Automatic Slip Sheet Recognition in Warehouse using Sim-To-Real Transfer Learning

Kai Schlauersbach

Supervisors: Prof. Dr. Andreas Thümmel, Prof. Dr. Timo Schürg

Darmstadt University of Applied Sciences

Introduction

In the development of autonomous *Order Picking Systems (OPS)* like *KION Group's RoCaP* robot, detecting slipsheets (thin separation layers between product layers) emerged as a key challenge. These slipsheets obstruct product visibility and accessibility, preventing the robot from continuing its picking process. A promising solution is training a Deep Neural Network (DNN) to detect or classify slipsheets in camera images. However, collecting sufficient labeled training data is difficult, especially in real warehouse environments. To address this, synthetic data generation using simulation tools like *NVIDIA Omniverse* [3] becomes a viable alternative. Its Replicator toolkit enables the randomized creation of labeled image datasets under varied conditions. This allows training visual models (e.g., *YOLO* [2], *EfficientNet* [4] or *ResNet* [1]) using simulated data, with the goal of achieving strong performance on real-world tasks, also known as Sim-to-Real Transfer Learning.

Research Questions

This thesis revolves around three main research questions.

- Can synthetic data be used to effectively fine-tune pre-trained visual models for real-world tasks and what is the best way of creating synthetic data?
- To what extent does supplementing synthetic data with real-world data improve model accuracy, and what strategies best minimize the simulation-to-reality gap?
- To what extent does data augmentation of real data contribute to improved generalization in models trained with hybrid datasets?

Synthetic Data Generation (SDG)

To create synthetic data usable for the training of DNN the Simulation-to-Reality Gap has to be addressed. This involves two main challenges:

- The *Appearance Gap*, which refers to visual differences between real and synthetic images due to rendering limitations or material complexity.
- The *Content Gap*, which concerns differences in object variety, placement, and scene diversity between simulations and real-world scenarios.

To overcome these gaps, rendering quality must be improved, realistic materials used and scene variability increased. Domain randomization, the continuous and varied alteration of scenes, is a key technique to help models generalize to real-world data.
To achieve this, two different simulations were developed using the *NVIDIA Omniverse Replicator* framework. These simulations included various warehouse models, packing pattern algorithms to determine the optimal arrangement of products on a pallet, and a full physics simulation to realistically model slipsheets.

Figure 1. Example image simulator approach 1

Figure 2. example image simulator approach 2

Data Augmentation of Real Data

In addition to the synthetic dataset generated through simulation, a small set of real-world data was collected for testing and fine-tuning. These recordings were conducted both in a live warehouse environment to ensure high realism, and in a test facility to capture data from a different domain.
To increase the size of the dataset *Data Augmentation* in different levels was applied to test which level scores the best results. The pipeline first applies a *horizontal flip*, then *dropouts*, *channel dropouts*, *affine transformations* and lastly *domain specific augmentations* (e.g. noise, blur).

Results

First it was examined if training on only synthetic data already leads to models usable in real-world applications. For this a *ResNet-50*, *EfficientNet-B0/-B1* and *YOLO11s-cls* algorithm were trained and tested. Then different approaches of mixing real and synthetic dataset and cross domain generalization were studied.

Figure 3. Comparison of models trained on synthetic data

All models perform subpar and are not sufficient for real-world use. Surprisingly the simplest model in *ResNet-50* performs the best and the more complex models cannot hold up. The last model shown in Figure 3 is another *YOLO11s-cls* model trained on synthetic data that was augmented to more closely match images recorded by a real camera.
To achieve better results real data had to be used in training in addition to the synthetic data. Two different methods for including the real data were tested: Mixing synthetic and real data and training on one big dataset and fine-tuning the models trained on synthetic data with the smaller real datasets.

The simpler models (*ResNet*, *EfficientNet*) all achieve better results when using the fine-tune method of integrating real data. The *YOLO11s-cls* model on the other hand performs better when applying the full train method. A reason for that may be the more complex layers used in this architecture, mainly the attention mechanism that needs a lot of training data to be be trained for a specific use case.

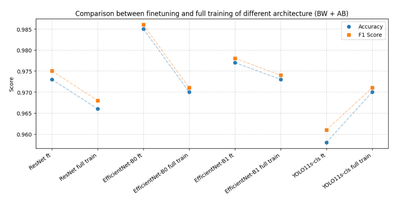

Figure 4. Comparison of full train and ft models

In addition to the different datasets used, cross domain tests were conducted using models trained only on real data taken from one domain and tested on another unknown domain while applying different levels of *data augmentation* to the real data. The results are as expected and the models perform well on the domain included in the training data and worse on the unknown domain. This results is observed over all architectures while different *data augmentation* levels are able to improve metrics slightly for some models. So when transferring the *RoCaP* robot to a new domain new data has to be recorded and a new model specialized on this domain has to be trained.

Results

The final model used for the integration in the *RoCaP* robot was an *EfficientNet-B0* model pretrained on synthetic data and fine-tuned on an unaugmented real dataset taken from all domains. The accuracy of the final model is 98.5% and the F1-score is 0.986. The training time needed for this model are around $1.54h$ for the training on synthetic data and an additional $7min$ for the fine-tuning. All misclassified images are borderline cases, suggesting that further hyperparameter tuning could improve performance.

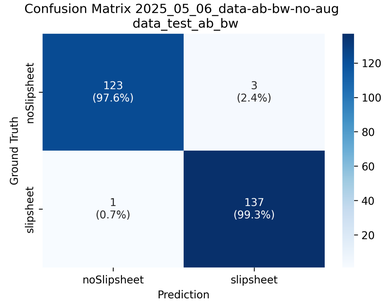

Figure 5. Confusion matrix final model

Conclusion

This study examined the effectiveness of synthetic data for training visual models, its combination with real-world data, and the impact of *data augmentation*. It was found that relying solely on synthetic data generated in *NVIDIA Omniverse* was insufficient for real-world performance, with the best model achieving only 87.5% accuracy. The high visual similarity between slipsheets and certain product layers likely contributed to these limitations. Incorporating even a small amount of real data significantly improved results. Fine-tuning pre-trained models with just 392 real images increased accuracy up to 98.5%, proving that real-world data is essential for robust model performance. Both training strategies (mixed datasets and fine-tuning) showed benefits, with their effectiveness depending on the model architecture. Additionally, *data augmentation* of real images helped improve cross-domain generalization, especially when training on data collected in a live warehouse environment. Although no single augmentation level outperformed the others, applying it consistently led to more stable results. Overall, combining synthetic data with augmented real-world data is key to achieving high-performing and generalizable models.
Future work includes integrating the model into *RoCaP's* picking process via a *Docker* container using *ROS2* camera data. While slipsheet removal is currently manual, automation is planned. Model efficiency can be improved using *TensorRT*. Enhancing the simulation with more realistic physics, textures, and object variety could enable fully synthetic training. Additionally, generating bounding boxes or keypoints would allow for object detection training. A two-phase classification approach could improve results on difficult cases. Finally, performance across domains may benefit from either fine-tuning per environment or creating a diverse, combined dataset for better generalization.

References

[1] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition, December 2015. URL `http://arxiv.org/abs/1512.03385`. arXiv:1512.03385 [cs].

[2] Glenn Jocher and Jing Qiu. Ultralytics yolo11, 2024. URL `https://github.com/ultralytics/ultralytics`.

[3] NVIDIA. NVIDIA Omniverse. URL `https://www.nvidia.com/de-de/omniverse/`.

[4] Mingxing Tan and Quoc V. Le. Efficientnet: Rethinking model scaling for convolutional neural networks, 2020. URL `https://arxiv.org/abs/1905.11946`.