# MixUDA: From Synthetic to Real Object Detection

Pablo Gil-Pérez, Daniel Cores, and Manuel Mucientes

Centro Singular de Investigación en Tecnoloxías Intelixentes (CiTIUS), Universidade de Santiago de Compostela, Santiago de Compostela, Spain
[pablo.gil, daniel.cores, manuel.mucientes]@usc.es

**Abstract.** Object detection has made remarkable progress in recent years, driven by advancements in deep learning and the availability of large-scale annotated datasets. However, these methods often require extensive labeled data, which may not be accessible for specific or emerging applications. This limitation has generated interest in Unsupervised Domain Adaptation (UDA), which facilitates knowledge transfer from a labeled source domain to an unlabeled and differently distributed target domain.
This study addresses the challenge of UDA between synthetic and real-world data. A methodology for generating synthetic datasets is proposed using AirSim and Unreal Engine, enabling the creation of highly customizable and diverse datasets. We also propose a Domain Adaptation technique, MixUDA, that maximizes the utility of the synthetic dataset to improve the performance of a model in a real domain. MixUDA is a UDA approach which uses a Mean Teacher architecture and employs pseudo-labels combined with two different image-mixing operations to achieve a smooth and progressive transition from the synthetic to the real domain: pseudo-mosaic and pseudo-mixup.
The obtained results demonstrate encouraging progress, as MixUDA surpasses state-of-the-art models D3T and MixPL by 1.18 and 4 AP points respectively, approaching performance of oracle models trained directly on the target domain. These findings suggest that synthetic datasets have significant potential in addressing data scarcity and improving model generalization, while also pointing to promising directions for further exploration in this area.

**Keywords:** Synthetic dataset · Unsupervised Domain Adaptation

## 1 Introduction

Object detection has experienced significant advancements in recent years, pushed forward by the development of sophisticated deep learning models and the availability of large-scale annotated datasets, such as ImageNet [2] or COCO [6]. However, the availability of large-scale and high-quality labeled data is not always feasible for specific, niche, or emerging applications. This limitation has led to growing interest in the field of Domain Adaptation (DA), which aims to

**Fig. 1.** Real domain image of SARD [11] dataset on the left and synthetic image on the right.

transfer knowledge from a source domain to a different distributed target domain, so that all prior knowledge of a model is leveraged for the application. In particular, Unsupervised Domain Adaptation (UDA) has concentrated increasing attention, as it closely aligns with real-world scenarios where labeled data in the target domain is nonexistent. Using techniques that bridge the distributional gap between source and target domains, UDA has the potential to enable robust object detection in challenging scenarios, such as changing environmental conditions and novel sensor modalities.

In this paper, we address the UDA challenge of transferring knowledge from synthetic to real domain. Domain transfer between synthetic and real datasets has many potential applications, since it allows the use of automatically generated data to train models in domains where acquiring large and varied quantities of data is challenging. On the other hand, the necessity to manually annotate images is also eliminated, as this process can also be carried out automatically and it is not necessary either to have labels in the real dataset, since this adaptation is performed in an unsupervised manner.

To address this problem, this work proposes a methodology for the creation of synthetic datasets, which serve as a basis for Domain Adaptation from synthetic images to real domain. The proposed approach uses two powerful tools, AirSim [12] and Unreal Engine, which allow the realistic simulation of diverse scenarios. In Fig. 1 an example of real domain image and a synthetic domain image created using these tools is shown.

However, creating a synthetic dataset that closely resembles the real domain is not sufficient for the detector to generalize effectively. In addition to having a realistic and varied dataset, it is essential to apply domain adaptation techniques that increases the generalization of the model. Otherwise, the model may learn to detect objects correctly in the synthetic domain but fail to generalize and accurately detect real objects.

For this reason, we propose MixUDA. MixUDA is a UDA method based on a Mean-Teacher architecture that leverages pseudo-labels to identify objects in target domain images and employs image-mixing techniques to adapt the model from the synthetic to the real domain. By using this method, the domain gap is reduced, and the model's performance on the target dataset is improved.

As a summary, the main contributions of this paper are as follows:

- A method for creating a synthetic dataset that allows for automatic scaling of the number of images, making it possible to train models in environments where acquiring large amounts of data is challenging. Additionally, this creation method does not require manual annotations, since this process is also automated.
- The MixUDA approach, a new UDA method that leverages a Mean-Teacher architecture and a novel way of mixing source and target domain images to enable a progressive adaptation through techniques inspired by the Semi-Supervised learning field.

## 2 State of the art

Object detection is a fundamental task in computer vision, which consists of localizing and classifying objects within an image. Most state-of-the-art approaches can be categorized into single-stage and two-stage detection models.

The common process shared by both single-stage and two-stage detectors is feature extraction, where the backbone processes the input image to generate a feature map encoding spatial and semantic information. Two-stage detectors, such as Faster R-CNN [10], use a Region Proposal Network (RPN) to generate candidate object regions, which are then refined and classified. In contrast, single-stage detectors such as FCOS [14] and YOLO [9] eliminate the Region Proposal Network (RPN) and directly perform dense predictions across the entire spatial domain of the image. This design offers faster performance but potentially sacrificing some accuracy.

Training this models requires large amounts of annotated data, so it leads to the increasing interest in synthetic data generation. Some approaches repurpose video games, such as DeepGTAV [5], which uses Grand Theft Auto V (GTAV) to generate high-quality annotated datasets. Alternatively, simulation platforms like AirSim [12], built on Unreal Engine, offer greater flexibility in creating photo-realistic virtual environments tailored to specific tasks. The advanced rendering capabilities of Unreal Engine make it a powerful tool for generating diverse and high-fidelity synthetic datasets.

Although synthetic datasets can be used to train detection models, achieving good performance on real-world images requires applying a Domain Adaptation (DA) process. The objective of Domain Adaptation consists of transferring knowledge from a labeled source domain to an unlabeled or sparsely labeled target domain, addressing distributional shifts due to environmental differences such as lighting or textures, as well as shifts in label distributions. There are different types of Domain Adaptation depending on the problem's context, but in this paper, we will focus on Unsupervised Domain Adaptation (UDA), which performs adaptation without using labeled images from the target domain.

UDA techniques align source and target domains using strategies such as Optimal Transport (OT) [7], which maps feature distributions at the cluster

level to reduce computational costs and class imbalance effects. Additionally, vision-language models like CLIP [8] have been integrated into UDA, as seen in DAMP [4], which employs visual and textual prompts for domain-invariant representations. Another key strategy is pseudo-labeling, exemplified by D3T [3], which addresses RGB-Thermal adaptation through a dual-teacher framework.

In this paper, we propose a novel UDA method inspired by techniques from Semi-Supervised Learning (SSL), as both fields share common characteristics, given that the objective os SSL is to leverage unlabeled data for model training. Specifically, we employ pseudo-mosaic and pseudo-mixup techniques, which are used in MixPL [1]. However, since the objectives of both approaches differ, there are also significant differences, such as the incorporation of a burn-in stage and the separate processing of images from each domain within the pipeline.

## 3 Synthetic Dataset Creation Process

This section details the creation of a synthetic dataset using AirSim, a tool based on the Unreal Engine graphics engine which allows the creation of very realistic environments. AirSim enables image capture from the created environment with multiple perspectives, including RGB, depth maps, and segmentation masks, facilitating the automatic extraction of object coordinates for computer vision tasks. The described process to create a synthetic dataset can be applied to a wide range of different problems.

The dataset generation begins with asset selection, including primary objects of interest and secondary elements to enhance diversity. It is also necessary to have a virtual environment in which to position the assets. Both the models and the virtual environment can be downloaded, reusing models that were originally created for other purposes, or they can be specifically created for the dataset. To place the models in the environment, different techniques can be used, or the assets can just be positioned randomly.

Once the virtual environment is constructed, the next step involves image extraction. Airsim's camera can be positioned in the environment simulating several perspectives. It is important to change the position and orientation of the camera to ensure good variability in the dataset, while performing a complete sweep over the plane of the environment. So instead of using fixed distances to move the camera, minor variations could be added.

For each captured image, AirSim can generate corresponding segmentation maps, providing pixel-level ground-truth. Bounding boxes can be extracted from these masks through color segmentation and contour detection. The process is described in Alg. 1 and an example of segmentation ground-truth images is shown in Fig. 2. Finally, in Fig. 3 examples of images from the dataset extracted under varying lighting conditions are shown.

**Algorithm 1** Synth Dataset Bounding Box Calculation

**foreach** *category_colors in category_color_list* **do**
  **foreach** *color in category_colors* **do**
    - Generate a binary mask of the image for the given color
    - Extract contours from the mask using the Suzuki-Abe algorithm [13].
    **foreach** *contour in contours* **do**
      - Determine the bounding rectangle using the contour's extreme points
      - Store the bounding box coordinates
  - Remove bounding boxes with an area smaller than the predefined threshold

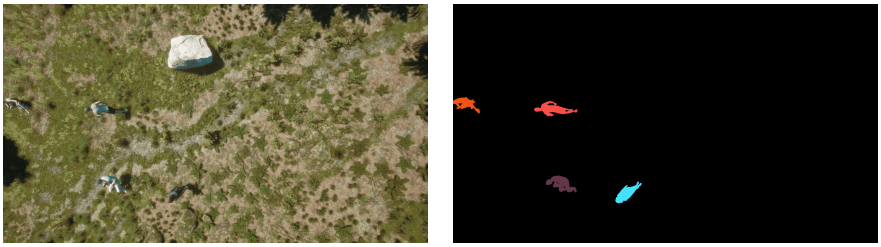

**Fig. 2.** On the left, RGB image extracted from the virtual environment, and on the right, the ground-truth image of the same scene.

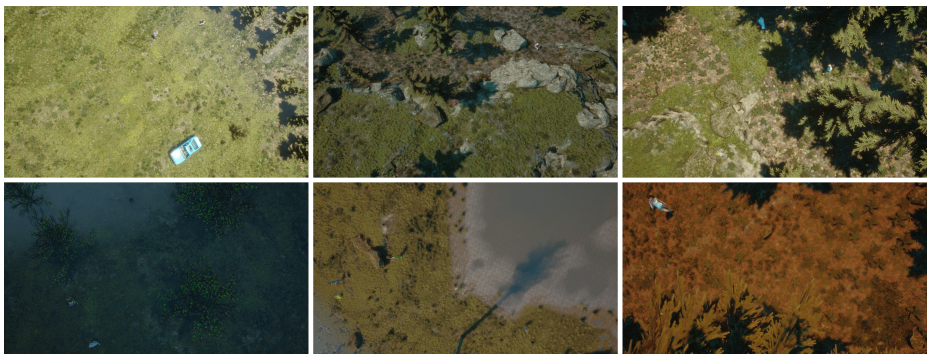

**Fig. 3.** Image examples of different iterations of the virtual environment. The light, asset position and orientation are different on each of the sweeps along the simulation.

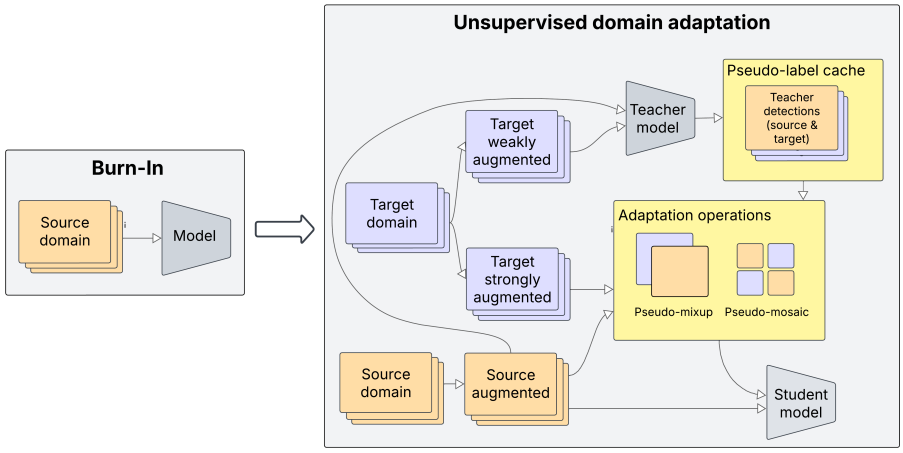


**Fig. 4.** Overview of MixUDA framework. The pipeline integrates labeled and unlabeled data on the training process. Labeled images –source domain– undergo a specific set of augmentations, while the unlabeled images –target domain– undergo weak augmentations in the case of the teacher, and strong in the case of the student. Through a cache of pseudo-labels created by the teacher, mosaic and mixup operations are applied and added to the set of input images of the student.

## 4 Methodology

We propose MixUDA, a novel UDA framework that is based on a Mean Teacher architecture which is trained using labeled data from a source –synthetic data– domain and pseudo-labeled data from both source and target –real data– domains. The key feature of this framework is the use of two distinct transformations for the progressive alignment between source and target domain images: pseudo-mixup and pseudo-mosaic. In Fig. 4 a schematic view of the approach is shown.

First, the model is trained in a fully supervised manner with the source dataset during the burn-in stage. These weights are then used as the starting point for both the teacher model and the student model in the mean-teacher framework.

Images from each of the domains receive a different type of transformations. Images from source domain are all augmented with the same set of operations. On the other hand, in the target domain, images which are used as input for the teacher model are transformed using weak augmentations such as image flip. In the case of the target domain images that are passed to the student model, strong augmentations are used like color and geometric transformations.

The teacher model is used to generate pseudo-labels for images, which are stored in the pseudo-label cache. Pseudo-labels are predicted detections assigned to unlabeled data by a pretrained model, in this case the teacher model. Images and pseudo-labels from both the source and target domains are stored, four per

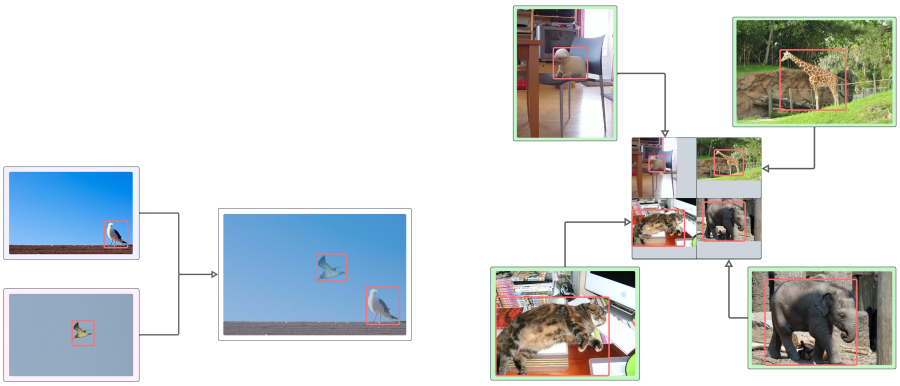

**Fig. 5.** The image on the left illustrates an example of the MixUp operation. The image on the right shows an example of the Mosaic operation. Images extracted from COCO dataset [6].

iteration, and even though the source data is labeled, we still generate pseudo-labels to add data from both domains to the cache. The pseudo-label cache has a size of 8 images, so only images from the last 2 iterations are kept.

The pseudo-labels are used as ground-truth to train the student model, but using them over the strongly augmented images in the case of the target domain images, ensuring that the teacher model generates accurate pseudo-labels while the student model learns to detect invariant features across the different strong augmentations. In the case of the source images, both models were already trained using them, so only strong augmentations are applied.

One of the core ideas of this proposal lies in its strategic use of Mosaic and MixUp operations to improve data diversity and to gradually adapt the model from source to target domain. Mosaic operation combines four images into a single one, while MixUp blends two images and their corresponding labels through a linear interpolation. An example of each of these transformations can be seen in Fig. 5.

These operations are applied relying on the following principles:

- **Progressive adaptation:** Since the model has already been trained with source domain images in the burn-in stage, introducing images that combine both domains allows for a gradual learning process of target domain features, ensuring that the transition is not abrupt in the early stages while reducing progressively the domain gap.
- **Scale:** During training, it is common for large object detections to exceed the number of ground-truth annotations, while medium and small objects are underrepresented. To address this imbalance, the mosaic operation is introduced, which effectively reduces the size of large objects, thereby increasing the representation of smaller objects in the dataset.

MixUp images are created using 4 images from the cache and 4 from the new batch, so 4 images are generated. To create the mosaic, 4 images from the cache are used having as a result 1 mosaic per batch. Moreover, The student model receives one source image with ground-truth detections. The student model's loss is computed based on these images as follows:

$$\mathcal{L} = \mathcal{L}_{\text{sup}} + w_{un} * \mathcal{L}_{\text{un}} \tag{1}$$

where:

- $\mathcal{L}_{\text{sup}}$ represents the loss calculated from supervised images. For classification, focal loss is used, while for regression generalized intersection over union loss is employed.
- $\mathcal{L}_{\text{un}}$ represents the loss calculated from pseudo-labeled images. Same loss functions as supervised loss are used.
- $w_{\text{un}}$ refers to the weight associated with the unsupervised loss.

On the other hand, the teacher model is updated using Exponential Moving Average (EMA), which maintains a smoothed version of the student's weights by exponentially decaying past parameters and providing more stable and consistent supervision, as follows:

$$\theta_{\text{t}}^{T} = \alpha \cdot \theta_{\text{t-1}}^{T} + (1 - \alpha) \cdot \theta_{\text{t}}^{S} \tag{2}$$

where:

- $\theta_{\text{t}}^{T}$ represents the weights of the teacher model at the current step $t$.
- $\theta_{\text{t-1}}^{T}$ represents the weights of the teacher model from the previous step $t-1$.
- $\theta_{\text{t}}^{S}$ referts to the weights of the student model at the current step $t$.
- $\alpha$ represents the momentum coefficient, which controls the weight given to the previous teacher parameters versus the current student parameters during the update.

## 5 Experimentation

### 5.1 Dataset description

The real domain dataset that has been selected to address the described problem is the Search and Rescue Image Dataset for Person Detection (SARD) [11]. This dataset, specifically designed for person detection in search and rescue scenarios, contains a total of 1,979 annotated aerial images. These images are captured from Unmanned Aerial Vehicles (UAV) in outdoor environments and include annotations for various poses of people, such as standing, sitting, or lying down, but for this study, only the "person" category is considered. The dataset provides high-resolution images (1920x1080 pixels), enabling precise annotation

and detection tasks. The domain is challenging due to the diversity of scenes, lighting conditions, and the small size of individuals in the aerial perspective.

Given that the SARD dataset consists of images extracted from four distinct aerial videos, this structure was split into training and testing subsets. Specifically, two videos were selected for training, resulting in 1,025 images, while the other two videos were allocated to testing, resulting in 954 images. This split ensures minimal overlap in scene content between the training and testing sets, thereby promoting a fair evaluation of models' performance. The train set labels are not used in the Domain Adaptation training processes.

The synthetic dataset is composed of a total of 9,683 images with a resolution of 1920x1080 pixels. The images were captured in a virtual environment simulating a forest and mountain landscape, viewed from a UAV perspective. To introduce variability, the dataset includes diverse lighting conditions.

This dataset focuses on the person class, with the positions of individuals varying across images to simulate movement. Examples of such images can be found in Sec. 3.

### 5.2 Results

The results of the different experiments are shown in Tab. 1, using AP and AP50 as metrics, which are the standard in object detection. AP50 is calculated as the area under the Precision-Recall curve using a 0.5 IoU threshold for considering a detection correct, while AP averages AP values over IoU thresholds from 0.5 to 0.95 in 0.05 increments. In the table, the results are divided based on the used detector model: Faster R-CNN and FCOS. For each detector, a fully supervised result is presented as a reference, since it is trained using the training set of the real dataset with its annotations. There is also a base model result, trained exclusively on the synthetic domain dataset, and finally, the results of the techniques used to adapt the synthetic domain to the real one along with their AP and AP50 improvements compared to the Base model. We compare MixUDA with MixPL [1], a Semi-Supervised Object Detection model implemented using Faster R-CNN and FCOS among other detectors, and D3T, a domain adaptation method based on FCOS.

Focusing first on the results of Faster R-CNN, we observe that our approach achieves the best performance, increasing AP by 4.6 points and AP50 by 11 points compared to the base model. These results outperform those obtained with MixPL, increasing AP by 4 points and AP50 by 5 points. Additionally, it is worth highlighting that, in terms of AP, our approach surpasses even the results obtained by training directly on the target domain, while it still lags slightly behind in AP50. This could be due to the fact that the synthetic training set is bigger than real one and offers greater diversity. As a result, applying domain adaptation enables better generalization, leading to improved object detection performance.

On the other hand, using FCOS detector, our approach again achieves the best results. MixUDA improves the results by 4.4 AP points and 8.4 AP50 points compared to the base model. These results also surpass D3T by 1.18 AP points

**Table 1.** The results are for two different detectors: Faster R-CNN and FCOS. For each detector, there is a fully supervised model trained directly on the target dataset highlighted in gray. We also have the Base model, which is trained exclusively on the source dataset, and the different domain adaptation techniques, also trained using labels from source dataset and whose results are compared against the Base model.

| **Model** | **AP** | **AP50** |
|---|---|---|
| Fully supervised (Faster R-CNN) | 22.91 | 55.29 |
| Base (Faster R-CNN) | 18.80 | 40.40 |
| MixPL [1] (Faster R-CNN) | $19.40_{+0.60}$ | $46.40_{+6.00}$ |
| **MixUDA (Faster R-CNN)** | $\mathbf{23.40_{+4.60}}$ | $\mathbf{51.40_{+11.00}}$ |
| Fully supervised (FCOS) | 18.40 | 50.30 |
| Base (FCOS) | 16.10 | 39.00 |
| D3T [3] (FCOS) | $19.32_{+3.22}$ | $41.90_{+2.90}$ |
| MixPL [1] (FCOS) | $18.10_{+2.00}$ | $45.40_{+6.40}$ |
| **MixUDA (FCOS)** | $\mathbf{20.50_{+4.40}}$ | $\mathbf{47.40_{+8.40}}$ |

and more notably, 5.5 AP50 points, and MixPL, by 2.4 points in AP and 2 points in AP50. As in the case of Faster R-CNN, our approach surpasses the oracle model in terms of AP, while still not reaching its AP50 value.

## 6 Conclusions

In this study, we have focused on the task of using synthetic data to train detection models in the context of object detection in real images. Using synthetic data offers advantages such as scalability in the number of images and not requiring human-annotated labels. To achieve this, we first established a methodology for creating synthetic datasets using the AirSim tool alongside the Unreal Engine graphics engine. Following this methodology, we built a synthetic dataset for person detection in natural environments from a UAV perspective.

We developed a novel UDA framework: MixUDA. This method is based on a Mean Teacher architecture, where the teacher model generates pseudo-labels for images from both domains. The key feature of this approach is the use of two operations for progressive alignment between the source and target domain images: pseudo-mixup and pseudo-mosaic.

The results demonstrate that MixUDA successfully outperforms other methods in terms of both AP and AP50, even surpassing the performance of a fully supervised model trained on the target dataset in the case of FCOS. Findings demonstrate that using a synthetic dataset can yield performance comparable to using real-world annotated images in the context of person detection, and MixUDA has proven to be an effective methodology to perform UDA from synthetic to real domain.

For future research, it would be valuable to extend the synthetic dataset creation methodology to other problem domains to evaluate its robustness and generalization. Additionally, exploring the applicability of MixUDA across different datasets would provide further insights into its effectiveness for various types of problems.

## 7 Acknowledgments

This research was partially funded by the Spanish Ministerio de Ciencia e Innovación (grant number PID2020-112623GB-I00, PID2023-149549NB-I00), and the Galician Consellería de Cultura, Educación e Universidade (grant numbers ED431C 2018/29 and ED431G2019/04). These grants are co-funded by the European Regional Development Fund (ERDF). Pablo Gil-Pérez is supported by the Spanish Ministerio de Universidades under the FPI national plan (grant number PRE2023-000607).

## References

1. Chen, Z., Zhang, W., Wang, X., Chen, K., Wang, Z.: Mixed pseudo labels for semi-supervised object detection. arXiv preprint arXiv:2312.07006 (2023)
2. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. IEEE Conference on Computer Vision and Pattern Recognition (CVPR) pp. 248–255 (2009)
3. Do, D.P., Kim, T., Na, J., Kim, J., Lee, K., Cho, K., Hwang, W.: D3t: Distinctive dual-domain teacher zigzagging across rgb-thermal gap for domain-adaptive object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 23313–23322 (2024)
4. Du, Z., Li, X., Li, F., Lu, K., Zhu, L., Li, J.: Domain-agnostic mutual prompting for unsupervised domain adaptation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 23375–23384 (2024)
5. Kiefer, B., Ott, D., Zell, A.: Leveraging synthetic data in object detection on unmanned aerial vehicles. In: 2022 26th international conference on pattern recognition (ICPR). pp. 3564–3571. IEEE (2022)
6. Lin, T.Y., Maire, M., Belongie, S., Bourdev, L., Girshick, R., Hays, J., Perona, P., Ramanan, D., Zitnick, C.L., Dollár, P.: Microsoft coco: Common objects in context. In: European conference on computer vision (ECCV). pp. 740–755. Springer (2014)
7. Liu, Y., Zhou, Z., Sun, B.: Cot: Unsupervised domain adaptation with clustering and optimal transport. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 19998–20007 (2023)
8. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: International conference on machine learning. pp. 8748–8763. PMLR (2021)
9. Redmon, J.: You only look once: Unified, real-time object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition (2016)
10. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: Towards real-time object detection with region proposal networks. IEEE transactions on pattern analysis and machine intelligence **39**(6), 1137–1149 (2016)

11. Sambolek, S., Ivasic-Kos, M.: Search and rescue image dataset for person detection - sard (2021). https://doi.org/10.21227/ahxm-k331, https://dx.doi.org/10.21227/ahxm-k331
12. Shah, S., Dey, D., Lovett, C., Kapoor, A.: Airsim: High-fidelity visual and physical simulation for autonomous vehicles. In: Field and Service Robotics: Results of the 11th International Conference. pp. 621–635. Springer (2018)
13. Suzuki, S., et al.: Topological structural analysis of digitized binary images by border following. Computer vision, graphics, and image processing **30**(1), 32–46 (1985)
14. Tian, Z., Shen, C., Chen, H., He, T.: FCOS: A simple and strong anchor-free object detector. IEEE transactions on pattern analysis and machine intelligence **44**(4), 1922–1933 (2020)