

Received: 10 January 2024

Revised: 6 February 2024

Accepted: 12 February 2024

Published: 16 February 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: workplace safety; construction industry; machine learning algorithms; object detection; faster R-CNN; few shot object detection; robotics

1. Introduction

On a global scale, workplace safety emerges as a paramount concern, with statistics underscoring the pervasive impact of occupational hazards. The International Labour Organization (ILO) reports an alarming annual toll of 2.78 million work-related fatalities worldwide, accompanied by an additional 374 million incidents of non-fatal workplace injuries and illnesses [1]. These occurrences impose substantial economic and societal burdens, underscoring the imperative for comprehensive safety measures across various industries. The construction sector, given its operational characteristics, significantly contributes to these statistics. With technological advancements, including the integration of machine vision algorithms and robotics, there exists a growing opportunity to elevate global workplace safety standards and curtail the human toll of occupational hazards on an international scale. Particularly, in the UK, 470,000 workers suffer skeletal illnesses due to their jobs and 142 are reported as fatally injured every year according to [2]. Some of these risks are due to the lack of supervisors who force or direct people to wear safety devices or prevent them from entering restricted areas.

Effective supervision on construction sites involves monitoring various aspects, including worker behavior, adherence to safety protocols, equipment operation, and regulatory

compliance. Traditionally, human supervisors shoulder the responsibility of overseeing these activities. However, the limitations of human monitoring, such as potential fatigue, supervision errors, and the inability to monitor every corner of an extensive construction site have prompted the exploration of motorized and robotic solutions.

Machine Learning algorithms, as a subset of artificial intelligence, have demonstrated remarkable abilities in pattern recognition, object detection, and decision making based on visual data [3–5]. Leveraging cameras and sensors strategically placed on construction sites, these algorithms can process visual information in real time to locate safety equipment, monitor its use, and help ensure consistent application of safety regulations. By seamlessly integrating into existing surveillance systems, machine vision algorithms can enhance the effectiveness of human supervisors and provide an additional layer of security oversight. Accordingly, accurate and timely detection and identification of safety and Personal Protective Equipment (PPE) such as helmets, highly visible jackets, and safety harnesses is critical to preventing accidents and minimizing hazards on construction sites. Robots that can cue workers 24/7 without tiring or distracting them can save workers' lives.

This paper aims to employ state-of-the-art machine learning methodologies for robust and timely detection of safety equipment to ensure human safety on construction sites. These methodologies can then be installed on robotic platforms for real-time and infinite inspection. Robotic automation of inspection duties can alleviate current manual process constraints and inefficiencies to enable periodic and reliable construction observation. Two algorithms for the accurate detection of safety equipment are explored and compared in this paper. The first algorithm is Faster R-CNN [6] with backbone ResNet-50 [7] to perform object detection and then compare the result of Faster R-CNN to that of Few-Shot Object Detection (FsDet) [8] which is our second algorithm. FsDet is chosen due to the time-intensive process of compiling datasets for network training in object recognition. Meanwhile, at the same time, another network is trained to compare outcomes, evaluating the effectiveness of novel training methods. Another aim for selecting FsDet is whether can it distinguish a safety gadget like a helmet from other similar caps or other similar things. The result is highly acceptable as it can recognize the safety gadgets by 50 percent from similar goods; this object detection method has not received much attention. Moreover, the number of datasets for safety gadgets is vast, and having such an opportunity to detect desired objects in short time by introducing at last 10 images can be counted as a major achievement.

Overall, the contributions of the paper can be summarized as follows:

- **Comparative Evaluation:** The paper provides a comprehensive comparison between Fast R-CNN and Few-Shot Object Detection methods for the robust and timely detection of Personal Protective Equipment (PPE) in construction sites. It systematically evaluates the strengths and weaknesses of both approaches, offering insights into their performance in the specific context of PPE detection.
- **Robustness Assessment:** The research contributes by conducting a thorough assessment of the robustness of Fast R-CNN and Few-Shot Object Detection in the challenging environment of construction sites. It delves into factors such as occlusions, varying lighting conditions, and diverse PPE types, shedding light on the methods' adaptability and reliability.
- **Application to Construction Site Safety:** This work extends the application of object detection techniques to the critical domain of construction site safety. By focusing on the detection of PPE, the paper addresses a crucial aspect of occupational health and safety, highlighting the potential impact of these methods in preventing accidents and ensuring compliance with safety regulations.
- **Insights for Practical Implementation:** The paper offers practical insights for the implementation of object detection systems in real-world construction site environments. It discusses considerations for deployment, challenges encountered during implementation, and recommendations for optimizing the performance of both Fast R-CNN and Few-Shot Object Detection models in construction site scenarios.

2. Related Work

Construction has been a fundamental human activity, and the integration of robots has significantly alleviated the burdens associated with construction tasks. Robots play versatile roles in construction, ranging from executing specific tasks to constructing entire structures, ensuring safety compliance, and performing demanding construction activities. In the context of safety, robots are tasked with detecting safety objects, such as helmets and high-visibility jackets, as highlighted in projects like [9,10]. Intelligent monitoring systems, addressing occasional non-compliance with Personal Protective Equipment (PPE) regulations, are deemed essential [10]. To ensure safety compliance, machine learning algorithms like YOLO (You Only Look Once) architectures, designed to recognize specific safety classes, are employed [11]. YOLO [12] approaches object detection as a regression problem, determining spatially separated bounding boxes and their corresponding class probabilities in a single neural network evaluation. This unified detection pipeline allows for direct end-to-end optimization, enabling the network to predict bounding boxes and class probabilities directly from full images. Object detection can be obtained by bi-directional cross-attention layers [13] in which early exits are eliminated and replaced with a classification branch. Obviously, feature extraction [3] is a key of object detection that can be found by varying orientation features the convolution rotates adaptively. In a more modern way, transformers (DETRs) employ a group of object queries to anticipate a collection of bounding boxes. These boxes are then arranged based on their confidence scores for classification, and the top-ranked predictions are chosen as the ultimate detection outcomes for the input image [14]. In DETRs, to remove the same detections, a hand-crafted NMS (non-maximum suppression) is used, and to improve accuracy “a hybrid matching scheme that combines the original one-to-one matching branch with an auxiliary one-to-many matching branch during training” is introduced [15].

Another approach involves reinforcement learning, where robots can learn through feedback, and imitation learning, which has significant implications for robotics. The intersection of robotics, reinforcement learning, and construction is explored in [16], providing a comprehensive review of research findings. Dataset preparation is pivotal in these endeavors, as seen in [17], where a combined dataset of collected and captured images was utilized. Regular monitoring of the built environment using robots is crucial for overseeing maintenance, assessing construction quality, monitoring progress, creating as-built models, and conducting safety inspections [18]. The evaluation of robots on construction sites, as discussed in [19], is essential to anticipate and proactively address challenges in practical robotic construction scenarios. Overall, the integration of robots in construction brings advancements in safety, efficiency, and proactive problem solving.

This paper aims to explore machine learning technologies to inspect construction sites to ensure compliance with worker personal safety regulations. The main part of the work focuses on object detection algorithms to facilitate the timely and robust detection of PPE. Various techniques have been developed for object detection, with approaches like those extracting multiple low-level features from an image before contextualizing them at a higher level showing exceptional performance [20]. In this landscape, Fast R-CNN emerges as a streamlined and efficient improvement over the SPPnet and R-CNN methodologies. Notably, it introduces a novel layer called ROI Pooling, which aggregates feature vectors of uniform length from individual proposals (or Regions of Interest—ROIs) within an image. This architectural enhancement marks a significant simplification compared to the three-step process of the original R-CNN [21].

However, for real-time object detection scenarios, the state-of-the-art capabilities of Convolutional Neural Networks (CNNs) are still widely recognized. The power of CNNs lies in their ability to automatically learn hierarchical representations of visual features from data. Despite their success, there are challenges in adapting CNNs to Few-Shot Learning scenarios where limited examples are available for novel classes. In contrast, Few-Shot Learning (FSL) approaches aim to address this limitation by training models to recognize objects with very few examples per class. The comparison between the

established efficacy of CNNs and the evolving landscape of Few-Shot Learning for real-time object detection becomes critical in understanding the trade-offs and advancements in the pursuit of accurate and efficient visual recognition systems.

Yadav's study [22] focuses on the timely enforcement of COVID-19 safety measures, employing deep learning for real-time detection of safe social distancing and face mask adherence. Cha et al. in [23] emphasize autonomous structural visual inspection, using region-based deep learning to detect multiple damage types in infrastructure, particularly in the context of bridges. In [24], authors propose object detection techniques for jobsite observation, aiming to enhance monitoring and reduce computing needs. Each study targets specific safety and inspection challenges, ranging from pandemic-related guidelines and rail integrity to structural damage detection and construction site monitoring.

In the specific context of monitoring construction environments, Tang et al. in [25] proposed a learned Human–Object Interaction (HOI) recognition model for personal protective equipment compliance checking by explicitly classifying worker–tool interactions. The proposed detection model was trained on a newly constructed image dataset for construction sites, achieving 52.9% average mean precision for 10 object categories and 89.4% average precision for detecting workers. Meanwhile, the authors of [26] introduced SODA and contributed to the field by providing a large-scale open-source object detection dataset for deep learning in construction. Recently, a robotic system equipped with perception sensors and intelligent algorithms was proposed in [27] to help construction supervisors remotely identify construction materials, detect component installations and defects, and generate report of their status and location information. In this work, building of an information model was employed for mobile robot navigation and localization while real-time object detection was realised with a data- and information-driven approach by way of incorporating offline training data, sensorial information, and the output of the building information model.

3. Object Detection in Construction Sites

Insufficient oversight is identified as a primary factor contributing to accidents and fatalities within construction sites. To address this challenge, implementing robotic systems capable of identifying and rectifying instances of non-compliance among workers regarding safety equipment usage is proposed. The initial step involves object detection, a crucial task within this approach. By leveraging advanced technologies like computer vision and machine learning, these robots can effectively pinpoint situations where workers deviate from prescribed safety protocols, thereby promoting a safer working environment.

The paper discusses two state-of-the-art methodologies for object detection, emphasizing the significance of the dataset's image volume in these approaches, excluding sections on meta-learning. Two separate networks for training to compare their performance and highlight the strengths and weaknesses of each approach are used. To ensure meaningful analysis, both networks operate under identical conditions and share the same backbone.

3.1. Faster R-CNN

There is lack of systematic evaluation of emerging paradigms in machine learning, particularly in the domain of Few-Shot Learning, concerning PPE detection. The complexity of PPE comprises various subcategories with dynamic variations in design, color, and configuration, making accurate identification challenging. The text underscores the need for meticulous training for each PPE class, including high-visibility jackets with diverse attributes, and highlights the labor-intensive process of curating an image dataset with detailed labeling and bounding boxes for traditional ML techniques. Training is performed on a PC with GPU Geforce with Cuda 3050 Ti to maintain the same conditions.

The Faster R-CNN network requires a backbone architecture to serve as the foundational convolutional layer. In our project, we deliberately opt for the ResNet-50 architecture as the chosen backbone. This decision is driven by ResNet-50's proven ability to intricately extract features from input images. The selection of ResNet-50 is grounded in its demon-

strated capability to meticulously capture detailed features, thereby significantly enhancing the network's proficiency in comprehensively and effectively capturing salient aspects embedded within the images.

3.1.1. Dataset for FRCNN

The surveillance of worker safety equipment has garnered significant attention, resulting in the availability of numerous datasets. However, each dataset is inherently designed for specific conditions, incorporating variations such as indoor and outdoor settings or limited brightness ranges. In an effort to construct a comprehensive dataset encompassing three fundamental classes—"helmet", "safety vest", and "face"—for both indoor and outdoor scenarios, we amalgamate two composite datasets, namely Pictor-v3 [28] and Apache [29]. Pictor-v3 consists of 1500 annotated images and around 4700 instances of workers wearing various combinations of PPE components, while Apache [29] contains 7350 images and 14,700 annotation masks. The images are divided by 70, 20, and 10 percent, respectively, for train/test/val. Most of them are 640×40 or 1221×921 , but generally, the minimum size is 480 and the maximum is 1223. Following this, a thorough review of the dataset involves plotting images alongside their corresponding bounding boxes to assess the accuracy of box delineations and labeling.

A critical step follows, which includes fixing inaccuracies in annotations and bounding box labels. Due to the merging of two separate datasets, an inherent imbalance occurs across the three classes. To address this, an additional processing of images is required to achieve a balance of cases for each individual class which are helmet, safety vest, and face. This process involves the careful selection of images to ensure parity, ultimately enhancing the representational efficiency of the dataset and laying the foundation for subsequent analyses in the field of security equipment detection.

After selecting images and making sure that labels and bounding boxes are fitted, it is time to make changes in dataset annotations to make them usable for the Faster R-CNN network. Pictor is in {image name, xmin, ymin, xmax, ymax} in a CSV file. Apache 2 is in {class, x_center y_center, width, height} in a text file. Faster R-CNN needs {Image name, width, height, class, xmin, ymin, xmax, ymax} [6]. To reach the dataset which is suitable for Faster R-CNN, a Python code is implemented and tested.

After the selection of images and ensuring the fitting of labels and bounding boxes, the next step involves modifying the dataset annotations to make them compatible with the Faster R-CNN network. Pictor's format is in image name, xmin, ymin, xmax, ymax within a CSV file, while Apache 2 follows {class, x_center y_center, width, height} in a text file. For Faster R-CNN compatibility, the required format is {Image name, width, height, class, xmin, ymin, xmax, ymax} [6]. To achieve this dataset format suitable for Faster R-CNN, a Python code is developed and tested.

To select the images from the dataset for training and testing separately, we can use *Shuffle = True* in implementing the code to select images for training and testing datasets at random. Then, we use *torch.utils.data.random_split* to split the dataset. Through this command, the dataset becomes split and the output is defined as *train_set*, *val_set*. Then, we convert the dataset of each image to the tensor; in fact, by applying the *torch.tensor* on all data, we can copy the data and convert them into the tensor array. By converting all tensor arrays into a dictionary, the data are ready to be an input of DataLoader. Then, using *torch.utils.data.DataLoader*, the DataLoader creates an iterable interface around the Dataset, facilitating convenient access to individual samples for further processing and analysis. This abstraction assists in efficiently handling large datasets during machine learning tasks, enhancing data manipulation and model training. By encapsulating the dataset within an iterable structure, the DataLoader simplifies the task of retrieving and utilizing data samples, ultimately contributing to the smoother and more effective development of machine learning models and algorithms [30].

It should be mentioned that Pictor-v3 contains around 1500 annotated images and around 4700 instances of workers wearing various combinations of PPE components [28], and Apache [29] is involved in training 7350 images and 14,700 annotation files.

After selecting images and making sure that labels and bounding boxes are fitted, it is time to make changes in dataset annotations to make it usable for the Faster RCNN network. Pictor is in {image name, xmin, ymin, xmax, ymax} in a CSV file. Apache 2 is in {class, x_center y_center, width, height} in a text file. Faster R-CNN needs {Image name, width, height, class, xmin, ymin, xmax, ymax} [6]. To arrive at the dataset suitable for Faster Rcn, a Python code is implemented and tested. Codes can be found in our GitHub repository, <https://github.com/RoxanaAz/dataset-converions> (accessed on 1 August 2023).

3.1.2. Implementation Code for Faster R-CNN

As mentioned above, we implemented a new code to modify and adapt the annotations of our dataset to a new format. Following this, we created the dataset in the new format and added new data. For the selection of images for separate training and testing datasets, the code was implemented with the option of *Suffle = True* allowing random selection of images for both training and testing. Then, we used *torch.utils.data.random_split* to split the dataset, generating two separated sets for training and validation (i.e., *train_set* and *val_set*). The next step involved converting the dataset of each image into a tensor. By applying *torch.tensor* to each data element, we efficiently copied and converted the data into a tensor array. This process was applied to all tensor arrays, and they were then organized into a dictionary. The resulting data structure was prepared to serve as input for DataLoader. Leveraging *torch.utils.data.DataLoader*, we created an iterable interface around the dataset, simplifying access to individual samples for subsequent processing and analysis. This abstraction aids in managing large datasets efficiently during machine learning tasks, enhancing data manipulation and model training. By encapsulating the dataset within an iterable structure, DataLoader streamlines the retrieval and utilization of data samples, ultimately contributing to the smoother and more effective development of machine learning models and algorithms [30].

In our implementation, specific hyperparameters were utilized, and details are provided below. Due to GPU constraints, batch size was restricted to one. After a comprehensive 50-epoch training process, we saved the model for subsequent image testing. Notably, the saved model was configured to operate on a CPU to ensure compatibility with the robots' CPU for future deployment. This deliberate choice guarantees the model's seamless functionality across diverse hardware environments, affirming its efficacy and robustness in real-world scenarios, particularly in the context of PPE detection. The corresponding training loss curve is depicted in Figure 1.

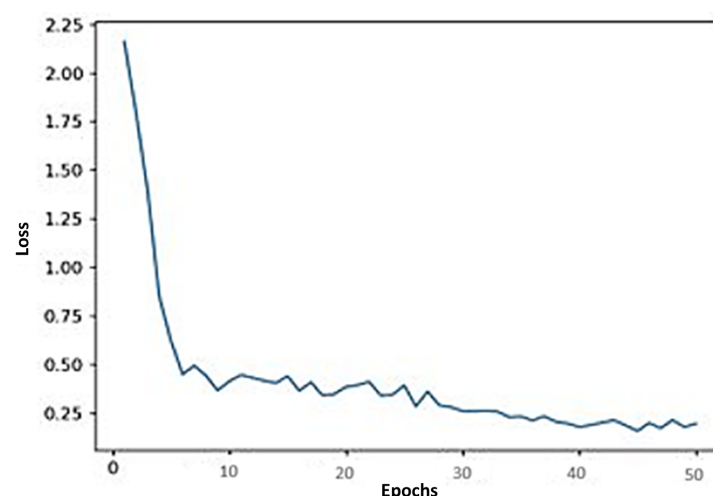


Figure 1. Loss curve of Faster R-CNN.

When configuring Faster R-CNN, it is essential to specify the number of classes as one more than the actual count. For instance, in our project encompassing classes like “helmet”, “safety vest”, and “worker”, we designated four classes. This additional class accounts for the background category, which Faster R-CNN reserves for parts of images without bounding boxes. Eventually, the model underwent testing on a construction site video, demonstrating its capability to successfully identify workers, along with their helmets and safety vests, showcasing its effectiveness in real-world scenarios. The images shown in Figure 2 are sample instances from the training dataset used [28,29].



Figure 2. Sample images from the training dataset.

3.2. Few-Shot Object Detection

Few-Shot Object Detection involves training a model to recognize and localize objects from a limited number of examples per class. This is particularly useful in scenarios where collecting a large labeled dataset is challenging or expensive. The process typically starts with selecting a pre-trained object detection model (Faster-rcnn) on a large dataset (Pascal VOC), which learns rich feature representations from a large dataset. In the few-shot context, the last layer is then fine-tuned while other parameters are frozen. In the second stage, fine-tuning on a small labeled dataset (from both base shots and novel ones) contains only a handful (less than 10 images) of instances for each target class. During fine-tuning, the last layer of the model adjusts its parameters to adapt to the selected few classes while retaining knowledge gained from the pre-training phase (Faster-rcnn). Additionally, architectural adjustments and hyperparameter tuning are performed to prevent overfitting and enhance the model’s ability to generalize to new classes with minimal examples and keep the data of the first step learning. The reason we selected few-shot classes is because they contained both novel one and base classes. The resulting Few-Shot Object Detection model can then identify objects from the novel classes with a limited number of training instances, making it applicable in scenarios where collecting extensive labeled data is challenging.

The methodology for Few-Shot Object Detection emphasizes leveraging transfer learning and meta-learning approaches. Transfer learning enables the model to inherit knowledge from a pre-trained model, while meta-learning techniques, such as MAML or Reptile, facilitate faster adaptation to new tasks with limited data. The model is trained to learn a meta-learner that can quickly adapt to novel classes using only a small number of examples. This meta-learning paradigm helps the model generalize effectively with minimal data, making it more suitable for scenarios where acquiring abundant labeled samples is impractical.

Exclusively training an object detector using a novel dataset tends to prompt overfitting and inadequate generalization owing to the restricted training dataset. In contrast, using the unbalanced fused data for training often yields a detector disproportionately skewed toward key classes. As a result, the detector struggles to accurately identify cases belonging to new classes, highlighting the inherent challenge of achieving a balance be-

tween training in different classes while avoiding limitations associated with overfitting and bias [31].

In the object detection process, it is common to start with an initial detector model incorporating a pre-trained backbone trained on classification data. This initial model is then further trained on a base dataset, yielding what is known as a base model. Many methodologies then extend this by training on the base model data, including new classes, culminating in the final iteration of the model [32].

3.2.1. Preparing Custom Datasets for FsDet

To create a custom dataset for FsDet, we followed the procedure recommended in [8]. The authors proposed a comprehensive five-step approach, which closely aligns with our dataset creation process. Their methodology introduces a two-stage fine-tuning technique (TFA) specifically tailored for Few-Shot Object Detection tasks. The foundational detection model employed in their work is Faster R-CNN, a two-stage object detector.

As we aimed to integrate our dataset into FsDet, we had to make modifications in the third step, which involved developing the dataset evaluator. We employed appropriate modules and functions for this purpose, ensuring compatibility with FsDet's requirements. The primary functions of this evaluator class included defining the dataset, resetting prediction lists, and preparing predictions, including bounding box coordinates, image_id, and classes. The final step involved implementing an evaluation function to calculate metrics such as Average Precision (AP), AP50, and AP75. As our dataset combines PASCAL VOC with a selection of images featuring helmets and safety jackets (i.e., 20 images for each class), an additional step involved developing code to create a new XML dataset conforming to PASCAL VOC standards. The code for this conversion process can be accessed on GitHub at <https://github.com/RoxanaAz/dataset-converions> (accessed on 1 August 2023).

In addition to the above, our training dataset is systematically divided into two sets: one comprises established base categories with abundant training data, while the other represents emerging novel categories (in our case, Equation (2), each accompanied by a sparse number of annotated instances. In the context of the K-shot object detection task, precisely K-annotated object instances are available for every category, indicating a scarcity of labeled instances for newly introduced objects. This scarcity of annotated instances for novel objects underscores the inherent challenge in training models to achieve accurate detection within this domain, as highlighted in the work by Kohler et al. [32].

3.2.2. Implementation Code for FsDet

Few-Shot Object Detection methodologies can be broadly classified into two categories: those utilizing single-input approaches and those employing dual-input methodologies. In our project, we opted for the dual-input paradigm. This involves the integration of two distinct inputs—namely the query set and the support set—through the designated backbone architecture. In this process, the support features extracted from the support image undergo global averaging pooling and are then aggregated with the query features. Within the framework of dual-input architecture, the support branch plays a crucial role in extracting relevant features from the support images. These extracted support features are subsequently combined with features from the query branch, resulting in an aggregation. This fusion process serves as a guidance mechanism for the detector, directing it to accurately identify object instances belonging to a specific category within the query image. Importantly, the query image is sourced from the support image's corresponding category, facilitating the model's ability to generalize and detect novel objects with only a limited number of annotated examples.

In our implementation, we chose to work with the FsDet architecture, as proposed by Wang et al. [33], designed specifically for Few-Shot Object Detection. The training strategy employed here involves a two-phase process to refine our approach. Initially, the entire object detector, i.e., Faster R-CNN, is trained on a dataset rich in base classes as defined in

Equation (1). In our case, this dense dataset consisted of a set of 1280 images used for both training and testing (70% and 30% respectively); instances of them are shown in Figure 3.

$$BaseClasses = \begin{bmatrix} "aeroplane", "bicycle", "bird", "boat", "bottle", "bus", "car", "cat", \\ "chair", "cow", "diningtable", "dog", "horse", "motorbike", "person", \\ "pottedplant", "sheep", "sofa", "train", "tv", "monitor" \end{bmatrix}, \quad (1)$$

$$NovelShots = ["helmet", "safetyVest"], \quad (2)$$

$$FewShots = \begin{bmatrix} "safetyVest", "bus", "cow", "helmet", "sofa" \\ "aeroplane", "helmet", "cow", "horse", "safetyVest" \\ "helmet", "cat", "motorbike", "safetyVest", "sofa" \end{bmatrix}. \quad (3)$$

Subsequently, fine-tuning is performed on the last layers of the detector using a balanced training set that includes both base and novel classes as shown in Equation (2). Importantly, the remaining model parameters are kept fixed during this fine-tuning process. To enhance training, instance-level feature normalization is introduced to the box classifier. It is noted here that to ensure network comparability with Faster R-CNN, our FsDet network adopts a congruent backbone architecture, mirroring that of our Faster R-CNN model—specifically the Faster R-CNN with a ResNet-50 backbone.



Figure 3. Indicative snapshots from the training dataset used in FsDet. Combination of base classes (i.e., first three rows) and novel classes (i.e., safety vest and helmet; last row).

In the second stage, a small balanced training set is established, including K shots per class and encompassing both base and novel classes. To investigate the results for different numbers of shots, the number of K varies in a range of $k = \{1, 2, 3, 5, 10\}$. It is noted here that 10 is the maximum number of shots we can test with FsDet, since more than 10 shots are defined as rare. By the end of this process, the Few-Shot Class formed in as shown in Equation (3). The initialization process involves randomizing the weights for the box prediction networks dedicated to novel classes. Subsequently, the fine-tuning procedure exclusively targets the box classification and regression networks, encompassing the last layers of the detection model while maintaining the entire feature extractor fixed. The loss function as described in Equation (4) is employed, with a reduced learning rate, 20%, of the

first stage's rate of [33]. In Equation (4), L_{rpn} is the loss of region proposal network, L_{cls} is cross-entropy loss, and L_{loc} is loss for the box regressor.

$$L = L_{rpn} + L_{cls} + L_{loc}. \quad (4)$$

At this stage, a classifier based on cosine similarity is considered for the second fine-tuning stage, introducing a mechanism to measure the similarity between feature vectors. Weight matrix $\omega \in \mathbb{R}^{d \times c}$ of box classifier C can be expressed as $[\omega_1, \omega_2, \dots, \omega_c]$, with $\omega_c \in \mathbb{R}^d$ representing the per-class weight vector. C 's output produces scaled similarity scores $S_{i,j}$ between the input feature $F(x)$ and the weight vectors of different classes.

$$s_{i,j} = \frac{\alpha F(x) \omega_i}{\|F(x)\| \|\omega_j\|}. \quad (5)$$

The entries $S_{i,j}$ are calculated from the similarity score between the i^{th} object proposal of input x and the weight vector of class j . While α acts as the scaling factor, set at 20 in our experiments. Instance-level feature normalization in the cosine similarity-based classifier aids in decreasing intra-class variance and enhancing detection accuracy for novel classes. This improvement comes with a relatively minor decline in the detection accuracy of base classes, particularly when the training examples are limited [33]. The novel classes are evaluated by mAPs, and our mAPs are a little lower than those reported in [33]. It should be considered that they employed ResNet-101 as the backbone; consistent with Meta R-CNN, they integrated FRCN+ ft-full into their framework. Our configuration is almost the same except the backbone, which is ResNet-50. We selected this backbone because the comparison between algorithms with the same backbone is more reasonable. Across various data splits and varying numbers of training shots, the approach which is TFA w/cos consistently surpasses the previous methods by a significant margin. Notably, the authors demonstrated an impressive twofold enhancement in one-shot scenarios. These remarkable changes were also seen in our results as well as reported in results of [33]. This shows that the backbone can play a role, but not a significant one. The improvement in learning and fine-tuning is more eye-catching, since these changes are up to 20 points in significance. To further compare the performance of the cosine similarity-based box classifier (TFA + w/cos) against a conventional FC-based classifier (TFA + w/fc), FsDet observations indicate that TFA + w/cos outperforms TFA + w/fc in instances with extremely low shots, such as one-shot scenarios. However, these results are comparable in higher shots like 10 shots.

In our implementation, a crucial component is the "Scaling Factor of Cosine Similarity". The impact of different scaling factors in the computation of cosine similarity must be carefully considered. As depicted in Equation (5), altering the value of α can influence the outcome of determining the class of an object. Notably, in the work by Wang et al. [33], the authors observed that setting $\alpha = 20$ yields significantly improved results. Therefore, in all reported results from our code, we adhered to the choice of $\alpha = 20$ as it consistently produced superior performance in the context of Few-Shot Object Detection. The scaling factor plays a critical role in shaping the sensitivity of the cosine similarity metric, and selecting an appropriate value is pivotal for achieving optimal results in the classification of objects within the Few-Shot Learning framework.

It is noteworthy that setting $\alpha = 20$ has shown superior performance in terms of both base Average Precision (AP) and novel AP on the PASCAL VOC dataset. Additionally, there is an improvement observed in the COCO dataset, albeit with a minor decrease in base AP. Despite the slight reduction in base AP, the notable enhancement in novel AP, particularly across both datasets, underscores the effectiveness of $\alpha = 20$ in prioritizing accurate detection for novel classes. This choice of scaling factor reflects a trade-off, achieving optimal performance for the specific task of Few-Shot Object Detection, particularly emphasizing the importance of accurate identification of objects from novel classes in real-world scenarios.

4. Results

To evaluate the effectiveness of each trained network, we employ a standard evaluation procedure that involves three distinct metrics: (i) average precision (AP), (ii) average recall (AR), and (iii) F1-score (F1). These metrics are computed using the following formulas [34]:

$$AP = \frac{tp}{tp + fp}, \quad (6)$$

$$AR = \frac{tp}{tp + fn}, \quad (7)$$

$$F1 = 2 * \frac{AP \cdot AR}{AP + AR}. \quad (8)$$

4.1. Results of Faster-RCNN

To assess the accuracy of the Faster-RCNN algorithm, we calculated the metrics described above on the test dataset described in Section 3.1.1 maintaining the following hyperparameters: *train_batch_size* = 1 defined as train batch size, *test_batch_size* = 1 as test batch size, number of classes = 4, *learning_rate* = 0.005, number of epochs = 50. The obtained results are summarized below (Table 1).

Table 1. Result of Faster R-CNN.

	Person	Helmet 1	Safety Vest
Recall	0.8	0.76	0.78
Precision	0.79	0.70	0.72
F1-score	0.79	0.73	0.75

The mean recall values for each class, namely person, helmet and safety_vest (0.8, 0.76, and 0.78, respectively), underscore the model's efficacy in correctly capturing instances of each category. Furthermore, the overall accuracy of 0.75 reflects the model's commendable performance in accurately identifying and classifying instances across the specified classes. These quantitative metrics provide a comprehensive overview of the model's proficiency in object detection and classification within the specified test dataset. It is worth noting that the test images revealed varying degrees of difficulty. While all classes displayed clear and distinguishable features, the model demonstrated high performance, particularly in challenging cases characterized by visual complexity, occlusions, or subtle variations within the class boundaries. Illustrative instances from the results on the test dataset are shown in Figure 4, where the class and the confidence level are given on top of each detected bounding box (red rectangles in the figures).



Figure 4. Sample result images from the test dataset with predicted bounding boxes along with class prediction and confidence level.

Results on Video Sequences. Moreover, due to the significant difference of object detection over videos and images, object recognition presents a difficult problem. Videos sometimes have unclear or moving elements that alter their stance or status. The changing background and erratic lighting present difficulties for item detection. On the other hand, temporal information of movies can offer more details than still photographs and is the tool for real-time applications. Accordingly, we tested the implemented model of Faster R-CNN to videos that were commercially available here: <https://www.shutterstock.com/video/search/construction> (accessed on 10 January 2024).

Indicative frames from the results on the FsDet algorithm running on the 30 s video sequence are illustrated in Figure 5. Video results can be seen here: <https://www.youtube.com/watch?v=tdy-BHR-UFM> (accessed on 10 January 2024), proving that the algorithm is robust for continuous inspection of construction sites even if the camera is far from the main scene. Interestingly, we see that the bounding boxes predictions are constant and are maintained during the video.



Figure 5. Object detection results of Faster R-CNN on videos sequences.

4.2. Results of FsDet

When incorporating novel classes into Few-Shot Learning for object detection, the outcome is influenced by the model's ability to generalize from a limited number of examples per novel class. If the novel classes are visually similar to those encountered during pre-training, the model may successfully adapt and recognize the new classes with a small number of shots. However, if the novel classes exhibit significant visual differences or if the available examples are insufficient for effective learning, the model may struggle to generalize, leading to lower detection performance on the novel classes. The success of incorporating novel classes in Few-Shot Learning hinges on finding a balance between the model's ability to transfer knowledge from pre-training and its capacity to adapt to the unique characteristics of the new classes with limited training instances.

In this paper, our primary objective was to delve into the application of Few-Shot Learning within a novel domain, specifically focusing on the inspection of construction workers. We sought to assess its capability for generalization in pertinent setups. The initial phase involved training our model with a generic dataset, as detailed in Section 3.2.2. However, the evolution of our approach led to the augmentation of this dataset, refining the model's ability to detect Personal Protective Equipment (PPE) objects such as safety vests and helmets. This transition reflects our commitment to adapting Few-Shot Learning to address specific challenges within the context of construction worker inspection.

During the testing phase, the parameters shown in Table 2 remain the same as those of the training. To compare the results between different novel sets, the COS (mAP50) is selected. The final results are shown in Table 3. The dataset is divided into five sections as introduced in matrix Section 3.2.2. Accordingly, *Novel Set 1* stands for the one shot of each novel class and, accordingly, *Novel Set 10* stands for ten shots per class. As expected, performance increases as the number of shots increases.

Table 2. Hyperparameters for FsDet.

<i>test_size</i>	<i>train_size</i>	Number of Classes	Number of Epochs
800	480	22	17

Table 3. Result of Few-Shot Learning (Backbone = FRCNN-R50).

Method/Shot	Novel Set 1	Novel Set 2	Novel Set 3	Novel Set 5	Novel Set 10
COS (mAP50)	36.8	30.1	43.4	54.6	55.2

To offer a qualitative insight into the derived object detection capabilities, we present visualizations of detected novel objects on the studied dataset [28,29] in Figure 6. Both successful and failed instances are showcased from the dataset, providing a comprehensive understanding of potential error patterns. For example, in most cases, valves are detected as helmets which strengthens the appearance of false-positive instances.



array of classes. The adaptability of Few-Shot Learning proves valuable in addressing unknown environments, thereby improving overall model robustness and versatility.

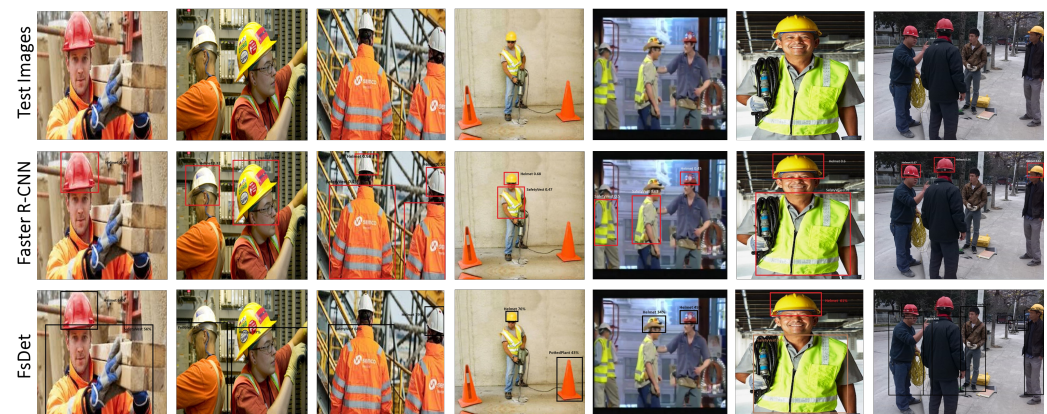


Figure 7. Comparative results of object detection strategies. In the first row, the raw test images are presented, while in the second and the third the object detection results of Faster R-Cnn and FsDet, respectively, are given. It is noted that in FsDet results, both correct and wrong predictions can be seen.

5. Discussion and Conclusions

In conclusion, our study focused on addressing the inherent challenges of object detection in construction sites, emphasizing the visibility of workers' clothing and essential equipment amid adverse conditions such as varying lighting, dust, and occlusions. We compared the performance of two algorithms, namely Faster R-CNN and Few-Shot Object Detection, using four realistic videos captured from construction sites. The results, as illustrated in Figure 7, highlight the superior accuracy of Faster R-CNN detection in accommodating the diverse and dynamic nature of construction site scenarios. This adaptability proved crucial in overcoming challenges associated with limited data and complex environmental conditions. Faster R-CNN object detection approach showcased its efficacy in accurately identifying objects, particularly workers and their safety gear, under challenging conditions. At the same time, it is important to note the significance of leveraging innovative methodologies like Few-Shot Learning to enhance object detection performance in real-world construction site environments, without the need of extensive and novel datasets from the targeted area.

Notably, the training process involved a dataset with limited multi-variety instances, yet Few-Shot Object Detection excelled in recognizing new instances and adapting to unfamiliar conditions. This adaptability is particularly noteworthy in real-world construction site applications, where the dynamic and unpredictable nature of the environment demands models that can effectively generalize to novel situations. The positive outcomes from testing highlight Few-Shot Learning as a promising approach for robust object detection in complex and dynamic settings, showcasing high accuracy in the performed testing, 55.2.

Our Faster R-CNN model achieved a mean Average Precision (mAP) of 73.8%, with precision scores of 0.79, 0.7, and 0.72 for the categories of person, helmet, and safety vest, respectively. In comparison, a referenced study [6] reports an mAP of 75.9%, combining the COCO 2007 and COCO 2012 datasets with VGG-16 as the backbone architecture. Notably, our choice was ResNet50 as the backbone. Precision scores for distinct classes were outlined in [6], varying from 63.6% for chairs to an impressive 86.6% for trains. The adoption of ResNet-5 instead of VGG-16 was aimed at enhancing frame rates [34], evident from their reported 5 fps compared to our 7 fps with ResNet-50. It is important to highlight that networks like YOLO achieve higher frame rates, reaching 45 fps, making them more suitable for video applications. However, our deliberate choice of Faster R-CNN serves to provide a consistent baseline and backbone architecture, facilitating a more direct and insightful evaluation of Few-Shot Object Detection results.

Future work involves the integration of robots equipped with scanning capabilities to monitor employees and their surroundings, ensuring compliance with safety requirements. A notable example is the "Spot-One" robot strategically positioned on construction sites to observe worker activities and verify the presence of essential safety equipment. This technological approach to monitoring is dynamic, minimizing the necessity for continuous human supervision and facilitating swift intervention in instances where deviations from safety standards are detected. The utilization of such robotic technologies holds promise for enhancing safety protocols on construction sites and streamlining monitoring processes through efficient automation.

Furthermore, in the envisioned future work, these robotic systems will not only monitor workers and their compliance with safety standards, but will also provide real-time guidance and alarms. This proactive approach aims to warn workers promptly and offer reminders about the proper usage of Personal Protective Equipment (PPE). By incorporating these real-time guidance and alarm functionalities, robotic systems contribute to creating a safer working environment, fostering continuous adherence to safety protocols, and ultimately mitigating potential risks on construction sites.

Author Contributions: Writing—original draft, R.A.; Writing—review and editing, M.K. and Y.P. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. UNGPs. A Safe and Healthy Working Environment. 2023. Available online: <https://unglobalcompact.org/take-action/safety-andhealth#:~:text=the%20new%20brief-,A%20Safe%20and%20Healthy%20Working%20Environment,working%20conditions%20every%20single%20day> (accessed on 11 February 2024).
2. Coombes, M. How Many Workplace Injuries Are Reported Each Year. 2022. Available online: <https://www.actassociates.co.uk/news/how-many-workplace-injuries-are-reported-each-year/> (accessed on 11 February 2024).
3. Pu, Y.; Wang, Y.; Xia, Z.; Han, Y.; Wang, Y.; Gan, W.; Wang, Z.; Song, S.; Huang, G. Adaptive Rotated Convolution for Rotated Object Detection. In Proceedings of the 2023 IEEE/CVF International Conference on Computer Vision (ICCV), Paris, France, 2–3 October 2023; pp. 6566–6577.
4. Moshayedi, A.J.; Roy, A.S.; Taravet, A.; Liao, L.; Wu, J.; Gheisari, M. A Secure Traffic Police Remote Sensing Approach via a Deep Learning-Based Low-Altitude Vehicle Speed Detector through UAVs in Smart Cities: Algorithm, Implementation and Evaluation. *Future Transp.* **2023**, *3*, 189–209. [CrossRef]
5. Yang, L.; Zheng, Z.; Wang, J.; Song, S.; Huang, G.; Li, F. AdaDet: An Adaptive Object Detection System Based on Early-Exit Neural Networks. *IEEE Trans. Cogn. Dev. Syst.* **2024**, *16*, 332–345. [CrossRef]
6. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2017**, *39*, 1137–1149. [CrossRef] [PubMed]
7. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27 June–30 June 2016; pp. 770–778. [CrossRef]
8. Available online: <https://github.com/ucbdrive/few-shot-object-detection> (accessed on 18 August 2023).
9. Saidi, K.S.; Bock, T.; Georgoulas, C. Robotics in Construction. In *Springer Handbook of Robotics*; Siciliano, B., Khatib, O., Eds.; Springer International Publishing: Cham, Switzerland, 2016; pp. 1493–1520. [CrossRef]
10. Ahmed, M.I.B.; Sarairoh, L.; Rahman, A.; Al-Qarawi, S.; Mhran, A.; Al-Jalaoud, J.; Al-Mudaifer, D.; Al-Haidar, F.; AlKhulaifi, D.; Youldash, M.; et al. Personal Protective Equipment Detection: A Deep-Learning-Based Sustainable Approach. *Sustainability* **2023**, *15*, 13990. [CrossRef]
11. Wang, Z.; Wu, Y.; Yang, L.; Thirunavukarasu, A.; Evison, C.; Zhao, Y. Fast Personal Protective Equipment Detection for Real Construction Sites Using Deep Learning Approaches. *Sensors* **2021**, *21*, 3478. [CrossRef] [PubMed]
12. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27 June–30 June 2016; pp. 779–788. [CrossRef]
13. Han, Y.; Han, D.; Liu, Z.; Wang, Y.; Pan, X.; Pu, Y.; Deng, C.; Feng, J.; Song, S.; Huang, G. Dynamic Perceiver for Efficient Visual Recognition. *arXiv* **2023**, arXiv:2306.11248.
14. Pu, Y.; Liang, W.; Hao, Y.; Yuan, Y.; Yang, Y.; Zhang, C.; Hu, H.; Huang, G. Rank-DETR for high quality object detection. *arXiv* **2023**, arXiv:2310.08854.

15. Jia, D.; Yuan, Y.; He, H.; Wu, X.; Yu, H.; Lin, W.; Sun, L.; Zhang, C.; Hu, H. Detrs with hybrid matching. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–27 June 2023; pp. 19702–19712.
16. Manuel Davila Delgado, J.; Oyedele, L. Robotics in construction: A critical review of the reinforcement learning and imitation learning paradigms. *Adv. Eng. Inform.* **2022**, *54*, 101787. [\[CrossRef\]](#)
17. Protik, A.A.; Rafi, A.H.; Siddique, S. Real-time Personal Protective Equipment (PPE) detection using Yolov4 and tensorflow. In Proceedings of the 2021 IEEE Region 10 Symposium (TENSYP), Grand Hyatt Jeju, Republic of Korea, 23–25 August 2021; pp. 1–6.
18. Halder, S.; Afsari, K. Robots in Inspection and Monitoring of Buildings and Infrastructure: A Systematic Review. *Appl. Sci.* **2023**, *13*, 2304. [\[CrossRef\]](#)
19. Pereira da Silva, N.; Eloy, S.; Resende, R. Robotic construction analysis: Simulation with virtual reality. *Heliyon* **2022**, *8*, e11039. [\[CrossRef\]](#) [\[PubMed\]](#)
20. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 580–587. [\[CrossRef\]](#)
21. Girshick, R. Fast R-CNN. In Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015; pp. 1440–1448. [\[CrossRef\]](#)
22. Yadav, S. Deep learning based safe social distancing and face mask detection in public areas for COVID-19 safety guidelines adherence. *Int. J. Res. Appl. Sci. Eng. Technol.* **2020**, *8*, 1368–1375. [\[CrossRef\]](#)
23. Cha, Y.J.; Choi, W.; Suh, G.; Mahmoudkhani, S.; Büyüköztürk, O. Autonomous structural visual inspection using region-based deep learning for detecting multiple damage types. *Comput.-Aided Civ. Infrastruct. Eng.* **2018**, *33*, 731–747. [\[CrossRef\]](#)
24. Seo, J.; Han, S.; Lee, S.; Kim, H. Computer vision techniques for construction safety and health monitoring. *Adv. Eng. Inform.* **2015**, *29*, 239–251. [\[CrossRef\]](#)
25. Tang, S.; Roberts, D.; Golparvar-Fard, M. Human-object interaction recognition for automatic construction site safety inspection. *Autom. Constr.* **2020**, *120*, 103356. [\[CrossRef\]](#)
26. Duan, R.; Deng, H.; Tian, M.; Deng, Y.; Lin, J. SODA: A large-scale open site object detection dataset for deep learning in construction. *Autom. Constr.* **2022**, *142*, 104499. [\[CrossRef\]](#)
27. Ilyas, M.; Khaw, H.Y.; Selvaraj, N.M.; Jin, Y.; Zhao, X.; Cheah, C.C. Robot-assisted object detection for construction automation: Data and information-driven approach. *IEEE/ASME Trans. Mechatronics* **2021**, *26*, 2845–2856.
28. Nath, N.D.; Behzadan, A.H.; Paal, S.G. Deep learning for site safety: Real-time detection of personal protective equipment. *Autom. Constr.* **2020**, *112*, 103085. [\[CrossRef\]](#)
29. Safety Helmet and Reflective Jacket Images of Individua. Available online: <http://www.kaggle.com/datasets/niravnaik/safety-helmet-and-reflective-jacket> (accessed on 17 August 2023).
30. Datasets & Dataloaders. Available online: https://pytorch.org/tutorials/beginner/basics/data_tutorial.html (accessed on 19 August 2023).
31. Yan, X.; Chen, Z.; Xu, A.; Wang, X.; Liang, X.; Lin, L. Meta r-cnn: Towards general solver for instance-level low-shot learning. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 9577–9586.
32. Köhler, M.; Eisenbach, M.; Gross, H.M. Few-Shot Object Detection: A Comprehensive Survey. *IEEE Trans. Neural Netw. Learn. Syst.* **2023**, *early access*. [\[CrossRef\]](#)
33. Wang, X.; Huang, T.E.; Darrell, T.; Gonzalez, J.E.; Yu, F. Frustratingly Simple Few-Shot Object Detection. *arXiv* **2020**, arXiv:cs.CV/2003.06957.
34. Koskinopoulou, M.; Raptopoulos, F.; Papadopoulos, G.; Mavrakis, N.; Maniadakis, M. Robotic Waste Sorting Technology: Toward a Vision-Based Categorization System for the Industrial Robotic Separation of Recyclable Waste. *IEEE Robot. Autom. Mag.* **2021**, *28*, 50–60. [\[CrossRef\]](#)

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.