Honorable Mention- Best Presentation Award

Combining simulated data, foundation models, and few real samples for training fine-grained object detectors

Friso G. Heslinga[a], Thijs A. Eker[a], Ella P. Fokkinga[a], Jan Erik van Woerden[a], Frank A. Ruis[a], Richard J. M. den Hollander[a], and Klamer Schutte[a]

[a]TNO - Intelligent Imaging, Oude Waalsdorperweg 63, the Hague, the Netherlands

ABSTRACT

Automatic object detection is increasingly important in the military domain, with potential applications including target identification, threat assessment, and strategic decision-making processes. Deep learning has become the standard methodology for developing object detectors, but obtaining the necessary large set of training images can be challenging due to the restricted nature of military data. Moreover, for meaningful deployment of an object detection model, it needs to work in various environments and conditions, in which prior data acquisition might not be possible. The use of simulated data for model development can be an alternative for real images and recent work has shown the potential for training a military vehicle detector using simulated data. Nevertheless, fine-grained classification of detected military vehicles, using training on simulated data, remains an open challenge.

In this study, we develop an object detector for 15 vehicle classes, containing similar appearing types, such as multiple battle tanks and howitzers. We show that combining few real data samples with a large amount of simulated data (12,000 images) leads to a significant improvement in comparison with using one of these sources individually. Adding just two samples per class improves the mAP to 55.9 [±2.6], compared to 33.8 [±0.7] when only simulated data is used. Further improvements are achieved by adding more real samples and using Grounding DINO, a foundation model pretrained on vast amounts of data (mAP = 90.1 [±0.5]). In addition, we investigate the effect of simulation variation, which we find is important even when more real samples are available.

Keywords: Deep learning; Simulated data; Object detection; Foundation model; Military vehicles

1. INTRODUCTION

Automatic object detection is increasingly important in various domains, including security,[1] robotics,[2] and the military.[3] Military applications of object detection include surveillance and reconnaissance as well as target identification and tracking. The vast amount of sensor data acquired by military platforms, both manned and unmanned, is likely to expand in the coming years, increasing the demand for automated analysis methods.

Deep learning[4] has become the standard methodology for developing object detectors and typically requires a large set of annotated data. Obtaining the necessary set of training images can be challenging due to the restricted nature of military data, resulting in a lack of access to certain objects of interest. Moreover, for meaningful deployment of an object detection model, it needs to work in various environments and conditions for which data acquisition might not be possible.

An alternative for collecting a large set of real data that includes the environment variations and the objects of interest is the use of simulated data. Simulated data for training deep learning methods has shown to be promising for various applications, including autonomous driving,[5] radar image segmentation,[6] and thermal infrared tracking.[7] Interestingly, even when a lot of real image data is available, the addition of simulated data can be beneficial to model performance.[8]

In recent work, we showed that simulated data can be used for training a military vehicle detector[9] using a Mask R-CNN model[10] with transformer backbone.[11] When sufficient simulation variation was included in the simulated dataset, evaluation on a set of real images with 4 different military vehicles led to a mean average

Corresponding author: Friso G. Heslinga. E-mail: fgheslinga@gmail.com

Synthetic Data for Artificial Intelligence and Machine Learning: Tools, Techniques, and Applications II, edited by Kimberly E. Manser, Christopher L. Howell, Raghuveer M. Rao, Celso De Melo, Proc. of SPIE Vol. 13035, 1303509 © 2024 SPIE · 0277-786X · doi: 10.1117/12.3013375

precision (mAP) of 0.76 and an mAP50 of 0.95. In the work, we identified several axes of simulation variation that are important to achieve good detection performance on real data.[9]

Nevertheless, fine-grained classification of a larger set of military vehicles remains an open challenge and previous work showed that simply expanding the pipeline to more object classes decreases the performance significantly. One potential strategy to address this, is to include a small amount of real data to the simulated dataset during training. An alternative is the use of a foundation model. Foundations models, such as Grounding DINO,[12] have recently been introduced and are trained on a very large set of data to obtain rich features for various downstream tasks.

So far, it remains unclear what the value is of combining simulated data with real data for training of a fine-grained military vehicle detector. In this study, we investigate the effect of having access to a few real data samples combined with a large amount of simulated data. We study this for Mask R-CNN and Grounding DINO and compare the benefits of combining both sets with the added value of simulation variation.

2. RELATED WORK

2.1 Object detection

Deep learning has become the standard methodology for object detection, powering popular architectures such as YOLO,[13] SSD,[14] R-CNN,[15] and their respective successors. In this study we use Mask R-CNN,[10] which is an extension of Faster R-CNN[16] with a parallel branch for predicting segmentations, as our baseline model, since it provided good results in previous work.[9] In recent years, Transformer based backbones have shown to be superior over their convolutional based counterpart,[17] especially when training with synthetic data.[18] Our Mask R-CNN uses a Swin-T backbone,[11] a type of Vision Transformer that constructs hierarchical feature maps by progressively merging image patches in deeper layers.

2.2 Joined training of real and simulated data

When training object detectors exclusively with simulated data, a gap is observed in how well these models perform on real-world data. This gap highlights the challenge for simulating data to fully replicate all complex variations and nuances present in real-world scenarios. Adding even a small amount of real data to the mix could significantly enrich the training set and allow the model to learn from this. Therefore, a method to bridge the synthetic-to-real gap is to combine simulated and real data in the training process. This approach aims to leverage the strengths of both data types.

There are two main methods to integrate simulated and real data.[19,20] The first involves sequential training, where the model is initially trained on simulated data and then fine-tuned. This approach allows the model to first understand the broad concepts of classes using synthetic data, before addressing the synthetic-to-real gap by adapting to a limited amount of real data. A potential drawback is however that the model overfits on the simulated data during the initial training phase, which may not provide an ideal foundation for subsequent fine-tuning on real data.

Alternatively, one can train a model using mixed batches, where simulated and real data are presented together during the training process. In this way, the model can learn from both data types simultaneously. It encourages the model to learn generalizable features across both datasets and reduces the risk of overfitting on the simulated data. In preliminary experiments we found that mixed training works best, so we adopted this approach for our study.

2.3 Foundation models

Self-supervised large vision models have marked significant advancements in computer vision research. Well known approaches include SimCLR,[21,22] which leverages contrastive loss, and DINO,[23] which not only improves upon contrastively trained methods but is also fully transformer-based. CLIP[24] was proposed to integrate vision and language, employing an image-text contrastive objective to learn similar embeddings for corresponding text-image pairs. GLIP[25] takes the principals of CLIP and extends them for the use of object detection and phrase grounding, where grounding refers to the process of matching words to specific regions within an image. Building

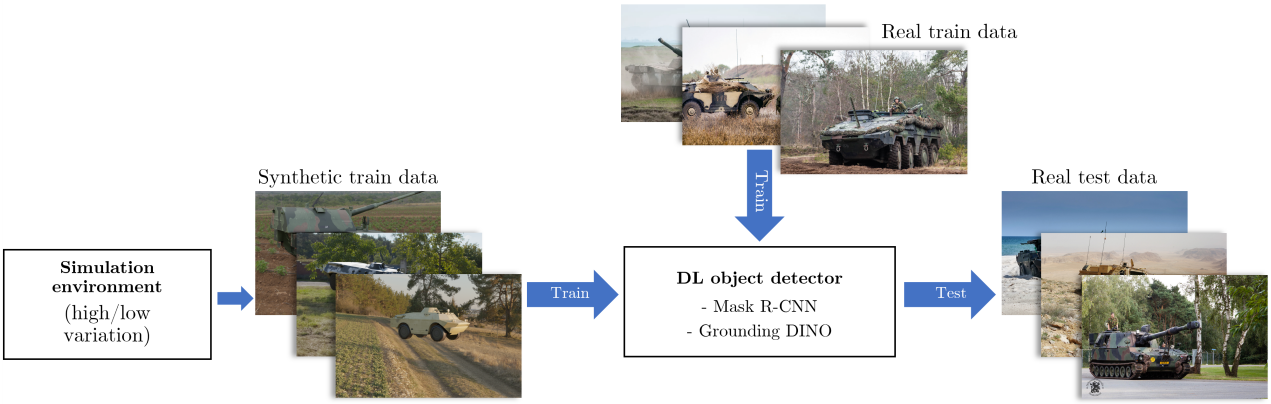

Figure 1: Overview of methods.

upon this, Grounding DINO[12] integrates ideas from both DINO and GLIP, improving zero-shot capabilities and downstream object detection performance. We use Grounding DINO as the second object model in this study.

The rich representations created by foundation models can be utilized for downstream tasks such as object detection, offering two main paths for adaptation to the specific task: zero-shot applications and transfer learning. In a zero-shot context, unsupervised vision models can utilize basic k-nearest neighbors (kNN) for object classification. This approach allows for the categorization of new objects based on the similarity of their feature representations to those of known objects, without requiring explicit example-based learning. Vision-language models further benefit from prompt engineering,[26,27] to guide the model to apply its learned representations to recognize and localize objects in images directly without labeled examples.

3. METHODS

An overview of our methods is depicted in Figure 1. Deep learning-based object detectors are trained using simulated or real data, or a combination of both sources. Evaluation is done on a separate set of real images. The experiments are performed with two object detection models: Mask R-CNN[10] and Grounding DINO.[12] We experiment with smaller sets of real training images as well as a simulated dataset that contains much less variation. In the next sections, we provide details about the datasets, object detection models, and experiments.

3.1 Datasets

For the fine-grained military vehicle detection, fifteen distinct types of military vehicles were selected. The list of vehicles is provided in Table 1 and includes armoured personnel carriers, reconnaissance vehicles, battle tanks, howitzers, and military trucks.

Table 1: Overview of the military vehicle classes.

Superclass	Classes
Armoured personnel carrier	Boxer, BTR-80, TPz Fuchs, Patria
Scout car	Fennek, BRDM-2
Battle tank	Leopard, M1 Abrams, T90, CV90
Howitzer	M109, 2S19 Msta, Panzerhaubitze 2000
Military truck	DAF YA 4440, Scania

3.1.1 Real data

A dataset of 959 real images was collected by scraping the internet. The data was split in sets of 360 train images (24 per class), 150 validation images (10 per class), and 449 test images (21-50 per class). The images

were annotated using the Computer Vision Annotation Tool (CVAT),[28] an open-source interactive annotation tool for videos and images, in combination with the Segment-Anything Model (SAM)[29] for fast semi-automatic segmentation of the military vehicles. Mask R-CNN requires ground truth segmentations in addition to bounding box annotations, while Grounding DINO only requires bounding boxes for training. In total 1,381 (488 train, 216 validation, 677 test) objects were annotated.

3.1.2 Simulation data

For generation of the simulated data, we used the same approach as Eker et al.[9] did for 13 classes. For this study we added the Scania and CV90 classes. In brief, 3D models were placed in High Dynamic Range (HDRI) scenes in Blender[30] and virtual pictures were acquired from various viewpoints. Images were generated with variation in the HDRI background scenes, object-camera distance, object yaw, pitch, and roll, model subtype, model texture, and model configuration. In total, 12,000 images (800 per class) were simulated, and we refer to this set as the *full variation* dataset.

To compare the effect of simulation variation (e.g. when only limited development time is available to construct a simulation pipeline) with the added value of including real data, we created a second simulated dataset. This *low variation* set also contains 800 images per class, but includes much less simulation variety. Table 2 shows the simulation settings of both datasets.

3.2 Models

The object detection models were implemented in the MMdetection framework,[31] which is based on a PyTorch backend.[32] Data augmentations were used for both the simulated and real dataset during training to increase data variability and prevent overfitting. Photometric distortions (contrast, saturation, and hue) are applied, followed by horizontal flipping, cropping, and scaling.

3.2.1 Mask R-CNN

Building upon our previous work,[9] we utilized a Mask R-CNN[10] with a SWIN-T transformer backbone,[11] with weights pre-trained on the COCO dataset.[33] The model was trained for 60,000 iterations (30 epochs) with batches of 12 images. The backbone was step-wise unfrozen in four stages after every 6 epochs. At the same epochs, the starting learning rate of 2×10^{-5} is decreased by a factor of four. A warm-up phase of 400 iterations was used before reaching the starting learning rate.

3.2.2 Grounding DINO

We chose MM-Grounding-DINO[34] as our foundation model, selecting a version pre-trained on a wide array of datasets including O365v1,[35] Flickr30K,[36] V3Det,[37] and GRIT,[38] and equipped with the SWIN-T backbone. We hypothesize that MM-grounding-DINO provides a superior starting point for fine-tuning on our real and synthetic military vehicle datasets. This is due to its training in an open-world, multi-modal setting which

Table 2: Overview of the simulation settings over the axes of variation for both simulated datasets. For a detailed explanation of the axes we refer to our previous study.[9]

Axis of variation	Full variation	Low variation
# of HDRI background scenes	100	4
Background resolution	12-24k	12-24k
Object-camera distance	Frequency inversely distributed with distance [10, 100] meter	21.6, 36 and 60 meters
Object yaw	Uniformly distributed [0, 360]°	0, 90, 180, 270°
Object pitch and roll	Roll rotation: $0 \pm \frac{\pi}{32}$ Pitch rotation: $\frac{\pi}{16} \pm \frac{\pi}{24}$	Pitch nor roll rotation
# of model textures	All available textures (3-4)	1 texture for all classes
# of model configurations	50% standard, 50% uniform distributed [min, max]	100% standard

improves the models generalization capabilities. Grounding DINO implements a end-to-end transformer based model, in contrast with the Mask R-CNN which only utilizes a transformed-based backbone. We fine-tuned this model for 5 epochs, since it convergences much quicker compared to the Mask R-CNN. The fine-tuning began with a learning rate of 1×10^{-5}, reduced by a factor of 10 at the start of the fourth epoch, without implementing a warm-up phase.

3.3 Experimental approach

We designed and executed a series of experiments to evaluate the value of simulated data in combination with a few real data samples, for both detection models. Every experiment is repeated three times to obtain a standard deviation of the performance metrics. In the first two experiments, only the simulated datasets with *full variation* and *low variation* were used for training. Next, we trained the models on the full real dataset of the 24 collected images, and on subsets of 8 and 2 samples. These subsets of the real training samples were made so that three different, non-overlapping, splits are used in the three repetitions. To explore the added value of combining both real and simulated data, the detector was also trained using mixed batches with equal amounts of real and simulated data. The combined experiments were repeated for both simulated datasets.

For all experiments, we wanted to ensure that the total dataset encompassed 24.000 images (800 real and 800 simulated images for all 15 classes for a combined experiment). Therefore, the real dataset was repeated (33, 100, and 400 times for 24, 8, and 2 real samples respectively). When training with either exclusively real or simulated data, we still required a dataset of 24.000 images in total. Thus, we double the aforementioned number of repeats for the real data or repeat the simulated data twice.

3.4 Evaluation

The mean average precision (mAP) is a popular metric for object detection tasks. It combines classification accuracy with localisation accuracy. The latter is determined by how well the predicted bounding boxes match the ground truth, measured by the Intersection over Union (IoU).[39] In addition, we compute the mAP50,[9] which is less strict in accepting the quality of the bounding box localisation and thus provides more insight in the classification accuracy. Both metrics are computed for the test set. The normalized confusion matrices are computed for each experiment, providing more information on how specific classes are predicted.

Table 3: The mAP and mAP50 of the two military vehicle detector networks trained using the different datasets with either simulated or real data, or a combination of both. LV = Low Variation.

	Mask R-CNN		**Grounding DINO**	
Training dataset	**mAP [± std]**	**mAP50**	**mAP**	**mAP50**
Simulated (LV)	19.2 [0.8]	28.1 [1.2]	23.3 [1.2]	25.8 [0.8]
Simulated	33.8 [0.7]	41.1 [0.8]	33.7 [3.2]	37.4 [3.6]
2 real	27.2 [2.8]	38.3 [3.7]	23.7 [3.8]	26.3 [3.9]
Simulated (LV) + 2 real	40.6 [2.4]	54.1 [2.3]	44.5 [6.3]	48.4 [6.5]
Simulated + 2 real	55.9 [2.6]	70.1 [2.1]	52.9 [4.5]	56.6 [4.3]
8 real	55.4 [1.0]	71.3 [1.6]	64.0 [1.3]	68.1 [1.8]
Simulated (LV) + 8 real	62.9 [0.9]	78.4 [0.3]	74.6 [2.0]	78.7 [2.4]
Simulated + 8 real	69.1 [1.2]	83.2 [1.1]	80.2 [1.0]	84.6 [0.6]
24 real	72.2 [0.6]	86.8 [0.5]	85.6 [1.1]	90.2 [1.0]
Simulated (LV) + 24 real	72.1 [2.0]	87.2 [0.9]	88.4 [0.3]	93.2 [0.0]
Simulated + 24 real	76.2 [0.5]	89.4 [0.4]	90.1 [0.5]	94.7 [0.4]

4. RESULTS

The mAP and mAP50 for the real test set images for all different training datasets and both models are presented in Table 3, including the standard deviation across the three iterations of each experiment. When only simulated data or only two real samples are used to train Mask R-CNN, the mAPs are 33.8 [±0.7] and 27.2

Figure 2: Bounding boxes predicted by the Mask R-CNN trained on 2 real samples in combination with the full variation simulated data, on test images containing the four battle tank classes. From left to right and top to bottom: M1A2 Abrams (photo by Staff. Sgt. Matthew Keeler/U.S. Army), T90, CV90 (photo by Martin Bos/DefensieFotografie Nederland), Leopard 2A6 (photo by 7th Army Training Command/Flickr). One Leopard is missed, the other detections are correct. The lines of the bounding boxes were thickened as a postprocessing step to enhance visibility.

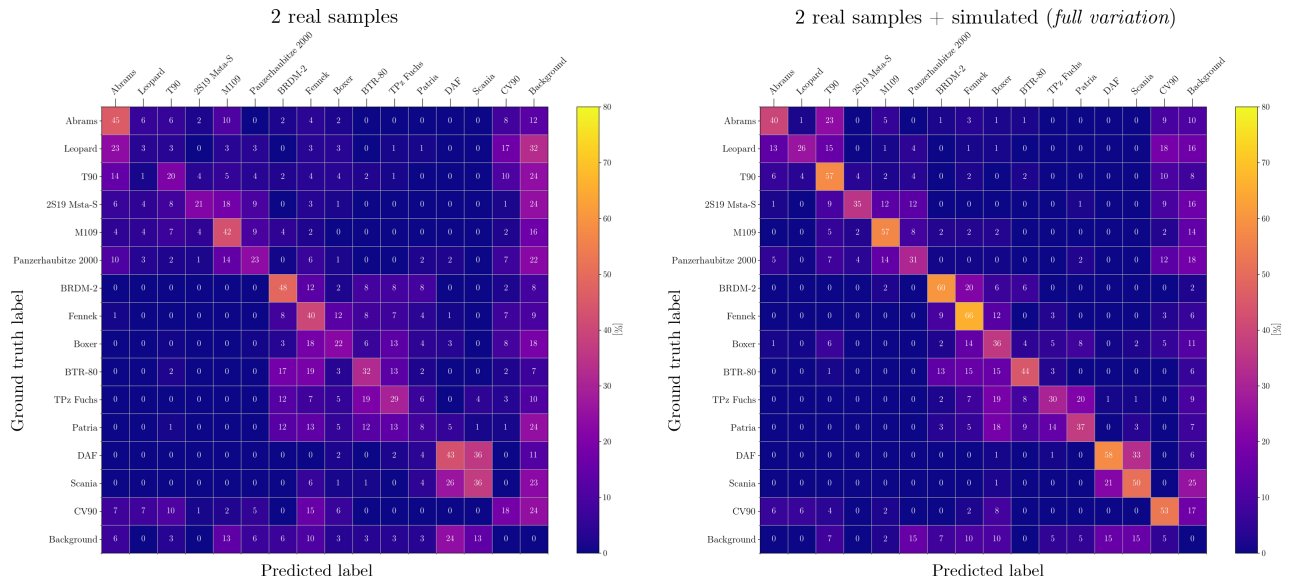

Figure 3: Confusion matrices for Mask R-CNN when trained on 2 real samples (left) and a combination of the simulated dataset with full variation and 2 real samples (right).

[±2.8] respectively. Combining both sources results in an mAP of 55.9 [±2.6] and in Figure 2 we show some

selected detections by this model for the different battle tanks (Leopard, M1 Abrams, T90, and CV90).

Figure 3 displays the confusion matrices for the Mask R-CNN models trained with 2 real samples only and trained with 2 real samples in combination with the simulated data. The improvement in mAP is clearly reflected by the differences between the confusion matrices. While some classes are almost never predicted correctly with only 2 real samples, this improves to substantial percentage when it is combined with simulated data, e.g. the accuracy for the *Leopard* class increases from 3% to 26%, for the *Patria* class from 8% to 37%.

Increasing the number of real samples further improves the mAP, both in combination with the simulated datasets and without. Figure 4 visualizes this trend for the experiments with Mask R-CNN. The effect of adding simulated data on top of real samples becomes smaller when more real samples are used. The results for the experiments with the *low variation* are all in between the experiments without simulated data and those with the *full variation* dataset.

Further improvements are obtained by using Grounding DINO as the object detection model. The best results are achieved when 24 real images per class are combined with full variation in the simulated dataset (mAP = 90.1 [±0.5]), which is substantially more than Mask R-CNN achieves for this combination (76.2 [±0.5]). Zooming in on the confusion matrices of Grounding DINO with 24 real images (Figure 5), we see a small benefit of adding simulated data. In comparison with the set without simulated data (mAP = 85.6 [±1.1]), we see minor improvements for correctly predicted objects for most classes and a reduction in missed detections.

Interestingly, when only very few real samples are available, Grounding DINO does not score better in our experiments. Mask R-CNN trained with two real samples in combination with *full variation* simulated data results in an mAP of 55.9 [±2.6] in comparison to 52.9 [±4.5] for Grounding DINO. When trained with only the *full variation* simulated data, mAP scores are similar for both models, while Grounding DINO score better when trained with the *low variation* set.

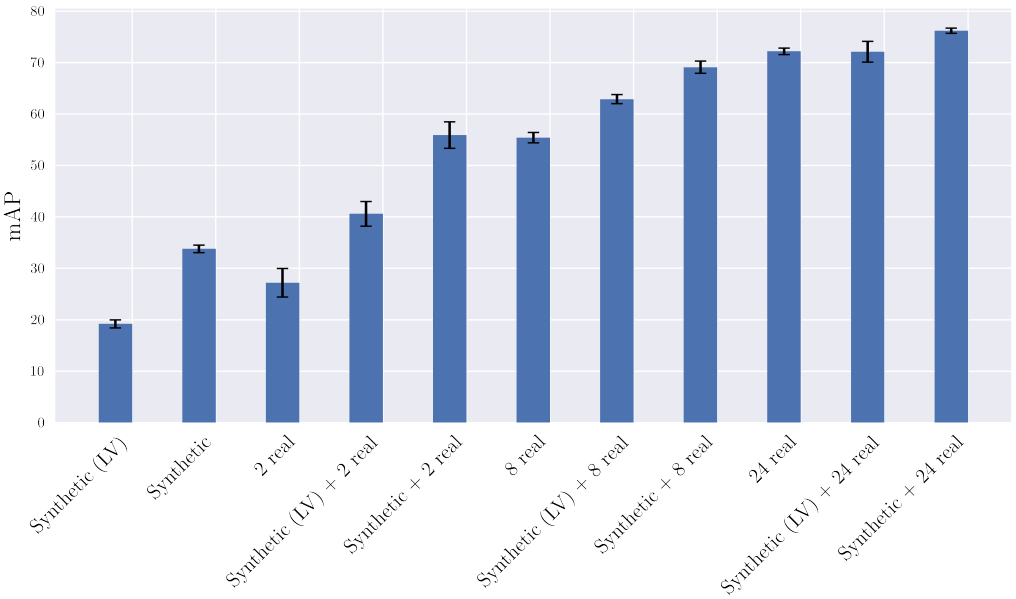

Figure 4: mAP scores for Mask R-CNN experiments with increasing number of real samples, combined with simulated data, either with full variation or low variation (LV). Error bars represent the observed standard deviation.

In general, repeating the experiments three times yields consistent results. Notable is the increase in standard deviation when training with a subset of only 2 real samples (2.6 - 3.0 percent) in comparison with a larger subset of 8 real samples or the whole set (ranging from 0.5 to 1.2 percent). mAP50 scores show a similar trend as mAP. The gap between the mAP and mAP50 indicates that part of the detection mistakes are related to the bounding boxes localization. These gaps are larger for Mask R-CNN than for Grounding DINO. This indicates that

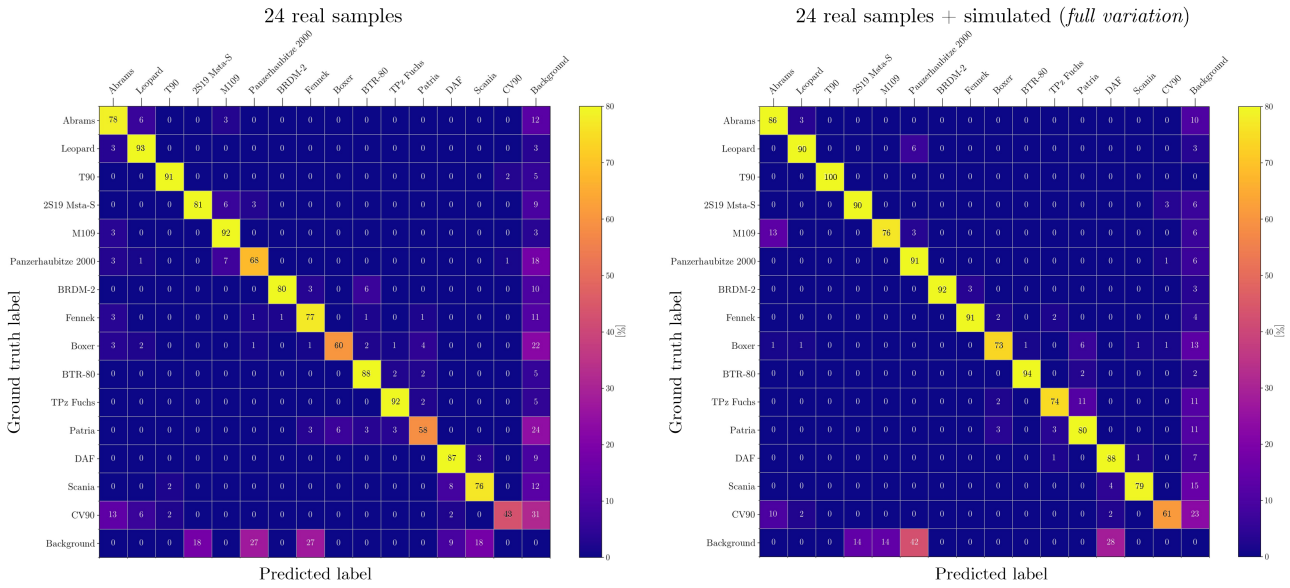

Figure 5: Confusion matrices for Grounding DINO when trained on 24 real samples (left) and a combination of the simulated dataset with full variation and 24 real samples (right).

the higher mAP scores for Grounding DINO are at least partly the result of more accurate bounding boxes predictions.

5. DISCUSSION

We previously showed that a deep learning-based military vehicle detector can be developed with only simulated data. While this detector performed well for four classes of quite distinct vehicles, extending the task to more classes substantially decreased performance.[9] The set of 15 vehicle types we use in this study contains similar-looking classes, such as multiple armoured personnel carriers, battle tanks, and howitzers, for which the nuances in appearance difference might not be well-represented in the simulated data. This limitation is likely related to the quality of the 3D models and other simulation factors, which could be addressed by spending more time on the simulation pipeline and simulated object details. However, taking into account limited time for simulation development, this might not be the most effective approach. Another strategy could be to collect more real data and use this to enhance the detectors performance. This presents an important trade-off: should one focus on generating and improving the simulated data, or on acquiring (more) real data?

In this study, we showed that for the development of a fine-grained military vehicle detector, combining few real data samples with a large amount of simulated data leads to a significant improvement in comparison with only using one of these sources. When only simulated data is used, even when using full variation in the simulation settings, an mAP score of 33.8 is obtained using Mask R-CNN, which is not sufficient for meaningful deployment. Similarly, a relatively low mAP score (27.2) is obtained when only two real samples are available per class. Combining the two sources leads to a substantial improvement (mAP = 55.9), which almost equals the sum of the mAPs for the two individual sources.

In the absence of simulated data, increasing the number of real samples from 2 to 8 and 24 improves the mAP substantially, which is line with the consensus that deep learning models benefit from more data. When more real samples are available, the addition of simulated data further improves performance, although the added benefit with Mask R-CNN seems to be smaller for 24 real samples.

Grounding DINO is a foundation model that has been pretrained on a large amount of data. We therefore hypothesized that this model might have a good performance even when providing a training dataset with only a small number of real samples per class. When using 24 real samples per class, Grounding DINO indeed shows

a high accuracy (mAP = 85.6) while Mask R-CNN scores 10 percents lower (mAP = 72.2). The benefit of Grounding DINO is even more apparent when 24 real samples are combined with the simulated dataset (mAP = 90.1, in comparison with 76.2 for Mask R-CNN). Interestingly, this advantage did not extend to scenarios with only 2 real samples per class, where Grounding DINO's performance was low (mAP = 23.7), even being slightly surpassed by Mask R-CNN (mAP = 27.2). This outcome suggests that, despite its potential for generalizability, Grounding DINO's large number of parameters may make it better suited to larger datasets. Alternatively, dedicating more time on prompt engineering the class names could potentially mitigate this performance shortfall, providing Grounding DINO with a more effective starting point for training with minimal data. Another possible explanation why Mask R-CNN works better for two real samples lies in the extra annotations that were available for Mask R-CNN. The ground truth segmentations, on top of the bounding box annotations, might have been helpful when very few examples were used for training.

The experiments with low and full simulation variation underline the importance of having sufficient simulation variation. In general, the mAP scores for experiments with the *low variation* dataset are in between training without simulated data and training with the *full variation* dataset. This indicated that, while less variation in the simulated data leads to a lower mAP, it still is beneficial in comparison with not adding any simulated data at all.

For all experiments with 2 and 8 real samples we used non-overlapping data splits to run three unique experiments. The confusion matrices of e.g. two real samples versus two real samples with simulated data were compared using the same splits. We noticed biases occurring in these comparisons, over- or underestimating specific classes. However, for a different split, the bias seems to be towards different classes, emphasizing the dependency on the specific samples when only 2 images are available per class. This might be caused by differences in image quality, or some samples not being very representative of the vehicle type.

In many situations, acquiring (more) real samples can be challenging or even impossible due to a lack of access. In such situations, further improvements can be achieved by enhancing the simulated data. This can be done by improving e.g., the details of the 3D models and background, and the configuration of the object in the image. Another strategy is to use generative AI to further augment the simulated images.[40] Potential solutions include image-to-image translation to generate more photo-realistic images,[41] local in-painting using Diffusion,[42] and image generation techniques where the simulated vehicle shape can be inputted as a prior, e.g. using Controllable Diffusion Models.[43]

The military vehicle detector presented in this paper was limited to a set of 15 classes, but given the excellent detection performance, we argue that the number of vehicles can be increased. Future research could also focus on investigating the effect of more difficult use-cases, for example when the objects of interest are further away or better camouflaged, or when challenging weather conditions make the images less clear.

In conclusion, we showed that a high-performing fine-grained military vehicle detector can be developed using Mask R-CNN and a large set of simulated data in combination with a few real samples. Further improvements can be achieved by using Grounding DINO, increasing the number of real images for training, and maximizing variation in the simulated dataset.

ACKNOWLEDGMENTS

The 3D models used in this study were provided by the Dutch Ministry of Defense.

REFERENCES

[1] van Rooijen, A., Bouma, H., Baan, J., and van Leeuwen, M., "Rapid person re-identification retraining strategy for flexible deployment in new environments," in [*Counterterrorism, Crime Fighting, Forensics, and Surveillance Technologies VI*], **12275**, 122750D, International Society for Optics and Photonics, SPIE (2022).

[2] Brohan, A., Brown, N., Carbajal, J., Chebotar, Y., Chen, X., Choromanski, K., Ding, T., Driess, D., Dubey, A., Finn, C., Florence, P., Fu, C., Arenas, M. G., Gopalakrishnan, K., Han, K., Hausman, K., Herzog, A., Hsu, J., Ichter, B., Irpan, A., Joshi, N., Julian, R., Kalashnikov, D., Kuang, Y., Leal, I., Lee, L., Lee,

T.-W. E., Levine, S., Lu, Y., Michalewski, H., Mordatch, I., Pertsch, K., Rao, K., Reymann, K., Ryoo, M., Salazar, G., Sanketi, P., Sermanet, P., Singh, J., Singh, A., Soricut, R., Tran, H., Vanhoucke, V., Vuong, Q., Wahid, A., Welker, S., Wohlhart, P., Wu, J., Xia, F., Xiao, T., Xu, P., Xu, S., Yu, T., and Zitkovich, B., "Rt-2: Vision-language-action models transfer web knowledge to robotic control," in [*arXiv preprint arXiv:2307.15818*], (2023).

[3] Heslinga, F. G., Ruis, F., Ballan, L., van Leeuwen, M. C., Masini, B., van Woerden, J. E., den Hollander, R. J. M., Berndsen, M., Baan, J., Dijk, J., and Huizinga, W., "Leveraging temporal context in deep learning methodology for small object detection," in [*Artificial Intelligence for Security and Defence Applications*], **12742**, SPIE Sensors + Imaging (2023).

[4] Lecun, Y., Bengio, Y., and Hinton, G., "Deep learning," *Nature* **521**, 436–444 (2015).

[5] Chen, Y., Li, W., Chen, X., and Van Gool, L., "Learning semantic segmentation from synthetic data: A geometrically guided input-output adaptation approach," in [*2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*], 1841–1850 (2019).

[6] Heslinga, F. G., Uysal, F., van Rooij, S. B., Berberich, S., and Caro Cuenca, M., "Few-shot learning for satellite characterization from synthetic ISAR images," *IET Radar, Sonar & Navigations* , 1–8 (2024).

[7] Zhang, L., Gonzalez-Garcia, A., van de Weijer, J., Danelljan, M., and Khan, F. S., "Synthetic data generation for end-to-end thermal infrared tracking," *IEEE Transactions on Image Processing* **28**, 1837–1850 (2018).

[8] Azizi, S., Kornblith, S., Saharia, C., Norouzi, M., and Fleet, D. J., "Synthetic data from diffusion models improves ImageNet classification," *ArXiv* **abs/2304.08466** (2023).

[9] Eker, T. A., Heslinga, F. G., Ballan, L., den Hollander, R. J., and Schutte, K., "The effect of simulation variety on a deep learning-based military vehicle detector," in [*Artificial Intelligence for Security and Defence Applications*], **12742**, 183–196, SPIE Sensors + Imaging (2023).

[10] He, K., Gkioxari, G., Dollár, P., and Girshick, R., "Mask R-CNN," (2018).

[11] Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, S., and Guo, B., "Swin transformer: Hierarchical vision transformer using shifted windows," (2021).

[12] Liu, S., Zeng, Z., Ren, T., Li, F., Zhang, H., Yang, J., Li, C., Yang, J., Su, H., Zhu, J., and Zhang, L., "Grounding DINO: Marrying DINO with grounded pre-training for open-set object detection," (2023).

[13] Redmon, J., Divvala, S., Girshick, R., and Farhadi, A., "You Only Look Once: Unified, real-time object detection," in [*2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*], 779–788 (2016).

[14] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., and Berg, A. C., "SSD: Single shot multibox detector," in [*Computer Vision - ECCV*], 21–37, Springer International Publishing (2016).

[15] Girshick, R., Donahue, J., Darrell, T., and Malik, J., "Rich feature hierarchies for accurate object detection and semantic segmentation," in [*Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*], 580–587 (2014).

[16] Ren, S., He, K., Girshick, R., and Sun, J., "Faster R-CNN: Towards real-time object detection with region proposal networks," in [*Advances in Neural Information Processing Systems*], Cortes, C., Lawrence, N., Lee, D., Sugiyama, M., and Garnett, R., eds., **28** (2015).

[17] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., and Houlsby, N., "An image is worth 16x16 words: Transformers for image recognition at scale," (2021).

[18] Ruis, F. A., Liezenga, A. M., Heslinga, F. G., Ballan, L., den Hollander, R. J., van Leeuwen, M. C., Masinia, B., Dijk, J., and Huizinga, W., "Improving object detector training on synthetic data by starting with a strong baseline methodology," in [*Synthetic Data for Artificial Intelligence and Machine Learning: Tools, Techniques, and Applications II*], **13035**, SPIE Defense + Commercial Sensing (2024).

[19] Nowruzi, F. E., Kapoor, P., Kolhatkar, D., Hassanat, F. A., Laganiere, R., and Rebut, J., "How much real data do we actually need: Analyzing object detection performance using synthetic and real data," *arXiv preprint arXiv:1907.07061* (2019).

[20] Seib, V., Lange, B., and Wirtz, S., "Mixing real and synthetic data to enhance neural network training–a review of current approaches," *arXiv preprint arXiv:2007.08781* (2020).

[21] Chen, T., Kornblith, S., Norouzi, M., and Hinton, G., "A simple framework for contrastive learning of visual representations," in [*International conference on machine learning*], 1597–1607, PMLR (2020).

[22] Chen, T., Kornblith, S., Swersky, K., Norouzi, M., and Hinton, G., "Big self-supervised models are strong semi-supervised learners," *arXiv preprint arXiv:2006.10029* (2020).

[23] Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., and Joulin, A., "Emerging properties in self-supervised vision transformers," in [*Proceedings of the International Conference on Computer Vision (ICCV)*], (2021).

[24] Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., and Sutskever, I., "Learning transferable visual models from natural language supervision," *CoRR* **abs/2103.00020** (2021).

[25] Li, L. H., Zhang, P., Zhang, H., Yang, J., Li, C., Zhong, Y., Wang, L., Yuan, L., Zhang, L., Hwang, J.-N., Chang, K.-W., and Gao, J., "Grounded language-image pre-training," in [*CVPR*], (2022).

[26] Lester, B., Al-Rfou, R., and Constant, N., "The power of scale for parameter-efficient prompt tuning," in [*Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*], Moens, M.-F., Huang, X., Specia, L., and Yih, S. W.-t., eds., 3045–3059, Association for Computational Linguistics, Online and Punta Cana, Dominican Republic (Nov. 2021).

[27] Jia, M., Tang, L., Chen, B.-C., Cardie, C., Belongie, S., Hariharan, B., and Lim, S.-N., "Visual prompt tuning," in [*European Conference on Computer Vision*], 709–727, Springer (2022).

[28] Opencv, "Opencv/cvat: Annotate better with CVAT, the industry-leading data engine for machine learning. used and trusted by teams at any scale, for data of any scale."

[29] Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A. C., Lo, W.-Y., et al., "Segment Anything," *arXiv preprint arXiv:2304.02643* (2023).

[30] Community, B. O., *Blender - a 3D modelling and rendering package*. Blender Foundation, Stichting Blender Foundation, Amsterdam (2018).

[31] Chen, K., Wang, J., Pang, J., Cao, Y., Xiong, Y., Li, X., Sun, S., Feng, W., Liu, Z., Xu, J., Zhang, Z., Cheng, D., Zhu, C., Cheng, T., Zhao, Q., Li, B., Lu, X., Zhu, R., Wu, Y., Dai, J., Wang, J., Shi, J., Ouyang, W., Loy, C. C., and Lin, D., "MMDetection: Open mmlab detection toolbox and benchmark," *arXiv preprint arXiv:1906.07155* (2019).

[32] Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Köpf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S., [*PyTorch: An Imperative Style, High-Performance Deep Learning Library*] (2019).

[33] Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., and Zitnick, C. L., "Microsoft COCO: Common objects in context," in [*Computer Vision – ECCV 2014*], Fleet, D., Pajdla, T., Schiele, B., and Tuytelaars, T., eds., 740–755 (2014).

[34] Zhao, X., Chen, Y., Xu, S., Li, X., Wang, X., Li, Y., and Huang, H., "An open and comprehensive pipeline for unified object grounding and detection," *arXiv preprint arXiv:2401.02361* (2024).

[35] Shao, S., Li, Z., Zhang, T., Peng, C., Yu, G., Zhang, X., Li, J., and Sun, J., "Objects365: A large-scale, high-quality dataset for object detection," in [*2019 IEEE/CVF International Conference on Computer Vision (ICCV)*], 8429–8438 (2019).

[36] Young, P., Lai, A., Hodosh, M., and Hockenmaier, J., "From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions," *Transactions of the Association for Computational Linguistics* **2**, 67–78 (2014).

[37] Wang, J., Zhang, P., Chu, T., Cao, Y., Zhou, Y., Wu, T., Wang, B., He, C., and Lin, D., "V3Det: Vast vocabulary visual detection dataset," in [*The IEEE International Conference on Computer Vision (ICCV)*], (October 2023).

[38] Gupta, T., Marten, R., Kembhavi, A., and Hoiem, D., "Grit: General robust image task benchmark," *ArXiv* **abs/2204.136533** (2022).

[39] Padilla, R., Netto, S. L., and Da Silva, E. A., "A survey on performance metrics for object-detection algorithms," in [*2020 international conference on systems, signals and image processing (IWSSIP)*], 237–242, IEEE (2020).

[40] Zhan, F., Yu, Y., Wu, R., Zhang, J., Lu, S., Liu, L., Kortylewski, A., Theobalt, C., and Xing, E., "Multimodal image synthesis and editing: The generative ai era," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **45**(12), 15098–15119 (2023).

[41] Zhang, C. and Shrivastava, A., "AptSim2Real: Approximately-paired sim-to-real image translation," *ArXiv* **abs/2303.12704** (2023).

[42] Yang, B., Gu, S., Zhang, B., Zhang, T., Chen, X., Sun, X., Chen, D., and Wen, F., "Paint by example: Exemplar-based image editing with diffusion models," *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* , 18381–18391 (2022).

[43] Fang, H., Han, B., Zhang, S., Zhou, S., Hu, C., and Ye, W.-M., "Data augmentation for object detection via controllable diffusion models," in [*Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*], 1257–1266 (2024).