Analysis of Training Object Detection Models with Synthetic Data

Bram Vanherle, Steven Moonen, Frank Van Reeth, Nick Michiels

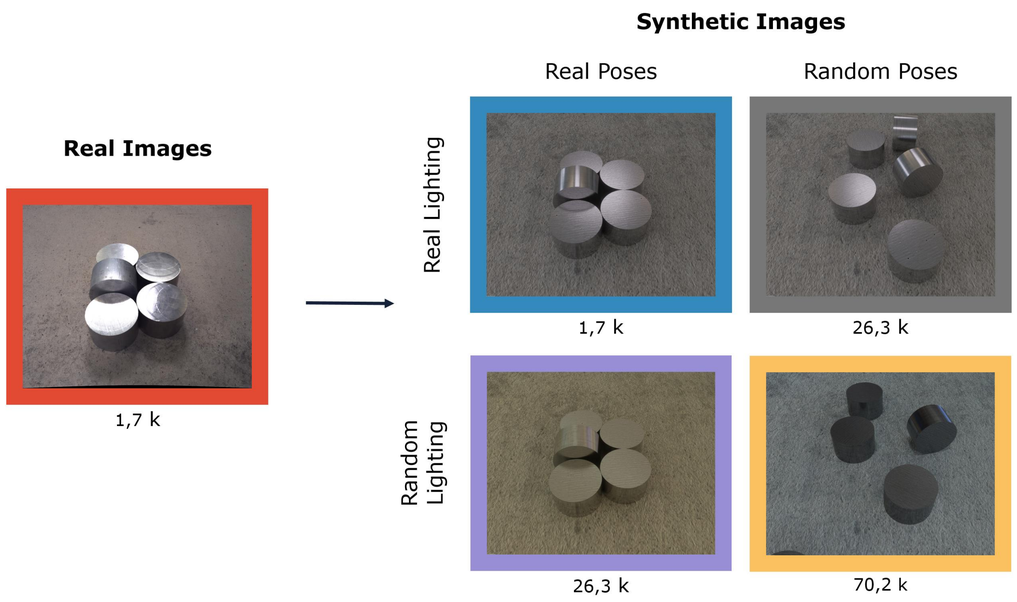
Hasselt University - tUL - Flanders Make - Expertise Centre for Digital Media
{firstname}.{lastname}@uhasselt.be

Introduction

Object Detection models are **data hungry**. Manual annotations are costly and error prone. **Synthetic training data** can provide cheap datasets, but trained models suffer from the **domain gap**. In this work we investigate if **matching the characteristics of the target domain** in the synthetic data is beneficial to generalization or whether some aspects can be left random. We combine this with modern deep learning techniques to provide a holistic analysis on how to efficiently train object detection models on synthetic data without losing too much accuracy.

The Dataset

For our experiments we use the DIMO dataset [1]. This dataset contains real images depicting five classes of objects. Additionally, the dataset contains four synthetic sets. The first synthetic dataset is an **exact digital twin** of the real dataset. The second and third datasets are digital twins, but with either **random poses or light conditions**. The fourth dataset has **random poses and random light**. These unique datasets allow us to research the impact of these variations in an isolated manner.



Experimental Setup

We trained many object detection models using **different datasets and deep learning techniques**. To test their ability to generalize, the trained models were validated on a test set of real images.
Mask-RCNN is used with a ResNet101 backbone. When transfer learning is mentioned, the model is initialized with weights trained on COCO and the backbone is frozen. Models are trained for **100 epochs** with Stochastic Gradient Descent using a **learning rate of 0.001** and a momentum of 0.9. A batch size of four is used and each epoch 1.000 images are used to train the model.

Conclusion

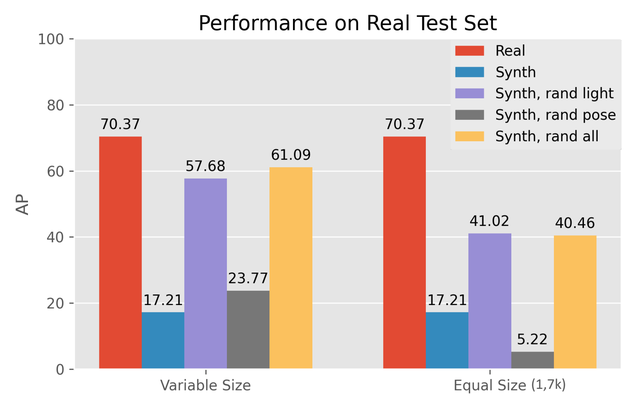
- **Recreating the target domain** in terms of object poses and lighting conditions **is helpful** when training on synthetic data, but only when transfer learning is used.
- **Modelling real object poses** in the training data leads to a larger **increase in performance** than modelling light conditions. We speculate this is due to the fact that higher level features such as shape are easier to simulate when rendering compared to low level features such as texture and light.
- Contrary to other research [2], we find that when using transfer learning, it can be beneficial to **retrain the entire network.**
- Where previous work advocated for more randomization [3], we conclude that the **benefit of adding more random synthetic images to a dataset is limited**. It is better to add more relevant synthetic images. Even better, is to add real images. When doing so, one should use finetuning.

[1] Dataset of Industrial Metal Objects, De Roovere et al.
[2] On Pre-Trained Image Features and Synthetic Images for Deep Learning, Hinterstoisser et al.
[3] Training Deep Networks with Synthetic Data: Bridging the Reality Gap by Domain Randomization, Tremblay et al.

Results

Does Scene Composition Matter?

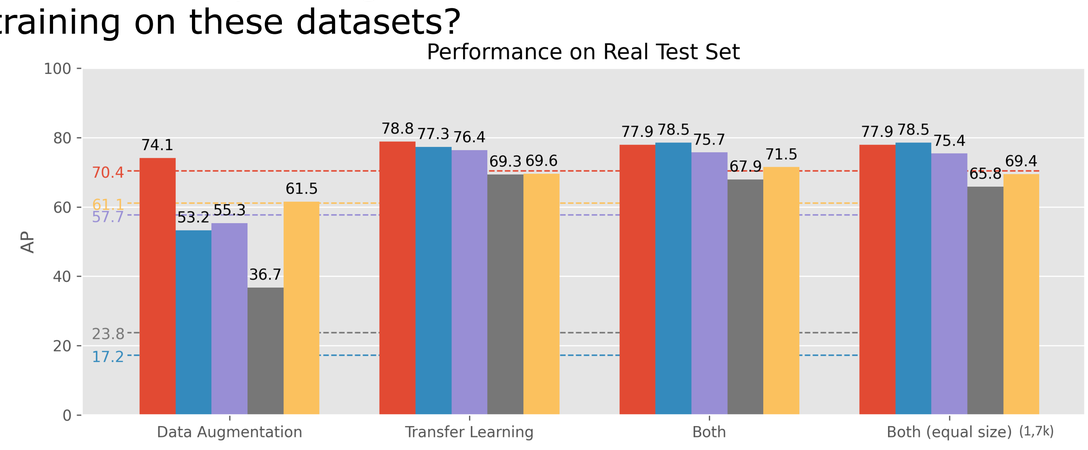
Is it beneficial to **model synthetic datasets to be similar to the target** domain in terms of object poses and lighting conditions?



- Matching Lighting and Poses hurts model performance.
- Randomizing lighting helps generalization

Impact of learning techniques?

Data augmentation (DA) and transfer learning (TL) have shown to improve generalization. What are their effects when training on these datasets?



- DA and TL especially are **very helpful**
- **Real pose and real light is now the best model**
- **Decreasing dataset** size only leads to small performance loss

What layers to retrain?

Layers Retrained	AP
All	**81,26**
Stage 3+	76,71
Stage 4+	80,77
Stage 5+	77,13
Heads	71,52

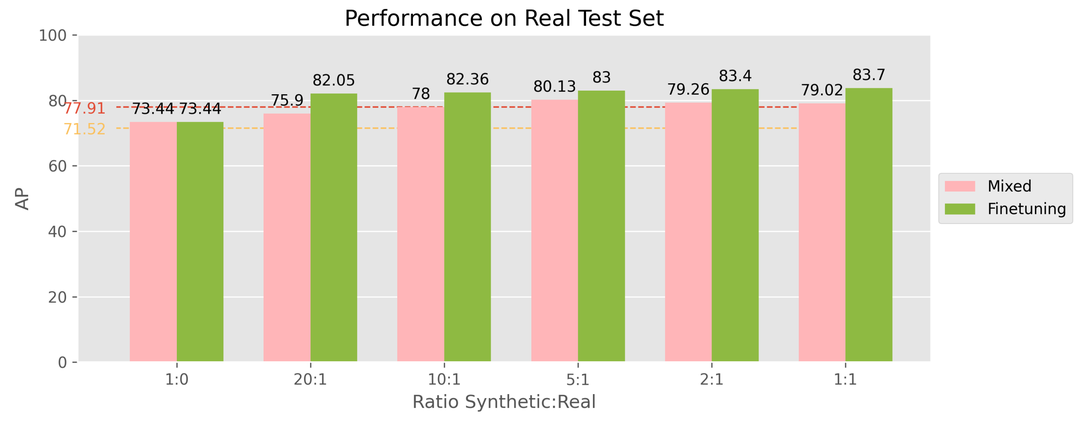
Retraining the entire network on fully random images leads to **large performance gains** compared to only the heads.

Do we need a lot of images?

Image Count	AP
1755	69,42
4387	72,23
8775	72,58
17550	**73,01**
35100	72,01
70200	71,52

Using a **small number of fully random images already gives decent performance**. Adding more increases the performance slightly. After 17k images, there is no performance gain anymore.

How to leverage real images?



Using a **small number of real images** together with fully random images leads to a **large performance increase**. **Finetuning works better** than mixing the datasets.

This study was supported by the Special Research Fund (BOF) of Hasselt University. The mandate ID is BOF20OWB24. Research was done in alignment with Flanders Make's PILS SBO project (R-9874).