Scalable Learning for Expanding Robot Vision

Horia Porav

Mansfield College
University of Oxford

A thesis submitted for the degree of
Doctor of Philosophy

Michaelmas 2020

Abstract

Vision is hard, but the rewards for cracking the problems that stand between us and robust robot vision more than make up for the arduous journey. This work aims to guide the reader through some ideas and concepts that try to make robot vision better under difficult conditions. We tackle the shortcomings of robot vision from three different directions: domain adaptation, image de-noising and finally training data generation. However, the boundaries between these categories are fluid, allowing for complementary operation.

Throughout this work, we successfully demonstrate domain adaptation for localisation across any weather and appearance using generative adversarial networks (GANs), and subsequently extend this to an unsupervised setting that generalises to more than one task. Similarly, we show that convolutional neural networks are a suitable choice for image de-noising - in the context of de-raining and inverse perspective mapping - and present data and ground-truth collection strategies that make this possible. Finally, we tackle the scarcity of training data via data generation. We propose an improvement to cycle-consistency GANs, and a very simple method for collecting images affected by adherent droplets complete with a clear ground-truth. Scene synthesis and randomization techniques are used for improved semantic segmentation models, finishing with an investigation of a collision mitigation system that leverages data created using a simulator.

The techniques described lead to significant gains under difficult conditions such as low illumination, large changes in appearance (domain shift) and noisy data, as well as allowing machine learning models to be trained using data that would normally be very hard, expensive or impossible to obtain.

Scalable Learning for Expanding Robot Vision



Horia Porav

Mansfield College

University of Oxford

A thesis submitted for the degree of

Doctor of Philosophy

Michaelmas 2020

Scalable Learning for Expanding Robot Vision

Candidate: Horia Porav
Supervisor: Professor Paul Newman
Examiners: Professor Philip Torr & Professor Cyrill Stachniss
Date of Examination: 10th May 2021

Mobile Robotics Group
Oxford Robotics Institute
Department of Engineering Science
University of Oxford

Acknowledgements

Personal

First of all, I would like to give my deepest regards to my supervisor and mentor Professor Paul Newman, for giving me the amazing opportunity to learn from and collaborate with him, and for giving me full autonomy to pursue any interest I had. One major idea instilled in me by working with Paul and the Oxford Robotics Institute (ORI) was that any piece of work - be it research, writing software or more admin-esque tasks - should be tackled in a way that easily allows extension and inclusion in other people's work. By doing so, it became incredibly easy to collaborate with colleagues and industry players alike, leading to co-authorships, patents, software licenses and even real-life deployment of some of our work. Oh, and Australia - thank you so much for that!

Secondly, Tom Bruls. To all our whiteboard sessions, and all the great ideas we had together. It was truly great having you as my colleague. May your jars of mayonnaise never spoil.

Thirdly, I would like to thank my wife Valentina Musat for all her support with developing ideas, writing papers, sketching pretty diagrams, talking me out of bad ideas and for trying to bake some sense of discipline into me - *There is no way to express how much I love you.*

To my parents - thank you for your unabated support all these years - Vă iubesc mult de tot!

I would also like to thank everyone else at the ORI for their open hearts and arms. You are all a great bunch, but I do have the suspicion that one of you may be a vampire.

Finally, I would also like to thank Peter Corke, Jurgen Leitner, Niko Sünderhauf and everyone else at the Australian Centre for Robotic Vision QUT for accepting me as one of their own during my exchange there.

Abstract

Vision is hard, but the rewards for cracking the problems that stand between us and robust robot vision more than make up for the arduous journey. This work aims to guide the reader through some ideas and concepts that try to make robot vision better under difficult conditions. We tackle the shortcomings of robot vision from three different directions: domain adaptation, image de-noising and finally training data generation. However, the boundaries between these categories are fluid, allowing for complementary operation.

Throughout this work, we successfully demonstrate domain adaptation for localisation across any weather and appearance using generative adversarial networks (GANs), and subsequently extend this to an unsupervised setting that generalises to more than one task. Similarly, we show that convolutional neural networks are a suitable choice for image de-noising - in the context of de-raining and inverse perspective mapping - and present data and ground-truth collection strategies that make this possible. Finally, we tackle the scarcity of training data via data generation. We propose an improvement to cycle-consistency GANs, and a very simple method for collecting images affected by adherent droplets complete with a clear ground-truth. Scene synthesis and randomization techniques are used for improved semantic segmentation models, finishing with an investigation of a collision mitigation system that leverages data created using a simulator.

The techniques described lead to significant gains under difficult conditions such as low illumination, large changes in appearance (domain shift) and noisy data, as well as allowing machine learning models to be trained using data that would normally be very hard, expensive or impossible to obtain.

Contents

List of Figures **xi**

List of Abbreviations **xiii**

1 Introduction **1**
1.1 Motivation . . . 1
1.2 Contributions . . . 6
1.2.1 Overall research contribution . . . 6
1.2.2 Domain adaptation . . . 6
1.2.3 Image de-noising . . . 7
1.2.4 Data generation . . . 8
1.3 Publications to date . . . 9
1.4 Roadmap . . . 11

2 A brief overview of relevant techniques **17**
2.1 Introduction . . . 17
2.2 Visual localisation . . . 18
2.2.1 Map-based representations . . . 19
2.2.2 Multi experience navigation . . . 20
2.3 Convolutional neural networks . . . 20
2.3.1 Intro to convolutions . . . 20
2.3.2 CNNs . . . 23
2.4 Generative adversarial networks . . . 23
2.4.1 Vanilla GAN . . . 25
2.4.2 Conditional GAN . . . 26
2.4.3 Cycle-consistency GAN . . . 27
2.4.4 GAN training tricks . . . 28

3 Domain adaptation **33**
3.1 Adversarial Training for Adverse Conditions . . . 34
3.2 Don't Worry About the Weather . . . 44

4 Image de-noising **55**
4.1 I Can See Clearly Now: Image Restoration via De-Raining 56
4.2 The Right (Angled) Perspective 65

5 Data generation **75**
5.1 Rainy screens: Collecting rainy datasets, indoors 76
5.2 Reducing steganography in cycle-consistency GAN architectures . . 84
5.3 Generating All the Roads to Rome 91
5.4 Imminent Collision Mitigation 101

6 Discussion **111**
6.1 Domain adaptation 111
6.2 Image de-noising 113
6.3 Data generation 114
6.4 Overall conclusion 117

References **119**

List of Figures

1.1 An example of both domain adaptation through style transfer **and** synthetic data generation . . . 3
1.2 Domain adaptation through image-to-image translation optimized to maximize performance on a downstream task . . . 4
1.3 An example of image restoration by removing the effect of adherent rain droplets . . . 4
1.4 An example of synthetic adherent rain droplets . . . 5
1.5 An example of data multiplication by synthesizing scenes from semantic labels . . . 5
1.6 Modern learned localisation techniques . . . 12

2.1 Convolution with no padding . . . 21
2.2 Convolution with padding . . . 22
2.3 Max pooling . . . 22
2.4 GAN setup . . . 24
2.5 Vanilla GAN architecture . . . 25
2.6 Early conditional GAN architecture . . . 26
2.7 Late conditional GAN architecture . . . 27
2.8 CycleGAN architecture . . . 29

List of Abbreviations

SLAM . . . . . Simultaneous Localisation and Mapping
GAN . . . . . Generative Adversarial Network
6DOF . . . . . 6-Degree Of Freedom
BDD . . . . . Berkeley DeepDrive
IPM . . . . . Inverse Perspective Mapping
mIOU . . . . . mean Intersection Over Union
RGB . . . . . Red Green Blue
CNN . . . . . Convolutional Neural Network
SIFT . . . . . Scale Invariant Feature Transform
SURF . . . . . Speeded Up Robust Features
BRIEF . . . . Binary Robust Independent Elementary Features
FAST . . . . . Features from Accelerated Segment Test
RANSAC . . . RANdom SAmple Consensus
UTM . . . . . Universal Transverse Mercator
2D 2-Dimensional
3D 3-Dimensional
ReLU . . . . . Rectified Linear Unit
ADAS . . . . . Advanced Driver Assistance Systems
GPU . . . . . Graphics Processing Unit
VAE . . . . . Variational AutoEncoder
DDPG . . . . . Deep Deterministic Policy Gradient
TTC . . . . . Time To Collision
RNN . . . . . Recurrent Neural Network
FID . . . . . Frechet Inception Distance
MUNIT . . . . Multimodal UNsupervised Image-to-image Translation
DRIT . . . . . Disentangled Representation for Image-to-image Translation

Robotum non sum.

I'm not a robot.

— Some robot.

1 Introduction

Contents

1.1 Motivation 1
1.2 Contributions 6
1.2.1 Overall research contribution 6
1.2.2 Domain adaptation 6
1.2.3 Image de-noising 7
1.2.4 Data generation 8
1.3 Publications to date 9
1.4 Roadmap 11

1.1 Motivation

This thesis presents a number of techniques which improve the state of the art of computer vision in conditions which are traditionally considered challenging.

The author's motivation for the work presented here is rooted in the assumption that it is often not computation that limits us, but suitable data to compute on. Hence our focus is in providing techniques which address what appears to be an insatiable thirst for data, while also reducing the costs associated with obtaining it. These techniques manifest themselves as data transformation, expansion, augmentation and synthesis. They each have their advantages and disadvantages, and thus may be used simultaneously and in a complementary way, while also combining

offline and online operation. Such techniques have proven useful in industry as well, with significant increases in performance brought by the use of Generative Adversarial Network (GAN) [1] based data transformation and generation.

One of the central themes of the author's research is that of visual localisation under any environmental conditions, day, night, winter, summer etc. This generally involves being able to match the current camera image to some image stored in a database (or a map), so that the robot can understand its topometric or precise 6DOF location in the world. A very common scenario is that the image stored in the map has a very different appearance than what the robot is currently seeing. This makes it very hard to perform matching and localisation.

So far, some attempts to alleviate this problem have included learning bespoke feature descriptors [2, 3] or directly using CNN features as descriptors [4], recording and storing maps of the same place under many different conditions [5], or using end-to-end monolithic deep learning approaches to directly output a pose [6].

Instead, the central proposal in this work revolves around transforming the camera images in a way that is meaningful and helpful for the tasks that are being performed, such as localisation or semantic segmentation. These challenges are attacked from three different directions:

- **domain adaptation**, which involves modifying the appearance or distribution of the image in such a way as to maximize performance in a target domain. One prominent example of this is style transfer [7, 8], e.g. transforming day-time images into night-time images and vice-versa, as shown in Figure 1.1. Another example is that of adapting images to a representation that maximizes the performance of downstream models [9], adeptly dubbed a *canonical* representation, as in Figure 1.2

- **image de-noising**, which involves a battery of image restoration techniques. An example of this is the removal of artifacts caused by adherent rain drops [10–13], as shown in Figure 1.3

- **data generation**, where new, synthetic data is generated and used in training downstream models. Examples of this include adding synthetic rain droplets to datasets [14], as shown in Figure 1.4, manipulating weather and appearance [15, 16], as in Figure 1.1, and synthesizing scenes from semantic labels [17–19], as shown in Figure 1.5



Figure 1.1: An example of both domain adaptation through style transfer **and** synthetic data generation. The images have been transformed using a cycle-consistency GAN trained to explicitly improve feature matching. In the context of domain adaptation, such a technique can be used as a real-time pre-processor of input images, as it was done in [15]. Alternatively, the resulting images can be used as training data, in conjunction with existing or pseudo ground-truth, as used in [9]

.

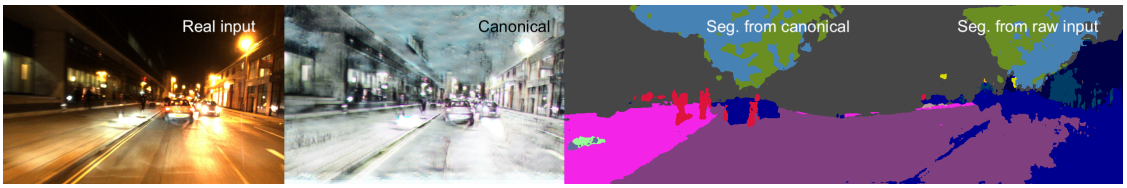


Figure 1.2: An alternative example of domain adaptation through image-to-image translation where the appearance of the translated images is not explicitly or implicitly conditioned on a style, but optimized directly to maximize performance on a downstream task. In this case, we train a light-weight image-to-image translation network (an adapter) to transform incoming images such that the performance of an off-the-shelf semantic segmentation model is maximized. From left to right, the camera image, the transformed image, the resulting segmentation on the transformed image, and the resulting segmentation on the initial, raw camera image. The performance on transformed images, while not perfect, is significantly improved, with no extra labelling effort.

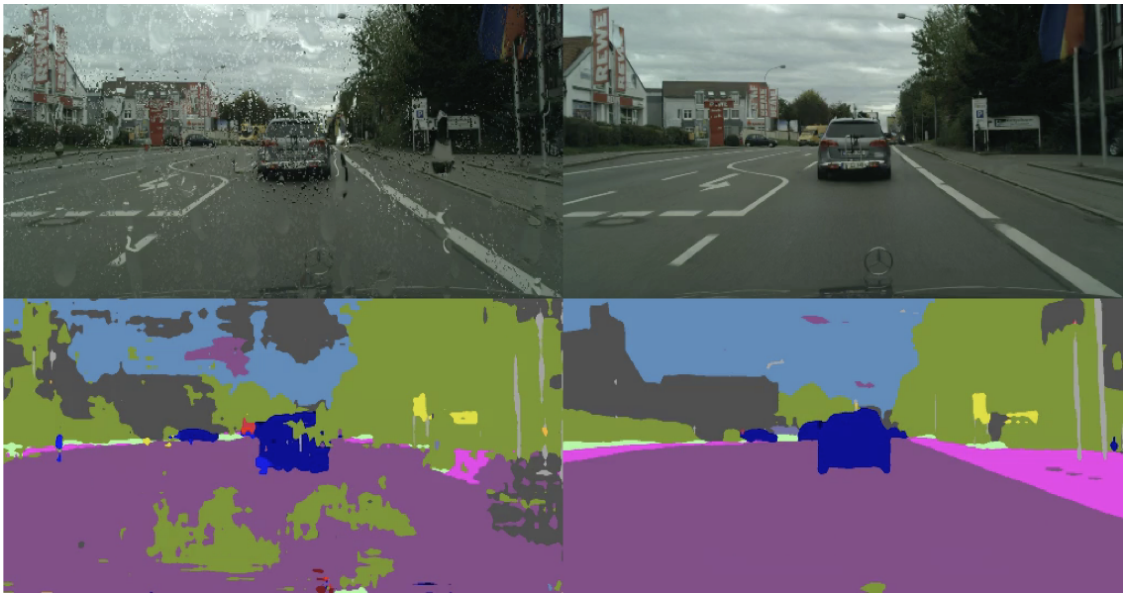


Figure 1.3: An example of image restoration by removing the effect of adherent rain droplets. Some tasks, such as semantic segmentation, are severely affected by adherent droplets (more examples of this in chapters 4 and 5, [13, 14]). Removing the effect of adherent droplets restores the performance of down stream tasks. However, training a model to restore such images requires large amounts of data with associated clear ground-truth images. This work presents effective methods of creating this training data and examples of models that are suitable for cleaning the affected images. The left column shows an input image affected by adherent droplets and the resulting semantic segmentation using an off-the-shelf segmentation model. The right column shows the same image cleaned by a de-raining model, and the resulting semantic segmentation using the same off-the-shelf model.



Figure 1.4: An example of synthetic adherent rain droplets. This works presents several methods of creating datasets of images affected by adherent water droplets, including vehicle-mounted rigs, workbench setups and, pictured here, computer-generated droplets using GPU shaders.



Figure 1.5: An example of data multiplication by synthesizing scenes from semantic labels. Full or partial scene synthesis via inpainting [17, 18] can be used to augment and multiply data, starting from an initial dataset. Here, we demonstrate road marker synthesis [19], and show that significant improvements in road marker segmentation can be achieved when fine-tuning a segmentation model with synthesized data extracted from an initial dataset, compared to the baseline model trained only on the initial dataset. There is no extra labelling involved, and all marker instances are extracted automatically. From top to bottom, the first row shows the initial images, the second row shows the predicted semantic segmentation maps (or labels, if available). The third row shows the manipulated semantic maps, where markers have been added, swapped or removed. The last row shows our hybrid synthesized images, obtained by blending parts of the original images with parts from fully-synthesized images.

1.2 Contributions

1.2.1 Overall research contribution

This thesis investigates new techniques and the applicability of existing techniques for improving the performance of vision-based tasks, such as semantic segmentation and visual localisation, under non-ideal and/or extreme weather and illumination conditions. As stated in Section 1.1, the individual contributions made through the work on which this thesis is based are classified into three categories: domain adaptation, image denoising and data generation. As it will be seen later throughout this thesis, these category boundaries are not fixed, with some concepts belonging to more than one category.

1.2.2 Domain adaptation

In the work that falls under this category, we investigate domain adaptation via image-to-image translation in two distinct but related cases. In the first case, we impose a specific style or distribution for the target translation, e.g. transforming an image from day time to night time appearance. In the second case, the style or distribution is optimized to maximize the performance of the resulting images on other tasks.

In chapter 3, we begin by presenting a method of improving visual place recognition and metric localisation under very strong appearance change, by learning an invertible generator that can transform the conditions of images, e.g. from day to night, summer to winter etc. This image transforming filter is explicitly designed to aid and abet feature-matching using a new loss based on SURF [20] detector and dense descriptor maps. A network is trained to output synthetic images optimised for feature matching given only an input RGB image, and these generated images are used to localize the robot against a previously built map using traditional sparse matching approaches. The results are benchmarked using multiple traversals of the Oxford RobotCar Dataset [21] over a year-long period, using one traversal as a map and the others to localise. We show that this method significantly improves place

recognition and localisation under changing and adverse conditions, while reducing the number of mapping runs needed to successfully achieve reliable localisation.

The chapter continues with a domain adaptation approach that transforms input images with arbitrary appearance or conditions into a normalized, "canonical" representation. Modern models that perform system-critical tasks such as segmentation and localization exhibit good performance and robustness under ideal conditions (i.e. daytime, overcast) but performance degrades quickly and often catastrophically when input conditions change. This method employs a domain adaptation system that uses light-weight input adapters to pre-processes input images, irrespective of their appearance, in a way that makes them compatible with off-the-shelf computer vision tasks that are trained only on inputs with ideal conditions. No fine-tuning is performed on the off-the-shelf models, and the system is capable of incrementally training new input adapters in a self-supervised fashion, using the computer vision tasks as supervisors, when the input domain differs significantly from previously seen domains. We report large improvements in semantic segmentation and topological localization performance on two popular datasets, RobotCar and BDD [22].

1.2.3 Image de-noising

The first part of chapter 4 presents a method for improving segmentation tasks on images affected by adherent rain drops and streaks. For this task, a novel stereo dataset was introduced, recorded using a system that allows one lens to be affected by real water droplets while keeping the other lens clear. A de-noising generator was trained using this dataset, demonstrating that it is effective at removing the effect of real water droplets, in the context of image reconstruction and road marking segmentation. To further test the de-noising approach, a method of adding computer-generated adherent water droplets and streaks to any images was implemented and used as a proxy to demonstrate the effectiveness of the de-raining model in the context of general semantic segmentation. The results were benchmarked using the CamVid [23] road marking segmentation dataset, Cityscapes [24] semantic segmentation datasets and the newly-created real-rain

dataset, showing significant improvement on all tasks. Additionally, the novel stereo dataset contains important aspects of synthetic data generation.

The second part of this chapter presents an adversarial learning approach for generating significantly improved Inverse Perspective Mapped (IPM) images from a single camera image in real time. The generated bird's-eye-view images contain sharper features (e.g. road markings) and a more homogeneous illumination, while (dynamic) objects are automatically removed from the scene, thus revealing the underlying road layout in an improved fashion. The framework is benchmarked using real-world data from the Oxford RobotCar Dataset, showing that scene understanding tasks directly benefit from the boosted IPM approach. This approach also contains important aspects of domain adaptation.

1.2.4 Data generation

Chapter 5 presents techniques related to (training) data generation, starting with an extension of the previous image de-raining work done in Section 4.1. Acquisition of data with adverse conditions in robotics is a cumbersome task due to the difficulty in guaranteeing proper ground truth and synchronising with desired weather conditions. A simple method is presented – recording a high resolution screen with a camera – that can generate diverse rainy images from existing clear ground-truth images. It is domain- and source-agnostic, simple and scales up. The setup makes it easier to leverage the diversity of existing datasets with auxiliary task ground-truth data, such as semantic segmentation, object positions etc. Rainy images with real adherent droplets and rain streaks are generated, based on Cityscapes and BDD, and a de-raining model is trained. Quantitative results are presented for image reconstruction and semantic segmentation, and qualitative results for an out-of-sample domain, showing that models trained with this data generalize well.

The chapter continues with a simple method of improving the suitability of data generated using cycle-consistency GANs in the context of day-to-night domain adaptation. While CycleGANs [8] produce visually pleasing outputs, they also encode hidden (steganographic) information about the source domain in the

generated images, which makes them less suitable as training data generators. We reduce the amount of steganographic information hidden in the generated images by introducing an end-to-end differentiable image de-noiser in between the two generators. The role of the de-noiser is to strip away the high frequency, low amplitude encoded information, making it harder for the generators to hide information that is invisible to the discriminator. The suitability of data generated using our simple method is benchmarked in the context of simple domain adaptation for semantic segmentation, comparing with standard CycleGAN [8], MUNIT [25] and DRIT [26], showing that the proposed method yields better performance.

Thirdly, the chapter presents a method for automatically generating new training pairs for road marking segmentation models. Principles of domain randomization are applied to the road layout and new images are synthesized from altered semantic labels. This work demonstrates that training on these synthetic pairs improves mean Intersection-over-Union (mIoU) for the segmentation of rare road marking classes during real-world deployment in complex urban environments, while performance for other classes is retained. The method can easily be scaled to all domains and conditions to generate large-scale road marking datasets, while avoiding manual labelling effort.

Finally, the chapter concludes with an investigation into the advantages of using a unified domain (semantic segmentation) as input to an autonomous vehicle driving task specialized in imminent collision avoidance, alongside the viability of formulating the objective of the training procedure using objective loss functions based on established delta-v models of injury severity.

1.3 Publications to date

- **Horia Porav**, Will Maddern and Paul Newman, "*Adversarial training for adverse conditions: Robust metric localisation using appearance transfer*", 2018 IEEE International Conference on Robotics and Automation (ICRA2018) [15]

- **Horia Porav**, Tom Bruls and Paul Newman, "*Don't Worry About the Weather: Unsupervised Condition-Dependent Domain Adaptation*", 2019 IEEE Intelligent Transportation Systems Conference (ITSC2019) [9]

- **Horia Porav**, Tom Bruls and Paul Newman, "*I Can See Clearly Now: Image Restoration via De-Raining*", 2019 International Conference on Robotics and Automation (ICRA2019) [13]

- Tom Bruls*, **Horia Porav***, Lars Kunze and Paul Newman, "*The Right (Angled) Perspective: Improving the Understanding of Road Scenes using Boosted Inverse Perspective Mapping*", 2019 IEEE Intelligent Vehicles Symposium (IV2019) (* equal contribution) [27]

- **Horia Porav**, Valentina-Nicoleta Musat, Tom Bruls and Paul Newman, "*Rainy screens: Collecting rainy datasets, indoors*", Arxiv preprint 2020 [14]

- **Horia Porav**, Valentina-Nicoleta Musat and Paul Newman, "*Reducing Steganography In Cycle-consistency GANs*", 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition - Vision for all Seasons Workshop (CVPR2019-V4AS) [28]

- Tom Bruls, **Horia Porav**, Lars Kunze and Paul Newman, "*Generating All the Roads to Rome: Road Layout Randomization for Improved Road Marking Segmentation*", 2019 IEEE Intelligent Transportation Systems Conference (ITSC2019) [19]

- **Horia Porav** and Paul Newman, "*Imminent Collision Mitigation with Reinforcement Learning and Vision*", 2018 IEEE Intelligent Transportation Systems Conference (ITSC2018) [29]

1.4 Roadmap

The rest of the thesis introduces key concepts that make this work possible in Chapter 2, followed by domain adaptation techniques in Chapter 3, image denoising approaches in Chapter 4, data generation methods in Chapter 5 and a discussion and concluding remarks in Chapter 6. The publications and ideas presented throughout chapters 3,4, and 5, follow a specific roadmap that builds upon and around the idea of image-to-image translation models as a core component.

We begin with an improvement to multi experience map-based localisation - a powerful technique, but which requires a new, complete traversal of the environment whenever localisation starts failing due to environmental conditions, which is an expensive process. This is made worse by the fact that new data must usually be associated with the existing map data (e.g. through loop-closures), which often fails if the domain gap is too large. Other sensor modalities such as LIDAR can help with this issues, but at a large increase in cost and complexity. The process could be significantly simplified (and costs reduced) if the already existing structured map data could be adapted to work under any new condition that the system is experiencing. Ideally, this should be done by collecting as little new data as possible, with as little supervision or labelling efforts as possible. Examples of previous learning-based approaches to this issue include learned descriptors [2, 30] and direct pose regression [6].

Conditional GANs and image-to-image translation techniques have evolved to produce data that can be directly used by existing localisation pipelines. As a novel solution, we propose to transform the appearance of existing map data into new, discretized styles/appearances that match the target domain of operation. This is accomplished using image-to-image translation and a novel loss designed to improve feature matching. Section 3.1 describes the approach and presents significant improvements to visual localisation performance.

While this method shows that performance can be significantly improved, there still remains the issue of discretized conditions and having to retrain the model for each new domain. Additionally, it is often not feasible to have a hand-crafted

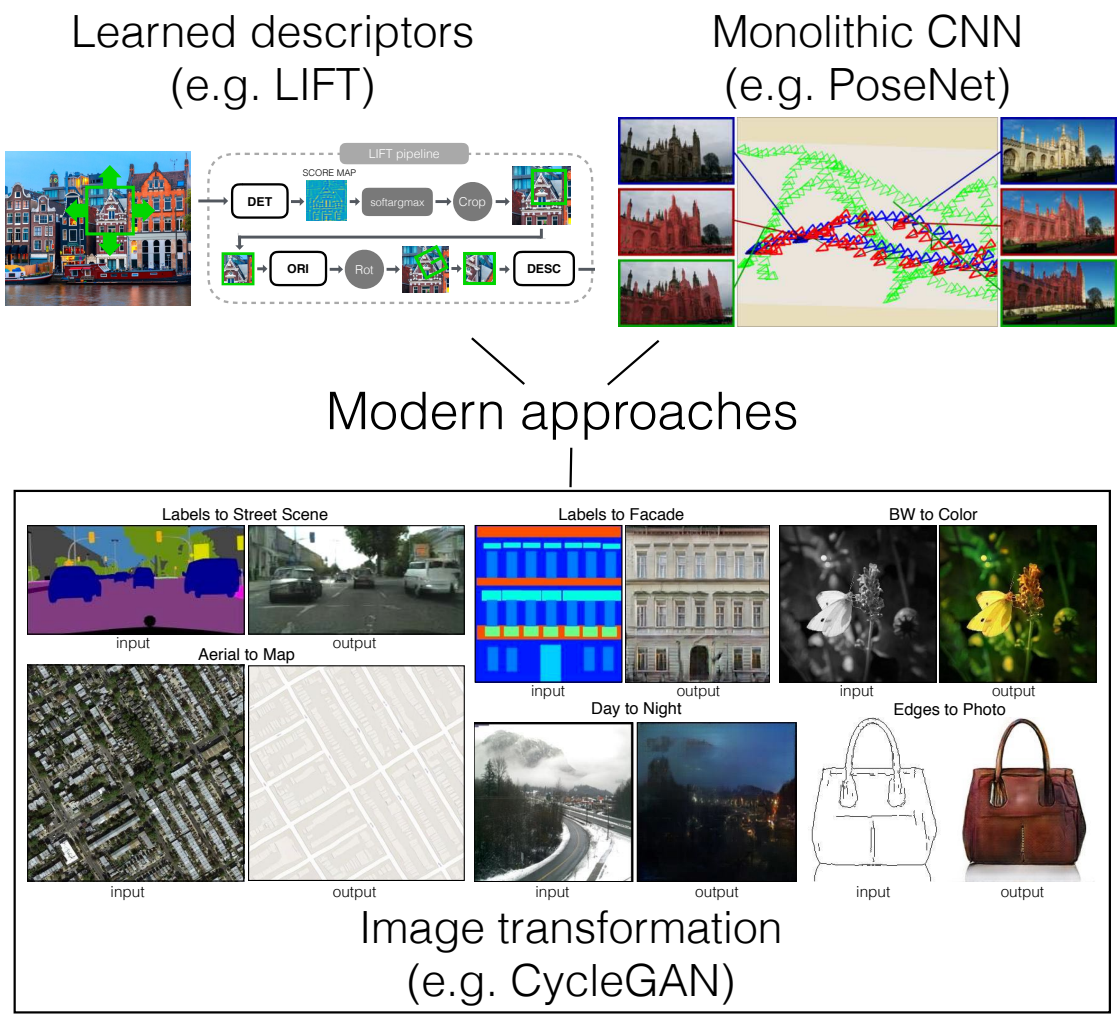


Figure 1.6: Modern learned localisation techniques: examples of existing techniques that improve the robustness and accuracy of localisation pipelines include learned feature detectors and descriptors such as LIFT, and monolithic CNNs that directly regress a pose. We propose using direct image translation to change the appearance of images, making matching and retrieval easier.

objective function for each task. A natural extension is a system that does not impose an explicit target appearance, instead optimizing the resulting images such that they improve the performance of downstream tasks. A further improvement is the ability to constantly evolve and expand the number of compatible target domains of operation, in a self-supervised fashion, yielding a system that improves as it traverses new environments. Section 3.2 describes the approach and the domain adaptation results obtained.

Both aforementioned methods make extensive use of cycle-consistency generative

models to leverage unlabelled or unpaired data. Models such as CycleGAN produce visually-pleasing outputs, but the pixel-wise reconstruction loss forces some information from the source domain to be encoded in the output as a high-frequency signal, and subsequently used by the second generator in reconstructing the original input [31], which is highly undesirable when the end goal is to produce training data for other models. Methods that reduce this effect exist, but tend to increase model complexity (limiting online or high-resolution operation) and may reduce training stability or the quality of the outputs [25, 26]. We introduce a method that maintains the simplicity and inherent stability of CycleGAN while reducing the amount of source information encoded in the output, providing training data with distributions that are more closely aligned with the target domain. Section 5.2 describes this approach and its impact on data generation.

However, it is not always required or necessary to translate or synthesize full, complete images or frames. Synthesizing only parts of frames has applications in domain randomization and re-biasing of datasets, and can be used to easily complement or extend existing datasets. Section 5.3 presents an application of synthesis to road marking segmentation, showing that performance can be drastically increased without any manual labelling effort, or similarly that costs can be reduced by extending a much smaller labelled dataset.

The methods described so far looked at image-to-image translation for domain adaptation, however image to image translation is applicable to a wide range of other problems in computer vision. One suitable area of application is that of image denoising, and in our specific case the removal of adherent water droplets, which affect the performance of semantic segmentation, object detection etc. Large convolutional models are inherently good at learning image priors and statistics [1, 32, 33] and thus are a good candidate for image restoration. The same class of architecture used in the domain adaptation experiments is modified for the task of removing the effects of adherent water droplets on camera lenses. However, while the previous approaches were suited to weak supervision, learned image restoration requires stronger supervision, to reduce or eliminate hallucinations, which are

especially dangerous in the case of field robotics. This highlights the need for a new dataset, requiring the design and implementation of a system for gathering images affected by rain droplets along with corresponding clean groundtruth. While previous methods required a static environment and could only capture images at a very low rate, we perform data capture in a dynamic environment, using a data collection vehicle, yielding a diverse dataset. The experiments show that image-to-image translation models are indeed very good at reconstructing corrupted images, establishing new state of the art results without the need for additional mechanisms such as attention [10]. Section 4.1 describes this approach and shows how model performance on rain-affected images can be restored.

The data collection mechanism presented above suffers from a number of limitations: it must be mounted on a data collection vehicle (or a movable platform), requires to be operated in clear weather, and data collected must then be labelled if extra annotations beside clean groundtruth are required (e.g. semantic segmentation). Ideally, one would like to make use of already-annotated datasets, while reducing or completely removing the need for human intervention (i.e. driving) and synchronization with weather. Section 5.1 extends and modifies the data collection method to allow addition of water droplets to datasets that already possess annotations. This improved technique uses a static rig, and once set-up does not require any human intervention. The results show that the resulting rainy images can be used to successfully train a de-raining model that significantly boosts performance on image segmentation tasks.

Another type of problem that can benefit from strong image/scene priors in the context of image to image translation is that of Inverse Perspective Mapping (IPM), which is used for tasks such as path planning, road scene understanding and ADAS systems. IPM is usually based on the assumption of a ground plane, which can break down due to irregularities in the road and a difficulty in defining and identifying the boundaries of the road surface. More importantly, its geometrical formulation coupled with camera sensor limitations produces image regions that are affected by data sparsity, leading to a drastic reduction in quality as a function of distance from

the camera. Features that are far away have a much lower effective resolution than features close by. To tackle this effect, we enhance the perspective transformation with denoising and smoothing, while also dividing it into multiple incremental steps. The motivation behind this is that interpolation leads to distortions and a loss of fidelity, while an image-to-image translation model can be used to restore this using learned scene priors. If done incrementally, the quality of the resulting final image may be improved. Similarly to the previous approaches, a supervised training approach is required. The assumptions behind IPM can mean that portions of the scene that are occluded in the image space become de-occcluded in the IPM images - minimizing hallucinations is thus crucial. We leverage a state of the art visual odometry pipeline to create a dataset with groundtruth, made possible by assuming that the features at a small distance from the camera retain their quality in the IPM space. Slices of consecutive frames are stitched together using poses from visual odometry, yielding high-quality ground-truth IPM images. Section 4.2 presents this approach, along with an IPM model trained with the generated groundtruth. Experimental results show that the method can significantly improve the quality of IPM images and the performance of scene understanding models on them.

Many of the methods mentioned thus far point towards increased availability of high quality semantic segmentation models under many different weather and illumination conditions. Bearing this in mind, Section 5.4 presents an application of collision avoidance using reinforcement learning, where the only input is a semantic segmentation map. Compared to approaches that consume raw images, this method has the advantage of being developed using data from driving simulators, while also potentially offering invariance to the target domain. Furthermore, the underlying model can be made much simpler and modular, and a number of frame prediction techniques can be applied to boost model performance. This work is an outlier in that it does not directly follow the roadmap laid thus far, other than by way of assumed general availability of semantic maps. However, it does represent a viable example of a downstream client or consumer for the data created, de-noised or translated using some of the methods described above.

2

A brief overview of relevant techniques

Contents

2.1 Introduction 17
2.2 Visual localisation 18
2.2.1 Map-based representations 19
2.2.2 Multi experience navigation 20
2.3 Convolutional neural networks 20
2.3.1 Intro to convolutions 20
2.3.2 CNNs 23
2.4 Generative adversarial networks 23
2.4.1 Vanilla GAN 25
2.4.2 Conditional GAN 26
2.4.3 Cycle-consistency GAN 27
2.4.4 GAN training tricks 28

2.1 Introduction

This chapter provides a summary description or the major/important prior art and existing methods that this thesis builds upon or makes use of. While each published paper included in this thesis includes its own literature summary, we take the opportunity here to call out seminal and foundational work which we draw on heavily, including visual localisation techniques, a quick primer on Convolutional Neural Networks (CNNs) and their application to Generative Adversarial Networks

(GANs) [1] and a few tricks for stabilizing GAN training.

2.2 Visual localisation

Vision-only localisation is of great importance in the robotics world, especially due to the low-cost and prevalence of cameras in almost any mobile device, from phones to cars. Roughly-speaking, visual localisation can be divided into two main categories: topological localisation or place recognition, which returns a global, topological location in a map, and metric localisation, which returns a precise pose, either global or relative to a specific location (or keyframe).

Topological visual localisation, or place recognition, involves recognising and matching images or scenes based on their appearance or common elements, without computing a precise pose, in a similar fashion to image retrieval. FABMAP [34] represents scenes using a local feature bag-of-words approach, while SeqSLAM [35] uses pixel values and image sequence information to match across changes in appearance, while others combine sideways-facing cameras, multi-scale image matching and sky-removal to improve the invariance to viewpoint changes. However, these methods break down in the presence of large changes in appearance and/or viewpoint. A common way of improving the accuracy or robustness of topological localisers is by using features extracted using a convolutional neural network [4].

On the other hand, metric localisation involves computing an N-DOF pose (e.g. a 6DOF pose in 3D space) with respect to a reference frame. Indirect methods typically use a combination of feature (SIFT, SURF, BRIEF, ORB, FAST [20, 36–39]) detection, extraction and matching between frames, followed by a pose estimation step that can additionally be made more robust to noise using techniques such as RANSAC [40]. Direct methods [41] make use of whole images and pixel information instead of using only sparse point-like features, and can exhibit higher robustness under changes in viewpoint but have a larger chance of breaking down under large changes in appearance and illumination, especially when a photometric error is used. Generally, indirect methods are much faster and less resource-intensive than direct methods, making them suitable candidates for mobile platforms.

Both topological and metric localisation typically make use of an encoding of the environment that the robot will be operating in. This can be a database of images or features extracted from images [5], or even a full pointcloud or dense reconstruction [42].

2.2.1 Map-based representations

A common solution to the localisation problem is to build a map of the robot's environment which allows unambiguous localisation irrespective of the robot's orientation and position. Two common methods involve building either a dense representation of the environment, through techniques such as structure-from-motion [43] or point-cloud integration, or a sparse encoding which stores some representative features of each (key)frame along with transforms (usually 6DOF) between frames [5, 44]. This thesis uses the latter as a framework for assessing performance of individual techniques, and as a backbone for storing and manipulating ground-truth and performing data association.

In the current context, a **map** is a database that contains, for each point of interest (keyframe) that makes up the environment of the robot, a set of images taken from the point of view of the robot and features extracted from them. This set can be extended using data from other sensor modalities, such as LIDAR [42], RADAR [45], UTM coordinates and even frames from cameras pointing in other directions. Such a set is referred to as a "node". Additionally, a list of 6DOF transforms between keyframes (nodes) is maintained, allowing the pose of any two arbitrary keyframes to be computed. Such a transform is referred to as an "edge". Thus, the map becomes a graph representation of the robot's environment. Ideally, this map allows for both topological localisation (or 'global' localisation, as in the kidnapped robot problem), where the closest node(s) to the robot is identified, and metric localisation, where a precise pose is computed with respect to the closest node(s).

2.2.2 Multi experience navigation

One of the central themes of my research is that of visual localisation under any environmental conditions and any time of day, be it night, day, winter, summer, etc. This generally involves being able to match the current camera image to some image or associated features stored in the map, so that the robot can understand its topological or precise location in the world. A very common scenario is that the image stored in the map has a **very** different appearance than what the robot is currently seeing. This makes it very hard to perform matching and localisation. Traditional sparse localisation methods generally use one or more types of keypoint features (such as SURF, SIFT, BRIEF or ORB) detected in each frame, followed by matching and pose solving steps, but these methods break down in the presence of large appearance changes. In order for these point-feature based methods to work, the mapping procedure needs to be performed for each category of appearance (sunny, snow, dawn, dusk, rain, night etc). A natural extension of the graph-based map allows for multiple mapping traversals, or 'experiences' to be encoded, each with its own graph. Each experience may be created under varying categories of appearance or by following a different path. Experiences may be linked using loop-closures. It is easy to see that it becomes difficult and expensive to create a map that covers all common categories of appearance, and that performing loop-closures between experiences through visual feature matching requires these experiences to be close to each other in appearance space.

2.3 Convolutional neural networks

2.3.1 Intro to convolutions

In general terms, convolution is the mathematical operation of combining two inputs, in order to produce an output which describes how one input is modified by the other and vice-versa. In image processing, 2D convolutions are extensively used in order to extract intrinsic characteristics present in the data such as edges, or modify the input image, such as blurring. As depicted in figure 2.1, a 2D convolution consists

in sliding a **kernel** (matrix of weights) over the 2D input, performing element-wise multiplication between the kernel and the input values at its current location, and summing up the results into a single value. By repeating the calculation for all the locations where the kernel is slid over, the resulting 2D output will be a collection of weighted sums of the input values, representing a **feature map** of the input.

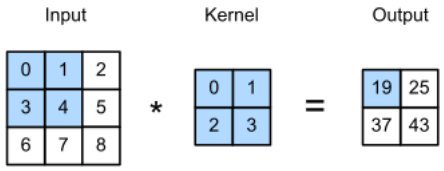


Figure 2.1: Convolution with no padding (image source: [46])

There are two main assumptions that convolutional neural networks are based on. One is locality of low-level features, and the other is that these features could exist in multiple locations. The second assumption is addressed by sliding the same kernel over the whole input, in order to extract the feature from different parts of the image. The first assumption is addressed by the choice of **kernel size** - the number of input features that are combined in order to produce one output of the feature map. A small kernel size will have a small **receptive field** since it only looks at a few pixels at once. The features are thus complex, small, and the information extracted is very rich. On the other hand, a large kernel will have a larger receptive field, the features are simpler and less information is extracted - making a large kernel size suitable for cases where the assumption of low-level locality is not that strong (for example, images that have a central structure, like a face).

The size of the output feature map also depends on how the input is sub-sampled, which is given by the **stride** of the kernel, i.e. the number of pixels that the kernel is shifted across the image at each step. A stride of length 1 will result in a large overlap between the input features, whereas a stride equal to half the size of the kernel results in no overlap at all. Generally, in order to make sure that a model is translational invariant, a stride of length 1 is chosen.

Since the convolution operation can lead to reduced size of the feature map as compared to the input size, **padding** the input with zeros is a general practice.

Additionally, the contribution of bordering pixels is enhanced, since they are brought closer to the centre of the input, in order to avoid "washing away" that information.

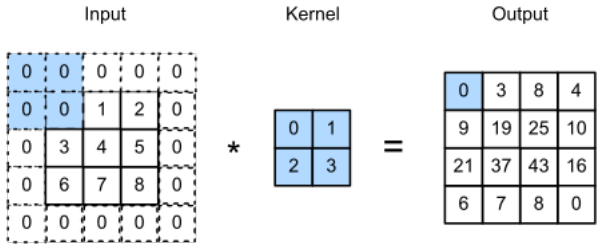


Figure 2.2: Convolution with padding (image source: [46])

One important characteristic of images is the high correlation between neighbouring pixels, which justifies the need to reduce the dimensionality of filter responses. This is accomplished by an operation called **pooling**: a discretization process in which a window is slid across the 2D input, and a function is applied on the values beneath the window. For **max pooling**, the maximum value from the window will be kept, whereas for **average pooling**, the average of the values will be computed.

Modern image translation models have come to rely less and less on pooling for dimensionality reduction, and more and more on kernel strides greater than 1.

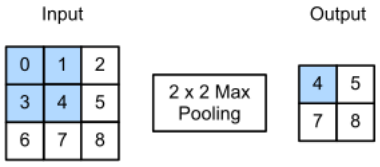


Figure 2.3: Max pooling (image source: [46])

Another important concept is the **activation function**, which is simply a transformation function that maps an input to an output. There are 3 types of activations: binary step functions (threshold-based), linear activation (the output signal is proportional to the input), and non-linear activations. Since the convolution is a linear operation, a non-linear activation function is needed in order to add non-linearity to the network, and to ensure that back-propagation is possible.

The most commonly applied non-linear activations are: **sigmoid**, hyperbolic tangent (**TanH**), rectified linear unit (**ReLU** [47]) and leaky rectified linear unit

(**Leaky ReLU** [48]). The sigmoid activation function has a smooth gradient, but its outputs are not centred around zero (introducing a bias for further layers), suffers from the vanishing-gradient problem (for input values close to the two extremes of the function range, the output hardly changes) and is also computationally inefficient. The TanH activation has zero-centred outputs, but suffers from the same problems as the sigmoid activation. The ReLU activation only activates if the input is larger than 0, thus making it computationally efficient, but for inputs that are negative or close to 0, some weights are not updated during the back-propagation process, leading to dead neurons. Leaky ReLU is the improved version of ReLU, which, for negative inputs will output a very small linear output, such that the gradient is non-zero, solving thus the dying ReLU problem.

2.3.2 CNNs

Convolutional Neural Networks (CNNs) are neural networks that are designed specifically to process multi-dimensional (usually 2D and 3D) input data using convolution operations. Similar to fully-connected networks, CNNs can have multiple layers, with each layer applying an N-dimensional convolution kernel over the output of the previous layer.

CNNs are a good candidate for finding patterns in data and learning image features without any explicit hand-engineering of convolutional kernels. By inspecting the activations of a CNN, it has been shown that the first convolutional layers learn simple features such as edges and simple textures, whereas deeper layers learn complex patterns and shapes of objects. A convolutional network is thus better that fully connected layers at capturing spatial and temporal dependencies, by applying relevant filters, while possessing a degree of invariance to translations in the input space.

2.4 Generative adversarial networks

A generative adversarial network is a type of architecture where two networks (usually CNNs) compete against each other in a minimax setup: one acting as a

forger and another one acting as a judge. The generator (forger) tries to produce data that comes from a particular distribution, while the discriminator (judge) tries to determine whether the data comes from that distribution or not.

Figure 2.4 presents a real-life analogy to the problem: although real money is printed by the central bank, counterfeiters can use paper, ink, and other resources to create fake bills. The purpose of the detective is to decide which bills are real and which are counterfeit, in order to punish the counterfeiters. Thus, the goal of the two entities is to minimize their own losses: the detective wants to be able to correctly discriminate between real and fake bills, whereas the counterfeiter wants to forge bills that look exactly like the real ones. However, to be able to create bills that look legitimate, the counterfeiter must learn the characteristics of the real bills, or in technical terms, the generator must learn the distribution of the real data.

Since the generator and discriminator are parametrized by two networks that need to be trained simultaneously, training a GAN involves solving the minimax problem over their parameters, and finding a balance between the two opposing forces. Unfortunately, because in the equilibrium, gains for one network can only come at the expense of the other, achieving the equilibrium point is difficult.

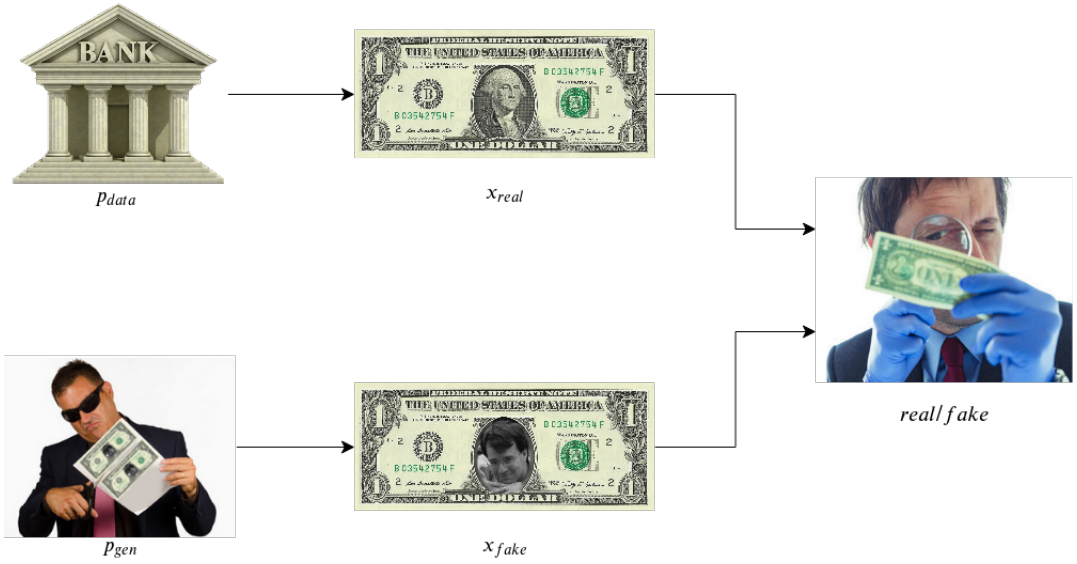


Figure 2.4: GAN setup: the forger tries to create a dollar bill that looks like the real one, while the detective tries to guess which dollars are real and which are counterfeit

2.4.1 Vanilla GAN

In the most basic GAN architecture [1], a generator $G(z)$ is trained to generate fake images $\hat{x}$ from input noise z, while the discriminator D is trained to distinguish real images x from fake images $\hat{x}$, by optimizing the following loss function:

$$\min_G \max_D L(D,G) = \mathbb{E}_{x\sim p_r(x)}[\log D(x)] + \mathbb{E}_{z\sim p_z(z)}[\log(1 - D(G(z)))] \tag{2.1}$$

The discriminator's goal is to maximize the loss, whereas the generator's goal is to minimize it, each of the networks tweaking its own parameters.

From D's point of view, maximizing the loss implies maximizing both terms. That is, for a real image x, D should output 1 (detecting a real image as real), such that the term $\log D(x)$ attains its maximum value 0. If instead, D outputs a value close to 0 (mistaking a real image for a fake image), the value of the term will be negative. For a fake image $\hat{x}$, D should output 0 (detecting a fake image as fake) such that the term $\log(1 - D(G(z)))$ attains its maximum value 0. If instead, D outputs a value close to 1 (mistaking a fake image for a real image), the value of the term will be negative.

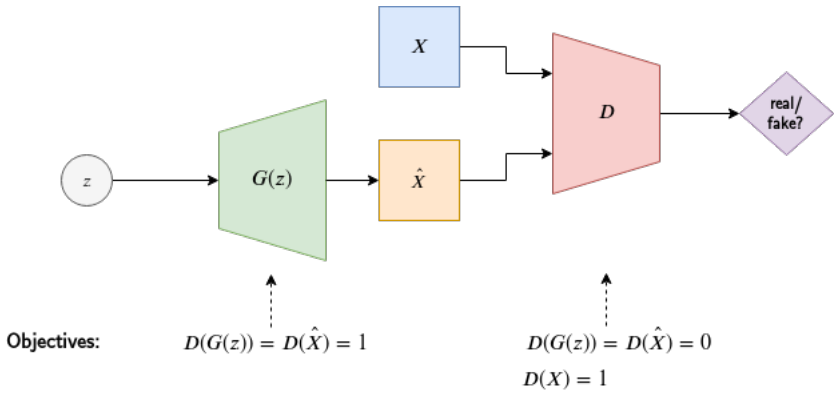


Figure 2.5: Vanilla GAN architecture. No direct pixel-wise loss is used. Instead, a discriminator network D is trained to distinguish between real images x and images $\hat{x}$ generated by the generator network G from a latent embedding z.

From G's point of view, minimizing the loss implies minimizing only the second term, because the first one does not depend on it. For a fake image $\hat{x}$, the generator

would want the discriminator to output 1 (to mistake a fake image for a real image), such that the term $\log(1 - D(G(z))) = \log(1 - D(\hat{x}))$ will attain its minimum negative value. If, on the other hand, the discriminator is not tricked (detecting the fake image as being fake) and outputs 1, the term will attain its maximum value 0.

In the ideal equilibrium, the generator has learned the training data distribution, thus the discriminator will be undecided whether a fake image is indeed fake or real or, in other words, should output a probability of 0.5.

2.4.2 Conditional GAN

Since the vanilla GAN only takes as input a noise vector z, the generative process is unconditional, and thus hard to control. In order to address this issue, conditional GANs [49] that take as input extra information have been proposed, in order to impose more structure in the output. The objective function thus becomes:

$$\min_G \max_D L(D, G) = \mathbb{E}_{x \sim p_r(x)}[\log D(x|y)] + \mathbb{E}_{z \sim p_z(z)}[\log(1 - D(G(z|y)))] \quad (2.2)$$

For example, an early version of a conditional GAN was presented in [49], where conditioning is accomplished by feeding extra information y through an additional layer to both the generator and discriminator. The extra information can come from class labels or other modalities.

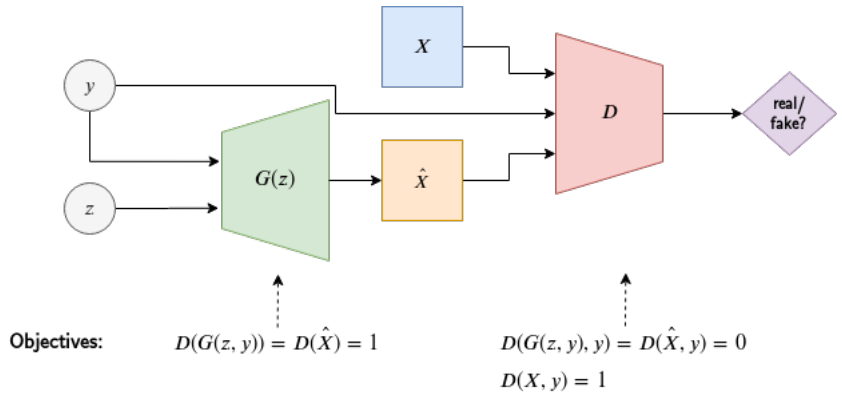


Figure 2.6: Early conditional GAN architecture. The architecture is similar to Vanilla GAN, but in this case the generator network G is additionally constrained by a class encoding y, allowing for conditional generation of outputs $\hat{x}$.

Later, conditional GANs were extended with encoders that process the real image x into a latent embedding [50], instead on relying on input noise z, as presented in figure 2.7.

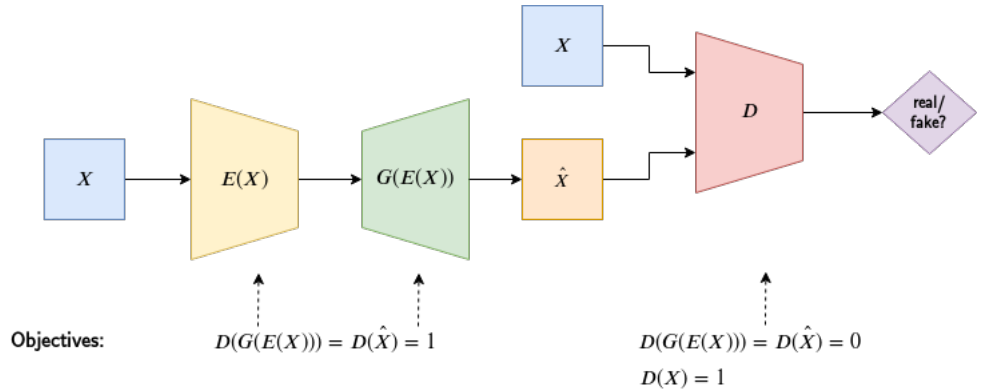


Figure 2.7: Late conditional GAN architecture. Instead of conditioning the generator G on a class encoding, an encoder E conditions the latent embedding z (or $E(x)$) on an input from the real domain x. In other words, z becomes an encoding of x in some lower-dimensional space.

2.4.3 Cycle-consistency GAN

CycleGANs [8] have been proposed for the task of image-to-image translation, and the architecture consists in combining 2 conditional GANs in a cycle-consistent architecture, as presented in figure 2.8 . The adversarial loss for the first generator is:

$$L(G_{AB}, D_B, A, B) = \mathbb{E}_{b\sim p(B)}[\log D_B(b)] + \mathbb{E}_{a\sim p(A)}[\log(1 - D_B(G_{AB}(a)))] \quad (2.3)$$

$$L(G_{AB}, D_B, A, B) = \mathbb{E}_{b\sim p(B)}[\log D_B(b)] + \mathbb{E}_{a\sim p(A)}[\log(1 - D_B(\hat{b}))] \quad (2.4)$$

Whereas the adversarial loss for the second generator is:

$$L(G_{BA}, D_A, B, A) = \mathbb{E}_{a\sim p(A)}[\log D_A(a)] + \mathbb{E}_{b\sim p(B)}[\log(1 - D_A(G_{BA}(b)))] \quad (2.5)$$

$$L(G_{BA}, D_A, B, A) = \mathbb{E}_{a\sim p(A)}[\log D_A(a)] + \mathbb{E}_{b\sim p(B)}[\log(1 - D_A(\hat{a}))] \quad (2.6)$$

As opposed to the GAN architectures presented earlier, the learning is unsupervised. For example, GANs are used to generate style images from sketches (paired dataset), whereas cycleGANs are trained on unpaired datasets - data where there

doesn't exist another style for the same image. Unfortunately, the discriminator is only able to impose constraints on the distribution of the image and not the structure as well, leading to an unconstrained mapping between the two domains. To solve this, a cycle-consistency loss is introduced, in order to diminish the space of possible mappings. The cycle-consistency loss is thus:

$$L(G_{AB}, G_{BA}) = \mathbb{E}_{a\sim p(A)}[\|G_{BA}(G_{AB}(a)) - a\|_1] + \mathbb{E}_{b\sim p(B)}[\|G_{AB}(G_{BA}(b)) - b\|_1] \tag{2.7}$$

Finally, the overall objective can be written as:

$$G^*_{AB}, G^*_{BA} = arg \min_{G_{AB},G_{BA}} \max_{D_A,D_B} L(G_{AB}, G_{BA}, D_A, D_B) \tag{2.8}$$

Where the full loss function is given by:

$$L(G_{AB}, G_{BA}, D_A, D_B) = L(G_{AB}, D_B, A, B) + L(G_{BA}, D_A, B, A) + \lambda L(G_{AB}, G_{BA}) \tag{2.9}$$

2.4.4 GAN training tricks

GANs are known to suffer from training instabilities, which most often lead to either mode-collapse or non-convergence. Some methods that are relatively simple to implement and do not have many hyperparameters that can be tuned are:

- learning rate scheduling [51, 52]: the learning rate is either reduced throughout the training run, or follows a periodic function (sine, triangular/sawtooth).
- progressive growing of input resolution: training starts with low-resolution images, low-dimensional or sparse data, and subsequently the resolution is slowly increased whenever the training loss (or a similar metric such as loss variance or performance on a downstream task) stops improving. Methods such as [53] progressively grow both the inputs and the model, but we've empirically determined that starting with a fully-parametrized model and only growing the inputs is satisfactory and gives most of the benefits of the full method.

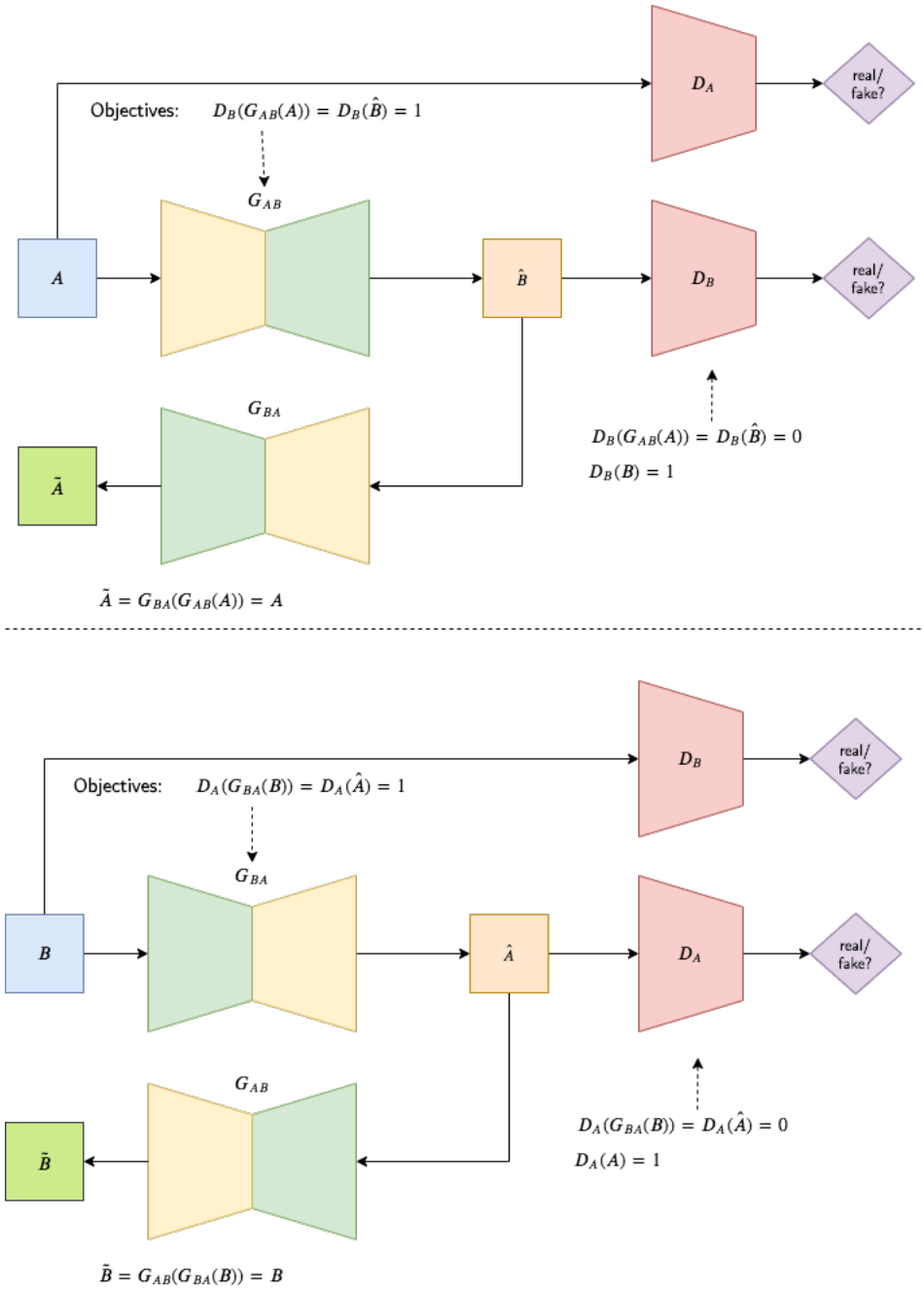


Figure 2.8: CycleGAN architecture. Two conditional GANs form a cycle, where an input A from a source domain is translated into a representation $\hat{B}$ from a target domain, followed by a reconstruction $\tilde{A}$ of the original input A from the source domain. The reverse cycle B-$\hat{A}$-$\tilde{B}$ is constructed as well. A discriminator loss is applied to each of $\hat{A}$ and $\hat{B}$. A pixel-wise reconstruction loss is applied between $\tilde{A}$ and A, and similarly $\tilde{B}$ and B. This architecture allows the use of unpaired inputs in training translation tasks. Diagram credits: Valentina Mușat

- residual skip connections: not all tasks require that the output is solely a decoding of an embedding (bottleneck), and instead many image-to-image translation tasks can be more easily reformulated as a residual or corrective transform, where the output becomes $y = x + F(x)$. For example, day to night translation is very conducive to additive or subtractive transformations, whereas a task such as reprojection or inverse perspective mapping, where the structure of the output may change, will most likely not benefit from direct transfer of information from inputs to outputs. Similar to the 'long' connections in DenseNet [54], we make use of a UNET-like [55] architecture with residual instead of concatenative skip connections.

- using an image pool to reduce model oscillations: one useful trick for smoothing out gradients is to keep a buffer or history of previously generated images [56], and, for the current iteration, to backpropagate using a generated image sampled from the buffer. The newest generated image then gets added to the buffer, randomly replacing another image from the buffer. This is, on one hand, conducive towards a smoother descent towards a minimum, and on the other hand can slightly improve the statistics learned by the discriminator by preventing (catastrophic) forgetting. A similar training stabilisation effect can be achieved by using a running average of the parameters of the model instead of directly updating the parameters using the latest gradients [57].

- using a patchGAN discriminator: instead of classifying the entire image as real or fake, a patchGAN discriminator classifies patches of the image as real or fake. This allows backpropagation updates to more granularly penalize models at the level of patches instead of at the level of a whole image, and produces a discriminator model that is more sensitive to higher frequency details, yielding generator models that produce better textures [17, 58].

- using multiscale discriminators: similar to [18], we make use of multi-scale discriminators, with one discriminator for each image scale. This allows for discriminators that have differentiated sensitivities to different frequencies in

the generated image. The discriminator operating on the smaller scales will have a larger effective receptive field, and will affect the lower frequencies and global features. The discriminator operating on larger scales will conversely have a smaller effective receptive field and be more conducive towards higher frequency details such as textures.

3
Domain adaptation

Contents

3.1 Adversarial Training for Adverse Conditions 34
3.2 Don't Worry About the Weather 44

3.1 Adversarial Training for Adverse Conditions: Robust Metric Localisation using Appearance Transfer

Problem: Traditional techniques used in robust visual localisation, such as point-feature detection, description and matching work well under ideal weather and illumination conditions, but break down quickly in the presence of changes in appearance and illumination.

Solution: The use of a preprocessing step that can transform the appearance of images, which is explicitly designed to aid and abet feature-matching. Summarily, this is done by using a cycle-consistency GAN architecture with a new bespoke loss based on the SURF detector and pixel-wise descriptor maps.

Status: Published at ICRA2018

Adversarial Training for Adverse Conditions: Robust Metric Localisation using Appearance Transfer

Horia Porav, Will Maddern and Paul Newman

***Abstract*— We present a method of improving visual place recognition and metric localisation under very strong appearance change. We learn an invertable generator that can transform the conditions of images, e.g. from day to night, summer to winter etc. This image transforming filter is explicitly designed to aid and abet feature-matching using a new loss based on SURF detector and dense descriptor maps. A network is trained to output synthetic images optimised for feature matching given only an input RGB image, and these generated images are used to localize the robot against a previously built map using traditional sparse matching approaches. We benchmark our results using multiple traversals of the Oxford RobotCar Dataset over a year-long period, using one traversal as a map and the other to localise. We show that this method significantly improves place recognition and localisation under changing and adverse conditions, while reducing the number of mapping runs needed to successfully achieve reliable localisation.**

I. Introduction

In this paper we present a way of improving the performance of camera-only metric localization in environments with large changes in appearance, (e.g. day-night) using images synthesized by a CNN, and introduce a new loss based on SURF detector and descriptor maps that improves the feature matching between the synthesized and real images.

Keypoint feature matching (such as SURF, SIFT, BRIEF or ORB) represents the leading approach in multi-map visual localisation systems, as used in [1], [2], [3], [4], [5] and many other commercial systems. While this works well when matching scenes with similar lighting conditions, performance quickly breaks down when the appearance of the scenes differs due to changes in illumination levels, or seasonal differences. Attempting to use standard descriptors will likely lead to poor or impoverished localisation. In order for these point-feature based methods to work, the mapping procedure would typically need to be performed for each category of appearance (sunny, snow, dawn, dusk, rain, night etc). While most approaches so far have attempted to improve the feature detectors and descriptors, we propose to instead modify the input images. In this way, the modified images can be used with any existing system based on feature-matching, without changing it. This paper builds on previous work by [6], [3] and [7], and is motivated by two drawbacks of the current approaches :

1) Mapping for all conditions is time consuming, expensive and may well be impossible in certain cases.
2) The quality of data collected during adverse conditions can be worse, leading to lower accuracy in the generated maps.

Authors are from the Oxford Robotics Institute, University of Oxford, UK. {horia,wm,pnewman}@robots.ox.ac.uk



Fig. 1. Improved feature-based localisation using synthesized images. The top row shows point feature matches between a day map keyframe (left) and the live camera night image (right). The bottom row shows the improved matches between the day map keyframe (left) and the synthetic day image (right) obtained by transforming the live camera night image. The key to our approach is explicitly accounting for feature detection and descriptor extraction using differentiable components of the synthetic image pipeline. Our system drastically reduces the amount of mapping needed to achieve successful localisation under challenging adverse conditions.

The key contribution of our work is to incorporate a differentiable feature detector and descriptor pipeline as part of an adversarially-trained network designed to synthesise images of the same location under different conditions. Building on the work of [7], we train a CNN to output synthetic images in two steps:

- We select a complete source-condition traversal and a small portion of a target-condition traversal and, without any alignment of the images, train a pair of image synthesizers using a cycle-consistency loss, discriminator loss and feature detector and descriptor losses.
- Afterwards, using metric 6DoF groundtruth, we select a small number of well aligned target- and source-condition image pairs, and separately fine-tune the image synthesizers using an L1 loss on the feature detector and descriptor outputs.

We expand on suggestions by [8] and [9] to speed up descriptor computations, and take inspiration from [10], who introduce differentiable HOG descriptors and use it in a pipeline to improve rendered images. We describe an algorithm for quickly and efficiently computing dense, per-pixel differentiable descriptors, and use it to compute descriptor losses alongside the reconstruction and discriminator losses. The synthetic images are then used in a stereo localisation pipeline based on [3]. This pipeline first performs place recognition, outputting candidate frames that are most likely to be from the same place as the robot's live frame, and

subsequently, uses keypoint feature matching to compute a metric pose between the live frame and the frame(s) retrieved during place recognition. We benchmark our localisation results against an RTK-GPS ground truth from the Oxford Robotcar Dataset introduced by [11] and present the results in section V.

II. Related Work

Topological Localisation: Synthetic images have seen use in topological localisation, with [12] showing that point-of-view makes a large difference when matching across images with large changes in appearance, and subsequently using Google Streetview panoramas to synthesize images that match the point of view of a query image. Co-visibility graphs for CNN features were used by [13], boosting the invariance to viewpoint changes by adding features from neighbouring views to the image that is currently queried, essentially creating 'in-between' views. In [14], features that co-occur for each image taken at different times of the day are combined into a unique representation that contains features identifiable in any image from that point of view, irrespective of illumination conditions.

Changes in an image across seasons were predicted by [15] using vocabularies of superpixels and used for place recognition. How a winter scene would look in the summer (and vice-versa) is predicted by first partitioning the image into superpixels, associating each superpixel with an entry in the winter vocabulary and then replacing the superpixel with its counterpart from the summer vocabulary by using a dictionary that maps between vocabularies. However, these synthetic views suffer from poor alignment or degradation of point features, meaning that metric localisation cannot be achieved.

Metric Localisation: In the context of metric localisation, [16] refine the multi-experience approach by introducing a step that creates a 'summary map' from multi-experience maps during online localisation, using a framework of sampling policies and ranking functions. However, their approach does not mitigate the need to perform mapping under various conditions. In [17], authors generate different views of the images to be matched using affine transforms and apply SIFT to these generated views, showing an increased robustness to changes in viewpoint, but without addressing changes in appearance.

High dynamic range image matching in the context of visual odometry is improved in [18] by training an LSTM deep neural network to produce temporally consistent enhanced representations of the images. However, visual odometry works with images that are temporally close with few lighting or appearance changes.

Appearance Transfer: Synthetic image techniques have been used for other tasks, but show promise in localisation contexts: [19] use local affine transforms in a color transfer technique to 'hallucinate' the appearance of an image at different times of day. In [20], images are decomposed into style and content representations using a VGG-19 network, followed by synthesization, starting from a white noise image and using gradient descent, of an image that matches the content representation of the input image and the style representation of the target image.

Closer to our approach, [7] demonstrate unsupervised image-to-image translation by training a pair of CNN generators, G that maps X to Y, and F that maps Y to X, and apply a 'cycle consistency' L1 loss between X and $F(G(X))$ along with discriminator losses L_{G} on the output of G and L_{F} on the output of F. Similarly, [21] train a pair of VAE-GANs for image to image translation using unaligned images, but do not use a cycle-consistency loss, instead choosing to partially share high-level layer weights and to share the latent-space encodings between the VAEs. The authors train on different domain transformation pairs, including semantic-labels to natural images and day images to night images, but do not demonstrate any topological or metric localisation applications. We believe we are the first to directly address metric localisation using an appearance transfer approach.

III. Learning to generate images

We base our detection and matching pipeline on the well-known SURF feature [22], and employ a 2-stage training strategy. First, we use a cycle-consistency architecture similar to [7], to train a generator to transform an input source image into a synthetic image with a target condition. This synthetic image is subsequently transformed by a second generator back into a reconstructed image that has the initial condition, with the process being repeated in the reverse direction. We then fine-tune our image generators independently using a well-aligned subset of the dataset.

In the first stage, two generators, G_{AB}, which transforms condition A into condition B, and G_{BA}, which transforms condition B into condition A, are trained using a collection of unpaired source and target images. A discriminator loss is applied on the synthetic images, and an L1 loss is applied between the reconstructed images and the input images. Additionally, we compute SURF detector response maps on the reconstructed and input images and apply an L1 loss between them, and similarly compute dense per-pixel SURF descriptor maps on the reconstructed and input images and apply an L1 loss between them; these methods are further described in III-A and III-B. The architecture of the first stage is shown in Fig. 2.

In the second stage, G_{AB} and G_{BA} are separately trained using a small dataset of aligned day and night images. The use of pixel-aligned images allows the generators to learn certain feature transformations that might have been uncaptured by the unsupervised method used in the first stage, which only learns to align the image distributions without any explicit pixel-wise mapping. This time, we apply the L1 loss between SURF detector response maps computed on aligned target images and synthetic images, and between dense descriptor response maps computed on aligned target and synthetic images. The architecture of the second stage is shown in Fig. 3.

The generator architecture is based on UResNet [23], which combines a UNet [24] with residual (ResNet) [25]

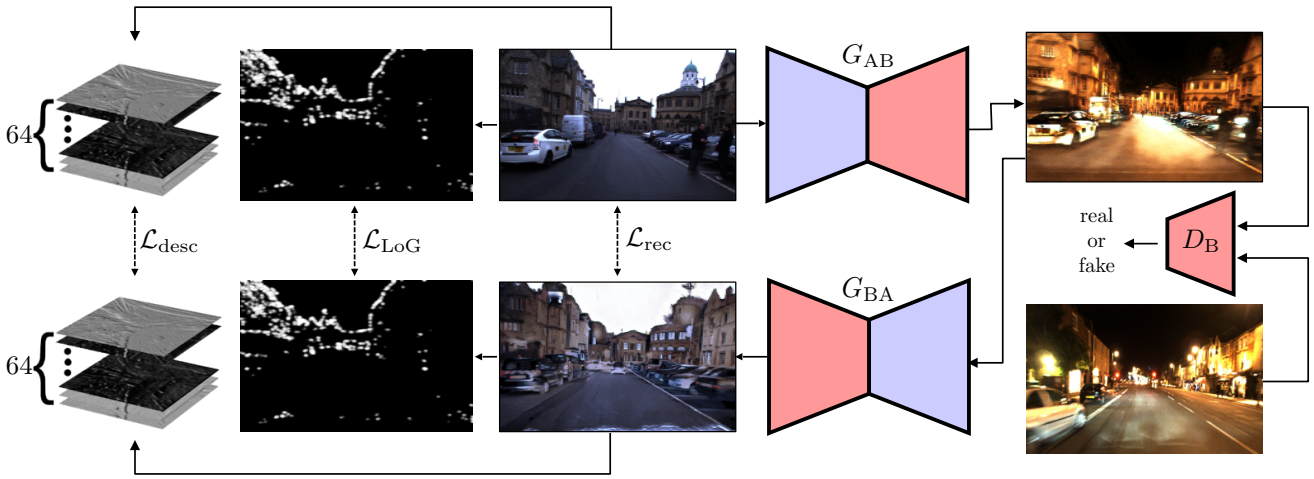


Fig. 2. Cycle-consistency architecture. This architecture is employed in stage 1 of the training process, and trains a pair of generators to transfer appearance from source to target images, and vice-versa, without requiring registration of images.

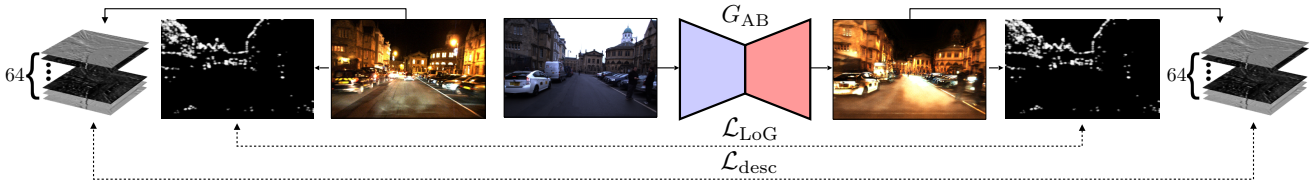


Fig. 3. Fine-tuning architecture. This architecture is employed in stage 2 of the training process on a well-aligned subset of the data in order to minimise the difference between feature detector and descriptor layers between different conditions. Note: discriminator is omitted from figure for compactness.

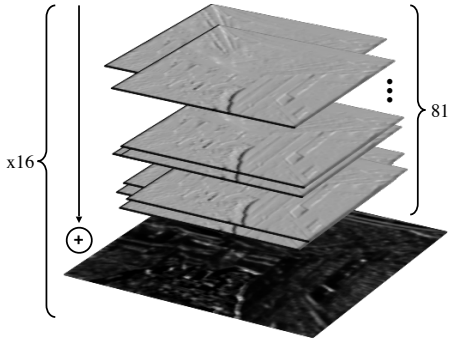


Fig. 4. HAAR response stack. For each of the 16 neighbourhood regions that compose a SURF descriptor for a specific scale, and for each of the X and Y directions, we stack 81 copies of the image convolved with the corresponding X- or Y-direction HAAR box filter, offset them according to a precomputed look-up table and multiply each response with a precomputed weight. We separately sum, along the stacking axis, the response maps and the absolute values of the response maps, for each direction, yielding 4x16 stacked matrices which represent the dense SURF descriptor.

modules. The internal architecture of the generator is shown in Fig. 5.

The discriminator architecture is a CNN with 5 layers. The first 4 layers are comprised of a convolution operation followed by instance normalisation and leaky ReLu, and the last layer is a convolution operation which outputs a $H/8 \times W/8$ map classifying the receptive field in the image space as real or fake, where H and W represent the height and width of the input image. Examples of generator results for both G_{AB} and G_{BA} are shown in Fig. 6 for a range of different condition pairs.

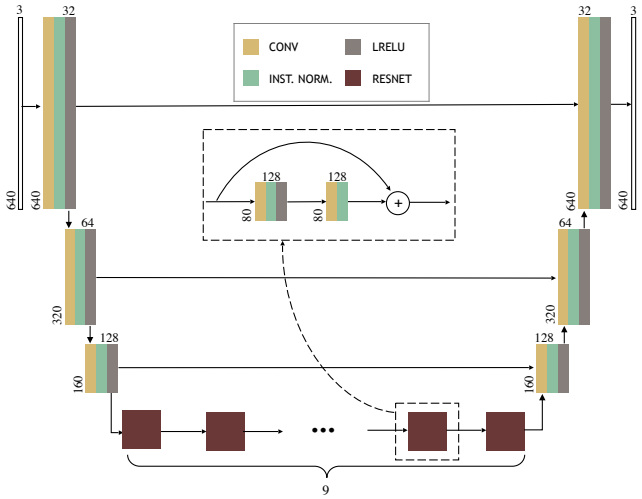


Fig. 5. Generator internal architecture. We employ 3 down-convolutional layers with stride 2, 9 ResNet blocks and 3 up-convolutional layers with fractional stride $1/2$, with skip connections between corresponding down- and up-convolutional layers. Each convolutional layer consists of a convolution operation, followed by instance normalisation and leaky ReLU. Each ResNet block consists of a convolution, followed by instance normalisation, leaky ReLU, a second convolution, instance normalisation and addition of the original block input to the resulting output.

A. SURF Detector Response Map

The SURF detector response map is obtained using a convolutional version of the original method of approximating the determinants of Hessians described in [22]. For each scale we generate three box filters to approximate the second-



Fig. 6. Example appearance transfer between different conditions using the cycle-consistency generators G_{AB} and G_{BA}. Each condition pair (Day/Night, Day/Dawn, Day/Rain, Day/Snow) produces a pair of generators, allowing us to perform bidirectional appearance transfer trained using only unordered collections of images from each condition.

order derivative of the gaussian $\frac{\partial^2}{\partial x^2}g(\sigma)$, $\frac{\partial^2}{\partial y^2}g(\sigma)$, $\frac{\partial^2}{\partial x \partial y}g(\sigma)$ on the X,Y and diagonal directions respectively. We convolve these filters with the image I, yielding the response maps $L_{\mathrm{xx}}(\sigma)$, $L_{\mathrm{yy}}(\sigma)$ and $L_{\mathrm{xy}}(\sigma)$.

Using the Hadamard product, the matrix of approximations of the determinant of Hessians is:

$$\det(\mathcal{H}_{\mathrm{approx}}) = L_{\mathrm{xx}} \circ L_{\mathrm{yy}} - 0.81 * L_{\mathrm{xy}} \circ L_{\mathrm{xy}} \qquad (1)$$

B. Dense SURF Descriptors

We adapt the methodology used in OpenSURF [26] and employ a fast, convolutional method for building dense per-pixel SURF descriptors, through which gradients can be passed. For each scale out of the N chosen scales, we precompute:

- a look-up table for the 81 relative offsets of pixel neighbours that are used to build SURF descriptors
- an $N \times 81$-matrix for the scale-specific Gaussian weights of the 81 offsets
- a 16-length column vector for the Gaussian weights of the 16 neighbourhoods
- HAAR-like box filters for both the X and Y directions

We then convolve the input image with the HAAR box filters and store the wavelet responses. For each chosen scale we stack 81 copies of the wavelet responses and multiply them with the scale-specific Gaussian weight.

Then, for each of the 16 pixel neighbourhoods that make up a SURF descriptor we:

- offset the stacked copies along the X and Y directions according to the offset look-up table
- multiply by the neighbourhood-specific Gaussian weight
- sum along the stack direction both the raw and absolute values for the X and Y-directions respectively, yielding 4 matrices.
- element-wise multiply each matrix with its specific Gaussian neighbourhood weight LUT
- stack the 4 resulting matrices

Finally, we normalize each column of the resulting 64 layer stack of $H \times W$ size matrices, where H and W are the height and width of the input images. This stack represents the dense per-pixel SURF descriptor, for each scale. The stacking and summing operation is depicted in Fig. 4.

Feature detectors and descriptors for day-night matching

are evaluated in [27], showing that most features are detected at small scales (< 10). Following our own experiments, we only compute SURF loss terms for the first 5 scales in order to speed-up the training process, without noticing any significant performance loss. An explanation for this could be that inside smaller pixel neighbourhoods, appearance changes between images with different conditions can be more uniform compared to larger neighbourhoods.

C. Losses

Similar to [7], we apply an adversarial loss through a discriminator on the output of each generator: discriminator D_{B} on the output of generator G_{AB}, and discriminator D_{A} on the output of generator G_{BA}. This loss is formulated as:

$$\mathcal{L}_{\mathrm{B}_{\mathrm{adv}}} = (D_{\mathrm{B}}(G_{\mathrm{AB}}(I_{\mathrm{A}})) - 1)^2 \quad (2)$$

$$\mathcal{L}_{\mathrm{A}_{\mathrm{adv}}} = (D_{\mathrm{A}}(G_{\mathrm{BA}}(I_{\mathrm{B}})) - 1)^2 \quad (3)$$

The adversarial objective $\mathcal{L}_{\mathrm{adv}}$ becomes:

$$\mathcal{L}_{\mathrm{adv}} = \mathcal{L}_{\mathrm{B}_{\mathrm{adv}}} + \mathcal{L}_{\mathrm{A}_{\mathrm{adv}}} \quad (4)$$

The discriminators are trained to minimize the following loss:

$$\mathcal{L}_{\mathrm{B}_{\mathrm{disc}}} = (D_{\mathrm{B}}(I_{\mathrm{B}}) - 1)^2 + (D_{\mathrm{B}}(G_{\mathrm{AB}}(I_{\mathrm{A}})))^2 \quad (5)$$

$$\mathcal{L}_{\mathrm{A}_{\mathrm{disc}}} = (D_{\mathrm{A}}(I_{\mathrm{A}}) - 1)^2 + (D_{\mathrm{A}}(G_{\mathrm{BA}}(I_{\mathrm{B}})))^2 \quad (6)$$

The discriminator objective $\mathcal{L}_{\mathrm{disc}}$ becomes:

$$\mathcal{L}_{\mathrm{disc}} = \mathcal{L}_{\mathrm{B}_{\mathrm{disc}}} + \mathcal{L}_{\mathrm{A}_{\mathrm{disc}}} \quad (7)$$

The cycle consistency loss [7] is applied between the input and reconstructed image, and between the SURF detector $Det(\cdot)$ and dense descriptor $Desc(\cdot)$ maps computed from these two images:

$$\mathcal{L}_{\mathrm{rec}} = \|I_{\mathrm{input}} - I_{\mathrm{reconstructed}}\|_1 \quad (8)$$

$$\mathcal{L}_{\mathrm{LoG}} = \|Det(I_{\mathrm{input}}) - Det(I_{\mathrm{reconstructed}})\|_1 \quad (9)$$

$$\mathcal{L}_{\mathrm{desc}} = \|Desc(I_{\mathrm{input}}) - Desc(I_{\mathrm{reconstructed}})\|_1 \quad (10)$$

The complete generator objective $\mathcal{L}_{\mathrm{gen}}$ becomes:

$$\mathcal{L}_{\mathrm{gen}} = \lambda_{\mathrm{rec}} * \mathcal{L}_{\mathrm{rec}} + \lambda_{\mathrm{LoG}} * \mathcal{L}_{\mathrm{LoG}} + \lambda_{\mathrm{desc}} * \mathcal{L}_{\mathrm{desc}} + \lambda_{\mathrm{adv}} * \mathcal{L}_{\mathrm{adv}} \quad (11)$$

Each λ term is a hyperparameter that weights the influence of each loss component. For the fine-tuning stage, where the target image is aligned with the input and synthetic images, the losses become:

$$\mathcal{L}_{\mathrm{F}_{\mathrm{LoG}}} = \|Det(I_{\mathrm{target}}) - Det(I_{\mathrm{synthetic}})\|_1 \quad (12)$$

$$\mathcal{L}_{\mathrm{F}_{\mathrm{desc}}} = \|Desc(I_{\mathrm{target}}) - Desc(I_{\mathrm{synthetic}})\|_1 \quad (13)$$

The fine-tuning objective $\mathcal{L}_{\mathrm{finetune}}$ becomes:

$$\mathcal{L}_{\mathrm{finetune}} = \lambda_{\mathrm{LoG}} * \mathcal{L}_{\mathrm{F}_{\mathrm{LoG}}} + \lambda_{\mathrm{desc}} * \mathcal{L}_{\mathrm{F}_{\mathrm{desc}}} \quad (14)$$

We wish to find the generator functions G_{AB}, G_{BA} such that:

$$G_{\mathrm{AB}}, G_{\mathrm{BA}} = \underset{G_{\mathrm{AB}}, G_{\mathrm{BA}}, D_{\mathrm{B}}, D_{\mathrm{A}}}{\arg\min} \mathcal{L}_{\mathrm{gen}} + \mathcal{L}_{\mathrm{disc}} + \mathcal{L}_{\mathrm{finetune}} \quad (15)$$

In the following section we describe how the network is trained to minimise the above losses.

IV. Experimental Setup

We chose 6 traversals from the Oxford RobotCar Dataset [11] collected up to 1 year apart, yielding 5 condition pairs: day-night, day-snow, day-dawn, day-sun and day-rain. For each of the traversals, the RTK-GPS ground truth was filtered and any data points with more than 25cm of translation standard deviation were discarded.

Training datasets for each condition pair were created from the entirety of the day traversal and a portion representing approximately 20% of the paired condition, to simulate a case where reasonable amounts of mapping data cannot be acquired. The remaining 80% of the paired condition was used to benchmark the performance of the synthetic images.

The well-aligned datasets used in the second training stages were created by selecting image pairs between which none or only a small viewpoint rotation existed. Image pairs with no translation or rotation misalignment were used as-is, and for those with small rotational differences the target images were affine-warped into the frame of the source images using the known poses provided by the RTK-GPS ground truth.

A. Training

For the cycle-consistency stage, we used a network training regimen similar to [7]. For each iteration we first trained the discriminator on a real target domain image and a synthetic image from a previous iteration with the goal of minimizing $\mathcal{L}_{\mathrm{disc}}$, and then trained the generators on input images to minimize $\mathcal{L}_{\mathrm{gen}}$. We used the Adam solver [28] with an initial learning rate set at 0.0002, a batch size of 1, $\lambda_{\mathrm{rec}} = 8$, $\lambda_{\mathrm{det}} = 2$, $\lambda_{\mathrm{desc}} = 2$ and $\lambda_{\mathrm{adv}} = 1$. For the fine-tuning stage, we trained on a small well-aligned subset of the dataset, minimizing $\mathcal{L}_{\mathrm{finetune}}$, with the same learning parameters.

B. Localisation

We used the trained generators G_{AB} to transform all of the day map frames into target-condition frames, and G_{BA} to transform the 5 types of target-condition frames into day-condition frames. To benchmark the synthetic images in the context of localisation, we used the experience-based navigation system of [3] which implements a feature-based topological localiser [29] followed by a geometric verification stage using RANSAC [30] and a nonlinear optimisation to minimise inlier reprojection errors. In contrast to adding the synthetic frames as a separate map, it is possible to accumulate feature correspondences from real to real image matching and synthetic to real image matching, leading to more robust and accurate solutions. The generator runs at approximately 1 Hz for images with a resolution of 1280×960, and at approximately 3 Hz for images with a resolution of 640×480 on an Nvidia Titan X GPU.

V. Results

A. Quantitative results

We benchmark our results taking into consideration both the frequency and the quality of localisation. Table I compares the root mean squared translation and rotation errors,

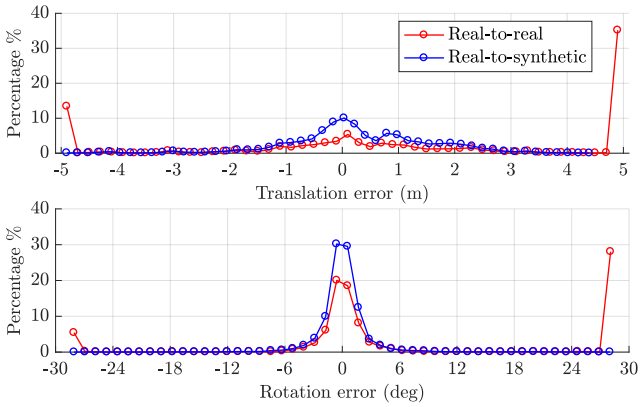


Fig. 7. Real day to real night localisation vs. Real day to synthetic day localisation : Histogram of translation and rotation error from ground truth. Translation outliers larger than 5 meters in absolute have been accumulated in the -5 and +5 meter bins. Rotation outliers larger than 30 degrees in absolute have been accumulated in the -30 and +30 degree bins.

with respect to the RTK-GPS ground truth, and the cumulative successfully localized portion as a percentage of the distance travelled in the case of day-night localisation. Results are shown for raw images, images obtained with an RGB-only implementation of [7], and images obtained using stages 1 and 2 of our solution. The results show a significant increase in the accuracy of localisation using synthetic images generated from the stage-1 model, and a further increase in accuracy from the stage-2 fine-tuned model.

Table II presents localisation results for a wide range of conditions transformed into day using stage-1 trained models, illustrating the performance of the method when localising relative to a single condition. In all cases the localisation rate is improved (often by a factor of 2) and the metric errors are reduced.

Fig. 7 shows the distribution of translation and rotation errors with respect to ground truth, for the raw image matching and for our best solution, in the case of day-night localisation. Overall we observe a large improvement in localisation accuracy with our solution, compared to raw images and images produced by an RGB-only implementation of [7].

Fig. 8 shows the number of match inliers as a function of the distance travelled, for the raw images and for our best solution, in the case of day-night localisation. We observe a significant increase in the number of inliers for real-to-synthetic matching, compared to real-to-real image matching.

Fig. 9 shows the probability of travelling a certain distance without localisation against the map (VO-based open-loop or dead-reckoning) when a localisation failure occurs. We observe a significant reduction in the distance to re-localisation against the map when using synthetic images generated using our solution. Surprisingly, images generated using an RGB-only implementation of [7] did not bring a large improvement in robustness.

B. Qualitative results

We present qualitative results in a series of locations throughout Oxford where matching between raw images

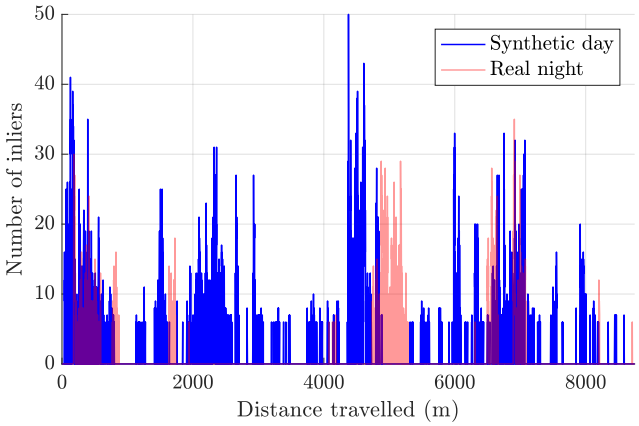


Fig. 8. Inlier count for real day vs. real night localisation and real day vs. synthetic day as a function of distance travelled.

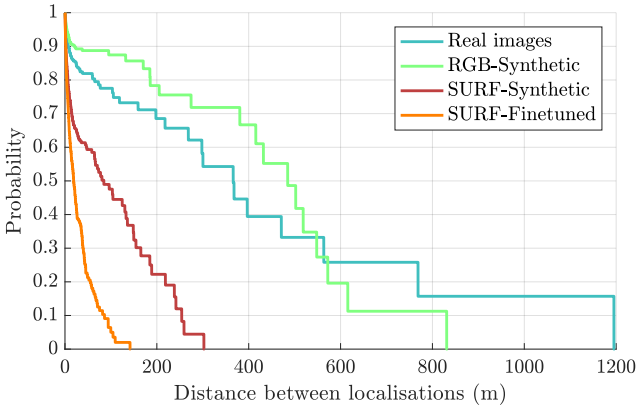


Fig. 9. Probability of travelling a certain distance without localisation against the map (VO-based open-loop or dead-reckoning) when a localisation failure occurs. We observe a large increase in the robustness of day-night localisation when using synthetic images. The map is from daytime, the input images are at night.

failed or produced a very small number of inliers. Fig. 10 presents matches between real images (top) and between real and synthetic images (bottom). Note how the learned image transformation (by construction) does a qualitatively good job reconstructing details that are described by the feature detector and descriptor, such as window frames.

VI. Conclusions

We have presented a system that yields robust localisation under very adverse conditions. The key to this work is transforming an input image in such a way as to enhance point wise matching to a stored image. This transformation is learnt using a cyclic GAN while explicitly accounting for the attributes feature detection and description stages. We motivated this extra-complexity by showing quantitatively that it is not sufficient to simply learn a transfer function for pixel-wise RGB values - one needs to account for feature detector and descriptor responses. Using modest target training data, which emulates scenarios where mapping is expensive, time-consuming or difficult, the resulting synthetic images consistently improved place recognition and metric localisation compared to baselines. This speaks to not only drastically

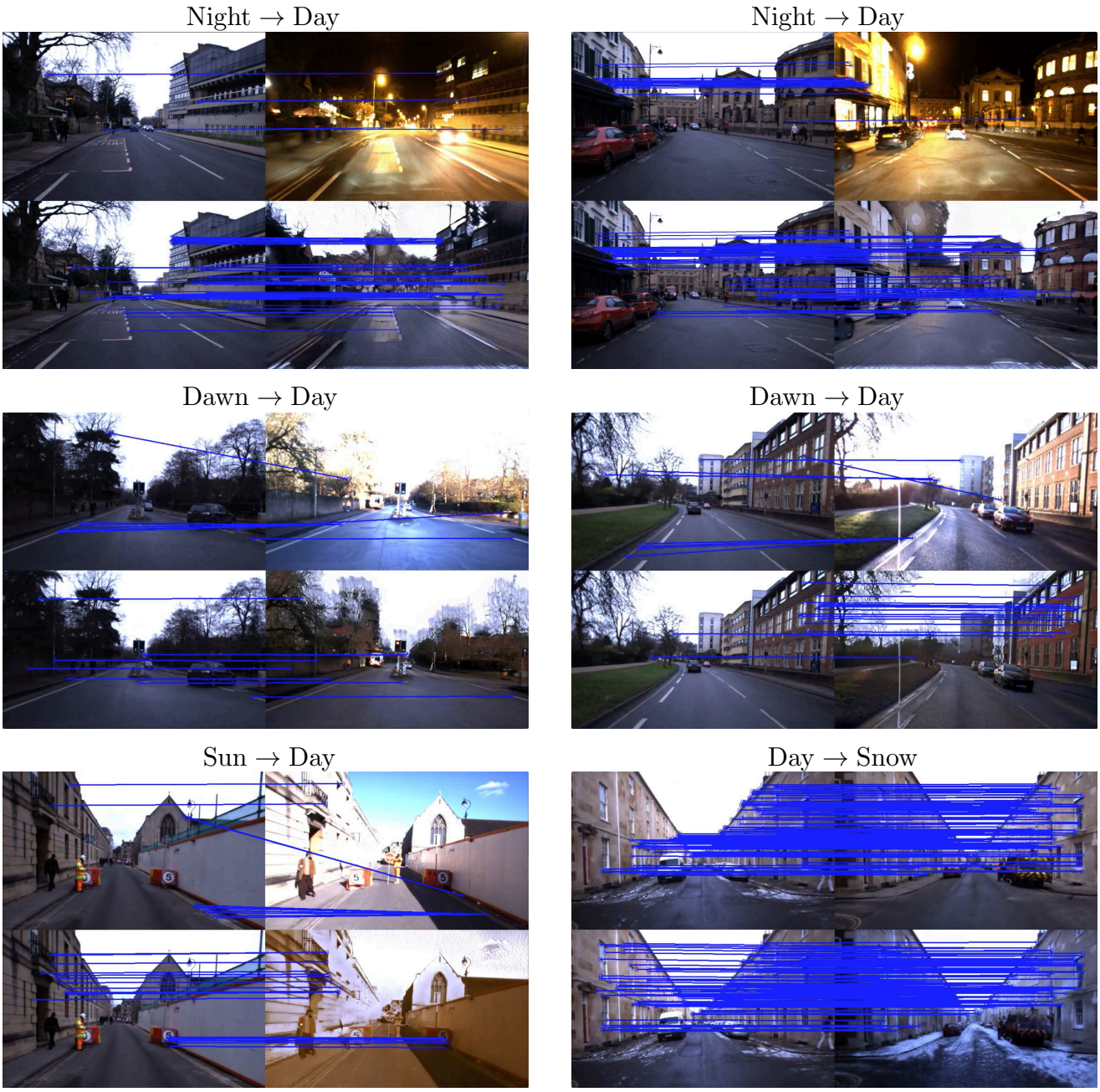


Fig. 10. Example feature-based localisation between different conditions using appearance transfer. In all cases our approach provides a greater number of inlier matches (blue) when using the transformed image compared to matching the original images captured under significantly different conditions. Top left and bottom left represent the real-day map image. Top right represents the original live camera frame. Bottom right represents the live camera frame transformed to match the condition of the map image. These results are obtained using stage-1 of our architecture.

Map vs. Traversal	**REFERENCE**			**RGB-only**			**SURF**			**SURF + finetune**		
	%	RMSE (m)	RMSE (°)	%	RMSE (m)	RMSE (°)	%	RMSE (m)	RMSE (°)	%	RMSE (m)	RMSE (°)
Real Day vs Real Night	19.5	12.74	24.06	—	—	—	—	—	—	—	—	—
Real Day vs Synthetic Day	—	—	—	25	3.59	11.38	51	2.43	4.06	60	1.99	3.32

TABLE I

METRIC LOCALISATION PERFORMANCE FOR DIFFERENT NETWORK ARCHITECTURES

Map vs. Traversal	REFERENCE			SYNTHETIC		
	%	RMSE (m)	RMSE (°)	%	RMSE (m)	RMSE (°)
Day vs Night	19.5	12.74	24.06	51.0	2.43	4.06
Day vs Snow	94.6	9.02	8.07	98.6	3.38	8.19
Day vs Dawn	31.9	11.59	7.04	66.8	2.55	6.30
Day vs Sun	33.0	32.35	82.04	71.0	9.40	24.52
Day vs Rain	27.2	11.28	9.85	58.6	2.91	7.84

TABLE II
METRIC LOCALISATION PERFORMANCE BETWEEN CONDITIONS

reducing the cost and inconvenience of mapping under diverse conditions, but also improving the efficacy of the maps produced when used in conjunction with our method. Finally, an architectural nicety is that our system processes the image streams outside of the localisation pipeline, either offline or online, and hence can be used naturally as a front end to many existing systems.

A video describing our results can be found at `https://www.youtube.com/watch?v=s8XV1Y6opig`

VII. Future work

Future work could involve a system that can deal with continuous changes in illumination and weather conditions, rather than binning them into discrete categories, and ideally learning a single generator model in order to accomplish this.

VIII. Acknowledgements

This work was supported by Oxford-Google DeepMind Graduate Scholarships and Programme Grant EP/M019918/1.

References

[1] C. Linegar, W. Churchill, and P. Newman, "Made to measure: Bespoke landmarks for 24-hour, all-weather localisation with a camera," in *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 2016-June, 2016, pp. 787–794.

[2] C. McManus, B. Upcroft, and P. Newman, "Learning place-dependant features for long-term vision-based localisation," *Autonomous Robots*, vol. 39, no. 3, pp. 363–387, 2015.

[3] W. Churchill and P. Newman, "Experience-based navigation for long-term localisation," *The International Journal of Robotics Research*, vol. 32, no. 14, pp. 1645–1661, 2013.

[4] R. Mur-Artal and J. D. Tardós, "ORB-SLAM2: an open-source SLAM system for monocular, stereo and RGB-D cameras," *CoRR*, vol. abs/1610.06475, 2016.

[5] P. Furgale and T. D. Barfoot, "Visual teach and repeat for long-range rover autonomy," *Journal of Field Robotics*, vol. 27, no. 5, pp. 534–560, may 2010.

[6] C. Linegar, W. Churchill, and P. Newman, "Work smart, not hard: Recalling relevant experiences for vast-scale but time-constrained localisation," in *Proceedings - IEEE International Conference on Robotics and Automation*, vol. 2015-June, no. June, 2015, pp. 90–97.

[7] J. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," *CoRR*, vol. abs/1703.10593, 2017.

[8] W. Yan, X. Shi, X. Yan, and L. Wang, "Computing OpenSURF on OpenCL and general purpose GPU," *International Journal of Advanced Robotic Systems*, vol. 10, 2013.

[9] T. B. Terriberry, L. M. French, and J. Helmsen, "Gpu accelerating speeded-up robust features," in *Proceedings of 3DPVT*, vol. 8, 2008, pp. 355–362.

[10] W. W.-C. Chiu and M. Fritz, "See the difference: Direct pre-image reconstruction and pose estimation by differentiating hog," *2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 468–476, 2015.

[11] W. Maddern, G. Pascoe, C. Linegar, and P. Newman, "1 Year, 1000km: The Oxford RobotCar Dataset," *The International Journal of Robotics Research (IJRR)*, vol. 36, no. 1, pp. 3–15, 2017.

[12] A. Torii, R. Arandjelović, J. Sivic, M. Okutomi, and T. Pajdla, "24/7 place recognition by view synthesis," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 07-12-June, 2015, pp. 1808–1817.

[13] S. Cascianelli, G. Costante, E. Bellocchio, P. Valigi, M. L. Fravolini, and T. A. Ciarfuglia, "Robust visual semi-semantic loop closure detection by a covisibility graph and CNN features," *Robotics and Autonomous Systems*, vol. 92, pp. 53–65, 2017.

[14] E. Johns and G.-Z. Yang, "Feature co-occurrence maps: Appearance-based localisation throughout the day," in *Robotics and Automation (ICRA), 2013 IEEE International Conference on*. IEEE, 2013, pp. 3212–3218.

[15] P. Neubert, N. Sunderhauf, and P. Protzel, "Appearance change prediction for long-term navigation across seasons," in *Mobile Robots (ECMR), 2013 European Conference on*. IEEE, 2013, pp. 198–203.

[16] P. Mühlfellner, M. Bürki, M. Bosse, W. Derendarz, R. Philippsen, and P. Furgale, "Summary Maps for Lifelong Visual Localization," *Journal of Field Robotics*, vol. 33, no. 5, pp. 561–590, aug 2016.

[17] G. Yu and J.-M. Morel, "Asift: An algorithm for fully affine invariant comparison," *Image Processing On Line*, vol. 1, pp. 11–38, 2011.

[18] R. Gomez-Ojeda, Z. Zhang, J. Gonzalez-Jimenez, and D. Scaramuzza, "Learning-based image enhancement for visual odometry in challenging hdr environments," *arXiv preprint arXiv:1707.01274*, 2017.

[19] Y. Shih, S. Paris, F. Durand, and W. T. Freeman, "Data-driven hallucination of different times of day from a single outdoor photo," *ACM Trans. Graph.*, vol. 32, no. 6, pp. 200:1–200:11, Nov. 2013.

[20] L. A. Gatys, A. S. Ecker, and M. Bethge, "Image style transfer using convolutional neural networks," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[21] M. Liu, T. Breuel, and J. Kautz, "Unsupervised image-to-image translation networks," *CoRR*, vol. abs/1703.00848, 2017.

[22] H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: Speeded up robust features," *Computer vision–ECCV 2006*, pp. 404–417, 2006.

[23] R. Guerrero, C. Qin, O. Oktay, C. Bowles, L. Chen, R. Joules, R. Wolz, M. Valdes-Hernandez, D. Dickie, J. Wardlaw, *et al.*, "White matter hyperintensity and stroke lesion segmentation and differentiation using convolutional neural networks," *arXiv preprint arXiv:1706.00935*, 2017.

[24] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2015, pp. 234–241.

[25] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[26] C. Evans, "Notes on the OpenSURF Library," *University of Bristol Tech Rep CSTR09001 January*, no. 1, p. 25, 2009.

[27] H. Zhou, T. Sattler, and D. W. Jacobs, "Evaluating local features for day-night matching," in *Computer Vision - ECCV 2016 Workshops - Amsterdam, The Netherlands, October 8-10 and 15-16, 2016, Proceedings, Part III*, 2016, pp. 724–736.

[28] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2014.

[29] M. Cummins and P. Newman, "Appearance-only slam at large scale with fab-map 2.0," *The International Journal of Robotics Research*, vol. 30, no. 9, pp. 1100–1123, 2011.

[30] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, jun 1981.

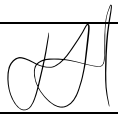
Statement of Authorship for joint/multi-authored papers for PGR thesis

To appear at the end of each thesis chapter submitted as an article/paper

The statement shall describe the candidate's and co-authors' independent research contributions in the thesis publications. For each publication there should exist a complete statement that is to be filled out and signed by the candidate and supervisor **(only required where there isn't already a statement of contribution within the paper itself).**

Title of Paper	Adversarial training for adverse conditions: Robust metric localisation using appearance transfer
Publication Status	Published
Publication Details	**H. Porav**, W. Maddern and P. Newman, "Adversarial Training for Adverse Conditions: Robust Metric Localisation Using Appearance Transfer," *2018 IEEE International Conference on Robotics and Automation (ICRA)*, Brisbane, QLD, 2018, pp. 1011-1018, doi: 10.1109/ICRA.2018.8462894.

Student Confirmation

Student Name:	Horia Porav		
Contribution to the Paper	All work except advice and editorial changes. My contributions included: - Initial research and generation of ideas - Implementation of models and of all software - Data processing and gathering - Data transformation, validation and curation - Data alignment and association for traversals under varying conditions - All experiments and analysis - Model curation and hyperparameter experimentation - Development of a differentiable SURF-based loss - Writing the paper, complete with diagrams, tables and graphs - Presenting the paper at ICRA 2018 - Creating the multimedia presentation and video for the paper		
Signature		Date	03 October 2020

Supervisor Confirmation

By signing the Statement of Authorship, you are certifying that the candidate made a substantial contribution to the publication, and that the description described above is accurate.

Supervisor name and title: Professor Paul Newman			
Supervisor comments			
Signature		Date	05-10-2020

This completed form should be included in the thesis, at the end of the relevant chapter.

3.2 Don't Worry About the Weather: Unsupervised Condition-Dependent Domain Adaptation

Problem: Current state-of-the-art semantic segmentation techniques produce good results with day-time, clean, overcast inputs. However, performance degrades significantly with changes in illumination or weather and in the presence of noise. Labelled data for non-ideal input conditions is scarce and difficult to obtain, which makes training models on such inputs hard.

Solution: Rather than training a segmentation model with multiple conditions, we train light-weight convolutional per-condition image adapters that transform the condition or appearance of the input image to one that maximizes the performance of the semantic segmentation model on that particular condition. We do not use a direct pixel-wise loss on the resulting adapted image, instead relying completely on the segmentation loss. The multi-condition dataset is synthetically generated from a daytime, overcast dataset using cycle-consistency GAN models, while the label ground-truths are approximated by running a state-of-the-art segmentation model on the reference, daytime dataset, making the method self-supervised.

Status: Published at ITSC2019

Don't Worry About the Weather: Unsupervised Condition-Dependent Domain Adaptation

Horia Porav, Tom Bruls and Paul Newman

***Abstract*— Modern models that perform system-critical tasks such as segmentation and localization exhibit good performance and robustness under ideal conditions (i.e. daytime, overcast) but performance degrades quickly and often catastrophically when input conditions change. In this work, we present a domain adaptation system that uses light-weight input adapters to pre-processes input images, irrespective of their appearance, in a way that makes them compatible with off-the-shelf computer vision tasks that are trained only on inputs with ideal conditions. No fine-tuning is performed on the off-the-shelf models, and the system is capable of incrementally training new input adapters in a self-supervised fashion, using the computer vision tasks as supervisors, when the input domain differs significantly from previously seen domains. We report large improvements in semantic segmentation and topological localization performance on two popular datasets, RobotCar and BDD.**

I. Introduction

Robust Computer Vision is paramount to the prevalence of general-purpose robotics, and even more so in fields such as autonomous transportation, where failures may be catastrophic. Modern models that perform system-critical tasks such as segmentation, detection, localization and classification - either traditional heuristics-based or learned - exhibit good performance and robustness under ideal conditions (i.e. daytime, overcast) but performance degrades quickly and often catastrophically when input conditions change. To have their breakthrough, real-world systems must work under varying illumination, weather and noise conditions, and in the long term will need the ability to adapt to new, unseen domains without explicit supervision.

As a type of domain adaptation technique, domain unification is the holy grail of visual perception, theoretically allowing models trained on samples with limited heterogeneity to perform adequately on scenes that are well out of the distribution of the training data. Domain unification can be applied within the vast distribution of natural images [1], [2], [3], between natural and synthetic images (computer-generated, whether through traditional 3D rendering or more modern GAN-based techniques) [4], [5] and even between different sensor modalities [6]. Additionally, domain unification can be implemented at different stages of a computer vision pipeline, ranging from direct approaches such as domain confusion [7], [8], [9], fine-tuning models on target domains [1] or mixture-of-expert approaches [10], etc.

However, a major limiting factor for all these approaches is the scarcity of labelled multi-modal data, driven by the high cost of manual labelling. Most approaches attempt to solve this shortcoming by using 3D-rendered simulations that programatically provide ground-truth [11], [12], or by using unsupervised techniques that adapt models based on auxiliary or proxy tasks [7], [8], [9], [13], [1].

Authors are from the Oxford Robotics Institute, University of Oxford, UK. {horia, tombruls, pnewman}@robots.ox.ac.uk



Fig. 1. Our method allows off-the-shelf models to work with new, unseen domains, without any specific fine-tuning. In this example, we use our input adapter to allow a segmentation model trained using only daytime examples to work under night-time conditions. Top-left quadrant is the input night-time image, top-right quadrant is the output of our input adapter. The bottom-left quadrant is the output of the segmentation model applied on the original night-time input image. The bottom-right quadrant is the output of the segmentation model applied on the output of our adapter. The initial segmentation result is unusable, while the result obtained by running the model on the output of our domain adaptation pipeline accurately classifies roads, pavement, pedestrians, bicycles, vegetation and buildings.

We propose a hybrid method, where multi-modal data is generated in an unsupervised fashion with approximated, high-quality ground truth, followed by supervised training of domain-adapters, for a battery of computer vision tasks, using this generated data and approximated ground truth. While this final step is supervised, the data used for supervision is itself created in an unsupervised fashion, making the entire pipeline unsupervised. To do so, we start by generating multi-modal training data: from a database of image sequences categorized using the time of day and weather conditions at their moment of recording, we select a daytime, overcast, clear **reference** sequence. We leverage the fact that modern computer-vision pipelines perform excellent (e.g. $\geqslant$ 0.83 mIOU on the Cityscapes multi-class segmentation validation set [14]) on inputs with ideal conditions, and thus we run this **reference** sequence through a set of off-the-shelf computer vision tasks and save the outputs as *approximated* ground truth - these results are not 100% identical to the real-world ground truth but they approximate it very well.

Secondly, we apply style transfer to the **reference** sequence in order to produce a set of sequences which are structurally and geometrically identical to the **reference** se-

quence but differ in appearance. The applied style is sourced from many other sequences, each possessing a different appearance. This step is key to our approach - retaining the structure and geometry of the **reference** sequence while varying appearance means that the *approximated* ground truth data is still valid: as an example, the same car but with varying appearances - once during daytime and once during nighttime - will still have the same ground truth footprint in a semantic segmentation map. At the end of these two steps we will have produced - in an unsupervised fashion - a set of sequences with varying appearances accompanied by task-specific ground truth data.

In step three, instead of training condition-dependent, separate task-specific models (e.g. a segmentation model for day-time, one for night-time etc.), we train lightweight condition-specific image adapters that are then used with vanilla, off-the-shelf task-specific models. The motivation for this is simple, but important: models that are invariant to input distributions are notoriously difficult to architect, parametrize and learn, while multi-model approaches such as mixture-of-experts do not scale well with the variance of the input distributions due to memory and runtime constraints. We tackle both these problems by training small, lightweight condition-specific convolutional input adapters, while a classifier-supervisor chooses the best adapter to be run, dependent on the distribution of the inputs. This approach adds minimal overhead to any off-the shelf task, benefits from parameter-counts that do not depend on the variances of the input distributions, and lends itself well to online learning of new conditions. Additionally, in contrast to the larger, task-specific models, the image adapter models tend to take up very little storage space and runtime memory and can be nearly-instantaneously loaded and re-loaded by the processing pipeline.

The final stage of our approach allows a robot or vehicle to incrementally adapt to a new, unseen domain: if the condition of the input images does not match one that the system has been previously trained on, the unsupervised style transfer pipeline will select a model that is closest to the current condition, clone it, and fine-tune this cloned model to be able to change the style of the **reference** sequence so that it matches the style of the current input images. Afterwards, data generated using this new model will be used to train - in a supervised fashion - an additional condition-specific image adapter that will allow upstream computer vision tasks to perform well on the new input image condition.

We benchmark our approach on two important tasks in computer vision and robotics: semantic segmentation and image retrieval/topological localization. This list is obviously not exhaustive, and the addition of extra supervisory tasks may lead to further improvements in performance.

Our main contributions include:

- Using cycle-consistency GANs to generate multi-condition training data with approximated ground truth for a battery of off-the-shelf computer vision tasks.
- Training input image adaptors by using the off-the-shelf computer vision models to generate a supervisory signal.
- Enabling online learning of new, unseen domains by leveraging the unsupervised data generation pipeline along with domains on which the data generation models have already been trained.
- Showing that training multiple lightweight adapter modules is better than training monolithic computer vision models that are invariant to input distributions.

Our qualitative and quantitative results are presented in section V.

II. Related Work

A. Computer Vision Tasks

Semantic Segmentation: Semantic segmentation is a key task in robotics, and modern approaches exhibit very good performance when input conditions are favourable. Deep convolutional models such as Deeplab V3+ [14], SDN [15] or PSPNet [16] achieve high class-mIOU figures ($\geqslant 80\%$) on benchmarks such as Cityscapes [17], but their performance breaks down fast when their inputs change due to different weather conditions, seasons or times of day. For this work, we chose DeepLab V3+ as the reference model due to its excellent open-source implementation and availability of results on a number of popular benchmarks.

Topological Localization: Similarly, widely used topological localization frameworks such as FABMAP [18], SeqSlam [19] or NetVLAD [20] achieve high recall and precision figures on clear, daytime images or when explicitly matching images with the same condition (e.g. winter-winter matching), but break down when the conditions of the locations to be matched differ. For this work, we chose NetVLAD as the reference topological localizer.

B. Domain Adaptation

Domain Confusion: The most common approaches fall under the umbrella of domain confusion, making use of a discriminator that forces features extracted by an encoder to follow a similar distribution for both a source and a target domain [7], [8], [9], [13], [21], [22]. The downside of these approaches is the lack of a direct loss for the target domain, which limits its upper bound on performance.

Style-Transfer: Other approaches attempt to directly train computer vision models using synthetic data generated via style-transfer, or to directly adapt the input data to the target domain. Notable approaches include those of [4], [5], [23], [3] and [2]. Generally, these methods seem to have the most promise of reducing the domain gap between real and synthetic images, hence our decision to generate training data using the approach of [24].

In [25], a style-transfer pipeline is trained incrementally by feeding the segmentation map, obtained at a previous adaptation step, as an auxiliary input at each incremental step. The downside is that this type of self-supervised approach assumes that high values in the softmax layer (using segmentation as an example) automatically correlate with higher prediction accuracies, whereas we only approximate ground truth labels once, on a reference, high-quality input sequence using models whose accuracies have actually been validated experimentally.

The closest to our approach is [1], where a semantic segmentation network is trained first with a day-time hand-labelled dataset, and then used to predict labels on intermediary datasets recorded at incremental types of twilight. The

twilight images and estimated labels are then used to further fine-tune the segmentation model, which is finally used to segment night-time images. In contrast, our approach only computes approximated labels on the **reference** condition and uses the rest of the conditions as a guide for style transfer. However, the approach of [1] could be used as a drop-in replacement for style transfer in the larger context of our framework. Similarly, [26] trains segmentation models with a mixture of source domain images and synthetic images with the style of an incrementally-shifted target domain. The main difference between these approaches and our work is that instead of directly training or fine-tuning computer-vision tasks, we train lightweight input adapters while using the performance (loss) on these tasks as a supervisory signal.

Additionally, [27] presents an approach where an encoder-decoder is trained to transform the appearance of input images to a reference appearance, but this is only benchmarked on scenes with small changes in appearance, in the context of 6-DOF localization.

C. Online Learning

The authors of [28] present an approach to incremental online domain adaptation, making use of unsupervised training by employing domain confusion at the level of encoder features from both target and source domains, while slowly shifting both domains through a range of incremental appearance changes (e.g. day to night). In our case, the unsupervised regimen is moved to the data-creation stage, with the generated data being used for supervised training of the input adapters, leading to better training stability and better performing models. Additionally, the authors present a method of reducing data storage requirements by approximating the feature distribution of the source data (or reference data, in our case) using a generative model, which could potentially be swapped for the reference sequence in our approach.

Other approaches include map-management for lifelong learning [29] and adaptation on a domain manifold [30]. Our incremental learning pipeline follows the spirit of these works by always choosing to fine-tune a seed model that was initially trained on data from a domain that is close to the current target domain.

D. Expert Systems

Finally, systems [31], [10] exist that attempt to achieve invariance to the input conditions by running multiple sub-models in parallel and combining their outputs using a weighting or gating scheme to yield the desired result. One major issue with this type of approach is that runtime memory and processing power requirements increase linearly with the number of expert components used in parallel. While we also produce a number of input adapters that is proportional to the number of target domains, our classifier will choose only one input adapter to be run, per target domain, leading to a very small computational and memory footprint at runtime.

III. Learning Condition-Dependent Representations

A. Synthetic Multi-Condition Data

The first step in our approach is data generation. From the Oxford Robotcar Dataset [32], we select a **reference** sequence - one that is daytime, clear, overcast - and a number of traversals with difficult conditions - night, rain, snow etc. We use these conditions, along with a cycle-consistency architecture GAN [24], to train generative models that can apply style transfer to the **reference** condition in order to create a number of synthetic sequences that maintain the structure and geometry of the **reference** condition - locations, shapes and topologies of both static and dynamic objects and of the overall scene - but exhibit variation in appearance. In the following paragraphs we offer a succinct introduction to cycle-consistency GANs and how we use them to generate new data. The explanations offered here are in no way exhaustive, and interested readers should refer to the work of [24] for further details.

Following the work of [24], we employ 2 generators: given an image I_{A} from domain A (**reference**) and an image I_{B} from domain B (night, rain, snow etc.), we use generator G_{AB} to translate an image style from domain A to domain B and generator G_{BA} to translate an image style from domain B back into domain A. An adversarial loss is applied on the output of each generator: discriminator D_{B} on the output of generator G_{AB}, and discriminator D_{A} on the output of generator G_{BA}. The adversarial losses are formulated as:

$$\mathcal{L}_{\mathrm{B}_{\mathrm{adv}}} = (D_{\mathrm{B}}(G_{\mathrm{AB}}(I_{\mathrm{A}})) - 1)^2 \tag{1}$$

$$\mathcal{L}_{\mathrm{A}_{\mathrm{adv}}} = (D_{\mathrm{A}}(G_{\mathrm{BA}}(I_{\mathrm{B}})) - 1)^2 \tag{2}$$

The complete adversarial objective to be minimized $\mathcal{L}_{\mathrm{adv}}$ is:

$$\mathcal{L}_{\mathrm{adv}} = \mathcal{L}_{\mathrm{B}_{\mathrm{adv}}} + \mathcal{L}_{\mathrm{A}_{\mathrm{adv}}} \tag{3}$$

We train the discriminators to minimize the following objective:

$$\mathcal{L}_{\mathrm{B}_{\mathrm{disc}}} = (D_{\mathrm{B}}(I_{\mathrm{B}}) - 1)^2 + (D_{\mathrm{B}}(G_{\mathrm{AB}}(I_{\mathrm{A}})))^2 \tag{4}$$

$$\mathcal{L}_{\mathrm{A}_{\mathrm{disc}}} = (D_{\mathrm{A}}(I_{\mathrm{A}}) - 1)^2 + (D_{\mathrm{A}}(G_{\mathrm{BA}}(I_{\mathrm{B}})))^2 \tag{5}$$

The complete discriminator objective to be minimized $\mathcal{L}_{\mathrm{disc}}$ is:

$$\mathcal{L}_{\mathrm{disc}} = \mathcal{L}_{\mathrm{B}_{\mathrm{disc}}} + \mathcal{L}_{\mathrm{A}_{\mathrm{disc}}} \tag{6}$$

A cycle-consistency loss [24] is applied between the reconstructed and input images:

$$\mathcal{L}_{\mathrm{rec}} = \|I_{\mathrm{input}} - I_{\mathrm{reconstructed}}\|_1 \tag{7}$$

The final generator objective $\mathcal{L}_{\mathrm{gen}}$ is:

$$\mathcal{L}_{\mathrm{gen}} = \lambda_{\mathrm{rec}} \cdot \mathcal{L}_{\mathrm{rec}} + \lambda_{\mathrm{adv}} \cdot \mathcal{L}_{\mathrm{adv}} \tag{8}$$

with each λ term representing a hyperparameter that weighs the importance of each individual objective. We want to find the optimal generators G_{AB}, G_{BA} that minimize the complete objective:

$$G_{\mathrm{AB}}, G_{\mathrm{BA}} = \underset{G_{\mathrm{AB}}, G_{\mathrm{BA}}, D_{\mathrm{B}}, D_{\mathrm{A}}}{\arg\min} \mathcal{L}_{\mathrm{gen}} + \mathcal{L}_{\mathrm{disc}} \tag{9}$$

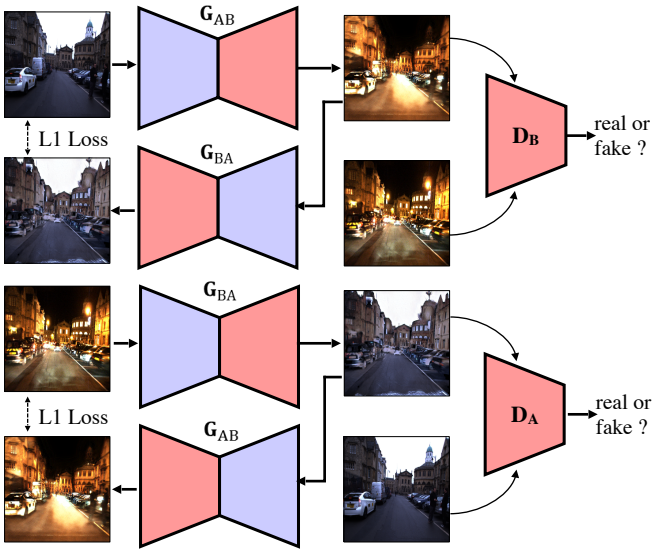


Fig. 2. The overall training architecture of the cycle-consistency GAN used to generate synthetic training data. We follow the training regimen described in [24] for each of the N pairings between the **reference** condition and a **target** condition.

We follow this methodology for N difficult conditions (domain B), always paired with the **reference** condition (domain A), yielding $2N$ generators. However, once the generators have converged, we only use the generators that apply the style of domains B to images from domain A - G_{AB} - to generate N versions of the **reference** sequence, each bearing the appearance of a sequence from domain B. Please note that for brevity we omit the condition-specific subscripts from the equations above. An overview of the CycleGAN architecture is shown if Figure 2.

B. Input Adapters

The second step in our approach is to use the data generated in the previous step to train a bank of adapters that preprocess the input images such that they follow a distribution similar to that of the training sets used to train the bank of tasks. We formulate our input adapters as convolutional encoder-decoders with 3 down-convolutions, a bottleneck with N_{res} ResNet [33] blocks and 3 transpose-convolutions. The input to our adaptors is a 3-channel RGB image, while the output is a 3-channel image compatible with the inputs of many well-known models (semantic segmentation, object detection, depth estimation etc). This configuration provides a light-weight solution that is easy to train using labelled data, with reduced storage requirements and a small run-time memory footprint.

For each input adapter F_{k} (specific to a k^{th} particular appearance), and each task T_m, we formulate the following loss:

$$\mathcal{L}_{T_m} = T_m(F_{\mathrm{k}}(G_{AB_k}(I_A))) - T_m(I_A) \tag{10}$$

where G_{AB_k} is the CycleGAN generator that transforms images from the **reference** condition to the k^{th} condition, and I_A is an input **reference** image. For each input adapter F_{k} specific to condition k out of N conditions, and M tasks, the final objective to be optimized becomes:

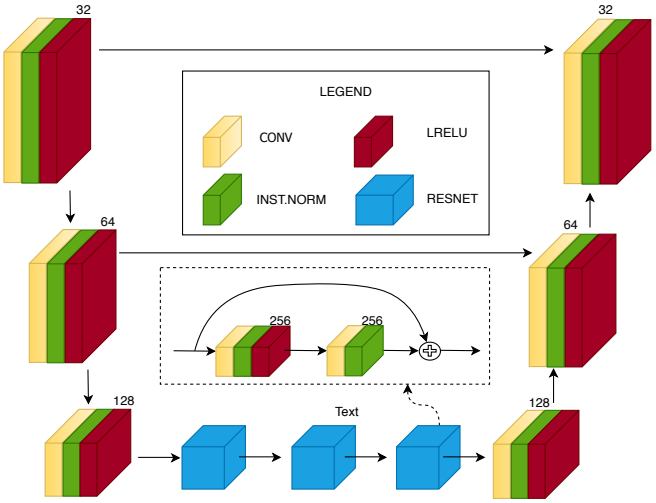


Fig. 3. The overall architecture of our input adapters. Under the assumption that a change in condition should not change the overall structure of the scene, we make use of skip-connections [34] and a ResNet [33] bottleneck to facilitate the direct transfer of features from the input side of the network to the output side.

$$F_{\mathrm{k}} = \arg\min_{F_{\mathrm{k}}} \sum_{m=1}^{M} \alpha_m \cdot \mathcal{L}_{T_m} \tag{11}$$

where α_m is a non-negative weight modulating the importance of each task T_m.

One key takeaway here is that F_{k} is essentially different from G_{BA_k} (the domain-specific generator that maps an image back to the appearance of the **reference** sequence) - we are not directly concerned with obtaining images that possess the appearance of the reference sequence, instead we want to obtain a processed image that maximizes the performance on the set of tasks T_m. This can be observed in Figure 6, second image from left.

C. Domain Classifier

We employ a domain classifier D to select the most suitable input adapter F_k that enables optimal performance on input images with the k^{th} condition. The classifier follows a largely traditional architecture comprised of 4 convolutional layers and 3 fully connected layers followed by a softmax layer, outputting an N-length vector. Given an input image I_A and a domain label t as an N-length one-hot encoding, we wish to find the parameters of the classifier D that minimizes the cross-entropy between the output of the classifier and the target label t:

$$D = \arg\min_{D} -\sum_{k=1}^{N} t_k \cdot \log(D(I_A)_k) \tag{12}$$

with k used to denote the element in each one-hot encoding. After training the classifier with N conditions, we additionally use the output of the penultimate fully-connected layer as a length-128 condition descriptor, which allows us to discriminatively identify domains that are outside of the original N domains used during training. To do this, we average the descriptors over a sequence of input images, and compare to other stored descriptors in the Euclidean space. A detailed explanation of how this is used is presented in Subsection III-E.

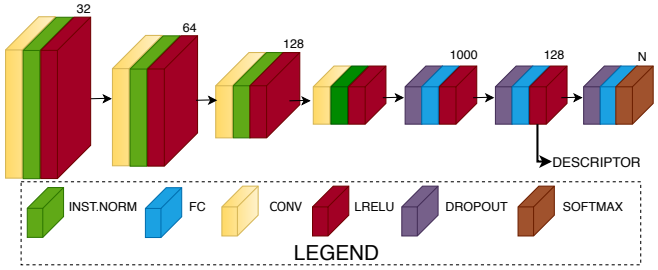


Fig. 4. Our classifier follows a traditional architecture, being composed of a series of down-convolutional layers followed by three fully connected layers. For on-line identification of new, unseen domains, we interpret the output of the penultimate fully-connected layer as a discriminative descriptor. Doing so allows us to identify an arbitrary number of domains, beyond the original N domains, without re-training the classifier.

D. Parameter Memory

After training each of the N condition-specific input adapters F_k parameters (weights) are stored in a memory (database) S. Additionally, the parameters of input adapters fine-tuned following the approach described in Subsection III-E are also stored in the same memory. The classifier described in Subsection III-C is used to select a set of optimal parameters to be used in the input adapter F_k. The memory can be queried in two ways: either by using an index k between 1 and N (the number of initial conditions) or by specifying a length-128 query descriptor and retrieving the set of parameters associated with the descriptor that is closest in the Euclidean space. To enable online learning of unseen domains, we additionally save the parameters of the cycle-consistency GAN generators using the same addressing scheme.

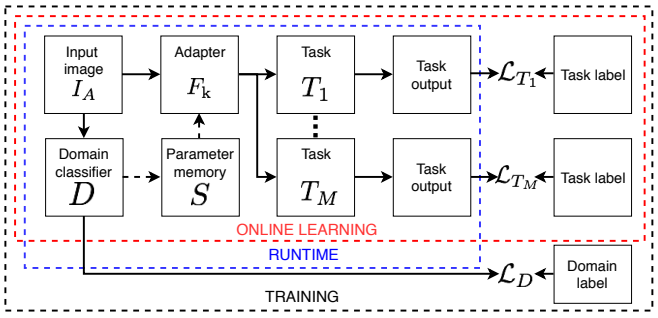


Fig. 5. An overview of our train- and test-time pipeline architecture. Given in input image I_A, the output of the classifier D is used to select a set of parameters for the input adapter F_k. The input adapter is then used to transform the input image into a representation that is better suited for the bank of M tasks T_m. During training, the performance of the tasks on the transformed input image is used as a corrective signal through the set of losses $\mathcal{L}_{T_m}$, along with a domain classification loss $\mathcal{L}_D$ used on the output of the classifier. At runtime only the components contained within the blue dotted rectangle are used. During online learning, the components contained within the red dotted rectangle are used. Solid arrow lines represent differentiable paths, while dotted arrow lines represent non-differentiable paths.

E. Online Learning

The pipeline described in the previous subsections can be extended to incremental, unsupervised, online learning of new, unseen domains without requiring any significant modifications to the existing system. We summarize and outline below the processed used:

- Given a continuous sequence of incoming images, we store the current frame and $T-1$ past frames in a buffer of length T that gets updated using a First-In-First-Out scheme
- For each frame in the buffer, we compute a length-128 condition descriptor using the penultimate layer of the classifier and average all the descriptors, yielding one single length-128 average descriptor
- If this average descriptor condition differs(in Euclidean space) by more than a threshold from the descriptors of any conditions previously trained on (i.e. the parameter memory S is unable to reliably identify the condition), the following training pipeline is triggered:
- We select the cycle-consistency GAN models closest to the current condition(using the condition descriptor), clone and fine-tune them for the current condition using the sequence stored in the buffer
- We use the newly trained generators from above to apply the new style to the **reference** condition to create a new training sequence
- We select the input adapter that is closest to the new condition(again using the descriptor), clone it and train it using the newly created training sequence from above
- We begin using this new adapter in the pipeline until the input condition changes significantly again

In the following section we describe our experimental setup.

IV. Experimental Setup

A. Creating multiple conditions

From the RobotCar Dataset [32], we choose $N = 7$ initial conditions: Snow, Dusk, Night, Night(rain), Night(low exp.), Shadows, Sun(glare) and a **reference** condition with a daytime, overcast condition. Additionally, we choose Sun (with ultrahigh exposure) as a condition not seen during initial training, to be used for online training.

B. Training

For training the cycle-consistency GAN models, we closely follow the approach from [24], and the reader is encouraged to consult the publication for further details. We train each of the initial N condition pairs for 100 epochs, and each online fine-tuning stage for 5 epochs. For training the input adapters, we use the Adam optimizer [35] with a base learning rate of 0.0005, $\beta_1 = 0.9$ and $\beta_2 = 0.999$. We have found that training an initial adapter on the **reference** condition (creating an identity function) and then using the parameters of this adapter as a 'seed' for training the initial N adapters greatly stabilizes and speeds up training. This provides the hint that most parametrisations for different condition adapters lie relatively close to each other on the parameter manifold, partly explaining the relative efficiency of our approach to incremental domain adaptation. We train the input adapters for 20 (offline) or 5 (online) epochs or until performance on the validation split stops increasing, whichever comes first.

C. The tasks

For our particular experiment, we chose $M = 2$ tasks: semantic segmentation and topological localisation, since they represent two critical components for robotics. For the semantic segmentation task, we chose to use DeepLab V3+,

as it produces state of the art results on a number of standard benchmarks [14]. We use a model checkpoint[1] trained on the Cityscapes dataset [17] that achieves 0.83% mIOU on the Cityscapes test split. For topological localisation, we chose the de-facto standard approach of computing place descriptors, NetVLAD [20], and used L2 matching. We use a model checkpoint[2] trained on the Pitts30K dataset [36].

Both of these architectures may be freely swapped with others as long as they are end-to-end differentiable. We set $\lambda = 1$ for semantic segmentation and $\lambda = 10$ for topological localisation, as we have noticed that increasing the importance of the localisation task improves overall performance without affecting the semantic segmentation task.

D. Performance

The input adapter performs inference at approximately 20 Hz for RGB inputs with a size of 640×480, on an Nvidia Titan V GPU. The chosen tasks have independent runtime performances of 3 Hz and 20 Hz for semantic segmentation and topological localisation, respectively.

We benchmark on-line learning by introducing an unseen condition (Sun with ultrahigh camera exposure). When starting from a system trained on $N = 7$ initial conditions described above, with the closest condition being Sun with glare, on-line training for the new domain takes approximately 30 minutes to first fine-tune the cycle-consistency GAN generators, followed by approximately 10 minutes to train the input adapter. This gives a complete cycle of 40 minutes, meaning that we can fine-tune for new domains approximately 36 times per day. This time should decrease with the addition of more conditions, as new domains could then benefit from 'closer' seeds when performing online training.

V. Results

For semantic segmentation, we create a testing split from the RobotCar Dataset [32] **reference** sequence and generate testing sequences with different conditions by applying style-transfer using the cycle-consistency GAN generators obtained during the training stage of our pipeline. This process yields sequences with 8 different conditions (the 7 initial conditions and one additional condition for testing online learning) and a common approximated ground truth. We again wish to remind the reader that testing is done on sequences derived from the **reference** sequence through style transfer (along with the approximated ground truth) due to a lack of semantic annotation for the RobotCar Dataset. To show the increase in segmentation performance on a dataset with hand-labelled groundtruth, we further test our system on the validation split of the BDD100K segmentation dataset [37]. As we train exclusively on data from the RobotCar Dataset, BDD is a domain that has *never* been seen during training of any components of our pipeline, and better reflects the usefulness of the proposed pipeline. As the BDD validation sequence does not have enough instances of each condition to also demonstrate online-learning, we freeze our system trained on 8 conditions ($N = 7$ initial conditions and 1 online-learned condition) and test it on BDD. For topological localisation, we test on **real** sequences (not produced using style-transfer) from the RobotCarDataset using the provided INS-RTK GPS ground truth, using a tolerance of 5 meters and reporting Precision-Recall and Area Under Curve (AUC).

A. Quantitative results

Semantic segmentation performance is significantly improved for RobotCar sequences, as can be seen in Table III. We compare our method (Indiv. adapters) of selecting input adapters with 3 other scenarios: a Baseline where the segmentation model is applied directly on the input image, one where we fine-tune the DeepLab segmentation model on all existing conditions (Deeplab-all) and one where we fine-tune individual DeepLab segmentation models for each condition, and use the classifier output to select the right model. The results show that our method consistently and significantly surpasses all other methods, with an average improvement of over 20 percentage points over the Baseline method. Additionally, night-time conditions show impressive gains in performance, with over 47 percentage points gained over Baseline for the **Night(rain)** condition. Table I presents results for semantic segmentation on the BDD dataset, which has never been seen during training. We test against the same 3 methods described above and again observe significant improvements, with over 5 percentage points gained in Mean Intersection Over Union compared to the Baseline method.

Similarly, topological localization shows a significant overall improvement, with an average of 10 percentage points of overall improvement in Area Under Curve (AUC), and very large improvements for Sun(glare) and Shadows. Night-time traversals are one exception where the improvements are still positive but smaller, as the task of detecting discriminative features is arguably harder than performing segmentation. Table II and Figures 8,7 present the results in more detail.

The condition classifier has an overall accuracy of 91%. The classifier confusion matrix is presented in Figure 9. The confusion of the reference, dusk and snow conditions does not lead, empirically, to a large drop in upstream task performance as there is a large degree of similarity between them.

B. Qualitative results

Additionally, we inspect segmentation results on **real** sequences (not produced using style-transfer) from the RobotCar Dataset. While they possess the same range of conditions as the ones the system was trained on, a ground truth is not available. We observe improvements in segmentation across the board, with the most important classes (vehicles, pedestrians, bicyclists etc) becoming distinguishable in even the most difficult conditions. An example for night-time is given in Figure 6.

C. Online learning results

To test our online learning capabilities, we run our system on a condition never before seen during training, **Sun(with ultrahigh camera exposure)**. The descriptor extracted from the penultimate layer of the classifier cannot be accurately matched to any stored descriptors, so the online training

[1] https://github.com/tensorflow/models/blob/master/research/deeplab/g3doc/model_zoo.md

[2] https://github.com/uzh-rpg/netvlad_tf_open



Fig. 6. Improvement of semantic segmentation on a **real** RobotCar night-time input. The first image is the input image, the second image is the output of the selected adapter, the third image is the result of running the segmentation model on the adapted image, while the last image is the result of running the segmentation model on the original, raw input image.

process is triggered. In the descriptor feature space, the closest condition stored is **Sun(glare)**, which is used as a seed for training the cycle-consistency GAN generators and a new input adapter. Results for this new condition are presented in Figures 7, 8 and in the *gray* shaded columns in Tables II and III. As with the initial condition, we observe a large and significant increase in performance for both topological localisation (over 15 percentage points) and semantic segmentation (11 percentage points).

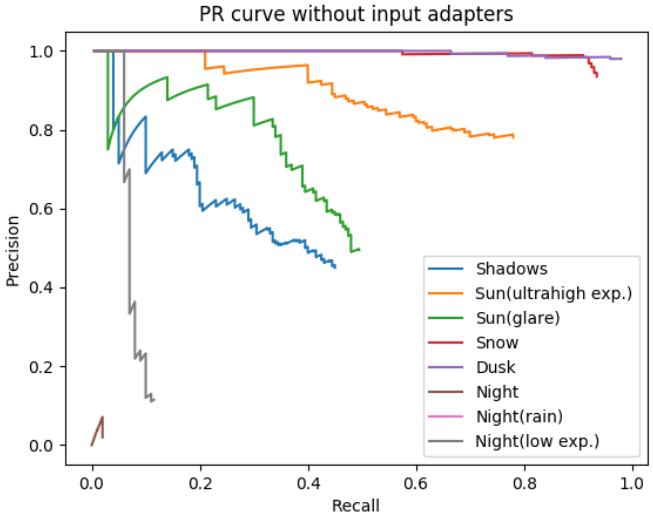


Fig. 7. RobotCar topological localisation Precision-Recall without input adapters. The AUC values can be found in Table II.

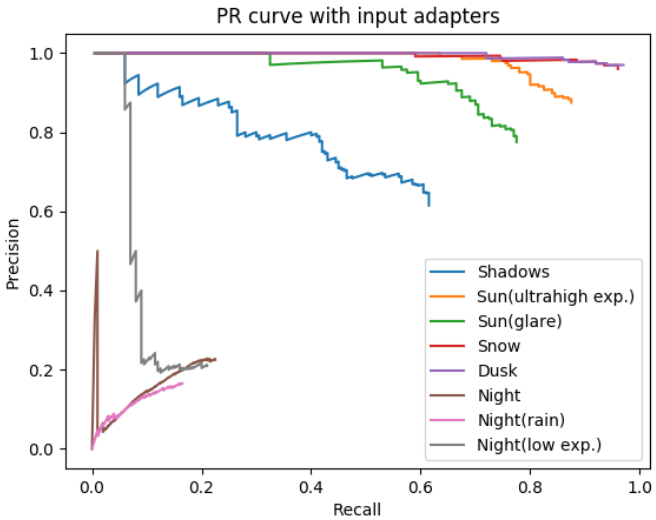


Fig. 8. RobotCar topological localisation Precision-Recall with input adapters. The AUC values can be found in Table II.

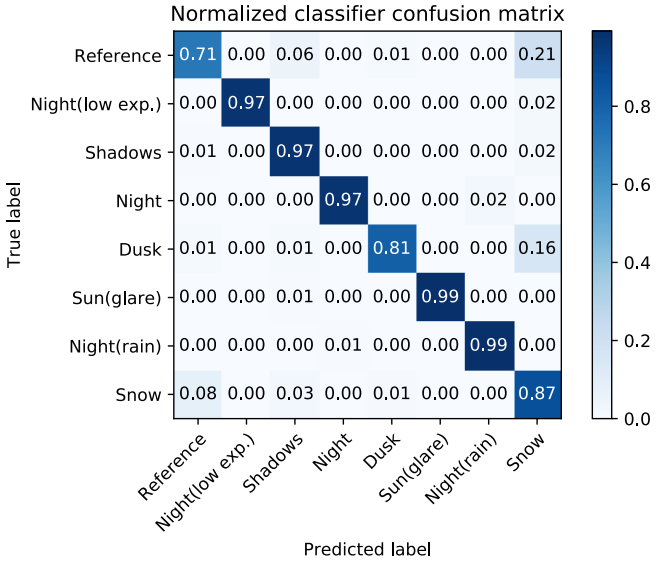


Fig. 9. RobotCar Condition Classifier confusion matrix. Our condition classifier achieves a 91% overall accuracy rate.

TABLE I
BDD SEGMENTATION MEAN INTERSECTION OVER UNION (MIOU)

No adapter	Deeplab-all	Indiv. Deeplabs	Indiv. adapters(**ours**)
0.4070	0.4100	0.4319	**0.4503**

VI. CONCLUSIONS

To prevent performance of computer vision tasks from degrading quickly and often catastrophically when input conditions change, we have presented a domain adaptation system that uses light-weight input adapters to pre-processes input images, irrespective of their appearance, in a way that makes them compatible with off-the-shelf computer vision tasks that are trained only on inputs with ideal conditions. No fine-tuning is performed on the off-the-shelf models, and the system is capable of incrementally training new input adapters in a self-supervised fashion, using the computer vision tasks as supervisors, when the input domain differs significantly from previously seen domains. We report large improvements in semantic segmentation and topological localization performance on two popular datasets, RobotCar and BDD. This work is presented as a framework, and each end-to-end differentiable component may be replaced with a better-performing counterpart, or with one that is better-suited for the task at hand, if available. Additionally, the training process may be extended to work with an arbitrary number of supervisory signals. Finally, our on-line training regimen benefits from convergence times that decrease as a function of the number of domains trained on.

VII. ACKNOWLEDGEMENTS

This work was supported by a Oxford-Google DeepMind Graduate Scholarship and EPSRC/UK Research and Innovation Programme Grant EP/M019918/1.

TABLE II

ROBOTCAR TOPOLOGICAL LOCALISATION AREA UNDER CURVE (AUC)

Method	Shadows	Sun(glare)	Snow	Dusk	Night	Night(rain)	Night(low exp.)	Sun(ultrahigh exp.)	Mean
No adapters	0.2928	0.3949	0.9263	**0.9710**	0.0009	0.0000	0.0719	0.7043	0.4202
With adapters(**ours**)	**0.4965**	**0.7404**	**0.9494**	0.9605	**0.0374**	**0.0191**	**0.0981**	**0.8593**	**0.5200**

TABLE III

ROBOTCAR SEGMENTATION MEAN INTERSECTION OVER UNION (MIOU)

Method	Shadows	Sun(glare)	Snow	Dusk	Night	Night(rain)	Night(low exp.)	*Sun(ultrahigh exp.)*	Mean
Baseline	0.6014	0.4316	0.5677	0.6156	0.1404	0.0859	0.1850	*0.4423*	0.3837
Deeplab-all	0.5375	0.4712	0.5055	0.5372	0.3465	0.3572	0.3593	*0.4821*	0.4495
Indiv. deeplabs	0.5594	0.4948	0.5303	0.5656	0.3948	0.4138	0.4184	*0.5120*	0.4861
Indiv. adapters(**ours**)	**0.6292**	**0.6419**	**0.6525**	**0.6327**	**0.5301**	**0.5627**	**0.5136**	***0.5500***	**0.5891**

REFERENCES

[1] D. Dai and L. V. Gool, "Dark model adaptation: Semantic image segmentation from daytime to nighttime," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, Nov 2018, pp. 3819–3824.

[2] H. Porav, W. Maddern, and P. Newman, "Adversarial training for adverse conditions: Robust metric localisation using appearance transfer," *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 1011–1018, 2018.

[3] A. Anoosheh, T. Sattler, R. Timofte, M. Pollefeys, and L. V. Gool, "Night-to-day image translation for retrieval-based localization," *CoRR*, vol. abs/1809.09767, 2018.

[4] L. Ma, X. Jia, S. Georgoulis, T. Tuytelaars, and L. V. Gool, "Exemplar guided unsupervised image-to-image translation with semantic consistency," in *International Conference on Learning Representations*, 2019.

[5] J. Hoffman, E. Tzeng, T. Park, J.-Y. Zhu, P. Isola, K. Saenko, A. Efros, and T. Darrell, "CyCADA: Cycle-consistent adversarial domain adaptation," in *Proceedings of the 35th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, J. Dy and A. Krause, Eds., vol. 80. Stockholmsmssan, Stockholm Sweden: PMLR, 10–15 Jul 2018, pp. 1989–1998.

[6] M. Limmer and H. P. A. Lensch, "Infrared colorization using deep convolutional neural networks," *CoRR*, vol. abs/1604.02245, 2016.

[7] E. Tzeng, J. Hoffman, N. Zhang, K. Saenko, and T. Darrell, "Deep domain confusion: Maximizing for domain invariance," *CoRR*, vol. abs/1412.3474, 2014.

[8] M. Wulfmeier, A. Bewley, and I. Posner, "Addressing appearance change in outdoor robotics with adversarial domain adaptation," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, October 2017.

[9] Z. Ren and Y. J. Lee, "Cross-domain self-supervised multi-task feature learning using synthetic imagery," *CoRR*, vol. abs/1711.09082, 2017.

[10] R. A. Jacobs, M. I. Jordan, S. J. Nowlan, and G. E. Hinton, "Adaptive mixtures of local experts," *Neural Comput.*, vol. 3, no. 1, pp. 79–87, Mar. 1991.

[11] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "CARLA: An open urban driving simulator," in *Proceedings of the 1st Annual Conference on Robot Learning*, 2017, pp. 1–16.

[12] G. Ros, L. Sellart, J. Materzynska, D. Vazquez, and A. M. Lopez, "The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[13] S. Sankar, Y. Balaji, A. Jain, S.-N. Lim, and R. Chellappa, "Learning from synthetic data: Addressing domain shift for semantic segmentation," 06 2018, pp. 3752–3761.

[14] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in *ECCV*, 2018.

[15] J. Fu, Y. Wang, and H. Lu, "Stacked deconvolutional network for semantic segmentation," *IEEE transactions on image processing : a publication of the IEEE Signal Processing Society*, 2017.

[16] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[17] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 3213–3223.

[18] M. Cummins and P. Newman, "FAB-MAP: Probabilistic Localization and Mapping in the Space of Appearance," *The International Journal of Robotics Research*, vol. 27, no. 6, pp. 647–665, 2008.

[19] M. J. Milford and G. F. Wyeth, "Seqslam: Visual route-based navigation for sunny summer days and stormy winter nights," in *2012 IEEE International Conference on Robotics and Automation*, May 2012, pp. 1643–1649.

[20] R. Arandjelović, P. Gronat, A. Torii, T. Pajdla, and J. Sivic, "NetVLAD: CNN architecture for weakly supervised place recognition," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[21] Y.-H. Tsai, W.-C. Hung, S. Schulter, K. Sohn, M.-H. Yang, and M. K. Chandraker, "Learning to adapt structured output space for semantic segmentation," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7472–7481, 2018.

[22] N. Souly, C. Spampinato, and M. Shah, "Semi and weakly supervised semantic segmentation using generative adversarial network," *CoRR*, vol. abs/1703.09695, 2017.

[23] Z. Murez, S. Kolouri, D. J. Kriegman, R. Ramamoorthi, and K. Kim, "Image to image translation for domain adaptation," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4500–4509, 2018.

[24] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Computer Vision (ICCV), 2017 IEEE International Conference on*, 2017.

[25] A. Dundar, M.-Y. Liu, T.-C. Wang, J. Zedlewski, and J. Kautz, "Domain stylization: A strong, simple baseline for synthetic to real image domain adaptation," *arXiv preprint arXiv:1807.09384*, 2018.

[26] C. Sakaridis, D. Dai, and L. Van Gool, "Semantic nighttime image segmentation with synthetic stylized data, gradual adaptation and uncertainty-aware evaluation," *ArXiv e-prints*, 2019.

[27] L. Clement and J. Kelly, "How to train a cat: Learning canonical appearance transformations for direct visual localization under illumination change," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 2447–2454, July 2018.

[28] M. Wulfmeier, A. Bewley, and I. Posner, "Incremental adversarial domain adaptation for continually changing environments," in *International Conference on Robotics and Automation (ICRA)*, 2018.

[29] M. Brki, M. Dymczyk, I. Gilitschenski, C. Cadena, R. Siegwart, and J. Nieto, "Map management for efficient long-term visual localization in outdoor environments," in *2018 IEEE Intelligent Vehicles Symposium (IV)*, June 2018, pp. 682–688.

[30] R. Gong, W. Li, Y. Chen, and L. V. Gool, "Dlow: Domain flow for adaptation and generalization," *CoRR*, vol. abs/1812.05418, 2018.

[31] H. Germain, G. Bourmaud, and V. Lepetit, "Efficient condition-based representations for long-term visual localization," *CoRR*, vol. abs/1812.03707, 2018.

[32] W. Maddern, G. Pascoe, C. Linegar, and P. Newman, "1 Year, 1000km: The Oxford RobotCar Dataset," *The International Journal of Robotics Research (IJRR)*, vol. 36, no. 1, pp. 3–15, 2017.

[33] ——, "1 Year, 1000km: The Oxford RobotCar Dataset," *The International Journal of Robotics Research (IJRR)*, vol. 36, no. 1, pp. 3–15, 2017.

[34] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, N. Navab, J. Hornegger, W. M. Wells, and A. F. Frangi, Eds. Cham: Springer International Publishing, 2015, pp. 234–241.

[35] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[36] R. Arandjelovi, P. Gronat, A. Torii, T. Pajdla, and J. Sivic, "Netvlad: Cnn architecture for weakly supervised place recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 6, pp. 1437–1451, June 2018.

[37] F. Yu, W. Xian, Y. Chen, F. Liu, M. Liao, V. Madhavan, and T. Darrell, "BDD100K: A diverse driving video database with scalable annotation tooling," *CoRR*, vol. abs/1805.04687, 2018.

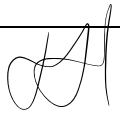
Statement of Authorship for joint/multi-authored papers for PGR thesis

To appear at the end of each thesis chapter submitted as an article/paper

The statement shall describe the candidate's and co-authors' independent research contributions in the thesis publications. For each publication there should exist a complete statement that is to be filled out and signed by the candidate and supervisor **(only required where there isn't already a statement of contribution within the paper itself).**

Title of Paper	Don't Worry About the Weather: Unsupervised Condition-Dependent Domain Adaptation
Publication Status	Published
Publication Details	**H. Porav**, T. Bruls and P. Newman, "Don't Worry About the Weather: Unsupervised Condition-Dependent Domain Adaptation," *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, Auckland, New Zealand, 2019, pp. 33-40, doi: 10.1109/ITSC.2019.8917073.

Student Confirmation

Student Name:	Horia Porav		
Contribution to the Paper	- Generating the ideas for the experiments, including the canonical representation for images and classifier-based selection of models - Conceiving and executing the pipeline architecture, including the models and all software implementations - Generating multiple conditions of the same reference datasets - Generating the NetVLAD descriptors for the reference datasets - Producing part of the pseudo-groundtruth for the segmentation dataset - Dataset curation - Executing the segmentation and localization experiments - Producing and collating the results - Writing the paper, along with diagrams, tables and figures - Designing the presentation for the paper		
Signature		Date	03 October 2020

Supervisor Confirmation

By signing the Statement of Authorship, you are certifying that the candidate made a substantial contribution to the publication, and that the description described above is accurate.

Supervisor name and title: Professor Paul Newman		
Supervisor comments		
Signature 	Date	05-10-2020

This completed form should be included in the thesis, at the end of the relevant chapter.

4
Image de-noising

Contents

4.1 I Can See Clearly Now: Image Restoration via De-Raining 56
4.2 The Right (Angled) Perspective 65

4.1 I Can See Clearly Now: Image Restoration via De-Raining

Problem: Segmentation tasks suffer from severely degraded performance on images affected by adherent rain drops and streaks. This is a common issue in outdoor robotic applications such as autonomous vehicles.

Solution: We introduce a novel stereo dataset recorded using a system that allows one lens to be affected by real water droplets while keeping the other lens clear. We additionally use a method of adding computer-generated adherent water droplets and streaks to any image. We train a de-noising generator using data produced by the above methods and show that it is effective at removing the effect of real water droplets, in the context of image reconstruction and road-marking segmentation.

Status: Published at ICRA2019

I Can See Clearly Now : Image Restoration via De-Raining

Horia Porav, Tom Bruls and Paul Newman

***Abstract*— We present a method for improving segmentation tasks on images affected by adherent rain drops and streaks. We introduce a novel stereo dataset recorded using a system that allows one lens to be affected by real water droplets while keeping the other lens clear. We train a denoising generator using this dataset and show that it is effective at removing the effect of real water droplets, in the context of image reconstruction and road marking segmentation. To further test our de-noising approach, we describe a method of adding computer-generated adherent water droplets and streaks to any images, and use this technique as a proxy to demonstrate the effectiveness of our model in the context of general semantic segmentation. We benchmark our results using the CamVid road marking segmentation dataset, Cityscapes semantic segmentation datasets and our own real-rain dataset, and show significant improvement on all tasks.**

I. Introduction

If we want machines to work outdoors and see while doing so, they have to work in the rain. When rain and lenses interact, computer vision becomes harder - wild local distortions of the image appear which dramatically impede image understanding tasks. However the distortions are not noise, they are structured, the light field is simply bent and attenuated, and accordingly can be modelled and reversed.

In this work we develop a filter which as a pre-processing step removes the effect of raindrops on lenses. Several tasks are affected by the presence of adherent water droplets on camera lenses or enclosures, such as semantic segmentation [1], localisation using segmentation [2], [3] or road marking segmentation [4]. In this paper we choose to use segmentation as an example task by which to test the effectiveness of our method. Many approaches so far have reached for multi-modal data [5], domain adaptation [6], [7] or training on synthetic data [8], however this can become awkward as:

1) Acquiring rainy images is time-consuming, expensive or impossible for many tasks or setups, especially in the case of supervised training, where ground truth data is needed.
2) Training, domain-adapting or fine-tuning each individual task with augmented data is intractable.

We take a different approach and build a system as an image preprocessor, the output of which is a cleaned, de-rained image that improves the performance of many tasks performed on the image.

We begin by creating a bespoke real-world small baseline stereo dataset where one lens is affected by real water droplets and the other is kept dry. The methodology and apparatus for doing so is presented in section IV-A. Using this dataset, we train a de-raining generator and show that it is able to both drastically improve the visual quality of images and restore performance on road marking segmentation tasks.

Authors are from the Oxford Robotics Institute, University of Oxford, UK. {horia,tombruls,pnewman}@robots.ox.ac.uk

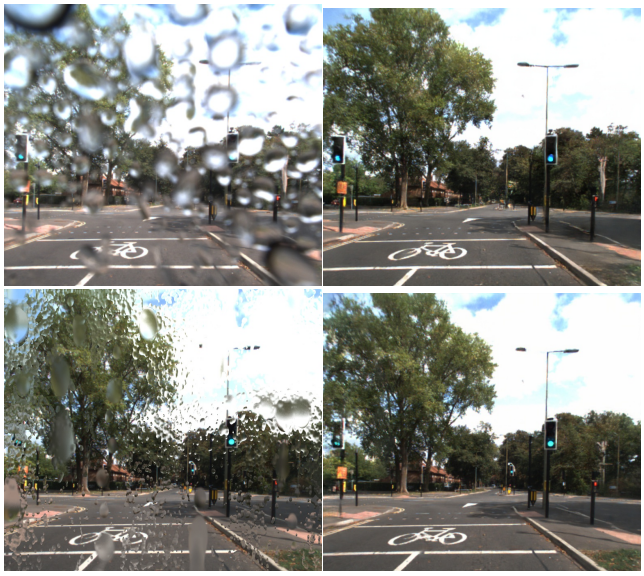


Fig. 1. We learn a de-noising generator that can remove noise and artefacts induced by the presence of adherent rain droplets and streaks. On the top left,input images that are affected by real rain drops. On the top right, the cleaned, de-rained images. On the bottom left, input images that are affected by computer-generated rain drops. On the bottom right, the cleaned, de-rained images.

Secondly, we describe a way of efficiently adding computer-generated adherent rain droplets and adherent streaks to any image using GPU shaders. This system is presented in section III-A. As the Cityscapes dataset provides a good groundtruth for segmentation but does not contain images with significant rain on the lens, we modify it using this technique and use it as a proxy to study the effects of rain on general semantic segmentation. Additionally, we create a synthetic rain dataset by adding computer-generated rain drops to a full Oxford RobotCar dataset [9] and to the CamVid [10] dataset.

Our main contributions include:

- a de-raining model that produces state of the art results;
- using computer-generated water drops as a proxy to study the effects of rain on segmentation for datasets that provide a ground truth but do not normally contain rainy images; and
- a real-world very-narrow-baseline stereo dataset with rainy & clear images covering a wide array of dynamic scenes.

Our aim is to show that pre-processing the image leads to better performance as compared to training, retraining or fine-tuning a task-specific model with rain-augmented data.

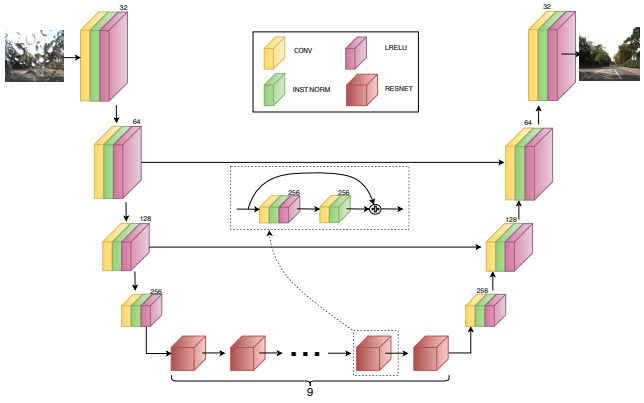


Fig. 2. The internal architecture of our generator. We motivate the addition of additive skip connections by observing that much of the structure of the input image should be kept, along with illumination levels and fine details.

We benchmark our de-raining model on the following tasks:

- Road marking segmentation and image restoration on a real-world small baseline stereo dataset where one lens is affected by real water droplets and the other is kept dry and clear.
- Image reconstruction on the real-world dataset of [11].
- Road marking segmentation and image restoration on CamVid [10] and RobotCar [9] imagery with computer-generated droplets added.
- Semantic segmentation on Cityscapes [1] imagery with computer-generated droplets added.

The quantitative and qualitative results are presented in section V.

II. Related Work

Generally speaking, the quality of an image can be affected in two ways by bad weather conditions. Firstly, contaminants in the atmosphere, such as falling rain, fog, smog or snow will hinder visibility or partially occlude a scene but do not significantly distort the image. Secondly, adherent contaminants such as water droplets, which stick to transparent surfaces or lenses, tend to heavily distort the image, essentially acting as a secondary lens with various degrees of blurring. Several techniques are employed to clean the first type of images, such as those used by [12], [13], [14], [15], [16], however these techniques cannot be used to restore images affected by adherent rain, as the optics involved differ significantly from those of atmospheric droplets. The remainder of this section outlines some of the techniques used to tackle the effects of adherent rain droplets and adherent streaks.

Rain Modelling and Simulation: In the context of computer vision, several studies have attempted to model the structure and optical properties of adherent water droplets. The authors of RIGSEC [17], [18] model raindrops first as sections of a sphere and later account for the effect of gravity using 2D Bezier curves, and confirm experimentally that a physically correct droplet shape can be computed using this method. [19] additionally study and model the dark band around the edges of adherent drops, and show that a simplified model is enough to correctly undistort the image on the surface of the droplet.

We base our simple synthetic droplet model on the works of [17], [18] and [19], by storing proto-droplet normal maps which are subsequently warped and combined at run time using an approach similar to meta-balls [20].

Additionally, several small datasets have been created to benchmark the accuracy of de-raining techniques. In [21], water is sprayed on a glass pane fitted in front of a camera, but no ground truth is provided due to temporal illumination and scene changes. A video sequence where the lens is affected by real rain droplets is also provided, again without ground truth. The authors of [22] again use a glass pane sprayed with water to study the performance of their droplet detection and removal pipeline, but only offer ground truth for the position of the droplets. The first attempt to provide accurate ground truth is made by [11], in which images of static scenes are captured both with and without a glass pane sprayed with water in front of the camera. This process is, however, very difficult to scale to the number of images required by modern deep-learning approaches. To our best knowledge, we are the first to record a real-world large dataset of sequential dynamic scenes with an accurate, clear ground truth and a large variation in raindrop type and size.

Raindrop Detection and Removal: In [17] and [18], raindrops are detected by attempting to match a template of a synthetic raindrop at locations where the presence of a real drop is hypothesized. This approach breaks down when the shape of the real droplets differs significantly from that of the template. The authors of [22] take a different approach by observing that the motion inside droplets is between 1/30 and 1/20 slower than that in the scene. They use this information to detect raindrops and then attempt to restore the image by using a combination between image inpainting and recovering data from within the distorted image formed on the droplet. Both techniques use multi-frame information for image reconstruction, and are not applicable to single-images.

Multi-camera and pan-tilt setups are exploited by [23], [24], [25] and [26]. These techniques use disparities to detect droplets and subsequently attempt to replace the affected regions in one lens with information from the other lens. This approach does not work on single images and assumes that the same regions are not covered by rain in both frames.

Convolutional neural networks were used by [21] to restore images affected by dirt and rain. They use a simple 3-layer architecture, each with 512 units, which works well on small drops but breaks down with much larger contaminants. A much larger Generative Adversarial Network (GAN) model [27] is used by [11], along with attention [28]. They leverage their static dataset to provide a ground truth for the droplet attention mask and train a recurrent model that outputs a heatmap of the location of the droplets. This heatmap is then concatenated with the input image and run through the GAN. They produce state-of-the-art results and made their dataset publicly available, which has allowed us to directly compare our method with theirs.

III. Learning to Clean Images

A. Computer-Generated Synthetic Rain

We base our simple synthetic droplet model on the works of [17], [18] and [19], generate the locations of raindrops using a simple statistical approach, model the interactions between raindrops using metaballs [20] and implement its rendering efficiently using GPU shaders.

A proto-raindrop is created using a simple refractive model that assumes a pinhole camera. The refraction angle is encoded following a scheme similar to normal mapping [29] by using a 2D look-up table represented by the RED and GREEN channels of a texture T, with the thickness of the drop encoded in the BLUE channel of the same texture. This texture T is then masked using an alpha layer that allows blending of the water drops with the background image and other drops, as shown in Figure 3a. With the drop acting as a simple lens, the coordinate (x_r, y_r) of the world point that is rendered at the location (u, v) on the surface of a drop is given by the following simplified distortion model:

$$x_r = u + (R * B) \quad (1)$$
$$y_r = v + (G * B). \quad (2)$$

Each image location (u, v) has a probability P_r of becoming the center of a proto-raindrop whose dimensions are scaled along the horizontal and vertical directions by a tuple of random values S_x and S_y. For each timestep, the center of a droplet may undergo a slip of D_x pixels along the horizontal and D_y pixels along the vertical direction as a function of the droplet diameter d:

$$D_x, D_y = \begin{cases} 0, 0 & d \leq 4mm \\ x \sim \mathcal{N}(0, 3), P_d * 5 & d > 4mm, \end{cases}$$

where P_d represents the probability of slip along the vertical direction and x denotes the random deviation of the slip along the horizontal direction. We empirically choose a maximum of 5 pixels of vertical displacement.

For each timestep, droplets that are close to each other are merged using the metaballs approach [20], as shown in Figure 3b. By default, each texture location $T(u, v)$ that does not fall under a droplet encodes a normal that is perpendicular to the background image. Finally, the image is sampled using the normal map defined by the texture T to produce a result similar to the one in the top-left corner of Fig 1. Using this technique we have created three synthetic rain datasets:

- synthetic rain added to CamVid, complete with road marking ground truth;
- synthetic rain added to Cityscapes, complete with semantic segmentation ground truth; and
- synthetic rain added to the dry images from our stereo dataset, complete with road marking ground truth.



Fig. 3. Metaballs.

B. The de-raining network

The de-raining network architecture is based on Pix2PixHD [30]. The architecture is shown in Fig. 2. We employ 4 down-convolutional layers with stride 2, followed by 9 ResNet [31] blocks and 4 up-convolutional layers. We motivate the addition of skip connections by observing that most of the structure of the input image should be kept, along with illumination levels and fine details.

To promote better generalization and inpainting, we refrain from using any direct pixel-wise loss and instead use a combination of adversarial, perceptual, and multi-scale discriminator feature losses. The discriminator architecture is a CNN with 5 layers, similar to PatchGAN [32]. We present the full structure of the losses in the next section.

C. Losses

Similar to [33], we apply an adversarial loss through a discriminator on the output of the generator. This loss is formulated as:

$$\mathcal{L}_{adv} = (D(G(I_{\text{rainy}})) - 1)^2. \quad (3)$$

The discriminator is trained to minimize the following loss:

$$\mathcal{L}_{disc} = (D(I_{\text{clear}}) - 1)^2 + (D(I_{\text{de-rained}}))^2, \quad (4)$$

where I_{derained} is sampled from a pool of previously derained images.

The perceptual loss [34] is applied between the label and reconstructed image:

$$\mathcal{L}_{\text{perc}} = \sum_{i=1}^{n_{VGG}} \frac{1}{w_i^{perc}} \|VGG(I_{\text{clear}})_i - VGG(G(I_{\text{rainy}}))_i\|_1, \quad (5)$$

where n_{VGG} represents the number of VGG layers that are used to compute the loss and $w_i^{perc} = 2^{(n_{VGG}-i)}$ weighs the importance of each layer.

Additionally, a multi-scale discriminator feature loss [30] is applied between the label and reconstructed image:

$$\mathcal{L}_{\text{msadv}} = \sum_{i=1}^{n_{ADV}} \frac{1}{w_i^{adv}} \|D(I_{\text{clear}})_i - D(G(I_{\text{rainy}}))_i\|_1, \quad (6)$$

where n_{ADV} represents the number of discriminator layers that are used to compute the loss and $w_i^{adv} = 2^{(n_{ADV}-i)}$ weighs the importance of each layer.

The complete generator objective $\mathcal{L}_{\text{gen}}$ becomes:

$$\mathcal{L}_{\text{gen}} = \lambda_{\text{adv}} * \mathcal{L}_{\text{adv}} + \lambda_{\text{perc}} * \mathcal{L}_{\text{perc}} + \lambda_{\text{msadv}} * \mathcal{L}_{\text{msadv}}. \quad (7)$$

Each λ term is a hyperparameter that weights the importance of each term of the loss equation. We wish to estimate the generator G and discriminator D functions such that:

$$G, D = \arg\min_{G,D} \mathcal{L}_{\text{gen}} + \mathcal{L}_{\text{disc}}. \quad (8)$$

In the following section we describe how the network is trained to minimise the above losses.

IV. Experimental Setup

A. Stereo rain dataset

In this section we present the hardware used to record our narrow-baseline stereo dataset that allows one lens to be affected by real water droplets while keeping the other lens clear. The camera setup is shown in Figure 8. A 3D-printed bi-partite chamber is sandwiched between two acrylic clear panels and placed in front of the two lenses, with the left-hand section of the chamber being kept dry at all times, while



Fig. 4. CamVid road marking segmentation results. From left to right: rainy input image, segmentation result on rainy image, derained input image, segmentation result on derained image.

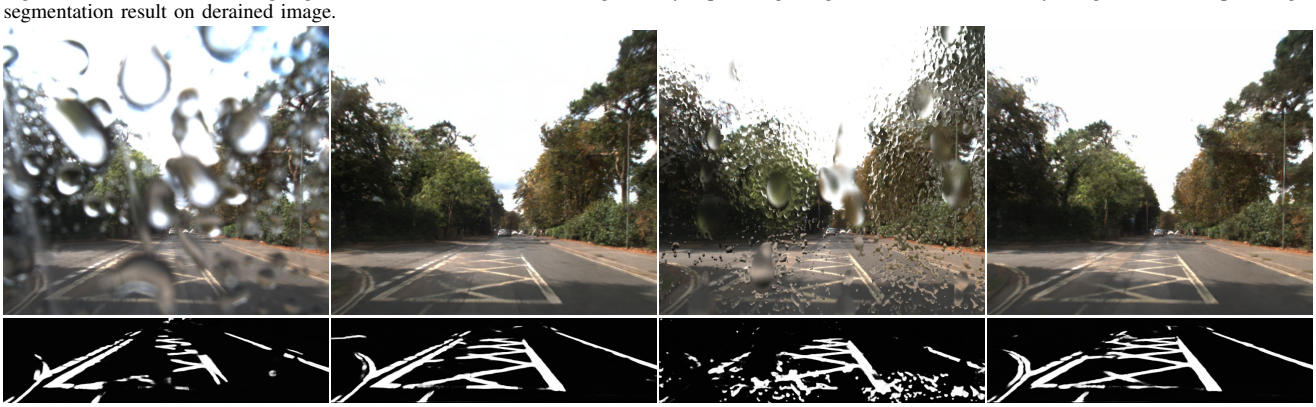


Fig. 5. RobotCar road marking segmentation results.First column shows a RobotCar(R) real rain image and segmentation result. Second column shows the derained real rain image and segmentation result. Third column shows a RobotCar(S) computer-generated rain image and segmentation result. Fourth column shows the derained computer-generated rain image and segmentation result.

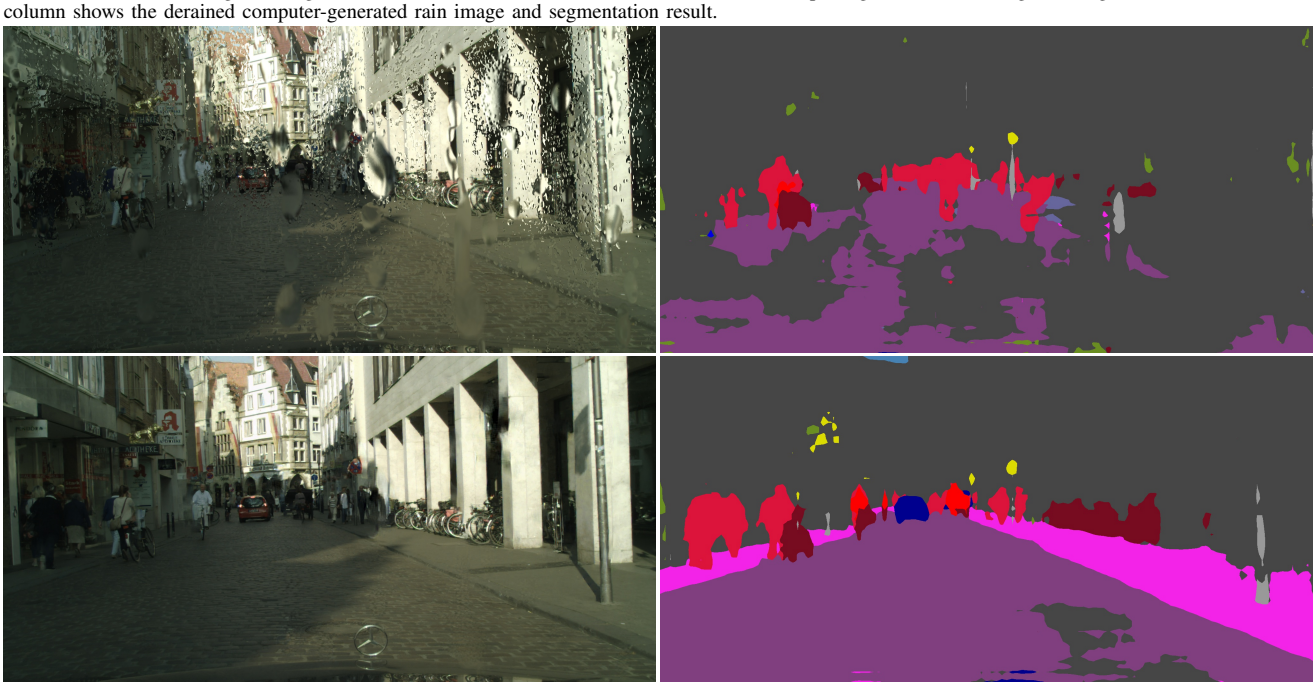


Fig. 6. Cityscapes semantic segmentation results. The first row shows a rainy image on the left and its corresponding semantic segmentation on the right. The second row shows the derained image on the left and its corresponding semantic segmentation on the right.



Fig. 7. An example from our stereo dataset. The image on the left is produced by the left lens, which is affected by water drops. The image in the middle is produced by the dry right hand lens. The image on the right is the road marking segmentation ground truth.

the right-hand section is sprayed with water droplets using an internal nozzle fitted at the top of the chamber. The angle of this chamber with respect to the axes of the cameras can be modified to simulate a slanted windscreen or enclosure, and the distance from the lenses can be increased or decreased accordingly to replicate different levels of focus or blur on the droplets.

The nozzle spans the entire width of the right chamber and is capable of producing water droplets with a diameter between 1mm and 8mm, as well as streaks of water. This variability is achieved by modulating the water pressure using a number of pulse width modulation regimes. The water is drained from the bottom of the chamber and is returned to a storage tank for recirculation. The cameras used are Point Grey Grasshopper 2 with 4.5 mm F/1.4 lenses, a baseline of 29 mm and automatic synchronisation. The system is fully portable and the water is completely contained within the circuit formed by the right chamber, pump and tank.

We have collected approximately 50000 pairs of images by driving in and around the city of Oxford. The image pairs are undistorted, cropped and aligned. We have selected 4818 image pairs to form a training, validation and testing dataset. From the testing partition, we have created ground truth road marking segmentations for 500 images. An example from our dataset is shown in Figure 7.

Compared to the painstakingly-collected dataset of [11], our setup is a set-and-forget approach: once the stereo camera has been mounted on a vehicle, it is trivial to collect large amounts of well-synchronised and well-aligned pairs of images.

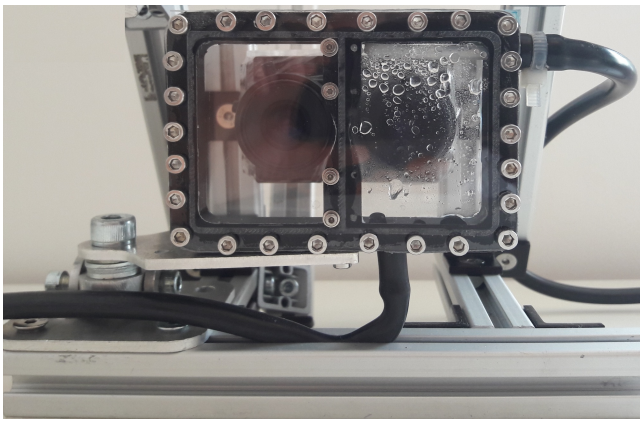


Fig. 8. Our small-baseline stereo camera setup. A bi-partite chamber with acrylic clear panels is placed in front of the lenses, with the left-hand section being kept dry at all times, while the right-hand section is sprayed with water droplets using an internal nozzle.

B. Training

We used a network training regimen similar to [30]. For each iteration we first trained the discriminator on a clear image and a de-rained image from a previous iteration with the goal of minimizing $\mathcal{L}_{\text{disc}}$, and then trained the generator on rainy input images to minimize $\mathcal{L}_{\text{gen}}$. We used the Adam solver [35] with an initial learning rate set at 0.0002, a batch size of 1, $\lambda_{\text{adv}} = 1$, $\lambda_{\text{perc}} = 1$ and $\lambda_{\text{msadv}} = 1$.

C. Segmentation Tasks

We used the trained generator G to de-rain all of the rainy input images. To benchmark both the images with computer-generated water drops and the images with real water drops, in the context of road marking segmentation, we used the approach of [4] which trains a U-Net to segment road markings in a binary way. To benchmark the computer-generated water drop images in the context of semantic segmentation, we used DeepLab v3 [36] which has achieved state-of-the-art performance on the Cityscapes dataset.

The generator runs at approximately 1 Hz for images with a resolution of 1280×960, and at approximately 3 Hz for images with a resolution of 640×480 on an Nvidia Titan X GPU.

V. Results

We benchmark our results taking into consideration several metrics across several tasks, and also present results on the quality of the image reconstruction.

A. Quantitative results

Table I presents results for road marking segmentation, in the case of RobotCar with real water drops (R), RobotCar with computer-generated water drops (S) and CamVid with computer-generated water drops (S). Our baseline is represented by the performance of clear images tested on models that were trained using clear images (REFERENCE). For both RobotCar (R), Robotcar (S), and the CamVid (S) datasets, the results show a severely degraded performance when testing rainy images on models that were trained using clear images (RAINY). Retraining the road marking segmentation models with a dataset augmented with rainy images will lead to an improvement in performance (AUGM). However, de-raining the images using our method and testing them on a model trained using clear images (DERAINED) restores the performance of the segmentation to levels that are close to the baseline recorded on clear images. Figure 4 shows road marking segmentation results on CamVid, before and after deraining. Figure 5 shows road marking segmentation results on RobotCar(R)&(S), before and after deraining.

As expected, re-training the segmentation model with a dataset that is augmented with rainy images helps to improve performance, however using a specialised de-raining preprocessing step significantly outperforms this approach, even when tested on a model trained exclusively with clear images. This is the expected advantage of having a model dedicated, in its entirety, to a specific image-to-image mapping task (de-raining), which narrows the variety of images fed to the segmentation task.

Table II presents results for semantic segmentation on the Cityscapes dataset. We benchmark 4 different combinations of models and datasets:

- Cityscapes-clear images tested on a model trained using Cityscapes-clear images;
- Cityscapes-rainy images tested on a model trained using Cityscapes-clear images;
- Cityscapes-rainy images tested on a model trained using Cityscapes-clear and Cityscapes-rainy images; and

TABLE I

ROAD MARKING SEGMENTATION RESULTS

Dataset	REFERENCE(CLEAR)				RAINY				AUGM.				DERAINED			
	Prec.	Rec.	F1	IOU	Prec.	Rec.	F1	IOU	Prec.	Rec.	F1	IOU	Prec.	Rec.	F1	IOU
RobotCar(R)	0.627	0.918	0.734	0.594	0.512	0.628	0.550	0.396	0.486	0.807	0.593	0.434	**0.603**	**0.841**	**0.689**	**0.544**
RobotCar(S)	0.627	0.918	0.734	0.594	0.364	0.595	0.437	0.287	0.654	0.770	0.690	0.541	**0.661**	**0.816**	**0.715**	**0.569**
CamVid(S)	0.576	0.927	0.699	0.551	0.353	0.576	0.425	0.279	0.457	**0.771**	0.563	0.405	**0.520**	0.755	**0.603**	**0.444**

TABLE II

CITYSCAPES SEMANTIC SEGMENTATION RESULTS

Cityscapes Model vs. Dataset	mIOU
CLEAR on CLEAR	0.692
RAINY on CLEAR	0.405
RAINY on AUGMENTED	0.611
DERAINED on CLEAR	**0.651**

TABLE III

RECONSTRUCTION RESULTS

Dataset	RAW		DERAINED	
	PSNR	SSIM	PSNR	SSIM
RobotCar-Rainy(R)	13.02	0.5574	**22.82**	**0.8188**
RobotCar-Rainy(S)	16.80	0.6134	**25.17**	**0.8699**
CamVid-Rainy(S)	16.89	0.6064	**22.11**	**0.7524**
Qian et al.[11](R)	24.09	0.8518	**31.55**	**0.9020**

- Cityscapes-derained(Cityscapes-rainy preprocessed using our deraining model) images tested on a model trained using Cityscapes-clear images.

Similar to the case of road marking segmentation, we notice the same severe degradation of performance when testing with rainy images (RAINY on CLEAR) as compared to the baseline (CLEAR on CLEAR). Again, the performance of derained images tested on a model trained using clear images (DERAINED on CLEAR) is significantly better than the performance of rainy images tested on a model trained using a dataset augmented with rainy images (RAINY on AUGMENTED). Figure 6 shows semantic segmentation results on Cityscapes, before and after deraining.

B. Reconstruction results

Table III presents results on the quality of the image reconstruction using two widely used image-quality metrics, Peak signal-to-noise ratio (PSNR), and Structural similarity (SSIM). We benchmark our model on our real-world RobotCar-Rainy (R) dataset, RobotCar-Rainy with computer-generated rain (S), CamVid-Rainy with computer-generated rain (S), and on the dataset provided by [11]. The RAW column shows the quality of the rainy images, while the DERAINED column shows the quality of the de-rained images, all relative to their clear ground truth. We show that in all cases, de-raining the rain-affected images using our preprocessor significantly increases the quality of the images, as compared to the reference case where raw rainy images are used. Both the real-world rainy dataset images and the images with computer-generated rain are significantly more degraded than the rainy images provided by [11], as seen in column RAW.

Table IV presents reconstruction results on the reference rainy dataset provided by [11]. We show that we achieve state-of-the-art PSNR reconstruction results on images affected by real water drops and only slightly lower SSIM, while, in contrast to [11], not requiring an attention [28] mechanism, which simplifies and speeds up inference and training.

TABLE IV

RECONSTRUCTION QUALITY COMPARISON TO STATE OF THE ART

Model vs. Dataset	Dataset from [11]	
	PSNR	SSIM
Original	24.09	0.8518
Eigen13[21]	28.59	0.6726
Pix2Pix[37]	30.14	0.8299
Qian et al.(no att.)[11]	30.88	0.8670
Qian et al.(full att.)[11]	31.51	**0.9213**
Ours(no att.)	**31.55**	0.9020

VI. CONCLUSIONS

We have presented a system that restores performance of images affected by adherent raindrops on important segmentation tasks. Our results show that road marking segmentation, an important task for autonomous driving systems, is severely affected by adherent rain and that performance can be restored by first running the images through a de-raining preprocessor. Similarly, we show the same reduction and restoration of performance in the case of semantic segmentation, a task that is important in many fields. Additionally, we produce state-of-the-art results in terms of the quality of image restoration, while being able to run in real time. Finally, our system processes the image streams outside of the segmentation pipeline, either offline or online, and hence can be used naturally as a front end to many existing systems. The dataset will be made available at `https://ciumonk.github.io/RobotCar-rainy/`, along with a video describing our results at `https://ciumonk.github.io/RobotCar-rainy/video.html`.

VII. FUTURE WORK

Future work may involve designing a mechanism for producing computer-generated rain that is indistinguishable from real rain in terms of its usefulness in training models that quantitatively rather than qualitatively improve performance on image-based tasks.

VIII. ACKNOWLEDGEMENTS

This work was supported by Oxford-Google DeepMind Graduate Scholarships and Programme Grant EP/M019918/1. The authors wish to thank Valentina Musat for labelling the road markings in our dataset.

REFERENCES

[1] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 3213–3223.

[2] E. Stenborg, C. Toft, and L. Hammarstrand, "Long-term visual localization using semantically segmented images," *CoRR*, vol. abs/1801.05269, 2018.

[3] J. L. Schönberger, M. Pollefeys, A. Geiger, and T. Sattler, "Semantic visual localization," *CoRR*, vol. abs/1712.05773, 2017.

[4] T. Bruls, W. Maddern, A. A. Morye, and P. Newman, "Mark yourself: Road marking segmentation via weakly-supervised annotations from multimodal data," in *Robotics and Automation (ICRA), 2018 IEEE International Conference on*. IEEE, 2018, p. in press.

[5] A. Valada, J. Vertens, A. Dhall, and W. Burgard, "Adapnet: Adaptive semantic segmentation in adverse environmental conditions," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, May 2017, pp. 4644–4651.

[6] Y. Chen, W. Li, C. Sakaridis, D. Dai, and L. V. Gool, "Domain adaptive faster R-CNN for object detection in the wild," *CoRR*, vol. abs/1803.03243, 2018.

[7] M. Wulfmeier, A. Bewley, and I. Posner, "Addressing appearance change in outdoor robotics with adversarial domain adaptation," *CoRR*, vol. abs/1703.01461, 2017.

[8] G. Ros, L. Sellart, J. Materzynska, D. Vazquez, and A. M. Lopez, "The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[9] W. Maddern, G. Pascoe, C. Linegar, and P. Newman, "1 Year, 1000km: The Oxford RobotCar Dataset," *The International Journal of Robotics Research (IJRR)*, vol. 36, no. 1, pp. 3–15, 2017.

[10] G. J. Brostow, J. Fauqueur, and R. Cipolla, "Semantic object classes in video: A high-definition ground truth database," *Pattern Recognition Letters*, vol. 30, no. 2, pp. 88–97, 2009.

[11] R. Qian, R. T. Tan, W. Yang, J. Su, and J. Liu, "Attentive generative adversarial network for raindrop removal from a single image," *CoRR*, vol. abs/1711.10098, 2017.

[12] J. Chen and L. Chau, "A rain pixel recovery algorithm for videos with highly dynamic scenes," *IEEE Transactions on Image Processing*, vol. 23, no. 3, pp. 1097–1104, March 2014.

[13] J. Kim, J. Sim, and C. Kim, "Stereo video deraining and desnowing based on spatiotemporal frame warping," in *2014 IEEE International Conference on Image Processing (ICIP)*, Oct 2014, pp. 5432–5436.

[14] ——, "Video deraining and desnowing using temporal correlation and low-rank matrix completion," *IEEE Transactions on Image Processing*, vol. 24, no. 9, pp. 2658–2670, Sept 2015.

[15] W. Ren, S. Liu, H. Zhang, J. Pan, X. Cao, and M.-H. Yang, "Single image dehazing via multi-scale convolutional neural networks," in *European conference on computer vision*. Springer, 2016, pp. 154–169.

[16] X. Fu, J. Huang, X. Ding, Y. Liao, and J. Paisley, "Clearing the skies: A deep network architecture for single-image rain removal," *CoRR*, vol. abs/1609.02087, 2016.

[17] M. Roser and A. Geiger, "Video-based raindrop detection for improved image registration," in *2009 IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops*, Sept 2009, pp. 570–577.

[18] M. Roser, J. Kurz, and A. Geiger, "Realistic modeling of water droplets for monocular adherent raindrop recognition using bézier curves," in *Computer Vision – ACCV 2010 Workshops*, R. Koch and F. Huang, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 235–244.

[19] S. You, R. T. Tan, R. Kawakami, Y. Mukaigawa, and K. Ikeuchi, "Waterdrop stereo," *CoRR*, vol. abs/1604.00730, 2016.

[20] J. F. Blinn, "A generalization of algebraic surface drawing," *ACM Trans. Graph.*, vol. 1, no. 3, pp. 235–256, July 1982.

[21] D. Eigen, D. Krishnan, and R. Fergus, "Restoring an image taken through a window covered with dirt or rain," in *2013 IEEE International Conference on Computer Vision*, Dec 2013, pp. 633–640.

[22] S. You, R. T. Tan, R. Kawakami, Y. Mukaigawa, and K. Ikeuchi, "Adherent raindrop modeling, detectionand removal in video," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 9, pp. 1721–1733, Sept 2016.

[23] A. Yamashita, M. Kuramoto, T. Kaneko, and K. T. Miura, "A virtual wiper - restoration of deteriorated images by using multiple cameras," in *IROS*. IEEE, 2003, pp. 3126–3131.

[24] A. Yamashita, T. Kaneko, and K. T. Miura, "A virtual wiper-restoration of deteriorated images by using a pan-tilt camera," in *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA '04. 2004*, vol. 5, April 2004, pp. 4724–4729 Vol.5.

[25] A. Yamashita, Y. Tanaka, and T. Kaneko, "Removal of adherent waterdrops from images acquired with stereo camera," in *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Aug 2005, pp. 400–405.

[26] M. Kuramoto, A. Yamashita, T. Kaneko, and K. T. Miura, "Removal of adherent waterdrops in images by using multiple cameras," in *MVA*, 2002, pp. 80–83.

[27] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, ser. NIPS'14. Cambridge, MA, USA: MIT Press, 2014, pp. 2672–2680.

[28] V. Mnih, N. Heess, A. Graves, and K. Kavukcuoglu, "Recurrent models of visual attention," *CoRR*, vol. abs/1406.6247, 2014.

[29] J. Cohen, M. Olano, and D. Manocha, "Appearance-preserving simplification," in *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques*, ser. SIGGRAPH '98. New York, NY, USA: ACM, 1998, pp. 115–122.

[30] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, A. Tao, J. Kautz, and B. Catanzaro, "High-resolution image synthesis and semantic manipulation with conditional GANs," in *Computer Vision and Pattern Recognition (CVPR), 2018 IEEE Conference on*. IEEE, 2018, pp. 1–13.

[31] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[32] C. Li and M. Wand, "Precomputed real-time texture synthesis with markovian generative adversarial networks," in *European Conference on Computer Vision*. Springer, 2016, pp. 702–716.

[33] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networkss," in *Computer Vision (ICCV), 2017 IEEE International Conference on*, 2017.

[34] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in *European Conference on Computer Vision*. Springer, 2016, pp. 694–711.

[35] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[36] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, "Encoder-decoder with atrous separable convolution for semantic image segmentation," in *ECCV*, 2018.

[37] P. Isola, J. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," *CoRR*, vol. abs/1611.07004, 2016.

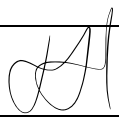
Statement of Authorship for joint/multi-authored papers for PGR thesis

To appear at the end of each thesis chapter submitted as an article/paper

The statement shall describe the candidate's and co-authors' independent research contributions in the thesis publications. For each publication there should exist a complete statement that is to be filled out and signed by the candidate and supervisor **(only required where there isn't already a statement of contribution within the paper itself).**

Title of Paper	I Can See Clearly Now: Image Restoration via De-Raining
Publication Status	Published
Publication Details	**H. Porav**, T. Bruls and P. Newman, "I Can See Clearly Now: Image Restoration via De-Raining," *2019 International Conference on Robotics and Automation (ICRA)*, Montreal, QC, Canada, 2019, pp. 7087-7093, doi: 10.1109/ICRA.2019.8793486.

Student Confirmation

Student Name:	Horia Porav		
Contribution to the Paper	- Generating the ideas for the paper and experiments, including the stereo rig - Designing and executing the stereo rig, pump and reservoir, along with accompanying control and data capture software - Calibrating and aligning the cameras - All software implementations, including models/networks - Training models - Implementing the GPU-shader based synthetic rain generation for the secondary experiments - Labelling part of the road marker segmentation dataset - Designing and running the experiments - Running the data collection - Writing the paper, along with diagrams, tables and figures - Presenting the paper at ICRA 2019		
Signature		Date	03 October 2020

Supervisor Confirmation

By signing the Statement of Authorship, you are certifying that the candidate made a substantial contribution to the publication, and that the description described above is accurate.

Supervisor name and title: Professor Paul Newman			
Supervisor comments			
Signature		Date	05-10-2020

This completed form should be included in the thesis, at the end of the relevant chapter.

4.2 The Right (Angled) Perspective: Improving the Understanding of Road Scenes using Boosted Inverse Perspective Mapping

Problem: Inverse Perspective Mapped images are useful for a number of robotics tasks (e.g. road marking detection, path planning), but since far-away real-world features are represented by a smaller pixel area as compared to identical close-by features, a direct consequence of applying a full perspective transformation to the input is increased unnatural blurring and stretching of features at further distance. Being able to produce sharp and clear IPM images can boost the performance of a number of scene understanding tasks.

Solution: To overcome these challenges, we present an adversarial learning approach which produces a significantly improved IPM in real time from a single front-facing camera image. An Incremental Spatial ResNet Transformer is introduced to address the significant appearance change between the frontal view and IPM. Our model divides the full perspective transformation into a series of N_{STRes} smaller incremental perspective transformations, each followed by a refinement of the transformed feature space using a ResNet block. The intuition behind this is that the slight blurring that occurs as a result of each perspective transformation is restored by the ResNet block that follows it. We demonstrate reliable, qualitative results in different environments and under varying lighting conditions. Furthermore, we show that the boosted IPM view allows for improved hierarchical scene understanding.

Status: Published at IV2019

The Right (Angled) Perspective: Improving the Understanding of Road Scenes using Boosted Inverse Perspective Mapping

Tom Bruls*, Horia Porav*, Lars Kunze, and Paul Newman

***Abstract*— Many tasks performed by autonomous vehicles such as road marking detection, object tracking, and path planning are simpler in bird's-eye view. Hence, Inverse Perspective Mapping (IPM) is often applied to remove the perspective effect from a vehicle's front-facing camera and to remap its images into a 2D domain, resulting in a top-down view. Unfortunately, however, this leads to unnatural blurring and stretching of objects at further distance, due to the resolution of the camera, limiting applicability. In this paper, we present an adversarial learning approach for generating a significantly improved IPM from a single camera image in real time. The generated bird's-eye-view images contain sharper features (e.g. road markings) and a more homogeneous illumination, while (dynamic) objects are automatically removed from the scene, thus revealing the underlying road layout in an improved fashion. We demonstrate our framework using real-world data from the Oxford RobotCar Dataset and show that scene understanding tasks directly benefit from our boosted IPM approach.**

I. Introduction

Autonomous vehicles need to perceive and fully understand their environment to accomplish their navigation tasks. Hence, scene understanding is a critical component within their perception pipeline, not only for navigation and planning, but also for safety purposes. While vehicles use different types of sensors to interpret scenes, cameras are one of the most popular sensing modalities in the field, due to their low cost as well as the availability of well-established image processing techniques.

In recent years, deep learning approaches based on images have been very successful and significantly improved the performance of autonomous vehicles in the context of semantic scene understanding [1], [2]. Many of these approaches take images from a front-facing camera as their input. However, interpretations (i.e. segmented pixels) in this perspective cannot be related directly to the vehicle's action space, which is often encoded in a local and/or global coordinate system.

Images of front-facing cameras as well as their interpretations need to be transformed into a different coordinate system (or view) to be effectively utilized within tasks such as lane detection [3], [4], road marking detection [5], road topology detection [6], [7], object detection/tracking [8]–[10], as well as path planning and intersection prediction [11], [12]. This transformation is commonly referred to as Inverse Perspective Mapping (IPM). IPM takes the frontal view as input, applies a homography, and produces a top-down view of the scene by mapping the pixels to a different 2D-coordinate frame, which is also known as *bird's-eye view*. In practice, this works well in the immediate proximity of the vehicle (assuming the road surface is planar). However, the geometric properties of objects in the distance are affected unnaturally by this non-homogeneous mapping, as shown in Fig. 1. This limits the performance of applications in terms of their accuracy and the distance at which they can be reliably applied. More crucially, however, is the effect of inaccurate mappings on the semantic interpretation of scenes, where small inaccuracies can lead to significant qualitative differences. As we demonstrate in Section V-B (Table I), these qualitative differences can manifest themselves in many ways, including missing lanes and/or late detection of stop



Fig. 1. Boosted Inverse Perspective Mapping (IPM) to improve the understanding of road scenes. *Left:* Top-down view created by applying a homography-based IPM to the front-facing image (*top*), leading to unnatural blurring and stretching of objects at further distance. *Right:* Improved top-down view generated by our Incremental Spatial Transformer GAN, containing sharper features and a homogeneous illumination, while dynamic objects (i.e. the two cyclists) are automatically removed from the scene.

∗ equal contribution

Authors are from the Oxford Robotics Institute, Dept. Engineering Science, University of Oxford, UK. {tombruls, horia, lars, pnewman}@robots.ox.ac.uk

lines (or other critical road markings).

To overcome these challenges, we present an adversarial learning approach which produces a significantly improved IPM in real time from a single front-facing camera image. This is a difficult problem which is not solved by existing methods, due to the large difference in appearance between the frontal view and IPM. State-of-the-art approaches for cross-domain image translation tasks train (conditional) Generative Adversarial Networks (GANs) to transform images to a new domain [13], [14]. However, these methods are designed to perform aligned appearance transformations and struggle when views drastically change [15]. The latter work, in which a synthetic dataset with *perfect* ground-truth labels is used to learn IPM, is closest to ours.

We demonstrate in this paper that we are able to generate reliable, improved IPM for larger scenes than in [15], which are therefore able to directly aid scene understanding tasks. We achieve this in real time using real-world data collected under different conditions with a single front-facing camera. Consequently, we must deal with *imperfect* training labels (see Section IV) created from a sequence of images and ego-motion. An Incremental Spatial Transformer GAN is introduced to address the significant appearance change between the frontal view and IPM. Compared to analytic IPM approaches our learned model is (1) more realistic with sharper contours at long distance, (2) invariant to extreme illumination under different conditions, and (3) removes dynamic objects from the scene to recover the underlying road layout. We make the following contributions in this paper:

- we introduce an Incremental Spatial Transformer GAN for generating boosted IPM in real time;
- we explain how to create a dataset for training IPM methods on real-world images under different conditions; and
- we demonstrate that our boosted IPM approach improves the detection of road markings as well as the semantic interpretation of road scenes in the presence of occlusions and/or extreme illumination.

II. Related Work

Improved IPM As indicated in Section I, many applications can be found in the literature that apply IPM. They rely on three assumptions: (1) the camera is in a fixed position with respect to the road, (2) the road surface is planar, and (3) the road surface is free of obstacles. Remarkably, relatively few approaches exist that aim to improve inaccurate IPM, in case one or more of these assumptions are not satisfied.

Several works have tried to adjust for inaccuracies caused by invalidity of the first two assumptions. The authors of [16], [17] used vanishing point detection, [18] estimated the slope of the road according to the lane markings, and [19] employed motion estimation obtained from SLAM. Invalidity of the third assumption is tackled in [20] by using a laser scanner to exclude obstacles from being transformed to IPM. Another approach [21]–[23] creates a look up table for all pixels, by taking into account the distance of objects on the road surface, in order to reduce artefacts at further distance. However, these methods generally assume simple environments (i.e. highway). Contrarily, we learn a non-linear mapping more suited for urban scenes.

Very recently, [15] proposed the first learning approach for IPM using a synthetic dataset. The authors introduced BridgeGAN which employs the homography IPM to bridge the significant appearance gap between the frontal view and bird's-eye view. In contrast, we use real-world data and consequently *imperfect* labels to generate boosted IPM for larger scenes. Therefore, our learned mapping is directly beneficial for scene understanding tasks (see Section V-B).

Semantic IPM Several methods use the semantic relations between the two views for different tasks. In [24], [25] conditional random fields in the frontal view and IPM are optimized to retrieve a coarse semantic bird's-eye-view map from a sequence of camera images. A joint optimization net is trained in [26], [27] to align the semantic cues of the two views. The authors then train a GAN to synthesize a ground-level panorama from the coarse semantic segmentation. However, because aerial images differ significantly in appearance from the ground view, there is a lack of texture and detail in the synthesized images. We generate a more detailed IPM by learning a direct mapping of the pixels from the frontal view which is more useful for autonomous driving applications.

GANs for Novel View Synthesis The rise of GANs has made it possible to generate new, realistic images from a learned distribution. In order to guide the generation process towards a desired output, GANs can be conditioned on an input image [13], [28]. Until now, these methods were restricted to perform aligned appearance transformations.

In [29], the spatial transformer module was introduced to learn transformations of the input to improve classification tasks. The authors of [30], [31] used similar ideas to synthesize new views of 3D objects or scenes. More recently, these two fields were combined in [32], [33]. In the latter work, realistic compositions of objects are generated for a new viewpoint. However, these techniques are limited to toy datasets or distort real-world scenes with dynamic objects.

III. Boosted IPM using an Incremental Spatial Transformer GAN

A. Network Overview

As a starting point, we use a state-of-the-art architecture similar to the global enhancer of [28], without employing feature or instance maps. Additionally, as we expect a slight change in scale from the homography-based IPM image to the stitched training labels (see Section IV), we refrain from using any pixel-wise losses and instead use multi-scale discriminator losses [28] combined with a perceptual loss [34], [35] based on VGG16 [36]. While VGG16 is trained on the ImageNet [37] dataset, thus being more suitable for frontal rather than bird's-eye-view images of road scenes, we still leverage the stability of its encoded features in this study. Retraining VGG16 on bird's-eye-view images of road scenes or swapping it out for a more suitable model, may improve

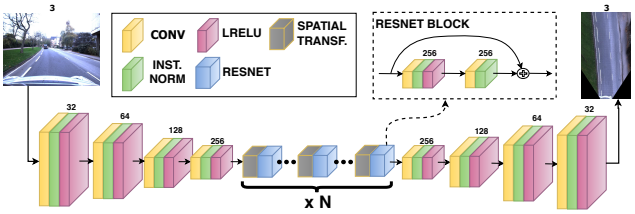


Fig. 2. The architecture of the generator of the network. The bottleneck of the model contains a series of N sequential blocks. Each block performs an incremental perspective transformation of n degrees, so that the bottleneck as a whole transforms the features from frontal to bird's-eye view. After every transformation, the features are sharpened by a ResNet block before the next transformation is applied. This process is depicted in more detail in Figure 3.

the quality of generated images, but this is beyond the scope of this study.

Our model follows a largely traditional downsample-bottleneck-upsample architecture, where we reformulate the bottleneck portion of the model as a series of N_{STRes} blocks that perform incremental perspective transformations followed by feature enhancement. Each block contains a Spatial Transformer (ST) [29] followed by a ResNet layer [38]. The structure of the generator is presented in Fig. 2. For an in-depth description of the remaining architecture, the reader is directed towards the paper and supplemental material of [28].

B. Spatial ResNet Transformer

Since far-away real-world features are represented by a smaller pixel area as compared to identical close-by features, a direct consequence of applying a full perspective transformation to the input is increased unnatural blurring and stretching of features af further distance. To counteract this effect, our model divides the full perspective transformation into a series of N_{STRes} smaller incremental perspective transformations, each followed by a refinement of the transformed feature space using a ResNet block [38]. The intuition behind this is that the slight blurring that occurs as a result of each perspective transformation is restored by the ResNet block that follows it, as conceptually visualized in Fig. 3. To maintain the ability to train our model end-to-end, we apply these incremental transforms using a Spatial Transformer [29].

Intuitively, a Spatial Transformer is a mechanism, which can be integrated in a deep-learning pipeline, that warps an image using a parametrization (e.g. an affine or homography transformation matrix) conditioned on a specific input signal. Formally, each incremental spatial transformer is an end-to-end differentiable sampler, represented in our case by two major components:

- a convolutional network which receives an input I of size $H_I * W_I * C$, where H_I, W_I and C represent the height, width, and number of channels of the input respectively, and outputs a parametrization M_{loc} of a perspective transformation of size $3 * 3$, and;
- a Grid Sampler which takes I and M_{loc} as inputs, creates a mapping matrix M_{map} of size $H_O * W_O * 2$, where H_O and W_O represent the height and width of the output O. M_{map} maps homogeneous coordinates $[x, y, 1]^T$ to their new warped position given by $M_{\text{loc}} * [x, y, 1]^T$. Finally M_{map} is used to construct O in the following way: $O(x, y) = I(M_{\text{map}}(x, y, 1), M_{\text{map}}(x, y, 2))$.

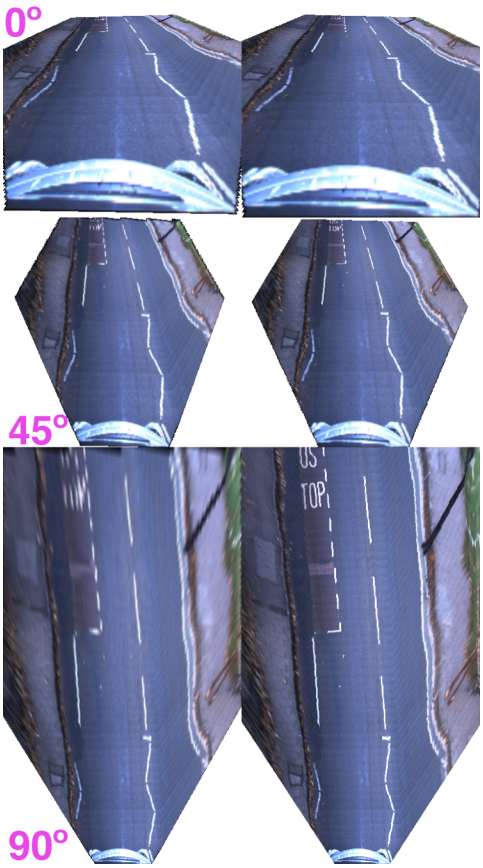


Fig. 3. Conceptual visualization of the sequential incremental transformations (i.e. $N = 3$, from $0°$ to $90°$ degrees down the rows) occurring in the bottleneck of the generator. The left column shows the features immediately after the transformation is applied, consequently they are stretched and blurred (e.g. BUS STOP letters). The right column shows how the ResNet blocks learns to sharpen these features to create the improved IPM before the next transformation is applied. Note that in reality the bottleneck has 512 feature maps instead of the 3 RGB channels depicted here for demonstration purposes.

In practice, it is non-trivial to train a spatial transformer (and even less trivial; a sequence of spatial transformers) on inputs with a large degree of self-similarity, such as road scenes. To stabilize the training procedure, for each incremental spatial transformer, we decompose $M_{\text{loc}} = M_{\text{locref}} * M_{\text{locpert}}$, where M_{locref} is initialized with an approximate parametrization of the desired incremental homography, and M_{locpert} is the actual output of the convolutional network and represents a learned perturbation or refinement of M_{locref}.

C. Losses

Our architecture stems from [28], but does not make use of any instance feature maps. Due to the potential misalignment between the output of the network and the labels, we rely on

a multi-scale discriminator loss and a perceptual loss based on VGG16. With a generator G, k^{th} scale discriminators D_k, and $\mathcal{L}_{\text{GAN}}(G, D_k)$ being the traditional GAN loss defined over $k = 3$ scales as in [28], the final objective thus becomes:

$$\mathcal{L}_{\text{tot}} = \min_{G}((\max_{D_1,D_2,D_3} \sum_{k=1,2,3} \mathcal{L}_{\text{GAN}}(G, D_k)) + \lambda_{\text{FM}} \sum_{k=1,2,3} \mathcal{L}_{\text{FM}}(G, D_k) + \lambda_{\text{VGG}} \mathcal{L}_{\text{VGG}}(G)), \quad (1)$$

where $\mathcal{L}_{\text{FM}}(G, D_k)$ is the multi-scale discriminator loss:

$$\mathcal{L}_{\text{FM}(G,D_k)} = \sum_{i=1}^{l_D} \frac{1}{w_i} \| D_k(I_{\text{label}})_i - D_k(G(I_{\text{input}}))_i \|_1 \quad (2)$$

and $\mathcal{L}_{\text{VGG}}(G)$ is the perceptual loss:

$$\mathcal{L}_{\text{VGG}(G)} = \sum_{i=1}^{l_P} \frac{1}{w_i} \| \text{VGG}(I_{\text{label}})_i - \text{VGG}(G(I_{\text{input}}))_i \|_1, \quad (3)$$

with l_D denoting the number of discriminator layers used in the discriminator loss, l_P denoting the number of layers from VGG16 that are utilized in the perceptual loss, and I_{input} and I_{label} being the input and label images, respectively. The weights $w_i = 2^{l-i}$ are used to scale the importance of each layer used in the loss.

D. Implementation details

We choose $N_{\text{STRes}} = 6$, $N_{\text{downsample}} = 4$, $N_{\text{upsample}} = 4$ and $l_D = l_P = 4$. Furthermore, for training, we employ the Adam solver using a base learning rate set at 0.0002, and a batch size of 1, training for 200 epochs. For the loss trade-off, we empirically set $\lambda_{\text{FM}} = 5$ and $\lambda_{\text{VGG}} = 2$. We train our network using 8416 overcast and 4894 nighttime labels. At run time, the network performs inference in real time (≈ 20 Hz) using an NVIDIA TITAN X.

IV. Creating Training Data for Boosted IPM

To evaluate our approach, we use the Oxford RobotCar Dataset [39], which features a 10-km route through urban environments under different weather and lighting conditions.

In order to create training labels which are a better representation of the real world than the standard, homography-based IPM, we use a sequence of images from the front-facing camera and corresponding visual odometry [40], and merge them into a single bird's-eye-view image.

From the sensor calibrations and the camera's intrinsic parameters, we compute the transformation which defines the one-to-one mapping between the pixels of the front-facing camera and the bird's-eye view. Then, using the relative transform obtained by visual odometry between the current image frame of the sequence and the initial frame, we stitch the respective pixels of the current frame into the IPM image at the correct pixel positions. This operation is performed iteratively, overwriting previous IPM pixels with more accurate pixels of subsequent frames, until the vehicle has reached the end of its field of view of the initial image.

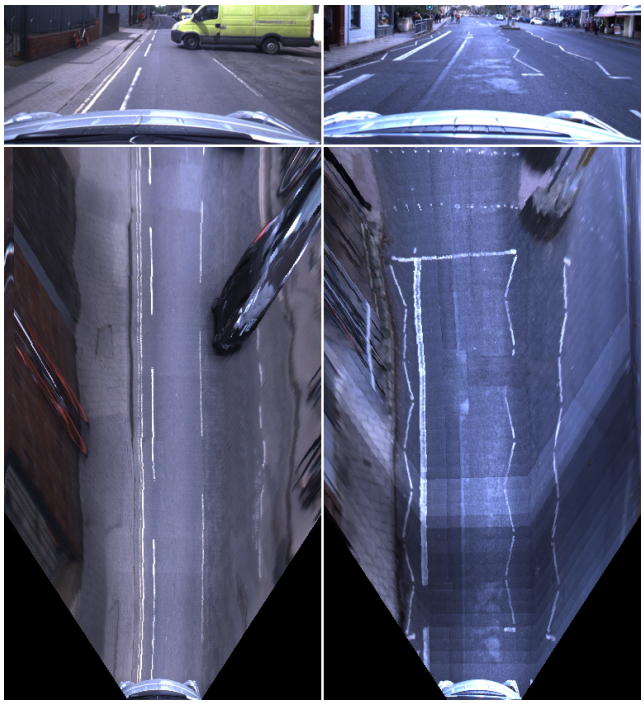


Fig. 4. Examples of created training pairs (which show the difficulties of using real-world data) by stitching IPM images generated from future front-facing camera images using the ego-motion obtained from visual odometry. The *left* example illustrates 1) movement of dynamic objects by the time the images are stitched and 2) stretching of objects because they are assumed to be on the road surface. The *right* example shows a significant change of illumination conditions. *Both* show inaccuracies at further lateral distance (e.g. wavy curb) because of sloping road surface and possibly imprecise motion estimation.

As the training labels are created from real-world data (in contrast to the synthetic data of [15]), their quality is limited by several aspects (see examples in Fig. 4):

- Minor inaccuracies in the estimation of the rotation of the vehicle and sloping road surface can lead to imprecise stitching at further lateral distance.
- Consecutive image frames may vary significantly in terms of lighting (e.g. due to overexposure), leading to illumination differences in the label which do not naturally occur in the real-world.
- Dynamic objects in the front-facing view will appear in a different position in future frames. Consequently, they will appear in unexpected places in the label.
- Objects above the road plane (e.g. vehicles, bicyclists, intersection islands, etc.) undergo a large deformation due to the view transformation. We cannot obtain accurate labels for these in real-world scenarios.

Due to the aforementioned drawbacks, no direct relation exists between the output (boosted IPM) of our network and the stitched labels. Therefore, it is impossible to incorporate a direct pixel-wise loss function, or employ super-resolution generating networks such as [41]. On the other hand, since we use a sequence of future images, regions that were previously occluded by (dynamic) objects in the initial view are potentially revealed later. This gives the network the

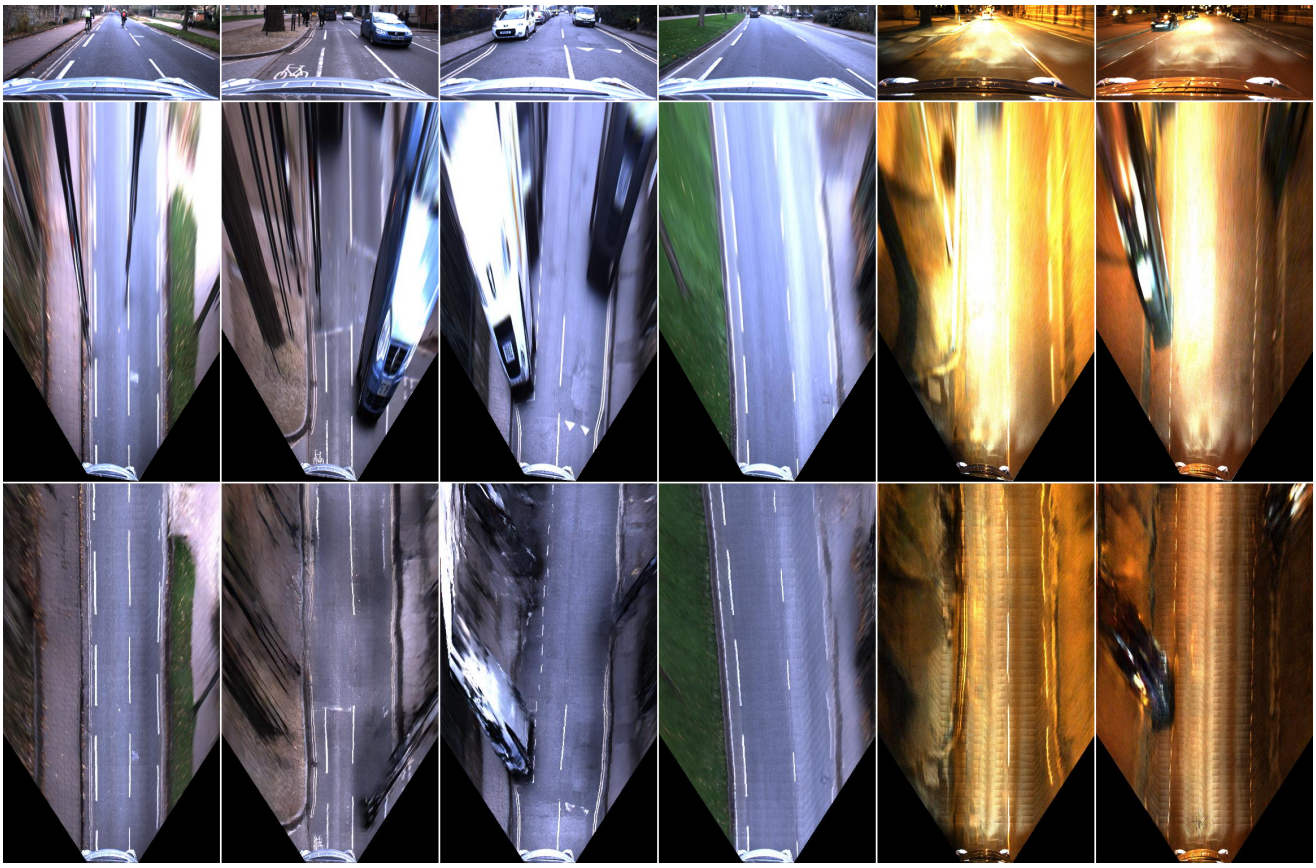


Fig. 5. Boosted IPM generated by the network (*bottom*) under different conditions compared to traditional IPM generated by applying a homography (*middle*) to the front-facing camera image (*top*). The boosted birds-eye-view images contain sharper features (e.g. road markings), more homogeneous illumination, and automatically remove (dynamic) objects from the scene. Consequently, we infer the underlying road layout, which is directly beneficial for various tasks performed by autonomous vehicles.

ability to learn the underlying road layout irrespective of occlusions or extreme illumination.

V. Experimental Results

In this section we present qualitative results generated under different conditions. Due to the nature of the problem, it is extremely hard to capture ground-truth labels in the real world (see Section IV), and thus to present quantitative results for our approach. Furthermore, the synthetic dataset used in [15] is not publicly available. However, we demonstrate that our boosted IPM has a significant qualitative effect on the semantic interpretation of real-world scenes. Lastly, we show some limitations of the presented framework.

A. Qualitative Evaluation

Fig. 5 shows qualitative results on a RobotCar test dataset. The results demonstrate that the network has learned the underlying road layout of various urban traffic scenarios. Semantic road features such as parking boxes (i.e. small separators) and stop lines are inferred correctly. Furthermore, dynamic objects, which occlude parts of the scene, are removed and replaced by the correct road/lane boundaries, making the representation more suitable for scene understanding and planning. The boosted IPM contains sharper road markings, which improves the performance of tasks such as lane detection. Lastly, the new view offers a more homogeneous illumination of the road surface, which is beneficial for all tasks that require image processing.

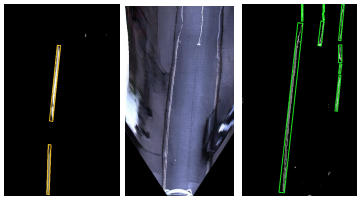
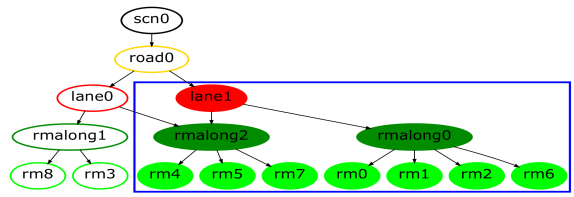
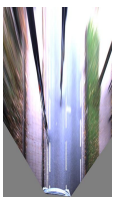
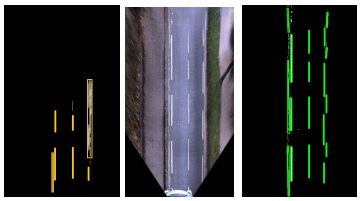
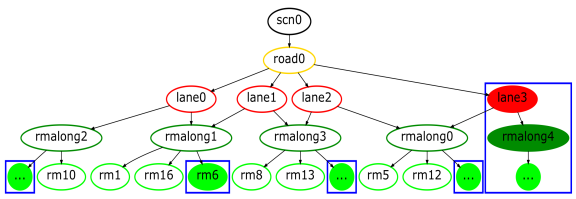
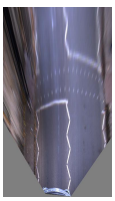
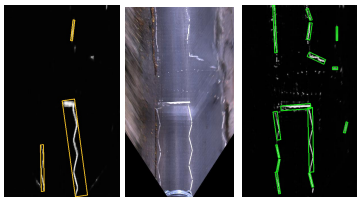
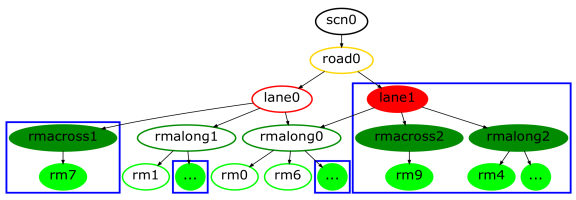
Additionally, we show that our framework is not limited to datasets recorded under overcast conditions. Although artificial lighting during night time introduces colour variations in the output, we are still able to significantly improve the representation of the underlying layout of the scene.

B. Employing Boosted IPM for Scene Interpretation

We demonstrate the effectiveness of our improved IPM approach for the application of road marking detection [42] and scene interpretation [43] (cf. Table I). Table I shows the original front-facing camera image, the bird's-eye views (homography-based as well as our boosted IPM) and their corresponding road marking detections, and the generated graph-based scene description.

The input to the scene interpretation process is the binary image mask of the detected road markings. Within these experiments this input is either provided by the homography-based IPM or by our boosted IPM. We then cluster the road marking pixels into groups and compute a set of spatial

TABLE I
QUALITATIVE EFFECTS OF IPM METHODS ON ROADMARKING DETECTION AND SCENE INTERPRETATION

Original	Road marking Detection [42]		Scene Interpretation [43]
	Homography	Boosted IPM	(generated from detected road markings)
			
			
			

properties and relations. Based on the spatial information and a learned probabilistic grammar, which captures the road layout of scenes, we generate a hierarchical, graph-based scene description including information about roads, lanes and road markings (which are grounded in image space). The reader is directed towards [43] for more details.

As the overall scene interpretation is based on the segmentation of road markings, the quality of the road marking detection has a major impact on the generated scene graph, as we will demonstrate later. Experimentally, we have verified that boosted IPM allows us to more robustly detect road markings (1) at greater distance and (2) in more detail, and (3) infer road markings occluded by dynamic objects such as cars and cyclists. These improvements are possible because boosted IPM contains sharper features with more consistent geometric properties (at further distance) and learns the underlying road layout.

We have trained a road marking detection network for each view separately with an equivalent setup according to [42]. The increase in performance for road marking detection has immediate consequences for the interpretation of scenes. In general, all interpretations (scene graphs) benefit from more accurate road marking detection. Table I depicts qualitative differences in the scene graphs[1]. In the following we discuss the individual scenes.

Scene (A) The vehicle approaches a pedestrian crossing which is signaled by the upcoming zig-zag lines (visible at the top of the image). While these road markings are visible to the human eye in the homography-based IPM, the trained road marking detection network was not able to detect them because of the stretching and blurring at further distance. However, our boosted IPM produced a bird's-eye-view image with sharper contours for the zig-zag lines and correct reconstruction of the road markings occluded by the vehicle. This resulted in an improved scene graph which not only captured the right boundary of the ego lane, but also a previously undetected second lane on the right. Such qualitative differences have substantial impact on the planning and decision making of the vehicle.

Scene (B) The vehicle drives on a road with four lanes — two inner lanes for vehicles and two outer lanes for cyclists — and experiences a change in illumination (from a darker foreground to a brighter background). This is clearly visible in the homography-based IPM and consequently leads to a poor detection of road markings. In contrast, our boosted approach produces a top-down view which inpaints learned semantic cues (i.e. road markings) directly over the overexposed area and also excludes the two cyclists. Hence, the resulting scene graph captures more detail as well as an extra lane which was missed in the segmentation resulting from the standard approach.

Scene (C) The vehicle approaches a pedestrian crossing which is indicated by both zig-zag and stop lines. Again, the distorted and blurry image resulting from the homography-based IPM leads to a poor detection of road markings. Our boosted approach has generated a more detailed view which led to better road marking detection including the successful

[1]In the scene graphs, the qualitative differences resulting from our boosted IPM method are indicated by filled nodes grouped in blue boxes.

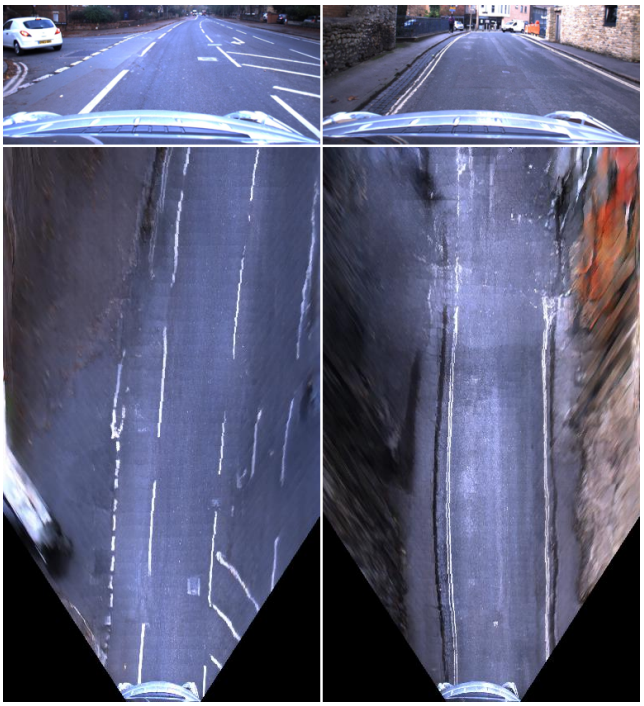


Fig. 6. Two cases in which the output of the network does not accurately depict the top-down view of the scene. In the *left* image, the road marking arrow is deformed, because we cannot employ a pixel-wise loss. In the *right* image, the road surface is not flat (sloping upwards), consequently the spatial transformer blocks attempt to map parts of the scene above the horizon, for which the features are not learned.

identification of the stop lines. The resulting scene graph based on the homography-based IPM not only misses a lane, but crucially also both stop lines.

Such qualitative differences clearly demonstrate the advantage of our proposed method as they have a direct impact on planning and decision making of autonomous vehicles. While the detection and interpretation of road markings at a greater distance will enable an autonomous vehicle to adapt its behaviour earlier, the detection of road markings behind moving objects will lead to performance that is more robust and safer even when the scene is partly occluded.

C. Failure Cases

Under certain conditions, the boosted IPM does not accurately depict all details of the bird's-eye view of the scene.

As we cannot enforce a pixel-wise loss during training (Section IV), the shape of certain road markings is not accurately reflected (illustrated in Fig. 6). Improvement of the representation of these structural elements will be investigated in future work.

Furthermore, the spatial transformer blocks assume that the road surface is more or less planar (and perpendicular to the z-axis of the vehicle). When this assumption is not satisfied, the network is unable to accurately reflect the top-down scene at further distance. This might be solved by providing/learning the rotation of the road surface with respect to the vehicle.

VI. Conclusion

We have presented an adversarial learning approach for generating boosted IPM from a single front-facing camera image in real time. The generated results show sharper features and a more homogeneous illumination, while (dynamic) objects are automatically removed from the scene. Overall, we infer the underlying road layout, which is directly beneficial for tasks performed by autonomous vehicles such as road marking detection, object tracking, and path planning.

In contrast to existing approaches, we used real-world data collected under different conditions, which introduced additional issues due to varying illumination and (dynamic) objects, making it impossible to employ a pixel-wise loss during training. We have addressed the significant appearance change between the views by introducing an Incremental Spatial Transformer GAN.

We have demonstrated reliable, qualitative results in different environments and under varying lighting conditions. Furthermore, we have shown that the boosted IPM view allows for improved hierarchical scene understanding.

Consequently, our boosted IPM approach can have a significant impact on a wide range of applications in the context of autonomous driving including scene understanding, navigation, and planning.

References

[1] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 12, pp. 2481–2495, Dec 2017.

[2] L. Schneider, M. Cordts, T. Rehfeld, D. Pfeiffer, M. Enzweiler, U. Franke, M. Pollefeys, and S. Roth, "Semantic stixels: Depth is not enough," in *2016 IEEE Intelligent Vehicles Symposium (IV)*, June 2016, pp. 110–117.

[3] D. Neven, B. De Brabandere, S. Georgoulis, M. Proesmans, and L. Van Gool, "Towards end-to-end lane detection: An instance segmentation approach," *arXiv preprint arXiv:1802.05591*, 2018.

[4] W. Song, Y. Yang, M. Fu, Y. Li, and M. Wang, "Lane detection and classification for forward collision warning system based on stereo vision," *IEEE Sensors Journal*, vol. 18, no. 12, pp. 5151–5163, June 2018.

[5] B. Mathibela, P. Newman, and I. Posner, "Reading the road: Road marking classification and interpretation," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 4, pp. 2072–2081, Aug 2015.

[6] A. L. Ballardini, D. Cattaneo, S. Fontana, and D. G. Sorrenti, "An online probabilistic road intersection detector," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, May 2017, pp. 239–246.

[7] S. Schulter, M. Zhai, N. Jacobs, and M. Chandraker, "Learning to look around objects for top-view representations of outdoor scenes," *arXiv preprint arXiv:1803.10870*, 2018.

[8] J. Dequaire, P. Ondrka, D. Rao, D. Wang, and I. Posner, "Deep tracking in the wild: End-to-end tracking using recurrent neural networks," *The International Journal of Robotics Research*, vol. 37, no. 4-5, pp. 492–512, 2018.

[9] N. Engel, S. Hoermann, P. Henzler, and K. Dietmayer, "Deep object tracking on dynamic occupancy grid maps using RNNs," *arXiv preprint arXiv:1805.08986*, 2018.

[10] N. Simond and M. Parent, "Obstacle detection from ipm and super-homography," in *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct 2007, pp. 4283–4288.

[11] N. Lee, W. Choi, P. Vernaza, C. B. Choy, P. H. S. Torr, and M. Chandraker, "Desire: Distant future prediction in dynamic scenes with interacting agents," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017, pp. 2165–2174.

[12] A. Zyner, S. Worrall, and E. Nebot, "Naturalistic driver intention and path prediction using recurrent neural networks," *arXiv preprint arXiv:1807.09995*, 2018.

[13] P. Isola, J. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017, pp. 5967–5976.

[14] J. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *2017 IEEE International Conference on Computer Vision (ICCV)*, Oct 2017, pp. 2242–2251.

[15] X. Zhu, Z. Yin, J. Shi, H. Li, and D. Lin, "Generative adversarial frontal view to bird view synthesis," in *2018 International Conference on 3D Vision (3DV)*, Sep. 2018, pp. 454–463.

[16] M. Nieto, L. Salgado, F. Jaureguizar, and J. Cabrera, "Stabilization of inverse perspective mapping images based on robust vanishing point estimation," in *2007 IEEE Intelligent Vehicles Symposium*, June 2007, pp. 315–320.

[17] D. Zhang, B. Fang, W. Yang, X. Luo, and Y. Tang, "Robust inverse perspective mapping based on vanishing point," in *Proceedings 2014 IEEE International Conference on Security, Pattern Analysis, and Cybernetics (SPAC)*, Oct 2014, pp. 458–463.

[18] M. Bertozzi, A. Broggi, and A. Fascioli, "An extension to the inverse perspective mapping to handle non-flat roads," in *IEEE International Conference on Intelligent Vehicles. Proceedings of the 1998 IEEE International Conference on Intelligent Vehicles*, vol. 1, 1998.

[19] J. Jeong and A. Kim, "Adaptive inverse perspective mapping for lane map generation with slam," in *2016 13th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)*, Aug 2016, pp. 38–41.

[20] M. Oliveira, V. Santos, and A. D. Sappa, "Multimodal inverse perspective mapping," *Information Fusion*, vol. 24, pp. 108 – 121, 2015.

[21] C.-C. Lin and M.-S. Wang, "A vision based top-view transformation model for a vehicle parking assistant," *Sensors*, vol. 12, no. 4, pp. 4431–4446, 2012.

[22] P. Cerri and P. Grisleri, "Free space detection on highways using time correlation between stabilized sub-pixel precision ipm images," in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, April 2005, pp. 2223–2228.

[23] J. M. M. García and N. Y. Ershadi, "A new strategy of detecting traffic information based on traffic camera : modified inverse perspective mapping," *Journal of Electrical Engineering, Technology and Interface Utilities*, vol. 10, no. 2, pp. 1101–1118, March 2017.

[24] S. Sengupta, P. Sturgess, L. Ladick, and P. H. S. Torr, "Automatic dense visual semantic mapping from street-level imagery," in *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct 2012, pp. 857–862.

[25] G. Máttyus, S. Wang, S. Fidler, and R. Urtasun, "Hd maps: Fine-grained road segmentation by parsing ground and aerial images," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016, pp. 3611–3619.

[26] M. Zhai, Z. Bessinger, S. Workman, and N. Jacobs, "Predicting ground-level scene layout from aerial imagery," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017, pp. 4132–4140.

[27] K. Regmi and A. Borji, "Cross-view image synthesis using geometry-guided conditional gans," *arXiv preprint arXiv:1808.05469*, 2018.

[28] T. Wang, M. Liu, J. Zhu, A. Tao, J. Kautz, and B. Catanzaro, "High-resolution image synthesis and semantic manipulation with conditional gans," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, June 2018, pp. 8798–8807.

[29] M. Jaderberg, K. Simonyan, A. Zisserman *et al.*, "Spatial transformer networks," in *Advances in neural information processing systems*, 2015, pp. 2017–2025.

[30] X. Yan, J. Yang, E. Yumer, Y. Guo, and H. Lee, "Perspective transformer nets: Learning single-view 3d object reconstruction without 3d supervision," in *Advances in Neural Information Processing Systems*, 2016, pp. 1696–1704.

[31] T. Zhou, S. Tulsiani, W. Sun, J. Malik, and A. A. Efros, "View synthesis by appearance flow," in *European conference on computer vision*. Springer, 2016, pp. 286–301.

[32] D. J. Rezende, S. A. Eslami, S. Mohamed, P. Battaglia, M. Jaderberg, and N. Heess, "Unsupervised learning of 3D structure from images," in *Advances in Neural Information Processing Systems*, 2016, pp. 4996–5004.

[33] S. Azadi, D. Pathak, S. Ebrahimi, and T. Darrell, "Compositional GAN: Learning conditional image composition," *arXiv preprint arXiv:1807.07560*, 2018.

[34] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in *European Conference on Computer Vision*. Springer, 2016, pp. 694–711.

[35] A. Dosovitskiy and T. Brox, "Generating images with perceptual similarity metrics based on deep networks," in *Advances in Neural Information Processing Systems*, 2016, pp. 658–666.

[36] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[37] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[38] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016, pp. 770–778.

[39] W. Maddern, G. Pascoe, C. Linegar, and P. Newman, "1 year, 1000 km: The oxford robotcar dataset," *The International Journal of Robotics Research*, vol. 36, no. 1, pp. 3–15, 2017.

[40] W. Churchill, "Experience based navigation: Theory, practice and implementation," Ph.D. dissertation, University of Oxford, Oxford, United Kingdom, 2012.

[41] C. Ledig, L. Theis, F. Huszr, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi, "Photo-realistic single image super-resolution using a generative adversarial network," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017, pp. 105–114.

[42] T. Bruls, W. Maddern, A. A. Morye, and P. Newman, "Mark yourself: Road marking segmentation via weakly-supervised annotations from multimodal data," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, May 2018, pp. 1863–1870.

[43] L. Kunze, T. Bruls, T. Suleymanov, and P. Newman, "Reading between the lanes: Road layout reconstruction from partially segmented scenes," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, Nov 2018, pp. 401–408.

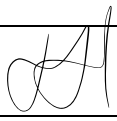
Statement of Authorship for joint/multi-authored papers for PGR thesis

To appear at the end of each thesis chapter submitted as an article/paper

The statement shall describe the candidate's and co-authors' independent research contributions in the thesis publications. For each publication there should exist a complete statement that is to be filled out and signed by the candidate and supervisor **(only required where there isn't already a statement of contribution within the paper itself).**

Title of Paper	The Right (Angled) Perspective: Improving the Understanding of Road Scenes Using Boosted Inverse Perspective Mapping
Publication Status	Published
Publication Details	T. Bruls*, **H. Porav***, L. Kunze and P. Newman, "The Right (Angled) Perspective: Improving the Understanding of Road Scenes Using Boosted Inverse Perspective Mapping," *2019 IEEE Intelligent Vehicles Symposium (IV)*, Paris, France, 2019, pp. 302-309, doi: 10.1109/IVS.2019.8814056. ***equal contribution**

Student Confirmation

Student Name:	Horia Porav		
Contribution to the Paper	Equal contribution by myself and the first author. My contributions included: - Generating ideas related to the use of incremental denoising - Introducing the Spatial ResNet Transformer to improve IPM images - Implementing the incremental denoising network - Generating initial training data from simulation to validate the concept - Training the network - Generating the results - Writing the Methodology and Implementation details and equally contributing to the rest of the sections in the paper		
Signature		Date	03 October 2020

Supervisor Confirmation

By signing the Statement of Authorship, you are certifying that the candidate made a substantial contribution to the publication, and that the description described above is accurate.

Supervisor name and title: Professor Paul Newman			
Supervisor comments			
Signature		Date	05-10-2020

This completed form should be included in the thesis, at the end of the relevant chapter.

5
Data generation

Contents

5.1 Rainy screens: Collecting rainy datasets, indoors . . . 76
5.2 Reducing steganography in cycle-consistency GAN architectures . . . 84
5.3 Generating All the Roads to Rome . . . 91
5.4 Imminent Collision Mitigation . . . 101

5.1 Rainy screens: Collecting rainy datasets, indoors

Problem: Acquisition of data with adverse conditions in robotics is a cumbersome task due to the difficulty in guaranteeing proper ground truth and synchronising with desired weather conditions. In our previous work on de-raining images, we built a stereo rig where one lens was affected by adherent droplets and the other lens was kept clean. The rig was mounted on a data collection vehicle which allowed for potentially limitless data acquisition, but two drawbacks existed: the actual weather needed to be favourable in order to allow the acquisition of clear ground-truth images, and furthermore the ground-truth was limited to clean images, without the benefit of semantic segmentation or image annotations (unless the images were to be manually labelled afterwards). Additionally, even though our stereo baseline was small, our homography-based image alignment exhibited a small degree of misalignment.

Solution: To further simplify the data collection process, reduce the misalignment issues and potentially extend the types of ground-truth available, we built a static monocular rig that is able to record images shown on a high-resolution computer monitor through a transparent surface that is affected by adherent droplets (and potentially other contaminants such as dirt). This allowed us to fully automate the data acquisition process, and more importantly offered the opportunity to process datasets that already contain an associated ground-truth (e.g. segmentation maps, object annotations etc.). Additionally, since the images appeared on an actual flat surface, our use of a homography to register the images was formally correct, yielding a much smaller degree of misalignment between the rain-enhanced images and their (extended, varied) ground-truth.

Status: Published on ArXiv preprint

Rainy screens: Collecting rainy datasets, indoors.

Horia Porav
Oxford Robotics Institute,
University of Oxford
horia@robots.ox.ac.uk

Valentina-Nicoleta Musat
Oxford Brookes University
16075442@brookes.ac.uk

Tom Bruls
Oxford Robotics Institute,
University of Oxford
tombruls@robots.ox.ac.uk

Paul Newman
Oxford Robotics Institute,
University of Oxford
pnewman@robots.ox.ac.uk

Abstract

Acquisition of data with adverse conditions in robotics is a cumbersome task due to the difficulty in guaranteeing proper ground truth and synchronising with desired weather conditions. In this paper, we present a simple method - recording a high resolution screen - for generating diverse rainy images from existing clear ground-truth images that is domain- and source-agnostic, simple and scales up. This setup allows us to leverage the diversity of existing datasets with auxiliary task ground-truth data, such as semantic segmentation, object positions etc. We generate rainy images with real adherent droplets and rain streaks based on Cityscapes and BDD, and train a de-raining model. We present quantitative results for image reconstruction and semantic segmentation, and qualitative results for an out-of-sample domain, showing that models trained with our data generalize well.

1. Introduction

Bad weather conditions are among the biggest challenges in robotics. Fog, snow, haze, rain and dirt can affect both the visual perception of humans and severely diminish the performance of computer vision tasks.

The effects of impurities on visual perception are intricate, but can be classified in two main categories. Firstly, by impurities in the atmosphere that occlude the scene (snow, rain streaks, fog, smog) but which have no distorting effects. Secondly, by impurities that adhere to transparent surfaces and have distorting effects. For example, adherent rain drops act as a fish-eye-lens due to their convex shape, with the image formed inside the droplet being a result of light rays coming from an area that is larger than what the droplet is occluding.



Figure 1. Top row: rainy image and its semantic segmentation. Bottom row: derained image and its semantic segmentation. The rainy image was obtained from a clear, dry-condition dataset (Cityscapes), using an indoor, automated data collection rig consisting of a high-resolution monitor, a DSLR camera, glass panes and water spray nozzles. This setup facilitates the creation of high-quality rainy datasets complete with a perfectly aligned clear, dry groundtruth and **any** other attributes of the original dataset such as semantic segmentation labels, depth etc. Once set up, our method provides automatic, unsupervised image capture, with a wide variety of water droplet sizes and streak patterns.

Modelling adherent droplets is a difficult task due to the various sizes and shapes that they can take, distortion, zoom, blur, glare, the effect of gravity, and cohesion and adhesion of water (27). Moreover, although the scene is in focus, the images formed inside droplets in front of the camera is usually blurry and opaque. Finally, if the light refracted inside the droplet comes from a large part of the scene, the image formed in the droplet becomes gray (21).

To the best of our knowledge, there are three main methods through which rainy images with ground-truth are gen-

erated in existing literature. The first is manual, and involves using a glass pane to capture a clean and a rainy image sequentially (16). While this offers perfect alignment, it is tedious, and does not scale up to dynamic scenes due to motion and changes in illumination. The second method involves using stereo cameras, where one lens is kept clean and one rainy, with both images captured simultaneously (15). Although the scenes are more diverse and the method scales up, it assumes that the two images can be aligned using a homography which, unfortunately, only works on planar scenes. The third method involves computer-generated rain, which scales up, but there is a large domain gap between real rain and computer-generated rain (15).

Building datasets with both real rain and clear ground truth under hard-to-control outdoor environments is very difficult for static images, let alone scenes containing moving objects. One has to account for changes in illumination, motion, occlusion/dis-occlusion, changes in appearance, and so on. For this reason, many of the datasets focus on static scenes, while those with video sequences do not offer non-rainy ground truth, apart from (15).

We propose a simple method for capturing diverse raindrops, that can make use of existing datasets with clear images and auxiliary task ground truth, but also of arbitrary footage. Our method is inspired by (15), which use a water pump to spray water on a glass pane, and (6), which use a windshield plane that can be tilted and translated in front of the camera. The difference is that we capture monocular images by recording a high-resolution screen. The advantages of this approach are threefold:

- perfectly aligned image pairs;
- high scalability, with no requirement for a mobile platform such as a vehicle since all data collection happens indoors, unsupervised, and in a controlled environment; and
- diverse, in and out of focus droplets and streaks.

In this way, we are able to easily generate image pairs with high diversity, and further demonstrate that data gathered indoors using this method can efficiently be used to both train and pre-train deraining models that generalize well.

2. Related work

Rain modelling: A photometric raindrop model is used by (6) and (17) to construct the appearance of raindrops by tracing the light rays that pass through them, with the droplet boundary modelled by a sphere with equal contact angles. In a follow-up paper (18), 2D Bezier curves were used to incorporate raindrop deformation due to the effect of gravity. Later in (28), the dark band around them is produced with the total reflection phenomenon (light from the environment being reflected back into the waterdrop). Based on (6), (17) and (28), the authors of (15) propose an approach similar to meta-balls (1), where they model synthetic waterdrops by warping normal maps of droplets.

Adding droplets: In the experimental setup of (4), rain is simulated by spraying water on a glass pane in front of the camera, but due to changes in illumination and subject motion, no ground truth is provided. Additionally, a video sequence with real rain is provided, without ground truth. The authors of (27) run experiments to study the effects of disturbance of light sources in environment, various shapes and sizes of raindrops, blurred raindrops and glare, in a droplet detection and removal pipeline in videos. They use a glass pane on which they spray water and capture videos in the real world, however ground truth is only available for the droplets' position.

Complete reconstruction ground truth (clean and rainy image pairs) was first attempted in (16). To ensure alignment and keep the same refractive index, they use both a sprayed glass pane and a clean glass pane. Additionally, they ensure that the atmospheric conditions and background remain constant during the two shots. They provide 1119 image pairs, however the process is difficult to scale up, and since background needs to be kept constant between the two shots, the images contain only static scenes.

On the other hand, the authors of (15) use a bi-partite chamber that is placed in front of a small-baseline stereo camera, keeping one part dry while the other part is sprayed with water. The system was mounted on a vehicle and image sequences were taken while driving. The authors provide 50000 pairs of undistorted, cropped and aligned images with both rain streaks and raindrops of various shapes and sizes. Although the system is portable and image acquisition is simpler, the method requires a mobile platform, and the alignment is based on a homography that assumes the world is flat, resulting in a small degree of misalignment between image pairs.

Raindrop detection and removal: The classical approaches for detecting and removing raindrops rely on a combination of image segmentation, pattern recognition, ellipse fitting, template matching (9), (19), (5), (17), (18), multiple image analysis (27), or multiple and/or rotating cameras (11), (25), (24) and (26). More recent approaches employ Convolutional Neural Networks and/or adversarial frameworks (4), (16), (13), while in (21) both classical methods and CNNs are combined.

Soil and dirt removal: Since cameras mounted outside the vehicle are exposed to adverse conditions, fully autonomous cars need a system to detect mud and trigger a cleaning mechanism. Recent work done in this area is that of (22), which create a dataset with opaque and transparent soiling used to train a soiling detection network. However, since the

acquisition of realistic muddy images is a difficult task, they propose a GAN-based architecture to generate artificial soil on clean images. Similarly, the authors of (4) generate synthetic dirt for training purposes, and validate their results on both synthetic dirt, and real dirt captured using a glass pane. The authors of (14) study how object detection is affected when degrading the image quality, by introducing synthetic fog, frost, snow and other corruptions and effects.

3. Methodology

3.1. Data acquisition

We build a rig consisting of a camera on a tripod, a tiltable glass pane simulating a car windshield, multiple water spray nozzles with a recirculating pump and collector tray, and a waterproofed high-resolution (2560x1440) computer monitor. The setup is presented in figure Fig 2.



Figure 2. Rain-maker camera rig. Note that the datasets are collected with the room lights OFF!

We start by calibrating our camera for the desired zoom level, and roughly align it such that its optical axis is normal to the screen. For our particular experiment, the surface of the screen was 320mm from the outer lens, the glass pane was offset 170mm from the screen surface, and tilted to 20 degrees, simulating a windshield. After setting the focus, we capture a test pattern that is used to estimate a homography between the monitor and the camera image. Since the image displayed on the screen already represents a 2D projection of the world, we have the advantage of being able to accurately align the camera image with the ground truth image using the estimated homography.

The first pass of data acquisition involves capturing clear images through the glass pane, without any water, in order to account for the refraction of the glass. The second pass involves turning on the water nozzles (with a randomized pressure regimen) and capturing the same images again, resulting in corresponding rainy data. The data acquisition process must take place in a dark room to minimize stray reflections and changes in illumination. All data acquisition is done automatically and unattended, at approximately 1 Hz, once the rig has been set up, and any misalignment is corrected using the computed homography. Note that the datasets are collected with the room lights OFF, without any natural or artificial lighting.

3.2. The deraining model

The principal purpose of our study is not to present a new architecture but to show that a high resolution monitor can be adequately used for data acquisition, hence we base our convolutional de-raining model on the widely-used architecture of Pix2PixHD (23), with 4 convolutional layers, 6 ResNet (7) blocks and 4 up-convolutional layers. We add additive skip connections between each down- and up-convolutional layer, observing that a large part of the input structure, illumination and details can be kept and copied at the output. Similar to (23) and (15), we do not make use of any unstructured pixel-wise losses, instead choosing a PatchGAN (12) adversarial loss combined with discriminator feature losses and a VGG-based perceptual loss.

3.3. Losses

On the generator output, we apply an adversarial loss:

$$\mathcal{L}_{adv} = (D(G(I_{\text{rainy}})) - 1)^2. \quad (1)$$

With the discriminator being trained to minimize:

$$\mathcal{L}_{disc} = (D(I_{\text{clear}}) - 1)^2 + (D(I_{\text{de-rained}}))^2, \quad (2)$$

A VGG-based perceptual loss (10) is additionally used:

$$\mathcal{L}_{\text{perc}} = \sum_{i=1}^{n_{VGG}} \frac{1}{w_i^{perc}} \| VGG(I_{\text{clear}})_i - VGG(G(I_{\text{rainy}}))_i \|_1 \quad (3)$$

with n_{VGG} denoting the number of layers used in the loss and $w_i^{perc} = 2^{(n_{VGG}-i)}$ weighing the contribution of each layer. We also use a multi-scale discriminator feature loss (23):

$$\mathcal{L}_{\text{msadv}} = \sum_{i=1}^{n_{ADV}} \frac{1}{w_i^{adv}} \| D(I_{\text{clear}})_i - D(G(I_{\text{rainy}}))_i \|_1, \quad (4)$$

with n_{ADV} denoting the number of discriminator layers used in the loss and $w_i^{adv} = 2^{(n_{ADV}-i)}$ weighing the contribution of each layer.

Finally, the full generator objective $\mathcal{L}_{\text{gen}}$ is:

$$\mathcal{L}_{\text{gen}} = \lambda_{\text{adv}} * \mathcal{L}_{\text{adv}} + \lambda_{\text{perc}} * \mathcal{L}_{\text{perc}} + \lambda_{\text{msadv}} * \mathcal{L}_{\text{msadv}} \quad (5)$$

The hyperparameters λ modulate the importance of the individual terms of the loss. We estimate a discriminator D and generator G such that:

$$G, D = \arg\min_{G,D} \mathcal{L}_{\text{gen}} + \mathcal{L}_{\text{disc}}. \quad (6)$$



Figure 3. Cityscapes-Rainy dataset. Left to right: Rainy, Clear ground-truth, Segmentation ground-truth.

Dataset	Model (ours) trained on	RAINY PSNR	RAINY SSIM	DERAINED PSNR	DERAINED SSIM
Cityscapes-Rainy	Cityscapes-Original labels	18.20	0.6865	**25.76**	**0.8817**
Cityscapes-Rainy	Cityscapes-Photographed labels	18.20	0.6865	**21.20**	**0.8294**
BDD-Rainy	Cityscapes-Original labels + BDD-Original labels	17.84	0.6500	**24.02**	**0.8587**

Table 1. Reconstruction results

3.4. Training

We employ a similar regimen to (15), training all models for 100 epochs, using the Adam optimizer, with a base learning rate of 0.0002.

4. Results

We collect rainy versions of Cityscapes(3) and BDD(29) datasets, and study reconstruction and segmentation quality.

4.1. Reconstruction results

Image reconstruction results are presented in Table 1. We train three different models, one using the original Cityscapes clear images as ground-truth (**Cityscapes-Original**), one using the photographed Cityscapes clear images as ground-truth (**Cityscapes-Photograph**), and one trained on both the the original Cityscapes and original BDD clear images. We present the deraining performance of these three models on both Cityscapes-Rainy and BDD-Rainy. All models significantly improve the quality of the rain-affected images. Secondly, Table 2 shows the performance of our model on a third-party dataset by Qian et al.(16), containing 861 training images and 59 testing images, acquired using a glass-pane manually sprayed with water and a DSLR camera in an outdoor environment. On one hand, we show that our model architecture achieves state-of-the-art results when trained on the target dataset. On the other hand, training solely on Cityscapes-Rainy does not produce top results. We observe that the model performs adequately on droplets that are in-focus or nearly in-focus, but inadequately on diffuse droplets, since these were not present in our training data. However, we show that fine-tuning (**Ours - trained on Cityscapes-Rainy, finetuned on Qian-sample(112 img.)**) the model with a very small random sample of the target training set (approximately 10%) yields excellent performance, showing that pre-training a model on our automatically-collected data significantly reduces the need for labelled target domain data.

Model vs. Dataset	Dataset of (16) PSNR	Dataset of (16) SSIM
RAW	24.09	0.8518
Eigen13(4)	28.59	0.6726
Pix2Pix(8)	30.14	0.8299
Qian et al.(no att.)(16)	30.88	0.8670
Qian et al.(full att.)(16)	31.51	**0.9213**
Ours - trained on Qian-Full (862 img.)	**31.55**	0.9020
Ours - trained on Cityscapes-Rainy	27.52	0.8716
Ours - trained on Cityscapes-Rainy, finetuned on Qian-sample(112 img.)	30.21	0.8953

Table 2. Reconstruction quality comparison to state of the art

4.2. Segmentation results

We show that rainy images severely degrade semantic segmentation performance, and that deraining the images **restores** performance. We use an off-the shelf segmentation model (DeepLab v3+(2), trained on the original, vanilla Cityscapes dataset), and derain Cityscapes-Rainy images using the model with top-performing reconstruction results (results in Table 3 and Fig. 4). The segmentation model achieves an mIOU of 0.79 on the original Cityscapes dataset, 0.71 on photographed (dry) Cityscapes, 0.17 on rainy Cityscapes, and a significantly improved value of 0.63 on rainy images that were derained using our model. Addi-

tionally, we show in the first two rows of Table 3 that taking photographs of a high-resolution monitor does not lead to a large loss in segmentation quality, as compared to the original images. We believe that the difference in performance (mIOU of 0.71 vs. 0.79 for the original images) can be reduced further by using a higher-resolution setup, better lens and a better alignment methodology.

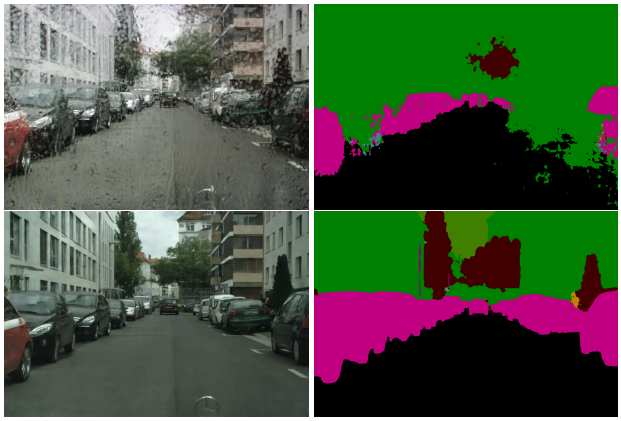


Figure 4. Segmentation results. Top: rainy image. Bottom: derained image.

Cityscapes Img. vs. Segm. Model	mIoU
CLEAR-Original	0.7901
CLEAR-Photographed	0.7137
RAINY	0.1776
DERAINED	**0.6327**

Table 3. Cityscapes Semantic segmentation results

4.3. Qualitative results

We show that our model generalises to other camera setups by recording a real-rainy dataset from inside a moving vehicle, using a camera from a mid-level smartphone. We observe satisfactory reconstruction performance for varying droplet and splatter shapes. We present an example of de-raining of images from such an **unseen** real domain in Figure 5. Outside of quantitative results on third-party rain datasets presented in Table 2 and on our own data presented in Table 1, we are severely limited in offering further numerical analyses by the lack of third-party rainy datasets with either a clear groundtruth or semantic segmentation labels.

5. Conclusion

We have shown that a very simple automated setup - a high resolution computer screen, a glass pane sprayed with water nozzles, and a camera - is suitable for generating rainy data from arbitrary datasets, and more importantly from those with ground truth for auxiliary tasks such as semantic



Figure 5. Qualitative results on **unseen** real-rain domain captured using a smartphone camera, from inside a moving vehicle.

segmentation. Both our quantitative and qualitative results show that data collected in this fashion can be effectively used to train image de-raining and de-noising models that generalise well to other environments, and that this data is valuable for closing domain gaps via fine-tuning, especially when the target data is hard to acquire.

Currently, the system is able to tackle adherent rain, while other effects such as atmospheric misting or fog are out of the scope of this work. Attempts at reducing the impact of these other effects can be found throughout the literature, with a good example being (20). Additionally, while the current implementation involves a windscreen-like setup where windscreen wipers could be envisioned to work and alleviate some of the issues stemming from adherent rain, there are many other camera/sensor locations where wipers would be impractical and which would benefit from our approach. Finally, even if effects such as tail-light reflections on the wet ground cannot be captured or introduced, this does not reduce the usefulness nor the validity of datasets obtained using this method when used for image reconstruction and denoising purposes. While such a dataset does not capture the full effects of rain, it can be used, at a minimum, to massively and effectively reduce the amount (and cost) of labelled/ground-truth real rainy data that needs to be collected.



Figure 6. Left: proof of concept. Right: muddy image.

6. Limitations and future work

The proposed setup is only able to simulate adherent contaminants. For example, wind, heavy snow, hailstorm and other atmospheric contaminants cannot be simulated if they are supposed to change the structure and appearance of objects already captured in the image. Additionally, the ease of generating data with this device comes at the cost of losing some image quality (see Table 3, second row). Another adherent contaminant that creates an obscuring effect

is dirt (dust, mud etc). Soil is synthetically generated in (22), whereas we propose as future work to use a rolling clear film to gather real mud (see Fig. 6).

References

[1] J. F. Blinn. A generalization of algebraic surface drawing. *ACM Trans. Graph.*, 1(3):235–256, July 1982. 2

[2] L. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. *CoRR*, abs/1802.02611, 2018. 4

[3] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 4

[4] D. Eigen, D. Krishnan, and R. Fergus. Restoring an image taken through a window covered with dirt or rain. In *2013 IEEE International Conference on Computer Vision*, pages 633–640, Dec 2013. 2, 3, 4

[5] E. Fouad and E. Abdelhak. *Adherent Raindrop Detection Based on Morphological Operations: Volume 5: Advanced Intelligent Systems for Computing Sciences*, pages 324–331. 01 2019. 2

[6] J. C. Halimeh and M. Roser. Raindrop detection on car windshields using geometric-photometric environment construction and intensity-based correlation. In *2009 IEEE Intelligent Vehicles Symposium*, pages 610–615, June 2009. 2

[7] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 3

[8] P. Isola, J. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. *CoRR*, abs/1611.07004, 2016. 4

[9] K. Ito, K. Noro, and T. Aoki. An adherent raindrop detection method using MSER. In *2015 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA)*, pages 105–109, Dec 2015. 2

[10] J. Johnson, A. Alahi, and L. Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European Conference on Computer Vision*, 2016. 3

[11] M. Kuramoto, A. Yamashita, T. Kaneko, and K. Miura. Removal of adherent waterdrops in images by using multiple cameras. pages 80–83, 01 2002. 2

[12] C. Li and M. Wand. Precomputed real-time texture synthesis with markovian generative adversarial networks. In *European Conference on Computer Vision*, pages 702–716. Springer, 2016. 3

[13] H. Lin, X. Fu, C. Jing, X. Ding, and Y. Huang. Aˆ2net: Adjacent aggregation networks for image raindrop removal. *CoRR*, abs/1811.09780, 2018. 2

[14] C. Michaelis, B. Mitzkus, R. Geirhos, E. Rusak, O. Bringmann, A. S. Ecker, M. Bethge, and W. Brendel. Benchmarking robustness in object detection: Autonomous driving when winter is coming. *CoRR*, abs/1907.07484, 2019. 3

[15] H. Porav, T. Bruls, and P. Newman. I can see clearly now : Image restoration via de-raining. *CoRR*, abs/1901.00893, 2019. 2, 3, 4

[16] R. Qian, R. T. Tan, W. Yang, J. Su, and J. Liu. Attentive generative adversarial network for raindrop removal from a single image. *CoRR*, abs/1711.10098, 2017. 2, 4

[17] M. Roser and A. Geiger. Video-based raindrop detection for improved image registration. *2009 IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops*, pages 570–577, 2009. 2

[18] M. Roser, J. Kurz, and A. Geiger. Realistic modeling of water droplets for monocular adherent raindrop recognition using bézier curves. In *ACCV Workshops*, 2010. 2

[19] C. S Vijay, R. Bhat, and V. Ragavan. Raindrop detection considering extremal regions and salient features. *Electronic Imaging*, 2018:1–6, 01 2018. 2

[20] C. Sakaridis, D. Dai, and L. V. Gool. Semantic foggy scene understanding with synthetic data. *CoRR*, abs/1708.07819, 2017. 5

[21] B. Ulfwi. Adherent raindrop detection, 2018. Student Paper. 1, 2

[22] M. Uricár, P. Krízek, G. Sistu, and S. Yogamani. Soilingnet: Soiling detection on automotive surround-view cameras. *CoRR*, abs/1905.01492, 2019. 2, 6

[23] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, A. Tao, J. Kautz, and B. Catanzaro. High-resolution image synthesis and semantic manipulation with conditional GANs. In *Computer Vision and Pattern Recognition (CVPR), 2018 IEEE Conference on*, pages 1–13. IEEE, 2018. 3

[24] A. Yamashita, T. Kaneko, and K. T. Miura. A virtual wiper-restoration of deteriorated images by using a pan-tilt camera. In *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA '04. 2004*, volume 5, pages 4724–4729 Vol.5, April 2004. 2

[25] A. Yamashita, M. Kuramoto, T. Kaneko, and K. Miura. A virtual wiper - restoration of deteriorated images by using multiple cameras. pages 3126 – 3131 vol.3, 11 2003. 2

[26] A. Yamashita, Y. Tanaka, and T. Kaneko. Removal of adherent waterdrops from images acquired with stereo camera. In *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 400–405, Aug 2005. 2

[27] S. You, R. T. Tan, R. Kawakami, Y. Mukaigawa, and K. Ikeuchi. Adherent raindrop modeling, detection and removal in video. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(9):1721–1733, Sep. 2016. 1, 2

[28] S. You, R. T. Tan, R. Kawakami, Y. Mukaigawa, and K. Ikeuchi. Waterdrop stereo. *CoRR*, abs/1604.00730, 2016. 2

[29] F. Yu, W. Xian, Y. Chen, F. Liu, M. Liao, V. Madhavan, and T. Darrell. BDD100K: A diverse driving video database with scalable annotation tooling. *CoRR*, abs/1805.04687, 2018. 4

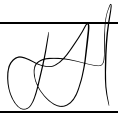
Statement of Authorship for joint/multi-authored papers for PGR thesis

To appear at the end of each thesis chapter submitted as an article/paper

The statement shall describe the candidate's and co-authors' independent research contributions in the thesis publications. For each publication there should exist a complete statement that is to be filled out and signed by the candidate and supervisor **(only required where there isn't already a statement of contribution within the paper itself).**

Title of Paper	Rainy screens: Collecting rainy datasets, indoors
Publication Status	Published
Publication Details	**Horia Porav**, Valentina-Nicoleta Musat, Tom Bruls and Paul Newman, "*Rainy screens: Collecting rainy datasets, indoors*", Arxiv Preprint 2020

Student Confirmation

Student Name:	Horia Porav		
Contribution to the Paper	All work except advice and editorial changes. Major contributions include: - Building the monocular rig, pump and reservoir - Writing the control software for the droplet spraying system - Calibrating the camera, and writing the capture and synchronization software - Aligning the resulting images with their groundtruth - Implementing the de-raining network and all accompanying software implementations - Running all experiments and data collection - Creating the Rainy-BDD and Rainy-Cityscapes - Writing the paper		
Signature		Date	03 October 2020

Supervisor Confirmation

By signing the Statement of Authorship, you are certifying that the candidate made a substantial contribution to the publication, and that the description described above is accurate.

Supervisor name and title: Professor Paul Newman			
Supervisor comments			
Signature		Date	05-10-2020

This completed form should be included in the thesis, at the end of the relevant chapter.

5.2 Reducing steganography in cycle-consistency GAN architectures

Problem: Cycle-consistency GAN architectures suffer from a tendency to hide high-frequency steganographic information in the generated outputs [31]. This information is then used by the complementary generators to reconstruct the inputs, leading to an artificially low reconstruction loss and a lower quality of the generated outputs. The presence of hidden information also reduces the usability of cycle-consistency architectures for training data generation.

Solution: During training, end-to-end differentiable image compression techniques (e.g. [59], [60]) can be used to remove this high-frequency information while maintaining image quality. The removal of high-frequency information prevents the generators from learning a shared encoding that would otherwise artificially improve the reconstruction quality (and reduce the loss and magnitude of gradients) when closing the cycle of each generator pair.

Status: Published at CVPR-V4AS 2019

This CVPR Workshop paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

Reducing Steganography In Cycle-consistency GANs

Horia Porav
Oxford Robotics Institute
University of Oxford
horia@robots.ox.ac.uk

Valentina Musat
Oxford Brookes University
Oxford
valentina.musat-2016@brookes.ac.uk

Paul Newman
Oxford Robotics Institute
Oxford
pnewman@robots.ox.ac.uk

Abstract

In this work we present a simple method of improving the suitability of data generated using cycle-consistency GANs in the context of day-to-night domain adaptation. While CycleGANs produce visually pleasing outputs, they also encode hidden (steganographic) information about the source domain in the generated images, which makes them less suitable as training data generators. We reduce the amount of steganographic information hidden in the generated images by introducing an end-to-end differentiable image denoiser in between the two generators. The role of the denoiser is to strip away the high frequency, low amplitude encoded information, making it harder for the generators to hide information that is invisible to the discriminator. We benchmark the suitability of data generated using our simple method in the context of simple domain adaptation for semantic segmentation, comparing with standard CycleGAN, MUNIT and DRIT and show that our method yields better performance.

1. Introduction

Cycle-consistency GANs have become popular ever since their introduction, and have been applied for artistic purposes and domain adaptation in fields like robotics and medical image analysis, where acquiring paired images from other domains is at best expensive, and at worst impossible. It was recently shown [3], however, that generated images contain hidden (steganographic) information about the source domain, and that the generators can end up "hallucinating" content in order to fool the discriminators. The abject simplicity that has made CycleGANs so popular is also its weak point: in order to minimize the reconstruction loss, the two generators cooperatively develop a common, high-frequency encoding for source domain information which is often not picked up by the discriminators. This inclusion of encoded source data raises concerns regarding the suitability of the generated data for domain adaptation training or fine-tuning, and in critical setups such as medical analysis or autonomous driving. A simple example is that of generating night time images from day time images, for training a semantic segmentation task. The segmentation model will learn to decode the steganographic daytime information, which won't be present in the real night time images, leading to a lower bound on performance, spurious results, and an inefficient use of model parameters.



Figure 1: Our method reduces the amount of information hidden in the generated images of cycle-consistency GANs. Top row, from left to right: input day time image, fake night time image and reconstructed day image generated by standard CycleGAN. Bottom row: input day time image, fake night time image and reconstructed day image generated by our improved CycleGAN. In contrast to the standard CycleGAN model, our method drastically reduces the amount of steganographic information hidden in the generated image, and as such is unable to accurately reconstruct the input image.

In this work, inspired by the study of [6] in combating adversarial noise, we present a simple method of reducing the amount of cooperation between the two generators by introducing an end-to-end differentiable image denoiser in between the two generators. The role of the denoiser is to strip away the high frequency, low amplitude information, making it harder for the generators to hide information that is invisible to the discriminator, without having to employ an over-parameterised discriminator that would

destabilize training. We benchmark the suitability of data generated using our method in the context of day-to-night domain adaptation for semantic segmentation. To focus on the improvement brought by having better data, we simply apply day-to-night style transfer to the training split of the Cityscapes [4] dataset and train a semantic segmentation model (Deeplab V3+[2]) on the generated data. We test the trained segmentation models on the test split (no train split available) of the NightTimeDriving[5] dataset. We compare our method with standard CycleGAN[12], MUNIT[8] and DRIT[9], and show that the semantic segmentation model trained on data generated using our method has better performance.

2. Related Work

2.1. Cycle-consistency GANs and steganography

CycleGANs are a popular tool for unpaired image-to-image translation, as opposed to previous methods that require image pairs from both domains. Using an adversarial loss to train a generative model implies learning a transformation $G_{AB} : A \rightarrow B$ such that the distribution of images from domain A rendered in domain B cannot be distinguished from the distribution of real images from domain B. However, this poses a problem, since there can be many images whose distribution is similar to domain B, but with meaningless content. In order to preserve the structure of the image, a cycle consistency constraint is employed, by introducing a second transformation model $G_{BA} : B \rightarrow A$ [11].

Although advantageous, CycleGANs pose a major drawback, as shown in a study by [3], where aerial photographs are translated to maps. The authors find that the model hides information about the aerial image into the generated map (even in solid-color areas), ensuring that aerial images will be reconstructed back with the finest details. By adding high-frequency noise, they compose a map b^* (that is visually similar to an original map b_0) and show that the generator G_{BA} can be forced to recreate a particular aerial photograph a^* from b^*. As the difference between the b^* and b_0 required by the generator in order to produce a^* decreases during training, they conclude that G_{BA} is colluding with G_{AB}'s adversarial attack, where G_{AB} encodes information about the source image and G_{BA} learns to decode the hidden information and recreate the source image in such detail that would otherwise be difficult to reconstruct, since there are many aerial photos that can correspond to one map.

2.2. Reducing steganography in cycle-consistency GANs

In a high-fidelity image to image translation task, [7] tackle steganography by ensuring an uncooperative setup and adding a residual/style path. In the traditional cooperative setup, the networks are trained on each other's output, which means that if the second network is able to compensate for the first network's error, then the first network doesn't need to improve. The proposed solution is to only train the networks when input is real, instead of both real and generated. Their architecture consists of two networks: a disentangler network D, which splits the image from domain V into C and R, where C is another image domain and R is the residual between the two domains, and an entangler network E that merges C and R back to V. In the first learning cycle, D is updated based on the reconstruction error of v, where $D(v) = (c', r')$ and $E(D(v)) = v'$, whereas in the second cycle, E is updated based on the reconstructions of r and c, where E generates $E(c, r) = v'$ and D disentangles $D(v') = (c', r')$. In both cycles the input data is real: in the first cycle, v is an image from domain V, in the second cycle, c is an image from domain C and r is generated from a random v.

As one characteristic of CycleGAN is the deterministic one-to-one mapping, [1] propose Augmented-CycleGAN in order to ensure many-to-many mappings across domains. By allowing the two generators to be conditional on a latent space Z, one could sample different $z \sim p(z)$ to generate multiple samples in the target domain, while holding the source domain fixed. However, the cycle-consistency reconstruction error implicitly assumes unimodality, since all images generated across samples z will have to be close to the original image, for the loss to be minimized. Their approach is thus to learn mappings between augmented spaces $A \times Z_b$ and $B \times Z_a$, where Z_a and Z_b are latent spaces that encode missing information: Z_a holds information about a that does not exist in the generated image b and vice-versa. Their loss function is amended to include two adversarial and two reconstruction losses for latent codes Z_a and Z_b. They check for steganography by introducing noise in the source domain images and evaluating the reconstruction error in the target domain and conclude that information is captured in the latent codes rather than being hidden in the generated images.

We draw our inspiration from [6], who study the effect of JPEG compression on adversarial perturbation and conclude that small perturbations can be eliminated using compression. We replace the non-differentiable JPEG compression with a learned denoising network.

3. Learning to reduce steganography

3.1. The image denoising network

As most of the hidden information inside cycle-consistency GANs is represented by high-frequency, low-amplitude information [3], we choose to use a differentiable denoiser to reduce this type of information. The intuition is that, within each cycle, the generators will be forced to encode a better structure-preserving output, using lower fre-

quency representations, while still minimizing discriminator losses. The architecture is presented in Figure 2.

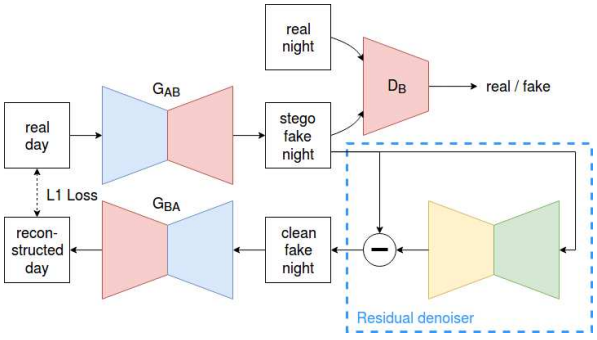


Figure 2: The architecture of one of the two cycles used to train traditional cycle-consistency GANs, augmented with a differentiable image denoising network in between the two generators. The discriminator always receives an unfiltered generated image.

We make use of a residual image denoiser R, which, invariant to the domain, takes and image I_z affected by noise z and extracts the noise component z. The original, clean image I is then reconstructed by subtracting the noise from the noise-affected image:

$$I_z = I + z \tag{1}$$

$$R(I_z) \approx z \tag{2}$$

$$I = I_z - R(I_z) \tag{3}$$

3.2. Cycle-Consistency GANs

Following [12] and using the image denoiser R described above, we use 2 generators: given an image I_A from domain A (day) and an image I_B from domain B (night), we employ generator G_AB to translate I_A to domain B and generator G_BA to translate the image back into the original domain. We add the image denoiser R in between the two generators. On the output of each generator, we apply an adversarial loss: discriminators D_B for generator G_AB, and D_A for G_BA respectively. The adversarial losses are:

$$\mathcal{L}_{\mathrm{B}_\mathrm{adv}} = (D_\mathrm{B}(G_\mathrm{AB}(I_\mathrm{A})) - 1)^2 \tag{4}$$

$$\mathcal{L}_{\mathrm{A}_\mathrm{adv}} = (D_\mathrm{A}(G_\mathrm{BA}(I_\mathrm{B})) - 1)^2 \tag{5}$$

The final adversarial objective $\mathcal{L}_\mathrm{adv}$ to be minimized becomes:

$$\mathcal{L}_\mathrm{adv} = \mathcal{L}_{\mathrm{B}_\mathrm{adv}} + \mathcal{L}_{\mathrm{A}_\mathrm{adv}} \tag{6}$$

The discriminators are trained to minimize:

$$\mathcal{L}_{\mathrm{B}_\mathrm{disc}} = (D_\mathrm{B}(I_\mathrm{B}) - 1)^2 + (D_\mathrm{B}(G_\mathrm{AB}(I_\mathrm{A})))^2 \tag{7}$$

$$\mathcal{L}_{\mathrm{A}_\mathrm{disc}} = (D_\mathrm{A}(I_\mathrm{A}) - 1)^2 + (D_\mathrm{A}(G_\mathrm{BA}(I_\mathrm{B})))^2 \tag{8}$$

The final discriminator objective $\mathcal{L}_\mathrm{disc}$ to be minimized becomes:

$$\mathcal{L}_\mathrm{disc} = \mathcal{L}_{\mathrm{B}_\mathrm{disc}} + \mathcal{L}_{\mathrm{A}_\mathrm{disc}} \tag{9}$$

Using the image denoiser R, a cycle-consistency loss [12] is computed between the input images and the reconstructed images:

$$\mathcal{L}_{\mathrm{A}_\mathrm{rec}} = \|I_\mathrm{A} - \hat{I_A}\|_1 \tag{10}$$

$$\mathcal{L}_{\mathrm{B}_\mathrm{rec}} = \|I_\mathrm{B} - \hat{I_B}\|_1 \tag{11}$$

where

$$\hat{I_A} = G_{BA}(G_{AB}(I_A) - R(G_{AB}(I_A))) \tag{12}$$

$$\hat{I_B} = G_{AB}(G_{BA}(I_B) - R(G_{BA}(I_B))) \tag{13}$$

The generator objective $\mathcal{L}_\mathrm{gen}$ becomes:

$$\mathcal{L}_\mathrm{gen} = \lambda_\mathrm{rec} * \mathcal{L}_\mathrm{rec} + \lambda_\mathrm{adv} * \mathcal{L}_\mathrm{adv} \tag{14}$$

with each λ weighing the importance of individual objectives. The generators G_AB, G_BA that minimize the complete objective are:

$$G_\mathrm{AB}, G_\mathrm{BA} = \underset{G_\mathrm{AB}, G_\mathrm{BA}, D_\mathrm{B}, D_\mathrm{A}}{\arg\min} \mathcal{L}_\mathrm{gen} + \mathcal{L}_\mathrm{disc} \tag{15}$$

4. Experimental Setup

We compare a baseline semantic segmentation model (Deeplab V3+ with Mobilenet backbone [2]) trained on the Cityscapes train split, and the same model fine-tuned with data generated using standard CycleGAN, MUNIT [8], DRIT[9] and our method. We chose these methods for their relative popularity and similar complexity to our method.

4.1. The denoising network

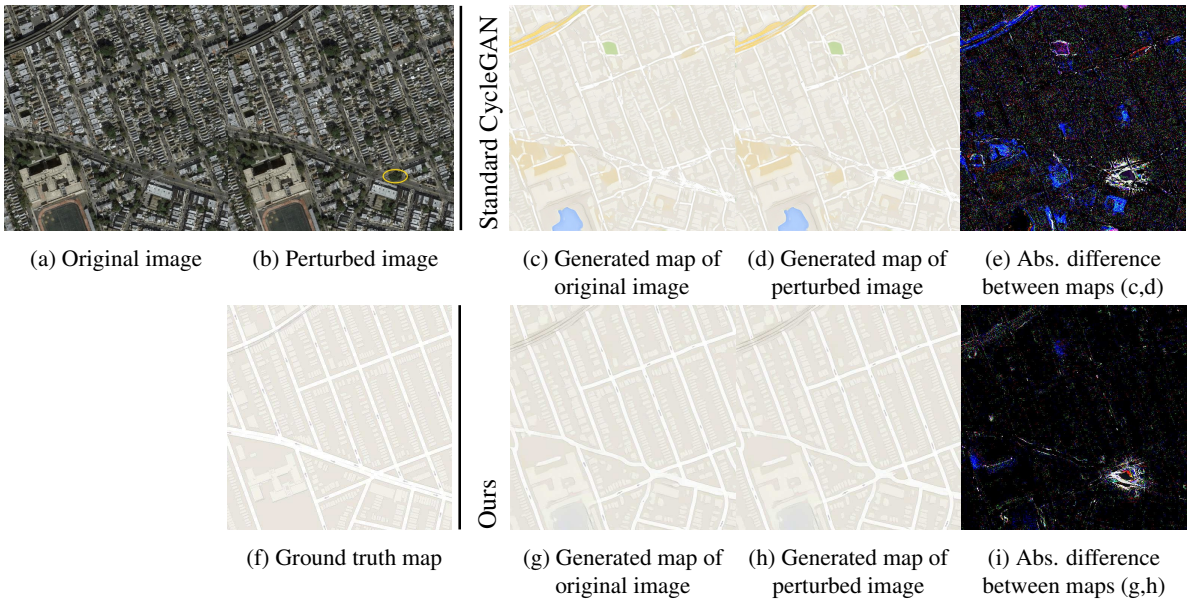
We make use of the reference implementation of DNCNN[1] [10], which is trained for blind Gaussian denoising with a large range of noise levels ($\sigma \in [0, 55]$). The network is **frozen** during our training and inference runs.

4.2. Training, generating data and fine-tuning

For training the data generation pipeline, we divide the training split of Cityscapes dataset in two equal shuffled sets of 1488 images each, and use the first set as the day-domain dataset, while using the testing split of NightTimeDriving as the night-domain dataset. For standard CycleGAN and our method, we follow the regimen of [12] and train for 50 epochs. For MUNIT, we follow the regimen described in [8] and train for 100000 iterations. Similarly, for DRIT, we follow the authors of [9] and train for 105000 iterations.

To test the quality of the generated data, we apply day-to-night style transfer to the second set of 1488 Cityscapes images, using generators trained with each of the frameworks mentioned above, producing images at both 512x256

[1] https://github.com/cszn/DnCNN



(a) Original image (b) Perturbed image (c) Generated map of original image (d) Generated map of perturbed image (e) Abs. difference between maps (c,d)

(f) Ground truth map (g) Generated map of original image (h) Generated map of perturbed image (i) Abs. difference between maps (g,h)

Figure 3: We demonstrate qualitatively that out method significantly reduces the amount and distribution of steganographic information. We add a perturbation (circled in yellow) to an input image and observe the effect it produces on the output of the generator. The first row shows results using Standard CycleGAN, while the second row shows results from our method. In contrast to Standard CycleGAN, where the perturbation is encoded **throughout** the image and with large amplitudes, our method encodes this information mostly locally, around the spatial location of the perturbation and along edges with large contrast. Note that the difference is computed over images that have **NOT** been denoised by the denoiser, and represent the raw output of the generator. Additionally, note that our results are much **closer** to the ground truth map.

and 1024x512 resolutions, and use this data to fine-tune the semantic segmentation model. Note that, for our method, the denoising network is not used at runtime. Finally, we evaluate the baseline segmentation model and each fine-tuned model on the test split of the NightTimeDriving [5] dataset, measuring Mean Intersection Over Union (mIOU). As night-time datasets with groundtruth annotations are scarce, the NightTimeDriving test split is used both as a night-time style for data generation, and as the test set for semantic segmentation. However, pollution of the generated data from the style information is minimal, and all methods are benchmarked with exactly the same setup.

5. Results

5.1. Quantitative results

While all methods performed far better than the baseline, the results presented in Table 1 indicate that our simple denoising strategy produces superior training data compared to MUNIT, DRIT and standard CycleGAN, at both resolutions. Note that our aim is to compare methods in relative terms and not to obtain absolute state-of-the-art results.

5.2. Qualitative results

In Figure 3, we show that our method significantly reduces the amount and distribution of steganographic information. Following the experiment proposed in [3], we first train our denoising-CycleGAN on an aerial-photo↔map dataset. We add a perturbation (circled in yellow) to an input image and observe the effect it produces on the output of the generator by computing the absolute difference between the map obtained from the unperturbed image and the map obtained from the perturbed image. In contrast to Standard CycleGAN, where the perturbation is encoded **throughout** the image and with large amplitudes, images generated using our method encode this information mostly locally, around the spatial location of the perturbation and along edges with large contrast.

Table 1: Semantic segmentation mIOU for all methods

	Trained on				
Training resolution	Source Domain	MUNIT	CycleGAN	DRIT	Ours
512x256	0.1305	0.1820	0.2231	0.2361	**0.2630**
1024x512	0.1305	0.2323	0.2700	0.2709	**0.2971**

6. Conclusions

We have presented a simple way to reduce steganographic information within cycle-consistency GANs and better align generated images with a target distribution. We have demonstrated this quantitatively by showing that using the improved generated images to fine-tune a semantic segmentation model leads to better performance on a real-world difficult dataset, and qualitatively by showing that the extent and amplitude of hidden information is reduced.

References

[1] Amjad Almahairi, Sai Rajeswar, Alessandro Sordoni, Philip Bachman, and Aaron C. Courville. Augmented cyclegan: Learning many-to-many mappings from unpaired data. *CoRR*, abs/1802.10151, 2018. 2

[2] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *ECCV*, 2018. 2, 3

[3] Casey Chu, Andrey Zhmoginov, and Mark Sandler. Cyclegan, a master of steganography. *CoRR*, abs/1712.02950, 2017. 1, 2, 4

[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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3213–3223, 2016. 2

[5] Dengxin Dai and Luc Van Gool. Dark model adaptation: Semantic image segmentation from daytime to nighttime. In *IEEE International Conference on Intelligent Transportation Systems*, 2018. 2, 4

[6] Gintare Karolina Dziugaite, Zoubin Ghahramani, and Daniel M. Roy. A study of the effect of JPG compression on adversarial images. *CoRR*, abs/1608.00853, 2016. 1, 2

[7] Adam W. Harley, Shih-En Wei, Jason M. Saragih, and Katerina Fragkiadaki. Image disentanglement and uncooperative re-entanglement for high-fidelity image-to-image translation. *CoRR*, abs/1901.03628, 2019. 2

[8] Xun Huang, Ming-Yu Liu, Serge Belongie, and Jan Kautz. Multimodal unsupervised image-to-image translation. In *ECCV*, 2018. 2, 3

[9] Hsin-Ying Lee, Hung-Yu Tseng, Jia-Bin Huang, Maneesh Kumar Singh, and Ming-Hsuan Yang. Diverse image-to-image translation via disentangled representations. In *European Conference on Computer Vision*, 2018. 2, 3

[10] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang. Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. *IEEE Transactions on Image Processing*, 26(7):3142–3155, July 2017. 3

[11] 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. 2

[12] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networkss. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, 2017. 2, 3

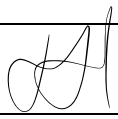
Statement of Authorship for joint/multi-authored papers for PGR thesis

To appear at the end of each thesis chapter submitted as an article/paper

The statement shall describe the candidate's and co-authors' independent research contributions in the thesis publications. For each publication there should exist a complete statement that is to be filled out and signed by the candidate and supervisor **(only required where there isn't already a statement of contribution within the paper itself).**

Title of Paper	Reducing Steganography In Cycle-consistency GANs
Publication Status	Published
Publication Details	**Horia Porav**, Valentina-Nicoleta Musat and Paul Newman, "*Reducing Steganography In Cycle-consistency GANs*", 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition - Vision for all Seasons Workshop (CVPR2019-V4AS)

Student Confirmation

Student Name:	Horia Porav		
Contribution to the Paper	All work except advice and editorial changes. Major contributions include: - Generating the ideas relating to the mitigation of collusion between generators in cycle-consistency architectures - Implementing the cycle-consistency + residual denoiser architecture - Running all experiments and generating training data for the segmentation models - Generating training data using DRIT and MUNIT - Performing the perturbation and qualitative experiments - Training the segmentation models - Performing the quantitative experiments - Writing the paper, along with diagrams and tables - Presenting the paper at CVPR-V4AS 2019		
Signature		Date	03 October 2020

Supervisor Confirmation

By signing the Statement of Authorship, you are certifying that the candidate made a substantial contribution to the publication, and that the description described above is accurate.

Supervisor name and title: Professor Paul Newman			
Supervisor comments			
Signature		Date	05-10-2020

This completed form should be included in the thesis, at the end of the relevant chapter.

5.3 Generating All the Roads to Rome: Road Layout Randomization for Improved Road Marking Segmentation

Problem: Road markings and road boundaries are paramount to driving, both for humans and autonomous vehicles. While human drivers are able to parse most markings even under difficult appearance and illumination conditions, machine learning models require sizeable amounts of labelled training data collected under various conditions. However, producing vast quantities of road marking labels is costly, time-consuming, and simply infeasible for every domain and condition. Additionally, data obtained from driving simulators is often affected by a much-too-large domain gap.

Solution: Starting from an initial automatically labelled or hand-labelled dataset, we apply principles of domain randomization to the road layout and synthesize new images from altered semantic labels using a state-of-the-art Generative Adversarial Network. We demonstrate that training on these synthetic pairs improves segmentation performance on rare road marking classes during real-world deployment in complex urban environments by more than 12 percentage points as compared to the baseline model trained only with human labelled data, while performance for other classes is retained. Our framework can easily be scaled to all domains and conditions to generate large-scale road marking datasets, while avoiding most manual labelling effort.

Status: Presented at ITSC2019

Generating All the Roads to Rome: Road Layout Randomization for Improved Road Marking Segmentation

Tom Bruls, Horia Porav, Lars Kunze, and Paul Newman

arXiv:1907.04569v1 [cs.RO] 10 Jul 2019

***Abstract*— Road markings provide guidance to traffic participants and enforce safe driving behaviour, understanding their semantic meaning is therefore paramount in (automated) driving. However, producing the vast quantities of road marking labels required for training state-of-the-art deep networks is costly, time-consuming, and simply infeasible for every domain and condition. In addition, training data retrieved from virtual worlds often lack the richness and complexity of the real world and consequently cannot be used directly. In this paper, we provide an alternative approach in which new road marking training pairs are automatically generated. To this end, we apply principles of domain randomization to the road layout and synthesize new images from altered semantic labels. We demonstrate that training on these synthetic pairs improves mIoU of the segmentation of rare road marking classes during real-world deployment in complex urban environments by more than 12 percentage points, while performance for other classes is retained. This framework can easily be scaled to all domains and conditions to generate large-scale road marking datasets, while avoiding manual labelling effort.**

I. Introduction

Safety-critical systems, such as automated vehicles, need interpretable and explainable decision-making for real-world deployment. An important aspect for improving interpretability of such systems is the ability to explain scenes semantically. More specifically, planning the behaviour of an automated vehicle through an urban traffic environment requires understanding of the *road rules*. These are conveyed to the traffic participants by the markings painted on the road.

Although semantic reasoning about road markings is ideally performed at an object and scene level [1], state-of-the-art deep learning methods perform semantic segmentation at the pixel level. This, however, requires thousands of pixel-labelled images for different environments and conditions, which is a problem for several reasons. Firstly, it is impossible to label every pixel of every image for every city in every condition manually. Secondly, simple data augmentation techniques [2] (e.g. flipping, translating, adjusting contrast, etc.) do not deliver the necessary diversity to adapt to all encountered environments and conditions [3].

Even if more efficient hand-labelling techniques become available in the future, we still face the issue of *edge cases* that appear very infrequently in regular driving. In the context of road marking segmentation, data collection during regular driving creates extremely imbalanced datasets. For example, zigzag markings (which indicate a pedestrian crossing, Fig. 1) are encountered rarely, but their detection is critical for safe operation. Resampling or applying a class-weighted loss function are not viable solutions for small, hand-labelled datasets, since these simply contain insufficient examples of rare classes for proper generalization. Retrieving more examples is labour intensive in terms of driving and labelling time. Consequently, trained classifiers show decreased performance on infrequently-occurring classes [4].

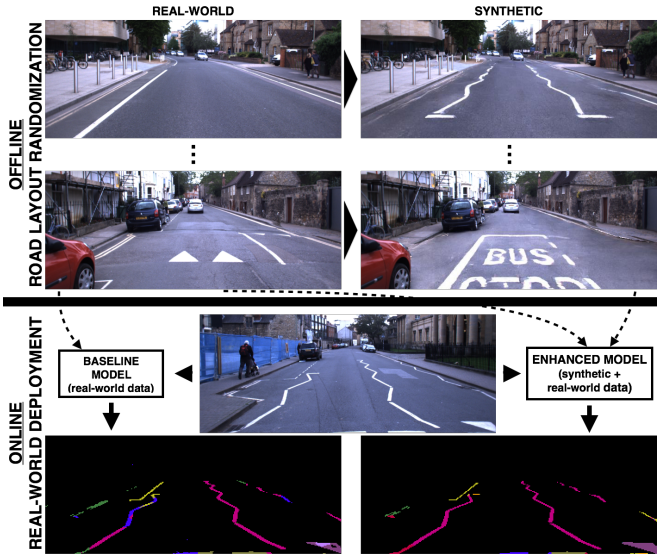


Fig. 1. Road layout randomization for improved road marking segmentation, while avoiding manual labelling. *Offline:* Firstly, new images for training road marking segmentation networks are automatically generated by synthesizing new road surfaces from altered semantic labels. *Online:* Subsequently, mIoU of the segmentation of rare road markings (e.g. zigzags shown in pink) is improved by more than 12 percentage points during *real-world* deployment by the enhanced model trained on a hybrid dataset when compared to a baseline model only trained on the real-world dataset.

The latter problem could be solved by creating a virtual environment (i.e. simulator), in which the desired road markings can be reproduced as many times as necessary. However, this introduces several new challenges. Firstly, even though state-of-the-art simulators can appear realistic to the human-eye, their fidelity lacks the richness and complexity of the real world and consequently there is still an apparent domain gap between simulated environments and their real-world equivalent. As a result, domain adaptation techniques need to be applied for real-world deployment [5], [6]. Secondly, although we might be able to generate simulated environments from real-world data in the future [7], at present their design remains a manual, costly, and time-consuming task. Besides, since urban environments can vary substantially between countries, there is a need for highly-configurable

Authors are from the Oxford Robotics Institute, Dept. Engineering Science, University of Oxford, UK. {tombruls, horia, lars, pnewman}@robots.ox.ac.uk

virtual worlds, which increases the labour cost.

Recently, alternative methods have been developed [8] to synthesize new, photo-realistic scenes for a domain of interest by employing Generative Adversarial Networks (GANs). These approaches require relatively little human effort and can easily extend to all kinds of different conditions [9]. This provides the ability to generate large-scale datasets for semantic scene understanding in a domain of interest at low cost. Most of these frameworks take real-world scenes and augment them by placing or removing objects (e.g. cars, pedestrians, etc.). This can be done randomly [10] or more naturally by learning from real-world examples [11], [12].

Similarly, we place instances of chosen road markings into newly-synthesized, photo-realistic scenes, which are then used to train a road marking segmentation network. In this way, we generate sufficient examples of *rare* road marking classes to achieve the generalization performance required during real-world deployment, as visualized in Fig. 1. However, placing new road markings coherently into the scene is difficult, since there are many dependencies such as the type of road / intersection, traffic lights, parked cars, etc. that need to be taken into account. We avoid solving this hard problem by employing the principles of domain randomization [13]. More concretely, we place road markings at random places on the road surface, not necessarily coherent with other elements in the scene. In this way, we perform *road layout randomization*. Real-world scenes encountered during deployment then appear as samples of the broadened distribution on which the model was trained.

We demonstrate quantitatively that training on these synthetic labels improves mIoU of the segmentation of rare road marking classes, for which it is expensive to attain sufficient real-world examples, during real-world deployment in complex urban environments by more than 12 percentage points. To take full advantage of the synthetic labels we introduce a new class-weighted cross-entropy loss which balances the training. Furthermore, we show qualitatively that the segmentation performance for other classes is retained.

We make the following contributions in this paper:

- We present a method for generating large-scale road marking datasets for a domain of interest by leveraging principles of domain randomization, while avoiding expensive manual effort.
- We introduce a new class-weighted cross-entropy loss to balance the training on synthetic datasets with large class-wise imbalance in terms of their occurrence.
- We demonstrate a real-time framework for improving the segmentation of (rare) road marking classes in *real-world*, complex urban environments.

II. Related Work

Road Marking Segmentation: Deep networks are increasingly used to perform lane detection in highway scenarios [14]–[16]. However, the urban environments and road markings targeted in this paper are substantially different and more complex, and thus require a different approach. This problem has seen significantly fewer deep learning solutions, due to a lack of large-scale datasets containing road markings. The first large-scale semantic road marking dataset was recently introduced in [17], however it is extremely expensive to manually expand this to all environments and conditions.

Road marking segmentation as demonstrated in [4] is closest to the application of this paper. The authors train a network for semantic road marking segmentation and improve their results by predicting the vanishing point simultaneously. In contrast to this paper, they require thousands of hand-labelled images, which is very labour expensive. Alternatively, the authors of [18] hand-label road markings such as arrows and bicycle signs and train an object detection network to predict bounding boxes instead of pixel segmentations. In previous work [19] (includes more extensive review), we have introduced a weakly-supervised approach for binary road marking segmentation, which is used here to acquire road marking labels for real-world scenes.

Synthetic Training for Automated Driving Tasks: To prevent costly and time-consuming manual labelling of training data, many approaches leverage synthetic datasets. Early works trained on purely virtual data to perform object detection [20], [21] or semantic segmentation [5], [6].

However, virtual data lacks the richness and complexity of the real world. A possible alternative is to augment real-world data. For the task of semantic segmentation this means either generating new, photo-realistic images from semantic labels [8], [22], [23] or enriching semantic labels with virtually-generated information [24]. Both of these principles are applied in this paper. For object detection tasks, the main difficulty is to place the (dynamic) objects coherently into the scene. The simplest solution is random object placement (i.e domain randomization) [10]. Alternatively, the authors of [25], [26] place photo-realistic, synthetic cars into real-world images by taking into account the geometry of the scene. The most recent approaches [11], [12], [27], [28] learn context-aware object placement from real-world examples. However, placing dynamic objects, such as pedestrians, seems less complex than road markings, because the space of realistic solutions is less restrictive. Therefore, we place road markings randomly onto the road surface in this paper.

Scene Manipulation: Recently, several approaches have been introduced for more complex scene manipulation, beyond simple augmentation. Additional sensor modalities are used in [29] to offer the flexibility (e.g. different view points) of a virtual simulator, while generating data with the fidelity and richness of real-world images. The authors of [30] introduce a probabilistic programming language to synthesize complex scenarios from existing domain knowledge. Another system [31] offers similar levels of control, while the camera sensor is modelled accurately at the same time. These frameworks potentially offer a way to generate improved training data for our approach.

III. Generating Synthetic Training Pairs

In this section, we explain in detail how to generate synthetic training pairs for road marking segmentation networks to improve performance during real-time deployment, as

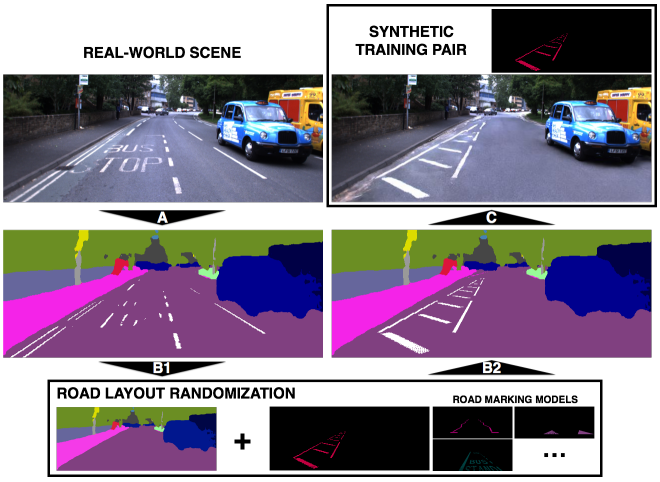


Fig. 2. Road layout randomization: generating synthetic training data based on real-world scenes. The process has the following steps (as described in the respective subsections of Section III): (A) semantic segmentation of the real-word scene is acquired, (B1) the road markings are removed and replaced with road surface, (B2) instances of chosen road markings (modelled according to the UK Highway Code) are placed randomly on the road surface, and finally (C) the road surface of the original image is replaced with a GAN-synthesized, photo-realistic alternative based on the altered semantic label. The composite image is then paired with the generated road marking label.

shown in Fig. 2. We demonstrate that this framework can be employed on any driving dataset even when no ground-truth semantic or road marking labels are available.

A. Retrieving Semantic Labels for Real-World Scenes

In order to generate synthetic training pairs for road marking segmentation, the road layout of semantic labels of real-world scenes is altered and from these new, photo-realistic images are synthesized. Ground-truth semantic labels are not required for the domain of interest, since semantic segmentation of reasonable (i.e. sufficient) quality can be acquired from a model pretrained on the Cityscapes dataset[1]. In this way, we retrieve semantic labels of real-world scenes from the Oxford RobotCar dataset [32], as shown in Fig. 3.

Unfortunately, the available model is not trained to segment road markings (Cityscapes does not contain road marking masks). However, semantic labels including road markings and their corresponding real-world images are necessary to train the GAN described in Section III-C. We prevent manual labelling of road markings by employing the techniques of [19] to generate large quantities of road marking annotations automatically. Because these annotations are generated automatically, they are not equivalent to the ground-truth, however they have proven to be sufficient for training purposes if regularization techniques are applied. The road markings are added to the semantic labels acquired from the Cityscapes model, as visualized in Fig. 3.

B. Road Layout Randomization

To form new road marking training pairs, we alter the road layout (i.e. road markings) of the retrieved semantic labels

[1]https://github.com/tensorflow/models/blob/master/research/deeplab/g3doc/model_zoo.md

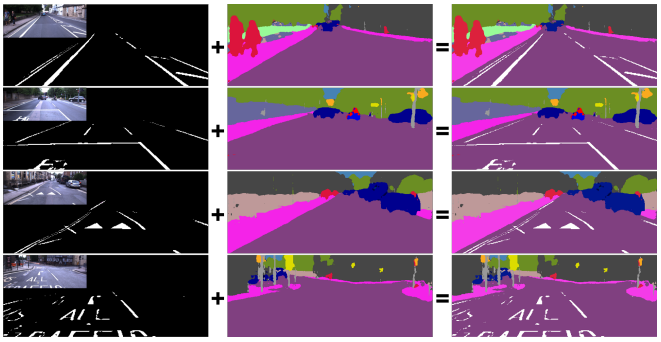


Fig. 3. We augmented the Oxford Robotcar Dataset with semantic labels including road markings to train the CGAN described in Section III-C. The semantic segmentation label is retrieved from inference with a pretrained Cityscapes model and combined with automatically generated road marking annotations from [19]. The resulting labels are not perfect ground-truth, but they are sufficient for the task and can be acquired at low cost.

and subsequently synthesize a new corresponding image. In order to rebalance datasets collected during regular driving, we create new semantic labels with road markings which occur relatively infrequently in the real world (e.g. pedestrian crossing, arrows, etc.). By training the road marking segmentation network on the rebalanced dataset, the goal is to improve the performance for these respective *rare* classes, while at the same time retaining the overall performance.

As mentioned before, the type and placement of road markings is dependent on many factors of the scene such as the type of road, traffic lights, and even the traffic participants. Altering all of these coherently according to the real world is difficult and seems similar in terms of complexity to designing a simulator. Therefore, we choose to leverage domain randomization principles [10]. We vary position (and scale accordingly), rotation, quantity, and partial occlusion of the road markings that are placed into the environment and in that way perform *road layout randomization* to create vast quantities of new training pairs automatically. For accurate placement, we use the camera sensor calibration of the vehicle and assume that the road surface is planar and horizontal. Training the network on many randomly-generated pairs improves generalization in newly-encountered, real-world scenes, which then appear as variations of the distribution on which the network was trained.

Concretely, we start by erasing the original road markings from the real-world semantic labels and subsequently place a new road marking instance onto the cleared road surface. The classes are realistically modelled according to the UK Highway Code so that their shape, size, colour and configuration (e.g. zigzags appear in dual or triple configurations) resemble the real world. Some examples for different classes of rare road markings are given in Fig. 4.

C. Synthesizing Photo-Realistic Images

In order to create a synthetic training pair, we train a Conditional Generative Adversarial Network (CGAN), as introduced in [8], to synthesize a photo-realistic RGB image for the altered semantic label (from Section III-B). In this



Fig. 4. Examples of newly-synthesized training images for several rare road marking classes (i.e. zigzag, diagonal stripes, bus stop, and small warning triangles) by employing road layout randomization. The top two rows show images of real-world scenes of the Oxford RobotCar dataset together with the corresponding (partial) semantic labels (Section III-A). The third row visualizes the altered semantic labels in which instances of chosen road markings are placed randomly on the road surface (Section III-B). The last row presents the newly-synthesized road surfaces substituted into the real-world images. The GAN is able to generate road surfaces with photo-realistic textures, lighting, and even degradation as exemplified on the letters of the bus stops.

framework the generator G aims to synthesize the RGB images, while the discriminator D tries to distinguish synthesized from real-world images. The CGAN is trained in a supervised setting using real-world images and corresponding semantic labels retrieved in Section III-A. After the training is completed a photo-realistic image can be synthesized by the generator from the altered semantic labels generated in Section III-B, as shown in Fig. 4.

More specifically, the framework incorporates several advancements over previous works which make it possible to generate higher-resolution images. Firstly, the generator architecture follows a traditional downsample-bottleneck-upsample model, but splits into a global generator and a local enhancer, where the local component is forced to learn high-resolution details for the stabilized features of the global component. Secondly, to overcome discriminator capacity limitations which arise from training with high-resolution images, the framework incorporates three similar discriminators that work on different scales. The discriminators with bigger receptive field enforce more globally consistent image generation, while the smaller receptive fields steer the generator towards more realistic, fine-level details. Lastly, the traditional GAN loss is augmented to include a feature matching loss based on the discriminator. Formally, following the architecture described in [8], given $K = 3$ discriminators D_k, each operating on a different scale, along with the input and label images $I^{\text{SEG}}_{\text{input}}$ and $I^{\text{RGB}}_{\text{label}}$, respectively, the final objective to be minimized is:

$$\mathcal{L}_{\text{tot}} = \min_{G}((\max_{D_1,D_2,D_3} \sum_{k=1,2,3} \mathcal{L}_{\text{GAN}}(G, D_k)) + \lambda_{\text{FM}} \sum_{k=1,2,3} \mathcal{L}_{\text{FM}}(G, D_k) + \lambda_{\text{VGG}} \mathcal{L}_{\text{VGG}}(G)). \tag{1}$$

Here, $\mathcal{L}_{\text{GAN}}(G, D_k)$ represents the usual GAN loss (see [8]) defined over K scales, $\mathcal{L}_{\text{FM}}(G, D_k)$ is the discriminator feature loss defined over K scales:

$$\mathcal{L}_{\text{FM}(G,D_k)} = \sum_{i=1}^{l_D} \frac{1}{w_i} \| D_k(I^{\text{RGB}}_{\text{label}})_i - D_k(G(I^{\text{SEG}}_{\text{input}}))_i \|_1, \tag{2}$$

with l_D defining the number of layers from the discriminator used in the discriminator feature loss and $\mathcal{L}_{\text{VGG}}(G)$ being the perceptual loss:

$$\mathcal{L}_{\text{VGG}(G)} = \sum_{i=1}^{l_P} \frac{1}{w_i} \| \text{VGG}(I^{\text{RGB}}_{\text{label}})_i - \text{VGG}(G(I^{\text{SEG}}_{\text{input}}))_i \|_1, \tag{3}$$

with l_P defining the number of layers from an ImageNet-trained network (in this case VGG16) used in computing the perceptual loss. The factors $w_i = 2^{l-i}$ are utilized to scale the weight of each network layer used in computing the losses. We train the model on 3351 overcast training pairs while using the settings as specified in [8] to generate images with a resolution of 256×640.

Unfortunately, the RobotCar dataset does not contain any boundary or instance labels (as used in [8]) necessary to generate sharp, high-quality images. Consequently, the generated images can be smudgy around object boundaries (e.g. rows of parked cars are merged because of the image perspective, as exemplified in [8]) and contain unnatural artifacts. Therefore, we choose to substitute only the newly-generated road surface and keep the rest of the original image intact. The RobotCar dataset contains sufficient real-world images so that no background duplicates have to exist in the new road marking dataset. In this way, we are able to generate a large-scale urban datasets for road marking segmentation, while avoiding expensive manual labelling.

The above-described framework can easily be extended to different (weather and lighting) conditions by training condition-specific models. If it is not possible to retrieve semantic labels of sufficient quality under difficult conditions, a state-of-the-art invertible generator, that can transform the images into the desired appearance similar to [9], [33], can be employed. In this way the semantic label acquired from the

overcast image can be paired with an image which resembles a different weather or lighting condition.

IV. Training for Road Marking segmentation

In this section, the network trained for road marking segmentation is described in detail, along with some important considerations that have to be taken into account when rebalancing datasets.

A. Network Architecture

Deep networks for road marking segmentation have several advantages over traditional heuristic or shallow-learning pipelines. Firstly, they are more robust to spatial deformations, degradation, and partial occlusion. Secondly, the scene context can be leveraged to improve semantic segmentation and thereby understand the road rules. For instance, similarly-shaped road markings (e.g. lane separators and separators that mark a parking spot) can be classified differently based on their place in the scene and relationship with other objects, whereas this is difficult to accomplish with traditional rule-based systems.

We train a U-Net model [34], but include batch normalization and dropout as regularization techniques. These are paramount in our framework, since we train on partial labels that are generated automatically. Dropout allows the network to extend its prediction towards road marking pixels that were wrongly assigned to the background in the partial labels, because they share more similarities with the road marking class than the background class. The architecture and training settings used are similar to our previous work [19], with the major exception that the output now predicts multiple classes of road markings instead of a binary segmentation. More specifically, the output of the network is computed by applying a channel-wise softmax activation over the final feature maps and assigning a class to each respective pixel by taking the channel-wise $\arg\max$ over the output channels, yielding a one-channel discrete class activation map.

At run time, the Tensorflow implementation of the network performs inference on an input image in real-time ($\sim$62.5 Hz) on an NVIDIA TITAN Xp GPU.

B. Balancing of the Classes

As mentioned before, datasets collected during regular driving are extremely imbalanced in terms of the occurrences of particular road marking classes. For instance, zigzag markings are only found in $\sim 7\%$ of the images, whereas lane separators occur in $\sim 70\%$. Solutions such as resampling the dataset or applying a class-weighted loss function are not viable for small, hand-labelled datasets, because they simply contain an insufficient number of examples of the rare classes to generalize well to unseen cases during deployment.

In this paper, we opt for a different approach in which we synthesize new training pairs for rare classes automatically and add them to an existing dataset. This ensures that there are enough examples of these classes for the network to learn from. However, it is not obvious how to produce a rebalanced dataset including synthetic training pairs that is optimal for training. To counteract the fact that we might add too many synthesized training pairs, we experiment with three types of class-weighted cross-entropy losses:

1) Equal weighting (EQ) of all classes irrespective of their occurrence in the dataset.
2) Median frequency balancing (FB) [35], in which each pixel is weighted by
$$w_c = \frac{\text{median}\,(F)}{f_c}, \tag{4}$$
where $F = \{f_1, \ldots, f_C\}$ with f_c denoting the total number of pixels of class c divided by the total number of pixels in labels where c is present and C the total number of classes.
3) Median total balancing (TB), in which each pixel is weighted by
$$w_c = \frac{\text{median}\,(G)}{f_c + n_c}, \tag{5}$$
where $G = \{f_1 + n_1, \ldots, f_C + n_C\}$ with f_c equivalent to 2) and n_c denoting the number of labels in which class c is present divided by the total number of training pairs.

It is important to note that median frequency balancing only corrects for the fact that some classes naturally occupy less pixels in the images. For instance, dotted lines indicating a pedestrian crossing are smaller in accumulated area than an alternative zebra crossing. However, median frequency balancing does not account for imbalance in occurrences across the dataset; whether $\sim 7\%$ of the images contain zigzag markings or $\sim 70\%$, the weight remains the same as long as their pixel size remains equivalent. This is not ideal, since we artificially create an imbalance in the number of occurrences by adding labels of specific classes. The third weighting function, introduced in this paper, is designed to take this into account, balancing the average pixel area as well as the imbalance in occurrences across the dataset.

V. Experimental Results

In this section we describe the experimental setup and the datasets that we have created, before we present the quantitative and qualitative results.

A. Experimental Setup

We have selected four types of rare road markings for evaluation: bus stops, diagonal stripes (must not enter), small warning triangles, and zigzag markings. These classes function as a proof of concept, but the framework can be applied to any class (i.e. model) of road markings. For quantitative pixel-wise evaluation, we have hand-labelled 102, 102, 96, and 102 *real-world* images containing bus stops, diagonal structures, small warning triangles, and zigzag markings, respectively. Note that in these images only these respective classes were labelled and all other classes present were ignored (see Fig. 7). While we train all models to predict the *full* set of 20 different road markings and show these results qualitatively, we only evaluate the four selected classes quantitatively. We define the pixel-wise metrics $\text{PRE} = \frac{\text{TP}}{\text{TP+FP}}$, $\text{REC} = \frac{\text{TP}}{\text{TP+FN}}$, $\text{F}_1 = 2 * \frac{\text{PRE*REC}}{\text{PRE+REC}}$, and

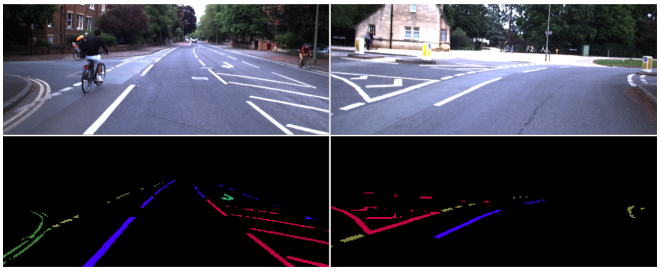


Fig. 5. Examples of the partial labels created by semantically classifying the binary annotations of [19]. Although not perfect ground-truth, these labels can be used to train a baseline model to predict the full set of road markings.

$\text{IoU} = \frac{\text{TP}}{\text{TP+FP+FN}}$ with TP, FP, and FN denoting the true positive, false positive, and false negative pixels, respectively. In contrast to binary classification, all metrics are evaluated at the operating point defined by taking the channel-wise $\arg\max$ over the multi-class output on a per image basis and averaged over the test set, without any further fine tuning of the operating characteristics. Furthermore, we have hand-labelled 25 real-world images for each respective class for validation. We train until convergence and select the epoch for testing in which the mIoU is highest among the evaluations on the validation set. It should be noted that road marking segmentation is arguably a harder task than scene segmentation, because road marking elements are fairly small in general, often degraded, and the different types share many visual and geometric similarities. State-of-the-art approaches achieve a mIoU of around 40%, however a benchmark has only been established recently [17].

As a reasonable baseline, 1000 partial, binary labels generated by [19] collected during regular driving were hand-labelled class-wise. Although not equivalent to the ground-truth, we have proven in [19] and will demonstrate again in Section V-C that these labels are sufficient to achieve full segmentation, when regularization techniques are applied. A few examples are given in Fig. 5. The labels contain the 20 different types of road markings, so that the network functions as a full road marking segmentation system. However, many classes occur too infrequently to achieve state-of-the-art performance, because the network fails to generalize to new scenarios during deployment. For instance, the baseline dataset only contains 63, 109, 39, and 74 images with bus stops, diagonal stripes, small warning triangles, and zigzag markings, respectively. For the other experiments, we add synthetic training pairs of the four classes to the baseline dataset. In this way, the network still predicts all 20 classes, but is given a sufficient number of labels of the rare classes to improve generalization during real-world deployment.

B. Quantitative Evaluation

In order to understand how the number of added synthetic images influences the performance, we have added different numbers of synthetic zigzag pairs to the baseline dataset, while keeping the other classes constant. The results for the three different cross-entropy losses are presented in Fig. 6.

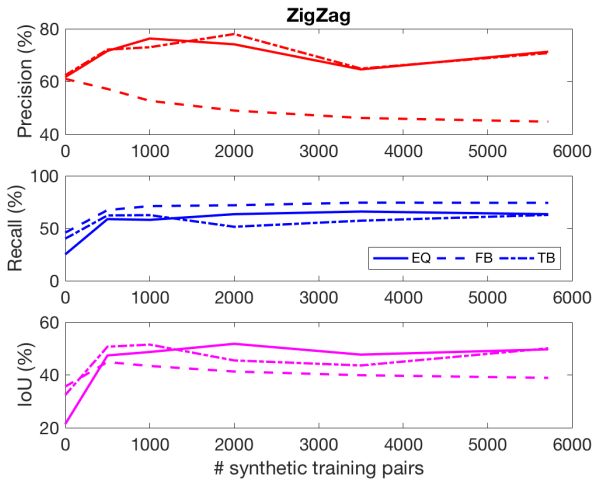


Fig. 6. Pixel-wise performance of zigzag segmentation when training with different cross-entropy losses for a variable number of synthetic images added to the baseline dataset.

The following key observations can be made:

- Adding as little as 500 synthetic training pairs already makes a substantial difference in terms of overall performance.
- Adding more than 2000 synthetic training pairs does not provide extra benefits in general. Further performance increase beyond this level might require higher-quality, more diverse, more coherent synthetic images.
- FB struggles to balance training as more synthetic pairs are added, due to the fact that it does not account for occurrence imbalance across the dataset. The precision drops significantly as the network learns from an abundance of zigzag markings and starts classifying other classes incorrectly as zigzag.
- TB alleviates the precision drop of FB, but does not outperform EQ consistently among all metrics.

Assuming that these observations hold similarly for the other classes, 1000 synthetic training pairs of each respective class were added to the baseline dataset as a proof of concept to train enhanced networks (with the different loss functions). From the results, as presented in Table I, the following key observations can be made:

- By adding synthetic training pairs, IoU performance similar to the state-of-the-art can be achieved when only very few real-world examples are available. mIoU is increased by 12.4% (comparing the best baseline and enhanced models) without using any manual labelling effort.
- The enhanced networks always achieve better overall performance (i.e IoU) by a substantial margin for the equivalent cost function. Segmentation performance can thus be boosted cheaply by the presented framework.
- TB outperforms FB in terms of F_1 and IoU in general, because it accounts for the class imbalance across the dataset that was artificially created by adding synthetic pairs. TB offers a good trade-off between high precision achieved by EQ and high recall achieved by FB.

TABLE I
PIXEL-WISE PERFORMANCE FOR RARE CLASSES FOR THE BASELINE (B) AND ENHANCED (E) MODELS

		BUS STOP				DIAGONAL				TRIANGLE				ZIGZAG				MEAN			
Model	**Loss**	**PRE**	**REC**	**F_1**	**IoU**	**PRE**	**REC**	**F_1**	**IoU**	**PRE**	**REC**	**F_1**	**IoU**	**PRE**	**REC**	**F_1**	**IoU**	**PRE**	**REC**	**F_1**	**mIoU**
B	EQ	61.6	17.8	27.6	16.1	59.1	24.6	34.7	21.8	60.7	41.1	49.0	34.4	65.9	22.5	33.6	20.1	61.8	26.5	36.2	23.1
B	FB	64.7	26.3	37.3	22.8	58.8	31.0	40.6	26.4	60.1	47.7	53.2	36.9	64.7	34.8	45.3	29.5	62.1	35.0	44.1	28.9
B	TB	62.2	19.1	29.2	17.1	58.4	33.3	42.4	27.7	59.9	51.6	53.4	39.4	62.3	31.3	41.7	26.5	60.7	33.8	42.2	27.7
E	EQ	**74.8**	28.5	41.2	26.3	**73.0**	40.9	52.4	35.8	**61.4**	46.9	53.2	38.4	**69.4**	49.8	58.0	40.7	**69.7**	41.5	51.2	35.3
E	FB	54.8	**62.9**	**58.6**	**40.1**	45.8	**67.4**	54.6	39.1	46.9	73.8	57.3	37.9	51.0	**66.6**	57.8	40.3	49.6	**67.7**	57.1	39.4
E	TB	58.5	55.3	56.8	39.2	51.4	59.1	**55.0**	**40.2**	50.8	**75.1**	**60.6**	**43.4**	61.4	57.3	**59.3**	**42.5**	55.5	61.7	**57.9**	**41.3**

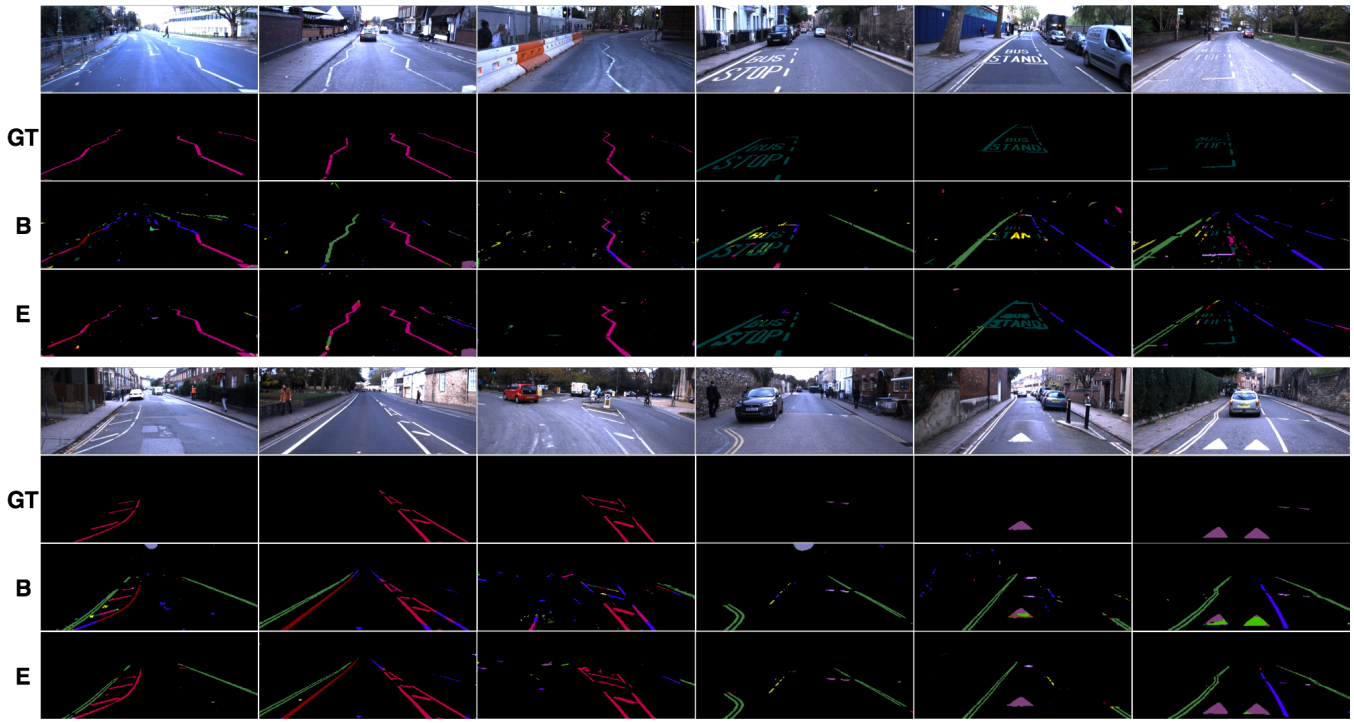


Fig. 7. Road marking segmentation (full set of classes) in traffic environments with rare classes. The *top two* rows of each scene show the input image together with the corresponding ground-truth (GT) label of the rare class, which is used for quantitative evaluation. The *bottom two* rows of each scene depict the segmentation results for the best performing baseline (B) and enhanced model (E), respectively. The enhanced model provides more consistent and correct segmentation of the rare classes, while retaining reasonable and sometimes achieving improved performance for other classes (e.g. *green* double boundaries, *blue* separators, *yellow* parking spot separators, etc.).

C. Qualitative Evaluation

In Fig. 7, the best baseline and enhanced models are compared qualitatively for different traffic scenes. All networks are trained to predict the full set of 20 different road marking classes, however scenes with the respective rare classes are selected for visualization.

It is clear that adding synthetic images to the training set results in more consistent and correct segmentation of the rare classes, while retaining reasonable and sometimes achieving improved performance for other classes. The latter could be caused by the general increase of the number of training examples and/or better balancing of the cost function. The enhanced model trained with TB offers more satisfying (i.e. less noisy) visual results than the baseline model trained with FB. Furthermore, it is clear that full segmentation of the road marking elements is possible from partial labels when regularization techniques are applied correctly. Thus, this framework offers an effective and efficient step towards a road marking classification system for automated driving pipelines.

VI. CONCLUSION

We have presented a weakly-supervised approach for improving road marking segmentation in complex urban environments. To this end, we alter semantic labels of real-world scenes with instances of chosen road markings using domain randomization principles and synthesized corresponding, photo-realistic images to generate vast quantities of synthetic training pairs, thereby avoiding the need for expensive manual labelling. During deployment, we predict 20 classes of road markings in real time and we have demonstrated quantitatively that this framework improves mIoU of rare classes by more than 12 percentage points and thus reaches state-of-the-art performance with very few real-world labels. This is achieved by introducing a new class-weighted cross-entropy loss to balance the training of

synthetic datasets. Furthermore, we have shown qualitatively that the segmentation performance for other classes is retained. The presented framework can easily be extended to include other classes or work under different conditions and results can be expected to improve as more advanced synthesizing networks will emerge in the future. Hence, road layout randomization is an effective and efficient technique to enhance road marking classification systems in automated driving pipelines.

ACKNOWLEDGMENT

The work has been supported by the EPSRC/UK Research and Innovation Programme Grant EP/M019918/1 (Mobile Autonomy: Enabling a Pervasive Technology of the Future). We acknowledge the support of NVIDIA Corporation with the donation of Titan Xp and Titan V GPUs.

REFERENCES

[1] L. Kunze, T. Bruls, T. Suleymanov, and P. Newman, "Reading between the lanes: Road layout reconstruction from partially segmented scenes," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, Nov 2018, pp. 401–408.

[2] E. D. Cubuk, B. Zoph, D. Mane, V. Vasudevan, and Q. V. Le, "Autoaugment: Learning augmentation policies from data," *arXiv preprint arXiv:1805.09501*, 2018.

[3] R. Krajewski, T. Moers, and L. Eckstein, "VeGAN: Using GANs for augmentation in latent space to improve the semantic segmentation of vehicles in images from an aerial perspective," in *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, Jan 2019, pp. 1440–1448.

[4] S. Lee, J. Kim, J. S. Yoon, S. Shin, O. Bailo, N. Kim, T. Lee, H. S. Hong, S. Han, and I. S. Kweon, "VPGNet: Vanishing point guided network for lane and road marking detection and recognition," in *2017 IEEE International Conference on Computer Vision (ICCV)*, Oct 2017, pp. 1965–1973.

[5] Y. Chen, W. Li, X. Chen, and L. Van Gool, "Learning semantic segmentation from synthetic data: A geometrically guided input-output adaptation approach," *arXiv preprint arXiv:1812.05040*, 2018.

[6] A. Dundar, M.-Y. Liu, T.-C. Wang, J. Zedlewski, and J. Kautz, "Domain stylization: A strong, simple baseline for synthetic to real image domain adaptation," *arXiv preprint arXiv:1807.09384*, 2018.

[7] R. Cura, J. Perret, and N. Paparoditis, "Streetgen: In base city scale procedural generation of streets: road network, road surface and street objects," *arXiv preprint arXiv:1801.05741*, 2018.

[8] T. Wang, M. Liu, J. Zhu, A. Tao, J. Kautz, and B. Catanzaro, "High-resolution image synthesis and semantic manipulation with conditional GANs," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, June 2018, pp. 8798–8807.

[9] T. Park, M.-Y. Liu, T.-C. Wang, and J.-Y. Zhu, "Semantic image synthesis with spatially-adaptive normalization," *arXiv preprint arXiv:1903.07291*, 2019.

[10] J. Tremblay, A. Prakash, D. Acuna, M. Brophy, V. Jampani, C. Anil, T. To, E. Cameracci, S. Boochoon, and S. Birchfield, "Training deep networks with synthetic data: Bridging the reality gap by domain randomization," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, June 2018, pp. 1082–10 828.

[11] D. Lee, S. Liu, J. Gu, M.-Y. Liu, M.-H. Yang, and J. Kautz, "Context-aware synthesis and placement of object instances," in *Advances in Neural Information Processing Systems*, 2018, pp. 10 414–10 424.

[12] A. Prakash, S. Boochoon, M. Brophy, D. Acuna, E. Cameracci, G. State, O. Shapira, and S. Birchfield, "Structured domain randomization: Bridging the reality gap by context-aware synthetic data," *arXiv preprint arXiv:1810.10093*, 2018.

[13] J. Tobin, R. Fong, A. Ray, J. Schneider, W. Zaremba, and P. Abbeel, "Domain randomization for transferring deep neural networks from simulation to the real world," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Sep. 2017, pp. 23–30.

[14] B. De Brabandere, W. Van Gansbeke, D. Neven, M. Proesmans, and L. Van Gool, "End-to-end lane detection through differentiable least-squares fitting," *arXiv preprint arXiv:1902.00293*, 2019.

[15] N. Garnett, R. Cohen, T. Pe'er, R. Lahav, and D. Levi, "3D-LaneNet: end-to-end 3D multiple lane detection," *arXiv preprint arXiv:1811.10203*, 2018.

[16] M. Ghafoorian, C. Nugteren, N. Baka, O. Booij, and M. Hofmann, "EL-GAN: Embedding loss driven generative adversarial networks for lane detection," in *Computer Vision – ECCV 2018 Workshops*, L. Leal-Taixé and S. Roth, Eds. Cham: Springer International Publishing, 2019, pp. 256–272.

[17] X. Huang, X. Cheng, Q. Geng, B. Cao, D. Zhou, P. Wang, Y. Lin, and R. Yang, "The ApolloScape dataset for autonomous driving," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, June 2018, pp. 1067–10 676.

[18] T. M. Hoang, P. H. Nguyen, N. Q. Truong, Y. W. Lee, and K. R. Park, "Deep retinanet-based detection and classification of road markings by visible light camera sensors," *Sensors*, vol. 19, no. 2, 2019.

[19] T. Bruls, W. Maddern, A. A. Morye, and P. Newman, "Mark yourself: Road marking segmentation via weakly-supervised annotations from multimodal data," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, May 2018, pp. 1863–1870.

[20] A. Gaidon, Q. Wang, Y. Cabon, and E. Vig, "Virtualworlds as proxy for multi-object tracking analysis," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016, pp. 4340–4349.

[21] M. Johnson-Roberson, C. Barto, R. Mehta, S. N. Sridhar, K. Rosaen, and R. Vasudevan, "Driving in the matrix: Can virtual worlds replace human-generated annotations for real world tasks?" in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, May 2017, pp. 746–753.

[22] S. Liu, J. Zhang, Y. Chen, Y. Liu, Z. Qin, and T. Wan, "Pixel level data augmentation for semantic image segmentation using generative adversarial networks," *arXiv preprint arXiv:1811.00174*, 2018.

[23] K. Li, T. Zhang, and J. Malik, "Diverse image synthesis from semantic layouts via conditional IMLE," *arXiv preprint arXiv:1811.12373*, 2018.

[24] Q. Geng, F. Lu, X. Huang, S. Wang, X. Cheng, Z. Zhou, and R. Yang, "Part-level car parsing and reconstruction from single street view," *arXiv preprint arXiv:1811.10837*, 2018.

[25] H. A. Alhaija, S. K. Mustikovela, L. Mescheder, A. Geiger, and C. Rother, "Augmented reality meets computer vision: Efficient data generation for urban driving scenes," *International Journal of Computer Vision*, vol. 126, no. 9, pp. 961–972, 2018.

[26] H. A. Alhaija, S. K. Mustikovela, A. Geiger, and C. Rother, "Geometric image synthesis," *arXiv preprint arXiv:1809.04696*, 2018.

[27] R. Khirodkar, D. Yoo, and K. M. Kitani, "VADRA: Visual adversarial domain randomization and augmentation," *arXiv preprint arXiv:1812.00491*, 2018.

[28] J. Fang, F. Yan, T. Zhao, F. Zhang, D. Zhou, R. Yang, Y. Ma, and L. Wang, "Simulating LiDAR point cloud for autonomous driving using real-world scenes and traffic flows," *arXiv preprint arXiv:1811.07112*, 2018.

[29] W. Li, C. Pan, R. Zhang, J. Ren, Y. Ma, J. Fang, F. Yan, Q. Geng, X. Huang, H. Gong *et al.*, "AADS: Augmented autonomous driving simulation using data-driven algorithms," *arXiv preprint arXiv:1901.07849*, 2019.

[30] D. J. Fremont, X. Yue, T. Dreossi, S. Ghosh, A. L. Sangiovanni-Vincentelli, and S. A. Seshia, "Scenic: Language-based scene generation," *CoRR*, vol. abs/1809.09310, 2018.

[31] Z. Liu, M. Shen, J. Zhang, S. Liu, H. Blasinski, T. Lian, and B. Wand ell, "A system for generating complex physically accurate sensor images for automotive applications," *arXiv e-prints*, p. arXiv:1902.04258, Feb 2019.

[32] W. Maddern, G. Pascoe, C. Linegar, and P. Newman, "1 year, 1000 km: The Oxford Robotcar dataset," *The International Journal of Robotics Research*, vol. 36, no. 1, pp. 3–15, 2017.

[33] H. Porav, W. Maddern, and P. Newman, "Adversarial training for adverse conditions: Robust metric localisation using appearance transfer," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, May 2018, pp. 1011–1018.

[34] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2015, pp. 234–241.

[35] D. Eigen and R. Fergus, "Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture," in *2015 IEEE International Conference on Computer Vision (ICCV)*, Dec 2015, pp. 2650–2658.

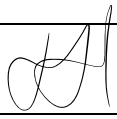
Statement of Authorship for joint/multi-authored papers for PGR thesis

To appear at the end of each thesis chapter submitted as an article/paper

The statement shall describe the candidate's and co-authors' independent research contributions in the thesis publications. For each publication there should exist a complete statement that is to be filled out and signed by the candidate and supervisor **(only required where there isn't already a statement of contribution within the paper itself).**

Title of Paper	Generating All the Roads to Rome: Road Layout Randomization for Improved Road Marking Segmentation
Publication Status	Published
Publication Details	T. Bruls, **H. Porav**, L. Kunze and P. Newman, "Generating All the Roads to Rome: Road Layout Randomization for Improved Road Marking Segmentation," *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, Auckland, New Zealand, 2019, pp. 831-838, doi: 10.1109/ITSC.2019.8916785.

Student Confirmation

Student Name:	Horia Porav		
Contribution to the Paper	- Generating initial ideas regarding domain randomization using data synthesis - General paper editing - Giving advice on experiments - Defined the training objectives in Section III-C of the paper		
Signature		Date	03 October 2020

Supervisor Confirmation

By signing the Statement of Authorship, you are certifying that the candidate made a substantial contribution to the publication, and that the description described above is accurate.

Supervisor name and title: Professor Paul Newman			
Supervisor comments			
Signature		Date	05-10-2020

This completed form should be included in the thesis, at the end of the relevant chapter.

5.4 Imminent Collision Mitigation with Reinforcement Learning and Vision

Problem: What might an autonomous vehicle do during an accident, including the cases where a collision is inevitable? Modern Advanced driver-assistance systems (ADAS) are generally relegated to either straight-line braking, or simple lane changing. The major drawback of braking-only systems is that they cannot avoid collisions where the distance to the obstacle is smaller than the total stopping distance. Lane-changing systems either use a simple heuristic (with or without path planning) or learn a policy using Reinforcement Learning in a simplified simulation which does not account for full scene information.

Solution: We construct a model, given camera images as input, that is capable of learning and predicting the dynamics of obstacles, cars and pedestrians, and train our policy using this model. We assume that the input images can easily be processed into semantic segmentation maps using the domain adaptation, de-noising and data generation techniques described thus far in this manuscript, and thus design the model to work in the semantic segmentation space, drastically reducing the complexity and parameter requirements for some of its components.

Two policies that control both braking and steering are compared against a baseline where the only action taken is (conventional) braking in a straight line. The two policies are trained using two distinct reward structures, one where any and all collisions incur a fixed penalty, and a second one where the penalty is calculated based on already established delta-v models of injury severity. The results show that both policies exceed the performance of the baseline, with the policy trained using injury models having the highest performance.

Status: Published at ITSC2018

Imminent Collision Mitigation with Reinforcement Learning and Vision

Horia Porav and Paul Newman

***Abstract*— This work examines the role of reinforcement learning in reducing the severity of on-road collisions by controlling velocity and steering in situations in which contact is imminent. We construct a model, given camera images as input, that is capable of learning and predicting the dynamics of obstacles, cars and pedestrians, and train our policy using this model. Two policies that control both braking and steering are compared against a baseline where the only action taken is (conventional) braking in a straight line. The two policies are trained using two distinct reward structures, one where any and all collisions incur a fixed penalty, and a second one where the penalty is calculated based on already established delta-v models of injury severity. The results show that both policies exceed the performance of the baseline, with the policy trained using injury models having the highest performance.**

I. Introduction

Both in the case of human-driven vehicles and autonomous vehicles, ensuring the safety of all traffic participants is paramount and often *the* prime motivator in autonomous vehicle research. Part of this endeavour is considering what to do if a situation arose in which some sort of collision was inevitable. We need not here exhaustively contemplate the many conceivable causes of such a situation - it could be entirely due to uncontrollable third-party decision or an "act-of-god" event. However, we ought to consider what appropriate actions can be taken in the moments after it becomes apparent that a collision is likely or inevitable and before impact occurs. There is potential to reduce the total harm and injuries sustained during a collision scenario, especially in the case where emergency braking systems by themselves are unable to prevent a collision [1].

Current approaches can be grouped under two major categories:

- emergency systems that only brake, without any steering input[2],[1]
- emergency systems that are capable of both braking and steering, usually in the form of lane changing[3],[4]

The major drawback of braking-only systems is that they cannot avoid collisions where the distance to the obstacle is smaller than the total stopping distance. Brake-and-steer systems attempt to improve this situation by triggering a lane change or a swerve if the system detects that straight line braking is not enough. However, to date and to the best of our knowledge, these systems either use a simple heuristic(with or without path planning[4]) or learn a policy using Reinforcement Learning[1] in a simplified simulation.

In this work we build a recurrent model that predicts the movement of obstacles, pedestrians and vehicles in a way inspired by [5] and [1]. Under the hood, our model has the following key components: a Variational Autoencoder (VAE) which reduces the dimensionality of the data and compresses incoming observations into a latent representation, an RNN which learns to predict the next latent representation given the current one and a controller that uses these latent representations and learns to take actions using Deep Deterministic Policy Gradients (DDPG)[6]. We extend the simulation system of [7] to include both pedestrian and car collisions, along with randomized pedestrians on sidewalks and vehicular traffic on the road around the collision site and focus on Time to Collision (TTC) of 1.5 seconds and under [1].

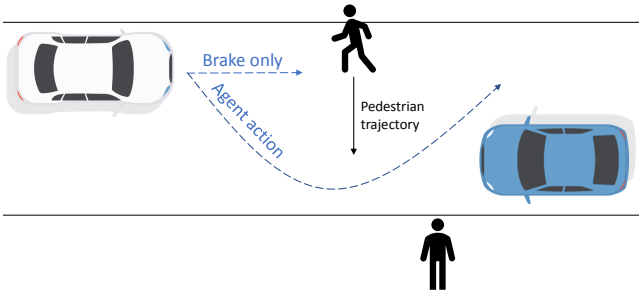


Fig. 1. The imminent collision mitigation system is trained to choose the action that leads to the least amount of injury, as measured by empirical models which relate delta-v to measures of injury and loss. Our simulated environment contains a mixture of pedestrians and vehicles that can take part in an imminent collision scenario. In this particular example we show a pedestrian that is running across the street in front of the vehicle containing the collision mitigation system, along with incoming traffic and a pedestrian on the sidewalk.

The major assumptions that we make is that other traffic participants cannot themselves actively avoid the collision by braking or changing direction. We present our results in V and show that training a brake-and-steer policy in a complex simulation could reduce the amount of pedestrian and car occupant injury compared to the baseline of braking in a straight line, especially for time-to-collision (TTC)[9] values between 0.5 and 1.5 seconds. In our tests, using standard models of injury as a function of delta-v for vehicle occupants and as a function of impact speed for pedestrians, we see a reduction of up to 60% in the collision rate for pedestrians and car occupants.

II. Related Work

In this section we review related work and collision mitigation and on injury risk estimation.

A. Related Work

In [2], the authors present an assessment algorithm that determines if collisions are avoidable by taking into consideration all vehicles detected and their constraints and

Authors are from the Oxford Robotics Institute, Dept. Engineering Science, University of Oxford, UK. {horia, pnewman}@robots.ox.ac.uk

having the emergency braking being triggered only when collision becomes unavoidable. Similar to our work, their approach does not rely on assumptions regarding vehicle path or road infrastructure, but only vehicle constraints and other detected traffic participants, and their approach can be applied to various crash scenarios, unlike other approaches that only model rear-end collisions. In an earlier study, [3] present a tool for determining the optimal maneuver in collision-avoidance scenarios. A clearance curve is generated for varying speeds, along which hazard states can be avoided only by an optimal maneuver. Their result show that at high speeds, lane-change is more advantageous than a full stop, this highlighting the importance of a more diversified action space. In [8] a cruise control system is presented, aimed at reducing collisions within a platoon using a collision avoidance mode involving either emergency brake or collision avoidance action whenever new data is acquired from sensors. [9] propose a model based collision avoidance algorithm in order to approximate a set of actions (break, accelerate or steer) that can be taken by the driver and determine whether immediate assistance is required. Similar to our work, the model can be used for all traffic scenarios and all kinds of traffic participants. [4] propose a nonlinear model predictive control that aims at lowering the risk of other hazardous situations that result from the vehicles attempts to avoid a collision. The model takes into account the vehicle dynamics (minimum and maximum steering wheel angle and acceleration) by implementing a 2-level architecture: a controller that provides the path/state which avoids the collision, and another controller that aids the vehicle at following the proposed path.

A system capable of early detecting a pedestrians intention of crossing the road and performing an evasive maneuver if avoidance by braking is impossible is presented in [10]. However, they rely on the existence of a Road Side Unit placed in dangerous road spots in order to detect pedestrian intention and send this information to the On Board Unit placed in the vehicle.

As opposed to most previous research, [1] propose a model-free collision avoidance system using Deep Reinforcement Learning (DRL). They derive a balanced reward function for an autonomous braking system based on DRL, where the action space allows 4 choices: no braking, weak braking, medium and strong. The reward function consists of one component that penalizes the agent for braking too early while the second one is a penalty for collision with the pedestrian and takes into account the velocity of the vehicle in order to reflect the degree of damage. Due to an unstable learning performance (collisions rarely occuring), the authors use memory replay[11] in order to remind the agent of collisions, whatever the present policy. The results show that collision rates are 0 for time-to-collision larger or equal to 1.5 seconds. This prompts us to study TTC values below $1.5s$.

B. Injury risk literature

Various metrics are being employed in research as a measure of crash severity, while this makes for grim reading they do provide widely adopted quantitative, data driven models of accident outcomes. They include the Acceleration Severity Index (ASI), Occupant Impact Velocity (OIV) and Delta-V. However, [12] show that the former two do not offer significant predictive advantage over the latter. Since its emergence in the 1970s, Delta-V has been the traditional metric for crash severity and is defined as the absolute change between pre-collision velocity and post-collision velocity, with the assumption that larger differences in velocities are correlated with more severe injuries ([13]):

$$\Delta v = |v_{after} - v_{before}| \tag{1}$$

In [14] the fatality risk of pedestrians as given by the vehicle's speed on impact using the GIDAS dataset (German In-Depth Accident Study) is studied. The dataset includes data from 2127 pedestrians that were involved in accidents between 1999 and 2007. They present a now widely-adopted approximation of the fatality risk as:

$$P = \frac{1}{1 + e^{6.9 - 0.09|v|}} \tag{2}$$

where v is the velocity when impacting the pedestrian (km/h).

The authors of [15] introduced a widely used model for the probability of a severe injury of car occupants during a frontal impact, which was later confirmed and extended by [16] in a larger study. Both papers analyzed crash tests that were published by National Highway Traffic Safety Administration (USA) in order to determine how the risk of severe injury of the front seat occupants of a passenger car is influenced by the impact speed during a frontal crash. They approximate the fatal injury risk to be given by

$$P = \left(\frac{\Delta v}{71}\right)^4 \tag{3}$$

where Δv is expressed in miles/hour, and the value of P is clamped to 1.0. Moving forward, and without prejudice, we will be using Equations 2 and 3 as quantitative models of accident severity. Nothing in the coming sections requires these particular models and they could be substituted out at the reader's discretion. What we propose here is a framework rather than a fixed implementation or a judgment call on the correct way to measure accident severity.

III. Learning to Avoid and Mitigate Collisions

Here we investigate the role of machine learning in managing and mitigating collisions. Our goal is a lower collision cost, especially in the cases of lower time-to-collision values. Our "agent" is the vehicle we can execute steering and brake control over. At runtime our agent is provided only with a sequence of images of the road scene. Given this we need to learn a mapping from images to a 2-channel control sequence which improves outcomes over conventional baseline straight line braking. In this way the system can choose as how and when to brake and steer as a function of instantaneous speed

and time to collision. We will consider and compare two loss/reward strategies which will be explained in Section III-B. In the remaining subsections we assume some familiarity with the recent reinforcement learning literature in which for completeness we give the details of our learning procedure. However the experiment and results in Sections IV and V can be consumed / read as independently of this section.

A. Reinforcement learning using DDPG

We use Deep Deterministic Policy Gradient (DDPG)[6] because it is off-policy, making it suitable for rapidly discovering optimal policies in simulation by employing a stochastic policy for improved exploration, while tackling the much easier task of learning a deterministic policy. DDPG employs two networks, a policy network (the actor) and a Q-value estimator network (the critic). Using the current state as as input to both networks, the actor will output actions from a continuous action space, while the critic estimates a Q value based on the output of the actor. The actor network weights are then updated using deterministic policy gradient[17], and the critic weights are updated using the gradient of the temporal-difference signal, similar to [17].

DDPG uses experience replay[11] as a variance-reduction technique and target networks[17] to stabilize training. As a difference from DPG[17], rather than directly copying the weights of the local networks θ_L to the target networks θ_T every N steps, DDPG performs a soft update of its target network weights θ_T using an update rate τ between 0 and 1:

$$\theta_T = (1-\tau) * \theta_T + \tau * \theta_L \tag{4}$$

The local critic network is updated by minimizing the loss function:

$$\mathcal{L}(\theta_{\mathrm{L}}^{\mathrm{Q}}) = \frac{1}{N}\sum_i \Big((R+\gamma Q(s_{i+1}, a_{i+1}|\theta_L^Q)) - Q(s_i, a_i|\theta_T^Q)\Big)^2 \tag{5}$$

Here, R is the reward, Q is the Q-value approximator function, the local critic network is parametrized by θ_L^Q, the target critic network is parametrized by θ_T^Q, γ is the time-horizon discount factor and N is the size of the minibatch sampled from the experience replay buffer. The current state and action are represented by s_i and a_i, and the future state and action are represented by s_{i+1} and a_{i+1}. The local actor network is updated using the gradient of the local critic network with respect to the actions a_i, multiplied by the gradient of the local actor network with respect to θ_L^μ (chain rule) [17]:

$$\nabla_{\theta_L^\mu}\mu = \frac{1}{N}\sum_i \left[\nabla_a Q(s_i, a_i|\theta_L^Q)\nabla_{\theta^\mu}\mu(s_i|\theta_L^\mu)\right] \tag{6}$$

Here, μ is the policy function, the local actor network is parametrized by θ_L^μ and the target actor network is parametrized by θ_T^μ. The parameterisation, origin, and form of the reward R, action a, and state s will now be explained in the following subsections.

B. Reward structure

As this research aims to be an objective comparison of the agent's behaviour as a result of the learning strategies, we propose two reward structures for the reinforcement-learning agent: one where the penalties are uniform (-1 for any type of collision) and another one that follows the literature on severe risk injury and pedestrian fatality risk.

While the work of [1] focuses on pedestrians only, our proposed reward structures will also take into account another category of traffic participants: car-occupants. Secondly, as injury risk has been approximated in previous research, we hope that the second reward structure will more closely follow real collision scenarios, where car-occupants are better protected than pedestrians due to the car's energy absorption properties. Additionally, this empirical approach allows us to avoid any hyper-parameters or weights in the reward functions.

Reward Strategy 1 The first reward function is defined as:

$$R_1 = -(nr_{ped} + nr_{occ}) \tag{7}$$

Here nr_{ped} represents the number of pedestrians and nr_{occ} represents the number of car occupants involved in the accident. This strategy simply counts the number of people involved in a collision.

Reward Strategy 2 The second reward function combines the empirical models of "injury" Equations (2) and (3) and so explicitly accounts for the degree of injury, and is defined as:

$$R_2 = -\Big(nr_{ped}\frac{1}{1+e^{6.9-0.09|v|}} + nr_{occ}\Big(\frac{\Delta v * 0.621}{71}\Big)^4\Big) \tag{8}$$

In the occupant injury component, 0.621 is used to convert the speed from km/h in miles/hour to fit Joksch's[15] original model.

Note that when it comes to comparing outcomes using Strategy 1 or Strategy we will use, in both cases, the empirically derived Equations 2 and 3 to numerically and continuously represent total injury.

C. Action space

In line with previous model-based research that includes steering and braking as collision-avoidance options, we propose an extension of [1] agent's action space to include both braking and steering. Additionally, we let the actions be continuous instead of discrete in order to allow more degree of control.

The actions a_t are encoded as a pair of integer values between -1 and 1, the first value encoding the steering angle, the second value encoding the position of the throttle(for positive values) or brake(for negative values). For steering, the range -1 to 1 maps the range between the maximum left and the maximum right steering angles. For braking, the range -1 to 0 maps a braking force causing a decelleration between $-9.8m/s^2$ to $0m/s^2$.

D. General Simulation Architecture

Clearly in this context the use of simulation is required. One cannot readily or ethically explore the space of accident vs. action in real life. Accordingly, our system as presented here appears as a back-end used in conjunction with CARLA's semantic segmentation output based on Citiscapes[18] classes,which is fed to our system in the shape of 64*64 1-channel images, with 13 corresponding classes(None, Buildings, Fences, Other, Pedestrians, Poles, RoadLines, Roads, Sidewalks, Vegetation, Vehicles, Walls, Traffic Signs) encoded as integer values between 0-12. Our choice is based on the recent performance of real-time semantic segmentation approaches[19],[20], and we assume that such a system is now readily available as an input preprocessor. Similar to [5], we develop and train our system in multiple phases, as this allows us to use supervised learning where possible and to test individual components. We begin by recording observations obs_t and actions a_t produced by an agent exploring the environment of CARLA using the built-in waypoint based autopilot[7] with Gaussian noise added to the controls. The observations obs_t that we collect are represented by tuples of a semantically-segmented(using Citiscapes classes) images of the environment I_t, as shown in Fig. 4 and the vehicle forward velocity v_t, for each time t. The actions a_t are encoded as described in section III-C. In the first phase, a Variational Autoencoder (VAE)[21] learns to compress the incoming observations from the environment obs_t into a latent representation z_t with a Gaussian distribution. In the second phase, we train a Recurrent Neural Network (RNN) to predict the next latent representation z_{t+1}, given the current latent representation z_t and the current action a_t. Finally,in the third phase, we train the agent using DDPG by encoding the current incoming frame using the VAE, predicting the next state using the recurrent network and using the latent representation z_t and the predicted state z_{t+1} as input to our controller, giving the agent access to the current state of the environment and to a prediction of the future state. The agent then takes an action in the simulated Carla environment. An overview of the architecture of the system is shown in Fig. 3.

E. Phase 1

Following [5], in Phase 1 we train a Variational Autoencoder (VAE)[21] to compress incoming 2D observations (images) obs_t into a 128-length latent representation z_t with a Gaussian distribution. The literature on VAEs is vast, and the reader is invited to study, for example, [22] and [21] for a more in-depth look at VAEs. The main reasoning behind using a VAE is two-fold: 1)we wish to reduce the dimensionality of our data such that it becomes easier to train the RNN and controller and 2)we wish to only keep features that are relevant to encoding the position and type of objects found in the incoming semantically-segmented images.

F. Phase 2

In phase 2, we first convert all of the recorded 2D image observations obs_t into latent space representations z_t. Following this we produce training data by creating a sequence of tuples (z_t, z_{t+1}, a_t) for t between 1 and N, the total number of recorded frames in a sequence. We then train an LSTM-RNN[23] to predict the next latent representation z_{t+1} given the current latent representation z_t and the current action a_t using the sequence of tuples. An example of applying the VAE decoder on the predicted latent representations can be seen in Fig 2. The RNN layout uses an LSTM cell with 512 hidden units. Details about the implementation can be referenced from [5].

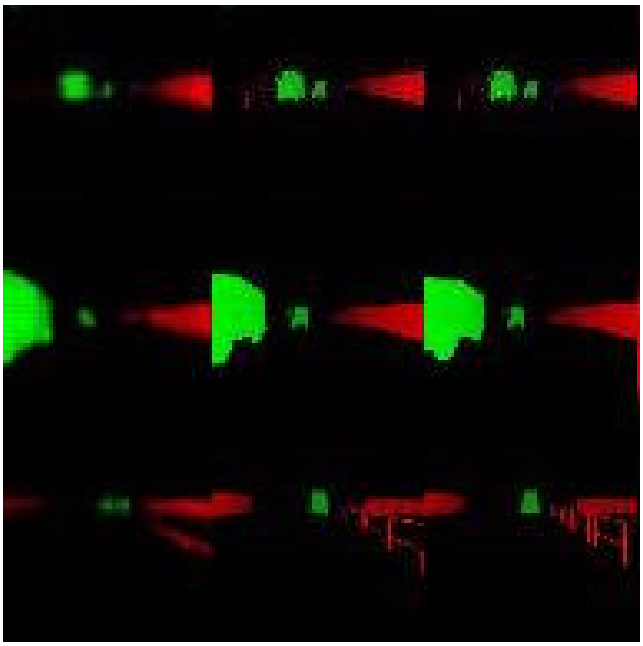


Fig. 2. Images obtained by applying the VAE decoder on latent representations. The right column represents decoded state z_{t-1}. The middle column represents decoded state z_t. The left column represents the output of the decoder applied to state z_{t+1} as predicted by the RNN using z_t and the internal state of the LSTM cell. The green blob represents a semantically-segmented vehicle coming towards the camera, while the red patches represent fences on the side of the road.

G. Phase 3

In phase 3, we train our agent in the CARLA environment using DDPG, as described in III-A. For each incoming observation obs_t, we transform it using the VAE into a latent representation z_t and predict a future state z_{t+1}. The input to both the policy and the Q-value functions is a concatenation of z_t and z_{t+1}, giving the agent access to the current state of the environment and to a prediction of the future state.

IV. Experimental Setup

A. Generating and Simulating Collision Scenarios

To generate a diverse set of imminent collision scenarios, we randomly sample collision parameters from the following distributions:

- $TTC \sim \mathcal{U}(0.25,\ 1.5)s$
- Collision velocity $v_c \sim \mathcal{U}(1,\ 30)m/s$
- Pedestrian velocity $v_p \sim \mathcal{U}(1,\ 5)m/s$
- Number of cars $n_c \sim [0...10]$
- Number of other pedestrians $n_p \sim [0...10]$

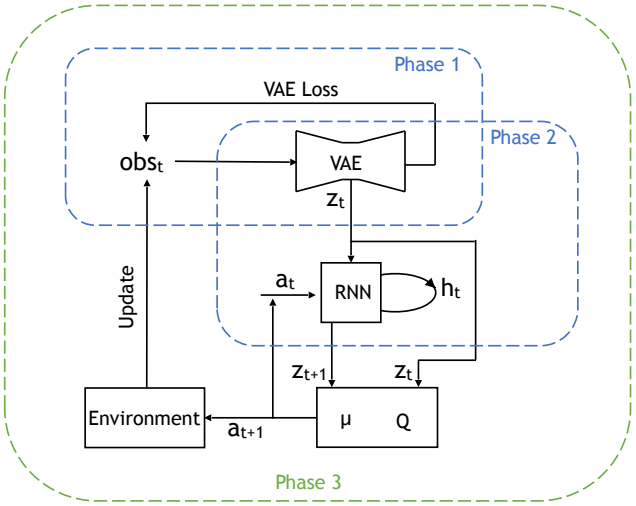


Fig. 3. An overview of our architecture. The dashed rounded rectangles encompass the components used in each phase, as described in III-D and the following subsections.

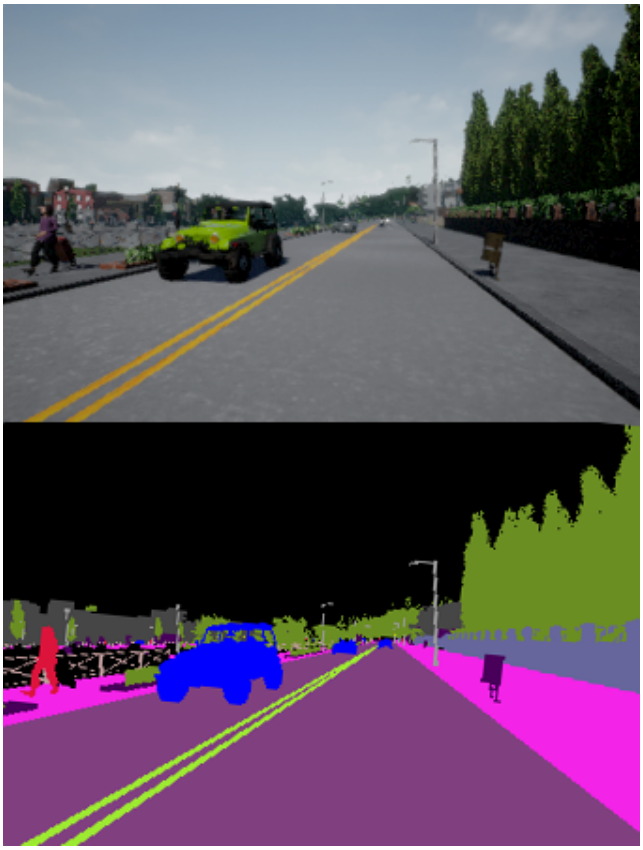


Fig. 4. Images from the CARLA environment. On the top we show an RGB image depicting a street view with traffic. On the bottom we show the semantic segmentation of the top image, with cars in blue and pedestrians in red.

- Probability of pedestrian infraction $P_p \sim \mathcal{U}(0,\ 1)$
- Probability of car infraction $P_c \sim \mathcal{U}(0,\ 1)$

For each scenario, if the probability of pedestrian collision is higher than 0.5, we generate a random trajectory that intersects the pedestrian with the car at a point that will be reached in TTC seconds if the velocity of the car is maintained. Similarly, if the probability of a car collision is higher than 0.5, we spawn a vehicle that will intersect with the car at a point that is TTC seconds away if the velocity is kept constant.

B. Training

We train the VAE for 2 epochs on a dataset containing 200000 frames. We train the RNN for 10 epochs on 850 sequences of approximately 230 frames each, captured in the CARLA environment at 15FPS, with a time-limit of 15 seconds for each sequence. We make use of the Adam [24] solver for all training, with an initial learning rate of 0.0001, and a batch size of 32. No other hyperparameters are set or used.

C. Testing

To test the effectiveness of our policies, we create a bank of test scenarios by sweeping through TTC, collision speeds and traffic participant setups, with fixed seeds for any randomly-generated component to allow for repeatability. We then test the 2 policies and the baseline policy using identical test scenarios, and record for each collision scenario, along with the scenario setup, the following metrics:

- Whether there was a collision with a pedestrian
- Whether there was a collision with a vehicle
- Whether there was a collision with a static object
- Injury severity for each collision, for each participant
- Velocity of car at the beginning of the collision
- Whether the car has left its lane

These metrics allow us to study,for each policy type and for each Time-To-Collision interval:

- The percentage of pedestrian collisions avoided
- The percentage of car collisions avoided
- The percentage of collisions with static obstacles (e.g. poles, walls etc)
- The percentage of severe injuries

D. Performance

The agent network takes approximately 100 ms for images with a resolution of 64×64 on an Intel I7 processor and 30 ms for the same resolution on an NVIDIA Tesla GPU.

V. Results

Fig. 5 shows, on each column, in the following order: a)Pedestrian collision rate percentage point improvement over baseline; b)pedestrian severe injury risk percentage point improvement over baseline; c)car collision rate percentage point improvement over baseline; d)car occupant severe injury risk percentage point improvement over baseline; All are expressed as a function of TTC and initial velocity. First row presents results for the policy trained using Reward $R1$, while the second row presents results for the policy trained using Reward $R2$. We notice that both policies offer very favourable outcomes as compared to the baseline of braking-only.

We observe a significant increase in the number of collisions avoided as compared to baseline, especially for TTC

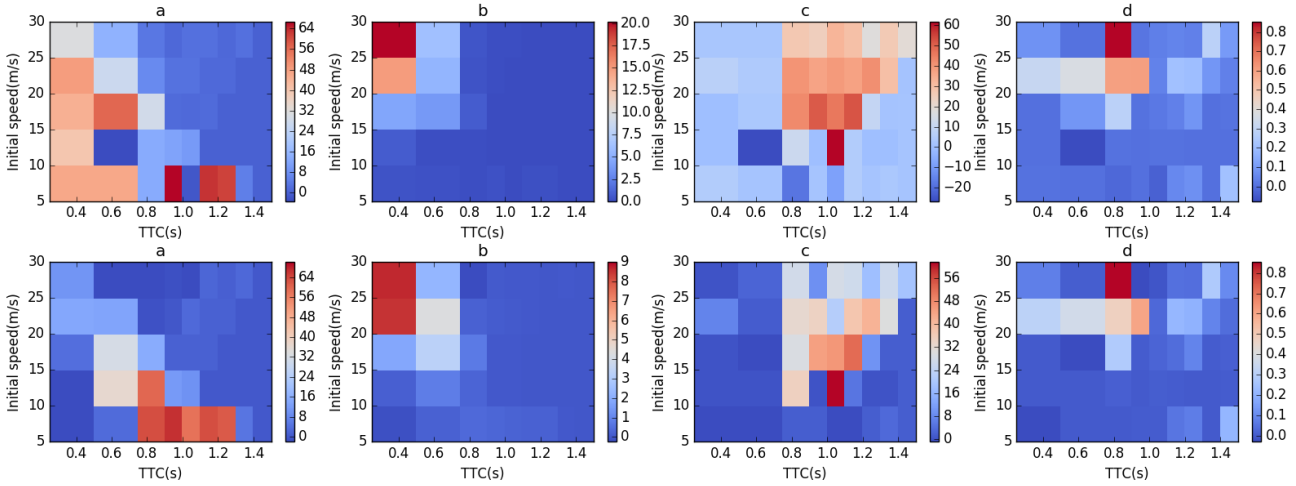


Fig. 5. First row presents results for the policy trained using Reward 1. Second row presents results for the policy trained using Reward 2. Column descriptions: a) Pedestrian collision rate percentage point reduction over baseline; b) Pedestrian severe injury risk percentage point reduction over baseline; c) Car collision rate percentage point reduction over baseline; d) Car occupant severe injury risk percentage point reduction over baseline; The more red an area is(i.e, the higher the numbers), the lower the rate of collision or risk of severe injury compared to a baseline of straight-line braking-only.

Policy type	**TTC 0.5s**			**TTC 0.75s**			**TTC 1.0s**			**TTC 1.5s**		
	10 (m/s)	20 (m/s)	30 (m/s)	10 (m/s)	20 (m/s)	30 (m/s)	10 (m/s)	20 (m/s)	30 (m/s)	10 (m/s)	20 (m/s)	30 (m/s)
Straight-line braking	0.886	5.40	6.08	0.263	0.360	1.650	0.015	0.064	0.828	0.0	0.0	0.0
Brake&Steer, reward 1	0.168	2.462	2.860	0.0	0.189	1.799	0.0	0.0	0.0	0.0	0.0	0.0
Brake&Steer, reward 2	0.054	1.087	3.487	0.0	0.172	1.633	0.0	0.0	0.0	0.0	0.0	0.0

TABLE I

PEDESTRIAN PROBABILITY (%) OF SEVERE INJURY

Policy type	**TTC 0.5s**			**TTC 0.75s**			**TTC 1.0s**			**TTC 1.5s**		
	10 (m/s)	20 (m/s)	30 (m/s)	10 (m/s)	20 (m/s)	30 (m/s)	10 (m/s)	20 (m/s)	30 (m/s)	10 (m/s)	20 (m/s)	30 (m/s)
Straight-line braking	0.0	0.372	0.829	0.008	0.603	1.240	0.006	0.042	0.015	0.001	0.002	0.245
Brake&Steer, reward 1	0.001	0.467	0.579	0.001	0.754	0.495	0.0	0.052	0.001	0.0	0.0	0.0
Brake&Steer, reward 2	0.0	0.001	0.663	0.0	0.124	0.0	0.0	0.0158	0.0	0.0	0.0	0.0

TABLE II

CAR OCCUPANT PROBABILITY (%) OF SEVERE INJURY

TTC(s)	0.5	0.75	0.9	1.1	1.3	1.5
Col. rate % (ours)	**21.8**	**5.0**	**1.4**	**0.0**	**0.0**	0.0
Col. rate % [1]	NA	NA	61.29	18.85	0.74	0.0

TABLE III

PEDESTRIAN COLLISION RATE (%) COMPARISON BETWEEN OUR RESULTS AND THOSE OF CHAE ET AL.[1].

values between $0.25s$ and $1.0s$ for pedestrians, and for TTC values between $0.75s$ and $1.3s$ for car collisions. We observe no overall degradation of performance as compared to the braking-only baseline, suggesting that the controller has learned to effectively gauge when simple braking is enough and when steering input is needed to mitigate the effects of a collision.

We also observe a significant improvement in the risk of severe injury for pedestrians, especially in the problematic[1] area of TTC under $0.9s$ and speeds over $15m/s$.

Table I compares the performance of the trained agent against the braking-only baseline for pedestrian risk of severe injuries. Both policies lead to a lower incidence of severe injuries compared to straight-line braking, with the policy trained using $R2$ showing outright improvement in all areas, with the exception of TTC $0.75s$ at $30m/s$.

Table II compares the performance of the trained agent against the braking-only baseline for car occupant injuries. We observe that the policy trained using $R2$ shows an outright improvement in all areas.

Table III compares the pedestrian collision results obtained using our $R2$ policy with the results published by [1]. We show a reduction in the rate of severe injuries across the whole TTC range between $0.9s$ and $1.5s$, and show much lower collision rates at TTC of $0.75s$ and $0.5s$ than the rates reported by [1] for TTC of $0.9s$.

VI. CONCLUSIONS

This paper has asked an unusual question: "what might an autonomous vehicle do during an accident, including the cases where a collision is inevitable?". There is, of course, no obviously correct answer, nor do we intend here to offer a view on what *should* happen. Instead, we ask what control

behaviours are learned if we utilize existing models of injury severity and at the same time leverage the ability of an Autonomous Vehicle to control its complete trajectory - thereby differentiating from conventional ADAS systems.

Our architecture makes several contributions to the topic of investigating collision mitigation using simulation and machine learning. The training of the VAE distills relevant information from the CARLA simulations such that both the RNN and the controller become much easier to train. And of course the use of Reinforcement Learning in the controller allows the system freedom to learn complex control profiles.

We compare our outcomes, which allow for steering and braking control, to those arising from straight-line emergency braking systems. We trained our system using two reward constructs - one which simply counts participants in collisions, and one which takes into account empirically derived measures of severity. In all three cases (our two strategies and the baseline) we measure the outcome using the same empirical measures of injury, Eq.(2) and Eq.(3).

We have presented a vision-based emergency collision mitigation system that, under simulation, reduces the total severity of human injury. Our empirical results show that allowing the vehicle to brake *and* steer yields preferable outcomes when used with the policies we learned. In some cases by a 60% margin.

VII. Acknowledgements

This work was supported by an Oxford-Google DeepMind Graduate Scholarship and EPSRC Programme Grant EP/M019918/1. The authors would also like to thank Valentina Musat and Tom Bruls for their help.

References

[1] H. Chae, C. M. Kang, B. Kim, J. Kim, C. C. Chung, and J. W. Choi, "Autonomous braking system via deep reinforcement learning," *CoRR*, vol. abs/1702.02302, 2017.

[2] N. Kaempchen, B. Schiele, and K. Dietmayer, "Situation assessment of an autonomous emergency brake for arbitrary vehicle-to-vehicle collision scenarios," *Trans. Intell. Transport. Sys.*, vol. 10, no. 4, pp. 678–687, Dec. 2009.

[3] Z. Shiller and S. Sundar, "Emergency lane-change maneuvers of autonomous vehicles," vol. 120, 03 1998.

[4] C. Choi, Y. Kang, and S. Lee, "Emergency collision avoidance maneuver based on nonlinear model predictive control," in *2012 IEEE International Conference on Vehicular Electronics and Safety (ICVES 2012)*, July 2012, pp. 393–398.

[5] D. Ha and J. Schmidhuber, "World Models," *ArXiv e-prints*, Mar. 2018.

[6] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," *ArXiv e-prints*, Sept. 2015.

[7] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "CARLA: An open urban driving simulator," in *Proceedings of the 1st Annual Conference on Robot Learning*, 2017, pp. 1–16.

[8] A. Ferrara and C. Vecchio, "Second order sliding mode control of vehicles with distributed collision avoidance capabilities," *Mechatronics*, vol. 19, no. 4, pp. 471 – 477, 2009, robotics and Factory of the Future, New Trends and Challenges in Mechatronics.

[9] M. Brannstrom, E. Coelingh, and J. Sjoberg, "Model-based threat assessment for avoiding arbitrary vehicle collisions," *IEEE Transactions on Intelligent Transportation Systems*, vol. 11, no. 3, pp. 658–669, Sept 2010.

[10] S. Khler, B. Schreiner, S. Ronalter, K. Doll, U. Brunsmann, and K. Zindler, "Autonomous evasive maneuvers triggered by infrastructure-based detection of pedestrian intentions," in *2013 IEEE Intelligent Vehicles Symposium (IV)*, June 2013, pp. 519–526.

[11] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, p. 529, feb 2015.

[12] D. J. Gabauer and H. C. Gabler, "Comparison of roadside crash injury metrics using event data recorders," *Accident Analysis and Prevention*, vol. 40, no. 2, pp. 548 – 558, 2008.

[13] S. G. Shelby, "Delta-v as a measure of traffic conflict severity," 2011.

[14] E. Rosen and U. Sander, "Pedestrian fatality risk as a function of car impact speed," vol. 41, pp. 536–42, 06 2009.

[15] H. C. Joksch, "Velocity change and fatality risk in a crash–a rule of thumb." *Accident; analysis and prevention*, vol. 25 1, pp. 103–4, 1993.

[16] L. Evans, "Driver injury and fatality risk in two-car crashes versus mass ratio inferred using newtonian mechanics," *Accident Analysis and Prevention*, vol. 26, no. 5, pp. 609 – 616, 1994.

[17] D. Silver, G. Lever, N. Heess, T. Degris, D. Wierstra, and M. Riedmiller, "Deterministic Policy Gradient Algorithms," in *ICML*, Beijing, China, June 2014.

[18] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," June 2016.

[19] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking Atrous Convolution for Semantic Image Segmentation," *ArXiv e-prints*, June 2017.

[20] C. Peng, X. Zhang, G. Yu, G. Luo, and J. Sun, "Large Kernel Matters – Improve Semantic Segmentation by Global Convolutional Network," *ArXiv e-prints*, Mar. 2017.

[21] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *CoRR*, vol. abs/1312.6114, 2013.

[22] D. J. Rezende, S. Mohamed, and D. Wierstra, "Stochastic backpropagation and approximate inference in deep generative models," in *Proceedings of the 31st International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, E. P. Xing and T. Jebara, Eds., vol. 32, no. 2. Bejing, China: PMLR, 22–24 Jun 2014, pp. 1278–1286.

[23] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.

[24] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference for Learning Representations (ICLR), San Diego*, 2015.

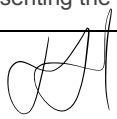
Statement of Authorship for joint/multi-authored papers for PGR thesis

To appear at the end of each thesis chapter submitted as an article/paper

The statement shall describe the candidate's and co-authors' independent research contributions in the thesis publications. For each publication there should exist a complete statement that is to be filled out and signed by the candidate and supervisor **(only required where there isn't already a statement of contribution within the paper itself).**

Title of Paper	Imminent Collision Mitigation with Reinforcement Learning and Vision
Publication Status	Published
Publication Details	**H. Porav** and P. Newman, "Imminent Collision Mitigation with Reinforcement Learning and Vision," *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, Maui, HI, 2018, pp. 958-964, doi: 10.1109/ITSC.2018.8569222.

Student Confirmation

Student Name:	Horia Porav		
Contribution to the Paper	All work except advice and editorial changes. My contributions included: - Defining the initial ideas revolving around training an agent to both steer and brake in the event of an imminent collision - Defining the two training rewards, using widely known measures of impact severity - Designing and implementing the whole agent pipeline, including training procedures - Experimenting with future frame prediction - Heavily modifying the CARLA driving simulator to support pedestrian entity control - Designing the collision scenarios for the simulation - Running all experiments and gathering all the data - Collating the data and producing the results - Writing the paper, including diagrams, tables and figures - Presenting the paper at ITSC 2018		
Signature		Date	03 October 2020

Supervisor Confirmation

By signing the Statement of Authorship, you are certifying that the candidate made a substantial contribution to the publication, and that the description described above is accurate.

Supervisor name and title: Professor Paul Newman			
Supervisor comments			
Signature		Date	05-10-2020

This completed form should be included in the thesis, at the end of the relevant chapter.

6 Discussion

Contents

6.1 Domain adaptation 111
6.2 Image de-noising 113
6.3 Data generation 114
6.4 Overall conclusion 117

In this final chapter, we review and discuss the methods designed and devised in the previous chapters, concluding with a short remark on future directions.

6.1 Domain adaptation

Modern machine learning models that perform system-critical tasks such as segmentation and localization exhibit good performance and robustness under ideal conditions (i.e. daytime, overcast) but performance degrades quickly and often catastrophically when input conditions change. Further increasing the robustness of models is usually hindered by any combination of cost, duration, labelling efforts and availability or scarcity of data. Our approach to domain adaptation significantly reduces labelling efforts, reliance on scarce data and implicitly the cost of operation under a plurality of conditions.

Image to image translation was used to significantly improve place recognition and localisation under changing and adverse conditions, while reducing the number of mapping runs needed to successfully achieve reliable localisation, thus reducing the cost and complexity of mapping.

This idea is further extended by relaxing the constraint that generated images should be natural images, in a system that uses light-weight input adapters to pre-processes input images, irrespective of their appearance, in a way that makes them compatible with off-the-shelf computer vision tasks that are trained only on inputs with ideal conditions. Furthermore, by having a lower complexity of the domain adaptation model, real-time fine-tuning of the input adapters becomes possible. This system can be extended to any task that is differentiable or for which a cost function can be computed, and has modest supervision requirements, possessing the ability to run entirely self-supervised.

An interesting note here is that the images produced by our light-weight adapters frequently bear little resemblance – by visual inspection – to the natural images that were used to train the off-the-shelf models (usually under ideal conditions), but work great as inputs to these models, highlighting the multi-modality of the input to output transfer functions learned. This in turn raises questions regarding the suitability of metrics that rely on the similarity of features, such as Fréchet Inception Distance (FID) [51], in the context of image translation for practical domain adaptation or training data synthesis, as seen in the data generation section.

Light-weight image pre-processors could speed up the time to deployment for robots operating under varied real-world conditions and reduce labelling efforts by orders of magnitude. Additionally, under the self-supervised mode of operation, the reference labelled dataset can be translated into the style of the incoming camera frames, then processed through a target model and compared against the reference ground-truth to produce a proxy for model uncertainty.

6.2 Image de-noising

Out of the wide range of tasks that fall under the umbrella of image de-noising, we chose to tackle two problems that are relevant to field robotics, image de-raining and IPM de-noising and undistortion. GANs are intuitively a good choice for these tasks, since adversarial training is suitable for dealing with missing or incomplete data where a multi-modal solution is acceptable. Using a specific learning schedule, training stabilisation tricks and carefully selected data helps to keep GAN hallucinations to a minimum.

Segmentation tasks suffer from severely degraded performance on images affected by adherent rain drops and streaks. This is a common issue in outdoor robotic applications such as autonomous vehicles. Acquiring rainy images is time-consuming, expensive or impossible for many tasks or setups, especially in the case of supervised training, where ground truth data is needed. To solve this issue, we introduced methodology for a novel stereo dataset recorded using a system that allows one lens to be affected by real water droplets while keeping the other lens clear. The stereo rig allowed for variations in the distance between the lens and the surface affected by adherent droplets, mimicking different possible setups. Our qualitative experiments (extended by the *Rainy screens* data generation work) demonstrated that droplets deposited on a clear surface using a variable pressure nozzle are a good substitute for real rain droplets. Our quantitative experiments showed that a de-noising generative model trained using this data is effective at removing the effect of real water droplets, in the context of image reconstruction and road-marking segmentation.

Inverse Perspective Mapped images are useful for a number of robotics tasks (e.g. road marking detection, path planning) and have also seen increasing presence in ADAS systems of higher-end cars. Since far-away real-world features are represented by a smaller pixel area as compared to identical close-by features, a direct consequence of applying a full perspective transformation to the input image is increased unnatural blurring and stretching of features at further distance. Additionally, it is relatively hard to obtain training data for IPM models due to physical constraints for camera placement. Past approaches have used simulation

data from 3D game engines (e.g. CARLA [61]) which introduce a large domain shift or 2D projections of coloured point clouds, which are hard and expensive to capture and require involved post-processing steps. Being able to produce sharp and clear IPM images can boost the performance of a number of scene understanding tasks. Similarly to the case of image de-raining, GANs are intuitively and empirically suitable for IPM, especially in the presence of simplified assumptions such as a ground plane.

In contrast to existing approaches, we successfully addressed the IPM ground-truth availability issue by stitching slices of sequential frames using relative poses from a state-of-the-art visual odometry pipeline. This allowed for automatic creation of large datasets, with the only external input being an estimated ground plane homography.

We addressed the significant appearance change between the views by introducing an Incremental Spatial ResNet Transformer GAN, which divides the full perspective transformation into a series of smaller incremental perspective transformations, each followed by a refinement of the transformed feature space using a ResNet block. The intuition behind this is that the slight blurring that occurs as a result of each perspective transformation is restored by the ResNet block that follows it. We demonstrated reliable, qualitative results in different environments and under varying lighting conditions. Furthermore, we showed that the boosted IPM view allows for improved hierarchical scene understanding.

6.3 Data generation

While both the domain adaptation and image de-noising categories contained sizeable data generation components, the works under the present category dealt with a more general approach to data generation, with a larger invariance to the actual target domain. As data labelling is both a very important and very costly part of machine learning and system validation, improvements that reduce our reliance on labelling are an important research avenue. We have successfully investigated methods for automating scarce data collection, developed a simple extension to

cycle consistency GANs that improves the quality of the generated training data, improved road marker segmentation via scene synthesis and leveraged simulation data to demonstrate a collision mitigation system.

We extended the work carried out during our initial de-raining efforts by focusing on developing the simplest and cheapest automatic way of obtaining images affected by adherent water droplets with a clear ground-truth. Our experiments successfully demonstrated that a quality dataset can be collected by simply recording a high-resolution computer monitor through a glass pane affected by adherent water droplets. The procedure is completely automatic once set-up and only requires an initial extrinsic calibration between the camera and the monitor frame and an intrinsic calibration of the camera, which can all be done with off-the-shelf tools. The ease of setup and the invariance to the image stream source opens up the possibility of collecting a very large dataset that spans multiple sources and integrates already existing datasets that possess ground-truth labels for related tasks such as semantic segmentation.

We examined, in the context of training data generation, the shortcomings of image translation using cycle consistency GANs and investigated how these could be improved. During training, the two generators in a cycle consistency architecture learn to encode and decode the details of the inputs from the source domain in a way that cannot be detected by the discriminator of the target domain. This reduces the efficacy of the cycle consistency component and limits the use of the generated images as training data. This is due to the presence of encoded source domain data, which can then be leveraged by downstream models to lower their loss during training instead of actually learning target domain features. Our experiments confirmed that the source domain information is encoded as a high frequency component and successfully demonstrated that using a differentiable denoiser can efficiently strip this steganographic information. This resulted in significant increases in the quality of the training data, while also benefiting from the increased stability of the simpler cycle-consistency model over more complex architectures that attempt to explicitly disentangle style and content [25, 26]. While night-time domain

adaptation is an extreme example where performance is hindered by the lack of features in under-exposed areas, there are many other domain shift problems where features are present but very different - simple models such as cycle consistency GANs have proven to be effective in these cases, and we believe our approach is a straight-forward addition that can significantly boost their performance.

Scene synthesis from semantic segmentation maps, either partial as in our work or full as in [18] is a very interesting proposition for training data generation. The input semantic maps are relatively easy to compose or manipulate, and can also be obtained from 3D engines or simulators. Multi-modal generation is useful for domain randomization and model robustness, while instance attribute manipulation (pose, style etc.) can help to model scarce data or scenarios that are hard to capture in real life. In the case of road marker synthesis, we demonstrate surprisingly large increases in both precision, recall and mIOU when further fine-tuning the baseline model with as few as 500 synthesized images, as compared to the baseline model trained only on real data. This result signals that scene synthesis could have a very large impact on performance in under-represented domains while simultaneously slashing the effort and cost required to improve model performance.

Finally, a vision-only collision mitigation system for urban driving was developed using the CARLA driving simulator [61]. By working directly in the semantic segmentation space instead of images, the system is designed to be invariant to the appearance of the inputs. We only make the assumption that segmentation models work reasonably well under good conditions and that they can be adapted to work under different conditions using the various domain adaptation, de-noising and data generation methods described in this manuscript. This work served as a demonstration of a domain-agnostic input for an ADAS system that learns to implicitly predict the dynamics of both objects in the scene and of the ego-vehicle, leading to significantly reduced measures of impact severity under the simulated scenarios. The fact that a system with simplified and normalized inputs and a relatively low parameter count can exhibit great performance is an indicator that perhaps modular or waterfall systems of highly specialised models are a

better approach than monolithic ones, especially when they allow synthetically-generated data to be leveraged.

6.4 Overall conclusion

Throughout this work, we successfully demonstrated domain adaptation for localisation across a number of weather and appearance variations using generative adversarial networks (GANs), and subsequently extended this to an unsupervised setting that generalises to more than one task. Similarly, we showed that convolutional neural networks are a great choice for image de-noising – in the context of de-raining and inverse perspective mapping – and presented data and ground-truth collection strategies that make this possible. Finally, we tackled the scarcity of training data via data generation, proposing an improvement to cycle-consistency GANs, introduced a very simple method for collecting images affected by adherent droplets complete with a clear ground-truth, developed scene synthesis and randomization techniques for improved semantic segmentation models, and investigated a collision mitigation system that leverages data created using a simulator.

The techniques described herein have lead to significant gains under difficult conditions such as low illumination, large changes in appearance (domain shift) and noisy data, as well as allowing machine learning models to be trained using data that would normally be very hard, expensive or impossible to obtain. The boundaries between domain adaptation, de-noising and data generation are fluid and the methods described can and should be used in a synergistic way, leveraging weak supervision and self-supervision wherever possible. Additionally, systems designed to be light-weight and modular can be used as front-ends for other models, further extending their usability. By doing so, we hope to bring field robotics one stepping stone closer to wider deployment.

Finally, we believe that the next important step in domain adaptation and data generation will revolve around train-time metrics that evaluate the suitability of generated data for use as training data. Currently, most approaches evaluate data separately, by training a downstream model and verifying its performance, which

introduces large amounts of lag and inefficiencies in the training of generative models and hinders model selection. Being able to easily determine the goodness of fit of synthetic data would liberate generative models as a tool for training data generation.

References

[1] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. "Generative Adversarial Nets". In: *Proceedings of the 27th International Conference on Neural Information Processing Systems (NIPS) - Volume 2*. Montreal, Canada: MIT Press, 2014, pp. 2672–2680.

[2] Chris Linegar, Winston Churchill, and Paul Newman. "Made to measure: Bespoke landmarks for 24-hour, all-weather localisation with a camera". In: *IEEE International Conference on Robotics and Automation (ICRA)*. 2016, pp. 787–794.

[3] Carl Doersch, Saurabh Singh, Abhinav Gupta, Josef Sivic, and Alexei A. Efros. "What Makes Paris Look like Paris?" In: *ACM Transactions on Graphics (SIGGRAPH)* 31.4 (2012), 101:1–101:9.

[4] Niko Suenderhauf, Sareh Shirazi, Adam Jacobson, Feras Dayoub, Edward Pepperell, Ben Upcroft, and Michael Milford. "Place Recognition with ConvNet Landmarks: Viewpoint-Robust, Condition-Robust, Training-Free". In: *Proceedings of Robotics: Science and Systems (RSS)*. Rome, Italy, 2015.

[5] Winston Churchill and Paul Newman. "Experience-based Navigation for Long-term Localisation". In: *The International Journal of Robotics Research (IJRR)* (2013).

[6] Alex Kendall, Matthew Grimes, and Roberto Cipolla. "PoseNet: A Convolutional Network for Real-Time 6-DOF Camera Relocalization". In: *IEEE International Conference on Computer Vision (ICCV)*. 2015, pp. 2938–2946.

[7] Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. "Image Style Transfer Using Convolutional Neural Networks". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016.

[8] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. "Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks". In: *IEEE International Conference on Computer Vision (ICCV)*. 2017, pp. 2242–2251.

[9] Horia Porav, Tom Bruls, and Paul Newman. "Don't Worry About the Weather: Unsupervised Condition-Dependent Domain Adaptation". In: *IEEE Intelligent Transportation Systems Conference (ITSC)*. 2019, pp. 33–40.

[10] Rui Qian, Robby T. Tan, Wenhan Yang, Jiajun Su, and Jiaying Liu. "Attentive Generative Adversarial Network for Raindrop Removal from A Single Image". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018, pp. 2482–2491.

[11] David Eigen, Dilip Krishnan, and Rob Fergus. "Restoring an Image Taken through a Window Covered with Dirt or Rain". In: *IEEE International Conference on Computer Vision (ICCV)*. 2013, pp. 633–640.

[12] Martin Roser and Andreas Geiger. “Video-based raindrop detection for improved image registration”. In: *IEEE 12th International Conference on Computer Vision (ICCV) Workshops*. 2009, pp. 570–577.

[13] Horia Porav, Tom Bruls, and Paul Newman. “I Can See Clearly Now: Image Restoration via De-Raining”. In: *International Conference on Robotics and Automation (ICRA)*. 2019, pp. 7087–7093.

[14] Horia Porav, Valentina-Nicoleta Musat, Tom Bruls, and Paul Newman. “Rainy screens: Collecting rainy datasets, indoors”. In: *CoRR* abs/2003.04742 (2020). arXiv: 2003.04742. URL: https://arxiv.org/abs/2003.04742.

[15] Horia Porav, Will Maddern, and Paul Newman. “Adversarial Training for Adverse Conditions: Robust Metric Localisation Using Appearance Transfer”. In: *IEEE International Conference on Robotics and Automation (ICRA)*. 2018, pp. 1011–1018.

[16] Asha Anoosheh, Torsten Sattler, Radu Timofte, Marc Pollefeys, and Luc Van Gool. “Night-to-Day Image Translation for Retrieval-based Localization”. In: *International Conference on Robotics and Automation (ICRA)*. 2019, pp. 5958–5964.

[17] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. “Image-to-Image Translation with Conditional Adversarial Networks”. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 5967–5976.

[18] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. “High-Resolution Image Synthesis and Semantic Manipulation with Conditional GANs”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018.

[19] Tom Bruls, Horia Porav, Lars Kunze, and Paul Newman. “Generating All the Roads to Rome: Road Layout Randomization for Improved Road Marking Segmentation”. In: *IEEE Intelligent Transportation Systems Conference (ITSC)*. 2019, pp. 831–838.

[20] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. “SURF: Speeded Up Robust Features”. In: *European Conference on Computer Vision (ECCV)*. Ed. by Aleš Leonardis, Horst Bischof, and Axel Pinz. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 404–417.

[21] Will Maddern, Geoff Pascoe, Chris Linegar, and Paul Newman. “1 Year, 1000km: The Oxford RobotCar Dataset”. In: *The International Journal of Robotics Research (IJRR)* 36.1 (2017), pp. 3–15.

[22] Fisher Yu, Haofeng Chen, Xin Wang, Wenqi Xian, Yingying Chen, Fangchen Liu, Vashisht Madhavan, and Trevor Darrell. “BDD100K: A Diverse Driving Dataset for Heterogeneous Multitask Learning”. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020, pp. 2633–2642.

[23] Gabriel J Brostow, Julien Fauqueur, and Roberto Cipolla. “Semantic object classes in video: A high-definition ground truth database”. In: *Pattern Recognition Letters* 30.2 (2009), pp. 88–97.

[24] 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: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 3213–3223.

[25] Xun Huang, Ming-Yu Liu, Serge Belongie, and Jan Kautz. "Multimodal Unsupervised Image-to-image Translation". In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018.

[26] Hsin-Ying Lee, Hung-Yu Tseng, Jia-Bin Huang, Maneesh Singh, and Ming-Hsuan Yang. "Diverse Image-to-Image Translation via Disentangled Representations". In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018.

[27] Tom Bruls, Horia Porav, Lars Kunze, and Paul Newman. "The Right (Angled) Perspective: Improving the Understanding of Road Scenes Using Boosted Inverse Perspective Mapping". In: *IEEE Intelligent Vehicles Symposium (IV)*. 2019, pp. 302–309.

[28] Horia Porav, Valentina Musat, and Paul Newman. "Reducing Steganography In Cycle-consistency GANs". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*. 2019.

[29] Horia Porav and Paul Newman. "Imminent Collision Mitigation with Reinforcement Learning and Vision". In: *IEEE Intelligent Transportation Systems Conference (ITSC)*. 2018, pp. 958–964.

[30] Kwang Moo Yi, Eduard Trulls, Vincent Lepetit, and Pascal Fua. "LIFT: Learned Invariant Feature Transform". In: *European Conference on Computer Vision (ECCV)*. Ed. by Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling. Cham: Springer International Publishing, 2016, pp. 467–483.

[31] Casey Chu, Andrey Zhmoginov, and Mark Sandler. "CycleGAN, a Master of Steganography". In: *CoRR* abs/1712.02950 (2017). arXiv: `1712.02950`. URL: `http://arxiv.org/abs/1712.02950`.

[32] David Bau, Hendrik Strobelt, William Peebles, Jonas Wulff, Bolei Zhou, Jun-Yan Zhu, and Antonio Torralba. "Semantic Photo Manipulation with a Generative Image Prior". In: *ACM Trans. Graph.* 38.4 (July 2019).

[33] Jinjin Gu, Yujun Shen, and Bolei Zhou. "Image Processing Using Multi-Code GAN Prior". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020, pp. 3009–3018.

[34] Mark Cummins and Paul Newman. "FAB-MAP: Appearance-Based Place Recognition and Mapping Using a Learned Visual Vocabulary Model". In: *Proceedings of the 27th International Conference on International Conference on Machine Learning (ICML)*. Haifa, Israel, 2010, 3–10.

[35] Michael J. Milford and Gordon. F. Wyeth. "SeqSLAM: Visual route-based navigation for sunny summer days and stormy winter nights". In: *IEEE International Conference on Robotics and Automation (ICRA)*. 2012, pp. 1643–1649.

[36] David G. Lowe. "Distinctive Image Features from Scale-Invariant Keypoints". In: *International Journal of Computer Vision (IJCV)* 60 (2004), pp. 91–110.

[37] Michael Calonder, Vincent Lepetit, Christoph Strecha, and Pascal Fua. "BRIEF: Binary Robust Independent Elementary Features". In: *Proceedings of the 11th European Conference on Computer Vision (ECCV): Part IV*. Berlin, Heidelberg: Springer-Verlag, 2010, 778–792.

[38] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. "ORB: An efficient alternative to SIFT or SURF". In: *International Conference on Computer Vision (ICCV)*. 2011, pp. 2564–2571.

[39] Edward Rosten and Tom Drummond. "Machine Learning for High-Speed Corner Detection". In: *Proceedings of the 9th European Conference on Computer Vision (ECCV) - Volume Part I*. Graz, Austria: Springer-Verlag, 2006, 430–443.

[40] Martin A. Fischler and Robert C. Bolles. "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography". In: *Commun. ACM* 24.6 (June 1981), 381–395.

[41] Jakob Engel, Thomas Schöps, and Daniel Cremers. "LSD-SLAM: Large-Scale Direct Monocular SLAM". In: *European Conference on Computer Vision (ECCV)*. 2014.

[42] Will Maddern, Geoffrey Pascoe, and Paul Newman. "Leveraging experience for large-scale LIDAR localisation in changing cities". In: *IEEE International Conference on Robotics and Automation (ICRA)*. 2015, pp. 1684–1691.

[43] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. 2nd ed. USA: Cambridge University Press, 2003.

[44] Raúl Mur-Artal, J. M. M. Montiel, and Juan D. Tardós. "ORB-SLAM: A Versatile and Accurate Monocular SLAM System". In: *IEEE Transactions on Robotics* 31.5 (2015), pp. 1147–1163.

[45] Sarah H. Cen and Paul Newman. "Precise Ego-Motion Estimation with Millimeter-Wave Radar Under Diverse and Challenging Conditions". In: *IEEE International Conference on Robotics and Automation (ICRA)*. 2018, pp. 6045–6052.

[46] Aston Zhang, Zachary C. Lipton, Mu Li, and Alexander J. Smola. *Dive into Deep Learning*. Available at `http://https://d2l.ai/chapter_convolutional-neural-networks/conv-layer.html` (accessed 26.02.2020). 2020.

[47] Vinod Nair and Geoffrey E. Hinton. "Rectified Linear Units Improve Restricted Boltzmann Machines". In: *Proceedings of the 27th International Conference on International Conference on Machine Learning (ICML)*. Madison, WI, USA, 2010, 807–814.

[48] Andrew L. Maas, Awni Y. Hannun, and Andrew Y. Ng. "Rectifier nonlinearities improve neural network acoustic models". In: *International Conference on Machine Learning (ICML) Workshop on Deep Learning for Audio, Speech and Language Processing*. 2013.

[49] Mehdi Mirza and Simon Osindero. "Conditional Generative Adversarial Nets". In: *CoRR* abs/1411.1784 (2014). arXiv: `1411.1784`. URL: `https://arxiv.org/abs/1411.1784`.

[50] Anders Boesen Lindbo Larsen, Søren Kaae Sønderby, Hugo Larochelle, and Ole Winther. "Autoencoding beyond Pixels Using a Learned Similarity Metric". In: *Proceedings of the 33rd International Conference on Machine Learning (ICML) - Volume 48*. New York, NY, USA, 2016, 1558–1566.

[51] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. "GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium". In: *Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS)*. Long Beach, California, USA: Curran Associates Inc., 2017, 6629–6640.

[52] Leslie N. Smith. "Cyclical Learning Rates for Training Neural Networks". In: *IEEE Winter Conference on Applications of Computer Vision (WACV)*. 2017, pp. 464–472.

[53] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. "Progressive Growing of GANs for Improved Quality, Stability, and Variation". In: *International Conference on Learning Representations (ICLR)*. 2018.

[54] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q. Weinberger. "Densely Connected Convolutional Networks". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 2261–2269.

[55] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. "U-Net: Convolutional Networks for Biomedical Image Segmentation". In: *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*. Springer International Publishing, 2015, pp. 234–241.

[56] Ashish Shrivastava, Tomas Pfister, Oncel Tuzel, Joshua Susskind, Wenda Wang, and Russell Webb. "Learning from Simulated and Unsupervised Images through Adversarial Training". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 2242–2251.

[57] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. "Improved Techniques for Training GANs". In: *Proceedings of the 30th International Conference on Neural Information Processing Systems (NIPS)*. Barcelona, Spain: Curran Associates Inc., 2016, 2234–2242.

[58] Chuan Li and Michael Wand. "Precomputed Real-Time Texture Synthesis with Markovian Generative Adversarial Networks". In: *European Conference on Computer Vision (ECCV)*. Vol. 9907. Springer, 2016, pp. 702–716.

[59] Fabian Mentzer, Eirikur Agustsson, Michael Tschannen, Radu Timofte, and Luc Van Gool. "Conditional Probability Models for Deep Image Compression". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018.

[60] Eirikur Agustsson, Michael Tschannen, Fabian Mentzer, Radu Timofte, and Luc Van Gool. "Generative Adversarial Networks for Extreme Learned Image Compression". In: *IEEE International Conference on Computer Vision (ICCV)*. 2019, pp. 221–231.

[61] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. "CARLA: An Open Urban Driving Simulator". In: *Proceedings of the 1st Annual Conference on Robot Learning (CoRL)*. 2017, pp. 1–16.

[62] Abhinav Valada, Johan Vertens, Ankit Dhall, and Wolfram Burgard. "AdapNet: Adaptive semantic segmentation in adverse environmental conditions". In: *IEEE International Conference on Robotics and Automation (ICRA)*. 2017, pp. 4644–4651.

[63] Johannes L. Schönberger, Marc Pollefeys, Andreas Geiger, and Torsten Sattler. "Semantic Visual Localization". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018, pp. 6896–6906.

[64] Erik Stenborg, Carl Toft, and Lars Hammarstrand. "Long-Term Visual Localization Using Semantically Segmented Images". In: *IEEE International Conference on Robotics and Automation (ICRA)*. 2018, pp. 6484–6490.

[65] Yuhua Chen, Wen Li, Christos Sakaridis, Dengxin Dai, and Luc Van Gool. "Domain Adaptive Faster R-CNN for Object Detection in the Wild". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018, pp. 3339–3348.

[66] Markus Wulfmeier, Alex Bewley, and Ingmar Posner. "Addressing appearance change in outdoor robotics with adversarial domain adaptation". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2017, pp. 1551–1558.

[67] Martin Roser, Julian Kurz, and Andreas Geiger. "Realistic Modeling of Water Droplets for Monocular Adherent Raindrop Recognition Using Bézier Curves". In: *Asian Conference on Computer Vision (ACCV) Workshops*. Berlin, Heidelberg, 2010, pp. 235–244.

[68] Kshitiz Garg and Shree K. Nayar. "Detection and removal of rain from videos". In: *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*. Vol. 1. 2004, pp. I–I.

[69] Shaodi You, Robby T. Tan, Rei Kawakami, Yasuhiro Mukaigawa, and Katsushi Ikeuchi. "Adherent Raindrop Modeling, Detection and Removal in Video". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* 38.9 (2016), pp. 1721–1733.

[70] Atsushi Yamashita, Yuu Tanaka, and Toru Kaneko. "Removal of adherent waterdrops from images acquired with stereo camera". In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2005, pp. 400–405.

[71] Shaodi You, Robby T. Tan, Rei Kawakami, Yasuhiro Mukaigawa, and Katsushi Ikeuchi. "Waterdrop Stereo". In: *CoRR* abs/1604.00730 (2016). arXiv: 1604.00730. URL: http://arxiv.org/abs/1604.00730.

[72] Masayuki Kuramoto, Atsushi Yamashita, Toru Kaneko, and Kenjiro T. Miura. "Removal of Adherent Waterdrops in Images by Using Multiple Cameras". In: *IAPR Workshop on Machine Vision Applications*. 2002, pp. 80–83.

[73] Atsushi Yamashita, Masayuki Kuramoto, Toru Kaneko, and Kenjiro T. Miura. "A virtual wiper - restoration of deteriorated images by using multiple cameras". In: *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Vol. 4. 2003, 3126–3131 vol.3.

[74] Atsushi Yamashita, Toru Kaneko, and Kenjiro T. Miura. "A virtual wiper-restoration of deteriorated images by using a pan-tilt camera". In: *IEEE International Conference on Robotics and Automation (ICRA)*. Vol. 5. 2004, 4724–4729 Vol.5.

[75] Kshitiz Garg and Shree K. Nayar. "Vision and Rain". In: *International Journal of Computer Vision (IJCV)* 75.1 (2007), pp. 3–27.

[76] Kshitiz Garg and Shree K. Nayar. "Photometric Model of a Rain Drop". In: *Technical Report, Department of Computer Science, Columbia University*. 2004.

[77] Kenneth V. Beard and Catherine Chuang. "A New Model for the Equilibrium Shape of Raindrops". In: *Journal of the Atmospheric Sciences* 44.11 (1987), pp. 1509–1524.

[78] Jin-Hwan Kim, Jae-Young Sim, and Chang-Su Kim. "Video Deraining and Desnowing Using Temporal Correlation and Low-Rank Matrix Completion". In: *IEEE Transactions on Image Processing* 24.9 (2015), pp. 2658–2670.

[79] Jin-Hwan Kim, Jae-Young Sim, and Chang-Su Kim. "Stereo video deraining and desnowing based on spatiotemporal frame warping". In: *IEEE International Conference on Image Processing (ICIP)*. 2014, pp. 5432–5436.

[80] Pierre Duthon, Frédéric Bernardin, Frédéric Chausse, and Michèle Colomb. "Methodology Used to Evaluate Computer Vision Algorithms in Adverse Weather Conditions". In: *Transportation Research Procedia* 14 (2016), pp. 2178–2187.

[81] Jie Chen and Lap-Pui Chau. "A Rain Pixel Recovery Algorithm for Videos With Highly Dynamic Scenes". In: *IEEE Transactions on Image Processing* 23.3 (2014), pp. 1097–1104.

[82] Wenqi Ren, Si Liu, Hua Zhang, Jinshan Pan, Xiaochun Cao, and Ming-Hsuan Yang. "Single image dehazing via multi-scale convolutional neural networks". In: *European Conference on Computer Vision (ECCV)*. Springer. 2016, pp. 154–169.

[83] Xueyang Fu, Jiabin Huang, Xinghao Ding, Yinghao Liao, and John Paisley. "Clearing the Skies: A Deep Network Architecture for Single-Image Rain Removal". In: *IEEE Transactions on Image Processing* 26.6 (2017), pp. 2944–2956.

[84] James F. Blinn. "A Generalization of Algebraic Surface Drawing". In: *ACM Trans. Graph.* 1.3 (July 1982), pp. 235–256.

[85] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. "Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation". In: *European Conference on Computer Vision (ECCV)*. 2018.

[86] German Ros, Laura Sellart, Joanna Materzynska, David Vazquez, and Antonio M. Lopez. "The SYNTHIA Dataset: A Large Collection of Synthetic Images for Semantic Segmentation of Urban Scenes". In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016.

[87] Tom Bruls, Will Maddern, Akshay A. Morye, and Paul Newman. "Mark Yourself: Road Marking Segmentation via Weakly-Supervised Annotations from Multimodal Data". In: *IEEE International Conference on Robotics and Automation (ICRA)*. 2018, pp. 1863–1870.

[88] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. “Perceptual losses for real-time style transfer and super-resolution”. In: *European Conference on Computer Vision (ECCV)*. Springer. 2016, pp. 694–711.

[89] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. “High-Resolution Image Synthesis and Semantic Manipulation with Conditional GANs”. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018, pp. 8798–8807.

[90] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. “Deep Residual Learning for Image Recognition”. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 770–778.

[91] Chuan Li and Michael Wand. “Precomputed real-time texture synthesis with markovian generative adversarial networks”. In: *European Conference on Computer Vision (ECCV)*. Springer. 2016, pp. 702–716.

[92] Diederik P Kingma and Jimmy Ba. “Adam: A method for stochastic optimization”. In: *3rd International Conference for Learning Representations (ICLR), San Diego*. 2015.

[93] Volodymyr Mnih, Nicolas Heess, Alex Graves, and Koray Kavukcuoglu. “Recurrent Models of Visual Attention”. In: *Proceedings of the 27th International Conference on Neural Information Processing Systems (NIPS) - Volume 2*. Montreal, Canada: MIT Press, 2014, 2204–2212.

[94] Jonathan Cohen, Marc Olano, and Dinesh Manocha. “Appearance-preserving Simplification”. In: *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*. ACM, 1998, pp. 115–122.