This CVPR 2020 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.

Deflating Dataset Bias Using Synthetic Data Augmentation

Nikita Jaipuria, Xianling Zhang, Rohan Bhasin, Mayar Arafa
Punarjay Chakravarty, Shubham Shrivastava, Sagar Manglani, Vidya N. Murali
Ford Greenfield Labs, Palo Alto
{njaipuri, xzhan258, rbhasin, marafa, pchakra5, sshriva5, smanglan, vnariyam}@ford.com

Abstract

Deep Learning has seen an unprecedented increase in vision applications since the publication of large-scale object recognition datasets and introduction of scalable compute hardware. State-of-the-art methods for most vision tasks for Autonomous Vehicles (AVs) rely on supervised learning and often fail to generalize to domain shifts and/or outliers. Dataset diversity is thus key to successful real-world deployment. No matter how big the size of the dataset, capturing long tails of the distribution pertaining to task-specific environmental factors is impractical. The goal of this paper is to investigate the use of targeted synthetic data augmentation - combining the benefits of gaming engine simulations and sim2real style transfer techniques - for filling gaps in real datasets for vision tasks. Empirical studies on three different computer vision tasks of practical use to AVs - parking slot detection, lane detection and monocular depth estimation - consistently show that having synthetic data in the training mix provides a significant boost in cross-dataset generalization performance as compared to training on real data only, for the same size of the training set.

1. Introduction

Data-hungry Deep Neural Networks (DNNs) thrive when trained on large datasets. The release of large-scale datasets (such as ImageNet [4], COCO [24], KITTI [10] and the relatively recent BDD100K [43]) coupled with progress in scalable compute has led to the use of DNNs for a wide variety of vision tasks for autonomous driving. State-of-the-art methods for most of these tasks, such as object detection, semantic segmentation and depth estimation to name a few [13, 29, 11], rely on supervised learning and often fail to generalize to unseen scenarios and/or datasets. Thus, dataset diversity is key to achieving successful deployment of DNNs for real-world vision tasks, especially in safety-critical applications.

Presence of bias in static datasets, such as selection bias, capture bias, label bias and negative set bias [40, 33] is a known problem in computer vision famously shown by the *Name That Dataset* experiment from Torralba *et al.* [40]. However, most of these well studied biases are task-agnostic and too general in nature. For instance, consider the task of lane detection which is one of the most common vision applications in autonomous driving. One way of addressing generic dataset selection biases is to simply augment data from multiple sources like highways, cities etc. But no matter how big the size of the dataset, it is extremely difficult to capture long tails of the distribution, and on the contrary, as shown in [40, 22], mixing different datasets often ends up hurting the final performance! This begs the question if it is ever possible to completely avoid such biases in realistic settings by means of careful data collection [32].

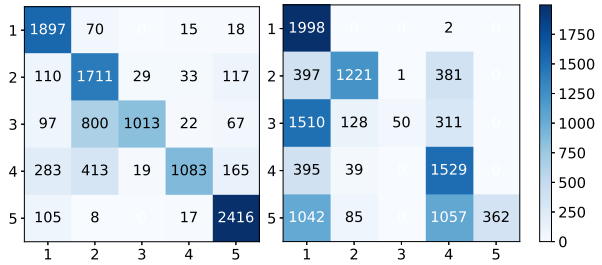


Figure 1: Comparison of confusion matrices from the ResNet-50 [14] based *Name That Dataset* classifiers described in Section 3.1 trained to distinguish between five different lane-detection datasets (left) and between the same five datasets with two of them (3 and 5) augmented with synthetic data (right). Note that synthetic data augmentation helps diffuse the strength of the diagonal indicating deflated dataset bias.

In this work, we focus on bias in the context of the noise distribution pertaining to task-specific environmental factors. We refer to it as *noise factor distribution bias*. For instance, instead of handling diversity by blindly collecting more data in our lane detection example, we chose to augment data with respect to task-specific noise factors, such as diversifying lane marker types, number of lanes in the scene, condition of lane markers, type of lane markers,

weather and lighting effects etc. We show how this could go a long-way in improving algorithm performance. Hoping to obtain such targeted diversity in real data from dashboard cameras in cars is likely futile because of the time it will take and the unavailability of sources.

One approach is to leverage advances in generative modeling to generate synthetic data for augmentation. Generative Adversarial Networks (GANs) [12] have shown immense progress in the past few years in image generation [20, 21]. While they have had huge success in graphics applications [30, 38, 23], synthetic data augmentation for improving performance of recognition models has seen limited success. One reason is the presence of noisy artifacts and semantic inconsistencies in the generated images [17, 31]. Alternatively, gaming-engine simulations can be used to generate semantically consistent data of desired task-specific scenarios, but the perceptual quality is far from realistic. Why not have the best of both worlds? In contrast to performing augmentation with either generated or simulated data, we first simply simulate candidate examples and then translate via unsupervised sim2real generative models [25, 18, 45].

We show that this simple two-stage augmentation when targeted to encourage task-specific noise diversity leads to huge gains in cross-dataset generalization performance. We demonstrate this empirically using three different case studies of computer vision tasks in an AV perception stack: (i) parking slot detection; (ii) lane detection; and (iii) monocular depth estimation. To isolate the effect of simply training on more data, in all of these tasks, synthetic data was used to replace some amount of real data in the training set. Results showed a significant boost in cross-dataset generalization performance, especially in cases where the real dataset was small in size and heavily biased. Moreover, model performance on the original test set was not hurt which further confirms that targeted synthetic data augmentation can go a long way in enriching the real biased dataset.

2. Related Work

Related work on dealing with dataset bias falls under two main categories: (i) Domain Adaptation (DA); and (ii) Transfer Learning. DA is one way of dealing with inherent bias in datasets and the problem of perception algorithms failing to generalize to different datasets. Fernando *et al*. [7] addressed DA by learning a mapping between the source and target datasets in the form of a common subspace between their distributions. One can also learn data specific embeddings subject to minimization of MMD between them [34] in an effort to bring the two distributions closer. A classifier can then act on the learnt embeddings. Optimal transport techniques have also been used to solve DA, with [2] minimizing the Wasserstein distance between the joint embedding and classifier label distributions of the two datasets. Wang *et al*. [41] provide a good taxonomy of DA techniques, including the more recent adversarial techniques based on GANs. Instead of relying on a hand-engineered loss function to bring the source and target data distributions close, these techniques use an adversarially trained discriminator network that attempts to differentiate between data from the two distributions. This discrimination can happen in: (i) the pixel space - where data from one domain is translated into the other using style transfer before being passed to the discriminator [26, 36]; (ii) latent space - where a discriminator learns to differentiate between the learned embeddings from the two domains [37] and; (ii) both the pixel and embedding space [15]. In cases where one has access to unpaired and unannotated data only from the two domains, one can use cycle consistency losses [25, 42, 45] for learning a common embedding between the two spaces. Often, we are concerned with DA for a particular task - for example image segmentation or depth estimation. Recent work has shown that using losses from an auxiliary task like image segmentation can help regularize the feature embeddings [15, 37]. These methods are most relevant to our work and future work will investigate how they fare against our approach of targeted synthetic data augmentation.

Transfer Learning is another way of dealing with dataset bias [39]. In contrast to such approaches, our method assumes no training data is available from the target domain (both for the task network and sim2real models), and that the target task is the same as the source task. Recent work [1, 19] has also focused on using synthetic data to augment real datasets for AV perception tasks. Meta-sim [19] parameterizes scene-grammar to generate a synthetic data distribution that is similar to real data and is optimized for a down-stream task and Alhaija *et al*. [1] augment real scene backgrounds with synthetically inserted objects for improved instance segmentation and object detection performance on real datasets. Our method, in contrast, investigates a general purpose, task agnostic approach to enriching real-world datasets using synthetic data.

3. Deflating Dataset Bias

The main objective of this paper is to test the hypothesis that targeted synthetic data augmentation can help deflate inherent bias in large-scale image datasets. For brevity, we will refer to this hypothesis as `H`. One way of testing `H` is to compare cross-dataset generalization performance of models trained on the original dataset (real) with models trained on augmented datasets (real+synthetic). In this paper, three supervised learning-based computer vision tasks: (i) parking slot detection; (ii) traffic lane detection; and (iii) monocular depth estimation are used as test-beds for the motivating hypothesis `H`, using the following methodology:

1. Simulate images and corresponding annotation using gaming engines for a diverse set of task-specific noise factors.
2. Use unsupervised generative modeling based sim2real methods such as [25, 18, 45] to translate the simulated images into photorealistic ones, that look like they are from the training domain.
3. Train task networks with different ratios of real and simulated data (from Step 1) or real and sim2real data (from Step 2). The size of the training set is kept constant across all experiments to isolate the improvement one can obtain by simply training on more data from the improvement due to deflated dataset bias. Also, the ratio of synthetic data in the training set was increased from 0% to 100% in continuous intervals of 10%.
4. Evaluate and compare cross-dataset generalization performance of all models from Step 3.

Sections 4, 5 and 6 describe the task-specific datasets, experiments and results.

3.1. Revisiting "Name That Dataset"

Torralba *et al*. [40] investigated the then state of object recognition datasets using the *Name That Dataset* experiment in which a 12-way linear SVM classifier was trained to distinguish between 12 datasets. The results showed strong *signatures* for each dataset - indicating inherent *bias* - despite the best efforts of their creators. We repeat the *Name That Dataset* experiment in the era of deep learning with a ResNet-50 [14] (pre-trained on ImageNet) trained to distinguish between five different lane-detection datasets - ApolloScape [16], BDD100K [43], CULane [29], Mapillary [28] and TuSimple[1]. 6000 images were randomly selected from each dataset and divided into training, validation and test sets. In a subsequent experiment, we replace 50% of the real data in two datasets - CULane and TuSimple - with sim2real translated images from VAE-GAN models based off of [25, 18] and trained on unpaired simulated and real CULane and simulated and real TuSimple images respectively. We chose to apply data augmentation to only these two datasets as they are also used for the lane detection experiments in Section 5 with readily available sim2real data on hand. Fig. 1 compares the confusion matrices of the two classifiers, with and without synthetic data augmentation. Here, the labels 1, 2, 3, 4 and 5 denote the ApolloScape, BDD100K, CULane, Mapillary and TuSimple datasets respectively. Consistent with the motivating hypothesis H, synthetic data augmentation diffuses the strength of the diagonal indicating deflated dataset bias.

[1] https://github.com/TuSimple/tusimple-benchmark/tree/master/doc/lane_detection

4. Case Study: Parking Slot Detection

The objective of this task is to detect empty parking slots in images taken from side vehicle cameras (see Fig. 2).

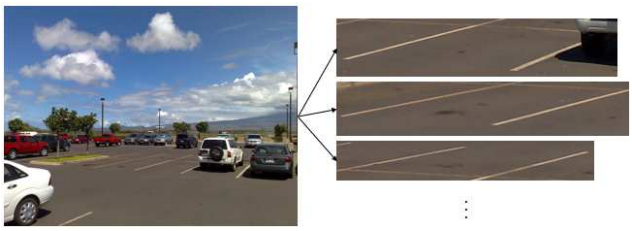


Figure 2: Illustrative example of empty parking slots detected (right) in a parking lot image (left).

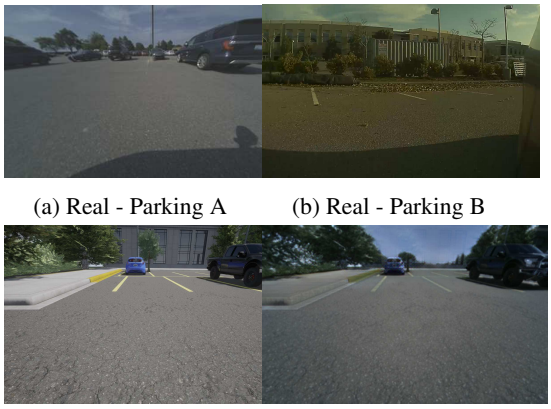
4.1. Dataset Description

Real Data: An internal parking dataset of bright daytime scenarios from two different parking lots (in Dearborn and Palo Alto) is used as the source of real data for this task. The Dearborn dataset has a total of 5907 images, for brevity, we will refer to this dataset as *Parking A*. The Palo Alto dataset has 602 images. We will refer to this dataset as *Parking B*. Fig. 3a and Fig. 3b show example images from the Parking A and Parking B datasets respectively to further motivate the large domain gap between them.

Synthetic Data: Simulated data for this task is generated using an Unreal Engine[2]-based simulation pipeline for a diverse set of noise factors such as different times of the day, cloud density, shadow intensity/cast location, ground textures, parking line damage levels and parking density. The variety of shadow intensities and locations, along with parking line damage and car density are in stark contrast to the homogeneity of the parking A dataset. Fig. 3c shows an example simulated image, visualizing the large domain gap between the simulated and real data from parking A. A sim2real VAE-GAN model (based on [25, 18]) trained on unpaired simulated images and real images from the Parking A dataset is used to translate the generated simulated data to look photorealistic. Fig. 3d shows the sim2real translated output for Fig. 3c. Note the realistic ground textures and lighting effects in Fig. 3d in contrast to Fig. 3c.

For the slot detection experiments in this paper, MobileNetV2 SSD [35, 27], pre-trained on COCO [24], was trained and tested on 300 $\times$ 300 parking lot images to detect open parking slots, as shown in Fig. 2. The Parking A dataset was split into a train and test set with 3545 images and 2362 images respectively. Given the small size of the Parking B dataset (602 images), it was used for testing only. Intersection over Union (IoU) of detected slots with ground truth empty slots is used as the metric for quantitative evaluation. Post training, model checkpoint with the

[2] https://www.unrealengine.com/en-US/



(a) Real - Parking A (b) Real - Parking B

(c) Simulated (d) Sim2Real Translated

Figure 3: Example images from the real and synthetic data used for the slot detection experiments.

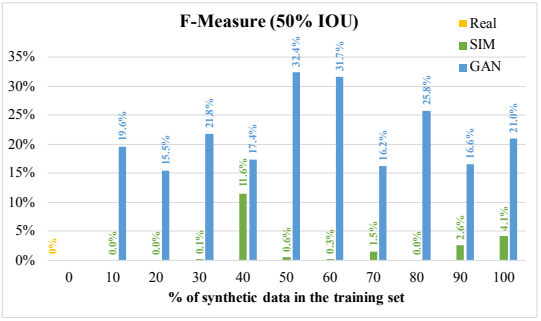


Figure 4: Plot of F-measure for slot detection models trained on a mix of real (Parking A) and synthetic images (either from simulation or from sim2real GAN) and tested on real Parking B images. As you move from left to right, the ratio of synthetic data in the training set increases.

best F-measure for 50% IoU on the Parking A test set is used for inference. The rest of this section describes the experiments performed to test our motivating hypothesis H.

4.2. Results

Fig. 4 shows the results of all slot detection models on the Parking B test set. Notice models trained on a mix of real and synthetic data (green and blue) significantly outperform the model trained on real data only (yellow). Moreover, across all ratios, models trained on a mix of real Parking A images and sim2real translated images (blue) do better than the models trained on a mix of real Parking A images and corresponding simulated images from Unreal Engine (green). Overall best performance (F-measure of 32.4%) is achieved by the model trained on a mix of real and GAN data in a 50:50 ratio. Table 1 summarizes the results from the plots in Fig. 4. For the synthetic data augmentation experiments, results are shown for the best model in terms of F-measure on cross-dataset testing. Additional insights into the number of true positives and false positives for cross-dataset testing with the models from Table 1 are provided in the Supplementary Material.

Table 1: Summary of results in Fig. 4. Here, A and B denote the Parking A and Parking B datasets. S denotes simulated images and G denotes the sim2real translated equivalent of S. For synthetic data augmentation rows, results are shown for the best model in terms of F-measure on cross-dataset testing in green for A + S and in blue for A + G.

Train	Test	Precision (↑)	Recall (↑)	F-Measure (↑)
A	A	95.1%	87.9%	91.4%
A + S (40%)	A	93.8%	87.7%	90.7%
A + G (50%)	A	94.2%	86.5%	90.2%
A	B	0%	0%	0%
A + S (40%)	B	**71.8%**	**6.3%**	**11.6%**
A + G (50%)	B	**67.0%**	**21.4%**	**32.4%**

4.3. Experiment Details

As shown in Table 1, MobileNetV2 SSD trained on Parking A results in a F-Measure of 91.4% on the Parking A test set (1st row). However, the same model when tested on the Parking B dataset results in a F-measure of 0% (4th row). It is a well known fact that supervised learning-based methods do not generalize across different domains. In this particular case the generalization performance is much worse than one might expect because of two main reasons: (i) the small size (relative to large-scale image datasets such as ImageNet [4] and COCO [24]) and low diversity (all daytime images from the same parking lot) of the Parking A dataset; (ii) the large domain gap between the two datasets. Increasing dropout regularization did not help improve generalization performance either - F-Measure remained constant at 0% for varying levels of dropout. The only improvement observed was in the number of false positives (more details are provided in Supplementary Material).

Thus, these results are consistent with the motivating hypothesis H. Additionally, as shown in the 2nd and 3rd rows of Table 1, synthetic data augmentation did not adversely affect the results on the Parking A test set which further strengthens the case for the use of synthetic data and especially GAN-translated data to enrich real-world datasets for supervised learning tasks.

5. Case Study: Traffic Lane Detection

The objective of this task is to detect lane boundaries in images taken from a front vehicle camera (see Fig. 5). Pan *et al*. [29] achieved state-of-the-art performance on this task with Spatial Convolutional Neural Networks (SCNNs). Their formulation is used as-is for all the lane detection experiments in this paper.

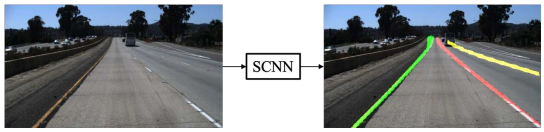


Figure 5: Lane detection schematic.

5.1. Dataset Description

Real Data: Following Pan *et al*. in [29], the *CULane* and *TuSimple*[3] datasets are used as real-world data sources. The CULane dataset has 88880 training images, 9675 validation images and 34680 test images - collected across diverse scenarios including urban, rural and highway environments. The TuSimple dataset has 3268, 358, and 2782 images for training, validation and testing respectively. Compared to CULane, TuSimple has highway scenes only.

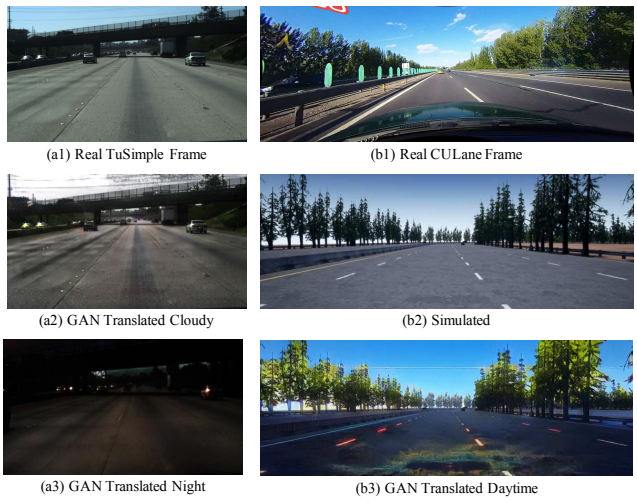


Figure 6: Example real, simulated and GAN-translated images used for lane detection.

Synthetic Data: For augmenting CULane, 88880 daytime highway images were generated using Unreal Engine by varying several noise factors such as the number of lanes, traffic density, sun intensity, location and brightness, road curvature, lane marker wear and tear etc. In testing the original implementation of SCNN, we found that the model performed poorest when lane lines were faint, in shadows or occluded by other vehicles. The change in sun intensity, its location and brightness helped create different shadow effects around the lane lines, giving the network more diverse data to train on. Varying traffic density and road curvature allowed for different occlusions of the lane line markings to produce more diverse data. Example synthetic images generated for this task are shown in Fig. 6. Following the method outlined in Section 3, a sim2real VAE-GAN model (based on [25, 18]) trained on unpaired simulated images and real images from CULane was used to translate the generated simulated data to look photorealistic. Fig. 6 shows the sim2real translated output for the given simulated image. Note the realistic ground textures and lighting effects in the GAN image in contrast to the simulated image.

[3] https://github.com/TuSimple/tusimple-benchmark/tree/master/doc/lane_detection

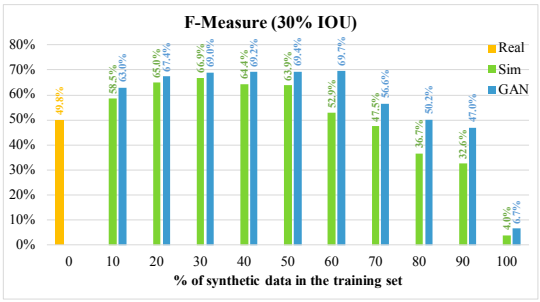


Figure 7: Plot of F-measure for models trained on a mix of CULane and synthetic images (from simulation or from sim2real VAE-GAN) and tested on TuSimple images.

5.2. Experiment Details

For the lane detection experiments in this paper, two types of experiments were performed:
Experiment I: Following Section 3, SCNN [29] is trained on a mix of CULane and synthetic images and tested on TuSimple. For results from SCNN trained on a mix of TuSimple and synthetic images and tested on CULane, please refer Supplementary Material. Models are trained on 800×288 images. For cross-testing, TuSimple images are padded (along width) to match the training resolution of 800×288 while simultaneously maintaining the original aspect ratio. IoU of detected lane lines with ground truth lane lines is used as the metric for quantitative evaluation.
Experiment II: In addition to the experiments described in Section 3, given that the TuSimple dataset has only daytime images while the CULane dataset has a diverse set of weather and lighting conditions (refer Section 5.1), we performed an additional set of experiments for this task to further test the motivating hypothesis $\mathtt{H}$ particularly in scenarios where synthetic data augmentation addresses the specific bias of weather and lighting effects. All synthetic data was generated by applying day-to-night and clear-to-cloudy VAE-GAN models (based off of the architecture in Ref. [25] and trained on BDD100K [43]) to TuSimple images. Fig. 6 shows an example GAN night and cloudy image. SCNN was trained on 512×288 images for this set of experiments and tested on downsized and then padded (along height) versions of CULane images that match the training resolution of 512×288 while simultaneously maintaining the original aspect ratio.

Table 2: Summary of results in Fig. 7. Here, A, A_N and B denote the CULane, CULane Night only and TuSimple datasets. S denotes simulated images and G denotes the sim2real translated equivalent of S. G_N and G_C denote real TuSimple images translated to nighttime and cloudy respectively. For synthetic data augmentation rows, results are shown for the best model in terms of F-measure on cross-dataset testing in green for A + S and in blue for A + G.

Train	Test	Precision (↑)	Recall (↑)	F-Measure (↑)
A	A	53.6%	70.6%	60.9%
A + S (30%)	A	53.5%	70.8%	60.9%
A + G (60%)	A	51.6%	68.0%	58.7%
A	B	47.8%	51.9%	49.8%
A + S (30%)	B	**63.6%**	**70.6%**	**66.9%**
A + G (60%)	B	**67.8%**	**71.6%**	**69.7%**
B	B	80.2%	91.7%	85.6%
B + G_N (10%)	B	80.3%	91.9%	85.7%
B + G_C (80%)	B	79.4%	90.6%	84.7%
B	A	2.8%	3.7%	3.2%
B + G_N (10%)	A	5.9%	7.8%	6.7%
B + G_C (80%)	A	**6.6%**	**8.7%**	**7.5%**
B	A_N	0.2%	0.3%	0.2%
B + G_N (10%)	A_N	2.3%	3.1%	2.6%

5.3. Results

Experiment I: Consistent with the cross-testing results in Section 4, as shown in Table 2, SCNN trained on CULane results in a F-Measure of 60.9% on the CULane test set (1st row) versus 49.8% on the TuSimple test set (4th row). This drop in accuracy can again be attributed to the large domain gap between the two datasets (see Fig. 6). Fig. 7 shows that models trained with a mix of real and sim2real translated data (blue) consistently outperform models trained with a mix of real and simulated data (green) in cross-testing. Moreover, as the ratio of synthetic data in the training set increases, the gap between models trained on GAN data and simulated data grows wider. Both these observations together verify the closeness of the GAN data to the real data as compared to just simulated data. More interestingly, for certain ratios of synthetic data, the models trained on a mix of real and synthetic data significantly outperform models trained with 100% real data. Table 2 (top) summarizes the results from the best models in terms of F-Measure - 69.7% for model trained on a 40:60 mix of real and GAN data and 66.9% for model trained on a 70:30 mix of real and sim data versus just 49.8% for model trained on 100% real data (note the size of the training dataset was held constant across all experiments). These results confirm that synthetic data augmentation can help deflate dataset bias and thus improve cross-dataset generalization performance. Again, similar to the observations in Section 4.2, the drop in accuracy on the original test set is minimal.

Experiment II: Consistent with previous results, SCNN trained on TuSimple gives an F-measure of 85.6% on the TuSimple test set versus only 3.2% on the CULane test set (7th row vs. 10th row in Table 2). The drop in accuracy is more prominent in this case as TuSimple is a much simpler dataset as compared to CULane both in terms of quantity and diversity. Table 2 shows that adding nighttime and cloudy data helps improve cross-dataset generalization performance, with models trained on a mix of real and GAN-generated cloudy data faring the best among all (12th row in green). Since CULane had the nighttime images labeled in their test set, we compared the performance of models trained on TuSimple only with models trained on a mix of TuSimple and GAN nighttime images and again, consistent with our motivating hypothesis H, the latter models do better (last row).

6. Case Study: Monocular Depth Estimation

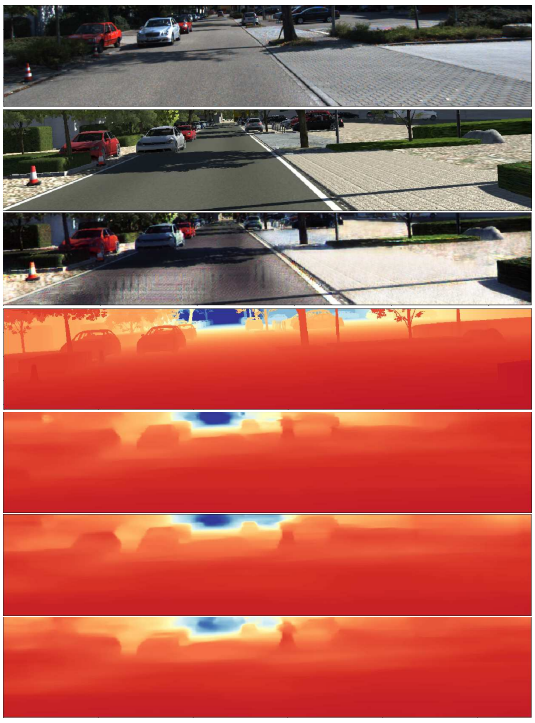


Figure 8: From top to bottom: KITTI RGB, vKITTI RGB, sim2real, ground truth depth, estimated depth A+S (60%), estimated depth A+G (60%) and estimated depth A+G (20%). Networks were trained with unpaired data. Paired images are used for illustrative purposes only.

In this case study, experiments are conducted for the task of estimating the depth in a scene from a single RGB image [6, 9, 11, 44]. We employ an encoder-decoder architecture with skip connections and train the network in a supervised fashion with MSE and edge-aware losses [11] between the ground truth and estimated depth maps.

6.1. Dataset Description

We use KITTI [10] and virtual KITTI (vKITTI) [8] as our real and simulated datasets. The vKITTI dataset is a scene-by-scene recreation of the KITTI tracking dataset, also using the Unreal gaming engine. However, we don't use any paired data for our experiments. We also do not use data from the same sequences as the real data for our simulated data.

Real Data: We use the KITTI odometry sequence 00, with a total of 4,540 images as our real training set - **A**. The KITTI Odometry sequences 02 and 05, with a cumulative 500 images, are used as the real test set - **B**. Ground truth depth is generated by using the OpenCV implementation of the stereo algorithm SGBM with WLS filtering on the left and right images. Note that since we did not make use of paired images between the simulated and real datasets, we could not use simulated depth as ground truth. Moreover, while the simulated recreation in vKITTI approaches that of real KITTI, the simulacrum is not exact, and this would have resulted in systematic biases in the learning of depth. This can be seen in rows 1 and 2 (KITTI and vKITTI) of Figure 8, where the virtual clone of the tree trunk on the right sidewalk is subtly different and slightly shifted.

Synthetic Data: We use data from vKITTI scenes 1, 2, 6, 18 and 20, under the Clone, Morning, 15L and 15R subsets, resulting in a total of 2,126 images per subset, and an overall total of 8,504 images. These vKITTI scenes are clones of the KITTI Tracking dataset (Clone), with variation in camera angles (15L/R) and time of the day (Morning). Note that the KITTI Tracking sequences (duplicated in vKITTI) are captured in a different environment compared to the KITTI Odometry dataset, which form part of our Real set. This variation in sequence geographical location, time of the day and camera pan angles represent the noise factors for this task. A set of randomly picked 4,540 images from this total is used as the source of simulated data for training - **S**. We use cycleGAN [45], trained with unpaired images from KITTI and vKITTI to convert the 4,540 sampled images from vKITTI to make them more realistic. This forms our sim2real translated dataset - **G**.

6.2. Experiment Details

As with the other tasks, we train the task network with different percentages of simulated (**A + S**) and sim2real (**A + G**) data, starting from 0% to 100% and test on KITTI sequences that were not seen during training (**B**). We use the Root Mean Squared Error (RMSE) metric to determine the performance of the network trained on a particular sim/real or sim2real/real mix, after limiting maximum depth to 100m. We provide detailed RMSE results in Figure 9. We also tested this task based on accuracy of depth estimation, measured as the ratio of correctly estimated depth pixels to the total number of depth pixels. These results are summarized, along with RMSE in Table 3 and more detailed results for accuracy are provided in the Supplementary Material. RMSE and accuracy are common metrics used in prior work on single image depth [6]. A lower value of RMSE indicates better performance while the same is true for a higher value for accuracy.

6.3. Results

Figure 9 shows RMSE for the different mixes of real (yellow), real + simulated (A+S, Sim, green) and real + sim2real (A+G, GAN, blue) training data. Some important highlights of the same are shown in Table 3.

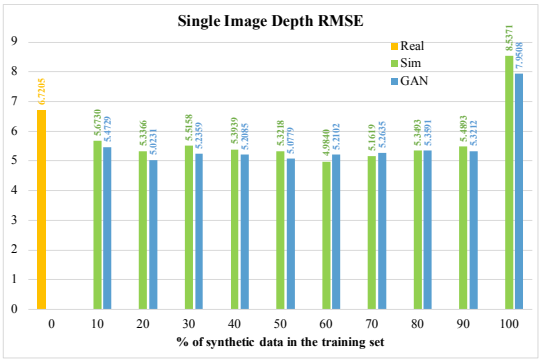


Figure 9: RMSE results for the single image depth task (lower is better).

From the RMSE numbers, one result is clear: having either simulated or sim2real data in the training mix is better than using only real data, for the same amount of total training data. This is shown by the yellow bar (only real data) being higher than the other mixes. Equally, having simulated (sim/sim2real) data alone (the last pair of bars in the RMSE figure) gives the worst results. The trends indicate that mixing sim2real (after converting the simulated data with the sim2real GAN pipeline) with real is better than mixing sim with real, when the percentage of sim/sim2real data is lower or equal to the percentage of real data ($10 - 50\%$ sim/sim2real), in the left half of the bar graphs. In other words, A + G seems to give a slight performance gain over A + S in the $10 - 50\%$ range. From Table 3, we see that the absolute best performer in terms of RMSE is 20% A + G and 60% A + S. Qualitative results are shown for a single image in Figure 8. Visually, the A + G (60%) network (trained with a 40/60 mix of real and sim2real data) seems to perform the best on this image, followed by A + S (60%). The top performer in terms of RMSE, A + G (20%) looks visually slightly worse.

Another important result to be highlighted is the fact that the network trained on just simulation data gains about 7% in terms of RMSE with the sim2real transformation, when tested on real data when using the accuracy numbers. This shows that sim2real from simulation to the source dataset,

Table 3: Summary of results for the single image depth task. Best results for A + S are in green, and best results for A + G are in blue.

Train	Test	RMSE (↓)	Accuracy(↑)
A	B	6.7205	0.9559
A + S (20%)	B	5.3366	0.9705
A + G (20%)	B	**5.0231**	0.9712
A + S (50%)	B	5.3218	0.9702
A + G (50%)	B	5.0779	**0.9721**
A + S (60%)	B	**4.9840**	**0.9723**
A + G (60%)	B	5.2102	0.9682

without any labelling from the source set, already gives a baseline boost. This, when mixed with real labelled data from the source set allows single image depth performance on the target set to rise further, and the perfect mix of real and simulated data lies in the 80/20 to 40/60 range, with sim2real showing minor improvements over just using simulated data in the mix.

We also conducted single image depth experiments using the NuScenes dataset [3] for real data and the CARLA simulation environment [5] for simulation data. These experiments indicated that the estimation of a depth map from a single image is highly dependent on the focal length and other intrinsic camera parameters. We were able to get good results on the NuScenes dataset by using data from CARLA, when the simulated camera on CARLA had been matched with the intrinsics the NuScenes camera. However, any mix of KITTI with NuScenes/CARLA during training completely confounded the algorithm and we do not include these experiments in this paper. We consider camera intrinsics an important consideration when generating simulation and sim2real data and one has to match these with the target dataset. The mixing of data across datasets captured with different focal length cameras requires more sophisticated techniques that are beyond the scope of this paper.

7. Discussion

As motivated in Section 1, dataset bias is a known problem in computer vision. However, most of the well studied sources of bias are task-agnostic. In this work, we focus on bias in the context of the noise distribution pertaining to task-specific environmental factors, referred to as *noise factor distribution bias*, and show that targeted synthetic data augmentation can help deflate this bias. For empirical verification, we use three different computer vision tasks of immense practical use - parking slot detection, lane detection and monocular depth estimation. Synthetic data for these tasks is generated via a simple two step process: (i) simulate images for a diverse set of task-specific noise factors and obtain corresponding ground truth; (ii) perform sim2real translation using GANs to make simulated images look like they are from the real training domain. The rest of this section summarizes the key insights obtained.

Across all three tasks, having synthetic data in the training mix provides a significant boost in cross-dataset generalization performance as compared to training on real data only, for the same size of the training set. Moreover, performance on the source domain test set was not adversely impacted which makes the case for synthetic data augmentation to enrich training datasets for these tasks stronger.

For both the slot detection and lane detection tasks, the best models in terms of F-Measure were those trained on a mix of real and sim2real translated data. For slot detection, the best model with 50% sim2real data in the training mix provided about 30% absolute improvement over the model trained on 100% real data. For lane detection, the best model with 60% sim2real data in the training mix performed about 40% better than the one trained on 100% real data. Another consistent observation across the two tasks is that models with a higher ratio of synthetic data ($> 50\%$) in the training mix do much better when the source of the synthetic data is sim2real data as opposed to simulated data.

In contrast, for the depth estimation task, the best model in terms of both RMSE and accuracy was the one with 60% simulated data (and not sim2real data) in the training mix that achieved a 25% improvement in RMSE over the model trained with 100% real data. We think this is because of two main reasons. First, depth estimation from a sensor (RGB camera) that is missing the 3rd dimension is an inherently hard task with every pixel contributing to the error metric. If we were solving some other problem in which 3D estimation can be parameterized - *e.g.* 3D bounding box detection from 2D images - instead of requiring prediction on a pixel level, we would expect to see a bigger gain with sim and sim2real data added in the training mix. Secondly, slot detection and lane detection are mostly dependent on higher-level features (such as edges) and appearance (such as exposure and lighting conditions). Sim2real is good at doing exactly this - matching higher-level features between the generated and real images and thus these two tasks significantly benefit from sim2real. Depth estimation, however, is dependent more on low-level features. Artifacts introduced by the GAN make it difficult to bridge the low-level feature discrepancies between the sim2real images and corresponding ground truth annotation obtained from simulation. Thus, as expected, for this task, as you go higher in terms of the ratio of synthetic data in the training mix ($> 50\%$), models trained on a mix of real and simulated data do better than those trained on a mix of real and sim2real data. However, the model trained on 100% sim2real data outperforms the one trained on 100% simulated data for this task as well.

Another interesting finding is that across all three tasks, the best models in terms of the chosen metrics were always those with 50%-60% synthetic data in the training mix. Although this makes intuitive sense, it requires more in-depth investigation which will be part of future work.

References

[1] Hassan Abu Alhaija, Siva Karthik Mustikovela, Lars Mescheder, Andreas Geiger, and Carsten Rother. Augmented reality meets computer vision: Efficient data generation for urban driving scenes. *International Journal of Computer Vision*, 126(9):961–972, 2018. 2

[2] Bharath Bhushan Damodaran, Benjamin Kellenberger, Rémi Flamary, Devis Tuia, and Nicolas Courty. Deepjdot: Deep joint distribution optimal transport for unsupervised domain adaptation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 447–463, 2018. 2

[3] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. *arXiv preprint arXiv:1903.11027*, 2019. 8

[4] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 1, 4

[5] Alexey Dosovitskiy, German Ros, Felipe Codevilla, Antonio Lopez, and Vladlen Koltun. Carla: An open urban driving simulator. *arXiv preprint arXiv:1711.03938*, 2017. 8

[6] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. In *Advances in neural information processing systems*, pages 2366–2374, 2014. 6, 7

[7] Basura Fernando, Amaury Habrard, Marc Sebban, and Tinne Tuytelaars. Unsupervised visual domain adaptation using subspace alignment. In *Proceedings of the IEEE international conference on computer vision*, pages 2960–2967, 2013. 2

[8] Adrien Gaidon, Qiao Wang, Yohann Cabon, and Eleonora Vig. Virtual worlds as proxy for multi-object tracking analysis. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4340–4349, 2016. 7

[9] Ravi Garg, Vijay Kumar BG, Gustavo Carneiro, and Ian Reid. Unsupervised cnn for single view depth estimation: Geometry to the rescue. In *European Conference on Computer Vision*, pages 740–756. Springer, 2016. 6

[10] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3354–3361. IEEE, 2012. 1, 7

[11] Clément Godard, Oisin Mac Aodha, and Gabriel J Brostow. Unsupervised monocular depth estimation with left-right consistency. In *CVPR*, volume 2, page 7, 2017. 1, 6

[12] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014. 2

[13] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. 1

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 1, 3

[15] Judy Hoffman, Eric Tzeng, Taesung Park, Jun-Yan Zhu, Phillip Isola, Kate Saenko, Alexei A Efros, and Trevor Darrell. Cycada: Cycle-consistent adversarial domain adaptation. *arXiv preprint arXiv:1711.03213*, 2017. 2

[16] Xinyu Huang, Peng Wang, Xinjing Cheng, Dingfu Zhou, Qichuan Geng, and Ruigang Yang. The apolloscape open dataset for autonomous driving and its application. *arXiv preprint arXiv:1803.06184*, 2018. 3

[17] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017. 2

[18] Nikita Jaipuria, Shubh Gupta, Praveen Narayanan, and Vidya N Murali. On the role of receptive field in unsupervised sim-to-real image translation. *arXiv preprint arXiv:2001.09257*, 2020. 2, 3, 5

[19] Amlan Kar, Aayush Prakash, Ming-Yu Liu, Eric Cameracci, Justin Yuan, Matt Rusiniak, David Acuna, Antonio Torralba, and Sanja Fidler. Meta-sim: Learning to generate synthetic datasets. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4551–4560, 2019. 2

[20] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017. 2

[21] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4401–4410, 2019. 2

[22] Aditya Khosla, Tinghui Zhou, Tomasz Malisiewicz, Alexei A Efros, and Antonio Torralba. Undoing the damage of dataset bias. In *European Conference on Computer Vision*, pages 158–171. Springer, 2012. 1

[23] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4681–4690, 2017. 2

[24] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 1, 3, 4

[25] Ming-Yu Liu, Thomas Breuel, and Jan Kautz. Unsupervised image-to-image translation networks. In *Advances in neural information processing systems*, pages 700–708, 2017. 2, 3, 5

[26] Ming-Yu Liu and Oncel Tuzel. Coupled generative adversarial networks. In *Advances in neural information processing systems*, pages 469–477, 2016. 2

[27] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016. 3

[28] Gerhard Neuhold, Tobias Ollmann, Samuel Rota Bulo, and Peter Kontschieder. The mapillary vistas dataset for semantic understanding of street scenes. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4990–4999, 2017. 3

[29] Xingang Pan, Jianping Shi, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Spatial as deep: Spatial cnn for traffic scene understanding. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018. 1, 3, 4, 5

[30] Taesung Park, Ming-Yu Liu, Ting-Chun Wang, and Jun-Yan Zhu. Semantic image synthesis with spatially-adaptive normalization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2337–2346, 2019. 2

[31] Suman Ravuri and Oriol Vinyals. Classification accuracy score for conditional generative models. In *Advances in Neural Information Processing Systems*, pages 12247–12258, 2019. 2

[32] Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do imagenet classifiers generalize to imagenet? *arXiv preprint arXiv:1902.10811*, 2019. 1

[33] Mengye Ren, Wenyuan Zeng, Bin Yang, and Raquel Urtasun. Learning to reweight examples for robust deep learning. *arXiv preprint arXiv:1803.09050*, 2018. 1

[34] Artem Rozantsev, Mathieu Salzmann, and Pascal Fua. Beyond sharing weights for deep domain adaptation. *IEEE transactions on pattern analysis and machine intelligence*, 41(4):801–814, 2018. 2

[35] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520, 2018. 3

[36] Swami Sankaranarayanan, Yogesh Balaji, Arpit Jain, Ser Nam Lim, and Rama Chellappa. Unsupervised domain adaptation for semantic segmentation with gans. *arXiv preprint arXiv:1711.06969*, 2, 2017. 2

[37] Swami Sankaranarayanan, Yogesh Balaji, Arpit Jain, Ser Nam Lim, and Rama Chellappa. Learning from synthetic data: Addressing domain shift for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3752–3761, 2018. 2

[38] Tamar Rott Shaham, Tali Dekel, and Tomer Michaeli. Singan: Learning a generative model from a single natural image. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4570–4580, 2019. 2

[39] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales. Learning to compare: Relation network for few-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1199–1208, 2018. 2

[40] Antonio Torralba, Alexei A Efros, et al. Unbiased look at dataset bias. In *Computer Vision and Pattern Recognition Conference*, volume 1, page 7. Citeseer, 2011. 1, 3

[41] Mei Wang and Weihong Deng. Deep visual domain adaptation: A survey. *Neurocomputing*, 312:135–153, 2018. 2

[42] Zili Yi, Hao Zhang, Ping Tan, and Minglun Gong. Dualgan: Unsupervised dual learning for image-to-image translation. In *Proceedings of the IEEE international conference on computer vision*, pages 2849–2857, 2017. 2

[43] Fisher Yu, Wenqi Xian, Yingying Chen, Fangchen Liu, Mike Liao, Vashisht Madhavan, and Trevor Darrell. Bdd100k: A diverse driving video database with scalable annotation tooling. *arXiv preprint arXiv:1805.04687*, 2018. 1, 3, 5

[44] Tinghui Zhou, Matthew Brown, Noah Snavely, and David G Lowe. Unsupervised learning of depth and ego-motion from video. In *CVPR*, volume 2, page 7, 2017. 6

[45] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232, 2017. 2, 3, 7