Supervised Virtual-to-Real Domain Adaptation for Object Detection Task using YOLO

1st Akbar Satya Nugraha
Departement of Informatics Engineering
Faculty of Computer Science
Brawijaya University
Malang, Indonesia
personal.akbarsn@gmail.com

2nd Novanto Yudistira
Departement of Informatics Engineering
Faculty of Computer Science
Brawijaya University
Malang, Indonesia
yudistira@ub.ac.id

3rd Bayu Rahayudi
Departement of Informatics Engineering
Faculty of Computer Science
Brawijaya University
Malang, Indonesia
ubay1@ub.ac.id

***Abstract*—Deep neural network shows excellent use in a lot of real-world tasks. One of the deep learning tasks is object detection. Well-annotated datasets will affect deep neural network accuracy. More data learned by deep neural networks will make the model more accurate. However, a well-annotated dataset is hard to find, especially in a specific domain. To overcome this, computer-generated data or virtual datasets are used. Researchers could generate many images with specific use cases also with its annotation. Research studies showed that virtual datasets could be used for object detection tasks. Nevertheless, with the usage of the virtual dataset, the model must adapt to real datasets, or the model must have domain adaptability features. We explored the domain adaptation inside the object detection model using a virtual dataset to overcome a few well-annotated datasets. We use VW-PPE dataset, using 5000 and 10000 virtual data and 220 real data. For model architecture, we used YOLOv4 using CSPDarknet53 as the backbone and PAN as the neck. The domain adaptation technique with fine-tuning only on backbone weight achieved a mean average precision of 74.457.**

YOLOv4, Object Detection, Virtual Dataset, Domain Adaptation, Personal Protective Equipment

I. Introduction

In the new spring of artificial intelligence, particularly within the subfield of machine learning, numerous notable advancements have demonstrated the viability of using machine learning for specific human tasks, such as object detection and classification [1]. However, the success of machine learning is heavily reliant on the availability of substantial amounts of real data and their corresponding labels.

In the era of big data, obtaining real input data to train machine learning algorithms is relatively straightforward for a wide range of applications. However, several fields require more extensive training data, which often necessitates manual curation to ensure usability.

Preparing a dataset for training is complex, especially for tasks like object detection, which require precise labeling of object anchors in each image. Training an anchor-based object detector with sparsely annotated data can lead to performance degradation [2].

Challenges such as data availability and the laborious process of data curation have prompted researchers to explore alternative methods. Among these emerging methods are synthetic data, computer-generated datasets, and virtual datasets. Virtual datasets have gained popularity due to their ability to provide abundant, accurately labeled data at a lower cost.

However, the use of virtual datasets presents a challenge: cross-domain shift. Cross-domain object detection is complex due to multi-level domain shifts in unseen domains [3]. Previous research has proposed various methods to address cross-domain shifts, ranging from incorporating domain-adapting layers [4] to developing hierarchical domain-consistent networks [3] to mitigate the challenges of using virtual data.

This study investigates a domain adaptation strategy that maximizes the utility of virtual domain data in real-world domains, thereby reducing the need for extensive real-world data. Specifically, we demonstrate how transfer learning on a well-established deep neural network can achieve state-of-the-art results in automatic visual media indexing when trained on virtually generated images of individuals wearing safety gear such as high-visibility jackets and helmets, followed by domain adaptation using a limited number of real image training examples.

II. Related Work

Object detection technologies have achieved amazing accuracies with faster speeds that were unimaginable a few years ago. Currently, YOLO [5] [6] [7] and RCNN [8] are the de facto standards for object detection tasks. Much of the research on object detection relies on huge, generic annotated datasets such as Pascal [9], ImageNet [10], MS COCO [11], or OpenImages [12]. These datasets collect a large amount of images from the web and are manually annotated.

With the need for vast amounts of data to achieve good accuracy, virtual or computer-generated datasets have gained significant interest. The use of virtual datasets began with research on detecting pedestrians, which showed promising results with a derivation rate of less than 2% [13]. Virtual datasets have also been utilized to study trained CNNs for qualitative and quantitative analysis of deep features [14].

The usage of data generated from games has also been explored in a few research studies. In [15], 50,000 labeled images from the game GTA-V were used with CNNs, demonstrating

979-8-3503-5409-6/24/$31.00 ©2024 IEEE
DOI 10.1109/CAI59869.2024.00242

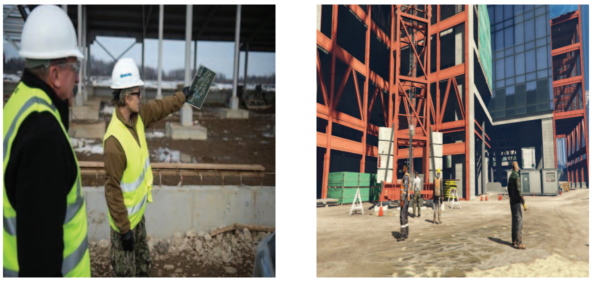
that the mean squared error for lane distance estimation is considerably small when using only a virtual dataset. In another study [16], Unreal Engine was used to generate datasets, enabling an RCNN model to detect a sofa from different viewpoints.

Datasets from GTA-V have shown promising results on tasks such as real people tracking and pose estimation [17]. Using Faster R-CNN on virtual datasets and validating the results on the KITTI dataset has also yielded positive outcomes [18]. Virtual datasets have also been used to train simple convolutional networks to detect objects belonging to various classes in videos [19].

Object detection models can also benefit from virtual datasets to achieve better accuracy. In [15], the use of a virtual dataset called SIM 10k alongside the real dataset Cityscapes for car detection resulted in an average precision of 51.6%. Another study utilized 140,000 virtual images and just 220 real images to train an object detection model for Personal Protective Equipment (PPE) detection with 76% accuracy [20]. Based on the research above, the usage of virtual datasets could lead to models with improved accuracy.

III. Methodology

A. Virtual Data



(a) Image for real dataset (b) Image for virtual dataset

Fig. 1. Image sample for real & virtual dataset

We utilized the VW-PPE dataset, comprising over 140,000 virtual images and 220 real images. The virtual images were generated using RAGE, the game engine for GTA-V, with each image having a width of 1088 and a height of 612. However, the real images varied in width and height. The VW-PPE dataset consists of seven object classes: Bare Head, Helmet, Ear Protection, Welding Mask, Bare Chest, High Visibility Vest, and Person. The virtual images were generated in 10 different locations on the game map, each with three weather and time variations. From the pool of 140,000 virtual images, we randomly sampled 5,000 and 10,000 images for this research. The real images were evenly split between training and test datasets in a 50:50 ratio. Sample images from the VW-PPE dataset are illustrated in Fig. 1.

B. YOLO

The architecture employed for object detection in our study is You Only Look Once (YOLO), a one-stage detector capable of performing image localization and classification simultaneously. Specifically, we utilized YOLOv4 for this task due to its customizable components. YOLOv4 incorporates CSPDarknet53 [21] as the backbone architecture, PAN [22] as the neck, and YOLOv3 detector layer [1] as the head.

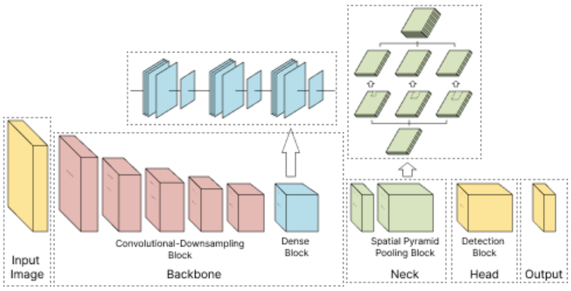

Input Image
Convolutional-Downsampling Block
Dense Block
Backbone
Spatial Pyramid Pooling Block
Neck
Detection Block
Head
Output

Fig. 2. YOLOv4 architecture

To assess the performance of our implementation, we utilized Intersection over Union (IoU) based on the area of the detected (D) and real (V) bounding boxes, along with Precision (Pr) and Recall (Rc). The confidence score associated with detected bounding boxes ranges from 0 to 1, and they are included in the output only if their confidence score exceeds a user-defined threshold. Based on these criteria, the mean Average Precision (mAP) is calculated as the average of the highest precision at various recall settings.

C. Loss Function

To ensure robust detection through the training of our deep learning model, we employed the YOLOv4 loss function. The initial component of the YOLOv4 loss function entails the complete Intersection over Union (IoU) loss formula, which computes loss based on the x and y coordinates of the bounding boxes' width and height [23].

$$\alpha = \frac{\upsilon}{(1 - IoU) + \upsilon'} \tag{1}$$

$$\upsilon = \frac{4}{\pi^2}(arctan\frac{w^{gt}}{h^{gt}} - arctan\frac{w}{h})^2 \tag{2}$$

Inside CIoU formula, there are 2 variables, which are α, a positive trade-off parameter, explained in Equation 1, and υ measures the consistency of aspect ratio, explained in Equation 2. So, the formula of L_{CIoU} is explained in Equation 3.

$$L_{CIoU} = \left[1 - IoU + \frac{\rho(b, b^{gt})}{c^2} + \alpha\upsilon\right] \tag{3}$$

The complete equation for YOLOv4 explained in Equation 4. In this function, we utilized the Complete Intersection over Union formula to compute loss using x and y coordinates, as well as the width and height of the bounding boxes [23].

$$
\begin{aligned}
L_{total} = L_{CIoU} \\
& - \sum_{i=0}^{S^2}\sum_{j=0}^{B} I_{ij}^{obj}\left[\hat{C}_i log(C_i) + (1 - \hat{C}_i log(1 - C_i))\right] \\
& - \lambda_{noobj}\sum_{i=0}^{S^2}\sum_{j=0}^{B} I_{ij}^{noobj}\left[\hat{C}_i log(C_i) + (1 - \hat{C}_i log(1 - C_i))\right] \\
& - \sum_{i=0}^{S^2} I_{ij}^{obj} \sum_{c \in classes} \left[\hat{p}_i(c) log(p_i(c)) + (1 - \hat{p}_i(c) log(1 - p_i(c)))\right]
\end{aligned} \tag{4}
$$

Inside Equation 4, second and third components were calculated as the confidence scores of objectness inside every grid cell. The variable of I_{ij}^{noobj} and I_{ij}^{obj} show the presence and absence of an object on that pixel, respectively. Value of I_{ij}^{obj} will be 1 if there are objects in the grid cell, and I_{ij}^{noobj} will be 1 if there is no object in the grid cell and 0 conversely. The variable of C_i and $\hat{C}_i$ are confidence scores of ground truth and prediction of whether there is an object or not, respectively. At the last component, there are $\hat{p}_i$ and p_i variables of actual and prediction class, respectively, for classification loss.

D. Domain adaptation

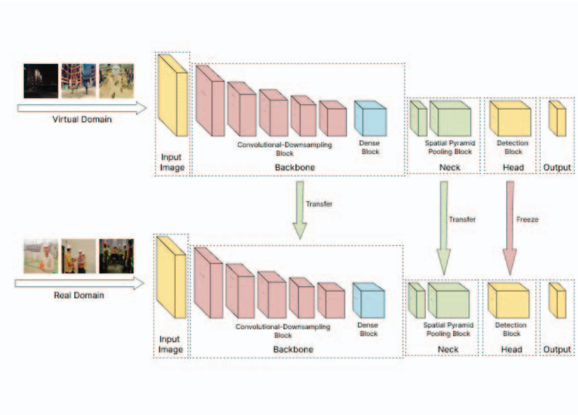


Fig. 3. SHOT Domain Adaptation Scheme

Solving cross-domain shift problem for using virtual datasets, we proposed using domain adaptation transfer learning. In specifically, we apply transfer learning to adapt YOLO to our case. Our premise is that a pre-trained network contains sufficient knowledge for us to specialize it for a new scenario using the transfer learning capabilities of deep neural networks and training sets generated from the virtual world.

The objective of transfer learning is to utilize the first already trained layers (i.e., those identifying low-level features) and update the final layers of the network in order to expand the detection capabilities to the new set of objects. With a trained deep convolutional neural network, its first layers have learned to identify increasingly complex features

In this experiment we used domain adaptation scheme based on SHOT (Source Hypothesis Transfer) [4] and explained in Fig. 3. For adjusting to and addressing the domain shift problem, we implemented the SHOT Domain Adaptation Scheme, where the last layer of the YOLO architecture utilized for detecting bounding boxes would be frozen. In addition to the weight of the freezing detecting layer, we will transfer the weight of the backbone and neck.

In supervised domain adaptation, the model is trained using labeled examples from the source domain and aims to adapt its performance to the target domain, where labeled data may be scarce or unavailable. SHOT starts with a supervised learning phase where the model is trained on labeled data from the source domain (e.g., virtual dataset). This initial training phase provides the model with knowledge about the task at hand and the characteristics of the source domain. After the initial supervised training on the source domain, SHOT proceeds with domain adaptation. Here, it leverages unlabeled examples from the target domain (e.g., real-world data) to refine the model's predictions and adapt its performance to the target domain. Despite the lack of explicit labels for the target domain data, SHOT uses pseudo-labels generated by the model's predictions on the unlabeled target domain data. These pseudo-labels effectively guide the model's learning process during domain adaptation.

E. SHOT domain adaptation and knowledge transfer

Domain adaptation aims to bridge the gap between the distribution of the source domain (where labeled data is available) and the target domain (where labeled data is scarce or unavailable). In SHOT, the model is initially trained on the source domain data (e.g., virtual dataset) with labeled examples. Then, it makes predictions on unlabeled examples from the target domain (e.g., real-world data). The predictions on the unlabeled target domain examples are treated as pseudo-labels. These pseudo-labels are used to retrain the model on a combination of the source and target domain data. Through this iterative process of self-training, the model learns to adapt to the target domain distribution, improving its performance on real-world data. Knowledge transfer refers to the process of transferring knowledge learned from one task or domain to another related task or domain. In the context of domain adaptation, knowledge transfer involves leveraging knowledge gained from the source domain to improve performance on the target domain. SHOT utilizes knowledge transfer by leveraging the labeled data from the source domain to guide the model's learning process on the target domain, even in the absence of labeled target domain data. While both domain adaptation with SHOT and knowledge transfer involves leveraging knowledge from a source domain, the key difference lies in the specific techniques and methodologies used to adapt the model to the target domain. Domain adaptation with SHOT focuses on iteratively refining the model's predictions on unlabeled target domain data using pseudo-labels generated during the self-training process. Knowledge transfer may involve various techniques such as fine-tuning, feature extraction, or model

Scheme	Total Sample Data	mAP
YR	220	0
2*YVR	5000	27.251
	10000	51.369
3*YCVR	5000	65.513
	10000	72.264
	20000	59.691
3*YCSVR	**5000**	**74.457**
	10000	72.096
	20000	73.369
2*YCMVR	5000	55.010
	10000	54.368
2*YCMSVR	5000	59.977
	10000	53.788

TABLE I
MAP RESULT FROM ALL TESTING SCHEME

distillation, where knowledge learned from the source domain is applied directly or indirectly to the target domain without the explicit use of pseudo-labels. In summary, while both domain adaptation with SHOT and knowledge transfer aims to transfer knowledge from a source domain to a target domain, they differ in their methodologies and specific techniques employed to achieve this goal.

IV. Experiments

This scheme is explained in Fig. 2. We trained 6 schemes, that is as below:

- Training from scratch using real dataset only (YR)
- Transfer learning from scratch (YVR)
- Transfer learning with pre-trained weight (YCVR)
- Transfer learning with domain adaptation scheme (YCSVR)
- Transfer learning with mosaic augmentation and pre-trained weight (YCMVR)
- Transfer learning with only backbone weight and mosaic augmentation (YCMSVR)

Based on Table 1, YR receives 0 mAP, since no detections achieved the confidence level. Utilizing 5000 sample data, the mAP for YVR hits 27.251; and using 10,000 sample data, the mAP reaches 51.369. Using virtual datasets as source domains before transferring learning to real-world datasets is a promising strategy for boosting mAP in object detection tasks, as demonstrated by these results.

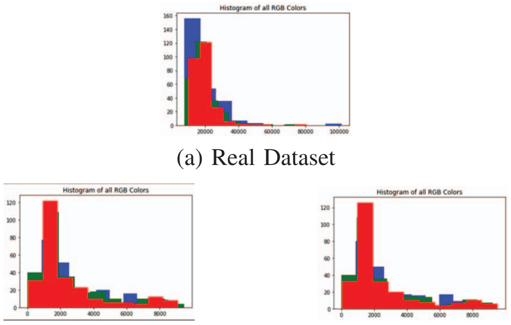
YCVR represents the increase from YVR, where the mAP for 5000 sample data is 65.515 and for 10000 sample data it is 72.264. Using pre-trained weight, even if it is cross-domain, increases the mAP for the object identification model based on this finding.

With 5000 virtual sample data, YCSVR achieves the best mAP score of 74.457; while utilizing 10,000 virtual sample data, the mAP score hits 72.096. Based on these findings, it appears that transfer learning utilizing the SHOT Domain Adaptation Scheme will increase mAP, but will struggle when the proportion of virtual domain data is considerably bigger than real domain data.

Lastly, with YCMVR and YCMSVR, it is demonstrated that mosaic augmentation decreases mAP. All YCMVR and YCMSVR tests reveal a mAP between 50 and 59, which is lower than YCVR.

Class	AP
Head	84.052
Helmet	**93.691**
Ear Protection	**42.292**
Welding Mask	86.364
Bare Chest	59.159
High Visibility Vest	87.637
Person	51.457

TABLE II
AVERAGE PRECISION OF EACH CLASS USING BEST SCHEME



(a) Real Dataset

(b) Sample 5.000

(c) Sample 10.000

Fig. 4. Average color histogram from all dataset used in research

Based on result in Table 1, it shows that mAP from YCSVR using sample 10.000 is lower than sample 5.000. This is because sampling process is random sampling, although class distribution is in the same ratio, the image is still different. Based on Fig. 4, it shows dataset from the real dataset is brighter than the 2 sample data in the virtual dataset. The issue with randomly sampling virtual datasets is that the average histogram color of each sampled virtual dataset will be darker than that of the actual dataset. Therefore, the domain shift problem in this experiment is due to random sampling, which makes virtual datasets darker than real datasets.

Table 2 explained of average precision of every class using best scheme which is YCSVR. It shows that helmet class has the highest average precision and ear protection class has the lowest average precision. This is because of the helmet class has most class label in dataset and ear protection has the fewest class label in dataset.

Using YCSVR models has good result for bounding box prediction and class classification. It is shown in Fig. 5.

V. Discussion

SHOT learns domain-specific feature encoders while keeping the source classifier module fixed. SHOT leverages the source hypothesis to encode distribution information from unseen source data, employing the same classifier module across different domain-specific feature encoders. By aiming to match target feature distributions to source feature distributions,

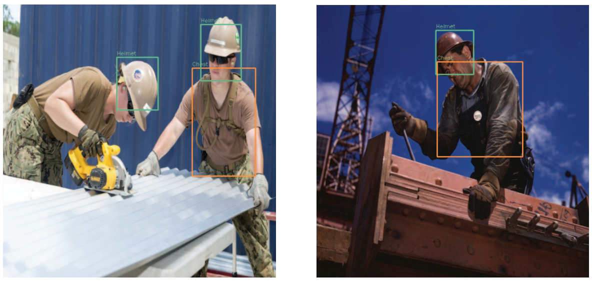


Fig. 5. Sample of detection of PPE objects

Mosaic augmentation combines multiple images into a single training sample by randomly cropping and stitching them together. While this can introduce diversity and increase the effective dataset size, it may also exacerbate domain shift issues when training on data from different domains. The combined images may contain inconsistent visual characteristics or context, making it challenging for the model to learn robust representations that generalize well across domains.

Moreover, mosaic augmentation introduces additional complexity to the training process by combining multiple images into one. This complexity can make it harder for the model to learn meaningful representations, especially in cross-domain scenarios where the underlying visual patterns may differ significantly between datasets. The model may struggle to disentangle the different sources of information present in the mosaic images, leading to confusion and degraded performance.

In cross-domain object detection, the goal is often to learn features that are invariant or robust to domain shifts. Mosaic augmentation may not encourage the model to learn such features effectively, as the combined images may contain conflicting visual cues from different domains. As a result, the model may struggle to generalize to new domains or exhibit poor performance on unseen data.

VI. Conclusion

Training a deep neural network in virtual environments has been proven to be of help when the number of the available and usable training dataset is low. In this paper, we try to research for personal protective equipment object detection with a few real data/images. In our experiment, we trained YOLOv4 on the virtual dataset and tested on real dataset. We also fine-tune the deep neural network with small real data. Based on the experiment that conducted, we found that performance of transfer learning only backbone weight is better than normal transfer learning, also we found out that using mosaic augmentation is not a good choice for training object detection cross-domain.

References

[1] Joseph Redmon and Ali Farhadi, "Yolov3: An incremental improvement," 2018.

[2] Jihun Yoon, Seungbum Hong, and Min-Kook Choi, "Semi-supervised object detection with sparsely annotated dataset," in *2021 IEEE International Conference on Image Processing (ICIP)*, 2021, pp. 719–723.

[3] Yuanyuan Liu, Ziyang Liu, Fang Fang, Zhanghua Fu, and Zhanlong Chen, "Hierarchical domain-consistent network for cross-domain object detection," in *2021 IEEE International Conference on Image Processing (ICIP)*, 2021, pp. 474–478.

[4] Jian Liang, Dapeng Hu, and Jiashi Feng, "Do we really need to access the source data? source hypothesis transfer for unsupervised domain adaptation," 2020.

[5] Joseph Redmon and Ali Farhadi, "Yolo9000: Better, faster, stronger," 2016.

[6] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao, "Yolov4: Optimal speed and accuracy of object detection," *CoRR*, vol. abs/2004.10934, 2020.

[7] Joseph Redmon and Ali Farhadi, "Yolov3: An incremental improvement," 2018.

[8] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," 2015.

[9] Mark Everingham, S. M. Ali Eslami, Luc Van Gool, Christopher K. I. Williams, John Winn, and Andrew Zisserman, "The pascal visual object classes challenge: A retrospective," *International Journal of Computer Vision*, vol. 111, no. 1, pp. 98–136, Jan 2015.

[10] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 248–255.

[11] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár, "Microsoft coco: Common objects in context," 2014.

[12] Alina Kuznetsova, Hassan Rom, Neil Alldrin, Jasper Uijlings, Ivan Krasin, Jordi Pont-Tuset, Shahab Kamali, Stefan Popov, Matteo Malloci, Alexander Kolesnikov, Tom Duerig, and Vittorio Ferrari, "The open images dataset v4," *International Journal of Computer Vision*, vol. 128, no. 7, pp. 1956–1981, mar 2020.

[13] Javier Marín, David Vázquez, David Gerónimo, and Antonio M. López, "Learning appearance in virtual scenarios for pedestrian detection," in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2010, pp. 137–144.

[14] Mathieu Aubry and Bryan Russell, "Understanding deep features with computer-generated imagery," .

[15] Mark Martinez, Chawin Sitawarin, Kevin Finch, Lennart Meincke, Alex Yablonski, and Alain Kornhauser, "Beyond grand theft auto v for training, testing and enhancing deep learning in self driving cars," 2017.

[16] Weichao Qiu and Alan Yuille, "Unrealcv: Connecting computer vision to unreal engine," 2016.

[17] Matteo Fabbri, Fabio Lanzi, Simone Calderara, Andrea Palazzi, Roberto Vezzani, and Rita Cucchiara, "Learning to detect and track visible and occluded body joints in a virtual world," 2018.

[18] Matthew Johnson-Roberson, Charles Barto, Rounak Mehta, Sharath Nittur Sridhar, Karl Rosaen, and Ram Vasudevan, "Driving in the matrix: Can virtual worlds replace human-generated annotations for real world tasks?," 2016.

[19] Erik Bochinski, Volker Eiselein, and Tomas Sikora, "Training a convolutional neural network for multi-class object detection using solely virtual world data," in *2016 13th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, 2016, pp. 278–285.

[20] Marco di Benedetto, Enrico Meloni, Giuseppe Amato, Fabrizio Falchi, and Claudio Gennaro, "Learning safety equipment detection using virtual worlds," in *2019 International Conference on Content-Based Multimedia Indexing (CBMI)*, 2019, pp. 1–6.

[21] Chien-Yao Wang, Hong-Yuan Mark Liao, I-Hau Yeh, Yueh-Hua Wu, Ping-Yang Chen, and Jun-Wei Hsieh, "Cspnet: A new backbone that can enhance learning capability of cnn," 2019.

[22] Shu Liu, Lu Qi, Haifang Qin, Jianping Shi, and Jiaya Jia, "Path aggregation network for instance segmentation," 2018.

[23] Zhaohui Zheng, Ping Wang, Wei Liu, Jinze Li, Rongguang Ye, and Dongwei Ren, "Distance-iou loss: Faster and better learning for bounding box regression," 2019.