UNIVERSITY OF TARTU
Institute of Computer Science
Computer Science Curriculum

Hannes Liik

Sim-to-Real Generalization of Computer Vision with Domain Adaptation, Style Randomization, and Multi-Task Learning

Master's Thesis (30 ECTS)

Supervisor: Tambet Matiisen, MSc

Tartu 2020

Sim-to-Real Generalization of Computer Vision with Domain Adaptation, Style Randomization, and Multi-Task Learning

Abstract:
In recent years, supervised deep learning has been very successful in computer vision applications. This success comes at the cost of a large amount of labeled data required to train artificial neural networks. However, manual labeling can be very expensive.

Semantic segmentation, the task of pixel-wise classification of images, requires painstaking pixel-level annotation. The particular difficulty of manual labeling for semantic segmentation motivates research into alternatives.

One solution is to use simulations, which can generate semantic segmentation ground truth automatically. Unfortunately, in practice, simulation-trained models have been shown to generalize poorly to the real world.

This work considers a simulation environment, used to train models for semantic segmentation, and real-world environments to evaluate their generalization. Three different approaches are studied to improve generalization from simulation to reality. Firstly, using a generative image-to-image model to make the simulation look realistic. Secondly, using style randomization, a form of data augmentation using style transfer, to make the model more robust to change in visual style. Thirdly, using depth estimation as an auxiliary task to enable learning of geometry.

Our results show that the first method, image-to-image translation, improves performance on environments similar to the simulation. By applying style randomization, the trained models generalized better to completely new environments. The additional depth estimation task did not improve performance, except by a small amount when combined with style randomization.

Keywords:
Computer Vision; Machine Learning; Deep Learning; Domain Adaptation; Data Augmentation; Multi-Task Learning; Convolutional Neural Networks

CERCS: P176 Artificial intelligence

Simulatsioonist pärismaailma üldistumine masinnägemises kasutades domeenikohandamist, stiili randomeerimist ja mitme ülesande õppimist

Lühikokkuvõte: Juhendamisega sügavõpe on viimastel aastatel olnud väga edukas masinnägemise ülesannetes. Selle saavutamiseks on vaja suurt hulka märgendatud andmeid. Käsitsi suurte andmestike annoteerimine võib olla väga kulukas. Semantiline segmenteerimine, ehk piltide pikslite klassifitseerimine, on näide ülesandest, mis vajab eriti töömahukat pikslitäpsusega märgendamist. Üks viis hoiduda käsitsi märgendamisest on kasutada simulatsiooni, mis teeb seda automaatselt. Praktika on aga näidanud, et simulatsiooni andmetel treenitud mudelid ei üldistu pärismaailma. Käesolevas töös uuritakse kolme meetodit, kuidas suurendada semantilise segmentatsiooni tehisnärvivõrkude üldistuvust simulatsioonist pärismaailma. Esiteks uuritakse generatiivse tehisnärvivõrgu kasutamist, et muuta andmestiku väljanägemist realistlikumaks (domeenikohandamine). Tulemused näitavad, et see meetod on efektiivne siis, kui pärismaailma keskkond on sarnane simulatsioonile. Teiseks uuritakse stiilide varieerimist, et muuta segmenteerimise närvivõrku vähem tundlikuks visuaalsetele muutustele. Stiilirandomeerimist kasutades üldistus treenitud tehisnärvivõrk ka keskkondadele, mis ei sarnane simulatsioonile. Kolmandaks katsetati lisaks segmentatsioonile ka pikslite sügavuse ennustamist lisaülesandena. Praktikas oli mitme ülesande korraga treenimist raske tööle saada ning näitas vähest kasu.

Võtmesõnad:

Masinnägemine; masinõpe, sügavõpe, domeenikohandamine; andmete rikastamine; mitme ülesande õppimine; konvolutsioonilised tehisnärvivõrgud

CERCS: P176 Tehisintellekt

Contents

1 Introduction

1.1 Deep Learning for Computer Vision

Deep learning is a sub-field of machine learning that uses artificial neural networks to extract high level representations of raw inputs. These representations are learned from data as opposed to being hand-crafted. Deep learning has been especially successful in computer vision - understanding and representing information from visual sensors. For example, the ImageNet Large Scale Visual Recognition Challenge [1] has been dominated by Convolutional Neural Networks (CNNs) [2]. The first such submission was AlexNet in 2012 [3]. While powerful, neural networks require large amounts of data to train. The ImageNet challenge, for example, uses approximately 1.2 million training images. For many tasks, such large datasets do not exist or are only permitted for use in academic research.

1.2 Semantic Segmentation

This work focuses on the task of semantic segmentation. Semantic segmentation is the pixel-wise classification of an image. It is considered to be one of the most complete perception tasks. Its dense and general representational capability makes it a perfect fit for robotics tasks, such as self-driving. An example of semantic segmentation from the Cityscapes [4] dataset is shown in Figure 1.

Unfortunately, labeling images for semantic segmentation is very tedious and time consuming. This results in a high cost per image, inhibiting creation of large datasets needed for deep learning. One way to get around expensive manual labeling is to use simulation environments to automatically extract semantic segmentation ground truth.

Figure 1. Example of semantic segmentation from the Cityscapes [4] dataset. Each pixel is overlayed with a color representing its class, e.g. red - human.

1.3 Simulation for training

We would like to use simulations for training neural networks for semantic segmentation because:

- Dense, expensive labelling (e.g. semantic segmentation) can be done automatically.
- It is possible to get accurate ground-truth for tasks such as monocular depth or surface normal estimation.
- Large amounts of samples can be generated at low cost.

Unfortunately, neural networks trained on simulation datasets have been shown to generalize poorly to the real world [5]. This is due to a gap between simulation and reality. Creating true to life environments in simulation is very difficult and expensive. For this reason, it is worth researching methods to increase generalization from environments which do not accurately match the real world.

Data augmentation can be applied to encourage trained models to learn features which generalize better. Domain adaptation is the transfer of knowledge learned for one environment (source domain) to a new one (target domain). In this work, we are interested in unsupervised domain adaptation in particular. This means we have training data with labels for the source domain (simulation) and unlabeled samples from the target domain (real world). Alternatively, training on multiple tasks should act as regularization, as the network needs to find a more efficient representation. Furthermore, more tasks means more learning signal. This thesis considers domain adaptation, data augmentation, and multi-task learning to achieve simulation to reality generalization.

1.4 Contributions

The contributions of this work are the following:

1. Unsupervised domain adaptation is evaluated and shown effective between a simulation and matching real world environment, but does not learn inductive representations that generalize well to other real world environments.
2. It is shown that a simple approach, learning a shape-based representation using style randomization [6,7], generalizes well to new environments. Furthermore, this approach even beats the tested domain adaptation approach on the target domain.
3. Finally, monocular depth estimation is tested as an auxiliary task. Though having potential, it is shown that this method is difficult to apply.

1.5 Contents

The contents of this work are organised as follows. First, the background of this thesis is covered in Section 2. Then, the datasets and model used in the experiments of this work are introduced in Section 3. In the main body of this work, three different approaches are explored in Sections 4, 5, and 6. The final results are combined in Section 7, followed by conclusions in Section 8.

2 Background

2.1 Simulations

A lot of work has been put into simulations and simulation-extracted datasets in the recent years. For example, CARLA is a simulation environment for self driving cars, which includes semantic segmentation and depth in addition to RGB images [8], shown in Figure 2. Virtual KITTI 2 is large simulation dataset which includes ground truth computer vision tasks such as semantic segmentation, depth and optical flow [9]. Video games, such as GTA 5, have also been used to extract datasets with rich variety [10]. This thesis uses a custom simulation environment (not created as a part of this thesis) made using AirSim [11].

Relatedly, reinforcement learning algorithms are often trained in simulation environments [12]. Robotics tasks, such as dexterous in-hand manipulation, have been learnt through simulation [13]. Though this work does not consider reinforcement learning or robotics applications, they are a source of inspiration.

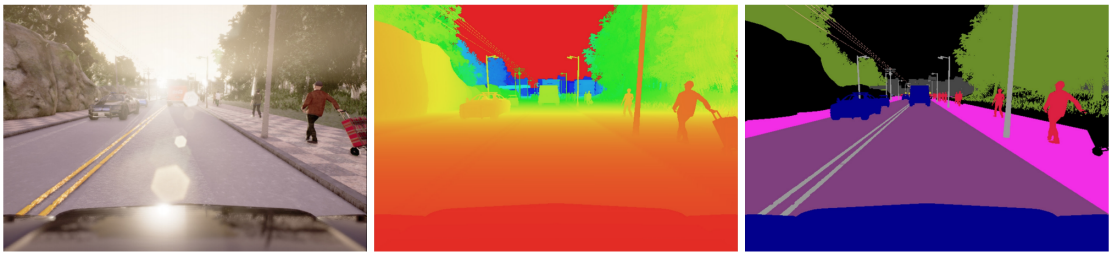


Figure 2. Example of RGB image, depth, and semantic segmentation from a scene in CARLA [8].

2.2 Unsupervised Domain Adaptation

Even slight departure from the neural network's training domain could hurt its performance [14]. This effect, called domain shift, has been observed on both simulation-to-real [14] and real-to-real dataset [15] shifts. A simple solution would be to fine-tune the network on samples from the new domain. It may be difficult to gather ground truth for the new domain, however. Often, unlabeled samples are easy to acquire. Unsupervised domain adaptation takes the approach of adapting a model to the target domain using unlabeled samples. A short overview of such methods follows.

2.2.1 Feature Alignment

One way to adapt a network to an unseen domain is to align the feature distributions extracted in the source and target domains using an additional loss. This can done by minimizing differences in higher order statistics of extracted features, i.e. moment matching [16–18]. Alternatively, an adversarial approach in which a discriminator tries

to classify the input as either from the source or target domain (shown in Figure 3). The objective is to maximize the domain confusion - if the features are similar across domains, the discriminator can't distinguish between them.

If the extracted features are similar in both the source and target domains, then the classifier using these features should intuitively perform similarly well. A limitation of such methods is that they only enforce the features from the target domain to look the same as source domain, possibly disregarding the semantic information necessary for performing the task.

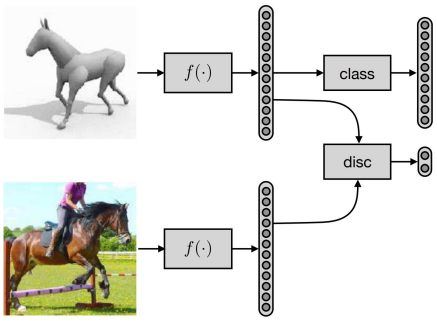


Figure 3. Visualization of adversarial feature-space alignment. Image adapted from Pinheiro, 2017 [19].

2.2.2 Image-to-Image Translation

Instead of making target domain features look like they are from the source domain, one could make source domain samples look like the target domain. This is possible by using Cycle-consistent Generative Adversarial Networks (CycleGANs) [20]. The method uses two sets of images, such as summer and winter, and learns to translate between them. Examples are shown in Figure 4.

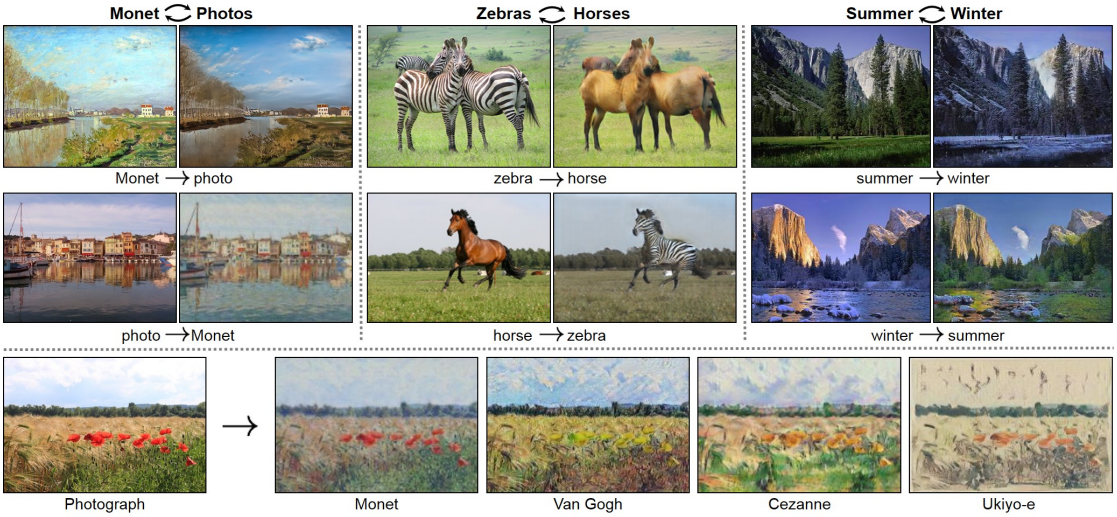


Figure 4. Image-to-image translation examples from CyleGAN [20].

The CycleGAN training scheme is shown in Figure 5. There are two mapping functions $G : X \rightarrow Y$ and $F : Y \rightarrow X$, and associated adversarial discriminators D_y and D_x for domains X and Y. For example X may be winter images and Y summer images. D_y encourages G to translate X into outputs indistinguishable from domain Y, and vice versa for D_x and F, as shown in Figure 5.a. While adversarial loss makes sure that the translated image looks like it belongs to the target domain, cycle-consistency loss makes sure it represents the same information as in the source image, as depicted in Figures 5.b and 5.c.

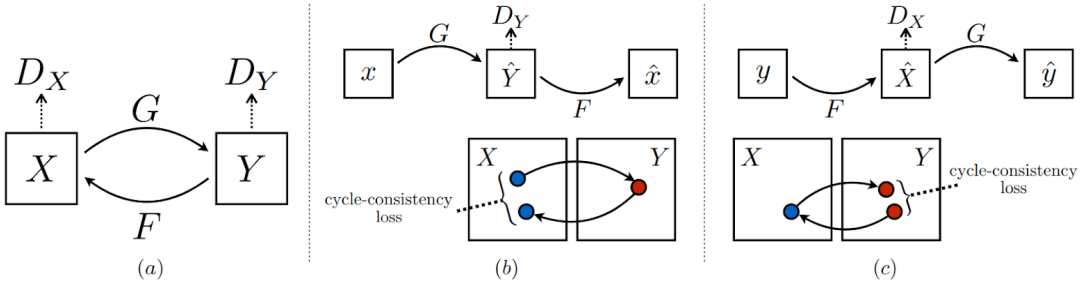


Figure 5. Training scheme of CyleGAN [20].

Cycle-Consistent Adversarial Domain Adaptation (CyCADA) [14] is an unsupervised domain adaptation method built upon CycleGAN, which competes with state-of-the-art to this day. Figure 6 depicts their proposed training scheme, where several ideas are combined:

- CycleGAN's adversarial (green *GAN loss*) and cycle-consistency (pink *Cycle loss*) losses.
- Feature matching using feature-level adversarial loss (orange *GAN loss*).
- Forcing semantic consistency in $source \rightarrow target$ translation with a feature level similarity loss (gray *Semantic Consistency loss*).
- Simultaneous optimization of target task (purple *Task loss*). This was omitted in experiments with semantic segmentation.

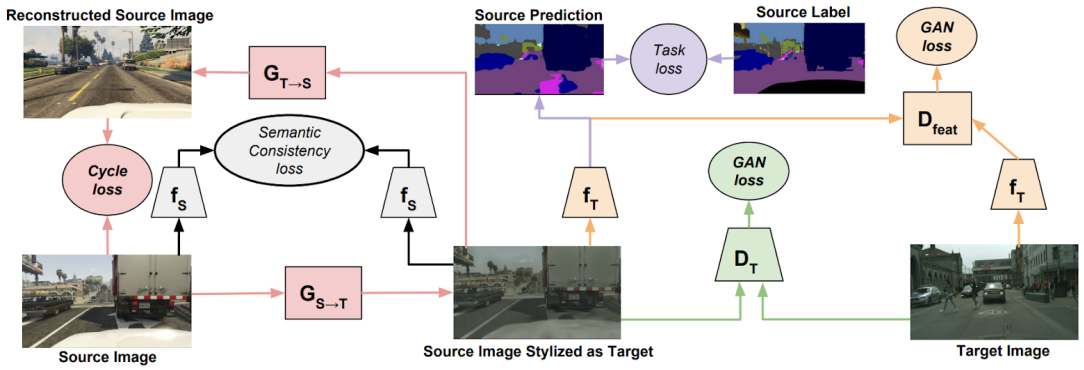


Figure 6. Training scheme of CyCADA [14].

This thesis considers the simplest form of image-to-image translation-based domain adaption: converting the simulation dataset to look like the real-world target dataset using CycleGAN. This was shown to be the most impactful part of CyCADA [14], so the thesis omits other parts, such as the semantic consistency loss, for simplicity.

2.3 Data Augmentation

Data augmentation is the generation of new samples by modifying or combining existing data. By covering a larger set of samples, the model should learn a more general representation. There is a variety of ways to modify an image while making use of its existing ground truth. Figure 7 gives a high-level overview of image augmentation methods. The described domain adaptation technique falls under "GAN Data Augmentation" from the data augmentation perspective. This thesis also experiments with the "Neural Style Transfer" approach, which is introduced after basic image manipulations.

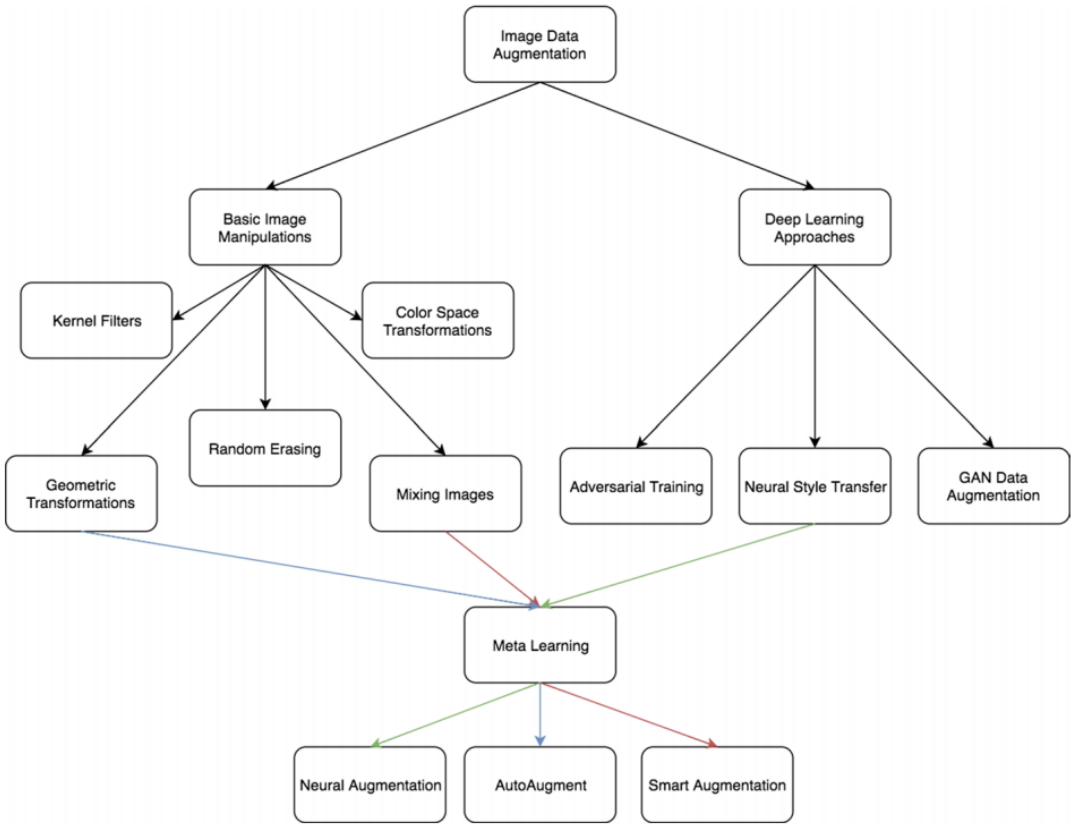


Figure 7. Taxonomy of image augmentation methods [21].

2.3.1 Basic Image Manipulations

The most common method of data augmentation is in the form of hand-coded manipulations. Figure 8 shows examples of such augmentations by the `imgaug` library [22]. Possible augmentations are include doing random left-right or up-down flips, rotation, cropping, erasing, color jitter, gaussian noise, blur, brightness, synthetic weather effects, and so on.

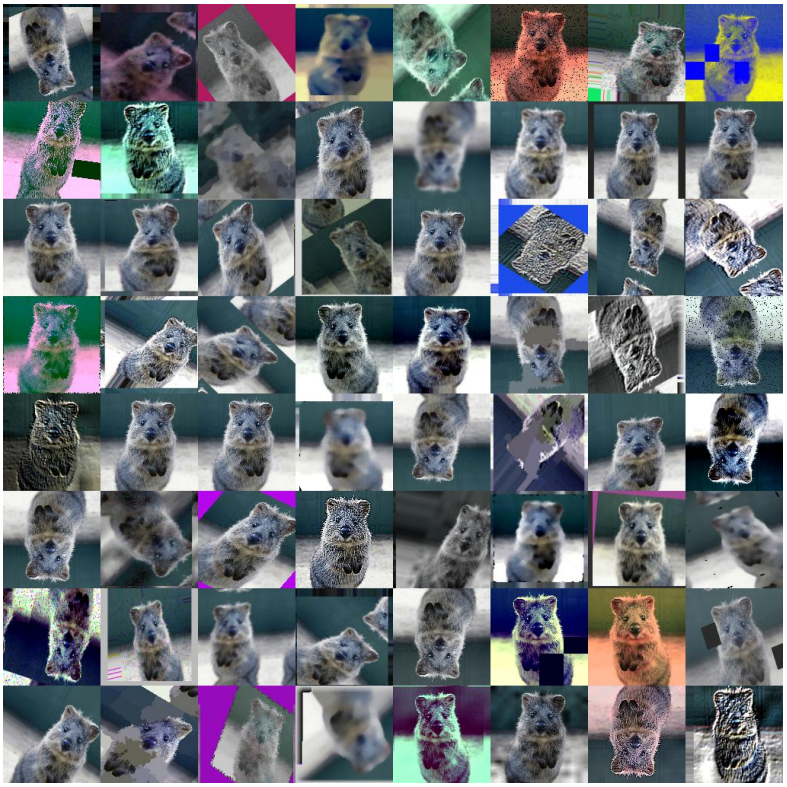


Figure 8. Image augmentations from the imgaug library [22].

With such variety, the question arises: which augmentations are appropriate, and how strong should they be? For example, in character recognition, left-right flips would not be appropriate, as letters are written in one direction. It can be difficult to put together a good combination of augmentations, as the search space is large. To solve this issue, several recent works have studied automatic search of augmentation combinations and parameters [23–25]. Currently, most augmentation search methods have been confined to classification tasks.

Instead using arbitrary augmentations for the sake of image diversity, another approach would be to look at the biases our model learns, and select augmentations that address them. In Section 5, this work considers style randomization and gives intuition why it is a form of augmentation particularly interesting for generalization from simulation to reality.

2.3.2 Style Randomization

Figure 9. Example of neural style transfer used to apply the style of a painting to an image [26].

Neural style transfer is a technique that blends the content of one image, and the style of another image together using neural networks [26]. An example is shown in Figure 9. Restyling images is a form of data augmentation which is especially interesting due to its ability to form complex non-uniform textures [6, 7].

Real-world images can be used to extract the target style, as experimented by Yue *et al.*, shown in Figure 10 [27]. Instead of taking styles from target images, the augmentation pipeline can be simplified by sampling random styles, as demonstrated by Jackson *et al.* [7]. The visual results of this method are shown in Figure 11. This thesis experiments with the latter method, style randomization, as applying styles from the real world does not seem to make simulation images more realistic.

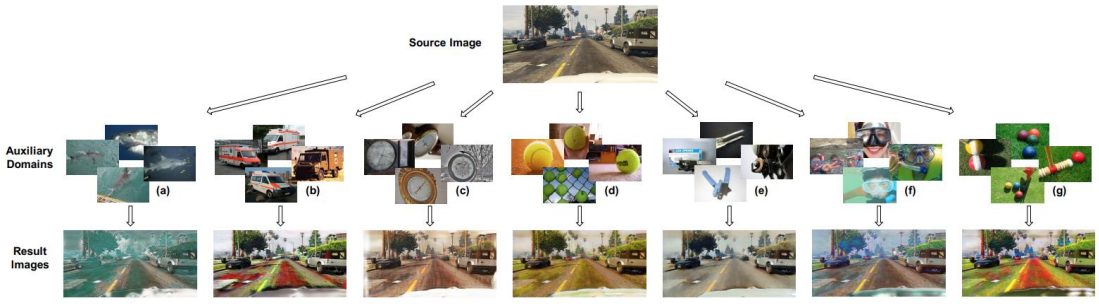


Figure 10. Simulation images with styles applied from ImageNet images [27].

Figure 11. Image of a cup with styles sampled from a random distribution [7].

2.3.3 Domain Randomization

In robotics, randomizing simulation parameters such as friction coefficients and environment textures has enabled policies trained in simulation to work on physical robots [13]. This method has the benefit that it is not targeted towards some selected environment, but optimizes for robustness on all environments.

Some recent works take inspiration from robotics and think of style randomization as a form of domain randomization [27]. Others look at it as data augmentation [7]. Both views are equally correct, and it is beneficial to consider different perspectives.

2.4 Multi-Task Learning

Research has shown that learning several tasks together can improve performance on all of them compared to training separately [28–30]. Simulation is perfect for multi-task learning, as ground truth for many different tasks can be generated, such as depth or surface normals. Such ground truth may be difficult or impossible in the real world. Furthermore, learning a set of vision tasks (shown in Figure 12) has been shown to be effective at enabling robots trained in simulation to generalize to the real world [31].

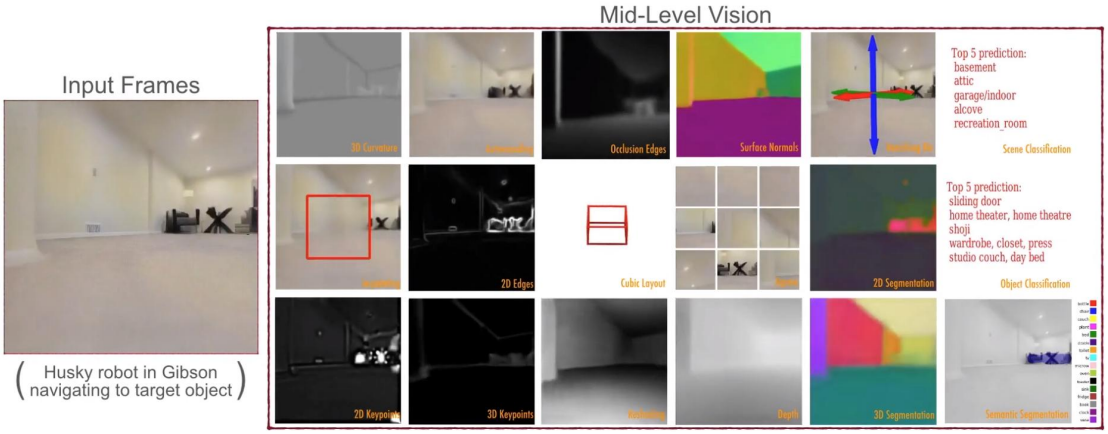


Figure 12. Multi-task learning of a variety of vision tasks for a robot [31]

.

2.5 Evaluation: Intersection over Union

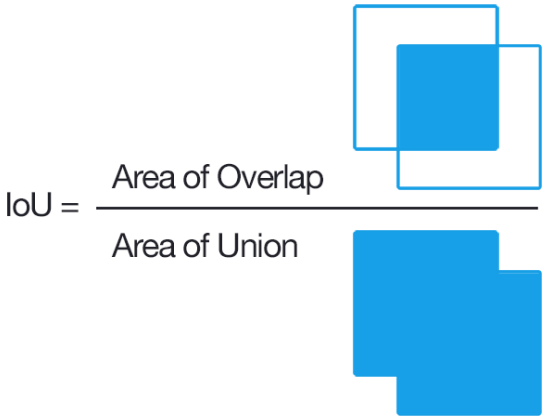


Figure 13. Intersection over union [32].

To quantitatively measure the performance of the trained models in our experiments, we use Mean Intersection over Union (MIoU), also known as Jaccard index. The intersection over union of a class in semantic segmentation in an image is the area overlap between the prediction and ground truth, divided by union of areas in the prediction and ground truth [32]. This is visualized in Figure 13. As this calculated separately for each class, we take the mean IoU of all classes to quantify the general quality of prediction.

It is one of the most common metrics to measure semantic segmentation performance. It has the quality that it is not dominated by large area predictions, such as roads or vegetation, and better reflects details like humans and lamp posts compared to alternative measures like pixel-wise accuracy.

3 Methods

3.1 Datasets

In order to study the generalization of semantic segmentation networks trained on simulation data, this work uses three environments:

1. A simulation environment that is closely modeled after the target real-world environment. The networks are only trained on labeled data from this simulation. Sample images with segmentation ground truth are shown in Figure 14. 5000 frames were extracted to form a dataset. The simulation environment was made by Mahir Gulzar as part of Milrem Nutikas UGV project, using the Unity engine and Airsim [11].

2. Target real-world environment with 460 images, manually labeled for testing. Example images are shown in Figure 15a. Referred to as Target-Set.

3. A collection of different real-world environments to test generalization beyond the target environment. Example images are shown in Figure 15b. A total of 190 images. Referred to as Gen-Set.

Generalizing to the Target-Set requires only the adaptation to the change of visual style from simulation to the real world, as content is otherwise the same. Generalizing to Gen-Set requires the model to learn inductive features, as it needs to extrapolate its knowledge to new environments.

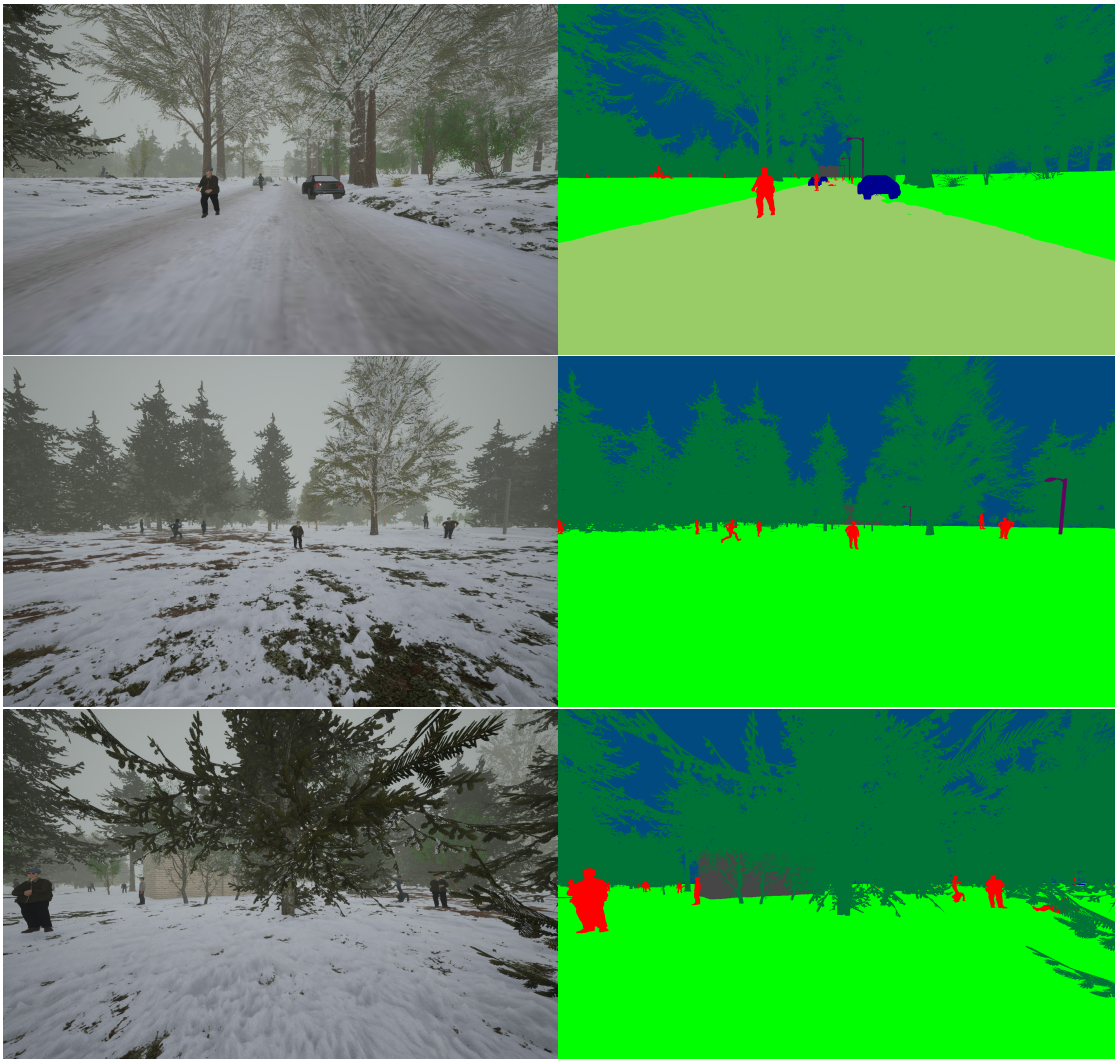
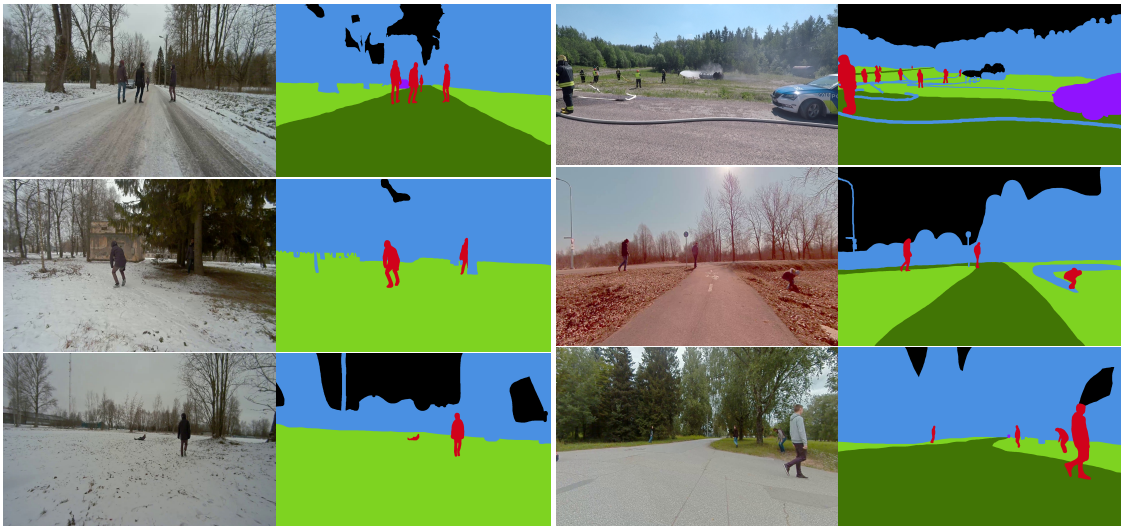


Figure 14. Examples from simulation-extracted semantic segmentation dataset. It is a snowy environment that is closely modeled after the real-world target environment (Target-Set). The left column shows the input RGB image and right column is the respective color-coded segmentation ground truth extracted from simulation. Examples of classes include: human (red), car (dark blue), grass (light-green), vegetation (dark-green).



(a) Target-Set (b) Gen-Set

Figure 15. Examples of the real world test datasets. (a) Target-Set: target environment, which the simulation was modeled after. (b) Gen-Set: real world generalization dataset, which includes images of different environments in addition to the target environment. The left column shows the input RGB image and right column is the respective color-coded segmentation ground truth. We use a simplified set of classes for real-world evaluation, to ease labeling. Examples of classes include: human (red), vehicles (purple), obstacles (e.g. trees and rocks; blue), drivable-fast (e.g. roads; dark green), drivable-slow (e.g. grass; light green)

The baseline performance achieved by training on simulation (none of the tested methods applied) and tested on the test sets is 0.2 MIoU on Target-Set, and 0.14 MIoU on Gen-Set. The baseline performance is included in all tables for comparison.

To measure the ”upper bound” of achievable performance on the test sets, a model is trained on real world training set (”Train on reality”), which is taken from the same environments of Target-Set and Gen-Set.

3.2 Model

This thesis uses the semantic segmentation architecture DeepLabV3 [33]. In particular, the implementation included in PyTorch's [34] vision library[1]. The main reason for this choice was that it is readily available in PyTorch.

[1] `https://pytorch.org/docs/stable/torchvision/models.html#torchvision.models.segmentation.deeplabv3_resnet50`

4 Unsupervised Domain Adaptation with CycleGAN

4.1 Introduction

This work experiments with CycleGAN [20], a image-to-image generative network, to make the simulation training set look like its real world counterpart.

4.2 Experiments

4.2.1 Method

Firstly, a CycleGAN model is trained to translate between simulation environment and target real-world environment. The whole simulation training set (5000 images) and real-world testing set (Target-Set, 460 images) are used for this. The official PyTorch implementation of CycleGAN[2] is used, with default hyperparameters, except `load_size=512` and `crop_size=420`.

The generated images look realistic from afar, which some exceptions. Objects with unpredictable poses and textures, such as cars and humans, were the most difficult for CycleGAN to generate. The ground truth for the original simulation images are also no longer very accurate. Examples are shown in Figure 16. The whole simulation dataset is transformed using the trained CycleGAN model.

[2] `https://github.com/junyanz/pytorch-CycleGAN-and-pix2pix`

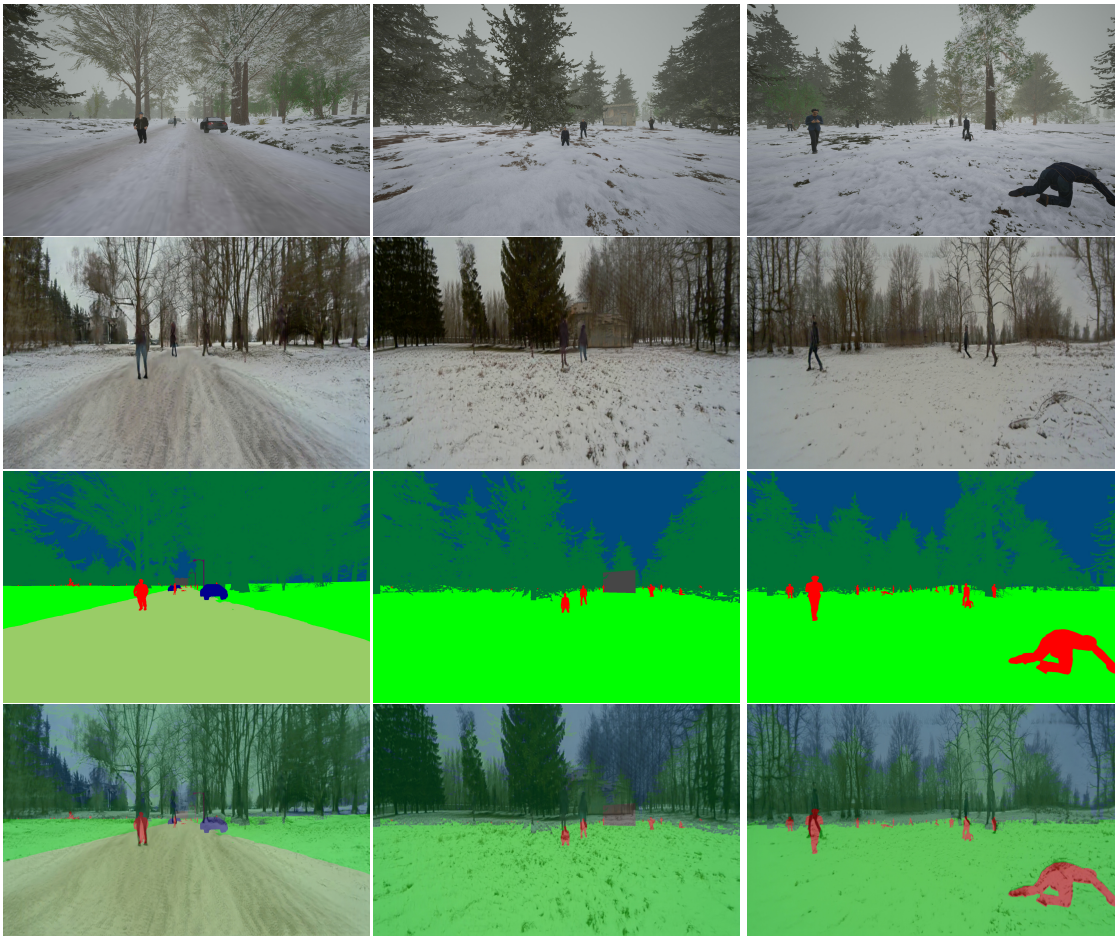


Figure 16. Examples of simulation images, what they look like after applying CycleGAN, and their semantic segmentation ground truth. Columns are 3 different samples. Rows are: 1) original, 2) the respective generated image, 3) semantic segmentation ground truth, and 4) generated image merged with ground truth. The generated images no longer perfectly match the ground truth, e.g. on rightmost sample, the crawling human is almost lost in the generated image, yet the network is expected to predict it is there.

4.2.2 Results

Table 1. Performance of domain adapted model on adaptation target set and generalization set compared to baseline and reality trained "upper bound" model. Measured in mean intersection over union

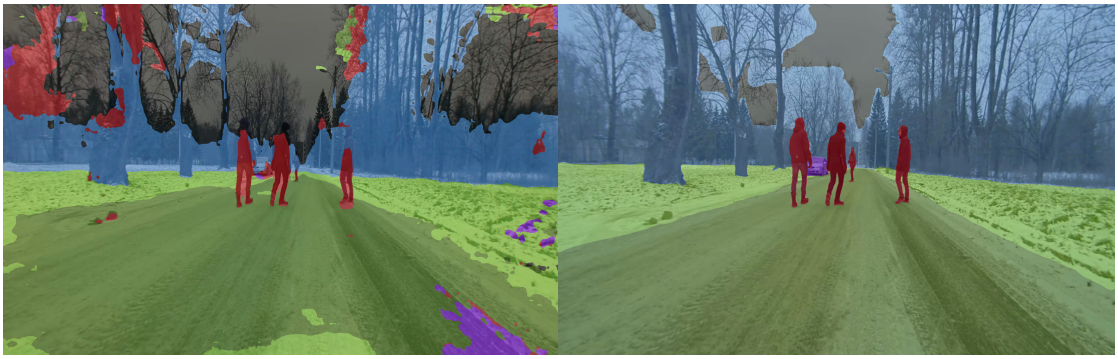
Method	MIoU Target-Set	MIoU Gen-Set
Baseline	0.2	0.14
Domain Adaptation	**0.42**	**0.19**
Train on reality (upper bound)	0.55	0.49

The semantic segmentation performance of a network trained on the unmodified simulation dataset and the domain adapted dataset in compared in Table 1. As expected, the model performance increases by a large margin on the dataset which it is adapted to. The generalization beyond the target dataset is poor, however. Sample predictions and ground truth are shown in Figure 17.

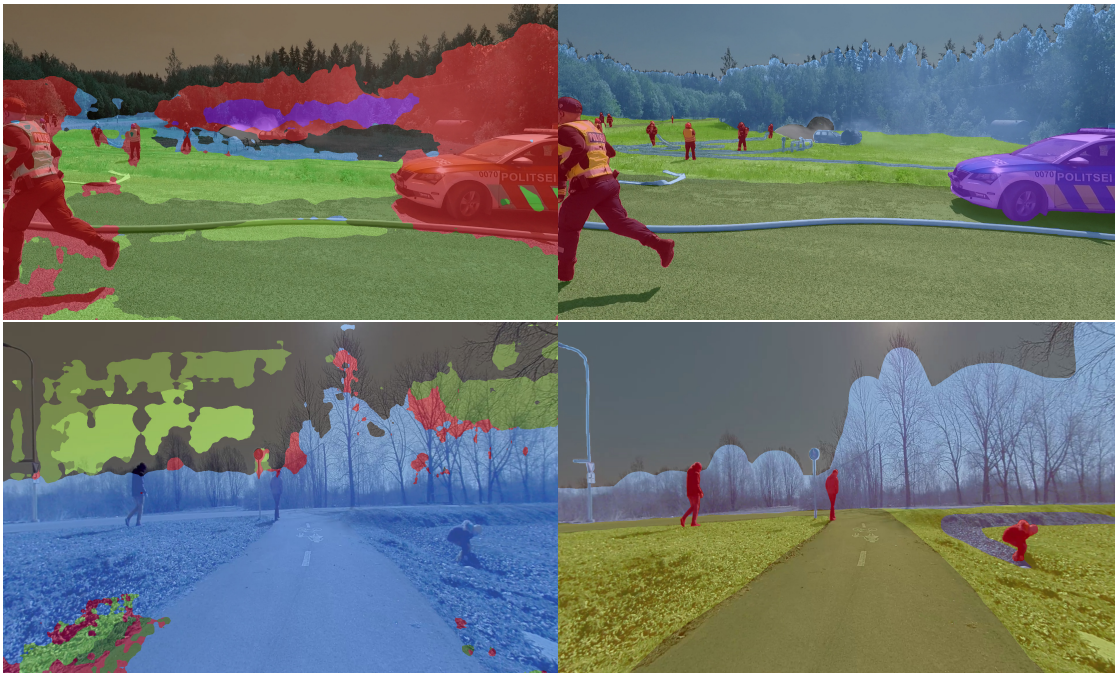
4.3 Limitations

Image-to-image translation for unsupervised domain adaptation has shown state-of-the-art performance in simulation-to-real generalization [14]. It has limitations, however, which motivate further work:

- Training image-to-image translation GANs is computationally expensive.
- The translation between domains may not be semantically accurate. This is shown in examples in Figure 16, where this effect is especially strong on humans. Perhaps the variety in human poses and appearance is overly challenging for the network to learn.
- Domain adaptation techniques require a target dataset. In the unsupervised case, labels are not needed, but adaptation is still done towards a target set of images, and generalization beyond them is not guaranteed.



(a) Target-Set



(b) Gen-Set

Figure 17. Example predictions by the domain adapted model (left) and ground truth (right). First sample is from Target-Set, and others from Gen-Set.

5 Style Randomization

5.1 Introduction

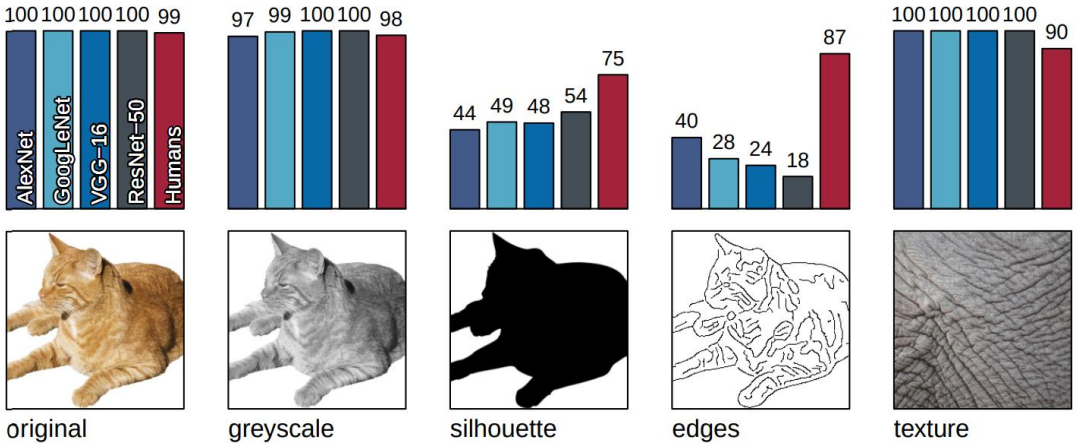


Figure 18. Accuracies and example stimuli for five different experiments without cue conflict [6]. Demonstrates ImageNet-trained neural networks' bias towards textures.

Geirhos *et al.* showed that convolutional neural networks tend to rely on textures instead of shape information (as shown in Figure 18), which should provide much stronger cues [6]. Furthermore, they demonstrated that by using style transfer as data augmentation, the trained networks were biased towards shapes and achieved better results on ImageNet. Though simpler augmentation methods exist, such as adding noise or shifting colors, style randomization has been shown to work better [6, 27].

The intuition that neural networks rely too much on texture is especially true in simulation. Due to the re-use of assets and textures, it is easy for the network to overfit such details and stop learning. Shapes of objects are more consistent across simulation and real world than textures, which motivates us to apply style randomization. While the data augmentation perspective says that the model should generalize better, the shape biasing perspective gives us a concrete explanation why.

5.2 Experiments

5.2.1 Method

A common way to do data augmentation via style transfer is to apply styles extracted from some source images [6, 27]. This thesis instead uses an implementation which samples random styles [7]. Their code is openly available[3]. This approach simplifies the training pipeline, as we have two hyperparameters to tune: augmentation strength and

[3] `https://github.com/philipjackson/style-augmentation`

probability. Augmentation strength $\alpha = 0.5$ is used, which is the default recommended by the authors [7]. We use augmentation probability $p = 0.5$, so half the images the model sees during training are the original simulation images, and the other half with a random style applied.

It should be noted that the augmentation can either be precomputed or done at run time. The first means storing style randomized images as a dataset. This takes hard-drive space, makes dataset management more difficult, and effectively limits the amount of augmentation one can do, but is more computationally efficient. We opted to do the augmentation at run time, which takes additional GPU power, and complicates multi-processed dataset loading. Our limited tests suggested that both methods results in equal model performance.

5.2.2 Results

The augmentations, shown in Figure 19, retain image semantics and the ground truth remains accurate. Furthermore, the augmented images are still easily understandable to humans, while being strong enough for augmentation purposes.

Results are shown in Table 2. The model trained with style randomization generalizes better than the domain-adapted model in the real-world target set (Target-Set), which was somewhat surprising. Furthermore, applying style randomization resulted in significantly better performance in the generalization test set (Gen-Set), compared to domain adaptation. Examples of the model's prediction area shown in Figure 20.

Table 2. Performance of model trained with style randomization compared to domain adaptation, baseline, and "upper bound" model trained on real world data. Measured in mean intersection over union

Method	MIoU Target-Set	MIoU Gen-Set
Baseline	0.2	0.14
Domain Adaptation	0.42	0.19
Style randomization	**0.50**	**0.34**
Train on reality + style randomization	0.54	0.44
Train on reality	0.55	0.49

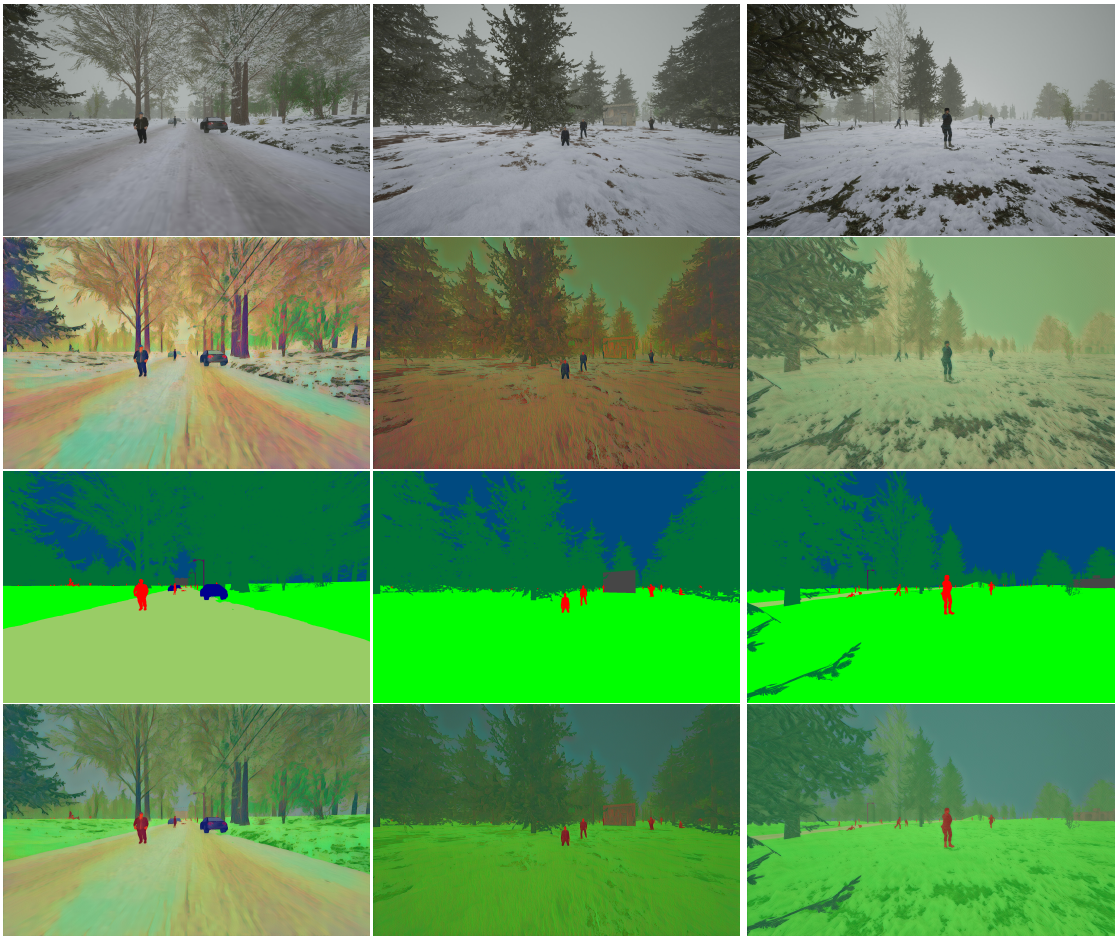


Figure 19. Examples of simulation images, what they look like after style randomization, and their semantic segmentation ground truth. Columns are 3 different samples. Rows are: 1) original, 2) the respective augmented image, 3) semantic segmentation ground truth, and 4) augmented image merged with ground truth.

Figure 20. Example predictions model trained with style randomization (left) and ground truth (right). First sample is from Target-Set, and others from Gen-Set.

5.3 Limitations

Style randomization, a neural data augmentation method for images for images, has shown to be an extremely effective, yet simple, way to generalize from simulation to the real world. There exist some limitations, however:

- Many (usually non-object) areas are often purely defined by their texture (e.g. roads, surface types). Strong enough perturbations to such textures may change their semantic meaning (e.g. making a dirt road look closer to asphalt).
- The augmentations made by style randomization can destroy some useful visual information. For example, shadows or consistent colors like green vegetation.

6 Multi-Task Learning with Depth

6.1 Introduction

Multi-Task Learning (MTL) is the simultaneous learning of several tasks with some shared representation. The efficacy of multi-task learning has long been known, and it is described that the training signals for the extra tasks serve as an inductive bias [28].

In Section 5, the intuition was introduced that by using style randomization, we bias the network to think that a pixel's class is determined by the shape that is formed in its surroundings. This is because changing up the textures via style randomization disproves the idea that the pixel class is defined by the local texture. Similarly, by adding auxiliary tasks, we limit the space of learnable representations, such that all the given tasks should be solvable with one general representation. This is to say, many possible representations are disproved, and only the more general ones remain. This can be considered as a form of regularization.

For several tasks to improve their shared representation, they should be related, such that they find similar features useful. What tasks should then be used for MTL? Zamir *et al.* studied how well different visual tasks are suited for transfer learning to other tasks, exploiting common structure between the tasks [35]. Standley *et al.* studied which tasks benefit from being learned jointly [29] by trying out different combinations. The tasks found to be most beneficial to be learned with semantic segmentation were surface normal prediction and depth estimation. Indeed, geometry is a defining property of objects, and networks should be able to extract it.

This intuition motivates the use of depth estimation as an auxiliary task. Perhaps by learning a representation which is able to predict the depth of each pixel, the model acquires some understanding of geometry. Geometry, being fairly consistent across simulation and reality, serves as an inductive bias. While surface normals are a more explicit representation of geometry, a sufficiently accurate depth image implicitly encodes surface normals as well. Depth is also available in most simulators [8, 9, 11].

6.2 Experiments

6.2.1 Method

For depth prediction, an additional head is added to the model. This means that segmentation and depth each have 3 independent layers after the shared backbone. Because the depth ground truth was strictly in the range [0, 255], we normalized it to [-1,1] and used the tanh activation to constrain predictions to that interval. We found that adding this constraint considerably improves prediction quality. Examples of depth ground truth are shown in Figure 21.

Figure 21. Examples of depth ground truth. Rows are 3 different samples. Columns are 1) the original image, and 2) its target depth ground truth. Each pixel in the depth ground truth is measured in meters from the camera in range 0–255. Brighter pixels are further away then dark pixels.

6.2.2 Results

The experiments showed that it is difficult to improve semantic segmentation performance using depth prediction as an auxiliary task. The results are listed in Table 3. Sample predictions are shown in Figure 22. The depth predictions can be used to project the camera image and segmentation to a point cloud, as seen in Figure 23. This allows better interpretability of the depth predictions. In summary, the findings are as follows:

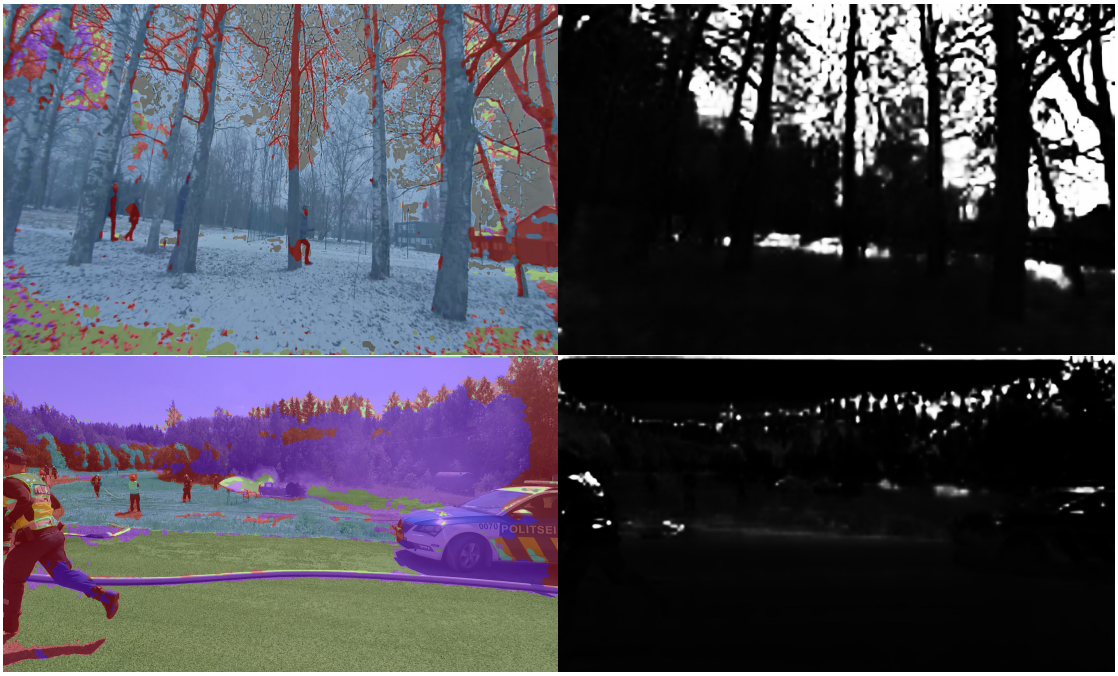
- Training is sensitive to the loss coefficients of the losses. In particular, the coeffi-

cients 1:0.15 for segmentation:depth had the best results and even slight changes to this ratio cost performance.

- Contrary to our hypothesis, adding the depth auxiliary loss did not increase generalization. Without using style randomization, the results were strictly worse than the baseline. Interestingly, by adding style randomization, the model performed about the same as without MTL. With enough tuning, combining the two methods produced the best results, but by a negligible margin.

Table 3. Comparison of different hyperparameter combinations. Segmentation loss coefficient is always 1. SR refers to using style randomization.

Method	depth coefficient	Batch size	MIoU Target-Set	MIoU Gen-Set
Baseline	0	11	0.2	0.14
Domain Adaptation	0	11	0.42	0.19
Style Randomization	0	11	**0.50**	0.34
Style Randomization	0	24	0.47	0.30
Baseline + depth	0.1	11	0.19	0.09
Baseline + depth	0.15	24	0.16	0.09
SR + depth	0.1	11	0.44	0.30
SR + depth	0.15	11	0.49	0.34
SR + depth	0.25	11	0.40	0.28
SR + depth	1.0	11	0.38	0.25
SR + depth	0.15	24	**0.50**	**0.36**
Train on reality	0	13	0.55	0.49



(a) Baseline + depth

(b) Style randomization + depth

Figure 22. Segmentation and depth predictions by **(a)** baseline + depth model, and **(b)** style randomized + depth prediction model.

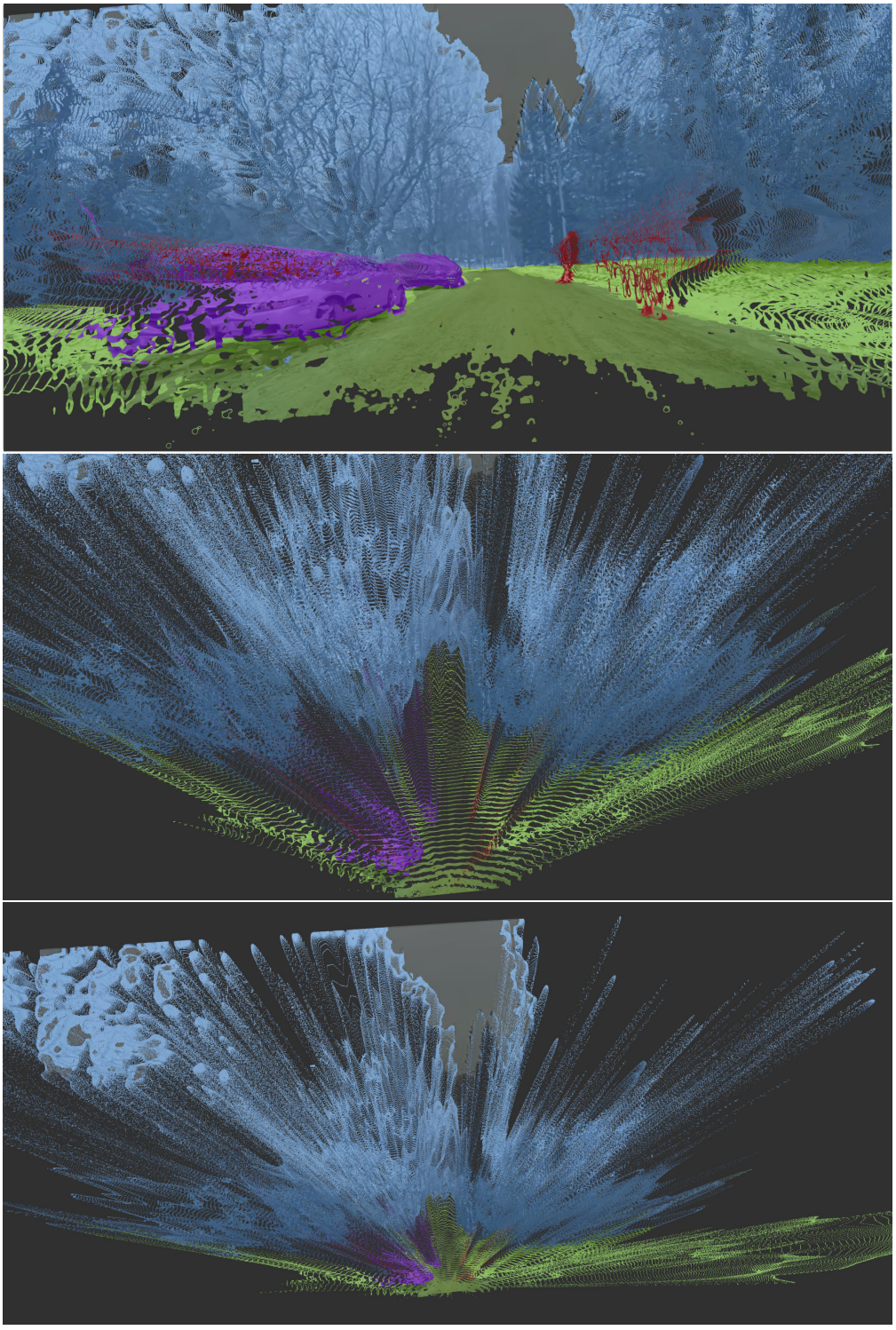


Figure 23. Pseudo point cloud made using the depth predictions. The RGB camera image and color-coded segmentation are projected onto the points. 3 different points of view are shown, from close to far.

6.3 Limitations

Though having potential, multi-task learning has several limitations in practice, as reflected in our results. To summarize:

- Multi-task learning can be hard to train. Several works have shown that a good set of loss weights or gradient manipulation may be needed to achieve a symbiotic relationship between the tasks [30, 36, 37].
- The training pipeline and model architecture have to be modified to accommodate auxiliary tasks.

7 Final Results

In this work, we considered three methods for simulation to real transfer: domain adaptation using CycleGAN, domain randomization via style randomization, and multi-task learning with depth prediction. We provide an ablation study in Table 4. The Target-Set column describes how well the model trained in a simulation environment generalizes to a similar real world environment (i.e. visual changes). The Gen-Set column describes how well the model generalizes to new real world environments (e.g. in summer, not winter), measuring how inductive is the learned model.

Table 4. Ablations of used methods. Best results of each column are in bold. Baseline is the model trained on simulation images without any of the tested methods applied. Final row is the "upper bound" model trained on real-world data from the test environments.

Method	MIoU Target-Set	MIoU Gen-Set	Samples
Baseline	0.2	0.14	
Domain Adaptation (DA)	0.42	0.19	Figure 17
Style Randomization (SR)	**0.50**	0.34	Figure 20
depth	0.19	0.09	Figure 22a
SR + depth	**0.50**	**0.36**	Figure 22b
DA + SR + depth	**0.50**	0.32	
Train on reality / upper bound	0.55	0.49	

8 Conclusion

This thesis considered three approaches for simulation to real world generalization of neural networks trained for semantic segmentation. The experiments were done with three datasets: the training dataset extracted from a simulation, a test dataset from a real world environment semantically similar to the simulation, and a generalization test dataset which is not similar to the simulation.

Firstly, this thesis experimented with unsupervised domain adaptation using CycleGAN. The generalization from the simulation to the target real environment improved significantly. The network failed to learn inductive representations, however, demonstrated by poor generalization to new environments.

Secondly, data augmentation using style randomization was tested. This approach was both easier to implement and enabled the model to generalize even to unseen environments in the real world.

Thirdly, multi-task learning was applied by using depth prediction as an auxiliary loss. Though the proposed intuition suggests that the network should generalize better, our experiments did not confirm this. Furthermore, this method showed to be highly sensitive to hyperparameters.

Finally, the tested methods were combined to see how well they complement each other. No benefit was observed when applying domain adaptation using CycleGAN and style randomization together. By combining depth estimation and style randomization, we were able to get the best results. The improvement was small, though, and probably not worth the effort required for successful multi-task learning.

Future work includes studying data augmentation using techniques that are more expressive than style transfer, such as generative networks similar to CycleGAN. Meta-learning (learning how to learn) is a related topic which was not covered in this thesis, but is interesting in the context of generalization. Multi-task learning was also not fully explored, as there are many tasks in addition to depth estimation to evaluate, as well as advanced methods to stabilize multi-task learning. Perhaps the multi-task learned models are more amenable to fine-tuning as well.

To summarize, this thesis found style randomization to be the best performing and simplest method to apply for simulation to real generalization. Being able do deploy simulation-trained models to the real world is highly valuable, as it allows training of algorithms that may be unfeasible or very expensive to do on real world data. As such, we will continue to see research into methods to advance sim-to-real generalization.

References

[1] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.

[2] Imagenet leaderboard. `https://paperswithcode.com/sota/image-classification-on-imagenet`. Online; Accessed: 29-04-2020.

[3] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. pages 1097–1105, 2012.

[4] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[5] Adrien Gaidon, Qiao Wang, Yohann Cabon, and Eleonora Vig. Virtual worlds as proxy for multi-object tracking analysis. *CoRR*, abs/1605.06457, 2016.

[6] Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A. Wichmann, and Wieland Brendel. Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. *CoRR*, abs/1811.12231, 2018.

[7] Philip T. G. Jackson, Amir Atapour Abarghouei, Stephen Bonner, Toby P. Breckon, and Boguslaw Obara. Style augmentation: Data augmentation via style randomization. *CoRR*, abs/1809.05375, 2018.

[8] Alexey Dosovitskiy, Germán Ros, Felipe Codevilla, Antonio López, and Vladlen Koltun. CARLA: an open urban driving simulator. *CoRR*, abs/1711.03938, 2017.

[9] Yohann Cabon, Naila Murray, and Martin Humenberger. Virtual kitti 2, 2020.

[10] Stephan R. Richter, Vibhav Vineet, Stefan Roth, and Vladlen Koltun. Playing for data: Ground truth from computer games. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *European Conference on Computer Vision (ECCV)*, volume 9906 of *LNCS*, pages 102–118. Springer International Publishing, 2016.

[11] Shital Shah, Debadeepta Dey, Chris Lovett, and Ashish Kapoor. Airsim: High-fidelity visual and physical simulation for autonomous vehicles. In *Field and Service Robotics*, 2017.

[12] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym. *CoRR*, abs/1606.01540, 2016.

[13] OpenAI, Marcin Andrychowicz, Bowen Baker, Maciek Chociej, Rafał Józefowicz, Bob McGrew, Jakub Pachocki, Arthur Petron, Matthias Plappert, Glenn Powell, Alex Ray, Jonas Schneider, Szymon Sidor, Josh Tobin, Peter Welinder, Lilian Weng, and Wojciech Zaremba. Learning dexterous in-hand manipulation. *CoRR*, 2018.

[14] Judy Hoffman, Eric Tzeng, Taesung Park, Jun-Yan Zhu, Phillip Isola, Kate Saenko, Alexei A. Efros, and Trevor Darrell. Cycada: Cycle-consistent adversarial domain adaptation. *CoRR*, abs/1711.03213, 2017.

[15] Judy Hoffman, Dequan Wang, Fisher Yu, and Trevor Darrell. Fcns in the wild: Pixel-level adversarial and constraint-based adaptation. *CoRR*, abs/1612.02649, 2016.

[16] Chao Chen, Zhihang Fu, Zhihong Chen, Sheng Jin, Zhaowei Cheng, Xinyu Jin, and Xian-Sheng Hua. Homm: Higher-order moment matching for unsupervised domain adaptation, 2019.

[17] Baochen Sun, Jiashi Feng, and Kate Saenko. Return of frustratingly easy domain adaptation. *CoRR*, abs/1511.05547, 2015.

[18] Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. *CoRR*, abs/1812.01754, 2018.

[19] Pedro Oliveira Pinheiro. Unsupervised domain adaptation with similarity learning. *CoRR*, abs/1711.08995, 2017.

[20] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. *CoRR*, abs/1703.10593, 2017.

[21] Connor Shorten and Taghi M. Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of Big Data*, 6(1):60, Jul 2019.

[22] Alexander B. Jung, Kentaro Wada, Jon Crall, Satoshi Tanaka, Jake Graving, Christoph Reinders, Sarthak Yadav, Joy Banerjee, Gábor Vecsei, Adam Kraft, Zheng Rui, Jirka Borovec, Christian Vallentin, Semen Zhydenko, Kilian Pfeiffer, Ben Cook, Ismael Fernández, François-Michel De Rainville, Chi-Hung Weng, Abner Ayala-Acevedo, Raphael Meudec, Matias Laporte, et al. imgaug. `https://github.com/aleju/imgaug`, 2020. Online; accessed 01-Feb-2020.

[23] Ekin Dogus Cubuk, Barret Zoph, Dandelion Mané, Vijay Vasudevan, and Quoc V. Le. Autoaugment: Learning augmentation policies from data. *CoRR*, abs/1805.09501, 2018.

[24] Ekin D. Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V. Le. Randaugment: Practical automated data augmentation with a reduced search space, 2019.

[25] Daniel Ho, Eric Liang, Ion Stoica, Pieter Abbeel, and Xi Chen. Population based augmentation: Efficient learning of augmentation policy schedules, 2019.

[26] L. A. Gatys, A. S. Ecker, and M. Bethge. Image style transfer using convolutional neural networks. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2414–2423, 2016.

[27] Xiangyu Yue, Yang Zhang, Sicheng Zhao, Alberto Sangiovanni-Vincentelli, Kurt Keutzer, and Boqing Gong. Domain randomization and pyramid consistency: Simulation-to-real generalization without accessing target domain data, 2019.

[28] Rich Caruana. Multitask learning. *Learning to Learn*, page 95–133, 1998.

[29] Trevor Standley, Amir Roshan Zamir, Dawn Chen, Leonidas J. Guibas, Jitendra Malik, and Silvio Savarese. Which tasks should be learned together in multi-task learning? *CoRR*, abs/1905.07553, 2019.

[30] Alex Kendall, Yarin Gal, and Roberto Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. *CoRR*, abs/1705.07115, 2017.

[31] Alexander Sax, Bradley Emi, Amir Roshan Zamir, Leonidas J. Guibas, Silvio Savarese, and Jitendra Malik. Mid-level visual representations improve generalization and sample efficiency for learning active tasks. *CoRR*, abs/1812.11971, 2018.

[32] Adrian Rosenbrok. Intersection over union (iou) for object detection. `http://www.pyimagesearch.com/2016/11/07/intersection-over-union-iou-for-object-detection`. Online; Accessed 13-05-2020.

[33] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *CoRR*, abs/1706.05587, 2017.

[34] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan

Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d' Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

[35] Amir Roshan Zamir, Alexander Sax, William B. Shen, Leonidas J. Guibas, Jitendra Malik, and Silvio Savarese. Taskonomy: Disentangling task transfer learning. *CoRR*, abs/1804.08328, 2018.

[36] Zhao Chen, Vijay Badrinarayanan, Chen-Yu Lee, and Andrew Rabinovich. Gradnorm: Gradient normalization for adaptive loss balancing in deep multitask networks. *CoRR*, abs/1711.02257, 2017.

[37] Tianhe Yu, Saurabh Kumar, Abhishek Gupta, Sergey Levine, Karol Hausman, and Chelsea Finn. Gradient surgery for multi-task learning, 2020.

Acknowledgements

This work has been supported by Milrem Robotics and the Archimedes Foundation as a part of the Nutikas UGV project (project number NSP153), and by StudyITin.ee.

I would like to thank my supervisor, Tambet Matiisen, for his thorough comments on this thesis, and Mahir Gulzar, for his work on the simulation and pseudo-pointcloud visualization.

Appendix

I. Licence

Non-exclusive licence to reproduce thesis and make thesis public

I, **Hannes Liik**,
*(*author's name)

1. herewith grant the University of Tartu a free permit (non-exclusive licence) to

   reproduce, for the purpose of preservation, including for adding to the DSpace digital archives until the expiry of the term of copyright,

   Sim-to-Real Generalization of Computer Vision with Domain Adaptation, Style Randomization, and Multi-Task Learning,
   *(*title of thesis)

   supervised by Tambet Matiisen.
   *(*supervisor's name)

2. I grant the University of Tartu a permit to make the work specified in p. 1 available to the public via the web environment of the University of Tartu, including via the DSpace digital archives, under the Creative Commons licence CC BY NC ND 3.0, which allows, by giving appropriate credit to the author, to reproduce, distribute the work and communicate it to the public, and prohibits the creation of derivative works and any commercial use of the work until the expiry of the term of copyright.

3. I am aware of the fact that the author retains the rights specified in p. 1 and 2.

4. I certify that granting the non-exclusive licence does not infringe other persons' intellectual property rights or rights arising from the personal data protection legislation.

Hannes Liik
15/05/2020