

Synthetic Data for Deep Learning

Sergey I. Nikolenko^{1,2}

¹Synthesis.ai, San Francisco, CA

²Steklov Institute of Mathematics at St. Petersburg, Russia

snikolenko@synthesis.ai

September 26, 2019

Abstract

Synthetic data is an increasingly popular tool for training deep learning models, especially in computer vision but also in other areas. In this work, we attempt to provide a comprehensive survey of the various directions in the development and application of synthetic data. First, we discuss synthetic datasets for basic computer vision problems, both low-level (e.g., optical flow estimation) and high-level (e.g., semantic segmentation), synthetic environments and datasets for outdoor and urban scenes (autonomous driving), indoor scenes (indoor navigation), aerial navigation, simulation environments for robotics, applications of synthetic data outside computer vision (in neural programming, bioinformatics, NLP, and more); we also survey the work on improving synthetic data development and alternative ways to produce it such as GANs. Second, we discuss in detail the synthetic-to-real domain adaptation problem that inevitably arises in applications of synthetic data, including synthetic-to-real refinement with GAN-based models and domain adaptation at the feature/model level without explicit data transformations. Third, we turn to privacy-related applications of synthetic data and review the work on generating synthetic datasets with differential privacy guarantees. We conclude by highlighting the most promising directions for further work in synthetic data studies.

Contents

1	Introduction	3
2	Synthetic data for basic computer vision problems	7
2.1	Low-level computer vision	7
2.2	Basic high-level computer vision	10
2.2.1	Datasets of basic objects	12
2.2.2	Improving high-level computer vision with synthetic data	14
2.3	Synthetic people	18
2.4	Character and text recognition	23
2.5	Visual reasoning	24
3	Synthetic simulated environments	26
3.1	Urban and outdoor environments: learning to drive	26
3.2	Datasets and simulators of indoor scenes	35
3.3	Robotic simulators	38
3.4	Vision-based applications in unmanned aerial vehicles	39
3.5	Computer games as virtual environments	42
4	Synthetic data outside computer vision	44
4.1	Synthetic data for neural programming	44
4.2	Synthetic data in bioinformatics	45
4.3	Synthetic data in natural language processing	47
5	Directions in synthetic data development	50
5.1	Domain randomization	50
5.2	Improving CGI-based generation	51
5.3	Compositing real data to produce synthetic datasets	53
5.4	Synthetic data produced by generative models	54
6	Synthetic-to-real domain adaptation and refinement	56
6.1	Synthetic-to-real refinement	56
6.1.1	Case study: GAN-based refinement for gaze estimation .	57
6.1.2	Refining synthetic data with GANs	59
6.1.3	Making synthetic data from real with GANs	64
6.2	Domain adaptation at the feature/model level	69
6.3	Domain adaptation for control and robotics	74
6.4	Case study: GAN-based domain adaptation for medical imaging	77
7	Privacy guarantees in synthetic data	82
7.1	Differential privacy in deep learning	82
7.2	Differential privacy guarantees for synthetic data generation .	83
7.3	Case study: synthetic data in economics, healthcare, and social sciences	86
8	Promising directions for future work	89
8.1	Procedural generation of synthetic data	89
8.2	From domain randomization to the generation feedback loop .	90
8.3	Improving domain adaptation with domain knowledge	92
8.4	Additional modalities for domain adaptation architectures	92
9	Conclusion	95

1 Introduction

Consider segmentation, a standard computer vision problem. How does one produce a labeled dataset for image segmentation? At some point, all images have to be manually processed: humans have to either draw or at least verify and correct segmentation masks. Making the result pixel-perfect is so laborious that it is commonly considered to be not worth the effort. Figure 1a-c shows samples from the industry standard *Microsoft Common Objects in Context* (MS COCO) dataset [363]; you can immediately see that the segmentation mask is a rather rough polygon and misses many finer features. It did not take us long to find such rough segmentation maps, by the way; these are some of the first images found by the “dog” and “person” queries.

How can one get a higher quality segmentation dataset? To manually correct all of these masks in the MS COCO dataset would probably cost hundreds of thousand dollars. Fortunately, there is a different solution: *synthetic data*. In the context of segmentation, this means that the dataset developers create a 3D environment with modes of the objects they want to recognize and their surroundings and then render the result. Figure 1d-e shows a sample frame from a synthetic dataset called *ProcSy* [317] (we discuss it in more detail in Section 2.2.2): note how the segmentation map is now perfectly correct. While 3D modeling is still mostly manual labor, this is a one-time investment, and as a result one can get a potentially unlimited number of pixel-perfect labeled data: not only RGB images and segmentation maps but also depth images, stereo pairs produced from different viewpoints, point clouds, synthetic video clips, and other modalities.

In general, many problems of modern AI come down to insufficient data: either the available datasets are too small or, also very often, even while capturing unlabeled data is relatively easy the costs of manual labeling are prohibitively high. *Synthetic data* is an important approach to solving the data problem by either producing artificial data from scratch or using advanced data manipulation techniques to produce novel and diverse training examples. The synthetic data approach is most easily exemplified by standard computer vision problems, as we have done above, but it is also relevant in other domains. Naturally, other problems arise, the most important of them being the problem of domain transfer: synthetic images, as you can see from Figure 1, do not look exactly like real images, and one has to make them as photorealistic as possible (a common theme in synthetic data research is whether realism is actually necessary; we will encounter this question several times in this survey) and/or devise techniques that help models transfer from synthetic training sets to real test sets; thus, domain adaptation becomes a major topic in synthetic data research and in this survey as well.

We begin with a few general remarks regarding synthetic data. First, note that synthetic data can be produced and supplied to machine learning models on the fly, during training, with software synthetic data generators, thus alleviating the need to ever store huge datasets; see, e.g., Mason et al. [397] who discuss this “on the fly” generation in detail. Second, while synthetic data is a rising field we know of no satisfactory general overview of the field, and this was our primary motivation for writing this survey. We note surveys that attempt to cover applications of synthetic data [119] and a special issue of the *International Journal of Computer Vision* [185], but hope that the present work paints a more

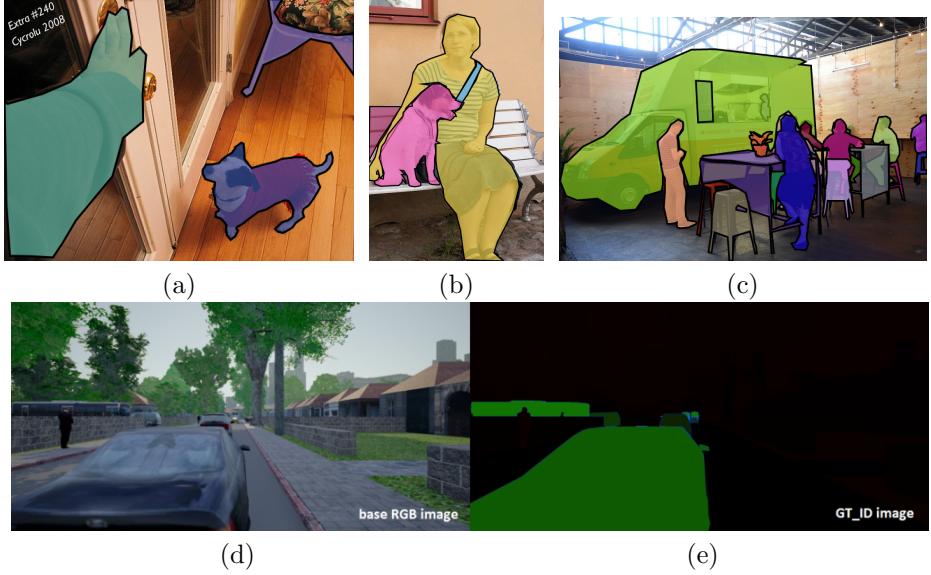


Figure 1: Sample images: (a-c) MS COCO [363] real data samples with ground truth segmentation maps overlaid; (d-e) *ProcSy* [317]: (d) RGB image, (e) ground truth segmentation map.

comprehensive picture.

Third, we distinguish between synthetic data and *data augmentation*; the latter is a set of techniques intended to modify real data rather than create new synthetic data. These days, data augmentation is a crucial part of virtually every computer vision pipeline; we refer to the surveys [544, 630] and especially recommend the *Alumentations* library [78] that has proven invaluable in our practice, but in this survey we concentrate on synthetic data rather than augmentation. Admittedly, the line between them is blurry, and some techniques discussed here could instead be classified as “smart augmentation”.

Fourth, we note a natural application of synthetic data in machine learning: testing hypotheses and comparing methods and algorithms in a controlled synthetic setting. Toy examples and illustrative examples are usually synthetic, with a known data distribution so that machine learning models can be evaluated on how well they learn this distribution. This approach is widely used throughout the field, sometimes for entire meta-analyses [65], and we do not dwell on it here; our subject is synthetic data used to transfer to real data rather than direct comparisons between models on synthetic datasets.

In the survey, we cover three main directions for the use of synthetic data in machine learning; we discuss all three, and below we give references to specific parts of the survey related to these directions.

1. Using synthetically generated datasets to train machine learning models directly. This is the approach often taken in computer vision, and most of this survey is devoted to variations of this approach. In particular, one can:

- train models on synthetic data with the intention to use them on real

data; we discuss through most of Sections 2, 3, and 4;

- train (usually generative) models that change (refine) synthetic data in order to make it more suitable for training; Section 6 is devoted to this kind of models.
2. Using synthetic data to augment existing real datasets so that the resulting hybrid datasets are better for training the models. In this case, the synthetic data is usually employed to cover parts of the data distribution that are not sufficiently represented in the real dataset, with the main purpose being to alleviate dataset bias. The synthetic data can either
- be generated separately with e.g., CGI-based methods for computer vision (see examples in Sections 2 and 3);
 - or be generated from existing real data with the help of generative models (see Section 6.1.3).
3. Using synthetic data to resolve privacy or legal issues that make the use of real data impossible or prohibitively hard. This becomes especially relevant for certain specific fields of application, among which we discuss:
- synthetic data in healthcare, which is not restricted to imaging but also extends to medical records and the like (Sections 7.3, 6.4);
 - synthetic data in finance and social science, where direct applications are hard but privacy-related ones do begin to appear (see Section 7.3);
 - synthetic data with privacy guarantees: many applications are sensitive enough to require a guarantee of privacy, for example from the standpoint of the differential privacy framework, and there has been an important line of work that makes synthetic data generation provide such guarantees (see Section 7).

The survey is organized as follows. Section 2 presents synthetic datasets and results for basic computer vision problems, including low-level problems such as optical flow or stereo disparity estimation (Section 2.1), basic high-level problems such as object detection or segmentation (Section 2.2), human-related synthetic data (Section 2.3), character and text recognition (Section 2.4), and visual reasoning problems (Section 2.5). In Section 3, we proceed to synthetic datasets that are more akin to full-scale simulated environments, covering outdoor and urban environments (Section 3.1), indoor scenes (Section 3.2), synthetic simulators for robotics (Section 3.3) and autonomous flying (Section 3.4), and computer games as simulation environments (Section 3.5).

Section 4 is devoted to other domains of application for synthetic data, including neural programming (Section 4.1), bioinformatics (Section 4.2), and natural language processing (Section 4.3). Section 5 discusses research intended to improve synthetic data generation: domain randomization (Section 5.1), development of methods for CGI-based generation (Section 5.2), synthetic data produced by “cutting and pasting” parts of real data samples (Section 5.3), and direct generation of synthetic data by generative models (Section 5.4).

Section 6 deals with the synthetic-to-real domain adaptation problem that we discussed above; there are many approaches here that can be broadly classified into synthetic-to-real refinement, where domain adaptation models are

used to make synthetic data more realistic (Section 6.1), and domain adaptation at the feature/model level, where the model and/or the training process is adapted rather than the data itself (Section 6.2); we also discuss case studies of domain adaptation for control and robotics (Section 6.3) and medical imaging (Section 6.4).

Section 7 is devoted to the privacy side of synthetic data: Section 7.1 introduces differential privacy, Section 7.2 shows how to generate synthetic data with differential privacy guarantees, and Section 7.3 presents a case study about private synthetic data in finance and related fields.

In an attempt to look forward, we devote Section 8 to directions for further work related to synthetic data that seem most promising: procedural generation of synthetic data (Section 8.1), closing the generation feedback loop (Section 8.2), introducing domain knowledge into domain adaptation (Section 8.3), and improving domain adaptation models with additional modalities that are easy to obtain in synthetic datasets (Section 8.4). Section 9 concludes the paper.

2 Synthetic data for basic computer vision problems

In this section, we present an overview of several directions for using synthetic data in computer vision, surveying both popular synthetic datasets that have been widely used in recent studies and the studies themselves. We organize this section by classifying datasets and models with respect to use cases, from generic object detection and segmentation problems to specific domains such as face recognition. All of these domains benefit highly from pixel-perfect labeling available by default in synthetic data, both in the form of classical computer vision labeling—bounding boxes for objects, segmentation masks—and labeling that would be very hard or impossible to do by hand: depth estimation, stereo image matching, 3D labeling in voxel space, and others.

In Section 2.1, we begin with low-level computer vision problems such as optical flow or stereo disparity estimation. Section 2.2 is devoted to basic high-level computer vision problems, including recognition of basic objects (Section 2.2.1) and improving general problems such as object detection or segmentation with synthetic data (Section 2.2.2). We also discuss several more specialized directions: human-related computer vision problems such as face recognition or crowd counting in Section 2.3, character and text recognition in Section 2.4, and visual reasoning problems in Section 2.5. We also refer to Table 1 for a brief overview of the major datasets considered in this chapter.

2.1 Low-level computer vision

Low-level computer vision problems include, in particular, *optical flow estimation*, i.e., estimating the distribution of apparent velocities of movement along the image, *stereo image matching*, i.e., finding the correspondence between the points of two images of the same scene from different viewpoints, *background subtraction*, and so on. Algorithms for solving these problems can serve as the foundation for computer vision systems; for example, optical flow is important for motion estimation and video compression. Low-level problems can usually be approached with methods that do not require modern large-scale datasets or much learning at all, e.g., classical differential methods for optical flow estimation. However, at the same time, ground truth datasets are very hard to label manually, and hardware sensors that would provide direct measurements of optical flow or stereo image correspondence are difficult to construct (e.g., commodity optical flow sensors simply run the same estimation algorithms).

All of these reasons make low-level computer vision one of the oldest problems where synthetic data was successfully used, originally mostly for evaluation. Works as far back as late 1980s [365] and early 1990s [38] presented and used synthetic datasets to evaluate different optical flow estimation algorithms. In 1999, Freeman et al. [178] presented a synthetically generated world of images, with labeling derived from the corresponding 3D scenes, designed to train and evaluate low-level computer vision algorithms.

A modern dataset for low-level vision is *Middlebury* presented by Baker et al. [34]; in addition to ground truth real life measurements taken with specially constructed computer-controlled lighting, they provide realistic synthetic imagery as part of the dataset and include it in the large-scale evaluation of

Name	Year	Ref	Size / comments
<i>Low-level computer vision</i>			
Tsukuba Stereo	2012	[450]	1800 high-res stereo image pairs
MPI-Sintel	2012	[79]	Optical flow from an animated movie
Middlebury	2014	[525]	33 high-res stereo datasets
Flying Chairs	2015	[152]	22K frame pairs with ground truth flow
Flying Chairs 3D	2015	[400]	22K stereo frames
Monkaa	2015	[400]	8591 stereo frames
Driving	2015	[400]	4392 stereo frames
UnrealStereo	2016	[700]	Data generation software
Underwater	2018	[431]	Underwater synthetic stereo pairs generator
<i>Datasets of basic objects</i>			
YCB	2015	[467]	77 objects in 5 categories
ShapeNet	2015	[93]	>3M models, 3135 categories, rich annotations
ShapeNetCore	2017	[672]	51K manually verified models from 55 categories
UnrealCV	2017	[467]	Plugin for UE4 to generate synthetic data
VANDAL	2017	[87]	4.1M depth images, >9K objects in 319 categories
SceneNet	2015	[232]	Automated indoor synthetic data generator
SceneNet RGB-D	2017	[401]	5M RGB-D images from 16K 3D trajectories
DepthSynth	2017	[455]	Framework for realistic simulation of depth sensors
PartNet	2018	[415]	26671 models, 573535 annotated part instances
Falling Things	2018	[594]	61.5K images of YCB objects in virtual envs
ADORESet	2019	[42]	Hybrid dataset for object recognition testing
<i>Datasets of synthetic people</i>			
ViHASi	2008	[475]	Silhouette-based action recognition
Agoraset	2014	[127]	Crowd scenes generator
LCrowdV	2016	[110]	1M videos, 20M frames with crowds
PHAV	2017	[132]	40K videos for action recognition (35 categories)
SURREAL	2017	[609]	145 subjects, 2.6K sequences, 6.5M frames
SyRI	2018	[32]	Virtual humans in UE4 with realistic lighting
GCC	2019	[625]	15K images with 7.6M subjects

Table 1: An overview of synthetic datasets discussed in Section 2.

optical flow and stereo correspondence estimation algorithms that they undertake. The *Middlebury* dataset played an important role in the development of low-level computer vision algorithms [524, 532], but its main emphasis was still on real imagery, as evidenced by its next version, *Middlebury 2014* [525].

Peris et al. [450] present the *Tsukuba CG Stereo Dataset* with synthetic data and ground truth disparity maps and show improvements in disparity classification quality. Butler et al. [79] presented a synthetic optical flow dataset *MPI-Sintel* derived from the short animated movie Sintel¹ produced as part of the Durian Open Movie Project. The main characteristic feature of *MPI-Sintel* is that it contains the same scenes with different render settings, varying quality and complexity; this approach can provide a deeper understanding of where different optical flow algorithms break down. This is an interesting idea that has not yet found its way into synthetic data for deep learning-based computer vision but might be worthwhile to investigate. An interesting study by Meister and Kondermann [407] shows that while real and (high-quality, produced with ray tracing) synthetic data yield approximately the same results for optical flow

¹<http://www.sintel.org/>

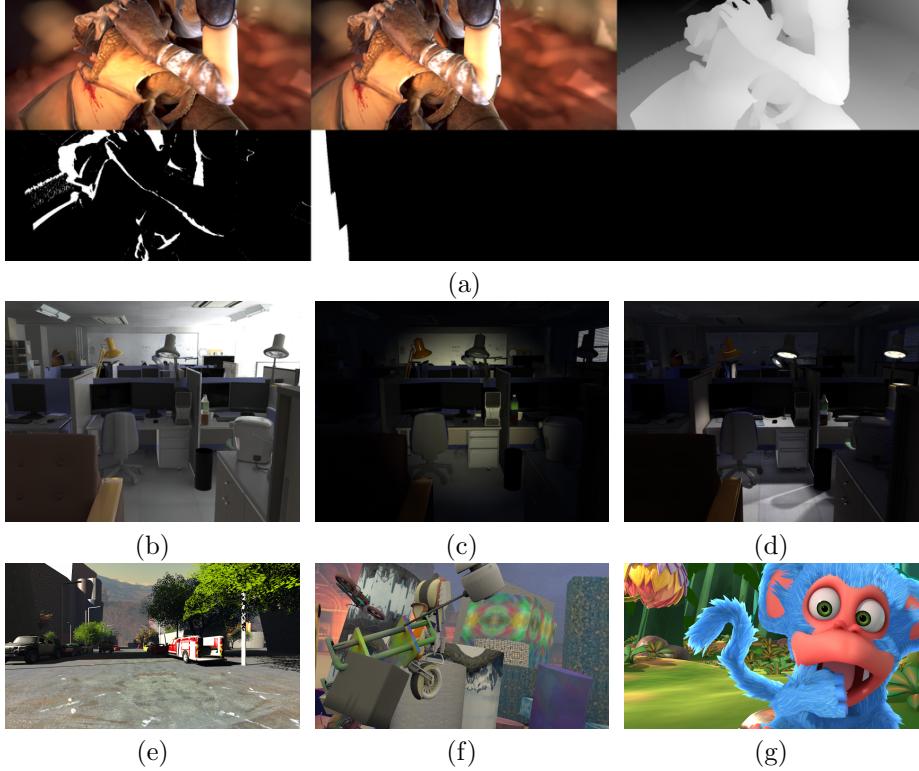


Figure 2: Sample images from synthetic low-level datasets: (a) *MPI-Sintel* [79] (left to right: left view, right view, disparities; bottom row shows occluded and out-of-frame pixels); (b-d) *Tsukuba CG Stereo Dataset* [450] with different illumination conditions: (b) daylight, (c) flashlight, (d) lamps; (e) *Driving* [400]; (f) *FlyingThings3D* [400]; (g) *Monkaa* [400].

detection in terms of mean endpoint error, the spatial distributions of errors are different, so synthetic data in this case may supplement real data in unexpected ways.

As the field moved from classical unsupervised approaches to deep learning, state of the art models began to require large datasets that could not be produced in real life, and after the transition to deep learning synthetic datasets started to dominate. Dosovitsky et al. [152] present a large synthetic dataset called *Flying Chairs* from a public database of 3D chair models, adding them on top of real backgrounds to train a CNN-based optical flow estimation model. Mayer et al. [400] extended this work from optical flow to disparity and scene flow estimation, presenting three synthetic datasets produced in Blender (similar to Sintel):

- *FlyingThings3D* with everyday objects flying along randomized trajectories,
- *Monkaa* from its namesake animated short film with soft nonrigid motion and complex details such as fur, and

- *Driving* with naturalistic dynamic outdoor scenes from the viewpoint of a driving car (for more outdoor datasets see Section 3.1).

The *Flying Chairs* dataset was also later extended with additional modalities to *ChairsSDHom* [271] with optical flow ground truth and *Flying Chairs 2* [272] with occlusion weights and motion boundaries.

The *UnrealStereo* dataset by Zhang et al. [700] is a data generation framework for stereo scene analysis based on the *Unreal Engine 4*, designed to evaluate the robustness of stereo vision algorithms to changes in material and other scene parameters. Many datasets that we describe below for high-level problems, such as SceneNet RGB-D [401] or SYNTHIA [499], also contain labeling for optical flow and have been used to train the corresponding models.

Olson et al. [431] consider an unusual special case for this problem: *underwater* disparity estimation. Their work is also interesting in the way they produce synthetic data: Olson et al. project real underwater images on randomized synthetic surfaces produced in Blender, and then use rendering tools developed to mimic the underwater sensors and characteristic underwater effects such as fast light decay and backscattering. They produce synthetic stereo image pairs and use the dataset to train disparity estimation models, with successful transfer to real images.

In a recent work, Mayer et al. [399] provide an overview of different synthetic datasets for low-level computer vision and compare them from the standpoint of training optical flow models. They come to interesting conclusions:

- first, for low-level vision synthetic data does not have to be realistic, *Flying Chairs* works just fine;
- second, it is best to combine different synthetic datasets and train in a variety of situations and domains; this ties into the domain randomization idea that we discuss in Section 5.1;
- third, while realism itself is not needed, it does help to simulate the flaws of a specific real camera; Mayer et al. show that simulating, e.g., lens distortion and blur or Bayer interpolation artifacts in synthetic data improves the results on a real test set afterwards.

The question of realism remains open for synthetic data, and we will touch upon it many times in this survey. While it does seem plausible that for low-level problems such as optical flow estimation “low-level realism” (simulating camera idiosyncrasies) is much more important than high-level scene realism, the answer may be different for other problems.

2.2 Basic high-level computer vision

Basic high-level computer vision problems, such as object detection or segmentation, fully enjoy the benefits of perfect labeling provided by synthetic data, and there is plenty of effort devoted to making synthetic data work for these problems. Since making synthetic data requires the development of 3D models, datasets usually also feature 3D-related labeling such as the depth map, labeled 3D parts of a shape, volumetric 3D data, and so on. There are many applications of these problems, including object detection for everyday objects and retail items (where a high number of classes and frequently appearing new

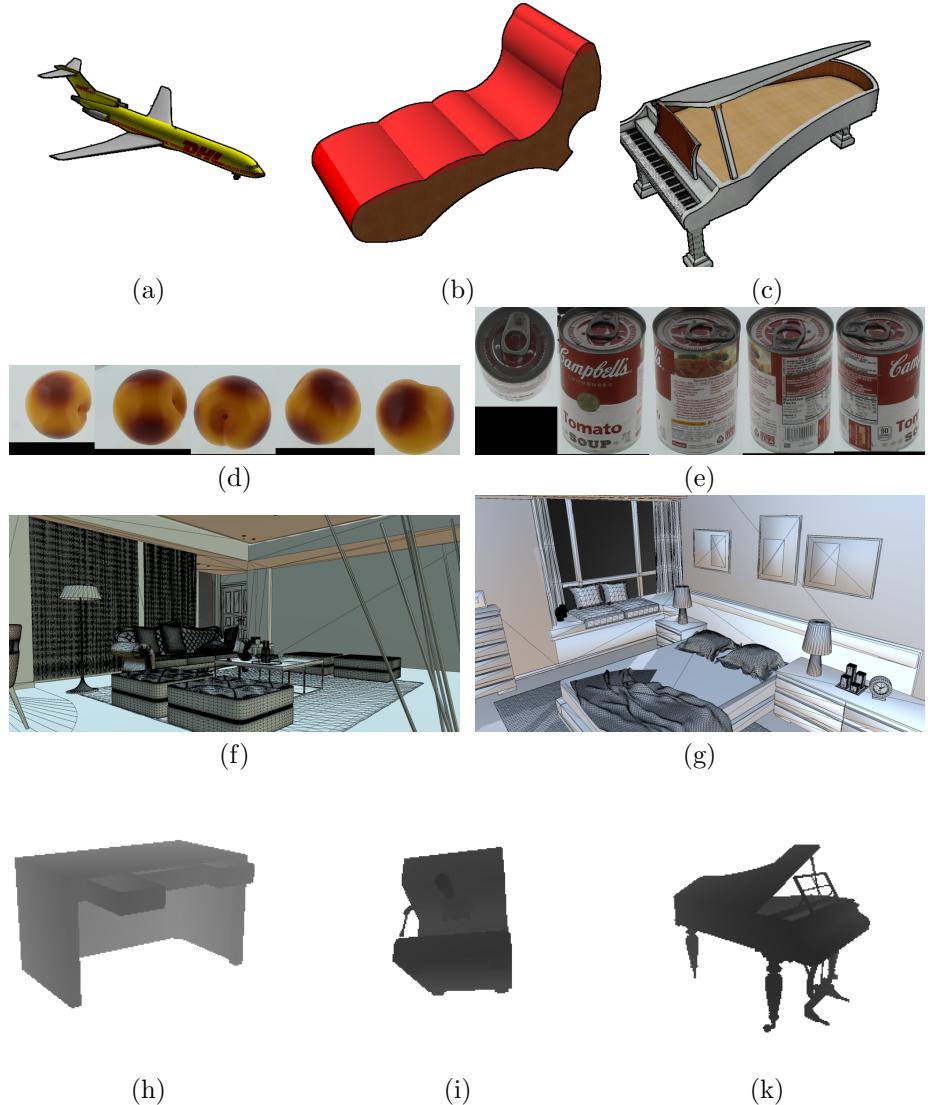


Figure 3: Sample images from synthetic datasets of basic objects: (a-c) shapes from *ShapeNet* [93]: (a) airplane, (b) chair, (c) grand piano; (d-e) from YCB Object and Model set [83]: (d) peach, (e) tomato soup can; (f-g) 3D scenes from *SceneNet* [232, 234]: (f) living room, (g) bedroom; (h-k) depth images from VANDAL [87]: (h) desk, (i) coffee maker, (k) grand piano.

classes make using real data impractical), counting and detection of small objects, basically all applications of semantic and instance segmentation (where manual labeling is especially hard to obtain), and more.

2.2.1 Datasets of basic objects

Many works apply synthetic data to recognizing everyday objects such as retail items, food, or furniture, and most of them draw upon the same database for 3D models. Developed by Chang et al. [93], *ShapeNet*² indexes more than three million models, with 220,000 of them classified into 3,135 categories that match WordNet synsets. Apart from class labels, *ShapeNet* also includes geometric, functional, and physical annotations, including planes of symmetry, part hierarchies, weight and materials, and more. Researchers often use the clean and manually verified *ShapeNetCore* subset that covers 55 common object categories with about 51,000 unique 3D models [672].

ShapeNet has become the basis for further efforts devoted to improving labelings. In particular, region annotation (e.g., breaking an airplane into wings, body, and tail) is a manual process even in a synthetic dataset, while shape segmentation models increasingly rely on synthetic data [670]; this also relates to the 3D mesh segmentation problem [541]. Based on *ShapeNet*, Yi et al. [671] developed a framework for scalable region annotation in 3D models based on active learning, and Chen et al. [104] released a benchmark dataset for 3D mesh segmentation. A recent important effort related to *ShapeNet* is the release of *PartNet* [415], a large-scale dataset of 3D objects annotated with fine-grained, instance-level, and hierarchical 3D part information; it contains 573,585 part instances across 26,671 3D models from 24 object categories. *PartNet* is mostly intended as a benchmark for 3D object and scene understanding, but the corresponding 3D models will no doubt be widely used to generate synthetic data.

One common approach to generating synthetic data is to reuse the work of 3D artists that went into creating the virtual environments of video games. For example, Richter et al. [493,494] captured datasets from the *Grand Theft Auto V* video game (see also Section 3.1). They concentrated on semantic segmentation; note that getting pixel-wise labels for segmentation still required manual labor, but the authors claim that by capturing the communication between the game and the graphics hardware, they have been able to cut the labeling costs (in annotation time) by orders of magnitude. Once the annotator has worked through the first frame, the same combinations of meshes, textures, and shaders reused on subsequent frames can be automatically recognized and labeled, and the annotators are only asked to label new combinations. In essence, the game engine provides perfect superpixels that are persistent across frames.

As *Grand Theft Auto V* and other games became popular for collecting synthetic datasets (see also Section 3.1), more specialized solutions began to appear. One such solution is *UnrealCV* developed by Qiu et al. [466,467], an open-source plugin for the popular game engine *Unreal Engine 4* that provides commands that allow to get and set camera location and field of view, get the set of objects in a scene together with their positions, set lighting parameters, modify properties of the objects such as material, and capture from the engine the image and depth ground truth for the current camera and lighting parameters. This allows to create synthetic image datasets from realistic virtual worlds.

Robotics has motivated the appearance of synthetic datasets with objects that might be subject for manipulation, usually with fairly accurate models of their physical properties. The computer vision objectives in these datasets usually relate to robotic perception and include segmentation, depth estimation,

²<https://www.shapenet.org/>

object pose estimation, and object tracking. In particular, Choi et al. [113] present a dataset of 3D models of household objects for their tracking filter, while Hodan et al. [248] provide a real dataset of textureless objects supplemented with 3D models of these objects that provide the 6D ground truth poses. Lee et al. [347] test existing tracking methods with simulated video sequences with occlusion effects. Papon and Schoeler [437] consider the problem of object pose and depth estimation in indoor scenes. They have developed a synthetic data generator and trained on 7000 randomly generated scenes with \approx 60K instances of 2842 pose-aligned models from the *ModelNet10* dataset [649], showing excellent results in transfer to real test data.

The *Yale-CMU-Berkeley* (YCB) Object and Model set presented by Calli et al. [83] contains a set of 3D models of objects commonly used for robotic grasping together with a database of real RGB-D scans and physical properties of the objects, which makes it possible to use them in simulations. The *Falling Things* (FAT) dataset by NVIDIA researchers Tremblay et al. [594] contains about 61500 images of 21 household objects taken from the YCB dataset and placed into virtual environments under a wide variety of lighting conditions, with 3D poses, pixel-perfect segmentation, depth images, and 2D/3D bounding box coordinates for each object; we show sample images from FAT on Figure 4.

Recent works begin to use synthetic datasets of everyday objects in more complex ways, in particular by placing them in real surroundings. Abu Alhaija et al. [5] and Georgakis et al. [198] propose procedures to augment real backgrounds with synthetic objects (see also Section 5.3 where we discuss placing real objects on real backgrounds). In [5], the backgrounds come from the KITTI dataset of outdoor scenes and the objects are synthetic models of cars, while in [198] the authors place synthetic objects into indoor scenes with an eye towards home service robots. Synthetic objects have been used on real backgrounds many times before, but the main distinguishing feature of [5] and [198] is that they are able to paste synthetic objects on real surfaces in a way consistent with the rest of the background scene. Abu Alhaija et al. developed a pipeline for automated analysis that recognized road surfaces on 360° panoramic images, but at the same time they conclude that the best (w.r.t. to the quality of the resulting segmentation model) way to insert the cars was to do it manually, and almost all their experiments used manual car placement. These experiments showed that state of the art models for instance segmentation and object detection yield better results on real validation tests when trained on scenes augmented with synthetic cars. Georgakis et al. use the algorithm from [581] to extract supporting surfaces from an image and place synthetic objects on such surfaces with proper scale; they show significant improvements by training on hybrid real+synthetic datasets. One of the latest and currently most advanced pipelines in this direction for autonomous driving is AADS [356] that we discuss in Section 3.1.

In general, by now researchers have relatively easy access to large datasets of 3D models of everyday objects to generate synthetic environments (we will see more of this in Sections 3.1 and 3.2), add synthetic objects as distractors to real images, place synthetic objects on real backgrounds in smarter ways, and so on. Although RGBD datasets with real scans are also increasingly available as the corresponding hardware becomes available (see, e.g., the survey [176] and [116]), they cannot compete with synthetic data in terms of the quality of labeling and diversity of environments (see also Section 5.1). In the next section,

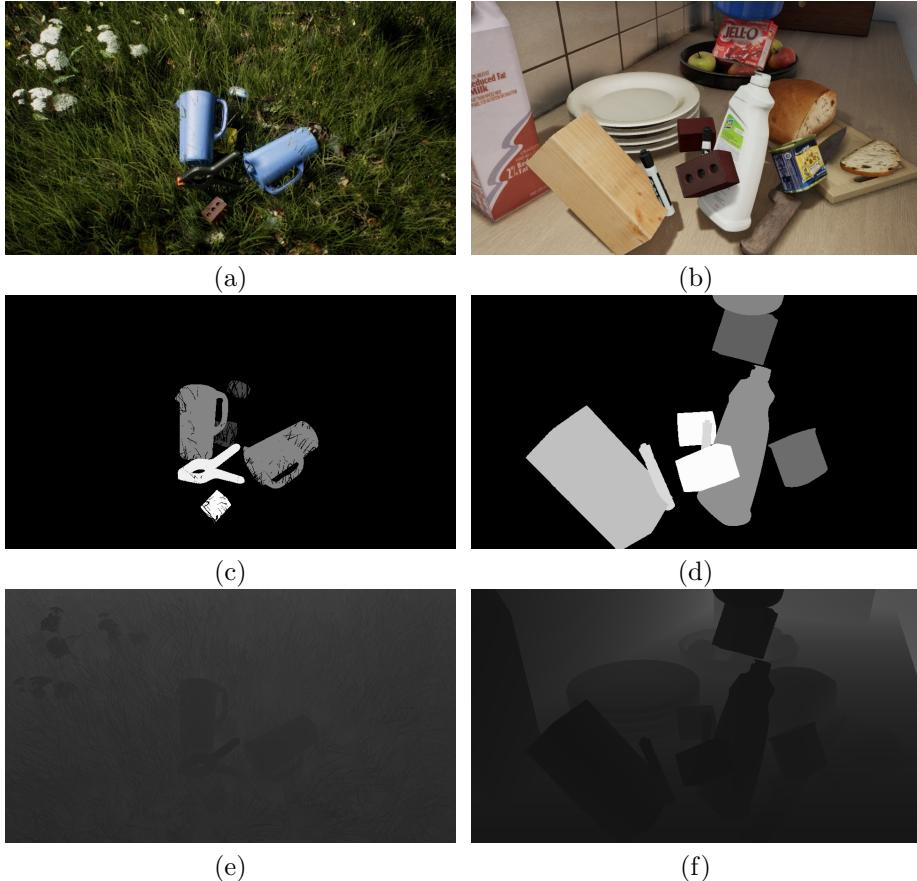


Figure 4: Sample images from the *Falling Things* dataset [594]: (a-b) RGB images, (c-d) ground truth segmentation maps; (e-f) depth maps.

we will see how this progress helps to solve the basic computer problems: after all, recognizing *synthetic* objects is never the end goal.

2.2.2 Improving high-level computer vision with synthetic data

A long line of work has used synthetic data for object detection. Peng et al. [445] trained an object detection framework on non-realistic images of 3D models superimposed on real backgrounds, noting, in particular, that the results improve when synthetic data is varied along 3D pose, texture and color. Bochinski et al. [64] were one of the first to train object detection CNNs on purely synthetic datasets and show that the results transfer to real world evaluation data. Rajpura et al. [477] developed a Blender-based synthetic scene generator for recognizing objects inside a refrigerator, showing improved results with a fully convolutional version of GoogLeNet [571] adapted for object detection. Bayraktar et al. [41] show improvements in object detection on a hybrid dataset in the context of robotics, extending a real dataset with images generated by the *Gazebo* simulation environment (see Section 3.3). In a recent work, Bayraktar et al. [42] test modern object recognition architectures such as *VGGNet* [333],

Inception v3 [570], *ResNet* [238], and *Xception* [117] by fine-tuning them on the *ADORESet* dataset that contains 2500 real and 750 synthetic images for each of 30 object categories in the context of robotic manipulation; they find that a hybrid dataset achieves much better recognition quality compared to purely synthetic or purely real datasets. Recent applications of synthetic data for object detection include the detection of objects in vending machines [623], objects in piles for training robotic arms [75], computer game objects [560], smoke detection [663], deformable part models [681], face detection in biomedical literature [140], drone detection [504], and more.

Recently, Nowruzi et al. [426] studied the question of how much real data is actually needed for object detection in comparison to synthetic data. Using SSD-MobileNet [254], they have compared different modes of training for a number of synthetic and real datasets. They conclude that fine-tuning models trained on synthetic datasets with a small amount of real data is preferable to mixed training on a hybrid dataset with the same amount of real data, and that photorealism appears to be less important than the diversity of synthetic data; this runs contrary to the conclusions of the works [601, 642] that we discuss below.

An interesting method of using synthetic data for object detection was proposed by Hinterstoisser et al. [246]. They note that training on purely synthetic data may give sub-par results due to the low-level differences between synthetic (rendered) images and real photographs. To avoid this, they propose to simply freeze the lower layers of, say, a pretrained object detection architecture and only train the top layers on synthetic data; in this way, basic features will remain suited for the domain of real photos while the classification part (top layers) can be fine-tuned for new classes. Otherwise, this is a straightforward test of synthetic data: Hinterstoisser et al. superimpose synthetic renderings on randomly selected backgrounds and fine-tune pretrained Faster-RCNN [489], R-FCN [134], and Mask R-CNN [237] object detection architectures with freezed feature extraction layers. They report that freezing the layers helps significantly, and different steps in the synthetic data generation pipeline (different domain randomization steps, see also Section 5.1) help as well, obtaining results close to training on a large real dataset.

Segmentation is another classical computer vision problem with obvious benefits to be had from pixel-perfect synthetic annotations. The above-mentioned *SceneNet RGB-D* dataset by McCormac et al. [401] comes with a study showing that an RGB-only CNN for semantic segmentation pretrained from scratch on purely synthetic data can improve over CNNs pretrained on ImageNet; as far as we know, this was the first time synthetic data managed to achieve such an improvement. The dataset is actually an extension of *SceneNet* [232, 234], an annotated model generator for indoor scene understanding that can use existing datasets of 3D object models and place them in 3D environments with synthetic annotation. By now, segmentation models are commonly trained with synthetic data: semantic segmentation is the main problem for most automotive driving models (Section 3.1) and indoor navigation models (Section 3.2), Grard et al. [209] do it for object segmentation in depth maps of piles of bulk objects, and so on.

Saleh et al. [515] note that not all classes in a semantic segmentation problem are equally suited for synthetic data. Foreground classes that correspond to objects (people, cars, bikes etc., i.e., *things* in the terminology of [241]) are well

suited for object detectors (that use shape a lot) but suffer from the synthetic-to-real transfer for segmentation networks because their textures (which segmentation models usually rely upon) are hard to make photorealistic. On the other hand, background classes (grass, road surface, sky etc., i.e., *stuff* in the terminology of [241]) look very realistic on synthetic images due to their high degree of “texture realism”, and a semantic segmentation network can be successfully trained on synthetic data for background classes. Therefore, Saleh et al. propose a pipeline that combines detection-based masks by Mask R-CNN [237] for foreground classes and semantic segmentation masks by *DeepLab* [101] for background classes.

Although most works on synthetic data use large and well-known synthetic datasets, there are many efforts to bring synthetic data to novel applications by developing synthetic datasets from scratch. For instance, O’Byrne et al. [427] develop a synthetic dataset for biofouling detection on marine structures, i.e., segmenting various types of marine growth on underwater images. Ward et al. [634] improve leaf segmentation for *Arabidopsis* plants for the CVPPP Leaf Segmentation Challenge by augmenting real data with a synthetic dataset produced with Blender. For a parallel challenge of leaf counting, Ubbens et al. [605] produce synthetic data based on an L-system plant model; they report improved counting results. Moiseev et al. [416] propose a method to generate synthetic street signs, showing improvements in their recognition. Neff et al. [422] use GANs to produce synthetically augmented data for small segmentation datasets (see Section 6.4). We also note that in other problems, such as video stream summarization, researchers are also beginning to use synthetic data [12].

Another important class of applications for CGI-based synthetic data relates to problems such as 3D pose, viewpoint, and depth estimation, where manual labeling of real data is very difficult and sometimes close to impossible. One of the basic problems here is 2D-3D alignment, the problem of finding correspondences between regions in a 2D image and a 3D model (this also implies pose estimation for objects). In an early work, Aubry et al. [29] solved the 2D-3D alignment problem for chairs with a dataset of synthetic CAD models. Gupta et al. [224] train a CNN to detect and segment object instances for 3D model alignment with synthetic data with renderings of synthetic objects. Su et al. [562] learn to recognize 3D shapes from several 2D images, training their multi-view CNNs on synthetic 2D views. Triyonoputro et al. [599] train a deep neural network on multi-view synthetic images to help visual servoing for an industrial robot. Liu et al. [371] perform indoor scene modeling from a single RGB image by training on a dataset of 3D models, and in other works [368] do 2D-3D alignment from a single image for indoor basic objects. Shoman et al. [543] use synthetic data for camera localization (a crucial part of tracking and augmented reality systems), using synthetic data to cover a wide variety of lighting and weather conditions. They use an autoencoder-like architecture to bring together the features extracted from real and synthetic data and report significantly improved results.

3D position and orientation estimation for objects, known as the 6-DoF (degrees of freedom) pose estimation, is another important computer vision problem related to robotic grasping and manipulation. NVIDIA researchers Tremblay et al. [596] approach it with synthetic data: using the synthetic data generation techniques we described in Section 2.2, they train a deep neural network and report the first state of the art network for 6-DoF pose estimation

trained purely on synthetic data. The novelty was that Tremblay et al. train on a mixture of domain randomized images, where distractor objects are placed randomly in front of a random background, and photorealistic images, where the foreground objects are placed in 3D background scenes obeying physical constraints; domain randomized images provide the diversity needed to cover real data (see Section 5.1) while realistic images provide proper context for the objects and are easier to transfer to real data. Latest results [391, 418, 471] show that synthetic data, especially with proper domain randomization for the data and domain adaptation for the features, can indeed successfully transfer 3D pose estimation from synthetic to real objects.

This also relates to depth estimation; synthetic renderings are easy to augment with pixel-perfect depth maps, and many synthetic datasets include RGB-D data. Carlucci et al. [87] created VANDAL, one of the first synthetic depth image databases, collecting 3D models from public CAD repositories for about 480 *ImageNet* categories of common objects; the authors showed that features extracted from these depth images by common CNN architectures improve object classification and are complementary to features extracted by the same architectures trained on *ImageNet*. Liebelt et al. [360] used 3D models to extract a set of 3D feature maps, then used a nearest neighbors approach to do multi-view object class detection and 3D pose estimation. Lee and Moloney [349] present a synthetic dataset with high quality stereo pairs and show that deep neural networks for stereo vision can perform competitively with networks trained on real data. *Siemens* researchers Planche et al. [455] consider the problem of more realistic simulation of depth data from real sensors and present *DepthSynth*, an end-to-end framework able to generate realistic depth data rather than purely synthetic perfect depth maps; they show that this added realism leads to improvements with modern 2.5D recognition methods.

Easy variations and transformations provided by synthetic data can not only directly improve the results by training, but also represent a valuable tool for studying the properties of neural networks and other feature extractors. In particular, Pinto et al. [453] used synthetic data to study the invariance of different existing visual feature sets to variation in position, scale, pose, and illumination, while Kaneva et al. [306] used a photorealistic virtual environment to evaluate image feature descriptors. Peng et al. [444], Pepik et al. [447], and Aubry and Russell [30] used synthetic data to study the properties of deep convolutional networks, in particular robustness to various transformations, since synthetic data is easy to manipulate in a predefined way.

Earlier works recognized that the domain gap between synthetic and real images does not allow to expect state of the art results when training on synthetic data only, so many of them concentrated on bridging this gap by constructing hybrid datasets. In particular, Vázquez et al. [612] considered pedestrian detection and proposed a scheme based on active learning: they initially train a detector on virtual data and then use selective sampling [121] to choose a small subset of real images for manual labeling, achieving results on par with training on a large real datasets while using 10x less real data.

Purely synthetic approaches were also used in early works, although mostly for problems where manual labeling would be even harder and noisier than for object detection or segmentation. The *Render for CNN* approach by Su et al. [563] outperformed real data with a hybrid synthetic+real dataset on the viewpoint estimation problem. Synthetic data helped improve 3D object pose

estimation in Gupta et al. [225] and multi-view object class detection in Liebelt and Schmid [359] and Stark et al. [558]; as an intermediate step, the latter work used synthetic data to learn shape models. Hattori et al. [236] trained scene-specific pedestrian detectors on a purely synthetic dataset, superimposing rendered pedestrians onto a fixed real scene background; synthetic data has also been used for pedestrian detection by Marin et al. [396].

We finish this section by returning to an important question for direct applications: how realistic must synthetic data be in order to help with the underlying computer vision problem? Early works often argued that photorealism is not necessary for good domain transfer results; see, e.g., [567]. This question was studied in detail by Movshovitz-Attias et al. [419]. With the example of the viewpoint estimation problem for cars, they showed that photorealistic rendering does indeed help, showed that the gap between models trained on synthetic and real data can often be explained by domain adaptation (i.e., adapting from a different real dataset would be just as hard as adapting from a synthetic one), and hybrid synthetic+real datasets can significantly outperform training on real data only.

Another data point is provided by Tsirikoglou et al. [601] who present a very realistic effort for the rendering of synthetic data, including Monte Carlo-based light transport simulation and simulation of optics and sensors, within the domain of rendering outdoor scenes (see also Section 3.1, where we discuss a continuation of this work by Wrenninge and Unger [642]). They show improved results in object detection over other synthetic datasets and conclude that “a focus on maximizing variation and realism is well worth the effort”.

2.3 Synthetic people

Synthetic models and images of people (both faces and full bodies) are an especially interesting subject for synthetic data. On one hand, real datasets here are even harder to collect due to several reasons:

- there are privacy issues involved in the collection of real human faces;
- the labeling for some basic computer vision problems is especially complex: while pose estimation is doable, facial keypoint detection (a key element for facial recognition and image manipulation for faces) may require to specify several dozen landmarks on a human face, which becomes very hard for human labeling [141, 575];
- finally, even in the presence of a large dataset it often contains biases in its composition of genders, races, or other parameters, sometimes famously so [327, 591].

On the other hand, there are complications as well:

- synthetic 3D models people and especially synthetic faces are much harder to create than models of basic objects, especially if sufficient fidelity is required to ;
- basic human-related tasks are very important in practice, so there already exist large real datasets for face recognition [84, 220, 355], pose estimation [24, 274, 373, 600], and other problems, which often limits synthetic

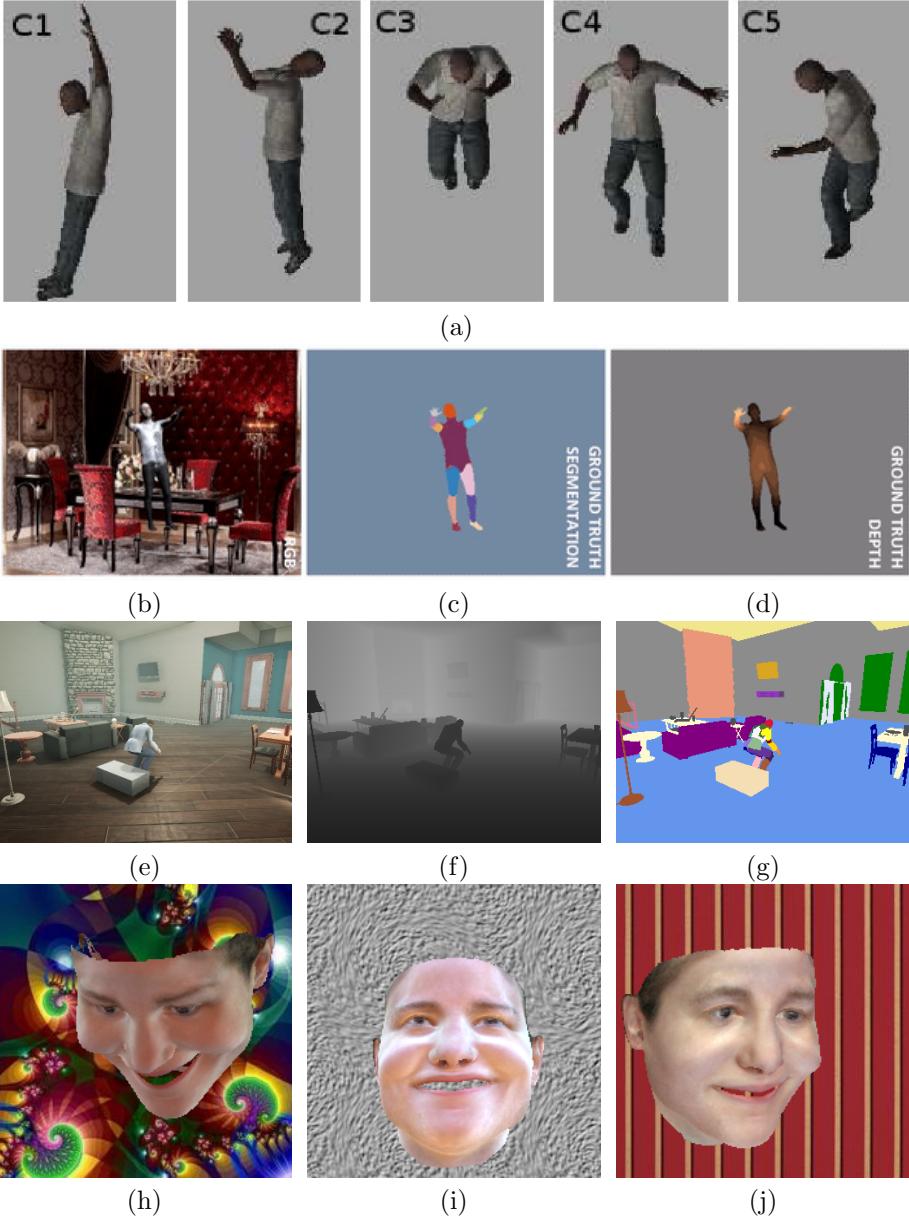


Figure 5: Sample images from human-related synthetic datasets: (a) video frames from *ViHASi* [475]; (b-d) a frame with ground truth from SUR-REAL [609]: (b) RGB image, (c) segmentation map, (d) depth map; (e-g) a frame from PHAV [132]: (e) RGB image, (f) segmentation map, (g) depth map; (h-j) sample synthetic faces with randomized backgrounds from [328,329] (based on the Basel Face Model).

data to covering corner cases, augmenting real datasets, or serving more exotic use cases.

This creates a tension between the quality of available synthetic faces and improvements in face recognition and other related tasks that they can provide. In this section, we review how synthetic people have been used to improve computer vision models in this domain.

In an early effort, Queiroz et al. [468] presented a pipeline for generating synthetic videos with automatic ground truth for human faces and the resulting *Virtual Human Faces Database* (VHuF) dataset with realistic face skin textures that can be extracted from real photos. Bak et al. [32] present the *Synthetic Data for person Re-Identification* (SyRI) dataset with virtual 3D humans designed with *Adobe Fuse CC* to make the models and *Unreal Engine 4* for high-speed rendering. Interestingly, they model realistic lighting conditions by using real HDR environment maps collected with light probes and panoramic photos.

While face recognition for full-face frontal high-quality photos has been mostly solved in 2D, achieving human and superhuman performance both for classification [575] and retrieval via embeddings [529], pose-invariant face recognition *in the wild* [149, 261, 620], i.e., under arbitrary angles and imperfect conditions, remains challenging. Here synthetic data is often used to augment a real dataset, where frontal photos usually prevail, with more diverse data points; we refer to Section 6.1.3 for a detailed overview of the works by Huang et al. [264] and Zhao et al. [707, 708] on GAN-based refinement.

An interesting approach to creating synthetic data for face recognition is provided by Hu et al. [256]. In their “Frankenstein” pipeline, they combine automatically detected body parts (eyes, mouth, nose etc.) from different subjects; interestingly, they report that the inevitable artifacts in the resulting images, both boundary effects and variations between facial patches, do not hinder training on synthetic data and may even improve the robustness of the resulting model.

There is also a related field of *3D-aided face recognition*. This approach uses a morphable synthetic 3D model of an abstract human face that has a number of free parameters; the model learns to tune these parameters so that the 3D model fits a given photo and then uses the model and texture from the photo either to produce a frontal image or to directly recognize photos taken from other angles. This is a classic approach, dating back to late 1990s [62, 63] and developed in many subsequent works with new morphable models [257, 268], deep learning used to perform the regression for morphing parameters [592], extended to 3D face scans [61], and so on; see, e.g., the survey [149] for more details. Xu et al. [665] use synthetic data to train their 3D-aided model for pose-invariant face recognition as well. Recent works used GANs to produce synthetic data for 3D-aided face recognition [708]; we discuss this approach in detail in Section 6.1.3.

In a large-scale effort to combat dataset bias in face recognition and related problems with synthetic data, Kortylewski et al. [328, 329] have developed a pipeline to directly create synthetic faces. They use the *Basel Face Model 2017* [199], a 3D morphable model of face shape [61, 63], and take special care to randomize the pose, camera location, illumination conditions, and background. They report significantly improved results for face recognition and facial landmark detection with the *OpenFace* framework [16, 520] and state of the art models for face detection and alignment [692] and landmark detection [478].

Human pose estimation is a very well known and widely studied problem [137, 373, 687] with many direct applications, so it is no wonder that the field

does not suffer from lack of real data, with large-scale datasets available [23, 291, 363] and state of the art models achieving impressive results [564, 568, 669]. However, synthetic data still can help. Ludl et al. [382] show that in corner cases, corresponding to rare activities not covered by available datasets, existing pose estimation models produce errors, but augmenting the training set with synthetic data that covers these corner cases helps improve pose estimation. Another specialized use-case has been considered by Rematas et al. in a very interesting application of pose estimation called “Soccer on Your Tabletop” [486]. They trained specialized pose and depth estimation models for soccer players and produced a unified model that maps 2D footage of a soccer match into a 3D model suitable for rendering on a real tabletop through augmented reality devices. For training, Rematas et al. used synthetic data captured from the *FIFA* video game series. These are model examples of how synthetic data can improve the results even when comprehensive real datasets are available.

Moving from still images to videos, we begin with human action recognition [460, 688]. ViHASi by Ragheb et al. [475] is a virtual environment and dataset for silhouette-based human action recognition. De Souza et al. [132] present PHAV (Procedural Human Action Videos), a synthetic dataset that contains 39,982 videos with more than 1,000 examples for each action of 35 categories. Inria and MPI researchers Varol et al. [609] present the SURREAL (Synthetic hUmans foR REAL tasks) dataset. They generate photorealistic synthetic images with labeling for human part segmentation and depth estimation, producing 6.5M frames in 67.5K short clips (about 100 frames each) of 2.6K action sequences with 145 different synthetic subjects.

Microsoft researchers Khodabandeh et al. [319] present the *DIY Human Action* generator for human actions. Their framework consists of a generative model, called the *Skeleton Trajectory GAN*, that learns to generate a sequence of frames with human skeletons conditioned on the label for the desired action, and a *Frame GAN* that generates photorealistic frames conditioned on a skeleton and a reference image of the person. As a result, they can generate realistic videos of people defined with a reference image that perform the necessary actions, and, moreover, the *Frame GAN* is trained on an unlabeled set of human action videos.

We also note here some privacy-related applications of synthetic data that are not about differential privacy (which we discuss in Section 7). For example, Ren et al. [491] present an adversarial architecture for video face anonymization; their model learns to modify the original real video to remove private information while at the same time still maximizing the performance of action recognition models.

As the problems become dynamic rather than static, e.g., as we move to recognizing human movements on surveillance cameras, synthetic data takes the form of full-scale simulated environments. This direction started a long time ago: already in 2007, The *ObjectVideo Virtual Video* (OVVV) system by Taylor et al. [582] used the *Half-Life 2* game engine with additional camera parameters designed to simulate real-world surveillance cameras to detect a variety of different events. Fernandez et al. [175] place virtual agents onto real video surveillance footage in a kind of augmented reality to simulate rare events. Qureshi and Terzopoulos [470] present a multi-camera virtual reality surveillance system.

An interesting human-related video analysis problem, important for au-

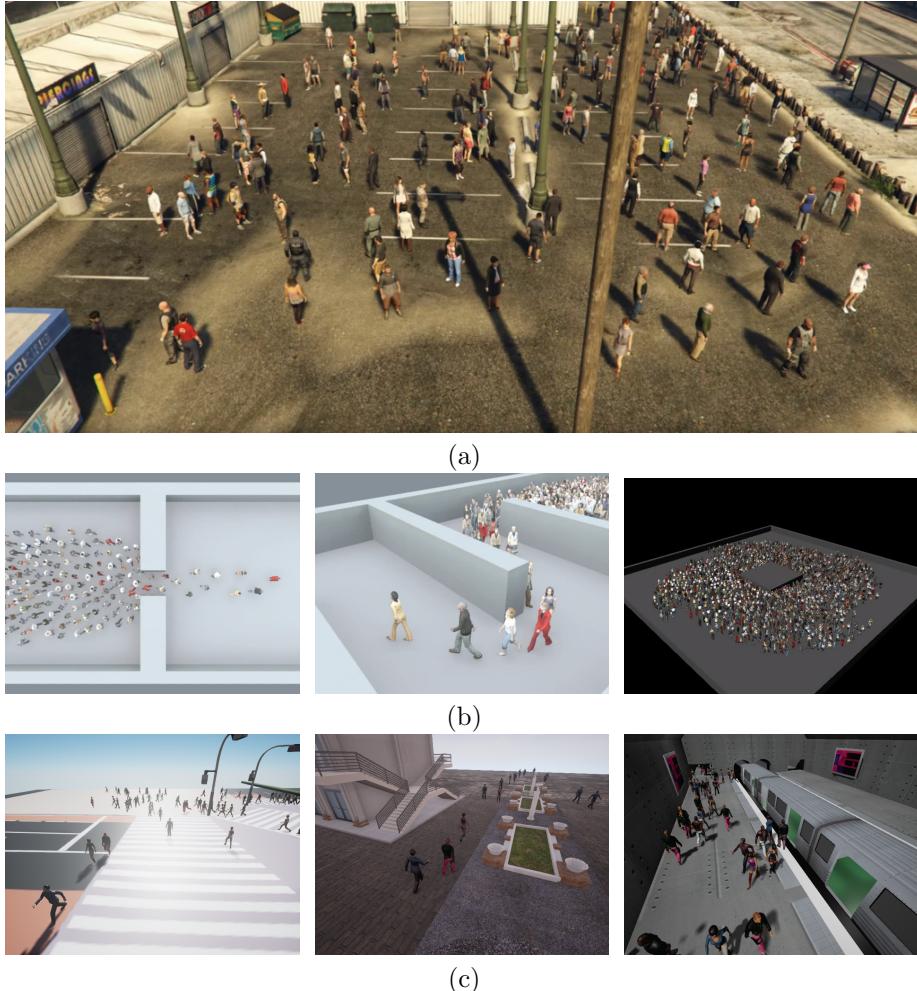


Figure 6: Sample images from synthetic crowd counting datasets: (a) *GTA5 Crowd Counting* [625]; (b) *Agoraset* [127]; (c) *LCrowdV* [110].

tonomous vehicles, is to predict pedestrian trajectories in an urban environment. Anderson et al. [19] develop a method for stochastic sampling-based simulation of pedestrian trajectories. They then train the *SocialGAN* model by Gupta et al. [221] that generates pedestrian trajectories with a recurrent architecture and uses a recurrent discriminator to distinguish fake trajectories from real ones; Anderson et al. show that synthetic trajectories significantly improve the results for a predictive model such as *SocialGAN*.

Another important application of datasets of synthetic people is crowd counting. In this case, collecting ground truth labels, especially if the model is supposed to do segmentation in addition to simple counting, is especially labor-intensive since crowd counting scenes are often highly congested and contain hundreds, if not thousands of people. Existing real datasets are either relatively small or insufficiently diverse; e.g., the UCSD dataset [89] and the Mall dataset [100] have both about 50,000 pedestrians but in each case collected from

a single surveillance camera, the ShanghaiTech dataset [697] has about 330,000 heads but only about 1200 images, again collected on the same event, and the UCF-QNRF dataset [270], while more diverse than previous ones, is limited to extremely congested scenes, with up to 12,000 people on the same image, and has only about 1500 images. The *LCrowdV* system [110] generates labeled crowd videos and shows that augmenting real data with a produced synthetic dataset improves the accuracy of pedestrian detection.

To provide sufficient diversity and scale, Wang et al. [625] presented a synthetic *GTA5 Crowd Counting* dataset collected with the help of the *Grand Theft Auto V* engine; the released dataset contains about 15,000 synthetic images with more than 7,5 million annotated people in a wide variety of scenes. They compare various approaches to crowd counting as a supervised problem, in particular their new spatial fully convolutional network (SFCN) model that directly predicts the density map of people on a crowded image. They report improved results when pretraining on GCC and then fine-tuning on a real dataset; they also consider GAN-based approaches that we discuss in Section 6.2. A more direct approach to generating synthetic data has been developed by Ekbatani et al. [168], who extract real pedestrians from images and add them at various locations on other backgrounds, with special improvement procedures for added realism; they also report improved counting results. Khadka et al. [316] also present a synthetic crowd dataset, showing improvements in crowd counting.

This ties into crowd analysis, where synthetic data is used to model crowds and train visual crowd analysis tools on rendered images [549]. Huang et al. [262] present virtual crowd models that could be used for such simulations. Courty et al. [127] present the *Agoraset* dataset for crowd analysis research that aims to provide realistic agent trajectories (done through the social force model by Heibling and Molnár [242]) and high-quality rendering with the Mental Ray renderer [157]; the dataset has 26 different characters and provides a variety of different scenes: corridor, flow around obstacles, escape through a bottleneck, and so on.

In general, we summarize that while the most popular problems such as frontal face recognition or human pose estimation are already being successfully solved with models trained on real datasets (because there has been sufficient interest for these problems to collect and manually label large-scale datasets), synthetic data remains very important for alleviating the effect of dataset bias in real collections, covering corner cases, and tackling other problems or basic problems with different kinds of data, e.g., in different modalities (such as face recognition with an IR sensor). We believe that there are important opportunities for synthetic data in human-related computer vision problems and expect this field to grow in the near future.

2.4 Character and text recognition

Various tasks related to text recognition, including optical character recognition (OCR), text detection, layout analysis and text line segmentation for document digitization, and others have often been attacked with the help of synthetic data, usually with synthetic text superimposed on real images. This is a standard technique in the field because text pasted in a randomized way often looks quite reasonable even with minimal additional postprocessing. Synthetic data was used for character detection and recognition in, e.g., [10, 169, 628, 686] and

for text block detection in, e.g., [276, 277]. Krishnan and Jawahar [331] use synthetic data to pretrain deep neural networks for learning efficient representations of handwritten word images. Jo et al. [284] train an end-to-end convolutional architecture that can digitize documents with a mixture of handwritten and printed text; to train the network, they produce a synthetic dataset with real handwritten text superimposed on machine-printed forms, with Otsu binarization applied before pasting.

There exist published datasets of synthetic text and software to produce them, in particular MJSynth [276] and SynthText in the Wild [222]. In the latter work, Gupta et al. use available depth estimation and segmentation solutions to find regions (planes) of a natural image suitable for placing synthetic text and find the correct rotation of text for a given plane. Moreover, recent works have used GAN-based refinement (see Section 6.1) to make synthetic text more realistic [165]. There are also synthetic handwriting generation models based on GANs that are conditioned on character sequences and produce excellent results [14, 281].

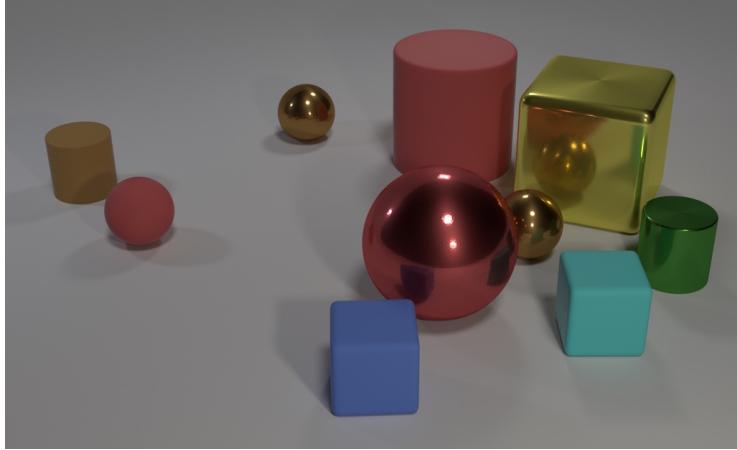
2.5 Visual reasoning

Visual reasoning is the field of artificial intelligence where models are trained to reason and answer questions about visual data. It is usually studied in the form of visual question answering (VQA), when models are trained to answer questions about a picture such as “What is the color of the small metal sphere?” or “Is there an equal number of balls and boxes?”.

There exist datasets for visual question answering based on real photographs, collected and validated by human labelers; they include the first large dataset called VQA [8] and its recent extension, VQA v2.0 [208]. However, the problem yields itself naturally to automated generation, so it is no wonder that synthetic datasets are important in the field.

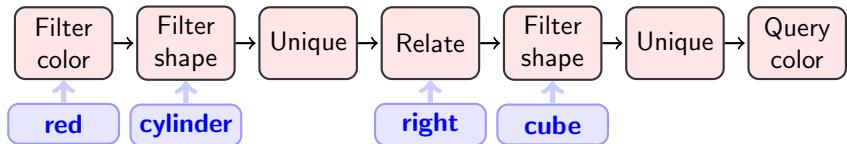
The most important synthetic VQA dataset is CLEVR (Compositional Language and Elementary Visual Reasoning), created by Johnson et al. in a collaboration between Stanford and Facebook Research [288]. It contains 100K rendered images with scenes composed of simple geometric shapes and about 1M (853K unique) automatically generated questions about these images. The intention behind this dataset was to enable detailed analysis of VQA models, simplifying visual recognition and concentrating on reasoning about the objects.

In CLEVR, scenes are represented as scene graphs [290, 330], where the nodes are objects annotated with attributes (shape, size, material, and color) and edges correspond to spatial relations between objects (“left”, “right”, “behind”, and “in front”). A scene can be rendered based on its scene graph with randomized positions of the objects. The questions are represented as functional programs that can be executed on scene graphs, e.g., “What color is the cube to the right of the white sphere?”. Different question types include querying attributes (“what color”), comparing attributes (“are they the same size”), existence (“are there any”), counting (“how many”), and integer comparison (“are there fewer”). When generating questions, special care is taken to ensure that the answer exists and is unique, and then the natural language question is generated with a relatively simple grammar. Figure 7 shows two sample questions and their functional programs: on Fig. 7a the program is a simple chain of filters, and Fig. 7b adds a logical connective, which makes the graph a tree.



(a)

(b) Chain-structured question: *What color is the cube to the right of the red cylinder?*



(c) Tree-structured question: *How many spheres are behind the blue thing and on the left side of the left cylinder?*

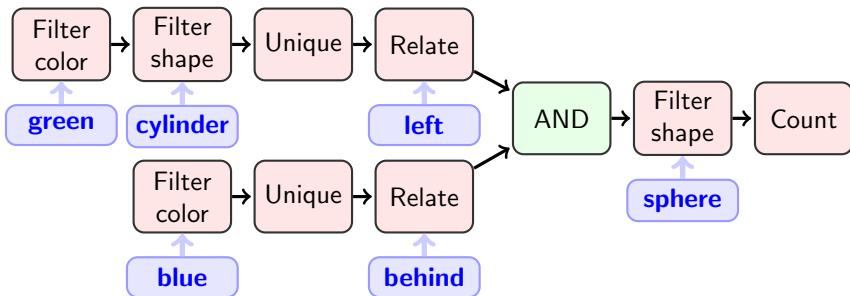


Figure 7: The CLEVR dataset [288]: (a) sample image; (b-c) sample visual reasoning questions.

We also note a recently published COG dataset produced by Google Brain researchers [667] that extends CLEVR’s ideas to video processing. It also contains synthetic visual inputs and questions generated from functional programs, but now questions can refer to time (e.g., “what is the color of the latest triangle?”). The authors also released a generator that can produce synthetic video-question pairs that are progressively more challenging and that have minimal response bias, an important problem for synthetic datasets (in this case, the generator begins with a balanced set of target responses and then generates videos and questions for them rather than the other way around).

3 Synthetic simulated environments

While collecting synthetic datasets is a challenging task by itself, it is insufficient to train, e.g., an autonomous vehicle such as a self-driving car or a drone, or an industrial robot. Learning to control a vehicle or robot often requires *reinforcement learning*, where an agent has to learn from interacting with the environment, and real world experiments to train a self-driving car or a robotic arm are completely impractical. Fortunately, this is another field where synthetic data shines: once one has a fully developed 3D environment that can produce datasets for computer vision or other sensory readings, it is only one more step to active interaction with this environment. Therefore, in most domains considered below we can see the shift from static synthetic datasets to interactive simulation environments.

Reinforcement learning (RL) agents are commonly trained on simulations because the interactive nature of reinforcement learning makes training in the real world extremely expensive. We discuss synthetic-to-real domain adaptation in this context in Section 6.3. However, in many works, there is no explicit domain adaptation: robots are trained on simulators and later fine-tuned on real data or simply transferred to the real world.

Table 2 shows a brief summary of datasets and simulators that we review in this section. To make the exposition more clear, we group together both environments and “static” synthetic datasets for outdoor (Section 3.1) and indoor (Section 3.2) scenes, including some works that use them to improve RL agents and other models. Next, we consider synthetic robotic simulators (Section 3.3) and vision-based simulators for autonomous flying (Section 3.4), finishing with an idea of using computer games as simulation environments in Section 3.5. Reinforcement learning in virtual environments remains a common thread throughout this section.

3.1 Urban and outdoor environments: learning to drive

An important direction of applications for synthetic data is related to navigation, localization and mapping (SLAM), or similar problems intended to improve the motion of autonomous robots. Possible applications include SLAM, motion planning, and motion for control for self-driving cars (urban navigation) [177, 346, 411, 435], unmanned aerial vehicles [11, 126, 305], and more; see also general surveys of computer vision for mobile robot navigation [68, 142] and perception and control for autonomous driving [443].

Before proceeding to current state of the art, we note an interesting historical fact: one of the first autonomous driving attempts based on neural networks, ALVINN [458], which used as input 30×32 videos supplemented with 8×32 range finder data, was already training on synthetic data. As early as 1989, the authors remark that “training on actual road images is logically difficult because... the network must be presented with a large number of training exemplars... under a wide variety of conditions” and proceed to describing a simulator. One of the first widely adopted full-scale visual simulation environments for robotics, *Gazebo* [324] (see Section 3.3), provided both indoor and outdoor environments for robotic control training.

In a much more recent effort, *Xerox* researchers Gaidon et al. [184] pre-

Name	Year	Ref	Engine	Size / comments
<i>Outdoor urban environments, driving</i>				
TORCS	2014	[651]	Custom	Game-based simulation engine
Virtual KITTI	2016	[184]	Unity	5 environments, 50 videos
GTAVision	2016	[292]	GTA V	GTA plugin, 200K images
SYNTHIA	2016	[499]	Unity	213K images
GTAV	2016	[494]	GTA V	25K images
VIPER	2017	[493]	GTA V	254K images
CARLA	2017	[153]	UE	Simulator
VIES	2018	[515]	Unity3D	61K images, 5 environments
ParallelEye	2018	[585]	Esri	Procedural gen, import from OSM
VIVID	2018	[341]	UE	Urban sim with emphasis on people
DeepDrive	2018	[469]	UE	Driving sim + 8.2h of videos
PreSIL	2019	[269]	GTA V	50K images with LIDAR point clouds
AADS	2019	[356]	Custom	3D models of cars on real backgrounds
WoodScape	2019	[674]	Custom	360° panoramas with fisheye cameras
ProcSy	2019	[317]	Esri	Procedural generation with varying conditions
<i>Robotic simulators and aerial navigation</i>				
Gazebo	2004	[324]	Custom	Industry standard robotic sim
MuJoCo	2012	[589]	Custom	Common physics engine for robotics
AirSim	2017	[539]	UE	Sensor readings, hardware-in-the-loop
CAD ² RL	2017	[513]	Custom	Indoor flying sim
X-Plane	2019	[555]	X-Plane	8K landings, 114 runways
Air Learning	2019	[332]	—	Platform for flying sims
VRGym	2019	[658]	UE	VR for human-in-the-loop training
ORRB	2019	[112]	Unity	Accurate sim used to train real robots
<i>Indoor environments</i>				
ICL-NUIM	2014	[233]	Custom	RGB-D with noise models, 2 scenes
SUNCG	2016	[557]	Custom	45K floors, 3D models
MINOS	2017	[521]	SUNCG	Indoor sim based on SUNCG
AI2-THOR	2017	[325]	Unity3D	Indoor sim with actionable objects
House3D	2018	[647]	SUNCG	Indoor sim based on SUNCG
Habitat	2019	[393]	Custom	Indoor sim platform and library

Table 2: An overview of synthetic datasets and virtual environments discussed in Section 3.

sented a photorealistic synthetic video dataset Virtual KITTI³ intended for object detection and multi-object tracking, scene-level and instance-level semantic segmentation, optical flow, and depth estimation. The dataset contains five different virtual outdoor environments created with the Unity game engine and 50 photorealistic synthetic videos. Gaidon et al. studied existing multi-object trackers, e.g., based on an improved min-cost flow algorithm [454] and on Markov decision processes [653]; they found minimal real-to-virtual gap. Note, however, that experiments in [184] were done on trackers trained on real data and evaluated on synthetic videos (and that’s where they worked well), not the other way around. In general, the *Virtual KITTI* dataset is much too small to train a model on it, it is intended for evaluation, which also explains the experimental setup.

Johnson-Robertson et al. [292], on the other hand, presented a method to

³The name comes from the KITTI dataset [197, 409] created in a joint project of the Karlsruhe Institute of Technology and Toyota Technological Institute at Chicago.

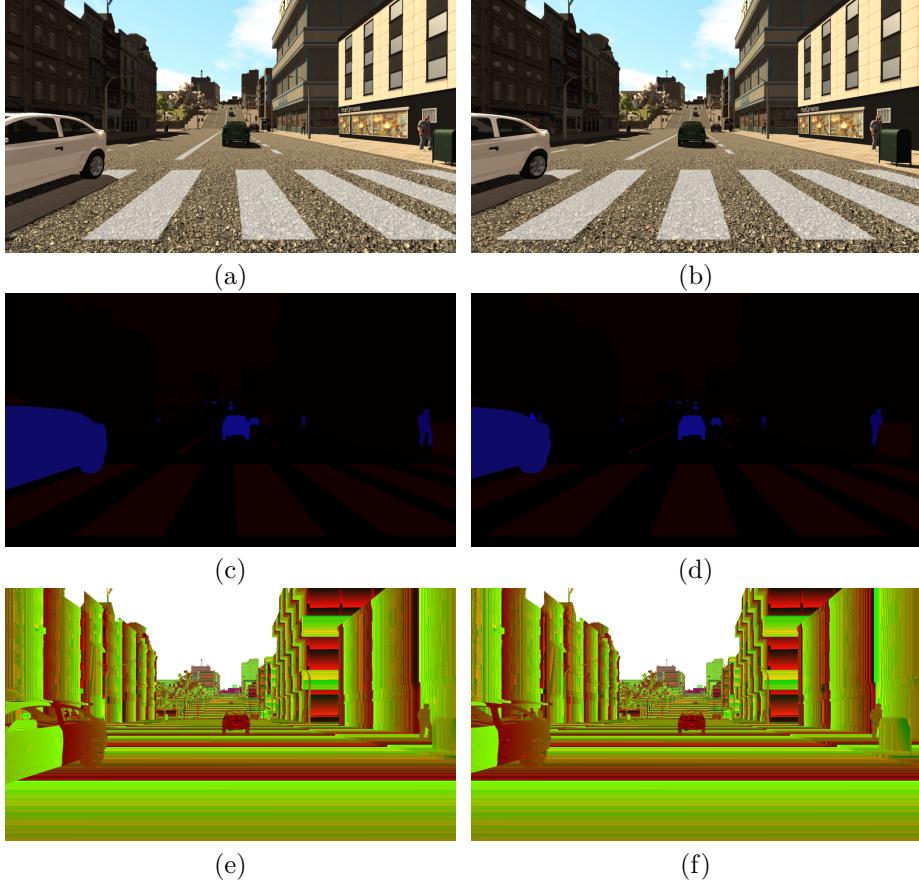


Figure 8: Sample images from SYNTHIA-SF [245]: (a-b) RGB ground truth (left and right camera); (c-d) ground truth segmentation maps; (e-f) depth maps (depth is encoded in the color as $R + 256 \cdot G + 256^2 \cdot B$).

train on synthetic data. They collected a large dataset by capturing scene information from the *Grand Theft Auto V* video game that provides sufficiently realistic graphics and at the same time stores scene information such as depth maps and rough bounding boxes in the GPU stencil buffer, which can also be captured; the authors developed an automated pipeline to obtain tight bounding boxes. Three datasets were generated, with 10K, 50K, and 200K images respectively. The main positive result of [292] is that a standard Faster R-CNN architecture [489] trained on 50K and 200K images outperformed on a real validation set (KITTI) the same architecture trained on a real dataset. The real training set was *Cityscapes* [123] that contains 2,975 images, so while the authors used more synthetic data than real, the difference is only 1-2 orders of magnitude. The VIPER and GTAV datasets by Richter et al. [493, 494] were also captured from *Grand Theft Auto V*; the latter provides more than 250K 1920×1080 images fully annotated with optical flow, instance segmentation masks, 3D scene layout, and visual odometry.

The SYNTHIA dataset presented by Ros et al. [499] provides synthetic im-

ages of urban scenes labeled for semantic segmentation. It consists of renderings of a virtual New York City constructed by the authors with the *Unity* platform and includes segmentation annotations for 13 classes such as pedestrians, cyclists, buildings, roads, and so on. The dataset contains more than 213,000 synthetic images covering a wide variety of scenes and environmental conditions; experiments in [499] show that augmenting real datasets with SYNTHIA leads to improved segmentation. Later, Hernandez-Juarez et al. [245] presented SYNTHIA-SF, the San Francisco version of SYNTHIA. We illustrate SYNTHIA with a sample frame (that is, two frames since the dataset contains two cameras) from the SYNTHIA-SF dataset on Figure 8.

Saleh et al. [515] presented a Unity3D framework called VEIS (Virtual Environment for Instance Segmentation); while not very realistic, it worked well with their detection-based pipeline (see Section 2.2.2). Li et al. [354] present a synthetic dataset with foggy images to simulate difficult driving conditions. We note the work of Lopez et al. [378] whose experiments suggest that the level of realism achieved in SYNTHIA and GTAV is already sufficient for successful transfer of object detection methods.

Tian et al. [585] present the *ParallelEye* synthetic dataset for urban outdoor scenes. Their approach is rather flexible and relies on previously developed *Esri CityEngine* framework [656] that provides capabilities for batch generation of 3D city scenes based on terrain data. In [585], this data is automatically extracted from the OpenStreetMap platform⁴. The 3D scene is then imported into the Unity3D game engine, which helped add urban vehicles on the roads, set up traffic rules, add support for different weather and lighting conditions. Tian et al. showed improvements in object detection quality for state of the art architectures trained on *ParallelEye* and tested on the real KITTI test set as compared to training on the real KITTI training set.

Li et al. [356] develop the *Augmented Autonomous Driving Simulation* (AADS) environment that is able to insert synthetic traffic on real-life RGB images. Starting from the real-life *ApolloScape* dataset for autonomous driving [265] that contains LIDAR point clouds, the authors remove moving objects, restore backgrounds by inpainting, estimate illumination conditions, simulate traffic conditions and trajectories of synthetic cars, preprocess the textures of the models according to lighting and other conditions, and add synthetic cars in realistic places on the road. In this way, a single real image can be reused many times in different synthetic traffic situations. This is similar to the approach of Abu Alhaija et al. [5] (Section 2.2.1) but due to available 3D information AADS can also change the observation viewpoint and even be used in a closed-loop simulator such as CARLA or *AirSim* (see below). We do not go into details on the already large and diverse field of virtual traffic simulation and refer to a recent survey [94].

Wrenninge and Unger [642] present the *Synscapes* dataset that continues the work of Tsirikoglou et al. [601] (see Section 2.2.2) and contains accurate photorealistic renderings of urban scenes (Fig. 10e-f), with unbiased path tracing for rendering, special models for light scattering effects in camera optics, motion blur, and more. They find that their additional efforts for photorealism do indeed result in significant improvements in object detection over GTA-based datasets, even though the latter have a wider variety of scenes and pedestrian

⁴<https://www.openstreetmap.org/>

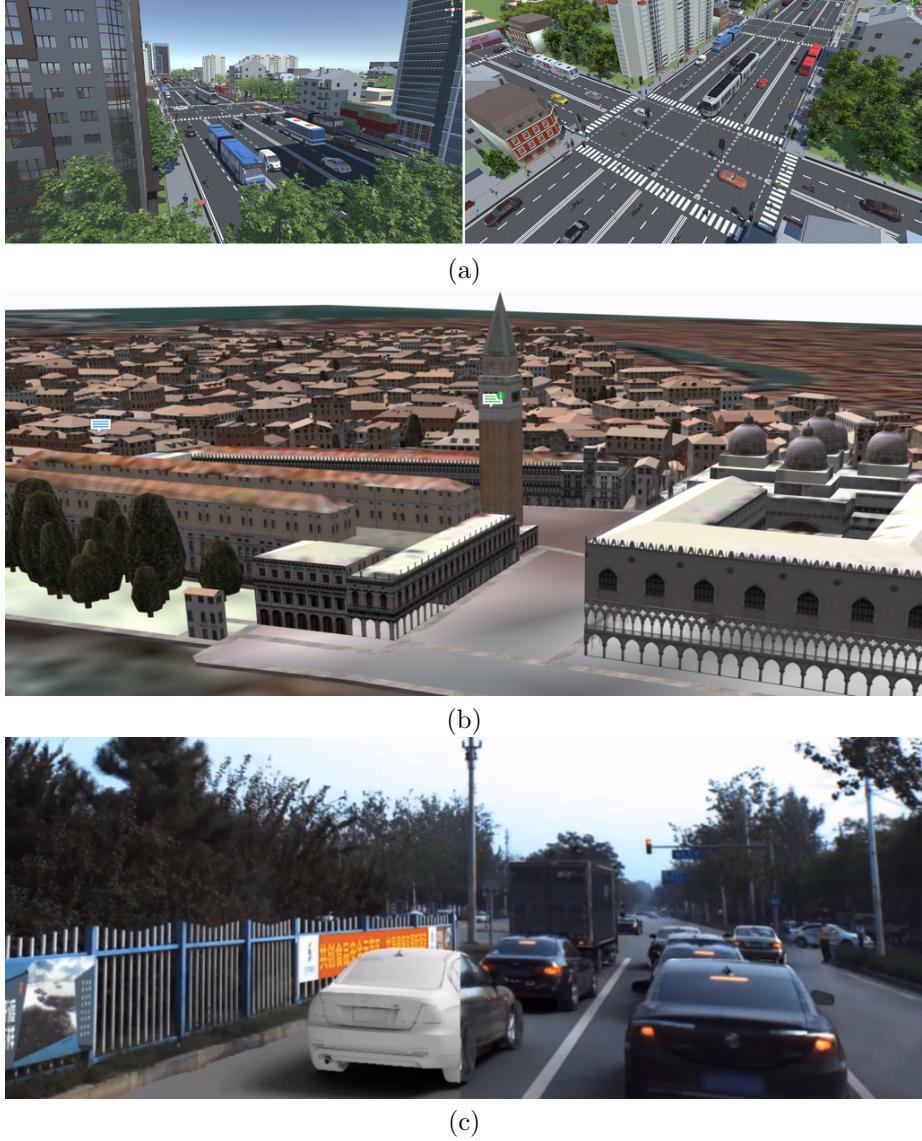


Figure 9: Sample images from synthetic outdoor datasets: (a) VEIS [515]; (b) *Esri CityEngine* Venice sample scene [656]; (c) AADS [356] (part of a frame from a showcase video).

and car models.

Khan et al. [317] introduce *ProcSy*, a procedurally generated synthetic dataset aimed at semantic segmentation (we showed a sample frame on Fig. 1c-d). It is modeling a real world urban environment, and its main emphasis is on simulating various weather and lighting conditions for the same scenes. The authors show that, e.g., adding a mere 3% of rainy images in the training set improves the mIoU of a state of the art segmentation network (in this case, Deeplab v3+ [102]) by as much as 10% on rainy test images. This again supports the

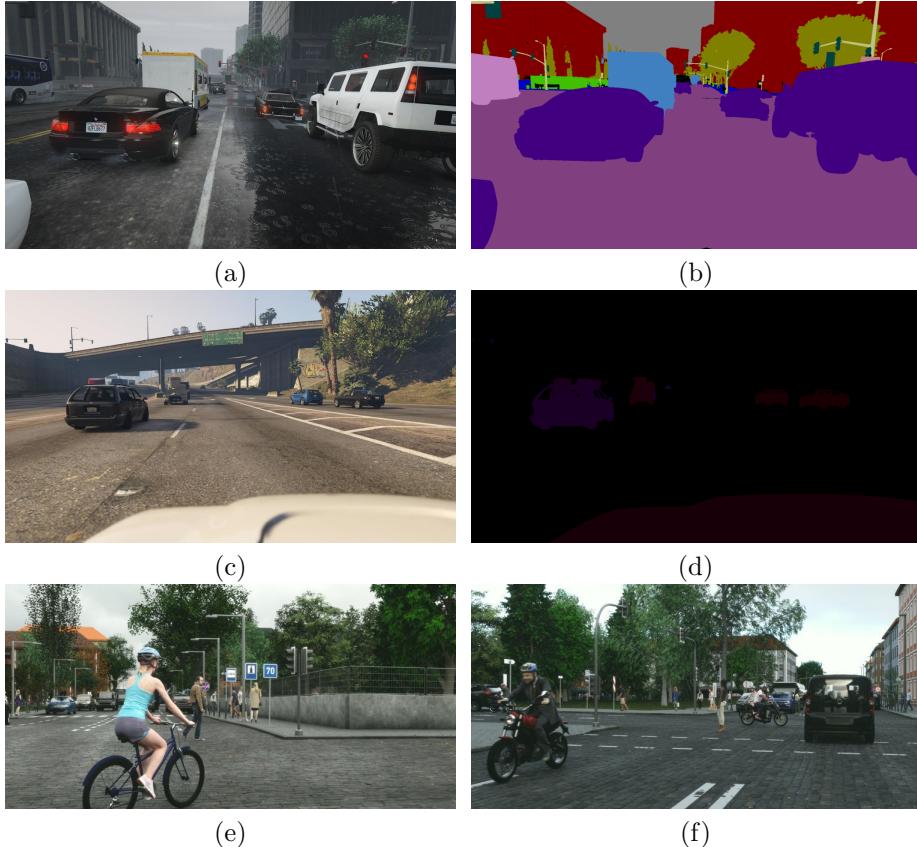


Figure 10: Sample images from synthetic outdoor datasets: (a-b) GTAV [494]: (a) RGB image, (b) ground truth segmentation; (c-d) VIPER [493]: (c) RGB image, (d) ground truth segmentation; (e-f) *Synscapes* [642].

benefits from using synthetic data to augment real datasets and cover rare cases; for a discussion of the procedural side of this work see Section 8.1.

Synthetic datasets with explicit 3D data (with simulated sensors) for outdoor environments are less common, although such sensors seem to be straightforward to include into self-driving car hardware. In their development of the *SqueezeSeg* architecture, Wu et al. [643, 644] added a LiDAR simulator to *Grand Theft Auto V* and collected a synthetic dataset from the game. *SynthCity* by Griffiths and Boehm [212] is a large-scale open synthetic dataset which is basically a huge point cloud of an urban/suburban environment. It simulates *Mobile Laser Scanner* (MLS) readings with a *Blender* plugin [214] and is specifically intended for pretraining deep neural networks. Yogamani et al. [674] present *WoodScape*, a multi-camera fisheye dataset for autonomous driving that concentrates on getting 360° sensing around a vehicle through panoramic fisheye images with a large field of view. They record 4 fisheye cameras with 190° horizontal field of view, a rotating LiDAR, GNSS and IMU sensors, and odometry signals with 400K frames with depth labeling and 10K frames with semantic segmentation labeling. Importantly for us, together with their real dataset they also released a

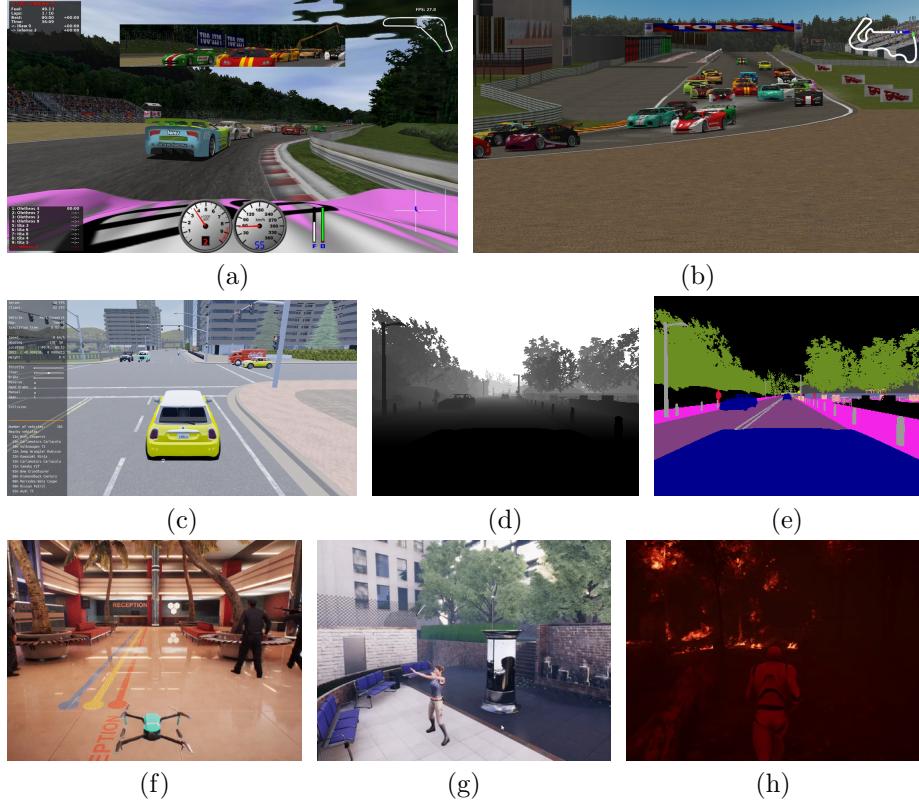


Figure 11: Sample images from outdoor environments: (a-b) TORCS [651]; (c-e) CARLA [153]; (f-h) VIVID [341].

synthetic part (10K frames) that matches their fisheye cameras, with the explicit purpose of helping synthetic-to-real transfer learning. This further validates the importance of synthetic data in autonomous driving.

Simulated environments rather than datasets are, naturally, also an important part of the outdoor navigation scene; below we describe the main players in the field and also refer to surveys [307, 565] for a more in-depth analysis of some of them. There is also a separate line of work related to developing more accurate modeling in such simulators, e.g., sensor noise models [412], that falls outside the scope of this survey.

TORCS⁵ (The Open Racing Car Simulator) [651] is an open source 3D car racing simulator that started as a game for *Linux* in the late 1990s but became increasingly popular as a virtual simulation platform for driving agents and intelligent control systems for various car components. TORCS provides a sufficiently involved simulation of racing physics, including accurate basic properties (mass, rotational inertia), mechanical details (suspension types etc.), friction profiles of tyres, and a realistic aerodynamic model, so it is widely accepted as useful as a source of synthetic data. TORCS has become the basis for the annual Simulated Car Racing Championship [375] and has been used in hundreds of works on autonomous driving and control systems (see Section 3.3).

⁵<http://torcs.sourceforge.net/>

CARLA (CAR Learning to Act) [153] is an open simulator for urban driving, developed as an open-source layer over *Unreal Engine 4* [310]. Technically, it operates similarly to [292], as an open source layer over *Unreal Engine 4* that provides sensors in the form of RGB cameras (with customizable positions), ground truth depth maps, ground truth semantic segmentation maps with 12 semantic classes designed for driving (road, lane marking, traffic sign, sidewalk and so on), bounding boxes for dynamic objects in the environment, and measurements of the agent itself (vehicle location and orientation). *DeepDrive* [469] is a simulator designed for training self-driving AI models, also developed as an *Unreal Engine* plugin; it provides 8 RGB cameras with 512×512 resolution at close to real time rates (20Hz), as well as a generated 8.2 hour video dataset.

VIVID (VIrtual environment for VIsual Deep learning), developed by Lai et al. [341], tackles a more ambitious problem: adding people interacting in various ways and a much wider variety of synthetic environments, they present a universal dataset and simulator of outdoor scenes such as outdoor shooting, forest fires, drones patrolling a warehouse, pedestrian detection on the roads, and more. VIVID is also based on the *Unreal Engine* and uses the wide variety of assets available for it; for example, NPCs acting in the scenes are programmed using *Blueprint*, an *Unreal* scripting engine, and the human models are animated by the *Unreal* animation editor. VIVID provides the ability to record video simulations and can communicate with deep learning libraries via the TCP/IP protocol, through the *Microsoft Remote Procedure Call* (RPC) library originally developed for *AirSim* (see Section 3.3).

As for reinforcement learning (RL) in autonomous driving, the original paper on the CARLA simulator [153] also provides a comparison on synthetic data between conditional imitation learning, deep reinforcement learning, and a modular pipeline with separated perception, local planning, and continuous control, with limited success but generally best results obtained by the modular pipeline. Many works on autonomous driving use TORCS [651] as a testbed, both in virtual autonomous driving competitions and simply as a well-established research platform. We do not aim to provide a full in-depth survey of the entire field and only note that despite its long history TORCS is being actively used for research purposes up to this day. In particular, Sallab et al. [516, 517] use it in their deep reinforcement learning frameworks for lane keeping assist and autonomous driving, Xiong et al. [661] add safety-based control on top of deep RL, Wang et al. [626] train a deep RL agent for autonomous driving in TORCS, Barati et al. [37] use it to add multi-view inputs for deep RL agents, Li et al. [352] develop *Visual TORCS*, a deep RL environment based on TORCS, Ando, Lubashevsky et al. [20, 381] use TORCS to study the statistical properties of human driving, Glassner et al. [202] shift the emphasis to trajectory learning, Luo et al. [383] use TORCS as the main test environment for a new variation of the policy gradient algorithm, Liu et al. [369] make use of the multimodal sensors available in TORCS for end-to-end learning, Xu et al. [576] train a segmentation network and feed segmentation results to the RL agent in order to unify synthetic imagery from TORCS and real data, and so on. In an interesting recent work, Choi et al. [114] consider the driving experience transfer problem but consider a transfer not from a synthetic simulator to the real domain but from one simulator (TORCS) to another (GTA V). Tai et al. [573] learn continuous control for mapless navigation with asynchronous deep RL in virtual environments.

Synthetic data in autonomous driving extends to other sensor modalities as

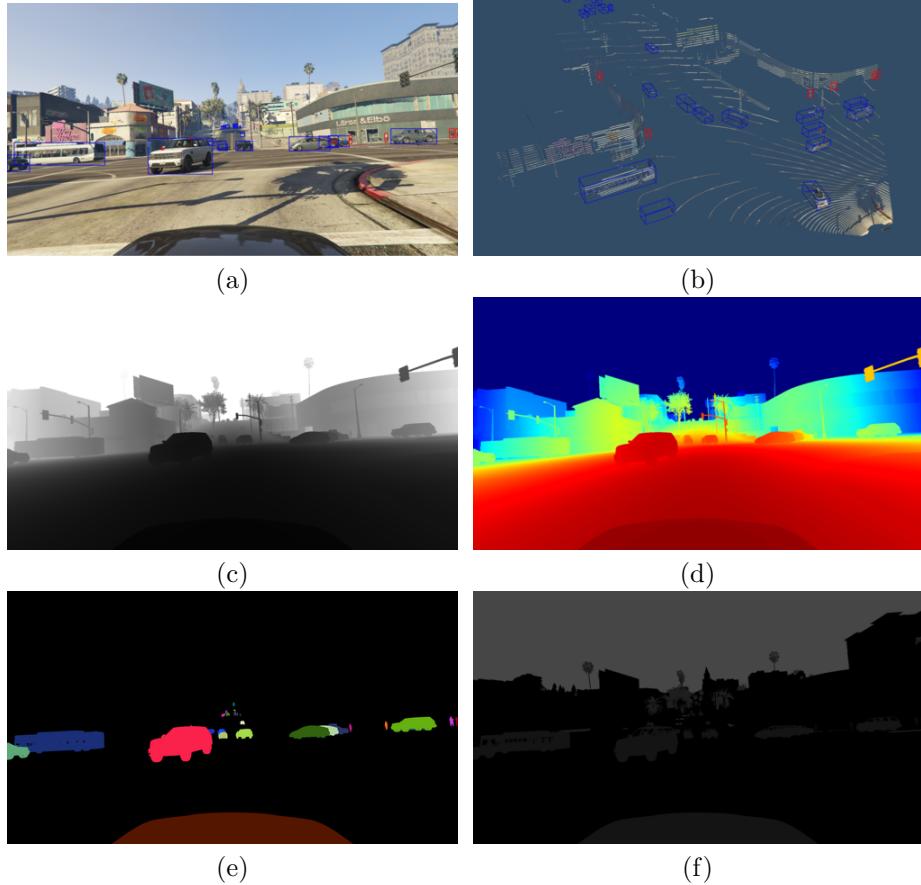


Figure 12: Sample images from the *PreSIL* dataset [269]: (a) RGB image, (b) point cloud, (c) black-and-white depth map, (d) color depth map, (e) segmentation map, (f) stencil buffer.

well. Thieling et al. [584] discuss the issues of physically realistic simulation for various robot sensors. Yue et al. [679] present a LIDAR point cloud generator based on the *Grand Theft Auto V* engine, showing significant improvements in point cloud segmentation when augmenting the KITTI dataset with their synthetic data. Sanchez et al. [572] generate synthetic 3D point clouds with the robotic simulator *Gazebo* (see Section 3.3). Wang et al. [619] develop a separate open source plugin for LIDAR point cloud generation. Fang et al. [173] present an augmented LIDAR point cloud simulator that can generate simulated point clouds from real 3D scanner data, extending it with synthetic objects (additional cars). The work based on GTA V has recently been continued by Hurl et al. [269] who have developed a precise LIDAR simulator within the GTA V engine and published the *PreSIL* (Precise Synthetic Image and LIDAR) dataset with over 50000 frames with depth information, point clouds, semantic segmentation, and detailed annotations; we use *PreSIL* to showcase on Fig. 12 the modalities available in modern synthetic datasets. There are also works on synthesizing specific elements of the environment, thus augmenting real data

with synthetic elements; for example, Bruls et al. [73] generate synthetic road marking layouts which improves road marking segmentation, especially in corner cases.

Outdoor simulated environments go beyond driving, however, with simulators and synthetic datasets successfully used for autonomous aerial vehicles. Most of them are intended for unmanned aerial vehicles (UAVs) and have an emphasis on plugging in robotic controllers, possibly even with hardware-in-the-loop approaches; we discuss these simulators in Sections 3.3 and 3.4.

3.2 Datasets and simulators of indoor scenes

Although, as we have seen in the previous section, the main emphasis of many influential applications remains in the outdoors, *indoor navigation* is also an important field where synthetic datasets are required. The main problems remain the same—SLAM and navigation—but the potential applications are now more in the field of home robotics, industrial robots, and embodied AI [398, 712]. There are large-scale efforts to create real annotated datasets of indoor scenes [92, 133, 548, 556, 652, 654], but synthetic data is increasingly being used in the field [693].

Currently, the main synthetic dataset for indoor navigation is SUNCG⁶ presented by Song et al. [557]. It contains over 45,000 different scenes (floors of private houses) with manually created realistic room layouts, 3D models of the furniture, realistic textures, and so on. All of the scenes are semantically annotated at the object level, and the dataset provides synthetic depth maps and volumetric ground truth data for the scenes. The original paper [557] presented state of the art results in semantic scene completion, but, naturally, SUNCG has been used for many different tasks related to depth estimation, indoor navigation, SLAM, and others [2, 108, 231, 385, 465], and it often serves as the basis for scene understanding competitions [566].

Interestingly, at the time of writing this survey the SUNCG website was down and the dataset itself unavailable due to a legal controversy over the data⁷; that is why Fig. 13c shows a standard showcase picture from [557] instead of data samples. While synthetic data can solve a lot of legal issues with real data (see Section 7 for a discussion of privacy concerns, for example), the data, and especially handmade or manually collected 3D models, are still intellectual property and can bring about problems of its own unless properly released to the public domain.

SUNCG has given rise to a number of simulation environments. Before SUNCG, we note the *Gazebo* platform mentioned above [324] (see also Section 3.3) and the V-REP robot simulation framework [496] that not only provided visual information but also simulated a number of actual robot types, further simplifying control deployment and development. Handa et al. [233] provide their own simulated living room environment and dataset ICL-NUIM (Imperial College London and National University of Ireland Maynooth) with special emphasis on visual odometry and SLAM; they render high-quality RGB-D images with ray tracing and take special care to model the noise in both depth and RGB channels.

⁶<http://suncg.cs.princeton.edu/>

⁷<https://futurism.com/tech-suing-facebook-princeton-data>

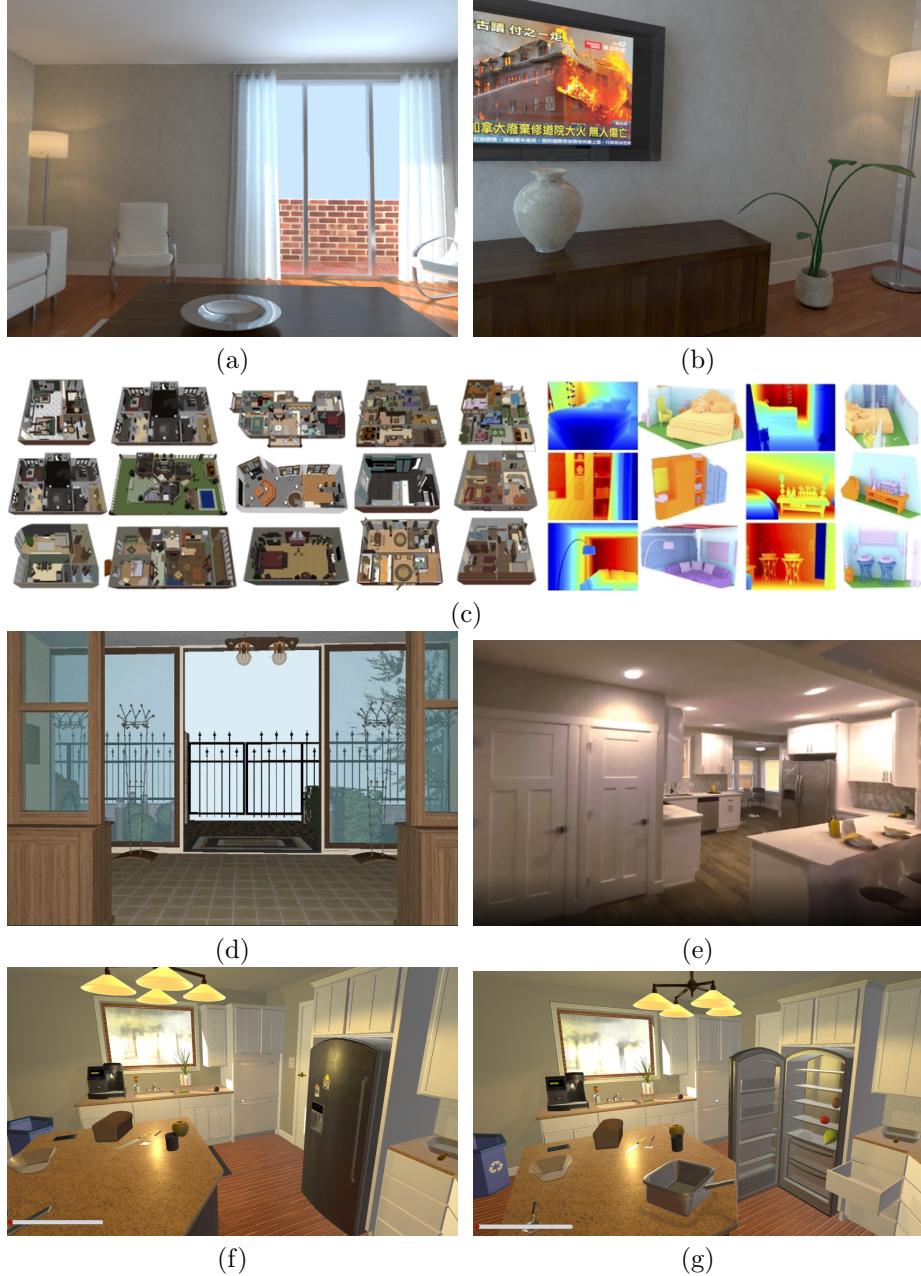


Figure 13: Sample images from indoor datasets and simulation environments: (a-b) ICL-NUIM [233]; (c) SUNCG [557]; (d) House3D [647]; (e) Habitat [393]; (f-g) AI2THOR [325].

MINOS by Savva et al. [521] is a multimodal simulator for indoor navigation (later superceded by Habitat, see below). Wu et al. [647] made SUNCG into a full-scale simulation environment named House3D⁸, with high-speed rendering

⁸<http://github.com/facebookresearch/House3D>

suitable for large-scale reinforcement learning. In House3D, a virtual agent can freely explore 3D environments taken from SUNCG while providing all the modalities of SUNCG. House3D has been famously used for navigation control with natural language: Wu et al [647] presented *RoomNav*, a task of navigation from natural language instructions and some models able to do it, while Das et al. [139] presented an embodied question answering model, where a robot is supposed to answer natural language questions by navigating an indoor environment. The AI2-THOR (The House Of inteRactions) framework [325] provides near photorealistic interactive environments with actionable objects (doors that can be opened, furniture that can be moved etc.) based on the *Unity3D* game engine (see example of the same scene after some actions applied to objects on Fig. 13f-g).

Zhang et al. [701] studied the importance of synthetic data realism for various indoor vision tasks. They fixed some problems with 3D models from SUNCG, improved their geometry and materials, sampled a diverse set of cameras for each scene, and compared OpenGL rendering against physically-based rendering (with Metropolis light transport models [613] and the Mitsuba renderer⁹) across a variety of lighting conditions. Their main conclusion is that, again, added realism is worth the effort: the quality gains are quite significant.

At the time of writing, the last (and very recent) major advance in the field is *Habitat*, a simulation platform for embodied AI developed by Facebook researchers Savva et al. [393]. Its simulator, called *Habitat-Sim*, presents a number of important improvements over previous work that we have surveyed in this section:

- dataset support: *Habitat-Sim* supports both synthetic datasets such as SUNCG [557] and real-world datasets such as Matterport3D [92] and Gibson [652];
- rendering performance: *Habitat-Sim* can render thousands of frames per second, 10-100x faster than previous simulators; the authors claim that “it is often faster to generate images using *Habitat-Sim* than to load images from disk”; this is important because simulation stops being a bottleneck in large-scale model training;
- humans-as-agents: humans can function as agents in the simulated environment, which allows to use real human behaviour in agent training and evaluation;
- accompanying library: the *Habitat-API* library defines embodied AI tasks and implements metrics for easy agent development.

Savva et al. also provide a large-scale experimental study of various state of the art agents, arriving at the conclusion (counter to previous research) that reinforcement learning-based agents outperform SLAM-based ones, and RL agents generalize best across datasets, including the synthetic-to-real generalization from SUNCG to Matterport3D and Gibson. This is an important finding for synthetic data in indoor navigation, and we expect it to be confirmed in later studies; moreover, we expect *Habitat* and its successors to become the new standard for indoor navigation and embodied AI research.

⁹<http://www.mitsuba-renderer.org/>

3.3 Robotic simulators

We have seen autonomous driving sims that mostly concentrate on accurately reflecting the outside world, modeling additional sensors such as LIDAR, and physics of the driving process. Simulators for indoor robots and unmanned aerial vehicles (UAV) add another complication: embedded hardware for such robots may be relatively weak and needs to be taken into account. Hence, robotic simulators usually support the *Robot Operating System*¹⁰ (ROS), a common framework for writing robot software. In some cases, simulators go as far as provide *hardware-in-the-loop* capabilities, where a real hardware controller can be plugged into the simulator; for example, hardware-in-the-loop approaches to testing UAVs have been known for a long time and represent an important methodology in flight controller development [7, 461, 618]. We also refer to the surveys [244, 389].

For a brief review, we highlight four works, starting with two standard references. *Gazebo*, originally presented by Koenig and Howard [324] and now being developed by OSRF (Open Source Robotics Foundation), is probably the best-known robotic simulation platform. It supports ROS integration out of the box, has been used in the DARPA Robotics Challenge [9], NASA Space Robotics Challenge, and others, and has been instrumental for thousands of research and industrial projects in robotics. *Gazebo* uses a realistic physical engine (actually, several different engines) that supports illumination and lighting effects, gravity, inertia, and so on; it can be integrated with robotic hardware via ROS and provides realistic simulation that often leads to successful transfer to the real world.

MuJoCo (Multi-Joint Dynamics with Contact) developed by Todorov [589] is a physics engine specializing on contact-rich behaviours, which abound in robotics. Both *Gazebo* and *MuJoCo* have become industry standards for robotics research, and surveying the full range of their applications goes far beyond the scope of this work. There are, of course, other platforms as well. For example, Gupta and Jarvis [223] present a simulation platform for training mobile robots based on the *Half-Life 2* game engine.

The other two works are, on the contrary, very recent and may well define a new industry standard in the near future. First, Xie et al. present *VRGym* [658], a virtual reality testbed for physical and interactive AI agents. Their main difference from previous work is the support of human input via VR hardware integration. The rendering and physics engine are based on *Unreal Engine 4*, additional multi-sensor hardware is capable of full body sensing and integration of human subjects to virtual environments, while a ROS bridge allows to easily communicate with robotic hardware. Xie et al. benchmark RL algorithms and show possibilities for socially aware models and intention prediction; *VRGym* is already being further extended into, e.g., *VRKitchen* by Gao et al. [192] designed to learn cooking tasks.

The second work is the *OpenAI Remote Rendering Backend* (ORRB) developed by OpenAI researchers Chociej et al. [112] able to render robotic environments and provide depth and segmentation maps in its renderings. ORRB emphasizes diversity, aiming to provide domain randomization effects (see Section 5.1); it is based on *Unity3D* for rendering and *MuJoCo* for physics, and it supports distributed cloud environments. ORRB has been used to train

¹⁰<https://www.ros.org/>

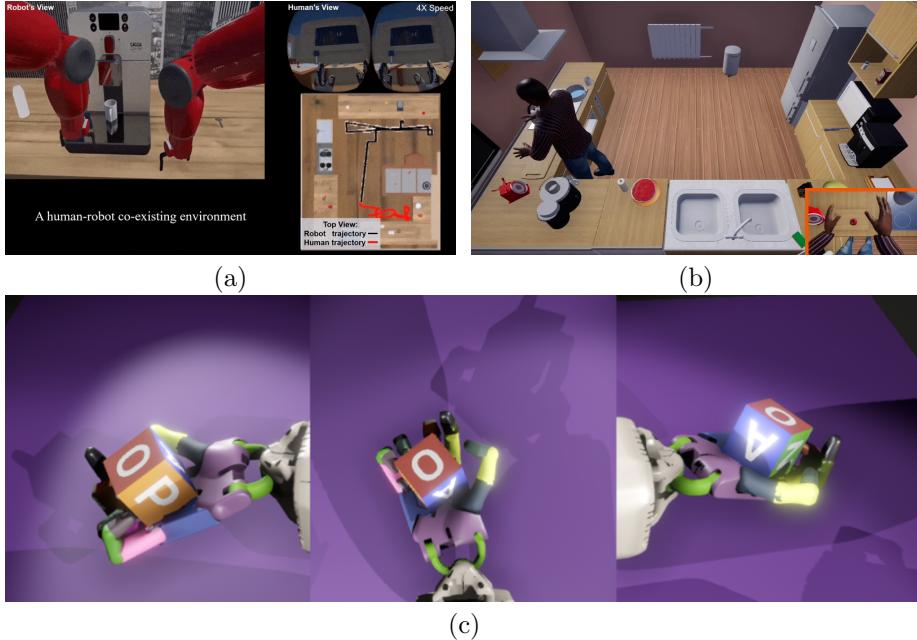


Figure 14: Sample images from robotic simulation environments: (a) *VR-Gym* [658]; (b) *VRKitchen* [192]; (c) *ORRB* [112].

Dactyl [432], a robotic hand for multi-finger small object manipulation developed by OpenAI; *Dactyl*'s RL-based policies have been trained entirely in ORRB simulation and then have been successfully transferred to the physical robot. We expect more exciting developments in such synthetic-to-real transfer for robotic applications in the near future.

3.4 Vision-based applications in unmanned aerial vehicles

A long line of research deals with vision-based approaches to operating unmanned aerial vehicles (UAV) [11]. Since in this case it is almost inevitable to use simulated environments for training, and often testing the model is also restricted to synthetic simulators (real world experiments are expensive outdoors and simply prohibited in urban environments), almost the entire field uses some kind of synthetic datasets or simulators. Classical vision-related problems for UAVs include three major tasks that UAVs often solve with computer vision:

- localization and pose estimation, i.e., estimating the UAV position and orientation in both 2D (on the map) and in the 3D space; for this problem, real datasets collected by real world UAVs are available [67, 495, 536, 561], and the field is advanced enough to use actual field tests, so synthetic-only results are viewed with suspicion, but existing research still often employs synthetic simulators, either handmade for a specific problem [635] or based on professional flight simulators [25];
- obstacle detection and avoidance, where real-world experiments are often too expensive even for testing [80];

- visual servoing, i.e., using feedback from visual sensors in order to maintain position, stability, and perform maneuvers; here synthetic data has usually been used in the form of hardware-in-the-loop simulators for the developed controllers [348], sometimes augmented with full-scale flight simulators [336] or specially designed “virtual reality” environments [523].

During the latest years, these classical applications of synthetic data for UAVs have been extended and taken to new heights with modern approaches to synthetic data generation. For example, in [395] the problem is to locate safe landing areas for UAVs, which requires depth estimation and segmentation into “horizontal”, “vertical”, and “uncertain” regions to distinguish horizontal areas that would be safe for landing. To train a convolutional architecture for this segmentation and depth estimation task, the authors propose an interesting approach to generating synthetic data: they begin with *Google Earth* data¹¹ and extract 3D scenes from it. However, since 3D meshes in *Google Earth* are far from perfect, the authors then map textures to the 3D scenes to obtain less realistic-looking images but ones for which the depth maps are known perfectly. The authors show that from a bird’s eye view the resulting images look quite realistic, and compare different segmentation architectures on the resulting synthetic dataset. Oyuki Rojas-Perez et al. [497] also compared the results obtained by training on synthetic and real datasets, with the results in favor of synthetic data due to the availability of depth maps for synthetic images. In a recent work, Castagno et al. [88] solve the landing site selection problem with a high-fidelity visual synthetic model of Manhattan rooftops, rendered with the *Unreal Engine* and provided to the simulated robots via *AirSim*.

Gazebo has been used for UAV simulations [183], but there are more popular specialized simulators in the field. *AirSim* [539] by *Microsoft* is a flight simulator that operates more like a robotic simulator such as *Gazebo* than an accurate flight sim such as *Microsoft Flight Simulator* or *X-Plane*: it contains a detailed physics engine designed to interact with specific flight controllers and providing a realistic physics-based vehicle model but also supports ROS to interact with the drones’ software and/or hardware. Apart from visualizations rendered with *Unreal Engine 4*, *AirSim* provides sensor readings, including a barometer, gyroscope, accelerometer, magnetometer, and GPS. *AirSim* is also available as a plugin for *Unreal Engine 4* which enables extensions and new projects based on *AirSim* simulations of autonomous vehicles. There are plenty of extensions and projects that use *AirSim* and provide interesting synthetic simulated environments (see also the survey [389]), in particular:

- Chen et al. [103] add realistic forests to use UAVs for forest visual perception;
- Bondi et al. [66] concentrate on wildlife preservation, extending the engine with realistic renderings of, e.g., the African savanna;
- in two different projects, Smyth et al. [552, 553] and Ullah et al. [606] simulate critical incidents (such as chemical, biological, or nuclear incidents or attacks) with an explicit goal of training autonomous drones to collect data in these virtual environments;

¹¹<https://www.google.com/earth/>

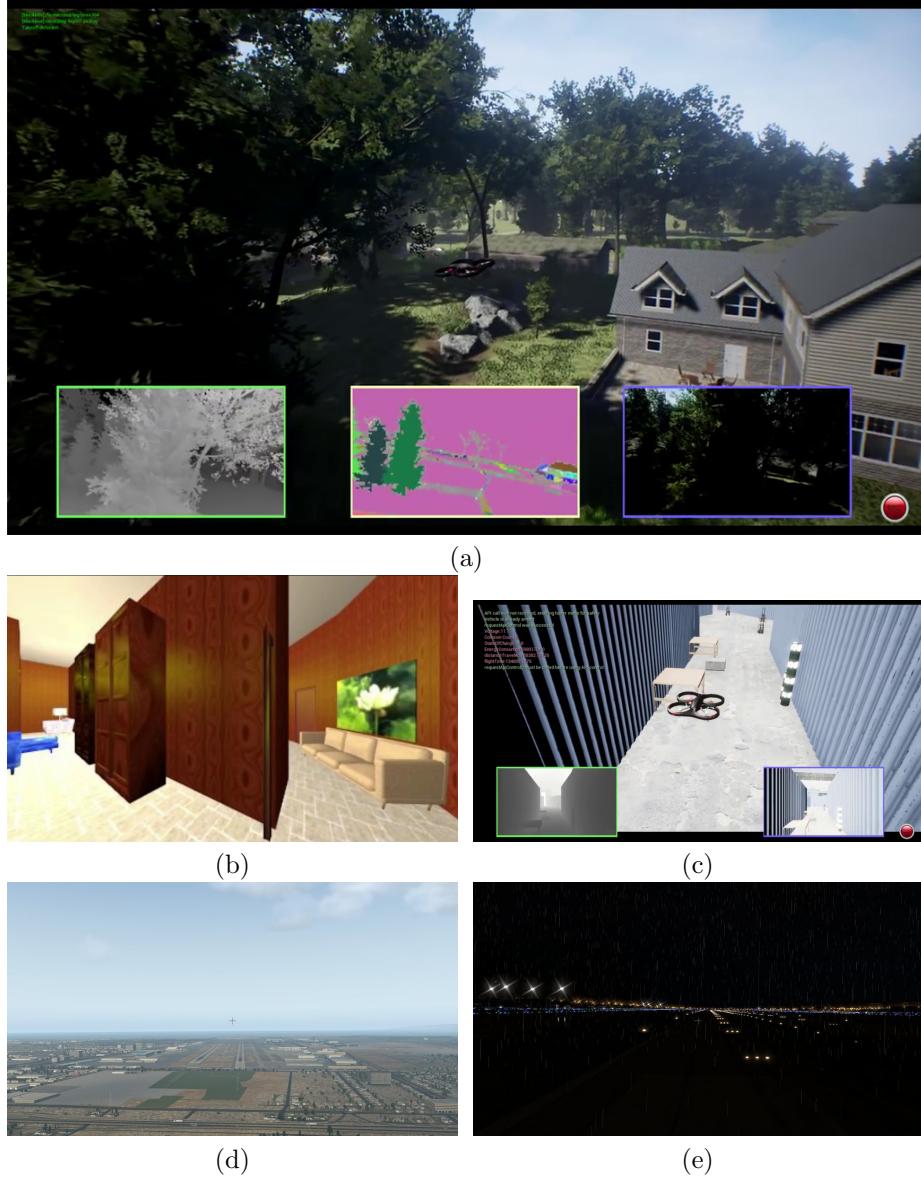


Figure 15: Sample images from flight simulators: (a) *AirSim* [539] drone demo with depth, segmentation, and RGB drone view on the bottom; (b) *CAD²RL* [513]; (c) *Air Learning* [332] (depth image and drone camera view on the bottom); (d-e) *XPlane* dataset [555].

- Huang et al. [260] extend *AirSim* with a natural language understanding model to simulate natural language commands for drones, and so on.

We also note that *Microsoft* itself recently extended *AirSim* to include autonomous car simulations, making the engine applicable to all the tasks we discussed in Section 3.1.

While *AirSim* and *Gazebo* have been the most successful simulation frame-

works, we also note other frameworks for multi-agent simulation that have been used for autonomous vehicles, usually without a realistic 3D rendered environment (see also a comparison of the frameworks in [420] and of their perception systems in [503]): JaSIM [188], a multiagent 3D environment model based on the *Janus* platform [194], *Repast Simphony* [425], an agent-based simulation library for complex adaptive systems, FLAME [321] that concentrates on parallel simulations, and JADE [48], a popular library for multi-agent simulations.

Sadeghi and Levine [513] present the CAD²RL framework for avoiding collisions while flying in indoor environments. They motivate the use of synthetic data by the domain randomization idea (see Section 5.1) and produce a wide variety of indoor simulated environments constructed in *Blender*. They do not use any real images during training, learning the Q-function for the reinforcement learning agent entirely on simulated data, with a fully convolutional neural network. The authors report improvements in a number of complex settings such as flying around corners, navigating through narrow corridors, flying up a staircase, avoiding dynamic obstacles, and so on.

The *Air Learning* platform recently presented by Krishnan et al. [332] is an end-to-end simulation environment for autonomous aerial robots that combines pluggable environment and physics engine (such as *AirSim* or *Gazebo*), learning algorithms, policies for robot control (implemented in, e.g., *TensorFlow* or *PyTorch*), and “hardware-in-the-loop” controllers, where a real flight controller can be plugged in and evaluated on the *Air Learning* platform. As an example, Krishnan et al. benchmark several reinforcement learning approaches to point-to-point obstacle avoidance tasks and arrive at important conclusions: for instance, it turns out that having more onboard compute (a desktop CPU vs. a *Raspberry Pi*) can significantly improve the result, producing almost 2x shorter trajectories.

Among vision-based synthetic datasets, we note the work by Solovev et al. [555] who present a synthetic dataset for airplane landing based on the XPlane flight simulator [342], which has been used for other UAV simulations as well [60, 193]. The main intention of Solovev et al. is to present a benchmark for representation learning by combining different modalities (RGB images and various sensors), but the resulting synthetic dataset is sufficiently large to be used for other applications: 93GB of images and sensor readings from 8K landings on 114 different runways.

Another interesting application where real and synthetic data come together is provided by Madaan et al. [386], who solve a truly life-or-death problem for UAVs: wire detection. They use real background images and superimpose them with renderings of realistic 3D models of wires. The authors vary different properties of the wires (material, wire sag, camera angle etc.) but do not make any attempts to adapt the wires to the semantics of the background image itself, simply pasting wires onto the images. Nevertheless, Madaan et al. report good results with training first on synthetic data and then fine-tuning on a small real dataset; synthetic pretraining proves to be helpful.

3.5 Computer games as virtual environments

Computer games and game engines have been a very important source of problems and virtual environments for deep RL and AI in general [85, 296, 526]. Much of the foundational work in modern deep RL has been done in the en-

vironments of 2D arcade games, usually classical *Atari* games [47, 414]. First, many new ideas for RL architectures or training procedures for robotic navigation were introduced as direct applications of deep RL to navigation in complex synthetic environments, in particular to navigating mazes in video games such as *Doom* [56] and *Minecraft* [429] or specially constructed 3D mazes, e.g., from the *DeepMind Lab* environment [44, 537]. *Doom* became something of an industry standard, with the *VizDoom* framework developed by Kempka et al. [314] and later used in many important works [15, 343, 480, 648].

Games of other genres, in particular real-time strategy games [18, 580], also represent a rich source of synthetic environments for deep RL; we note the *TorchCraft* library for machine learning research in *StarCraft* [569], synthetic datasets extracted from *StarCraft* [364, 410, 645], the *ELF* (Extensive, Lightweight, and Flexible) library platform with three simplified real-time strategy games [586], and the SC2LE (StarCraft II Learning Environment) library for *Starcraft II* [615]. Racing games have been used as environments for training end-to-end RL driving agents [451]. While data from computer games could technically be considered synthetic data, we do not go into further details on game-related research and concentrate on virtual environments specially designed for machine learning and/or transfer to real world tasks. Note, however, that synthetic data is already making inroads even into learning to play computer games: Justesen et al. [297] show that using procedurally generated levels improves generalization and final results for *Atari* games and can even produce models that work well when evaluated on a completely new level every time they play.

All of the above does not look too much in line with the general topic of synthetic data: while game environments are certainly “synthetic”, there is usually no goal to transfer, say, an RL agent playing *StarCraft* to a real armed conflict (thankfully). However, recent works suggest that there is potential in this direction. For example, while navigation in a first-person shooter is very different from real robotic navigation, successful attempts at transfer learning from computer games to the real world are already starting to appear. Karttunen et al. [313] present an RL agent for navigation trained on the *Doom* environment and transferred to a real life Turtlebot by freezing most of the weights in the DQN network and only fine-tuning a small subset of them. As computer games get more realistic, we expect such transfer to become easier.

4 Synthetic data outside computer vision

While computer vision remains the main focus of synthetic data applications, other fields also begin to use synthetic datasets, with some directions entirely dependent on synthetic data. In this section, we survey some of these fields. Specifically, in Section 4.1 we consider neural programming, in Section 4.2 discuss synthetic data generation and use in bioinformatics, and Section 4.3 reviews the (admittedly limited) applications of synthetic data in natural language processing.

4.1 Synthetic data for neural programming

One interesting domain where synthetic data is paramount is neural program synthesis and neural program induction. The basic idea of teaching a machine learning model to program can be broadly divided into two subfields: program induction aims to train an end-to-end differentiable model to capture an algorithm [143], while program synthesis tries to teach a model the semantics of a domain-specific language (DSL), so that the model is able to generate programs according to given specifications [308]. Basically, in program induction the network *is* the program, while in program synthesis the network *writes* the program. Naturally, both tasks require large datasets of programs together with their input-output pairs; since no such large datasets exist, and generating synthetic programs and running them in this case is relatively easy (arguably even easier than generating synthetic data for computer vision), all modern works use synthetic data to train the “neural computers”.

In program induction, the tasks are so far relatively simple, and synthetic data generation does not present too many difficulties. For example, Joulin and Mikolov [294] present a new architecture (stack-augmented recurrent networks) to learn regularities in algorithmically generated sequences of symbols; the training data, as in previous such works [200, 247, 251, 637], is synthetically generated by hand-crafted simple algorithms, including generating sequences from a pattern such as $a^n b^{2n}$ or $a^n b^m c^{n+m}$, binary addition (supervised problem asking to continue a string such as $110 + 10 =$), and similar. Zaremba and Sutskever [683] train a model to execute programs, i.e., map their textual representations to outputs; they generate training data as Python-style programs with addition, subtraction, multiplication, variable assignments, if-statements, and for-loops (but without nested loops), each ending with a `print` statement; see Fig. 16 for an illustration. Neural RAM machines [335] are trained on a number of simple tasks (access, increment, copy etc.) whose specific instances are randomly synthesized. The same goes for neural Turing machines [210] and neural GPUs [179, 300]: they are trained and evaluated on synthetic examples generated for simple basic problems such as copying, sorting, or arithmetic operations.

In neural program synthesis, again, the programs are usually simple and also generated automatically; attempts to collect natural datasets for program synthesis have begun only very recently [684]. Learning-based program synthesis (earlier attempts were based on logical inference [392]) began with learning string manipulation programs [218] and soon branched into deep learning, using recurrent architectures to guide search-based methods [614] or to generate programs in encoder-decoder architectures [106, 146]. In [76, 546], reinforcement

Input:

```
j=8584
for x in range(8):
    j+=920
    b=(1500+j)
    print((b+7567))
```

Target: 25011.**Input:**

```
i=8827
c=(i-5347)
print((c+8704) if 2641<8500 else 5308)
```

Target: 12184.

Figure 16: Sample synthetic programs from [683].

learning is added on top of a recurrent architecture in order to alleviate the *program aliasing* problem, i.e., the fact that many different equivalent programs provide equally correct answers while the training set contains only one. All of the above-mentioned models were trained on synthetic datasets of randomly generated programs (usually in the form of abstract syntax trees) run on randomized sets of inputs.

As a separate thread, we mention works on program synthesis for visual reasoning and, generally speaking, question answering [258, 289, 519]. To do visual question answering [8], models are trained to compose a short program based on the natural language question that will lead to the answer, usually in the form of an execution graph or a network architecture. This line of work is based on neural module networks [21, 91] and similar constructions [22, 259], where the network learns to create a composition of modules (in QA, based on parsing the question) that are also neural networks, all learned jointly (see, however, the critique in [533]). Latest works use architectures based on self-attention that have proven their worth across a wide variety of NLP tasks [351]. Naturally, most of these works use the CLEVR synthetic dataset for evaluation (see Section 2.5).

Generation of synthetic data itself has not been given much attention in this field until very recently, but the interest is rising. The work by Shin et al. [542] presents a study of contemporary synthetic data generation techniques for neural program synthesis and concludes that the resulting models do not capture the full semantics of the language, even if they do well on the test set. Shin et al. show that synthetic data generation algorithms, including the one from the popular *tensor2tensor* library [610], have biases that fail to cover important parts of the program space and deteriorate the final result. To fix this problem, they propose a novel methodology for generating distributions over the space of datasets for program induction and synthesis, showing significant improvements for two important domains: *Calculator* (which computes the results of arithmetic expressions) and *Karel* (which achieves a given objective with a virtual robot moving on a two-dimensional grid with walls and markers). We expect more research into synthetic data generation for neural program induction and synthesis to follow in the near future.

4.2 Synthetic data in bioinformatics

We use examples from the healthcare and biomedical domain throughout this survey; see, e.g., Sections 6.4 and 7.3. In this section, we concentrate on ap-

plications of synthetic data in bioinformatics that fall outside either producing synthetic medical images (usually through GANs, see Section 6.4) or providing privacy guarantees for sensitive data through synthetic datasets (Section 7.3). It turns out that there are still plenty, and synthetic data is routinely used and generated throughout bioinformatics; see also a survey in [111].

For many of these methods, generated synthetic data is the end goal rather than a tool to improve machine learning models. In particular, *de novo* drug design [235, 528] is a field that searches for molecules with desirable properties in a search space of about 10^{60} synthesizable molecules [195, 492], and the goal is to find (which in a space of this size rather means to *generate*) candidate molecules that would later have to be explored further in lab studies and then clinical trials. First modern attempts at *de novo* drug design used rule-based methods that simulate chemical reactions [528], but the field soon turned to generative models, in particular based on deep learning [99, 196]. In this context, molecules are usually represented in the SMILES format [636] that encodes molecular graphs as strings in a certain formal grammar, which makes it possible to use sequence learning models to generate new SMILES strings. Segler et al. [530] used LSTM-based RNNs to learn a chemical language model, while Gómez-Bombarelli et al. [205] trained a variational autoencoder (VAE) which is already a generative model capable of generating new candidate molecules. To further improve generation, Kusner et al. [338] developed a novel extension of VAEs called Grammar Variational Autoencoders that can take into account the rules of the SMILES formal grammar and make VAE outputs conform to the grammar. To then use the models to obtain molecules with desired properties, researchers used either a small set of labeled positive examples [530] or augment the RNN training procedure with reinforcement learning [280]. In particular, Olivecrona et al. [430] use a recurrent neural network trained to generate SMILES representations: first, a prior network (3 layers of 1024 GRU) is trained in a supervised way on the RDKit subset [344] of ChEMBL database [195], then an RL agent (with the same structure, initialized from the prior network) is fine-tuned with the REINFORCE algorithm to improve the resulting SMILES encoding. A similar architecture, but with a stack-augmented RNN [295] as the basis, which enables more long-term dependencies, was presented by Popova et al. [459]. We note a series of works by *Insilico* researchers Kadurin, Polykovskiy and others who applied different generative models to this problem:

- the work [298] trains a supervised adversarial autoencoder [390] with the condition (in this case, growth inhibition percentage for tumor cells after treatment) added as a separate neuron to the latent layer;
- in [299], Kadurin et al. compared adversarial autoencoders (AAE) with variational autoencoders (VAE) for the same problem, with new modifications to the architecture that result in improved generation;
- in [457], Polykovskiy et al. introduced a new AAE modification, entangled conditional adversarial autoencoder, to ensure the disentanglement of latent features; in this case, which is still quite rare for deep learning in drug discovery, a newly discovered molecule (a new inhibitor of Janus kinase 3) was actually tested in the lab and showed good activity and selectivity *in vitro*;

- Kuzmynikh et al. [339] presented a novel 3D molecular representation based on the wave transform that led to improved performance for CNN-based autoencoders and improved MACCS fingerprint prediction;
- Polykovskiy et al. [456] presented MOSES (Molecular Sets), a benchmarking platform for molecular generation models, which implemented and compared various generative models for molecular generation including CharRNN [531], VAE, AAE, and Junction Tree VAE [283], together with a variety of evaluation metrics for generation results.

The above works can be thought of as generation of synthetic data (e.g., molecular structures) that could be of direct use for practical applications.

Johnson et al. [286] undertake an ambitious project: they learn a generative model of the variation in cell and nuclear morphology based on fluorescence microscopy images. Their model is based on two adversarial autoencoders [390], one learning a probabilistic model of cell and nuclear shape and the other learning the interrelations between subcellular structures conditional on an encoding of the cell and nuclear shape from the first autoencoder. The resulting model produces plausible synthetic images of the cell with known localizations of subcellular structures.

One interesting variation of “synthetic data” in bioinformatics concerns learning from live experiments on synthetically generated biological material. For example, Rosenberg et al. [502] study alternative RNA splicing, in particular the functional effects of genetic variation on the molecular phenotypes through alternative splicing. To do that, they create a large-scale gene library with more than two million randomly generated synthetic DNA sequences, then used massively parallel reporter assays (MPRA) to measure the isoform ratio for all mini-genes in the experiment, and then used it to learn a (simple linear) machine learning model for alternative splicing. It turned out that this approach significantly improved prediction quality, outperforming state of the art deep learning models for alternative splicing trained on the actual human genome [660] in predicting the results of *in vivo* experiments. There is also a related field of imitational modeling for bioinformatics data that often results in realistic synthetic generators; e.g., Van den Bulcke et al. [608] provide a generator for synthetic gene expression data, able to produce synthetic transcriptional regulatory networks and simulated gene expression data while closely matching real statistics of biological networks.

4.3 Synthetic data in natural language processing

Synthetic data has not been widely used in natural language processing (NLP). In our opinion, there is a conceptual reason for this. Compare with computer vision: there, the process of synthetic data generation can be done separately from learning the models, and the essence of what the models are learning is, in a way, “orthogonal” to the difference between real and synthetic data. If I show you a cartoon ad featuring a new bottle of soda, you will be able to find it in a supermarket even though you would never confuse the cartoon with a real photo. In natural language processing, on the other hand, text generation is the hard problem itself. The problem of generating meaningful synthetic text with predefined target variables such as topic or sentiment is the subject of many studies in NLP, we are still a long way to go before it is solved, and it

is quite probable that when text generation finally reaches near-human levels, discriminative models for the target variables will easily follow from it, rendering synthetic data useless.

Nevertheless, there have been works that use data augmentation for NLP in a fashion that borders on using synthetic data. There have been simple augmentation approaches such as to simply drop out certain words [535]. A development of this idea shown in [631] switches out certain words, replacing them with random words from the vocabulary. The work [659] develops methods of data noising for language models, adding noise to word counts in a way reminiscent of smoothing in language models based on n -grams.

A more developed approach is to do data augmentation with synonyms: to expand a dataset, one can replace words with their synonyms, getting “synthetic sentences” that can still preserve target variables such as the topic of the text, its sentiment, and so on. The work [695] used this method directly to train a character-level network for text classification, while [187] tested augmentation with synonyms for morphology-rich languages such as Russian. In [172], augmentation with synonyms was used for low-resource machine translation, with an auxiliary LSTM-based language model used to recognize whether the synonym substitution is correct. The work [629], which concentrated on studying tweets, proposed to use embedding-based data augmentation, using neighboring words in the word vector space as synonyms. The work [322] extends augmentation with synonyms by replacing words in sentences with other words in paradigmatic relations with the original words, as predicted by a bi-directional language model at the word positions.

Techniques for generating synthetic text are constantly evolving. First, modern language models based on multi-head self-attention from the Transformer family, starting from the Transformer itself [611] and then further developed by BERT [145], OpenAI GPT [473], Transformer-XL [135], OpenAI GPT-2 [474], and GROVER [685], generate increasingly coherent text. Actually, Zellers et al. [685] claim that their GROVER model for conditional generation (e.g., generating the text of a news article given its title, domain, and author) outperforms human-generated text in the “fake news” / “propaganda” category in terms of style and content (evaluated by humans).

Moreover, recently developed models allow to generate text with GANs. This is a challenging problem because unlike, say, images, text is discrete and hence the generator output is not differentiable. There are several approaches to solving this problem:

- training with the REINFORCE algorithm and other techniques from reinforcement learning that are able to handle discrete outputs; this path has been taken in the pioneering model named SeqGAN [678], LeakGAN for generating long text fragments [219], and MaskGAN that learns to fill in missing text with an actor-critic conditional GAN [174], among others;
- approximating discrete sampling with a continuous function; these approaches include the Gumbel Softmax trick [337], TextGAN that approximates the arg max function [702], TextKD-GAN that uses an autoencoder to smooth the one-hot representation into a softmax output [226], and more;
- generating elements of the latent space for an autoencoder instead of di-

rectly generating text; this field started with adversarially regularized autoencoders by Zhao et al. [709] and has been extended into text style transfer by disentangling style and content in the latent space [285], disentangling syntax and semantics [36], DialogWAE for dialog modeling [215], Bilingual-GAN able to generate parallel sentences in two languages [479], and other works.

This abundance of generative models for text has not, however, led to any significant use of synthetic data for training NLP models; this has been done only in very restricted domains such as electronic medical records [216] (we discuss this field in detail in Section 7.3). We have suggested the reasons for this in the beginning of this section, and so far the development of natural language processing supports our view.

5 Directions in synthetic data development

In this section, we outline the main directions that intend to further improve synthetic data, making it more useful for a wide variety of applications in computer vision and other fields. In particular, we discuss the idea of domain randomization (Section 5.1) intended to improve the applications of synthetic datasets, methods to improve CGI-based synthetic data generation itself (Section 5.2), ways to create synthetic data from real images by cutting and pasting (Section 5.3), and finally possibilities to produce synthetic data by generative models (Section 5.4). The latter means generating useful synthetic data from scratch rather than domain adaptation and refinement, which we consider in a separate Section 6.

5.1 Domain randomization

Domain randomization is one of the most promising approaches to make straightforward transfer learning from synthetic data to real actually work. Consider a model that is supposed to train on $D_{\text{syn}} \sim p_{\text{syn}}$ and later be applied to $D_{\text{real}} \sim p_{\text{real}}$. The basic idea of domain randomization had been known since the 1990s [278] but was probably first explicitly presented and named in [588]. The idea is simple: let us try to make the synthetic data distribution p_{syn} sufficiently wide and varied so that the model trained on p_{syn} will be robust enough to work well on p_{real} .

Ideally, we would like to cover p_{real} with p_{syn} , but in reality this is never achieved directly. Instead, synthetic data in computer vision can be randomized and made more diverse in a number of different ways at the level of either constructing a 3D scene or rendering 2D images from it:

- at the scene construction level, a synthetic data generator (SDG) can randomize the number of objects, its relative and absolute positions, number and shape of distractor objects, contents of the scene background, textures of all objects participating in the scene, and so on;
- at the rendering level, SDG can randomize lighting conditions, in particular the position, orientation, and intensity of light sources, change the rendering quality by modifying image resolution, rendering type such as ray tracing or other options, add random noise to the resulting images, and so on.

The work [588] did the first steps to show that domain randomization works well; they used simple geometric shapes (polyhedra) as both target and distractor objects, random textures such as gradient fills or checkered patterns. The authors found that synthetic pretraining is indeed very helpful when only a small real training set is available, but helpful only if sufficiently randomized, in particular when using a large number of random textures. This approach was subsequently applied to a more ambitious domain by NVIDIA researchers Tremblay et al. [595], who trained object detection models on synthetic data with the following procedure:

- create randomized 3D scenes, adding objects of interest on top of random surfaces in the scenes;

- add so-called “flying distractors”, diverse geometric shapes that are supposed to serve as negative examples for object detection;
- add random textures to every object, randomize the camera parameters, lighting, and other parameters.

The resulting images were completely unrealistic, yet diverse enough that the networks had to concentrate on the shape of the objects in question. Tremblay et al. report improved car detection results for R-FCN [134] and SSD [372] architectures (but failing to improve Faster R-CNN [489]) on their dataset compared to Virtual KITTI (see Section 3.1), as well as improved results on hybrid datasets (adding a domain-randomized training set to COCO [363]), a detailed ablation study, and extensive experiments showing the effect of various hyperparameters.

Since then, domain randomization has been used and further developed in many works. Borrego et al. [69] aim to improve object detection for common objects, showing that domain randomization in the synthetic part of the dataset significantly improves the results. Tobin et al. [587] consider robotic grasping, a problem where the lack of real data is especially dire (see also Sections 3.3 and 6.3). They use domain randomization to generate a wide variety of unrealistic procedurally generated object meshes and textured objects for grasping, so that a model trained on them would generalize to real objects as well. They show that a grasping model trained entirely on non-realistic procedurally generated objects can be successfully transferred to realistic objects.

Up until recently, domain randomization had operated under the assumption that realism is not necessary in synthetic data. Prakash et al. [462] take the next logical step, continuing this effort to *structured* domain randomization. They still randomize all of the settings mentioned above, but only within realistic ranges, taking into account the structure and context of a specific scene.

Finally, another important direction is learning *how* to randomize. Van Vuong et al [616] provide one of the first works in this direction, concentrating on picking the best possible domain randomization parameters for sim-to-real transfer of reinforcement learning policies. They show that the parameters that control sampling over Markov decision processes is important for the quality of transferring the learned policy to a real environment and that these parameters can be optimized. We mark this as a first attempt and expect more works devoted to structuring and honing the parameters of domain randomization.

5.2 Improving CGI-based generation

The basic workflow of synthetic data in computer vision is relatively straightforward: prepare the 3D models, place them in a controlled scene, set up the environment (camera type, lighting etc.), and render synthetic images to be used for training. However, some works on synthetic data present additional ways to enhance the data not by domain adaptation/refinement to real images (we will discuss this approach in Section 6) but directly on the stage of CGI generation.

There are two different directions for this kind of added realism in CGI generation. The first direction is to make more realistic objects. For example, Wang et al. [623] recognize retail items in a smart vending machine; to simulate natural deformations in the objects, they use a surface-based mesh deformation

algorithm proposed in [624], introducing and minimizing a global energy function for the object’s mesh that accounts for random deformations and rigidity properties of the material (Wang et al. also use GAN-based refinement, see Section 6.1.2). Another approach, initiated by Rozantsev et al. [505], is to estimate the rendering parameters required to synthesize similar images from data; this approach ties into the synthetic data generation feedback loop that we discuss in Section 8.2.

The second direction is to make more realistic “sensors”, introducing synthetic data postprocessing that mimics the noise characteristics of real cameras/sensors. For example, we discussed *DepthSynth* by Planche et al. [455] (see Section 2.2.2), a system that makes simulated depth data more realistic, more similar to real depth sensors, while the OVVV system by Taylor et al. [582] (Section 2.3) and the ICL-NUIM dataset by Handa et al. [233] (Section 3.1) take special care to simulate the noise of real cameras. There is even a separate area of research completely devoted to better modeling of the noise and distortions in real world cameras [59].

Apart from added realism on the level of images, there is also the question of high-level coherence and realism of the scenes. While there is no problem with coherence when the scenes are done by hand, the scale of modern datasets requires to automate scene composition as well. We note a recent joint effort in this direction by NVIDIA, University of Toronto, and MIT: Kar et al. [309] present *Meta-Sim*, a general framework that learns to generate synthetic urban environments (see also Section 3.1). *Meta-Sim* represents the composition of a 3D scene with a *scene graph* and a *probabilistic scene grammar*, a common representation in computer graphics [715]. The goal is to learn how to transform samples coming from the probabilistic grammar so that the distribution of synthetic scenes becomes similar to the distribution of scenes in a real dataset; this is known as bridging the *distribution gap*. What’s more, *Meta-Sim* can also learn these transformations with the objective of improving the performance of networks trained on the resulting synthetic data for a specific task such as object detection (see also Section 8.2).

There are also a number of domain-specific developments that improve synthetic data generation for specific fields. For example, Cheung et al. [110] present *LCrowdV*, a generation framework for crowd videos that combines a procedural simulation framework that concentrates of movements and human behaviour and a rendering framework for image/video generation, while Anderson et al. [19] develop a method for stochastic sampling-based simulation of pedestrian trajectories (see Section 2.3).

In general, while computer graphics is increasingly using machine learning to speed up rendering (by, e.g., learning approximations to complex computationally intensive transformations) and improve the resulting 3D graphics, works on synthetic data seldom make use of these advances; a need to improve CGI-based synthetic data is usually considered in the direction of making it more realistic with refinement models (see Section 6.1). However, we do expect further interesting developments in specific domains, especially in situations where the characteristics of specific sensors are important (such as, e.g., LIDARs in autonomous vehicles).

5.3 Compositing real data to produce synthetic datasets

Another notable line of work that, in our opinion, lies at the boundary between synthetic data and data augmentation is to use combinations and fusions of different real images to produce a larger and more diverse set of images for training. This does not require the use of CGI for rendering the synthetic images, but does require a dataset of real images.

Early works in this direction were limited by the quality of segmentation needed to cut out real objects. For some problems, however, it was easy enough to work. For example, Eggert et al. [166] concentrate on company logo detection. To generate synthetic images, they use a small number of real base images where the logos are clearly visible and supplied with segmentation masks, apply random warping, color transformations, and blurring, and then paste the modified (segmented) logo onto a new background image, improving logo detection results. In Section 2.3 we have discussed the “Frankenstein” pipeline for compositing human faces [256].

The field started in earnest with the *Cut, Paste, and Learn* approach by Dwibedi et al. [159], which is based on the assumption that only *patch-level realism* is needed to train, e.g., an object detector. They take a collection of object instance images, cut them out with a segmentation model (assuming that the instance images are simple enough that segmentation will work almost perfectly), and paste them onto randomized background scenes, with no regard to preserving scale or scene composition. Dwibedi et al. compare different classical computer vision blending approaches (e.g., Gaussian and Poisson blending [449]) to alleviate the influence of boundary artifacts after the paste; they report improved instance detection results. The work on cut-and-paste was later extended with GAN-based models (used for more realistic pasting and inpainting) and continued in the direction of unsupervised segmentation by Remez et al. [487] and Ostyakov et al. [434].

Subsequent works extend this approach for generating more realistic synthetic datasets. Dvornik et al. [158] argue that an important problem for this type of data augmentation is to preserve visual context, i.e., make the environment around the objects more or less realistic. They describe a preliminary experiment where they placed segmented object at completely random positions in new scenes and not only did not see significant improvements for object detection on the VOC’12 dataset but actually saw the performance deteriorate, regardless of the distractors or strategies used for blending and boundary artifact removal. Therefore, they added a separate model (also a CNN) that predicts what kind of objects can be placed in a given bounding box of an image from the rest of the image with this bounding box masked out; then the trained model is used to evaluate potential bounding boxes for data augmentation, choose the ones with the best object category score, and then paste a segmented object of this category in the bounding box. The authors report improved object detection results on VOC’12.

Wang et al. [621] develop this into an even simpler idea of *instance switching*: let us switch only instances of the same class between different images in the training set; in this way, the context is automatically right, and shape and scale can also be taken into account. Wang et al. also propose to use instance switching to adjust the distribution of instances across classes in the training set and account for class importance by adding more switching for classes with

lower scores. The resulting PSIS (Progressive and Selective Instance Switching) system provides improved results on the MS COCO dataset for various object detectors including Faster-RCNN [489], FPN [362], Mask R-CNN [237], and SNIPER [550].

With the development of conditional generative models, this field has blossomed into more complex conditional generation, usually called *image fusion*, that goes beyond cut-and-paste; we discuss these extensions in Section 6.1.3.

5.4 Synthetic data produced by generative models

Generative models, especially Generative Adversarial Networks (GAN) [207], are increasingly being used for domain adaptation, either in the form of refining synthetic images to make them more realistic or in the form of “smart augmentation”, making nontrivial transformations on real data. We discuss these techniques in Section 6. Producing synthetic data directly from random noise for classical computer vision applications generally does not sound promising: GANs can only try to approximate what is already in the data, so why can’t the model itself do it? However, in a number of applications synthetic data produced by GANs directly from random noise, usually with an abstract condition such as a segmentation mask, can help; in this section, we consider several examples of these approaches.

Counting (objects on an image) is a computer vision problem that, formally speaking, reduce to object detection or segmentation but in practice is significantly harder: to count correctly the model needs to detect all objects on the image, missing no one. Large datasets are helpful for counting, and synthetic data generated with a GAN conditioned on the number of objects or a segmentation mask with known number of objects, either produced at random or taken from a labeled real dataset, proves to be helpful. In particular, there is a line of work that deals with leaf counting on images of plants: ARIGAN by Giuffrida et al. [201] generates images of arabidopsis plants conditioned on the number of leaves, Zhu et al. generate the same conditioned on segmentation masks [717], and Kuznichov et al. [340] generate synthetically augmented data that preserves the geometric structure of the leaves; all works report improved counting.

Santana and Hotz [518] present a generative model that can learn to generate realistic looking images and even videos of the road for potential training of self-driving cars. Their model is a VAE+GAN autoencoder based on the architecture from [345] that is combined with a recurrent transition model that learns realistic transitions in the embedded space. The resulting model produces synthetic videos that preserve road texture, lane markings, and car edges, keeping the road structure for at least 100 frames of the video. This interesting approach, however, has not yet led to any improvements in the training of actual driving agents.

It is hard to find impressive applications where synthetic data is generated purely from scratch by generative models; as we have discussed, this may be a principled limitation. Still, even a small amount of additional supervision may do. For example, Alonso et al. [14] consider adversarial generation of handwritten text (see also Section 2.4). They condition the generator on the text itself (sequence of characters), then generate handwritten instances for various vocabulary words and augment the real RIMES dataset [213] with the resulting synthetic dataset. Alonso et al. report improved performance in terms

of both edit distance and word error rate. This example shows that synthetic data does not need to involve complicated 3D modeling to work and improve results; in this case, all information Alonso et al. provided for the generative model was a vocabulary of words.

A related but different field considers unsupervised approaches to segmentation and other computer vision problems based on adversarial architectures, including learning to segment via cut-and-paste [487], unsupervised segmentation by moving objects between pairs of images with inpainting [434], segmentation learned from unannotated medical images [703], and more [57]. While this is not synthetic data *per se*, in general we expect unsupervised approaches to computer vision to be an important trend in the use of synthetic data.

6 Synthetic-to-real domain adaptation and refinement

So far, we have discussed direct applications where synthetic data has been used to augment real datasets of insufficient size or to create virtual environments for training. In this section, we proceed to methods that can make the use of synthetic data much more efficient. *Domain adaptation* is a set of techniques designed to make a model trained on one domain of data, the *source* domain, work well on a different, *target* domain. This is a natural fit for synthetic data: in almost all applications, we would like to train the model in the source domain of synthetic data but then apply the results in the target domain of real data.

In this section, we give a survey of domain adaptation approaches that have been used for such synthetic-to-real adaptation. We broadly divide the methods outlined in this section into two groups. Approaches from the first group operates on the data level, which makes it possible to extract synthetic data “refined” in order to work better on real data, while approaches from the second group operate directly on the model, its feature space or training procedure, leaving the data itself unchanged. We concentrate mostly on recent work related to deep neural networks and refer to, e.g., the survey [441] for an overview of earlier work.

In Section 6.1, we discuss synthetic-to-real refinement, where a model learns to make synthetic “fake” data more realistic with an adversarial framework; we begin with a case study on gaze estimation (Section 6.1.1) where this field has mostly originated from and then proceed to other applications of such refiners (Section 6.1.2) and GAN-based models that work in the opposite direction, making real data more “synthetic-like” (Section 6.1.3). In Section 6.2 we proceed to domain adaptation at the feature and model level, i.e., to methods that perform synthetic-to-real domain adaptation but do not necessarily yield more realistic synthetic data as a by-product. Section 6.3 is devoted to domain adaptation in control and robotics, and in Section 6.4 we present a case study of adversarial architectures for medical imaging, one of the fields where synthetic data produced with GANs can significantly improve results.

6.1 Synthetic-to-real refinement

The first group of approaches for synthetic-to-real domain adaptation work with the data itself. The models below can take a synthetic image and “refine” it, making it better for subsequent model training. Note that while in most works we discuss here the objective is basically to make synthetic data more realistic (and it is supported by discriminators that aim to distinguish refined synthetic data from real samples), this does not necessarily have to be the case; some early works on synthetic data concluded that, e.g., synthetic imagery may work better if it is less realistic, resulting in better generalization of the models; we discuss this, e.g., in Section 5.1.

We begin with a case study on a specific problem that kickstarted synthetic-to-real refinement and then proceed to other approaches, both refining already existing synthetic data and generating new synthetic data from real by generative manipulation.

6.1.1 Case study: GAN-based refinement for gaze estimation

One of the first successful examples of straightforward synthetic-to-real refinement was given by Apple researchers Shrivastava et al. in [545], so we begin by considering this case study in more detail and show how the research progressed afterwards. The underlying problem here is *gaze estimation*: recognizing the direction where a human eye is looking. Gaze estimation methods are usually divided into *model-based*, which model the geometric structure of the eye and adjacent regions, and *appearance-based*, which use the eye image directly as input; naturally, synthetic data is made and refined for the latter class of approaches.

Before [545], this problem had already been tackled with synthetic data. Wood et al. [640, 641] presented a large dataset of realistic renderings of human eyes and showed improvements on real test sets over previous work done with the *MPIIGaze* dataset of real labeled images [694]. Note that the usual increase in scale here is manifested as an increase in variability: *MPIIGaze* contains about 214K images, and the synthetic training set was only about 1M images, but all images in *MPIIGaze* come from the same 15 participants of the experiment, while the *UnityEyes* system developed in [641] can render every image in a different randomized environment, which makes the model significantly more robust.

Shrivastava et al. further improve upon this result by presenting a GAN-based system trained to improve synthesized images of the eyes, making them more realistic. They call this idea *Simulated+Unsupervised learning*, and in it they learn a transformation implemented with a *Refiner* network with the *SimGAN* adversarial architecture. SimGAN consists of a generator (refiner) G_θ^{REF} with parameters θ and a discriminator D_ϕ^{REF} with parameters ϕ ; see Fig. 17 for an illustration. The discriminator learns to distinguish between real and refined images with standard binary classification loss function

$$\mathcal{L}_D^{\text{REF}}(\phi) = -\mathbb{E}_S [\log D_\phi^{\text{REF}}(\hat{\mathbf{x}}_S)] - \mathbb{E}_T [\log (1 - D_\phi^{\text{REF}}(\mathbf{x}_T))],$$

where $\hat{\mathbf{x}}_S = G_\theta^{\text{REF}}(\mathbf{x}_S)$ is the refined version of \mathbf{x}_S produced by G_θ^{REF} . The generator, in turn, is trained with a combination of the realism loss $\mathcal{L}_{\text{real}}^{\text{REF}}$ that makes G_θ^{REF} learn to fool D_ϕ^{REF} and regularization loss $\mathcal{L}_{\text{reg}}^{\text{REF}}$ that captures the similarity between the refined image and the original one in order to preserve the target variable (gaze direction in [545]):

$$\begin{aligned} \mathcal{L}_G^{\text{REF}}(\theta) &= \mathbb{E}_S [\mathcal{L}_{\text{real}}^{\text{REF}}(\theta; \mathbf{x}_S) + \lambda \mathcal{L}_{\text{reg}}^{\text{REF}}(\theta; \mathbf{x}_S)], \text{ where} \\ \mathcal{L}_{\text{real}}^{\text{REF}}(\theta; \mathbf{x}_S) &= -\log (1 - D_\phi^{\text{REF}}(G_\theta^{\text{REF}}(\mathbf{x}_S))), \\ \mathcal{L}_{\text{reg}}^{\text{REF}}(\theta; \mathbf{x}_S) &= \|\psi(G_\theta^{\text{REF}}(\mathbf{x}_S)) - \psi(\mathbf{x}_S)\|_1, \end{aligned}$$

where $\psi(\mathbf{x})$ is a mapping to a feature space (that can contain the image itself, image derivatives, statistics of color channels, or features produced by a fixed extractor such as a pretrained CNN), and $\|\cdot\|_1$ denotes the L_1 distance. On Fig. 17, black arrows denote the data flow and green arrows show the gradient flow (on subsequent pictures, we omit the gradient flow to avoid clutter); $\mathcal{L}_{\text{real}}^{\text{REF}}(\theta)$ and $\mathcal{L}_D^{\text{REF}}(\phi)$ are shown in the same block since it is the same loss function differentiated with respect to different weights for G and D respectively.

In SimGAN, the generator is a fully convolutional neural network that consists of several ResNet blocks [238] and does not contain any striding or pooling, which makes it possible to operate on pixel level while preserving the global

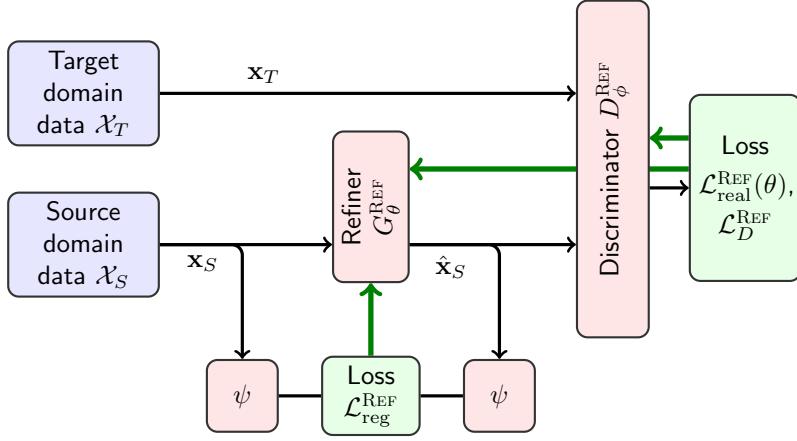


Figure 17: The architecture of SimGAN, a GAN-based refiner for synthetic data [545].

structure. The training proceeds by alternating between minimizing $\mathcal{L}_G^{\text{REF}}(\theta)$ and $\mathcal{L}_D^{\text{REF}}(\phi)$, with an additional trick of drawing training samples for the discriminator from a stored history of refined images in order to keep it effective against all versions of the generator. Another important feature is the locality of adversarial loss: D_ϕ^{REF} outputs a probability map on local patches of the original image, and $\mathcal{L}_D^{\text{REF}}(\phi)$ is summed over the patches.

SimGAN's ideas were later picked up and extended in many works. A direct successor of SimGAN, GazeGAN developed by Sela et al. [534], applied to synthetic data refinement the idea of CycleGAN for unpaired image-to-image translation [714]. The structure of GazeGAN contains four networks: G^{Gz} is the generator that learns to map images from the synthetic domain S to the real domain R, F^{Gz} learns the opposite mapping, from R to S, and two discriminators D_S^{Gz} and D_R^{Gz} learn to distinguish between real and fake images in the synthetic and real domains respectively. An overview of the GazeGAN architecture is shown on Fig. 18. It uses the following loss functions:

- the LSGAN [394] loss for the generator with label smoothing to 0.9 [448] to stabilize training:

$$\mathcal{L}_{\text{LSGAN}}^{\text{Gz}}(G, D, S, R) = \mathbb{E}_{\mathbf{x}_S \sim p_{\text{syn}}} [(D(G(\mathbf{x}_S)) - 0.9)^2] + \mathbb{E}_{\mathbf{x}_T \sim p_{\text{real}}} [D(\mathbf{x}_T)^2];$$

this loss is applied to both directions, as $\mathcal{L}_{\text{LSGAN}}^{\text{Gz}}(G^{\text{Gz}}, D_R^{\text{Gz}}, \mathcal{X}_S, \mathcal{X}_T)$ and $\mathcal{L}_{\text{LSGAN}}^{\text{Gz}}(F^{\text{Gz}}, D_S^{\text{Gz}}, \mathcal{X}_T, \mathcal{X}_S)$;

- the cycle consistency loss [714] designed to make sure both $F \circ G$ and $G \circ F$ are close to identity:

$$\begin{aligned} \mathcal{L}_{\text{Cyc}}^{\text{Gz}}(G^{\text{Gz}}, F^{\text{Gz}}) = & \mathbb{E}_{\mathbf{x}_S \sim p_{\text{syn}}} [\|F^{\text{Gz}}(G^{\text{Gz}}(\mathbf{x}_S)) - \mathbf{x}_S\|_1] + \\ & \mathbb{E}_{\mathbf{x}_T \sim p_{\text{real}}} [\|G^{\text{Gz}}(F^{\text{Gz}}(\mathbf{x}_T)) - \mathbf{x}_T\|_1]; \end{aligned}$$

- finally, a special gaze cycle consistency loss to preserve the gaze direction (so that the target variable can be transferred with no change); for this,

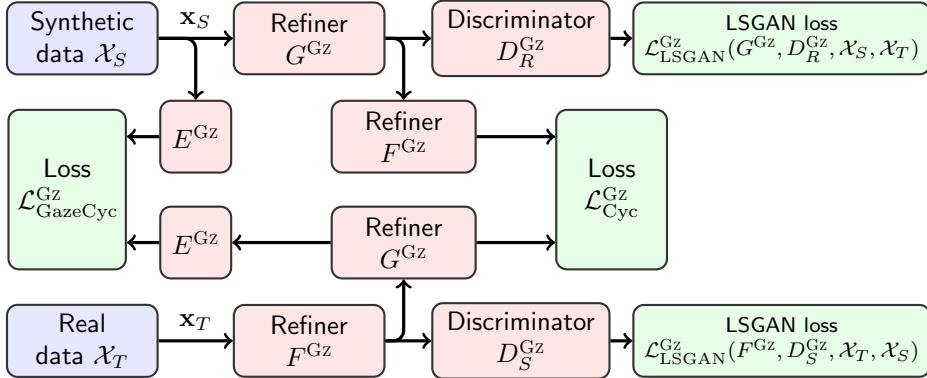


Figure 18: The architecture of GazeGAN [534]. Blocks with identical labels have shared weights.

the authors train a separate gaze estimation network E^{Gz} designed to overfit and predict the gaze very accurately on synthetic data; the loss makes sure E^{Gz} still works after applying $F \circ G$:

$$\mathcal{L}_{\text{GazeCyc}}^{Gz}(G^{Gz}, F^{Gz}) = \mathbb{E}_{\mathbf{x}_S \sim p_{\text{syn}}} [\|E^{Gz}(F^{Gz}(G^{Gz}(\mathbf{x}_S))) - E^{Gz}(\mathbf{x}_S)\|_2^2].$$

Sela et al. report improved gaze estimation results. Importantly for us, they operate not on the 30×60 grayscale images as in [545], but on 128×128 color images, and GazeGAN actually refines not only the eye itself but parts of the image (e.g., nose and hair) that were not part of the 3D model of the eye.

Finally, a note of caution: GAN-based refinement is not the only way to go. Kan et al. [304] compared three approaches to data augmentation for pupil center point detection, an important subproblem in gaze estimation: affine transformations of real images, synthetic images from *UnityEyes*, and GAN-based refinement. In their experiments, real data augmentation with affine transformations was a clear winner, with the GAN improving over *UnityEyes* but falling short of the augmented real dataset. This is one example of a general common wisdom: in cases where a real dataset is available, one should squeeze out all the information available in it and apply as much augmentation as possible, regardless of whether the dataset is augmented with synthetic data or not.

6.1.2 Refining synthetic data with GANs

Gaze estimation is a convenient problem for GAN-based refining because the images of eyes used for gaze estimation have relatively low resolution, and scaling GANs up to high-resolution images has proven to be a difficult task in many applications. Nevertheless, in this section we consider a wider picture of other GAN-based refiners applied for synthetic-to-real domain adaptation.

We begin with an early work in refinement, parallel to [545], which was done by *Google* researchers Bousmalis et al. [71]. They train a GAN-based architecture for pixel-level domain adaptation (PixelDA), using a basic style transfer GAN, i.e., they find by alternating optimization steps

$$\min_{\theta_G, \theta_T} \max_{\phi} \lambda_1 \mathcal{L}_{\text{dom}}^{\text{PIX}}(D^{\text{PIX}}, G^{\text{PIX}}) + \lambda_2 \mathcal{L}_{\text{task}}^{\text{PIX}}(G^{\text{PIX}}, T^{\text{PIX}}) + \lambda_3 \mathcal{L}_{\text{cont}}^{\text{PIX}}(G^{\text{PIX}}), \quad \text{where:}$$

- $\mathcal{L}_{\text{dom}}^{\text{PIX}}(D^{\text{PIX}}, G^{\text{PIX}})$ is the domain loss,

$$\mathcal{L}_{\text{dom}}^{\text{PIX}}(D^{\text{PIX}}, G^{\text{PIX}}) = \mathbb{E}_{\mathbf{x}_S \sim p_{\text{syn}}} [\log (1 - D^{\text{PIX}}(G^{\text{PIX}}(\mathbf{x}_S; \theta_G); \phi))] + \mathbb{E}_{\mathbf{x}_T \sim p_{\text{real}}} [\log D^{\text{PIX}}(\mathbf{x}_T; \phi)];$$

- $\mathcal{L}_{\text{task}}^{\text{PIX}}(G^{\text{PIX}}, T^{\text{PIX}})$ is the task-specific loss, which in [71] was the image classification cross-entropy loss provided by a classifier $T^{\text{PIX}}(\mathbf{x}; \theta_T)$ which is also trained as part of the model:

$$\begin{aligned} \mathcal{L}_{\text{task}}^{\text{PIX}}(G^{\text{PIX}}, T^{\text{PIX}}) = \\ \mathbb{E}_{\mathbf{x}_S, \mathbf{y}_S \sim p_{\text{syn}}} [-\mathbf{y}_S^\top \log T^{\text{PIX}}(G^{\text{PIX}}(\mathbf{x}_S; \theta_G); \theta_T) - \mathbf{y}_S^\top \log T^{\text{PIX}}(\mathbf{x}_S; \theta_T)]; \end{aligned}$$

- $\mathcal{L}_{\text{cont}}^{\text{PIX}}(G^{\text{PIX}})$ is the content similarity loss, intended to make G^{PIX} preserve the parts of the image related to the target variables; in [71], $\mathcal{L}_{\text{cont}}^{\text{PIX}}$ was used to preserve the foreground objects (that would later need to be classified) with a mean squared error applied to their masks:

$$\begin{aligned} \mathcal{L}_{\text{cont}}^{\text{PIX}}(G^{\text{PIX}}) = \mathbb{E}_{\mathbf{x}_S \sim p_{\text{syn}}} \left[\frac{1}{k} \|(\mathbf{x}_S - G^{\text{PIX}}(\mathbf{x}_S; \theta_G)) \odot \mathbf{m}(\mathbf{x})\|_2^2 - \right. \\ \left. - \frac{1}{k^2} ((\mathbf{x}_S - G^{\text{PIX}}(\mathbf{x}_S; \theta_G))^\top \mathbf{m}(\mathbf{x}))^2 \right], \end{aligned}$$

where $\mathbf{m}(\mathbf{x}_S)$ is a segmentation mask for the foreground object extracted from the synthetic data renderer; note that this loss does not “insist” on preserving pixel values in the object but rather encourages the model to change object pixels in a consistent way, preserving their pairwise differences.

Bousmalis et al. applied this GAN to the *Synthetic Cropped LineMod* dataset, a synthetic version of a small object classification dataset [638], doing both classification and pose estimation for the objects. They report improved results in both metrics compared to both training on purely synthetic data and a number of previous approaches to domain adaptation.

Many modern approaches to synthetic data refinement include the ideas of CycleGAN [714]. The most direct application is the *GeneSIS-RT* framework by Stein and Roy [559] that refines synthetic data directly with the CycleGAN trained on unpaired datasets of synthetic and real images. They show that a training set produced by image-to-image translation learned by CycleGAN improves the results of training machine learning systems for real-world tasks such as obstacle avoidance and semantic segmentation.

$T^2\text{Net}$ by Zheng et al. [710] uses synthetic-to-real refinement for depth estimation from a single image. This work also uses the general ideas of CycleGAN with a translation network that makes the images more realistic. The new idea here is that $T^2\text{Net}$ asks the synthetic-to-real generator $G_S^{T^2}$ not only to translate one specific domain (synthetic data) to another (real data) but also to work across a number of different input domains, making the input image “more realistic” in every case, as shown on Figure 19. In essence, this means that $G_S^{T^2}$ aims to learn the minimal transformation necessary to make an image realistic, in particular, it should not change real images much. In total, $T^2\text{Net}$ has the

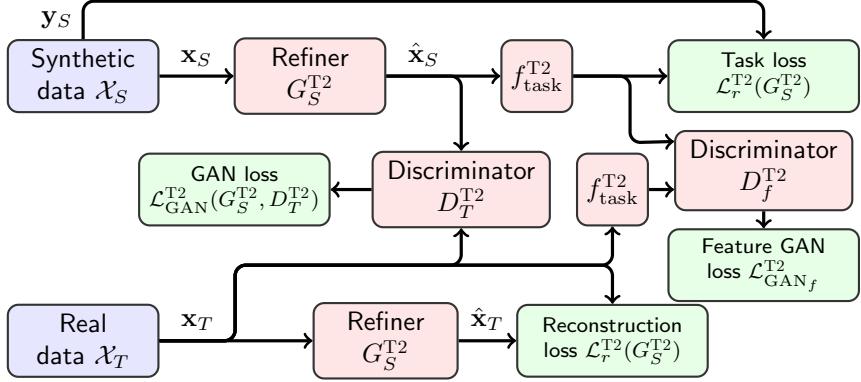


Figure 19: The architecture of T²Net [710]. Blocks with identical labels have shared weights.

generator loss function

$$\begin{aligned} \mathcal{L}^{\text{T2}} = & \mathcal{L}_{\text{GAN}}^{\text{T2}}(G_S^{\text{T2}}, D_T^{\text{T2}}) + \lambda_1 \mathcal{L}_{\text{GAN}_f}^{\text{T2}}(f_{\text{task}}^{\text{T2}}, D_f^{\text{T2}}) + \lambda_2 \mathcal{L}_r^{\text{T2}}(G_S^{\text{T2}}) \\ & + \lambda_3 \mathcal{L}_t^{\text{T2}}(f_{\text{task}}^{\text{T2}}) + \lambda_4 \mathcal{L}_s^{\text{T2}}(f_{\text{task}}^{\text{T2}}), \text{ where} \end{aligned}$$

- $\mathcal{L}_{\text{GAN}}^{\text{T2}}(G_S^{\text{T2}}, D_T^{\text{T2}})$ is the usual GAN loss for synthetic-to-real transfer with discriminator D_T^{T2} :

$$\begin{aligned} \mathcal{L}_{\text{GAN}}^{\text{T2}}(G_S^{\text{T2}}, D_T^{\text{T2}}) = & \mathbb{E}_{\mathbf{x}_S \sim p_{\text{syn}}} [\log(1 - D_T^{\text{T2}}(G_S^{\text{T2}}(\mathbf{x}_S)))] + \\ & \mathbb{E}_{\mathbf{x}_T \sim p_{\text{real}}} [\log D_T^{\text{T2}}(\mathbf{x}_T)]; \end{aligned}$$

- $\mathcal{L}_{\text{GAN}_f}^{\text{T2}}(f_{\text{task}}^{\text{T2}}, D_f^{\text{T2}})$ is the feature-level GAN loss for the features extracted from translated and real images with discriminator D_f^{T2} :

$$\begin{aligned} \mathcal{L}_{\text{GAN}_f}^{\text{T2}}(f_{\text{task}}^{\text{T2}}, D_f^{\text{T2}}) = & \mathbb{E}_{\mathbf{x}_S \sim p_{\text{syn}}} [\log D_f^{\text{T2}}(f_{\text{task}}^{\text{T2}}(G_S^{\text{T2}}(\mathbf{x}_S)))] + \\ & \mathbb{E}_{\mathbf{x}_T \sim p_{\text{real}}} [\log(1 - D_f^{\text{T2}}(f_{\text{task}}^{\text{T2}}(\mathbf{x}_T)))] ; \end{aligned}$$

- $\mathcal{L}_r^{\text{T2}}(G_S^{\text{T2}}) = \|G_S^{\text{T2}}(\mathbf{x}_T) - \mathbf{x}_T\|_1$ is the reconstruction loss for real images;
- $\mathcal{L}_r^{\text{T2}}(f_{\text{task}}^{\text{T2}}) = \|f_{\text{task}}^{\text{T2}}(\hat{\mathbf{x}}_S) - \mathbf{y}_S\|_1$ is the *task loss* for depth estimation on synthetic images, namely the L_1 -norm of the difference between the predicted depth map for a translated synthetic image $\hat{\mathbf{x}}_S$ and the original ground truth synthetic depth map \mathbf{y}_S ; this loss ensures that the translation does not change the depth map;
- $\mathcal{L}_s^{\text{T2}}(f_{\text{task}}^{\text{T2}}) = |\partial_x f_{\text{task}}^{\text{T2}}(\mathbf{x}_T)|^{-|\partial_x \mathbf{x}_T|} + |\partial_y f_{\text{task}}^{\text{T2}}(\mathbf{x}_T)|^{-|\partial_y \mathbf{x}_T|}$, where ∂_x and ∂_y are image gradients, is the task loss for depth estimation on real images; since ground truth depth maps are not available now, this regularizer is a locally smooth loss intended to optimize object boundaries, a common tool in depth estimation models [204].

Zheng et al. show that T²Net can produce realistic images from synthetic ones, conclude that end-to-end training is preferable over separated training (of the translation network and depth estimation network), and note that T²Net can achieve good results for depth estimation with no access to real paired data, even outperforming some (but not all) supervised approaches.

We note a few more interesting applications of refiner-based architectures. Wang et al. [633] use a classical refiner modeled after [545] for human motion synthesis and control. Their model first generates a motion sequence from a recurrent neural network and then refines it with a GAN; since the goal is to model and refine sequences, both generator and discriminator in the refiner also have RNN-based architectures. Dilipkumar [148] applied SimGAN to improve handwriting recognition. They generated synthetic handwriting images and applied SimGAN to refine them, with significantly improved recognition of real handwriting after training on a hybrid dataset.

A recent example that applies GAN-based refinement in a classical computer vision setting is provided by Wang et al. [623]. They consider the problem of recognizing objects inside an automatic vending machines; this is a basic functionality needed for monitoring the state of supplies and is usually done based on object detection. Wang et al. begin by scanning the objects, adding random deformations to the resulting 3D models (see Section 5.2), setting up scenes and rendering with settings matching the fisheye cameras used in smart vending machines. Then they refine rendered images with virtual-to-real style transfer done by a CycleGAN-based architecture. The novelty here is that Wang et al. separate foreground and background losses, arguing that style transfer needed for foreground objects is very different from (much stronger than) the style transfer for backgrounds. Thus, they use the overall loss function

$$\begin{aligned} \mathcal{L}^{\text{OD}} = & \mathcal{L}_{\text{GAN}}^{\text{OD}}(G^{\text{OD}}, D_T^{\text{OD}}, \mathcal{X}_S, \mathcal{X}_T) + \mathcal{L}_{\text{GAN}}^{\text{OD}}(F^{\text{OD}}, D_S^{\text{OD}}, \mathcal{X}_T, \mathcal{X}_S) + \\ & + \lambda_1 \mathcal{L}_{\text{cyc}}^{\text{OD}}(G^{\text{OD}}, F^{\text{OD}}) + \lambda_2 \mathcal{L}_{\text{bg}}^{\text{OD}} + \lambda_3 \mathcal{L}_{\text{fg}}^{\text{OD}}, \quad \text{where:} \end{aligned}$$

- $\mathcal{L}_{\text{GAN}}^{\text{OD}}(G, D, X, Y)$ is the standard adversarial loss for generator G mapping from domain X to domain Y and discriminator D distinguishing real images from fake ones in the domain Y ;
- $\mathcal{L}_{\text{cyc}}^{\text{OD}}(G, F)$ is the cycle consistency loss as used in CycleGAN [714] and detailed above;
- $\mathcal{L}_{\text{bg}}^{\text{OD}}$ is the background loss, which is the cycle consistency loss computed only for the background part of the images as defined by the mask \mathbf{m}_{bg} :

$$\begin{aligned} \mathcal{L}_{\text{bg}}^{\text{OD}} = & \mathbb{E}_{\mathbf{x}_T \sim p_{\text{real}}} [\| (G^{\text{OD}}(F^{\text{OD}}(\mathbf{x}_T)) - \mathbf{x}_T) \odot \mathbf{m}_{\text{bg}}(\mathbf{x}_T) \|_2] \\ & + \mathbb{E}_{\mathbf{x}_S \sim p_{\text{syn}}} [\| (F^{\text{OD}}(G^{\text{OD}}(\mathbf{x}_S)) - \mathbf{x}_S) \odot \mathbf{m}_{\text{bg}}(\mathbf{x}_S) \|_2]; \end{aligned}$$

- $\mathcal{L}_{\text{fg}}^{\text{OD}}$ is the foreground loss, similar to $\mathcal{L}_{\text{bg}}^{\text{OD}}$ but computed only for the hue channel in the HSV color space (the authors argue that color and profile are the most critical for recognition and thus need to be preserved the most), as denoted by \cdot^H below:

$$\begin{aligned} \mathcal{L}_{\text{fg}}^{\text{OD}} = & \mathbb{E}_{\mathbf{x}_T \sim p_{\text{real}}} [\| (G^{\text{OD}}(F^{\text{OD}}(\mathbf{x}_T))^H - \mathbf{x}_T^H) \odot \mathbf{m}_{\text{fg}}(\mathbf{x}_T) \|_2] \\ & + \mathbb{E}_{\mathbf{x}_S \sim p_{\text{syn}}} [\| (F^{\text{OD}}(G^{\text{OD}}(\mathbf{x}_S))^H - \mathbf{x}_S^H) \odot \mathbf{m}_{\text{fg}}(\mathbf{x}_S) \|_2]. \end{aligned}$$

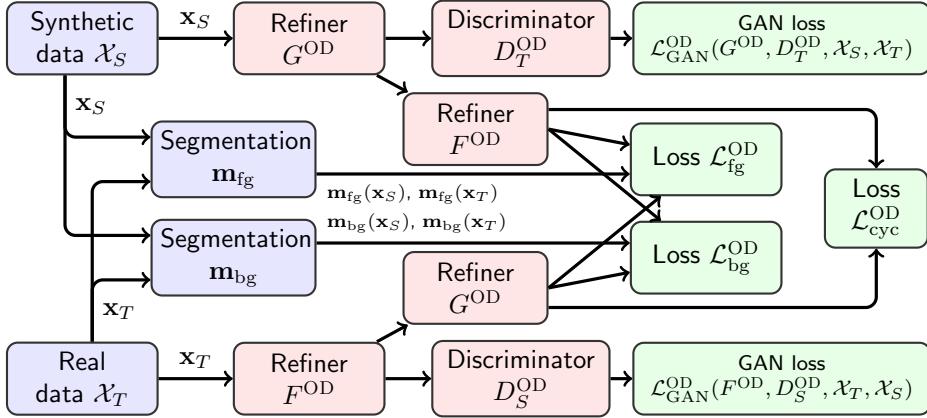


Figure 20: The architecture of the refiner used in [623]. Blocks with identical labels have shared weights.

Segmentation into foreground and background is done automatically in synthetic data and is made easy in [623] for real data since the camera position is fixed, and the authors can collect a dataset of real background templates from the vending machines they used in the experiments and then simply subtract the backgrounds to get the foreground part. As a result, Wang et al. report significantly improved results when using hybrid datasets of real and synthetic data for all three tested object detection architectures: PVANET [320], SSD [372], and YOLOv3 [482]. Even more importantly, they report a comparison between basic and refined synthetic data with clear gains achieved by refinement across all architectures.

Wang et al. [625] discuss synthetic-to-real domain adaptation in the context of crowd counting, a domain where synthetic data has been successfully used for a long time. They collect synthetic data with the *Grand Theft Auto V* engine, producing the so-called *GTA5 Crowd Counting* (GCC) dataset (see also Section 2.3). They also use a CycleGAN-based refiner from the domain of synthetic images \mathcal{X}_S to the domain of real images \mathcal{X}_T but remark that CycleGAN can easily lose local patterns and textures which is exactly what is important for crowd counting. Therefore, they modify the CycleGAN loss function with the Structural Similarity Index (SSIM) [713] that computes the similarity between images in terms of local patterns. Their final loss function is

$$\begin{aligned} \mathcal{L}^{\text{SSIM}} = & \mathcal{L}_{\text{GAN}}^{\text{SSIM}}(G^{\text{SSIM}}, D_T^{\text{SSIM}}, \mathcal{X}_S, \mathcal{X}_T) + \mathcal{L}_{\text{GAN}}^{\text{SSIM}}(F^{\text{SSIM}}, D_S^{\text{SSIM}}, \mathcal{X}_T, \mathcal{X}_S) \\ & + \lambda \mathcal{L}_{\text{cyc}}^{\text{SSIM}}(G^{\text{SSIM}}, F^{\text{SSIM}}, \mathcal{X}_S, \mathcal{X}_R) + \mu \mathcal{L}_{\text{SE}}^{\text{SSIM}}(G^{\text{SSIM}}, F^{\text{SSIM}}, \mathcal{X}_S, \mathcal{X}_R), \end{aligned}$$

where $G^{\text{SSIM}} : \mathcal{X}_S \rightarrow \mathcal{X}_T$ is the generator from synthetic to real domains, $F^{\text{SSIM}} : \mathcal{X}_T \rightarrow \mathcal{X}_S$ works in the opposite direction, D_T^{SSIM} and D_S^{SSIM} are the corresponding discriminators, $\mathcal{L}_{\text{GAN}}^{\text{SSIM}}$ is a standard GAN loss function, and $\mathcal{L}_{\text{cyc}}^{\text{SSIM}}$ is the cycle consistency loss as defined above, while $\mathcal{L}_{\text{SE}}^{\text{SSIM}}$ is a special loss function designed to improve SSIM:

$$\begin{aligned} \mathcal{L}_{\text{SE}}^{\text{SSIM}}(G^{\text{SSIM}}, F^{\text{SSIM}}, \mathcal{X}_S, \mathcal{X}_R) = & \mathbb{E}_{\mathbf{x}_S \sim p_{\text{syn}}} [1 - \text{SSIM}(\mathbf{x}_S, F^{\text{SSIM}}(G^{\text{SSIM}}(\mathbf{x}_S)))] \\ & + \mathbb{E}_{\mathbf{x}_T \sim p_{\text{data}}} [1 - \text{SSIM}(\mathbf{x}_T, G^{\text{SSIM}}(F^{\text{SSIM}}(\mathbf{x}_T)))] . \end{aligned}$$

Bak et al. [32] (see Section 2.3) use domain translation with synthetic people as a method for person re-identification. The domains in this model are represented by illumination conditions for the images; the model has access to M real source domains and N synthetic domains, with $N \gg M$, and the objective is to perform re-identification in an unknown target domain. To achieve this, Bak et al. first learned a generic feature representation from all domains, but the resulting model, even trained on a hybrid dataset, did not generalize well. Therefore, Bak et al. proceeded to domain adaptation done as follows: first choose the nearest synthetic with a separately trained domain identification network fine-tuned for illumination classification, then use the CycleGAN architecture (as shown above) to do domain translation from this synthetic domain to the target domain, and then use the to fine-tune the re-identification network. Bak et al. report improved results from the entire pipeline as compared to any individual parts in the ablation study.

In robotics, domain adaptation of synthetic imagery is not yet common, but some applications have appeared there as well. For example, Pecka et al. [442] train a CycleGAN-based domain adaptation model to learn a transformation from the data observed in a non-differentiable physics simulator and the data from a real robotic platform, showing improved sim-to-real policy transfer results.

6.1.3 Making synthetic data from real with GANs

A related idea is to generate synthetic data from real data by learning to transform real data with conditional GANs. This could either simply serve as “smart augmentation” to extend the dataset or, more interestingly, could “fill in the holes” in the data distribution, obtaining synthetic data for situations that are lacking in the original dataset.

Zhao et al. [707, 708] concentrated on applying this idea to face recognition in the wild, with different poses rather than by a frontal image. They continued the work of Tran et al. [593] (we do not review it in detail) and Huang et al. [264], who presented a TP-GAN (two-pathway GAN) architecture for frontal view synthesis: given a picture of a face, generate a frontal view picture. TP-GAN’s generator G_θ^{TP} has two pathways: a global network $G_{\theta^g}^{\text{TP}}$ that rotates the entire face and four local patch networks $G_{\theta_i^l}^{\text{TP}}$, $i = 1, \dots, 4$, that process local textures around four facial landmarks (eyes, nose, and mouth). Both $G_{\theta^g}^{\text{TP}}$ and $G_{\theta_i^l}^{\text{TP}}$ have encoder-decoder architectures with skip connections for multi-scale feature fusion. The discriminator D_ϕ^{TP} learns to distinguish real frontal face images $\mathbf{x}_{\text{front}} \sim \mathcal{D}_{\text{front}}$ from synthesized images $G_\theta^{\text{TP}}(\mathbf{x})$, $\mathbf{x} \sim p_{\text{data}}$:

$$\min_{\theta} \max_{\phi} \left[\mathbb{E}_{\mathbf{x}_{\text{front}} \sim \mathcal{D}_{\text{front}}} [\log D_\phi^{\text{TP}}(\mathbf{x}_{\text{front}})] + \mathbb{E}_{\mathbf{x} \sim p_{\text{data}}} [\log (1 - D_\phi^{\text{TP}}(G_\theta^{\text{TP}}(\mathbf{x})))] \right].$$

The synthesis loss function in TP-GAN is a sum of four loss functions:

- pixel-wise L_1 -loss between the ground truth frontal image and rotated image:

$$\mathcal{L}_{\text{pixel}}^{\text{TP}}(\mathbf{x}, \mathbf{x}^{\text{gt}}) = \frac{1}{W \times H} \sum_{i=1}^W \sum_{j=1}^H |G_\theta^{\text{TP}}(\mathbf{x})_{i,j} - \mathbf{x}_{i,j}^{\text{gt}}|,$$

where \mathbf{x}^{gt} is the ground truth frontal image corresponding to \mathbf{x} , and W and H are an image's width and height; this loss is measured at several different places of the network: output of $G_{\theta^g}^{\text{TP}}$, outputs of $G_{\theta_i^l}^{\text{TP}}$, and the final output of G_{θ}^{TP} ;

- symmetry loss

$$\mathcal{L}_{\text{sym}}^{\text{TP}} = \frac{1}{W/2 \times H} \sum_{i=1}^{W/2} \sum_{j=1}^H |G_{\theta}^{\text{TP}}(\mathbf{x})_{i,j} - G_{\theta}^{\text{TP}}(\mathbf{x})_{W-(i-1),j}|,$$

intended to preserve the symmetry of human faces;

- adversarial loss

$$\mathcal{L}_{\text{adv}}^{\text{TP}} = \frac{1}{N} \sum_{n=1}^N -\log D_{\phi}^{\text{TP}}(G_{\theta}^{\text{TP}}(\mathbf{x}_n));$$

- identity preserving loss

$$\mathcal{L}_{\text{ip}}^{\text{TP}} = \sum_l \frac{1}{W_l \times H_l} \sum_{i=1}^{W_l} \sum_{j=1}^{H_l} |F^l(G_{\theta}^{\text{TP}}(\mathbf{x}))_{i,j} - F^l(\mathbf{x}^{\text{gt}})_{i,j}|,$$

where F^l denotes the output of the l th layer of a face recognition network applied to \mathbf{x} and \mathbf{x}^{gt} (Huang et al. used Light CNN [646], and only used its last two layers in $\mathcal{L}_{\text{ip}}^{\text{TP}}$); this idea is based on *perceptual losses* [287], a popular idea in GANs designed to preserve high-level features when doing low-level transformations; in this case, it serves to preserve the person's identity when rotating the face.

As usual, the final loss function is a linear combination of the four losses above and a regularization term.

In [708], Zhao et al. propose the DA-GAN (Dual-Agent GAN) model that also works with faces but in the opposite scenario: while TP-GAN rotates every face into the frontal view, DA-GAN aims to fill in the “holes” in the real data distribution, rotating real faces so that the distribution of angles becomes more uniform. They begin with a 3D morphable model (see also Section 2.3) from [716], extracting 68 facial landmarks with the Recurrent Attentive-Refinement (RAR) model [655] and estimating the transformation matrix with 3D-MM [61]. However, the authors report that simulation quality dramatically decreases for large yaw angles, necessitating further improvement with the DA-GAN framework.

Again, DA-GAN's generator G_{θ}^{DA} maps a synthetized image to a refined one, $\hat{\mathbf{x}}_S = G_{\theta}^{\text{DA}}(\mathbf{x}_S)$. It is trained on a linear combination of three loss functions

$$\mathcal{L}_G^{\text{DA}} = -\mathcal{L}_{\text{adv}}^{\text{DA}} + \lambda_1 \mathcal{L}_{\text{ip}}^{\text{DA}} + \lambda_2 \mathcal{L}_{\text{pp}}^{\text{DA}},$$

and the discriminator D_{ϕ}^{DA} consists of two parallel branches (agents) that optimize $\mathcal{L}_{\text{adv}}^{\text{DA}}$ and $\mathcal{L}_{\text{ip}}^{\text{DA}}$ respectively. The loss functions are defined as follows:

- the adversarial loss $\mathcal{L}_{\text{adv}}^{\text{DA}}$ follows the BEGAN architecture introduced in [54]: this branch of D_{ϕ}^{DA} is an autoencoder that minimizes the Wasserstein distance with a boundary equilibrium regularization term:

$$\mathcal{L}_{\text{adv}}^{\text{DA}} = \sum_j |\mathbf{x}_{T,j} - D_{\phi}^{\text{DA}}(\mathbf{x}_T)_j| - k_t \sum_i |\hat{\mathbf{x}}_{S,i} - D_{\phi}^{\text{DA}}(\hat{\mathbf{x}}_S)_i|,$$

where, again, \mathbf{x}_T is a real image, $\hat{\mathbf{x}}_S$ is a refined image, and k_t is a boundary equilibrium regularization term continuously trained to maintain the equilibrium

$$\mathbb{E} \left[\sum_i |\hat{\mathbf{x}}_{S,i} - D_\phi^{\text{DA}}(\hat{\mathbf{x}}_S)_i| \right] = \gamma \mathbb{E} \left[\sum_j |\mathbf{x}_{T,j} - D_\phi^{\text{DA}}(\mathbf{x}_T)_j| \right]$$

for some diversity ratio γ (see [54, 708] for more details); in general, $\mathcal{L}_{\text{adv}}^{\text{DA}}$ is designed to keep the refined face in the manifold of real faces;

- the identity preservation loss $\mathcal{L}_{\text{ip}}^{\text{DA}}$, similar to $\mathcal{L}_{\text{ip}}^{\text{TP}}$, aims to make the refinement respect the identities, but does it in a different way; here the idea is to put both \mathbf{x}_T and \mathbf{x}_S through the same (relatively simple) face recognition network and bring its features together; DA-GAN uses for this purpose a classifier C_ϕ^{DA} trained on the bottleneck layer of D_ϕ^{DA} :

$$\begin{aligned} \mathcal{L}_{\text{ip}}^{\text{DA}} = & \frac{1}{N} \sum_j [-\mathbf{y}_j \log C_\phi^{\text{DA}}(\mathbf{x}_{T,j}) + (1 - \mathbf{y}_j) \log(1 - C_\phi^{\text{DA}}(\mathbf{x}_{T,j}))] \\ & + \frac{1}{N} \sum_j [-\mathbf{y}_j \log C_\phi^{\text{DA}}(\hat{\mathbf{x}}_{S,j}) + (1 - \mathbf{y}_j) \log(1 - C_\phi^{\text{DA}}(\hat{\mathbf{x}}_{S,j}))], \end{aligned}$$

where \mathbf{y}_j is the ground truth label;

- the pixel-wise loss $\mathcal{L}_{\text{pp}}^{\text{DA}}$ is the L_1 -loss intended to make sure that the pose (angle of inclination for the head) remains the same after refinement:

$$\mathcal{L}_{\text{pp}}^{\text{DA}} = \frac{1}{W \times H} \sum_{i=1}^W \sum_{j=1}^H |\mathbf{x}_{S,i,j} - \hat{\mathbf{x}}_{S,i,j}|.$$

In total, during training DA-GAN alternatively optimizes G_θ^{DA} and D_ϕ^{DA} with loss functions $\mathcal{L}_G^{\text{DA}}$ and $\mathcal{L}_D^{\text{DA}} = \mathcal{L}_{\text{adv}}^{\text{DA}} + \lambda_1 \mathcal{L}_{\text{ip}}^{\text{DA}}$. Following [54], to measure convergence DA-GAN tests the reconstruction quality together with proportion control theory, evaluating

$$\mathcal{L}_{\text{con}}^{\text{DA}} = \sum_j |\mathbf{x}_{T,j} - D_\phi^{\text{DA}}(\mathbf{x}_{T,j})| + \left| \gamma \sum_j |\mathbf{x}_{T,j} - D_\phi^{\text{DA}}(\mathbf{x}_{T,j})| - \sum_i |\hat{\mathbf{x}}_{S,i} - D_\phi^{\text{DA}}(\hat{\mathbf{x}}_{S,i})| \right|.$$

Apart from experiments done by the authors, DA-GAN was verified in a large-scale NIST IJB-A competition [171] where a model based on DA-GAN won the face verification and face identification tracks. This result heavily supports the general premise of using synthetic data: augmenting the dataset and balancing out the training data distribution with synthetic images proved highly beneficial in this case.

Inoue et al. [273] try to find a middle ground between synthetic and real data. They use two variational autoencoders (VAE) to reduce both synthetic and real data to a common pseudo-synthetic image space, and then train CNNs on images from this common space. The training sequence is as follows:

- train $\text{VAE}_1 : \mathcal{X}_S \rightarrow \mathcal{X}_S$ as an autoencoder;

- train VAE₂ : $\mathcal{X}_T \rightarrow \mathcal{X}_S$ where the decoder is fixed and shares weights with the decoder from VAE₁; as a result, VAE₂ has to learn to generate pseudo-synthetic images from real images;
- train a CNN for the task in question on synthetic data, using VAE₁ to map it to the common image space;
- during inference, use a composition of VAE₂ and CNN.

In the context of robotics, this kind of *real-to-sim* approach was continued by Zhang et al. [690] in a framework called “VR-Goggles for Robots”. It is based on the CycleGAN ideas as they were continued in CyCADA [249], a popular domain adaptation model that adds semantic losses to CycleGAN. The *VR-Goggles* model has two generators, $G_S^{\text{VRG}} : \mathcal{X}_T \rightarrow \mathcal{X}_S$ with discriminator D_T^{VRG} that distinguishes fake synthetic images and $G_T^{\text{VRG}} : \mathcal{X}_S \rightarrow \mathcal{X}_T$ with discriminator D_S^{VRG} that is defined in the domain of real images. The overall loss function is

$$\begin{aligned} \mathcal{L}^{\text{VRG}} = & \mathcal{L}_{\text{GAN}}^{\text{VRG}}(G_T^{\text{VRG}}, D_T^{\text{VRG}}; \mathcal{X}_S, \mathcal{X}_T) + \mathcal{L}_{\text{GAN}}^{\text{VRG}}(G_S^{\text{VRG}}, D_S^{\text{VRG}}; \mathcal{X}_T, \mathcal{X}_S) \\ & + \lambda_1 (\mathcal{L}_{\text{cyc}}^{\text{VRG}}(G_S^{\text{VRG}}, G_T^{\text{VRG}}; \mathcal{X}_T) + \mathcal{L}_{\text{cyc}}^{\text{VRG}}(G_T^{\text{VRG}}, G_S^{\text{VRG}}; \mathcal{X}_S)) \\ & + \lambda_2 (\mathcal{L}_{\text{sem}}^{\text{VRG}}(G_S^{\text{VRG}}; \mathcal{X}_T, f_S^{\text{VRG}}) + \mathcal{L}_{\text{sem}}^{\text{VRG}}(G_S^{\text{VRG}}; \mathcal{X}_S, f_S^{\text{VRG}})) \\ & + \lambda_3 (\mathcal{L}_{\text{shift}}^{\text{VRG}}(G_T^{\text{VRG}}; \mathcal{X}_S) + \mathcal{L}_{\text{shift}}^{\text{VRG}}(G_S^{\text{VRG}}; \mathcal{X}_T)), \text{ where} \end{aligned}$$

- $\mathcal{L}_{\text{GAN}}^{\text{VRG}}$ is the standard GAN loss:

$$\mathcal{L}_{\text{GAN}}^{\text{VRG}}(G_T^{\text{VRG}}, D_T^{\text{VRG}}; \mathcal{X}_S, \mathcal{X}_T) = \mathbb{E}_{\mathbf{x}_T \sim p_{\text{real}}} [\log D_T^{\text{VRG}}(\mathbf{x}_T)] + \mathbb{E}_{\mathbf{x}_S \sim p_{\text{syn}}, \mathbf{z}} [\log (1 - D_T^{\text{VRG}}(G_T^{\text{VRG}}(\mathbf{x}_S)))]$$

and similarly for $\mathcal{L}_{\text{GAN}}^{\text{VRG}}(G_S^{\text{VRG}}, D_S^{\text{VRG}}; \mathcal{X}_T, \mathcal{X}_S)$;

- $\mathcal{L}_{\text{sem}}^{\text{VRG}}$ is the semantic loss as introduced in CyCADA [249]; the idea is that if we have ground truth labels for the synthetic domain \mathcal{X}_S (in this case, we are doing semantic segmentation), we can train a network f_S^{VRG} on \mathcal{X}_S and then use it to generate pseudolabels for the domain \mathcal{X}_T where ground truth is not available; the semantic loss now makes sure that the results (segmentation maps) remain the same after image translation:

$$\begin{aligned} \mathcal{L}_{\text{sem}}^{\text{VRG}}(G_S^{\text{VRG}}; \mathcal{X}_T, f_S^{\text{VRG}}) &= \mathbb{E}_{\mathbf{x}_T \sim p_{\text{real}}} [\text{CE}(f_S^{\text{VRG}}(\mathbf{x}_T), f_S^{\text{VRG}}(G_S^{\text{VRG}}(\mathbf{x}_T)))], \\ \mathcal{L}_{\text{sem}}^{\text{VRG}}(G_T^{\text{VRG}}; \mathcal{X}_S, f_S^{\text{VRG}}) &= \mathbb{E}_{\mathbf{x}_S \sim p_{\text{syn}}} [\text{CE}(f_S^{\text{VRG}}(\mathbf{x}_S), f_S^{\text{VRG}}(G_T^{\text{VRG}}(\mathbf{x}_S)))], \end{aligned}$$

where CE denotes cross-entropy;

- $\mathcal{L}_{\text{shift}}^{\text{VRG}}$ is the *shift loss* that makes the image translation result invariant to shifts:

$$\begin{aligned} \mathcal{L}_{\text{shift}}^{\text{VRG}}(G_T^{\text{VRG}}; \mathcal{X}_S) &= \mathbb{E}_{\mathbf{x}_S, i, j} \left[\left\| G_T^{\text{VRG}}(\mathbf{x}_S)_{\left[\begin{smallmatrix} x \rightarrow i \\ y \rightarrow j \end{smallmatrix}\right]} - G_T^{\text{VRG}}\left(\mathbf{x}_S, \left[\begin{smallmatrix} x \rightarrow i \\ y \rightarrow j \end{smallmatrix}\right]\right) \right\|_2^2 \right], \\ \mathcal{L}_{\text{shift}}^{\text{VRG}}(G_S^{\text{VRG}}; \mathcal{X}_T) &= \mathbb{E}_{\mathbf{x}_T, i, j} \left[\left\| G_S^{\text{VRG}}(\mathbf{x}_T)_{\left[\begin{smallmatrix} x \rightarrow i \\ y \rightarrow j \end{smallmatrix}\right]} - G_S^{\text{VRG}}\left(\mathbf{x}_T, \left[\begin{smallmatrix} x \rightarrow i \\ y \rightarrow j \end{smallmatrix}\right]\right) \right\|_2^2 \right], \end{aligned}$$

where $\mathbf{x}_{\begin{bmatrix} x \rightarrow i \\ y \rightarrow j \end{bmatrix}}$ denotes the shifting operation by i pixels along the X-axis and j pixels along the Y-axis, and i and j are chosen uniformly at random up to the total downsampling factor of the network K (since the result will always be invariant to shifts of multiples of K).

Zhang et al. test their solution on the CARLA navigation benchmark [153] and show significant improvements.

James et al. [279] consider the same kind of approach for robotic grasping. Their model, *Randomized-to-Canonical Adaptation Networks* (RCAN), learn to map heavily randomized simulation images (with random textures) to a canonical (much simpler) rendered image and also map real images to canonical rendered images; interestingly, they achieve good results with a much simpler GAN architecture where additional losses simply bring together the segmentation masks and depth maps for simulated images, and there are no cycle consistency losses. An even simpler approach is taken by Yang et al. [668] who introduce *domain unification* for autonomous driving. Their model, called *DU-Drive*, consists of a generator that translates real images to simplified synthetic images and a discriminator that distinguishes them from actual synthetic images; the driving policy is then trained in the simulator.

Another idea for generating synthetic data from real is to compose parts of real images to produce synthetic ones. We have discussed the cut-and-paste approaches in 5.3; a natural continuation of these ideas would be to use more complex, semantic conditioning with a GAN-based architecture. For example, Joo et al. [293] provide a GAN-based architecture for generating a fusion image, where, say, one input \mathbf{x} provides the identity of a person, another input \mathbf{y} provides the shape (pose) of a person, and the result is $\hat{\mathbf{x}}$ which has the identity of \mathbf{x} and the shape of \mathbf{y} . Their FusionGAN architecture extends CycleGAN-like ideas to losses that distinguish between identity and shape of an image, introducing the concepts of *identity loss* and *shape loss*. FusionGAN relies on a dataset where there are several images with different shapes but the same identity (e.g., the same person in different poses; a dataset of videos would provide a simple example); its overall loss function is

$$\mathcal{L}^{\text{FUS}} = \mathcal{L}_I^{\text{FUS}} + \lambda \mathcal{L}_S^{\text{FUS}}, \text{ where}$$

- $\mathcal{L}_I^{\text{FUS}}$ is the *identity loss*

$$\begin{aligned} \mathcal{L}_I^{\text{FUS}}(G^{\text{FUS}}, D^{\text{FUS}}) &= \mathbb{E}_{\mathbf{x}, \mathbf{x}' \sim p_{\text{real}}(\mathbf{x})} [\|1 - D^{\text{FUS}}(\mathbf{x}, \mathbf{x}')\|_2] \\ &\quad + \mathbb{E}_{\mathbf{x} \sim p_{\text{real}}(\mathbf{x}), \mathbf{y} \sim p_{\text{real}}(\mathbf{y})} [\|D^{\text{FUS}}(\mathbf{x}, G^{\text{FUS}}(\mathbf{x}, \mathbf{y}))\|_2], \end{aligned}$$

i.e., the discriminator D^{FUS} learns to distinguish real pairs of images $(\mathbf{x}, \mathbf{x}')$ with the same identity (but different shapes) and fake pairs of images $(\mathbf{x}, G^{\text{FUS}}(\mathbf{x}, \mathbf{y}))$ where G^{FUS} is supposed to take the identity from \mathbf{x} ;

- $\mathcal{L}_S^{\text{FUS}}$ is the *shape loss* defined as

$$\mathcal{L}_{S_1}^{\text{FUS}}(G^{\text{FUS}}) = \mathbb{E}_{\mathbf{x}, \mathbf{x}' \sim p_{\text{real}}(\mathbf{x})} [\|\mathbf{x}' - G^{\text{FUS}}(\mathbf{x}, \mathbf{x}')\|_1]$$

when \mathbf{x} and \mathbf{x}' have the same identity, and

$$\mathcal{L}_{S_2a}^{\text{FUS}}(G^{\text{FUS}}) = \mathbb{E}_{\mathbf{x} \sim p_{\text{real}}(\mathbf{x}), \mathbf{y} \sim p_{\text{real}}(\mathbf{y})} [\|\mathbf{y} - G^{\text{FUS}}(\mathbf{y}, G^{\text{FUS}}(\mathbf{x}, \mathbf{y}))\|_1],$$

$$\mathcal{L}_{S_2b}^{\text{FUS}}(G^{\text{FUS}}) = \mathbb{E}_{\mathbf{x} \sim p_{\text{real}}(\mathbf{x}), \mathbf{y} \sim p_{\text{real}}(\mathbf{y})} [\|G^{\text{FUS}}(\mathbf{x}, \mathbf{y}) - G^{\text{FUS}}(G^{\text{FUS}}(\mathbf{x}, \mathbf{y}), \mathbf{y})\|_1],$$

i.e., $G^{\text{FUS}}(\mathbf{y}, G^{\text{FUS}}(\mathbf{x}, \mathbf{y}))$ should be the same as \mathbf{y} , with identity from \mathbf{y} and shape also from \mathbf{y} , and $G^{\text{FUS}}(G^{\text{FUS}}(\mathbf{x}, \mathbf{y}), \mathbf{y})$) should be the same as $G^{\text{FUS}}(\mathbf{x}, \mathbf{y})$, with identity from \mathbf{x} and shape from \mathbf{y} .

Similar ideas have been extended to animating still images [547], motion transfer [90], and image-to-image translation [384]. In general, these works belong to an interesting field of generative semantic manipulation with GANs. Important works in this direction include Mask-Contrasting GAN [358] that can modify an object to a different suitable category inside its segmentation mask (e.g., replace a cat with a dog), Attention-GAN [105] that performs the same task with an attention-based architecture, IterGAN [186] that attempts iterative small-scale 3D manipulations such as rotation from 2D images, and others. However, while this field produces very interesting works, so far we have not seen direct applications of such architectures to generating synthetic data. We believe that ideas similar to TP-GAN can also be fruitful in other domains, especially in situations where one- or few-shot learning is required so “smart augmentations” such as rotation can bring significant improvements.

6.2 Domain adaptation at the feature/model level

In the previous sections, we have considered models that perform domain adaptation (DA) at the data level, i.e., one can extract a part of the model that takes as input a data point from the source domain (in our case, a synthetic image) and map it to the target domain (domain of real images). However, the final goal of model design rarely involves the generation of more realistic synthetic images; they are merely a stepping stone to producing models that work better, e.g., in the absence of supervision in the target domain. Therefore, to make better use of synthetic data it makes sense to also consider *feature-level* or *model-level* domain adaptation, i.e., methods that work in the space of features or model weights and never go back to change the actual data.

The simplest approach to domain adaptation would be to share the weights between networks operating on different domains or learn an explicit mapping between them [118, 203]. While we mostly discuss other approaches, we note that simpler techniques based on weight sharing remain relevant for domain adaptation. In particular, Rozantsev et al. [504] recently presented a domain adaptation approach where two similar networks are trained on the source and target domain with special regularizers that bring their weights together; the authors evaluate their approach on synthetic-to-real domain adaptation for drone detection with promising results.

Another approach to model-level domain adaptation is related to mining relatively strong priors from real data that can then inform a model trained on synthetic data, helping fix problematic cases or incongruencies between the synthetic and real datasets. For example, Zhang et al. [698, 699] present a curriculum learning approach to domain adaptation for semantic segmentation of urban scenes. They train a segmentation network on synthetic data (specifically on the GTA dataset; see also Section 3.1) but with a special component in the loss function related to the general label distribution in real images:

$$\mathcal{L}^{\text{CURR}} = \frac{1}{|\mathcal{X}_S|} \sum_{\mathbf{x}_S \in \mathcal{X}_S} \mathcal{L}(\mathbf{y}_S, \hat{\mathbf{y}}_S) + \lambda \frac{1}{|\mathcal{X}_T|} \sum_{\mathbf{x}_T \in \mathcal{X}_T} \sum_k \mathcal{C}(p^k(\mathbf{x}_T), \hat{p}^k(\mathbf{x}_T)),$$

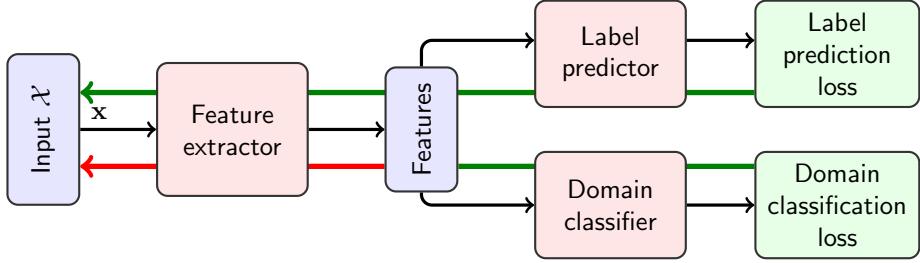


Figure 21: The high-level architecture of model-level domain adaptation from [190, 191]: the gradient flow (green) from the domain classification loss is reversed (becomes red) at the features.

where $\mathcal{L}(\mathbf{y}_S, \hat{\mathbf{y}}_S)$ is the pixel-wise cross-entropy, a standard segmentation loss, and $\mathcal{C}(p^k(\mathbf{x}_T), \hat{p}^k(\mathbf{x}_T))$ is the cross-entropy between the distribution of labels $\hat{p}(\mathbf{x}_T)$ in a real image \mathbf{x}_T that the network produces and $p(\mathbf{x}_T)$ is the real label distribution (superscript k denotes different kinds of label distributions). Note that $p(\mathbf{x}_T)$ is not available in the real data, so this is where curriculum learning comes in: the authors first train on synthetic a simpler model to estimate $p(\mathbf{x}_T)$ from image features and then use it to inform the segmentation model. Recent developments of this interesting direction shift from merely enforcing the label distribution to matching features on multiple different levels [263]. In particular, recent works [357, 680] have introduced the so-called *pyramid consistency loss* instead of $\mathcal{C}(p(\mathbf{x}_T), \hat{p}(\mathbf{x}_T))$ that tries to enforce consistency across domains on the activation maps of later layers of the network.

One of the main directions in model-level domain adaptation was initiated by Ganin and Lempitsky [190] who present a generic framework for unsupervised domain adaptation. Their approach, illustrated on Figure 21, consists of a *feature extractor*, a *label predictor* that performs the necessary task (e.g., classification) on extracted features, and a *domain classifier* that takes the same features and attempts to classify which domain the original input belonged to. The idea is to train the label predictor to perform as well as possible and at the same time train the domain classifier to perform as badly as possible; this is achieved with *gradient reversal*, i.e., multiplying the gradients by a negative constant as they pass from the domain classifier to the feature extractor. In a subsequent work, Ganin et al. [191] generalized this domain adaptation approach to arbitrary architectures and experimented with DA in different domains, including image classification, person re-identification, and sentiment analysis. We also note extensions and similar approaches to domain adaptation developed in [376, 377, 603] and the domain confusion metric that helps produce domain-invariant representation [604], but proceed to highlight the works that perform specifically synthetic-to-real domain adaptation.

Many general model-level domain adaptation approaches have been validated or subsequently extended to synthetic-to-real domain adaptation. Xu et al. [664] consider the pedestrian detection problem (this work is a continuation of [612], see Section 2.2.2). They adapt detectors trained on virtual datasets with a boosting-based procedure, assigning larger weights to samples that are similar to target domain ones. Sun and Saenko [567] propose a domain adaptation approach based on decorrelating the features of a classifier, both

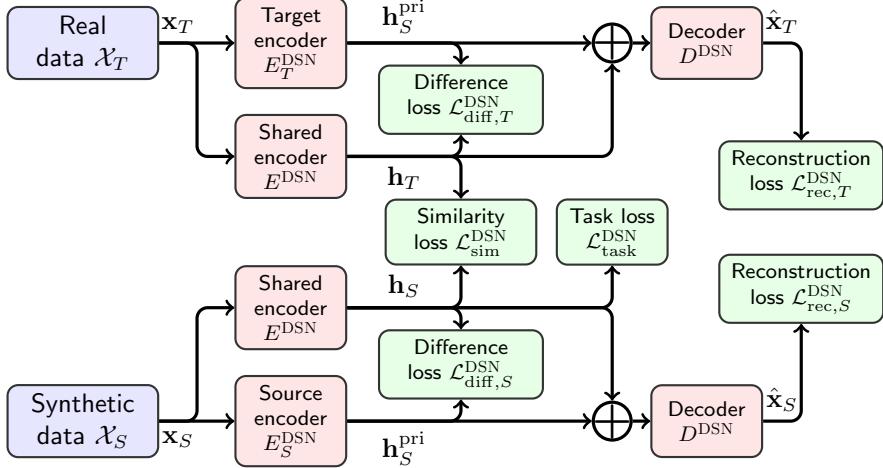


Figure 22: Architecture of the domain separation network [72]. Blocks with identical labels have shared weights.

in unsupervised and supervised settings. Later, López et al. [378] extended the SA-SVVM domain adaptation used in [612] to train deformable part-based models, using synthetic pedestrians from the SYNTHIA dataset (see Section 3.1) as the main example. In a parallel paper, the authors of SYNTHIA Ros et al. [500] used a simple domain adaptation technique called Balanced Gradient Contribution [501], where training on synthetic data is regularized by the gradient obtained on a (small) real dataset, to further improve their results on segmentation aided by synthetic data. Ren et al. [490] perform cross-domain self-supervised multi-task learning with synthetic images: their model predicts several parameters of an image (surface normal, depth, and instance contour) and at the same time tries to minimize the difference between synthetic and real data in feature space.

Domain separation networks by Bousmalis et al. [72], illustrated on Fig. 22, explicitly separate the shared and private components of both source and target domains. Specifically, they introduce a shared encoder $E^{\text{DSN}}(\mathbf{x})$ and two private encoders, E_S^{DSN} for the source domain and E_T^{DSN} for the target domain. The total objective function for a domain separation network is

$$\mathcal{L}^{\text{DSN}} = \mathcal{L}_{\text{task}} + \lambda_1 \mathcal{L}_{\text{rec}} + \lambda_2 \mathcal{L}_{\text{diff}} + \lambda_3 \mathcal{L}_{\text{sim}}, \text{ where}$$

- $\mathcal{L}_{\text{task}}^{\text{DSN}}$ is the supervised task loss in the source domain, e.g., for the image classification task it is

$$\mathcal{L}_{\text{task}}^{\text{DSN}} = -\mathbb{E}_{\mathbf{x}_S \sim p_{\text{syn}}} [\mathbf{y}^\top \log f^{\text{DSN}}(E^{\text{DSN}}(\mathbf{x}_S))],$$

where f^{DSN} is the classifier operating on the output of the shared encoder;

- $\mathcal{L}_{\text{rec}}^{\text{DSN}}$ is the reconstruction loss defined as the difference between the original samples \mathbf{x}_S and \mathbf{x}_T and the results of a shared decoder D^{DSN} that tries to reconstruct the images from a combination of shared and private

representations:

$$\begin{aligned}\mathcal{L}_{\text{rec}}^{\text{DSN}} = & -\mathbb{E}_{\mathbf{x}_S \sim p_{\text{syn}}} [\mathcal{L}_{\text{sim}}(\mathbf{x}_S, D^{\text{DSN}}(E_S^{\text{DSN}}(\mathbf{x}_S) + E_S^{\text{DSN}}(\mathbf{x}_S)))] \\ & -\mathbb{E}_{\mathbf{x}_T \sim p_{\text{real}}} [\mathcal{L}_{\text{sim}}(\mathbf{x}_T, D^{\text{DSN}}(E_T^{\text{DSN}}(\mathbf{x}_T) + E_T^{\text{DSN}}(\mathbf{x}_T)))]\end{aligned}$$

for some similarity metric \mathcal{L}_{sim} ;

- $\mathcal{L}_{\text{diff}}^{\text{DSN}}$ is the difference loss that encourages the hidden shared representations of instances from the source and target domains $E_S^{\text{DSN}}(\mathbf{x}_S)$ and $E_T^{\text{DSN}}(\mathbf{x}_T)$ to be orthogonal to their corresponding private representations $E_S^{\text{DSN}}(\mathbf{x}_S)$ and $E_T^{\text{DSN}}(\mathbf{x}_T)$; in [72], the difference loss is defined as

$$\mathcal{L}_{\text{diff}}^{\text{DSN}} = \left\| H_S^\top H_S^{\text{pri}} \right\|_F^2 + \left\| H_T^\top H_T^{\text{pri}} \right\|_F^2,$$

where H_S is the matrix of $E_S^{\text{DSN}}(\mathbf{x}_S)$, H_S^{pri} is the matrix of $E_S^{\text{DSN}}(\mathbf{x}_S)$, and similarly for H_T and H_T^{pri} ;

- $\mathcal{L}_{\text{sim}}^{\text{DSN}}$ is the similarity loss that encourages the hidden shared representations from the source and target domains $E_S^{\text{DSN}}(\mathbf{x}_S)$ and $E_T^{\text{DSN}}(\mathbf{x}_T)$ to be similar to each other, i.e., indistinguishable by a domain classifier trained through the gradient reversal layer as in [190]; in [72], this loss is composed of the cross-entropy for the domain classifier and maximal mean discrepancy (MMD) [211] for the hidden representations themselves.

Bousmalis et al. evaluate their model on several synthetic-to-real scenarios, e.g., on synthetic traffic signs from [416] and synthetic objects from the *LineMod* dataset [638].

Domain separation networks became one of the first major examples in domain adaptation with *disentanglement*, where the hidden representations are domain-invariant and some of the features can be changed to transition from one domain to another. Further developments include asymmetric training for unsupervised domain adaptation [514], DistanceGAN for one-sided domain mapping [52], co-regularized alignment [334], cross-domain autoencoders [206], multisource domain adversarial networks [705], continuous cross-domain translation [367], face recognition adaptation from images to videos with the help of synthetic augmentations [554], and more [689]; all of these advances may be relevant for synthetic-to-real domain adaptation but we will highlight some works that are already doing adaptation between these two domains.

The popular and important domains for feature-based domain adaptation are more or less the same as in domain adaptation on the data level, but feature-based DA may be able to handle higher-dimensional inputs and more complex scenes because the adaptation itself is done in an intermediate lower-dimensional space. As an illustrative example, let us consider feature-based DA for computer vision problems for outdoor scenes (see also Section 3.1). In their *FCNs in the Wild* model, Hoffman et al. [250] consider feature-based DA for semantic segmentation with fully convolutional networks (FCN) where ground truth is available for the source domain (synthetic data) but unavailable for the target domain (real data). Their unsupervised domain adaptation framework contains a feature extractor f^{FCNW} and the joint objective function

$$\mathcal{L}^{\text{FCNW}} = \mathcal{L}_{\text{seg}}^{\text{FCNW}} + \mathcal{L}_{\text{DA}}^{\text{FCNW}} + \mathcal{L}_{\text{MI}}^{\text{FCNW}}, \text{ where}$$

- $\mathcal{L}_{\text{seg}}^{\text{FCNW}}$ is the standard supervised segmentation objective on the source domain, where supervision is available;
- $\mathcal{L}_{\text{DA}}^{\text{FCNW}}$ is the domain alignment objective that minimizes the observed source and target distance in the representation space by training a discriminator (domain classifier) to distinguish instances from source and target domains; an interesting new idea here is to take as an instance for this objective not the entire image but a cell from a coarse grid that corresponds to the of high-level features that domain adaptation is supposed to bring together;
- $\mathcal{L}_{\text{MI}}^{\text{FCNW}}$ is the multiple instance loss that encourages pixels to be assigned to class c in such a way that the percentage of an image labeled with c remains within the expected range derived from the source domain.

Another direction of increasing the input dimension is to move from images to videos. Xu et al. [663] use adversarial domain adaptation to transfer object detection models—single-shot multi-box detector (SSD) [372] and multi-scale deep CNN (MSCNN) [81]—from synthetic samples to real videos in the smoke detection problem.

Chen et al. [107] construct the *Cross City Adaptation* model that brings together the features from different domains, again with semantic segmentation of outdoor scenes in mind. Their framework optimizes the joint objective function

$$\mathcal{L}^{\text{CCA}} = \mathcal{L}_{\text{task}}^{\text{CCA}} + \mathcal{L}_{\text{global}}^{\text{CCA}} + \mathcal{L}_{\text{class}}^{\text{CCA}}, \text{ where}$$

- $\mathcal{L}_{\text{task}}^{\text{CCA}}$ is the task loss, in this case cross-entropy between predicted and ground truth segmentation masks in the source domain;
- $\mathcal{L}_{\text{global}}^{\text{CCA}}$ is the global domain alignment loss, again defined as fooling the domain discriminator similar to *FCNs in the Wild*;
- $\mathcal{L}_{\text{class}}^{\text{CCA}}$ is the class-wise domain alignment loss, where grid cells are assigned soft class labels (extracted from the truth in the source domain and predicted in the target domain), and the domain classifiers and discriminators are trained and applied *class-wise*, separately.

As the title suggests, *Cross City Adaptation* is intended to adapt outdoor segmentation models trained on one city to other cities, but Chen et al. also apply it to synthetic-to-real domain adaptation from SYNTHIA to Cityscapes (see Section 3.1), achieving noticeable gains in segmentation quality.

Hong et al. [252] provide one of the most direct and most promising applications of feature-level synthetic-to-real domain adaptation. In their *Structural Adaptation Network*, the conditional generator $G_{\theta}^{\text{SDA}}(\mathbf{x}_S, \mathbf{z})$ takes as input the features $f_l^{\text{SDA}}(\mathbf{x}_S)$ from a low-level layer of the feature extractor (i.e., features with fine-grained details) and random noise \mathbf{z} and produces transformed feature maps that should be similar to feature maps extracted from real images. To achieve that, G_{θ}^{SDA} produces a noise map $\hat{G}_{\theta}^{\text{SDA}}(f_l^{\text{SDA}}(\mathbf{x}_S), \mathbf{z})$ and then adds it to high-level features: $G_{\theta}^{\text{SDA}}(\mathbf{x}_S, \mathbf{z}) = f_h^{\text{SDA}}(\mathbf{x}_S) + \hat{G}_{\theta}^{\text{SDA}}(f_l^{\text{SDA}}(\mathbf{x}_S), \mathbf{z})$. The optimization problem is

$$\min_{\theta, \theta'} \max_{\phi} (\mathcal{L}_{\text{GAN}}^{\text{SDA}}(G_{\theta}^{\text{SDA}}, D_{\phi}^{\text{SDA}}) + \lambda \mathcal{L}_{\text{task}}^{\text{SDA}}(G_{\theta}^{\text{SDA}}, T_{\theta'}^{\text{SDA}})), \text{ where}$$

- $\mathcal{L}_{\text{GAN}}^{\text{SDA}}$ is the GAN loss in the feature space:

$$\mathcal{L}_{\text{GAN}}^{\text{SDA}}(G_{\theta}^{\text{SDA}}, D_{\phi}^{\text{SDA}}) = \mathbb{E}_{\mathbf{x}_T \sim p_{\text{real}}} [\log D_{\phi}^{\text{SDA}}(\mathbf{x}_T)] + \mathbb{E}_{\mathbf{x}_S \sim p_{\text{syn}}, \mathbf{z}} [\log (1 - D_{\phi}^{\text{SDA}}(G_{\theta}^{\text{SDA}}(\mathbf{x}_S, \mathbf{z})))];$$

- $\mathcal{L}_{\text{task}}^{\text{SDA}}$ is the task loss for the pixel-wise classifier $T_{\theta'}^{\text{SDA}}$ which is trained end-to-end, together with the rest of the architecture; the task loss is defined as the pixel-wise cross-entropy between the segmentation mask $T_{\theta'}^{\text{SDA}}(G_{\theta}^{\text{SDA}}(\mathbf{x}_S, \mathbf{z}))$ produced by $T_{\theta'}^{\text{SDA}}$ on adapted features and the ground truth synthetic segmentation mask \mathbf{y}_S .

Hong et al. compare the Structural Adaptation Network with other state of the art approaches, including FCNs in the Wild [250] and cross-city adaptation [107], with source domain datasets SYNTHIA and GTA and target domain dataset Cityscapes; they conclude that this adaptation significantly improves the results for semantic segmentation of urban scenes.

To summarize, feature-level domain adaptation provides interesting opportunities for synthetic-to-real adaptation, but these methods still mostly represent work in progress. In our experience, feature- and model-level DA is usually a simpler and more robust approach, easier to get to work, so we expect new exciting developments in this direction and recommend to try this family of methods for synthetic-to-real DA (unless actual refined images are required).

6.3 Domain adaptation for control and robotics

In the field of control, joint domain adaptation is usually intended to transfer control policies learned in a simulated environment to a real setting. As we have already discussed in Sections 3.2 and 3.1, simulated environments are almost inevitable in reinforcement learning for robotics, as they allow to scale the datasets up compared to real data and cover a much wider range of situations than real data that could be used for imitational learning (see also the survey [574]). In this setting, domain adaptation is performed either for the control itself or jointly for the control and synthetic data. The field began even before deep learning; for instance, Saxena et al. [522] learned a model for estimating grasp locations for previously unseen objects on synthetic data.

In Cutler et al. [129], the results of training on a simulator serve as a prior for subsequent learning in the real world. Moreover, in [130, 131] Cutler et al. proceed to multifidelity simulators, training the reinforcement learning agent in a series of simulators with increasing realism; we note this idea as potentially fruitful for other domains as well.

DeepMind researchers Rusu et al. [509] studied the possibility for transfer learning from simulated environments to the real world in the context of end-to-end reinforcement learning for robotics. They use the general idea of *progressive networks* [508], an architecture designed for multitask learning and transfer where each subsequent column in the network solves a new task and receives as input the hidden activations from previous columns. Rusu et al. present a modification of this idea for robot transfer learning, then train the first column in the MuJoCo physics simulator [589], and then transfer to a real *Jaco* robotic arm, using the *Asynchronous Advantage Actor-Critic* (A3C) framework for reinforcement learning [413]. The authors report improved results for progressive networks compared to simple transfer via fine-tuning.

There are plenty of works that consider similar kinds of transfer learning, known in robotics as closing the *reality gap*. In particular, Bousmalis et al. [70] use a simulated environment to learn robotic grasping with a domain adaptation model called GraspGAN that makes synthetic images more realistic with a refiner (see Section 6.1.2); they argue that the added realism improves the results for control transfer. Tzeng et al. [602] propose a framework that combines supervised domain adaptation (that requires paired images) and unsupervised DA (that aligns the domains on the level of distributions); to do that, they introduce the notion of a “weak pairing” between images in the source and target domains and learn to find matching synthetic images to produce aligned data. The resulting model is successfully applied to training a visuomotor policy for real robots. Pan et al. [676] consider sim-to-real translation for autonomous driving; they convert synthetic images to a scene parsing representation and then generate a realistic image by a generator corresponding to this parsing representation; the reinforcement learning agent receives this image as part of its driving environment. An even simpler approach, taken, e.g., by Xu et al [577], would be to directly use the segmentation masks as input for the RL agent.

Researchers from *Wayve* Bewley et al. [55] perform domain adaptation for learning to drive from simulation; they claim to present the first end-to-end driving policy transferred from an (obviously supervised) synthetic setting to the fully unsupervised real domain. Their model does image translation and control transfer at the same time, learning the control on a jointly learned latent embedding space. The architecture consists of two encoders E_S^{WVE} and E_T^{WVE} , two generators G_S^{WVE} and G_T^{WVE} , two discriminators D_S^{WVE} and D_T^{WVE} , and a controller C^{WVE} . The image translator follows the MUNIT architecture [370], with two convolutional variational autoencoder networks that swap the latent embeddings to translate between domains, i.e., for $\mathbf{x}_S \sim \mathcal{X}_S$, $\mathbf{x}_T \sim \mathcal{X}_T$

$$\begin{aligned} \mathbf{z}_S &= E_S^{\text{WVE}}(\mathbf{x}_S) + \epsilon, & \hat{\mathbf{x}}_S &= G_S^{\text{WVE}}(\mathbf{z}_S), \\ \mathbf{z}_T &= E_T^{\text{WVE}}(\mathbf{x}_T) + \epsilon, & \hat{\mathbf{x}}_T &= G_T^{\text{WVE}}(\mathbf{z}_T), \end{aligned}$$

and the translation is to compute \mathbf{z}_S with E_S^{WVE} and then predict $\hat{\mathbf{x}}$ with G_T^{WVE} and vice versa. The overall generator loss function is

$$\begin{aligned} \mathcal{L}^{\text{WVE}} = & \lambda_0 \mathcal{L}_{\text{rec}}^{\text{WVE}} + \lambda_1 \mathcal{L}_{\text{cyc}}^{\text{WVE}} + \lambda_2 \mathcal{L}_{\text{ctrl}}^{\text{WVE}} + \lambda_3 \mathcal{L}_{\text{cycl}}^{\text{WVE}} + \\ & + \lambda_4 \mathcal{L}_{\text{LSGAN}}^{\text{WVE}} + \lambda_5 \mathcal{L}_{\text{perc}}^{\text{WVE}} + \lambda_6 \mathcal{L}_{\text{zrec}}^{\text{WVE}}, \quad \text{where} \end{aligned}$$

- $\mathcal{L}_{\text{rec}}^{\text{WVE}}$ is the L_1 image reconstruction loss in both domains:

$$\begin{aligned} \mathcal{L}_{\text{rec}}^{\text{WVE}}(\mathbf{x}_S) &= \|G_S^{\text{WVE}}(E_S^{\text{WVE}}(\mathbf{x}_S)) - \mathbf{x}_S\|_1, \\ \mathcal{L}_{\text{rec}}^{\text{WVE}}(\mathbf{x}_T) &= \|G_T^{\text{WVE}}(E_T^{\text{WVE}}(\mathbf{x}_T)) - \mathbf{x}_T\|_1; \end{aligned}$$

- $\mathcal{L}_{\text{cyc}}^{\text{WVE}}$ is the cycle consistency loss for both domains:

$$\mathcal{L}_{\text{cyc}}^{\text{WVE}}(\mathbf{x}_S) = \|G_S^{\text{WVE}}(E_T^{\text{WVE}}(G_T^{\text{WVE}}(E_S^{\text{WVE}}(\mathbf{x}_S)))) - \mathbf{x}_S\|_1$$

and similar for \mathcal{X}_T ;

- $\mathcal{L}_{\text{ctrl}}^{\text{WVE}}$ is the control loss that compares the controls produced on the training set with the autopilot: $\mathcal{L}_{\text{ctrl}}^{\text{WVE}}(\mathbf{x}_S) = \|C^{\text{WVE}}(E_S^{\text{WVE}}(\mathbf{x}_S)) - \mathbf{c}\|_1$ for ground truth control \mathbf{c} , and similar for \mathcal{X}_T ;

- $\mathcal{L}_{\text{cyctrl}}^{\text{WVE}}$ is the control cycle consistency loss that makes the controls similar for images translated to another domain:

$$\mathcal{L}_{\text{cyctrl}}^{\text{WVE}}(\mathbf{x}_S) = \|C^{\text{WVE}}(E_T^{\text{WVE}}(G_S^{\text{WVE}}(E_S^{\text{WVE}}(\mathbf{x}_S)))) - C^{\text{WVE}}(E_S^{\text{WVE}}(\mathbf{x}_S))\|_1,$$

and similar for \mathcal{X}_T ;

- $\mathcal{L}_{\text{LSGAN}}^{\text{WVE}}$ is the LSGAN adversarial loss applied to both generator-discriminator pairs (see above);
- $\mathcal{L}_{\text{perc}}^{\text{WVE}}$ is the perceptual loss (see above) for both image translation directions with instance normalization applied before, as shown in [266];
- $\mathcal{L}_{\text{zrec}}^{\text{WVE}}$ is the latent reconstruction loss: $\mathcal{L}_{\text{zrec}}^{\text{WVE}}(\mathbf{z}_S) = \|E_T^{\text{WVE}}(G_T^{\text{WVE}}(\mathbf{z}_S)) - \mathbf{z}_S\|_1$ and similar for \mathcal{X}_T .

Bewley et al. compare their approach with a number of transfer learning baselines, show excellent results for end-to-end learning to drive, and even perform real world experiments with the trained policy. Similar techniques have been used without synthetic data in the loop as well; e.g., Wulfmeier et al. [650] use a similar model for domain adaptation to handle appearance changes in outdoor robotics, i.e., changes in weather conditions, lighting, and the like.

We have already discussed the works of Inoue et al. [273], Zhang et al. [690], James et al. [279], and Yang et al. [668] who make real data more similar to synthetic for computer vision problems related to robotic grasping and visual navigation (see Section 6.1.3). Importantly, these models are not merely translating images but are also tested on real world robots. Zhang et al. not only show improvements in semantic segmentation results but also conduct real-world robotic experiments for indoor and outdoor visual navigation tasks, first training a navigation policy in a simulated environment and then directly deploying it on a robot in a real environment, while James et al. test their solution on a real robotic hand, training the *QT-Opt* policy [301] to grasp from a simulation with 5000 additional real life grasping episodes better than the same policy trained on 580,000 real episodes, a more than 99% reduction in required real world input.

Another direction where synthetic data might be useful for learning control is to generate synthetic behaviours to improve imitation learning [433]. Bansal et al. [35] discuss the insufficient data problem in imitation learning: for learning to drive, even 30 million real-world expert driving examples that combine into more than 60 days of driving is not sufficient to train an end-to-end driving model. To alleviate this lack of data, they present their imitation learning framework *ChauffeurNet* with data where synthetic perturbations have been introduced to expert driving examples. This allows to cover corner cases such as collisions and off-road driving, i.e., bad examples that should be avoided but that are lacking in expert examples altogether. Interestingly, perturbations are introduced into intermediate representations rather than in raw sensor input or controller outputs.

To sum up, closing the reality gap is one of the most important problems in the field of control and robotics. Important breakthroughs in this direction appear constantly, but there is still some way to go before self-driving cars and robotic arms are able to train in a simulated environment and then perfectly transfer these skills to the real world.

6.4 Case study: GAN-based domain adaptation for medical imaging

Medical imaging is a field where labeled data is especially hard to come by. First, while manual labeling is hard and expensive enough for regular computer vision problems, in medical imaging it is far more expensive because it cannot be crowdsourced to anonymous annotators: for most problems, medical imaging data can only be reliably labeled by a trained professional, often with a medical degree. Second, for obvious privacy reasons it is very hard to arrange for publishing real datasets, and collecting a large enough labeled dataset to train a standard object detection or segmentation model would in many cases require a concerted effort from several different hospitals; thus, with the exception of public competitions, most papers in the field use private datasets and are not allowed to share their data. Third, some pathologies simply do not have sufficiently large and diverse datasets collected yet. At the same time, often there are relatively large generic datasets available, e.g., of healthy tissue but not of a specific pathology of interest.

While we emphasize GAN-based generation methods, we note that there have been successful attempts to use rendered synthetic data for medical imaging tasks that are based on recent developments in medical visualization and rendering tools. For example, Mahmood et al. [388] use the recently developed cinematic rendering technique for CT [167] (a photorealistic simulation of the propagation of light through tissue) to train a CNN for depth estimation in endoscopy data.

GANs have been widely applied to generating realistic medical images [40, 46,326,423,666,673]. Moreover, since the images are domain-specific, the quality of GAN-produced images has relatively quickly reached the level where it can in many applications pass the “visual Turing test”, fooling even trained specialists; see, e.g., lung nodule samples generated in [120] or magnetic resonance (MR) images of the brain in [82, 227]. Therefore, it is no wonder that GAN-based domain adaptation (DA) techniques, especially based on fusing and augmenting real images, are increasingly finding their way into medical imaging. In this section, we give a brief overview of recent work in this domain.

In some works, synthetic data is generated from scratch, i.e., GANs are trained to convert random noise into synthetic images (see also Section 5.4). Frid-Adar et al. [180, 181] used two standard GAN architectures, Deep Convolutional GAN (DCGAN) [472] and Auxiliary Classifier GAN (ACGAN) [428] with class label auxiliary information, to generate synthetic computed tomography (CT) images of liver lesions. They report significantly improved results in image classification with CNNs when training on synthetic data compared to standard augmentations of their highly limited dataset (182 2D scans divided into three types of lesions). Baur et al. [666] attempt high resolution skin lesion synthesis, comparing several GAN architectures and obtaining highly realistic results even with a small training dataset. Han et al. [229, 230] concentrate on brain magnetic resonance (MR) images. They use progressively growing GANs (PGGAN) [311] to generate 256×256 MR images and then compare two different refinement approaches: SimGAN [545] as discussed in Section 6.1.1 and UNIT [370], an unsupervised image-to-image translation architecture that maps each domain into a shared latent space with a VAE-GAN architecture [345] (we remark that the original paper [370] also applies UNIT, among other things, to

synthetic-to-real translation). Han et al. report improved results when combining GAN-based synthetic data with classic domain adaptation techniques. Neff [422] uses a slightly different approach: to generate synthetic data for segmentation, he uses a standard WGAN-GP architecture [217] but generates image-segmentation pairs, i.e., images with an additional channel that shows the segmentation mask. Neff reports improved segmentation results with U-Net [498] after augmenting a real dataset with synthetic image-segmentation pairs. Mahmood et al. [387] show an interesting take on the problem by doing the reverse: they make real medical images look more like synthetic images in order to then apply a network trained on synthetic data (see also Section 6.1.3). With this approach, they improve state of the art results in depth estimation for endoscopy images.

In general, segmentation problems in medical imaging are especially hard to label, and segmentation data is especially lacking in many cases. In this context, recent works have often employed conditional GANs and pix2pix models to generate realistic images from randomized segmentation masks. For example, Bailo et al. [31] consider red blood cell image generation with the *pix2pixHD* model [627]. Namely, their conditional GAN optimizes

$$\begin{aligned} & \min_{G^{p2p}} \left[\max_{D_1^{p2p}, D_2^{p2p}} [\mathcal{L}_{GAN}^{p2p}(G^{p2p}, D_1^{p2p}) + \mathcal{L}_{GAN}^{p2p}(G^{p2p}, D_2^{p2p})] + \right. \\ & \quad \left. + \lambda_1 (\mathcal{L}_{FM}^{p2p}(G^{p2p}, D_1^{p2p}) + \mathcal{L}_{FM}^{p2p}(G^{p2p}, D_2^{p2p})) + \lambda_2 \mathcal{L}_{PR}^{p2p}(G^{p2p}(\mathbf{s}, E^{p2p}(\mathbf{x})), \mathbf{x}) \right], \end{aligned}$$

where:

- \mathbf{x} is an input image, \mathbf{s} is a segmentation mask (it serves as input to the generator), D_1^{p2p} and D_2^{p2p} are two discriminators that have the same architecture but operate on different image scales (original and 2x down-sampled), $\mathcal{L}_{GAN}^{p2p}(G, D)$ is the regular GAN loss;
- $E^{p2p}(\mathbf{x})$ is the feature encoder network that encodes low-level features of the objects with instance-wise pooling; its output is fed to the generator $G^{p2p}(\mathbf{s}, E^{p2p}(\mathbf{x}))$ and can be used to manipulate object style in generated images (see [627] for more details);
- \mathcal{L}_{FM}^{p2p} is the *feature matching loss* that makes features at different layers of the discriminators (we denote the input to the i th layer of D as $D^{(i)}$) match for \mathbf{x} and $G(\mathbf{s})$:

$$\mathcal{L}_{FM}^{p2p}(G, D) = \mathbb{E}_{(\mathbf{s}, \mathbf{x})} \left[\sum_{i=1}^L \frac{1}{N_i} \|D^{(i)}(\mathbf{s}, \mathbf{x}) - D^{(i)}(\mathbf{s}, G(\mathbf{s}, E^{p2p}(\mathbf{x})))\|_1 \right];$$

- \mathcal{L}_{PR}^{p2p} is the *perceptual reconstruction loss* for some feature encoder F :

$$\mathcal{L}_{PR}^{p2p}(G, D) = \mathbb{E}_{(\mathbf{s}, \mathbf{x})} \left[\sum_{i=1}^{L'} \frac{1}{M_i} \|F^{(i)}(\mathbf{x}) - F^{(i)}(G(\mathbf{s}, E^{p2p}(\mathbf{x})))\|_1 \right].$$

The real dataset in [31] consisted of only 60 manually annotated 1920×1200 RGB images (with another 40 images used for testing), albeit with a lot of

annotated objects (669 blood cells per image on average). Bailo et al. also developed a scheme for sampling randomized but realistic segmentation masks to use for synthetic data generation. They report improved segmentation results with FCN and improved detection with Faster R-CNN when trained on a combination of real and synthetic data.

Zhao et al. [706] consider the problem of generating filamentary structured images, such as retinal fundus and neuronal images, from a ground truth segmentation map, with an emphasis on generating images in multiple different styles. Their FILA-sGAN approach is based on GAN-based image style transfer ideas [287, 607]. Its generator loss function is

$$\mathcal{L}^{\text{FIL}} = \mathcal{L}_{\text{GAN}}^{\text{FIL}}(G^{\text{FIL}}, D^{\text{FIL}}) + \lambda_1 \mathcal{L}_{\text{cont}}^{\text{FIL}}(G^{\text{FIL}}) + \lambda_2 \mathcal{L}_{\text{sty}}^{\text{FIL}}(G^{\text{FIL}}) + \lambda_3 \mathcal{L}_{\text{TV}}^{\text{FIL}}(G^{\text{FIL}}), \text{ where}$$

- $G^{\text{FIL}} : \mathbf{y} \rightarrow \hat{\mathbf{x}}$ is a generator that takes a binary image and produces a “phantom” $\hat{\mathbf{x}}$, D^{FIL} is a synthetic vs. real discriminator, and $\mathcal{L}_{\text{GAN}}^{\text{FIL}}$ is the standard GAN loss;
- $\mathcal{L}_{\text{cont}}^{\text{FIL}}$ is the *content loss* that makes the filamentary structure of a generated phantom $\hat{\mathbf{x}}$ match the real raw image \mathbf{x} , evidenced through the features $\phi^{(i)}$ for some standard CNN feature extractor such as VGG-19:

$$\mathcal{L}_{\text{cont}}^{\text{FIL}}(G^{\text{FIL}}) = \sum_l \frac{1}{W_l H_l} \left\| \phi^{(l)}(\mathbf{x}) - \phi^{(l)}(\hat{\mathbf{x}}) \right\|_F^2,$$

where \mathbf{x} is the real raw image, l spans the CNN blocks and layers, W_l and H_l are the width and height of the corresponding feature maps, and $\|\cdot\|_F$ is the Frobenius matrix norm;

- $\mathcal{L}_{\text{sty}}^{\text{FIL}}$ is the *style loss* that minimizes the textural difference between $\hat{\mathbf{x}}$ and a style image \mathbf{x}_s :

$$\mathcal{L}_{\text{sty}}^{\text{FIL}}(G^{\text{FIL}}) = \sum_l \frac{\omega_l}{W_l H_l} \left\| \mathbb{G}^{(l)}(\mathbf{x}_s) - \mathbb{G}^{(l)}(\hat{\mathbf{x}}) \right\|_F^2,$$

where \mathbf{x}_s is the style image, $\mathbb{G}^{(l)}$ is the Gram matrix of the features in CNN block l , and ω_l is its weight (a hyperparameter);

- $\mathcal{L}_{\text{TV}}^{\text{FIL}}$ is the *total variation loss* that serves as a regularizer and encourages $\hat{\mathbf{x}}$ to be smooth:

$$\mathcal{L}_{\text{TV}}^{\text{FIL}}(G^{\text{FIL}}) = \sum_{i,j} \left(\|\hat{\mathbf{x}}_{i,j+1} - \hat{\mathbf{x}}_{i,j}\|_2^2 + \|\hat{\mathbf{x}}_{i+1,j} - \hat{\mathbf{x}}_{i,j}\|_2^2 \right).$$

As a result, Zhao et al. report highly realistic filamentary structured images generated from a segmentation map and a single style image in a variety of different styles. Importantly for us, they also report improved segmentation results with state of the art approaches to the corresponding segmentation task.

In other works, Hou et al. [253] use GANs to refine synthesized histopathology images (similar to SimGAN discussed in Section 6.1.1), with improved nucleus segmentation and glioma classification results. Tang et al. [578] use the *pix2pix* model [275] to generate realistic computed tomography (CT) images

from customized lymph node masks, reporting improved lymph node segmentation with U-Net [498]. In [579], the same researchers use the MUNIT (multi-modal image-to-image translation) model [266] to generate realistic chest X-rays with custom abnormalities, reporting improved segmentation with both U-Net and their developed model XLSor. Han et al. [228] were the first to apply 3D GAN-based DA to produce data for 3D object detection, i.e., bounding boxes; they use it in the context of synthetizing CT images of lung nodules.

In a related approach, synthetic data can be generated from real data, but in a different domain. For example, Zhang et al. [704] learn a CycleGAN-based architecture [714] to learn volume-to-volume (i.e., 3D) translation for unpaired datasets of CT and MR images (domain A and domain B respectively). Moreover, they augment the basic CycleGAN with segmentors S_A^{VOL} and S_B^{VOL} that help preserve segmentation mask consistency. In total, their model optimizes the loss function

$$\begin{aligned}\mathcal{L}^{\text{VOL}} = \mathcal{L}_{\text{GAN}}^{\text{VOL}}(G_A^{\text{VOL}}, D_A^{\text{VOL}}) + \mathcal{L}_{\text{GAN}}^{\text{VOL}}(G_B^{\text{VOL}}, D_B^{\text{VOL}}) + \lambda_1 \mathcal{L}_{\text{cyc}}^{\text{VOL}}(G_A^{\text{VOL}}, G_B^{\text{VOL}}) + \\ + \lambda_2 \mathcal{L}_{\text{shape}}^{\text{VOL}}(S_A^{\text{VOL}}, S_B^{\text{VOL}}, G_A^{\text{VOL}}, G_B^{\text{VOL}}), \quad \text{where}\end{aligned}$$

- $G_A^{\text{VOL}} : B \rightarrow A$ and $G_B^{\text{VOL}} : A \rightarrow B$ are CycleGAN generators, and D_A^{VOL} and D_B^{VOL} are discriminators in domains A and B respectively, trained to distinguish between real and synthetic (generated by G^{VOL}) images by the standard GAN loss function $\mathcal{L}_{\text{GAN}}^{\text{VOL}}$;
- $\mathcal{L}_{\text{cyc}}^{\text{VOL}}$ is the cycle consistency loss

$$\begin{aligned}\mathcal{L}_{\text{cyc}}^{\text{VOL}}(G_A^{\text{VOL}}, G_B^{\text{VOL}}) = \mathbb{E}_{\mathbf{x}_A} [\|G_A^{\text{VOL}}(G_B^{\text{VOL}}(\mathbf{x}_A)) - \mathbf{x}_A\|_1] \\ + \mathbb{E}_{\mathbf{x}_B} [\|G_B^{\text{VOL}}(G_A^{\text{VOL}}(\mathbf{x}_B)) - \mathbf{x}_B\|_1],\end{aligned}$$

- $S_A^{\text{VOL}} : A \rightarrow Y$ and $S_B^{\text{VOL}} : B \rightarrow Y$ are segmentors that produce 3D segmentation masks, and $\mathcal{L}_{\text{shape}}^{\text{VOL}}$ is the shape consistency loss

$$\begin{aligned}\mathcal{L}_{\text{shape}}^{\text{VOL}}(S_A^{\text{VOL}}, S_B^{\text{VOL}}, G_A^{\text{VOL}}, G_B^{\text{VOL}}) = \\ \mathbb{E}_{\mathbf{x}_B} \left[-\frac{1}{N} \sum_i \mathbf{y}_B^i \log S_A^{\text{VOL}}(G_A^{\text{VOL}}(\mathbf{x}_B))_i \right] \\ + \mathbb{E}_{\mathbf{x}_A} \left[-\frac{1}{N} \sum_i \mathbf{y}_B^i \log S_A^{\text{VOL}}(G_A^{\text{VOL}}(\mathbf{x}_B))_i \right],\end{aligned}$$

where \mathbf{y}_A and \mathbf{y}_B are ground truth segmentation results for \mathbf{x}_A and \mathbf{x}_B respectively.

Zhang et al. report that 3D segmentation in their architecture improves not only over the baseline model trained only on real data but also over the standard approach of fine-tuning S_A^{VOL} and S_B^{VOL} separately on generated synthetic data. Ben-Cohen et al. [51] present a similar architecture for cross-modal synthetic data generation of PET scans from CT images, also with improved segmentation results for lesion detection. Similar image-to-image translation techniques have been applied to generating images from 2D MR brain images to CT and back [282, 424, 639], PET to CT [27], cardiac CT to MR [96], virtual

H&E staining, including transformation from unstained to stained lung histology images [43] and stain style transfer [538], multi-contrast MRI (from contrast to contrast) [138], 3D cross-modality MRI [677], different styles of prostate histopathology [488], different datasets of chest X-rays [98], and others.

Model-based domain adaptation (Section 6.2) has also been applied in the context of medical imaging. Often it has been used to do domain transfer between different types of real images, e.g. between different parts of the brain [53] or from *in vitro* to *in vivo* images [122], but synthetic-to-real DA has also been a major topic. As early as 2013, Heimann et al. [240] generated synthetic training data in the form of digitally reconstructed radiographs for ultrasound transducer localization. To close the domain gap between synthetic and real images, they used standard instance weighting and found significant improvements in the resulting detections. Kamnitsas et al. [302] use unsupervised DA for brain lesion segmentation in 3D, switching from one type of MR images to another in domain adaptation. They use a state of the art 3D multi-scale fully convolutional segmentation network [303] and a domain discriminator that makes intermediate feature representations of the segmentation networks indistinguishable between the domains.

In general, GAN-based architectures for medical imaging, either generating synthetic data, adapting real data from other domains, or , represent promising directions of further research and will, in our opinion, define state of the art in the field for years to come. However, at present the architectures used in different works differ a lot, and comparisons across different GAN-based architectures are usually lacking: each work compares their architecture only with the baselines. Further research and large-scale experimental studies are needed to determine which architectures work best for various domain adaptation problems related to medical imaging.

7 Privacy guarantees in synthetic data

7.1 Differential privacy in deep learning

In many domains, real data is not only valuable but also sensitive; it should be protected by law, commercial interest, and common decency. The unavailability of real data is exactly what makes synthetic data solutions attractive in these domains—but models for generating synthetic data have to train on real datasets anyway, how do we know we are not revealing it? A famous paper by Dinur and Nissim [150] showed that a few database queries (e.g., taking sums or averages of subsets) suffice to bring about strong violations of privacy. If a machine learning model has trained on a dataset with a few outliers, how do we know it doesn’t “memorize” these outliers directly? For a sufficiently expressive model, such memorization is quite possible, and note that the outliers are usually the most sensitive data points. For instance, Carlini et al. [86] show that state of the art language models do memorize specific sequences of symbols, and one can extract, e.g., a secret string of numbers from the original dataset with a reasonably high success rate.

The field of differential privacy, pioneered by Dwork et al. [160, 162, 164] (the work [162] received the Gödel Prize in 2017), was largely motivated by considerations such as above. In the main definition of the field, a randomized algorithm A is called (ϵ, δ) -differentially private if for any two databases D and D' that differ in only a single point and any subset of outputs S

$$p(A(D) \in S) \leq e^\epsilon p(A(D') \in S) + \delta,$$

or, equivalently, for every point s in the output range of A

$$\left| \frac{p(A(D) = s)}{p(A(D') = s)} \right| \leq e^\epsilon \quad \text{with probability } 1 - \delta.$$

The intuition here is that an adversary who receives only the outputs of A should have a hard time learning anything about any single point in D . Differential privacy has many desirable qualities such as composability (allowing for modular design of architectures), group privacy (graceful degradation when independency assumptions break), and robustness to auxiliary information that an adversary might have.

In this section, we review the applications of differential privacy and related concepts to synthetic data generation. The purpose is similar: the release of a synthetic dataset generated by some model trained on real data should not disclose information regarding the individual points in this real dataset. Our review is slanted towards deep learning; for a more complete picture of the field we refer to the surveys in [58, 160]. However, we do note the efforts devoted to generating differentially private synthetic datasets in classical machine learning. Lu et al. [380] develop a model for making sensitive databases private by fixing a set of queries to the database and perturbing the outputs to ensure differential privacy. Zhang et al. present the *PrivBayes* approach [691]: construct a Bayesian network that captures the correlations and dependencies between data attributes, inject noise into the marginals that constitute this network, and then sample from the perturbed network to produce the private synthetic dataset. In a similar effort, the *DataSynthesizer* model by Ping et al. [452] is able to

take a sensitive dataset as input and generate a synthetic dataset that has the same statistics and structure but at the same time provides differential privacy guarantees.

We also note some privacy-related applications of synthetic data that are not about differential privacy. For example, Ren et al. [491] present an adversarial architecture for video face anonymization; their model learns to modify the original real video to remove private information while at the same time still maximizing the performance of action recognition models (see also Section 2.3).

As for deep learning, we begin with a brief overview of how to make complex high-dimensional optimization, such as training deep neural networks, respect privacy constraints. The basic approach to achieving differential privacy is to add noise to the output of A ; many classical works on the subject focus on estimating and reducing the amount of noise necessary to ensure privacy under various assumptions [160–163, 361]. However, it is not immediately obvious how to apply this idea to a deep neural network. There are two major approaches to achieving differential privacy in deep learning.

Abadi et al. [1] suggest a method for controlling the influence of the training data during stochastic gradient descent called Differentially Private SGD (DP-SGD). Specifically, they clip the gradients on each SGD iteration to a predefined value of the L_2 -norm and add Gaussian noise to the resulting gradient value. By careful analysis of the privacy loss variable, i.e., $\log \frac{p(A(D)=s)}{p(A(D')=s)}$ above, Abadi et al. show that the resulting algorithm preserves differential privacy under reasonable choices of the clipping and random noise parameters. Moreover, this is a general approach that is agnostic to the network architecture and can be extended to various first-order optimization algorithms based on SGD.

A year later, Papernot et al. [436] (actually, mostly the same group of researchers from Google) presented the Private Aggregation of Teacher Ensembles (PATE) approach. In PATE, the final “student” model is trained from an ensemble of “teacher” models that have access to sensitive data, while the “student” model only has access to (noisy) aggregated results of “teacher” models, which allows to control the disclosure and preserve privacy. A big advantage of this approach is that the “teacher” models can be treated as black-box while still providing rigorous differential privacy guarantees based on the same moments accounting technique from [1]. Incidentally, the best results were obtained with adversarial training for the “student” in a semi-supervised fashion, where the entire dataset is available for the “student” but labels are only provided for a subset of it, preserving privacy.

7.2 Differential privacy guarantees for synthetic data generation

The general approaches we have discussed in the previous section have been modified and applied for producing synthetic data with generative models, mostly, of course, with generative adversarial networks. Although the methods are similar, we note an important conceptual difference that synthetic data brings in this case. *Model release* approaches in the previous section assumed access to and full control of model training. *Data release* approaches (here we use the terminology from [598]) that perform synthetic data generation have the following advantages:

- they can provide private data to third parties to construct better models and develop new techniques or use computational resources that might be unavailable to the holders of sensitive data;
- moreover, these third parties are able to pool synthetic data from different sources, while in the model release framework this would require a transfer of sensitive data;
- synthetic data can be either traded or freely made public, which is an important step towards reproducibility of research, especially in such fields as bioinformatics and healthcare, where reproducibility is an especially important problem and where, at the same time, sensitive data abounds.

In this section, we discuss existing constructions of GANs that provide rigorous privacy guarantees for the resulting generated data. Basically, in the ideal case a differentially private GAN has to generate an artificial dataset that would be sampled from the same distribution p_{data} but with differential privacy guarantees as discussed above. One general remark that is used in most of these works is that in a GAN architecture, it suffices to have privacy guarantees or additional privacy-preserving modifications (such as adding noise) only in the discriminator since gradient updates for the generator are functions of discriminator updates. Another important remark is that in cases when we generate differentially private synthetic data, a drop in quality for subsequent “student” models trained on synthetic data is expected in nearly all cases, not because of any deficiencies of synthetic data vs. real in general but because the nature of differential privacy requires adding random noise to the generative model training.

Xie et al. [657] present the differentially private GAN (DPGAN) model, which is basically the already classical Wasserstein GAN [26, 217] but with additional noise on the gradient of the Wasserstein distance, in a fashion following the DP-SGD approach (Section 7.1). They apply DPGAN to generate electronic health records, showing that classifiers trained on synthetic records have accuracy approaching that of classifiers trained on real data, while guaranteeing differential privacy. This was further developed by Zhang et al. [696] who used the Improved WGAN framework [217] and obtained excellent results on the synthetic data generated from various subsets of the LSUN dataset [696], which is already a full-scale image dataset, albeit low-resolution (64×64). Beaulieu-Jones et al. [45] apply the same idea to generating electronic health records, specifically training on the data of the Systolic Blood Pressure Trial (SPRINT) data analysis challenge [154, 583], which are in nature low-dimensional time series. They used the DP-SGD approach for the Auxiliary Classifier GAN (AC-GAN) architecture [428] and studied how the accuracy of various classifiers drops when passing to synthetic data. Triastcyn and Faltings [598] continue this line of work and show that differential privacy guarantees can be obtained by adding a special Gaussian noise layer to the discriminator network. They show good results for “student” models trained on synthetically generated data for MNIST, but already at the SVHN dataset the performance degrades more severely.

Bayesian methods are a natural fit for differential privacy since they deal with entire distributions of parameters and lend themselves easily to adding extra noise needed for DP guarantees. A Bayesian variant of the GAN framework,

which provides representations of full posterior distributions over the parameters, was provided by Saatchi and Wilson [512]. Arnold et al. [28] adapted the BayesGAN framework for differential privacy by injecting noise into the gradients during training, shown by Wang et al. [632] to lead to DP guarantees. They apply the resulting DP-BayesGAN framework to microdata, i.e., medium-dimensional samples of 40 explanatory variables of different nature and one dependent variable.

As for the PATE framework, it cannot be directly applied to GANs since noisy aggregation of a PATE ensemble is not a differentiable function that could serve as part of a GAN discriminator. Ács et al. [6] proposed to use a differentially private clustering method to split the data into k clusters, then train a separate generative models (the authors tried VAE) on their own clusters, and then create a mixture of the resulting models that will inherit differential privacy properties as well. A recent work by Jordon et al. [675] circumvents the non-differentiability problem by training a “student-discriminator” on already differentially private synthetic data produced by the generator. The learning procedure alternates between updating “teacher” classifiers for a fixed generator on real samples and updating the “student-discriminator” classifier and the generator for fixed “teachers”. PATE-GAN works well on low-dimensional datasets but begins to lose ground on high-dimensional datasets such as, e.g., the UCI Epileptic Seizure Recognition dataset (with 184 features).

However, these results are still underwhelming; it has proven very difficult to stabilize GAN training with the additional noise necessary for differential privacy guarantees, which has not allowed researchers to progress to, say, higher resolution images so far. In a later work, Triastcyn and Faltings [597] consider a different approach: they use the *empirical DP* framework [3, 95, 124, 527], an approach that empirically estimates the privacy of a posterior distribution, and the modification that ensures privacy is usually a sufficiently diffuse prior. In this framework, evaluating the privacy would reduce to training a GAN on the original dataset D , removing one sample from D to obtain D' , retraining the GAN and comparing the probabilities of all outcomes, and so on, repeating these experiments enough times to obtain empirical estimates for ϵ and δ . For realistic GANs, a large number of retrainings is impractical, so Triastcyn and Faltings modify this procedure to make it operate directly on the generated set \tilde{D} rather than the original dataset D . They study the tradeoff of privacy vs. accuracy of the “student” models trained on synthetic data and show that GANs can fall into the region of practical values for both privacy and accuracy. Their proposed modification of the architecture (a single randomizing layer close to the end of the discriminator) strengthens DP guarantees while preserving good generation quality for datasets up to *CelebA* [374]; in fact, it appears to serve as a regularizer and improve generation.

Frigerio et al. [182] extend the DPGAN framework to continuous, categorical, and time series data. They use the Wasserstein GAN loss function [217], extending the moment accountant to this case. To handle discrete variables, the generator produces an output for every possible value with a softmax layer on top, and its results are sent to the discriminator. Bindschadler [58] presents a *seedbased* modification of synthetic data generation: an algorithm that produces data records through a generative model conditioned on some seed real data record; this significantly improves quality but introduces correlations between real and synthetic data. To avoid correlations, Bindschadler introduces

privacy tests that reject unsuitable synthetic data points. The approach can be used in complex models based on encoder-decoder architectures by adding noise to a seed in the latent space; it has been evaluated across different domains from census data to celebrity face images, the latter through a VAE/GAN architecture [345].

Finally, we note that synthetic data produced with differential privacy guarantees is also starting to gain legal status; in a technical report [50], Bellovin et al. from the Stanford Law School discuss various definitions of privacy from the point of view of what kind of data can be released. They conclude: “...as we recommend, synthetic data may be combined with differential privacy to achieve a best-of-both-worlds scenario”, i.e., combining added utility of synthetic data produced by generative models with formal privacy guarantees.

7.3 Case study: synthetic data in economics, healthcare, and social sciences

Synthetic data is increasingly finding its way into economics, healthcare, and social sciences in a variety of applications. We discuss this set of models and applications here since often the main concern that drives researchers in these fields to synthetic data is not lack of data *per se* but rather privacy issues. A number of models that guarantee differential privacy have already been discussed above, so in this section we concentrate on other approaches and applications.

As far back as 1993, Rubin [506] discussed the dangers of releasing microdata (i.e., information about individual transactions) and the extremely complicated legal status of data releases, as the released data might be used to derive protected information even if it had been masked by standard techniques. To avoid these complications, Rubin proposed to use imputed synthetic data instead: given a dataset with confidential information, “forget” and impute confidential values for a sample from this dataset, using the same background variables but drawing confidential data from the predictions of some kind of imputation model. Repeating the process for several samples, we get a multiply-imputed population that can then be released. In the same year (actually, the same special issue of the *Journal of Official Statistics*), Little [366] suggested to also keep the non-confidential part of the information to improve imputation. By now, synthetic datasets produced by multiple imputation are a well-established field of statistics, with applications to finance and economics [147, 484], healthcare [17], social sciences [77], survey statistics [4, 13], and other domains. Since the main emphasis of the present survey is on synthetic data for deep learning, we do not go into details about multiple imputation and refer to the book [156] and the main recent sources in the field [155, 476, 483, 485].

In a very recent work, Heaton and Witte [239] propose another interesting take on synthetic data in finance. They begin with the well-known problem of overfitting during backtesting: since there is a very large number of financial products and relatively short time series available for them, one can always find a portfolio (subset of products) that works great during backtesting, but it does not necessarily reflect future performance. The authors suggest to use synthetic data not to train financial strategies (they regard this as infeasible) but rather to *evaluate* developed strategies, generating synthetic data with a different distribution of abnormalities and testing strategies for robustness in these altered circumstances. Interestingly, the motivation here is not to improve

or choose the best strategies but to obtain evidence of their robustness that could be used for regulatory purposes. As a specific application, the authors use existing fraud detection algorithms to find anomalies in the Kaggle Credit Card Fraud Detection Dataset [136] and generate synthetic data that balances the found abnormalities.

However, at present, we know of no direct applications where synthetically generated financial time series that would lead to improved results in financial forecasting, developing financial strategies, and the like. In general, financial time series are notorious for not being amenable to either prediction or accurate modeling, and even with current state of the art economic models we can hardly hope to generate useful synthetic financial time series any more than we can hope to generate meaningful text (see Section 4.3).

As for healthcare, this is again a field where the need for synthetic data was understood very early, and this need was mostly caused by privacy concerns: hospitals are required to protect the confidentiality of their patients. Ever since the first works in this direction, dating back to early 1990s, researchers mostly concentrated on generating synthetic electronic medical records (EMR) in order to preserve privacy [39]. In more recent work, MDClone [405] is a system that samples synthetic EMRs from the distributions learned on existing cohorts, without actually reusing original data points. Walonoski et al. [617] present the *Synthea* software suite designed to simulate the lifespans of synthetic patients and produce realistic synthetic EMRs. McLaghlan [403] discusses realism in synthetic EMR generation and methods for its validation, and in another work presents a state transition machine that incorporates domain knowledge and clinical practice guidelines to generate realistic synthetic EMRs [402].

Another related direction of research concentrates not on individual EMRs but on modeling entire populations of potential patients. Synthetic micro-populations produced by Smith et al. [551] are intended to match various sociodemographic conditions found in real cities and use them in imitational modeling to estimate the effect of interventions. Moniz et al. [417, 511] create synthetic EMRs made available on the CDC Public Health grid for imitational modeling. Buczak et al. [74] generate synthetic EMRs for an outbreak of a certain disease (together with background records). Kartoun [312] progressed from individual EMRs to entire virtual patient repositories, concentrating on preserving the correct general statistics while using simulated individual records. However, most of this work does not make use of modern formalizations of differential privacy or recent developments in generative models, and only very recently researchers have attempted to bring those into the healthcare micro-data domain as well.

A direct application of GANs for synthetic EMR generation was presented by Choi et al. [115]. Their *medGAN* model consists of a generator G^{MED} , discriminator D^{MED} , and an autoencoder with encoder Enc^{MED} and decoder Dec^{MED} . The autoencoder is trained to reconstruct real data $\mathbf{x}_T \sim \mathcal{X}_T$, while G^{MED} learns to generate latent representations $G^{\text{MED}}(\mathbf{z})$ from a random seed \mathbf{z} such that D^{MED} will not be able to differentiate between $\text{Dec}^{\text{MED}}(G^{\text{MED}}(\mathbf{z}))$ and a real sample $\mathbf{x}_T \sim \mathcal{X}_T$. The privacy in the *medGAN* model is established empirically, and the main justification for privacy is the fact that *medGAN* uses real data only for the discriminator and never trains the generator on any real samples. We note, however, that in terms of generation *medGAN* is not perfect: for example, Patel et al. [440] present the *CorrGAN* model for correlated discrete data

generation (with no regard for privacy) and show improvements over *medGAN* with a relatively straightforward architecture.

The DP-SGD framework has also been applied to GANs in the context of medical data. We have discussed Beaulieu-Jones et al. [45] above. Another important application for synthetic data across many domains, including but not limited to finance, would be to generate synthetic time series. This, however, has proven to be a more difficult problem, and solutions are only starting to appear. In particular, Hyland et al. [170] present the Recurrent GAN (RGAN) and Recurrent Conditional GAN architectures designed to generate realistic real-valued multi-dimensional time series. They applied the architecture to generating medical time series (vitals measured for ICU patients) and reported successful generation and ability to train classifiers on synthetic data, although there was a significant drop in quality when testing on real data. Hyland et al. also discuss the possibility to use a differentially private training procedure, applying the DP-SGD framework to the discriminator and thus achieving differential privacy for the RGAN training. The authors report that after this procedure, synthetic-to-real test results deteriorate significantly but remain reasonable in classification tasks on ICU patient vitals.

Finally, we note another emerging field of research related to generating synthetic EMRs for the sake of privacy: generating clinical notes and free-text fields in EMRs with neural language models (see also Section 4.3). Latest advances in deep learning for natural language processing have led to breakthroughs in large-scale language modeling [144, 255, 473], and this has been applied to smaller datasets of clinical notes as well. Lee [350] uses an encoder-decoder architecture to generate chief complaint texts for EMRs. Guan et al. [216] propose a GAN architecture called *mtGAN* (medical text GAN) for the generation of synthetic EMR text. It is based on the SeqGAN architecture [678] and is trained with the REINFORCE algorithm; the primary difference is a condition added by Guan et al. to be able to generate EMRs for a specific disease or other features. Melamud and Shivade [408] compare LSTM-based language models for generating synthetic clinical notes, suggesting a new privacy measure and showing promising results. Further advances in this direction may be related to the recently developed differentially private language models [404].

8 Promising directions for future work

8.1 Procedural generation of synthetic data

The first direction that we highlight as important for further study in the field of synthetic data is *procedural generation*. Take, for instance, synthetic indoor scenes that we discussed in Section 3.2. Note that in the main synthetic dataset for indoor scenes, SUNCG, the 3D scenes and their layouts were created manually. While we have seen that this approach has an advantage of several orders of magnitude in terms of labeled images over real datasets, it still cannot scale up to millions of 3D scenes. The only way to achieve that would be to learn a model that can generate the *contents* of a 3D scene (in this case, an indoor environment with furniture and supported objects), varying it stochastically according to some random input seed. This is a much harder problem than it might seem: e.g., a bedroom is much more than just a predefined set of objects placed at random positions.

There is a large related field of procedural content generation for video games [243, 540, 590], but we highlight a recent work by Qi et al. [464] as representative for state of the art and more directly related to synthetic data. They propose a human-centric approach to modeling indoor scene layout, learning a stochastic scene grammar based on an attributed spatial AND-OR graph [715] that relates scene components, objects, and corresponding human activities. A scene configuration is represented by a parse graph whose probability is modeled with potential functions corresponding to functional grouping relations between furniture, relations between a supported object and supporting furniture, and human-centric relations between the furniture based on the map of sampled human trajectories in the scene. After learning the weights of potential functions corresponding to the relations between objects, MCMC sampling can be used to generate new indoor environments. Qi et al. train their model on the very same SUNCG dataset and show that the resulting layouts are hard to distinguish (by an automated state of the art classifier trained on layout segmentation maps) from the original SUNCG data.

In Section 3.1, we have already discussed *ProcSy* by Khan et al. [317]. In addition to randomizing weather and lighting conditions, another interesting part of their work is the procedural generation of cities and outdoor scenes. They base this procedural generation on the method of Parish and Müller [438], which is in turn based on the notion of Lindenmayer systems (L-systems) [463] and embodied in the *CityEngine* tool [656] (see Section 3.1). They use a part of real *OpenStreetMaps* data for the road network and buildings, but we hope that future work based on the same ideas can offer fully procedural modeling of cities and road networks.

We believe that procedural generation can lead to an even larger scale of synthetic data adoption in the future, covering cases when simply placing the objects at random is not enough. The nature of the model used for procedural generation may differ depending on the specific field of application, but in general for procedural generation one needs to train probabilistic generative models that allow for both learning from real or manually prepared synthetic data and then generating new samples.

8.2 From domain randomization to the generation feedback loop

The work [97] makes one of the first steps in a very interesting direction. They are also working on domain transfer, transferring continuous control policies for robotic arms from synthetic to real domain. But importantly, they attempt to close the feedback loop between synthetic data generation and domain transfer via domain randomization. Previous works on domain randomization (see above) manually tuned the distribution of simulation parameters $p_\phi(\xi)$ such that a policy trained on $D_{\xi \sim p_\phi}$ would perform well. In [97], the parameters of $p_\phi(\xi)$ are learned automatically via a feedback loop from the results of real observations.

A similar approach on the level of data augmentation was presented in [481]. As we discussed in Section 1, data augmentation differs from synthetic data in that it modifies real data rather than creates new; the modifications are usually done with predefined transformation functions (TFs) that do not change the target labels. This assumption is somewhat unrealistic: e.g., if we augment by shifting or cropping the image for image classification, we might crop out exactly the object that determines the class label. The work [481] relaxes this assumption, treating TFs as black boxes that might move the data point out of all necessary classes, into the “null” class, but cannot mix up different classes of objects. The authors train a generative sequence model with an adversarial objective that learns a sequence of TFs that would not move data points into the “null” class by training a null class discriminator D_ϕ^\emptyset and a generator G_θ for sequences of TFs $h_L \circ \dots \circ h_1$:

$$\min_{\theta} \max_{\phi} \mathbb{E}_{\tau \sim G_\theta} \mathbb{E}_{x \sim \mathcal{U}} \left[\log(1 - D_\phi^\emptyset(h_{\tau_L} \circ \dots \circ h_{\tau_1}(x))) \right] + \mathbb{E}_{x' \sim \mathcal{U}} \left[\log(D_\phi^\emptyset(x')) \right],$$

where \mathcal{U} is some distribution of (possibly unlabeled) data. Since TFs are not necessarily differentiable or deterministic, learning G_θ is defined in the syntax of reinforcement learning. Pashevich et al. [439] note that the space of augmentation functions is very large (for 8 different transformations they estimate to have $\approx 3.6 \cdot 10^{14}$ augmentation functions), and propose to use Monte Carlo Tree Search (MCTS) [125, 323] to find the best augmentations by automatic exploration. They apply this idea to augmenting synthetic images for sim-to-real policy transfer for robotic manipulation and report improved results in real world tasks such as cube stacking or cup placing. *Google Brain* researchers Cubuk et al. [128] continue this work, presenting a framework for learning augmentation strategies from data. Their approach is modeled after recent advances in automated architecture search [33, 49, 718, 719]: they use reinforcement learning to find the best augmentation policy composed of a number of parameterized operations. As a result, they significantly improve state of the art results on such classical datasets as CIFAR-10, CIFAR-100, and ImageNet.

Zakharov et al. [682] look at a similar idea from the point of view of domain randomization (see also Section 5.1). Their framework consists of a recognition network that does the basic task (say, object detection and pose estimation) and a deception network that transforms the synthetic input with an encoder-decoder architecture. Training is done in an adversarial way, alternating between two phases:

- fixing the weights of the deception network, perform updates of the recog-

nition network as usual, serving synthetic images transformed by the deception network as inputs;

- fixing the weights of the recognition network, perform updates of the deception network with the same objective but with reversed gradients, updating the deception network so as to make the inputs hard for the recognition network.

The deception network is organized and constrained in such a way that its transformations do not change the ground truth labels or change them in predictable ways. This adversarial framework exhibits performance comparable to state of the art domain adaptation frameworks and shows superior generalization capabilities (better avoiding overfitting).

There are two recent works that represent important steps towards closing this feedback loop. First, the *Meta-Sim* framework [309], which we discussed in Section 5.2 in the context of high-level procedural scene generation, also makes inroads in this direction: the distribution parameters for synthetic data generation are tuned not only to bring the synthetic data distribution closer to the real one but also to improve the performance on downstream tasks such as object detection. The difference here is that instead of low-level parameters of image augmentation functions *Meta-Sim* adapts high-level parameters such as the synthetic scene structure captured as a scene graph.

Second, the *Visual Adversarial Domain Randomization and Augmentation* (VADRA) model by Khirodkar et al. [318] makes the next step in developing domain randomization ideas: instead of simply randomizing synthetic data or making it similar to real, let’s learn a policy π_ω that generates rendering parameters in such a way that the downstream model learns best. They use the REINFORCE algorithm to obtain stochastic gradients for the objective $J(\omega)$ which consists of the downstream model performance (for supervised data) and the errors of a domain classifier (for unsupervised data; this is the “adversarial” part of VADRA). As a result, VADRA works much better for the syn-to-real transfer on problems such as object detection and segmentation than regular domain randomization.

Similar ideas have been recently explored by Mehta et al. [406], who present *Active Domain Randomization*, again learning a policy for generating better simulated instance, but this time in the context of generating Markov decision processes for reinforcement learning, Ruiz et al. [507], who also learn a policy π_ω that outputs the parameters for a simulator, learning to generate data to maximize validation accuracy, with reinforcement learning techniques, and Louppe et al. [379], who provide an inference algorithm based on variational approximations for fitting the parameters of a domain-specific non-differentiable simulator.

We believe that this meta-approach to automatically learning the best way to generate synthetic data, both high-level and low-level, is an important new direction that might work well for other applications too. In our opinion, this idea might be further improved by methods such as the SPIRAL framework by Ganin et al. [189] or neural painter models [267, 421, 711] that train adversarial architectures to generate images in the form of sequences of brushstrokes or higher-level image synthesis programs with instructions such as “place object X at location Y ”; these or other kinds of high-level descriptions for images

might be more convenient for the generation feedback loop. We expect further developments in this direction in the nearest future.

8.3 Improving domain adaptation with domain knowledge

To showcase this direction, we consider one more work on gaze estimation (see Section 6.1.1) that presents a successful application of a hybrid approach to image refinement. Namely, Wang et al. [622] propose a very different approach to generating synthetic data for gaze estimation: a hierarchical generative model (HGM) that is able to operate both top-down, generating new synthetic images of eyes, and bottom-up, performing Bayesian inference to estimate the gaze in a given new image.

The general structure of their approach is shown on Fig. 23. Specifically, Wang et al. design a probabilistic hierarchical generative shape model (HGSM) based on 27 eye-related landmarks that together represent the shape of a human eye. The model connects personal parameters that define variation between humans, visual axis parameters that define eye gaze, and eye shape parameters. The structure of HGSM is based on anatomical studies, and its parameters are learned from the UnityEyes dataset [641]. During generation, HGSM generates eye shape parameters (positions of the 27 landmarks in the eyeball’s spherical coordinate system) based on the given gaze direction.

The second part of the pipeline generates the actual images with a conditional BiGAN (bidirectional GAN) architecture [151]. Bidirectional GAN is an architecture that learns to transform data in both directions, from latent representations to the objects and back, while regular GAN’s learn to generate only the objects from latent representations. The conditional BiGAN (c-BiGAN) modification developed in [622] does the same with a condition, which in this case are the eye shape parameters produced by HGSM. As a result, the model by Wang et al. can work in both directions:

- generate eye images by sampling the gaze from a prior distribution, sampling 2D eye shape parameters from HGSM, and then using the generator G of c-BiGAN to generate a refined image;
- infer gaze parameters from an eye image by first estimating the eye shape through the encoder E of c-BiGAN and then performing Bayesian inference in the HGSM to find the posterior distribution of gaze parameters.

Wang et al. report performance improvements of the model itself applied to gaze estimation over [545] for sufficiently large training sets, and also show that the synthetic data generated by the model improves the results of standard gaze estimators (*LeNet*, as used in [694]).

In general, we mark the approach of [622] to combining probabilistic generative models that incorporate domain knowledge and GAN-based architectures as a very interesting direction for further studies. We believe this approach can be suitable for applications other than gaze estimation.

8.4 Additional modalities for domain adaptation architectures

Another natural idea that has not been used too widely yet is to use the additional data modalities such as depth maps or 3D volumetric data, which are

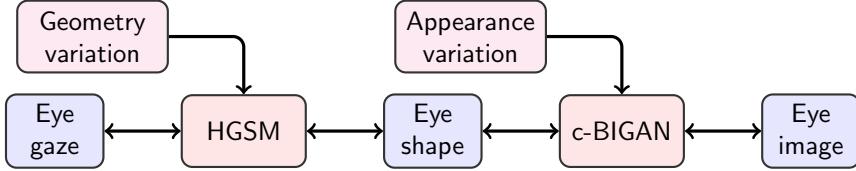


Figure 23: General structure of the hierarchical generative model for eye image synthesis and gaze estimation [622]. Left to right: top-down image synthesis pipeline; right to left: bottom-up eye gaze estimation pipeline.

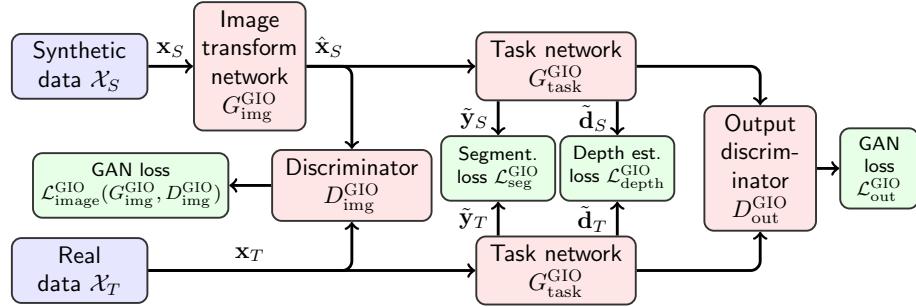


Figure 24: General structure of the GIO-Ada model with input- and output-level domain adaptation [109].

available for free in synthetic datasets but usually unavailable in real ones, to improve

Pioneering work in this direction has been recently done by Chen et al. [109]. They note that depth estimation and segmentation are related tasks that are increasingly learned together with multitask learning architectures [315, 446, 662] and then propose to use this idea to improve synthetic-to-real domain adaptation. Their model, called *Geometrically Guided Input-Output Adaptation* (GIO-Ada), is based on the PatchGAN architecture [275] intended for image translation.

We have illustrated the GIO-Ada model on Figure 24. Similar to the refiners considered in Section 6.1.2, they train a GAN generator to refine the synthetic image, but the refiner takes as input not just a synthetic image \mathbf{x}_S but an input triple $(\mathbf{x}_S, \mathbf{y}, \mathbf{d})$, where \mathbf{x}_S is a synthetic image, \mathbf{y} is its segmentation map, and \mathbf{d} is its depth map. Moreover, they also incorporate *output-level adaptation*, where a separate generator predicts segmentation and depth maps, and a discriminator tries to distinguish whether these maps came from a synthetic transformed image or from a real one. In this way, the model can use the depth information to obtain additional cues to further improve segmentation on real images, and the output-level adaptation brings segmentation results on synthetic and real domains closer together.

Specifically, the GIO-Ada model optimizes, in an adversarial way, the following objective:

$$\min_{G^{\text{GIO}}, G^{\text{GIO}}_{\text{task}}} \max_{D^{\text{GIO}}, D^{\text{GIO}}_{\text{out}}} [\mathcal{L}_{\text{seg}}^{\text{GIO}} + \lambda_1 \mathcal{L}_{\text{depth}}^{\text{GIO}} + \lambda_2 \mathcal{L}_{\text{image}}^{\text{GIO}} + \lambda_3 \mathcal{L}_{\text{out}}^{\text{GIO}}], \text{ where:}$$

- $G_{\text{img}}^{\text{GIO}}$ is an image transformation network that performs input-level adaptation, i.e., produces a transformed image $\hat{\mathbf{x}} = G_{\text{img}}^{\text{GIO}}(\mathbf{x}, \mathbf{y}, \mathbf{d})$, and $G_{\text{task}}^{\text{GIO}}$ is the output-level adaptation network that predicts the segmentation and depth maps $(\tilde{\mathbf{y}}, \tilde{\mathbf{d}}) = G_{\text{task}}^{\text{GIO}}(\mathbf{x})$;
 - $D_{\text{img}}^{\text{GIO}}$ is the image discriminator that tries to distinguish between $\hat{\mathbf{x}}_S$ and \mathbf{x}_T , and $D_{\text{out}}^{\text{GIO}}$ is the output discriminator that distinguishes between $(\tilde{\mathbf{y}}_S, \tilde{\mathbf{d}}_S) = G_{\text{task}}^{\text{GIO}}(\hat{\mathbf{x}}_S)$ and $(\tilde{\mathbf{y}}_T, \tilde{\mathbf{d}}_T) = G_{\text{task}}^{\text{GIO}}(\mathbf{x}_T)$;
 - $\mathcal{L}_{\text{seg}}^{\text{GIO}}$ is the segmentation cross-entropy loss $\mathcal{L}_{\text{seg}}^{\text{GIO}} = \mathbb{E}_{\mathbf{x}_S \sim p_{\text{syn}}} [\text{CE}(\mathbf{y}_S, \tilde{\mathbf{y}}_S)]$;
 - $\mathcal{L}_{\text{depth}}^{\text{GIO}}$ is the depth estimation L_1 -loss $\mathcal{L}_{\text{depth}}^{\text{GIO}} = \mathbb{E}_{\mathbf{x}_S \sim p_{\text{syn}}} [\|\mathbf{d}_S - \tilde{\mathbf{d}}_S\|_1]$;
 - $\mathcal{L}_{\text{image}}^{\text{GIO}}$ is the GAN loss for $D_{\text{img}}^{\text{GIO}}$:
- $$\mathcal{L}_{\text{image}}^{\text{GIO}} = \mathbb{E}_{\mathbf{x}_T \sim p_{\text{real}}} [\log D_{\text{img}}^{\text{GIO}}(\mathbf{x}_T)] + \mathbb{E}_{\mathbf{x}_S \sim p_{\text{syn}}} [\log(1 - D_{\text{img}}^{\text{GIO}}(\hat{\mathbf{x}}_S))];$$
- $\mathcal{L}_{\text{out}}^{\text{GIO}}$ is the GAN loss for $D_{\text{out}}^{\text{GIO}}$:
- $$\mathcal{L}_{\text{out}}^{\text{GIO}} = \mathbb{E}_{\mathbf{x}_T \sim p_{\text{real}}} [\log D_{\text{out}}^{\text{GIO}}(\tilde{\mathbf{y}}_T, \tilde{\mathbf{d}}_T)] + \mathbb{E}_{\mathbf{x}_S \sim p_{\text{syn}}} [\log(1 - D_{\text{out}}^{\text{GIO}}(\tilde{\mathbf{y}}_S, \tilde{\mathbf{d}}_S))].$$

Chen et al. report promising results on standard synthetic-to-real adaptations: Virtual KITTI to KITTI and SYNTHIA to Cityscapes (see Section 3.1). This, however, looks to us as merely a first step in the very interesting direction of using additional data modalities easily provided by synthetic data generators to further improve domain adaptation.

9 Conclusion

In this work, we have attempted a survey of one of the most promising general techniques on the rise in modern deep learning, especially computer vision: synthetic data. This source of virtually limitless perfectly labeled data has been explored in many problems, but we believe that many more potential use cases still remain.

In direct applications of synthetic data, we have discussed many different domains and use cases, from basic computer vision tasks such as stereo disparity estimation or semantic segmentation to full-scale simulated environments for autonomous driving, unmanned aerial vehicles, and robotics. In the domain adaptation part, we have surveyed a wide variety of generative models for synthetic-to-real refinement and for feature-level domain adaptation. As another important field of synthetic data applications, we have considered data generation with differential privacy guarantees. We have also reviewed the works dedicated to improving synthetic data generation and outlined potential promising directions for further research.

In general, throughout this survey we have seen synthetic data work well across a wide variety of tasks and domains. We believe that synthetic data is essential for further development of deep learning: many applications require labeling which is expensive or impossible to do by hand, other applications have a wide underlying data distribution that real datasets do not or cannot fully cover, yet other applications may benefit from additional modalities unavailable in real datasets, and so on. Moreover, we believe that synthetic data applications will be extended in the future. For example, while this survey does not yet have a section devoted to sound and speech processing, works that use synthetic data in this domain are already beginning to appear [353, 510]. As synthetic data becomes more realistic (where necessary) and encompasses more use cases and modalities, we expect it to play an increasingly important role in deep learning.

References

- [1] Martin Abadi, Andy Chu, Ian Goodfellow, H. Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, CCS ’16*, pages 308–318, New York, NY, USA, 2016. ACM.
- [2] Ali Abbasi, Sinan Kalkan, and Yusuf Sahillioglu. Deep 3d semantic scene extrapolation. *The Visual Computer*, 35:271–279, 2018.
- [3] J. Abowd, M. Schneider, and L. Vilhuber. Differential privacy applications to bayesian and linear mixed model estimation. *Journal of Privacy and Confidentiality*, 5(1):185–205, 2013.
- [4] John Abowd, Martha Stinson, and Gary Benedetto. Final report to the social security administration on the sipp/ssa/irs public use file project. Technical report, Longitudinal Employer–Household Dynamics Program, U.S. Bureau of the Census, Washington, DC, <https://hdl.handle.net/1813/43929>, 01 2006.
- [5] 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, Sep 2018.
- [6] Gergely Ács, Luca Melis, Claude Castelluccia, and Emiliano De Cristofaro. Differentially private mixture of generative neural networks. *CoRR*, abs/1709.04514, 2017.
- [7] Widyawardana Adiprawita, Adang Suwandi Ahmad, and Jaka Semibiring. Hardware in the loop simulator in UAV rapid development life cycle. *CoRR*, abs/0804.3874, 2008.
- [8] Aishwarya Agrawal, Jiasen Lu, Stanislaw Antol, Margaret Mitchell, C Lawrence Zitnick, Devi Parikh, and Dhruv Batra. VQA: Visual Question Answering. *International Journal of Computer Vision*, 123(1):4–31, 2017.
- [9] C.E. Aguero, N. Koenig, I. Chen, H. Boyer, S. Peters, J. Hsu, B. Gerkey, S. Paepcke, J.L. Rivero, J. Manzo, E. Krotkov, and G. Pratt. Inside the virtual robotics challenge: Simulating real-time robotic disaster response. *Automation Science and Engineering, IEEE Transactions on*, 12(2):494–506, April 2015.
- [10] O. Akbani, A. Gokrani, M. Quresh, F. M. Khan, S. I. Behlim, and T. Q. Syed. Character recognition in natural scene images. In *2015 International Conference on Information and Communication Technologies (ICICT)*, pages 1–6, Dec 2015.
- [11] Abdulla Al-Kaff, David Martín, Fernando García, Arturo de la Escalera, and José María Armingol. Survey of computer vision algorithms and applications for unmanned aerial vehicles. *Expert Systems with Applications*, 92:447 – 463, 2018.

- [12] Nada Jasim Al-Musawi and Saad Talib Hasson. Improving video streams summarization using synthetic noisy video data. *International Journal of Advanced Computer Science and Applications*, 6(12), 2015.
- [13] Andreas Alfons, Stefan Kraft, Matthias Templ, and Peter Filzmoser. Simulation of close-to-reality population data for household surveys with application to eu-silc. *Statistical Methods & Applications*, 20(3):383–407, Aug 2011.
- [14] Eloi Alonso, Bastien Moysset, and Ronaldo O. Messina. Adversarial generation of handwritten text images conditioned on sequences. *CoRR*, abs/1903.00277, 2019.
- [15] Samuel Alvernaz and Julian Togelius. Autoencoder-augmented neuroevolution for visual doom playing. *2017 IEEE Conference on Computational Intelligence and Games (CIG)*, pages 1–8, 2017.
- [16] Brandon Amos, Bartosz Ludwiczuk, and Mahadev Satyanarayanan. Openface: A general-purpose face recognition library with mobile applications. 2016.
- [17] Di An, Roderick J. A. Little, and James W. McNally. A multiple imputation approach to disclosure limitation for high-age individuals in longitudinal studies. *Statistics in Medicine*, 29(17):1769–1778, 2010.
- [18] P. Andersen, M. Goodwin, and O. Granmo. Deep rts: A game environment for deep reinforcement learning in real-time strategy games. In *2018 IEEE Conference on Computational Intelligence and Games (CIG)*, pages 1–8, Aug 2018.
- [19] Cyrus Anderson, Xiaoxiao Du, Ram Vasudevan, and Matthew Johnson-Roberson. Stochastic sampling simulation for pedestrian trajectory prediction. *ArXiv*, abs/1903.01860, 2019.
- [20] Hiromasa Ando, Ihor Lubashevsky, Arkady Zgonnikov, and Yoshiaki Saito. Statistical Properties of Car Following: Theory and Driving Simulator Experiments. *arXiv e-prints*, page arXiv:1511.04640, Nov 2015.
- [21] Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Dan Klein. Deep compositional question answering with neural module networks. *CoRR*, abs/1511.02799, 2015.
- [22] Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Dan Klein. Learning to compose neural networks for question answering. *CoRR*, abs/1601.01705, 2016.
- [23] M. Andriluka, L. Pishchulin, P. Gehler, and B. Schiele. 2d human pose estimation: New benchmark and state of the art analysis. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3686–3693, June 2014.
- [24] Mykhaylo Andriluka, Leonid Pishchulin, Peter Gehler, and Bernt Schiele. 2d human pose estimation: New benchmark and state of the art analysis. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014.

- [25] Cesario Vincenzo Angelino, Vincenzo Rosario Baraniello, and Luca Cicala. High altitude UAV navigation using imu, GPS and camera. In *Proceedings of the 16th International Conference on Information Fusion, FUSION 2013, Istanbul, Turkey, July 9-12, 2013*, pages 647–654, 2013.
- [26] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 214–223, International Convention Centre, Sydney, Australia, 06–11 Aug 2017. PMLR.
- [27] Karim Armanious, Chenming Yang, Marc Fischer, Thomas Küstner, Konstantin Nikolaou, Sergios Gatidis, and Bin Yang. Medgan: Medical image translation using gans. *CoRR*, abs/1806.06397, 2018.
- [28] Christian Arnold, Marcel Neunhoeffer, and Sebastian Sternberg. Releasing differentially private synthetic micro-data with bayesian gans, 2019.
- [29] Mathieu Aubry, Daniel Maturana, Alexei Efros, Bryan Russell, and Josef Sivic. Seeing 3d chairs: exemplar part-based 2d-3d alignment using a large dataset of cad models. In *CVPR*, 2014.
- [30] Mathieu Aubry and Bryan C. Russell. Understanding deep features with computer-generated imagery. In *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, ICCV ’15, pages 2875–2883, Washington, DC, USA, 2015. IEEE Computer Society.
- [31] Oleksandr Bailo, DongShik Ham, and Young Min Shin. Red blood cell image generation for data augmentation using conditional generative adversarial networks. *CoRR*, abs/1901.06219, 2019.
- [32] Slawomir Bak, Peter Carr, and Jean-François Lalonde. Domain adaptation through synthesis for unsupervised person re-identification. In *ECCV*, 2018.
- [33] Bowen Baker, Otkrist Gupta, Nikhil Naik, and Ramesh Raskar. Designing neural network architectures using reinforcement learning. *CoRR*, abs/1611.02167, 2016.
- [34] Simon Baker, Daniel Scharstein, J. P. Lewis, Stefan Roth, Michael J. Black, and Richard Szeliski. A database and evaluation methodology for optical flow. *International Journal of Computer Vision*, 92(1):1–31, Mar 2011.
- [35] Mayank Bansal, Alex Krizhevsky, and Abhijit S. Ogale. Chauffeurnet: Learning to drive by imitating the best and synthesizing the worst. *ArXiv*, abs/1812.03079, 2018.
- [36] Yu Bao, Hao Zhou, Shujian Huang, Lei Li, Lili Mou, Olga Vechtomova, Xin-yu Dai, and Jiajun Chen. Generating sentences from disentangled syntactic and semantic spaces. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6008–6019, Florence, Italy, July 2019. Association for Computational Linguistics.

- [37] Elaheh Barati, Xuewen Chen, and Zichun Zhong. Attention-based deep reinforcement learning for multi-view environments. *CoRR*, abs/1905.03985, 2019.
- [38] J. L. Barron, D. J. Fleet, and S. S. Beauchemin. Performance of optical flow techniques. *International Journal of Computer Vision*, 12(1):43–77, Feb 1994.
- [39] Randolph C. Barrows and Paul D. Clayton. Privacy, confidentiality, and electronic medical records. *Journal of the American Medical Informatics Association : JAMIA*, 3 2:139–48, 1996.
- [40] Christoph Baur, Shadi Albarqouni, and Nassir Navab. Melanogans: High resolution skin lesion synthesis with gans. *CoRR*, abs/1804.04338, 2018.
- [41] Ertugrul Bayraktar, Cihat Bora Yigit, and Pinar Boyraz. A hybrid image dataset toward bridging the gap between real and simulation environments for robotics. *Machine Vision and Applications*, 30:23–40, 2018.
- [42] Ertugrul Bayraktar, Cihat Bora Yigit, and Pinar Boyraz. A hybrid image dataset toward bridging the gap between real and simulation environments for robotics. *Machine Vision and Applications*, 30(1):23–40, Feb 2019.
- [43] N. Bayramoglu, M. Kaakinen, L. Eklund, and J. Heikkilä. Towards virtual h e staining of hyperspectral lung histology images using conditional generative adversarial networks. In *2017 IEEE International Conference on Computer Vision Workshops (ICCVW)*, pages 64–71, Oct 2017.
- [44] Charles Beattie, Joel Z. Leibo, Denis Teplyashin, Tom Ward, Marcus Wainwright, Heinrich Küttler, Andrew Lefrancq, Simon Green, Víctor Valdés, Amir Sadik, Julian Schrittwieser, Keith Anderson, Sarah York, Max Cant, Adam Cain, Adrian Bolton, Stephen Gaffney, Helen King, Demis Hassabis, Shane Legg, and Stig Petersen. Deepmind lab. *CoRR*, abs/1612.03801, 2016.
- [45] Brett K. Beaulieu-Jones, Zhiwei Steven Wu, Chris Williams, Ran Lee, Sanjeev P. Bhavnani, James Brian Byrd, and Casey S. Greene. Privacy-preserving generative deep neural networks support clinical data sharing. *bioRxiv*, 2018.
- [46] Andrew Beers, James M. Brown, Ken Chang, J. Peter Campbell, Susan Ostmo, Michael F. Chiang, and Jayashree Kalpathy-Cramer. High-resolution medical image synthesis using progressively grown generative adversarial networks. *CoRR*, abs/1805.03144, 2018.
- [47] Marc G. Bellemare, Yavar Naddaf, Joel Veness, and Michael Bowling. The arcade learning environment: An evaluation platform for general agents. *J. Artif. Int. Res.*, 47(1):253–279, May 2013.
- [48] Fabio Luigi Bellifemine, Giovanni Caire, and Dominic Greenwood. *Developing Multi-Agent Systems with JADE (Wiley Series in Agent Technology)*. John Wiley & Sons, Inc., USA, 2007.

- [49] Irwan Bello, Barret Zoph, Vijay Vasudevan, and Quoc V. Le. Neural optimizer search with reinforcement learning. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 459–468, International Convention Centre, Sydney, Australia, 06–11 Aug 2017. PMLR.
- [50] Steven M. Bellovin, Preetam K. Dutta, and Nathan Reitinger. Privacy and synthetic datasets. Technical report, SSRN, October 2018.
- [51] Avi Ben-Cohen, Eyal Klang, Stephen P. Raskin, Shelly Soffer, Simona Ben-Haim, Eli Konen, Michal Marianne Amitai, and Hayit Greenspan. Cross-modality synthesis from CT to PET using FCN and GAN networks for improved automated lesion detection. *CoRR*, abs/1802.07846, 2018.
- [52] Sagie Benaim and Lior Wolf. One-sided unsupervised domain mapping. In *Proceedings of the 31st International Conference on Neural Information Processing Systems, NIPS’17*, pages 752–762, USA, 2017. Curran Associates Inc.
- [53] Róger Bermúdez-Chacón, Carlos Becker, Mathieu Salzmann, and Pascal Fua. Scalable unsupervised domain adaptation for electron microscopy. In Sébastien Ourselin, Leo Joskowicz, Mert R. Sabuncu, Gozde Unal, and William Wells, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2016*, pages 326–334, Cham, 2016. Springer International Publishing.
- [54] David Berthelot, Tom Schumm, and Luke Metz. BEGAN: boundary equilibrium generative adversarial networks. *CoRR*, abs/1703.10717, 2017.
- [55] Alex Bewley, Jessica Rigley, Yuxuan Liu, Jeffrey Hawke, Richard Shen, Vinh-Dieu Lam, and Alex Kendall. Learning to drive from simulation without real world labels. *ArXiv*, abs/1812.03823, 2018.
- [56] Shehroze Bhatti, Alban Desmaison, Ondrej Miksik, Nantas Nardelli, N. Siddharth, and Philip H. S. Torr. Playing doom with slam-augmented deep reinforcement learning. *CoRR*, abs/1612.00380, 2016.
- [57] Adam Bielski and Paolo Favaro. Emergence of object segmentation in perturbed generative models. *ArXiv*, abs/1905.12663, 2019.
- [58] Vincent Bindschadler. *Privacy-Preserving Seedbased Data Synthesis*. PhD thesis, University of Illinois at Urbana-Champaign, 2018.
- [59] N. Bisagno and N. Conci. Virtual camera modeling for multi-view simulation of surveillance scenes. In *2018 26th European Signal Processing Conference (EUSIPCO)*, pages 2170–2174, Sep. 2018.
- [60] Adriano Bittar, Neusa Maria Franco de Oliveira, and Helosman Valente de Figueiredo. Hardware-in-the-loop simulation with x-plane of attitude control of a suav exploring atmospheric conditions. *Journal of Intelligent & Robotic Systems*, 73(1):271–287, Jan 2014.

- [61] V. Blanz, K. Scherbaum, and H. Seidel. Fitting a morphable model to 3d scans of faces. In *2007 IEEE 11th International Conference on Computer Vision*, pages 1–8, Oct 2007.
- [62] V. Blanz and T. Vetter. Face recognition based on fitting a 3d morphable model. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(9):1063–1074, Sep. 2003.
- [63] Volker Blanz and Thomas Vetter. A morphable model for the synthesis of 3d faces. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH ’99*, pages 187–194, New York, NY, USA, 1999. ACM Press/Addison-Wesley Publishing Co.
- [64] E. Bochinski, V. Eiselein, and T. Sikora. Training a convolutional neural network for multi-class object detection using solely virtual world data. In *2016 13th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 278–285, Aug 2016.
- [65] Verónica Bolón-Canedo, Noelia Sánchez-Maróño, and Amparo Alonso-Betanzos. A review of feature selection methods on synthetic data. *Knowl. Inf. Syst.*, 34(3):483–519, March 2013.
- [66] Elizabeth Bondi, Debadeepta Dey, Ashish Kapoor, Jim Piavis, Shital Shah, Fei Fang, Bistra N. Dilkina, Robert Hannaford, Arvind Iyer, Lucas Joppa, and Milind Tambe. Airsim-w: A simulation environment for wildlife conservation with uavs. In *COMPASS*, 2018.
- [67] Margherita Bonetto, Pavel Korshunov, Giovanni Ramponi, and Touradj Ebrahimi. Privacy in mini-drone based video surveillance. *2015 11th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG)*, 04:1–6, 2015.
- [68] Francisco Bonin-Font, Alberto Ortiz, and Gabriel Oliver. Visual navigation for mobile robots: A survey. *Journal of Intelligent and Robotic Systems*, 53(3):263, May 2008.
- [69] João Borrego, Atabak Dehban, Rui Figueiredo, Plinio Moreno, Alexandre Bernardino, and José Santos-Victor. Applying domain randomization to synthetic data for object category detection. *CoRR*, abs/1807.09834, 2018.
- [70] Konstantinos Bousmalis, Alex Irpan, Paul Wohlhart, Yunfei Bai, Matthew Kelcey, Mrinal Kalakrishnan, Laura Downs, Julian Ibarz, Peter Pastor, Kurt Konolige, Sergey Levine, and Vincent Vanhoucke. Using simulation and domain adaptation to improve efficiency of deep robotic grasping. *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4243–4250, 2018.
- [71] Konstantinos Bousmalis, Nathan Silberman, David Dohan, Dumitru Erhan, and Dilip Krishnan. Unsupervised pixel-level domain adaptation with generative adversarial networks. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 95–104, 2017.

- [72] Konstantinos Bousmalis, George Trigeorgis, Nathan Silberman, Dilip Krishnan, and Dumitru Erhan. Domain separation networks. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS’16, pages 343–351, USA, 2016. Curran Associates Inc.
- [73] Tom Bruls, Horia Porav, Lars Kunze, and Paul Newman. Generating all the roads to rome: Road layout randomization for improved road marking segmentation. *ArXiv*, abs/1907.04569, 2019.
- [74] Anna L. Buczak, Steven Babin, and Linda Moniz. Data-driven approach for creating synthetic electronic medical records. *BMC Medical Informatics and Decision Making*, 10(1):59, Oct 2010.
- [75] Elvijs Buls, Roberts Kadikis, Ričards Cacurs, and Jānis Ārents. Generation of synthetic training data for object detection in piles. In *International Conference on Machine Vision*, 2019.
- [76] Rudy Bunel, Matthew J. Hausknecht, Jacob Devlin, Rishabh Singh, and Pushmeet Kohli. Leveraging grammar and reinforcement learning for neural program synthesis. *CoRR*, abs/1805.04276, 2018.
- [77] Jan Pablo Burgard, Jan-Philipp Kolb, Hariolf Merkle, and Ralf Münnich. Synthetic data for open and reproducible methodological research in social sciences and official statistics. *AStA Wirtschafts- und Sozialstatistisches Archiv*, 11(3):233–244, Dec 2017.
- [78] Alexander V. Buslaev, Alex Parinov, Eugene Khvedchenya, Vladimir I. Iglovikov, and Alexandr A. Kalinin. Albumentations: fast and flexible image augmentations. *CoRR*, abs/1809.06839, 2018.
- [79] Daniel J. Butler, Jonas Wulff, Garrett B. Stanley, and Michael J. Black. A naturalistic open source movie for optical flow evaluation. In Andrew Fitzgibbon, Svetlana Lazebnik, Pietro Perona, Yoichi Sato, and Cordelia Schmid, editors, *Computer Vision – ECCV 2012*, pages 611–625, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.
- [80] J. Byrne and C. J. Taylor. Expansion segmentation for visual collision detection and estimation. In *2009 IEEE International Conference on Robotics and Automation*, pages 875–882, May 2009.
- [81] Zhaowei Cai, Quanfu Fan, Rogério Schmidt Feris, and Nuno Vasconcelos. A unified multi-scale deep convolutional neural network for fast object detection. *CoRR*, abs/1607.07155, 2016.
- [82] Francesco Calimeri, Aldo Marzullo, Claudio Stamile, and Giorgio Terracina. Biomedical data augmentation using generative adversarial neural networks. In Alessandra Lintas, Stefano Rovetta, Paul F.M.J. Verschure, and Alessandro E.P. Villa, editors, *Artificial Neural Networks and Machine Learning – ICANN 2017*, pages 626–634, Cham, 2017. Springer International Publishing.
- [83] B. Calli, A. Singh, A. Walsman, S. Srinivasa, P. Abbeel, and A. M. Dollar. The ycb object and model set: Towards common benchmarks for manipulation research. In *2015 International Conference on Advanced Robotics (ICAR)*, pages 510–517, July 2015.

- [84] Q. Cao, L. Shen, W. Xie, O. M. Parkhi, and A. Zisserman. Vggface2: A dataset for recognising faces across pose and age. In *International Conference on Automatic Face and Gesture Recognition*, 2018.
- [85] Emilio Capo. State of the art on: Deep learning for video games ai development. http://www.honours-programme.deib.polimi.it/2018-I/Deliverable1/CSE_CAPO_SOTA.pdf, 2018.
- [86] Nicholas Carlini, Chang Liu, Jernej Kos, Úlfar Erlingsson, and Dawn Song. The secret sharer: Measuring unintended neural network memorization & extracting secrets. *CoRR*, abs/1802.08232, 2018.
- [87] F. M. Carlucci, P. Russo, and B. Caputo. A deep representation for depth images from synthetic data. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1362–1369, May 2017.
- [88] Jeremy D. Castagno, Yu Yao, and Ella M. Atkins. Realtime rooftop landing site identification and selection in urban city simulation. *ArXiv*, abs/1903.03829, 2019.
- [89] A. B. Chan, Zhang-Sheng John Liang, and N. Vasconcelos. Privacy preserving crowd monitoring: Counting people without people models or tracking. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–7, June 2008.
- [90] Caroline Chan, Shiry Ginosar, Tinghui Zhou, and Alexei A. Efros. Everybody dance now. *ArXiv*, abs/1808.07371, 2018.
- [91] Khyathi Raghavi Chandu, Mary Arpita Pyreddy, Matthieu Felix, and Narendra Nath Joshi. Textually enriched neural module networks for visual question answering. *CoRR*, abs/1809.08697, 2018.
- [92] Angel X. Chang, Angela Dai, Thomas A. Funkhouser, Maciej Halber, Matthias Nießner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3d: Learning from RGB-D data in indoor environments. *CoRR*, abs/1709.06158, 2017.
- [93] Angel X. Chang, Thomas A. Funkhouser, Leonidas J. Guibas, Pat Hanrahan, Qi-Xing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li Yi, and Fisher Yu. Shapenet: An information-rich 3d model repository. *CoRR*, abs/1512.03012, 2015.
- [94] Qianwen Chao, Huikun Bi, Weizi Li, Tianlu Mao, Zhaoqi Wang, Ming C. Lin, and Zhigang Deng. A survey on visual traffic simulation: Models, evaluations, and applications in autonomous driving. *Computer Graphics Forum*, 0(0).
- [95] Anne-Sophie Charest and Yiwei Hou. On the meaning and limits of empirical differential privacy. *Journal of Privacy and Confidentiality*, 7(3):185–205, 2017.
- [96] Agisilaos Chartsias, Thomas Joyce, Rohan Dharmakumar, and Sotirios A. Tsaftaris. Adversarial image synthesis for unpaired multi-modal cardiac data. In *SASHIMIMICCAI*, 2017.

- [97] Yevgen Chebotar, Ankur Handa, Viktor Makoviychuk, Miles Macklin, Jan Issac, Nathan D. Ratliff, and Dieter Fox. Closing the sim-to-real loop: Adapting simulation randomization with real world experience. *CoRR*, abs/1810.05687, 2018.
- [98] Cheng Chen, Qi Dou, Hao Chen, and Pheng-Ann Heng. Semantic-aware generative adversarial nets for unsupervised domain adaptation in chest x-ray segmentation. *CoRR*, abs/1806.00600, 2018.
- [99] Hongming Chen, Ola Engkvist, Yinhai Wang, Marcus Olivecrona, and Thomas Blaschke. The rise of deep learning in drug discovery. *Drug Discovery Today*, 23(6):1241 – 1250, 2018.
- [100] Ke Chen, Chen Change Loy, Shaogang Gong, and Tony Xiang. Feature mining for localised crowd counting. In *BMVC*, 2012.
- [101] L. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(4):834–848, April 2018.
- [102] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, *Computer Vision – ECCV 2018*, pages 833–851, Cham, 2018. Springer International Publishing.
- [103] Lyujie Chen, Wufan Wang, and Jihong Zhu. Learning transferable uav for forest visual perception. *ArXiv*, abs/1806.03626, 2018.
- [104] Xiaobai Chen, Aleksey Golovinskiy, and Thomas Funkhouser. A benchmark for 3d mesh segmentation. *ACM Trans. Graph.*, 28(3):73:1–73:12, July 2009.
- [105] Xinyuan Chen, Chang Xu, Xiaokang Yang, and Dacheng Tao. Attention-gan for object transfiguration in wild images. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, *Computer Vision – ECCV 2018*, pages 167–184, Cham, 2018. Springer International Publishing.
- [106] Xinyun Chen, Chang Liu, and Dawn Song. Execution-guided neural program synthesis. In *International Conference on Learning Representations*, 2019.
- [107] Yi-Hsin Chen, Wei-Yu Chen, Yu-Ting Chen, Bo-Cheng Tsai, Yu-Chiang Frank Wang, and Min Sun. No more discrimination: Cross city adaptation of road scene segmenters. *CoRR*, abs/1704.08509, 2017.
- [108] Yueh-Tung Chen, Martin Garbade, and Juergen Gall. 3d semantic scene completion from a single depth image using adversarial training. *ArXiv*, abs/1905.06231, 2019.

- [109] Yuhua Chen, Wen Li, Xiaoran Chen, and Luc Van Gool. Learning semantic segmentation from synthetic data: A geometrically guided input-output adaptation approach. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.
- [110] Ernest Cheung, Tsan Kwong Wong, Aniket Bera, Xiaogang Wang, and Dinesh Manocha. Lcrowdv: Generating labeled videos for simulation-based crowd behavior learning. In Gang Hua and Hervé Jégou, editors, *Computer Vision – ECCV 2016 Workshops*, pages 709–727, Cham, 2016. Springer International Publishing.
- [111] Travers Ching, Daniel S. Himmelstein, Brett K. Beaulieu-Jones, Alexandr A. Kalinin, Brian T. Do, Gregory P. Way, Enrico Ferrero, Paul-Michael Agapow, Michael Zietz, Michael M. Hoffman, Wei Xie, Gail L. Rosen, Benjamin J. Lengerich, Johnny Israeli, Jack Lanchantin, Stephen Woloszynek, Anne E. Carpenter, Avanti Shrikumar, Jinbo Xu, Evan M. Cofer, Christopher A. Lavender, Srinivas C. Turaga, Amr M. Alexandari, Zhiyong Lu, David J. Harris, Dave DeCaprio, Yanjun Qi, Anshul Kundaje, Yifan Peng, Laura K. Wiley, Marwin H. S. Segler, Simina M. Boca, S. Joshua Swamidass, Austin Huang, Anthony Gitter, and Casey S. Greene. Opportunities and obstacles for deep learning in biology and medicine. *Journal of The Royal Society Interface*, 15(141):20170387, 2018.
- [112] Maciek Chociej, Peter Welinder, and Lilian Weng. ORRB - openai remote rendering backend. *CoRR*, abs/1906.11633, 2019.
- [113] C. Choi and H. I. Christensen. Rgb-d object tracking: A particle filter approach on gpu. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1084–1091, Nov 2013.
- [114] Dooseop Choi, Taeg-Hyun An, Kyoungwhan Ahn, and Jeongdan Choi. Driving experience transfer method for end-to-end control of self-driving cars. *CoRR*, abs/1809.01822, 2018.
- [115] Edward Choi, Siddharth Biswal, Bradley Malin, Jon Duke, Walter F. Stewart, and Jimeng Sun. Generating multi-label discrete patient records using generative adversarial networks. In Finale Doshi-Velez, Jim Fackler, David Kale, Rajesh Ranganath, Byron Wallace, and Jenna Wiens, editors, *Proceedings of the 2nd Machine Learning for Healthcare Conference*, volume 68 of *Proceedings of Machine Learning Research*, pages 286–305, Boston, Massachusetts, 18–19 Aug 2017. PMLR.
- [116] Sungjoon Choi, Qian-Yi Zhou, Stephen Miller, and Vladlen Koltun. A large dataset of object scans. *ArXiv*, abs/1602.02481, 2016.
- [117] François Chollet. Xception: Deep learning with depthwise separable convolutions. *CoRR*, abs/1610.02357, 2016.
- [118] S. Chopra, R. Hadsell, and Y. LeCun. Learning a similarity metric discriminatively, with application to face verification. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 1, pages 539–546 vol. 1, June 2005.

- [119] Lovish Chum, Anbumani Subramanian, Vineeth N. Balasubramanian, and C. V. Jawahar. Beyond supervised learning: A computer vision perspective. *Journal of the Indian Institute of Science*, 99(2):177–199, Jun 2019.
- [120] M. J. M. Chuquicusma, S. Hussein, J. Burt, and U. Bagci. How to fool radiologists with generative adversarial networks? a visual turing test for lung cancer diagnosis. In *2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018)*, pages 240–244, April 2018.
- [121] David Cohn, Les Atlas, and Richard Ladner. Improving generalization with active learning. *Machine Learning*, 15(2):201–221, May 1994.
- [122] Sailesh Conjeti, Amin Katouzian, Abhijit Guha Roy, Loïc Peter, Debdoot Sheet, Stéphane Carlier, Andrew Laine, and Nassir Navab. Supervised domain adaptation of decision forests: Transfer of models trained in vitro for in vivo intravascular ultrasound tissue characterization. *Medical Image Analysis*, 32:1 – 17, 2016.
- [123] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.
- [124] G. Cormode, C. M. Procopiuc, E. Shen, D. Srivastava, and T. Yu. Empirical privacy and empirical utility of anonymized data. In *2013 IEEE 29th International Conference on Data Engineering Workshops (ICDEW)*, pages 77–82, April 2013.
- [125] Rémi Coulom. Efficient selectivity and backup operators in monte-carlo tree search. In H. Jaap van den Herik, Paolo Ciancarini, and H. H. L. M. (Jeroen) Donkers, editors, *Computers and Games*, pages 72–83, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg.
- [126] Jonathan Courbon, Youcef Mezouar, Nicolas Guénard, and Philippe Martinet. Vision-based navigation of unmanned aerial vehicles. *Control Engineering Practice*, 18(7):789 – 799, 2010. Special Issue on Aerial Robotics.
- [127] Nicolas County, Pierre Allain, Clement Creusot, and Thomas Corpetti. Using the agoraset dataset: Assessing for the quality of crowd video analysis methods. *Pattern Recognition Letters*, 44:161 – 170, 2014. Pattern Recognition and Crowd Analysis.
- [128] Ekin Dogus Cubuk, Barret Zoph, Dandelion Mané, Vijay Vasudevan, and Quoc V. Le. Autoaugment: Learning augmentation policies from data. *CoRR*, abs/1805.09501, 2018.
- [129] M. Cutler and J. P. How. Efficient reinforcement learning for robots using informative simulated priors. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2605–2612, May 2015.

- [130] Mark Cutler, Thomas J. Walsh, and Jonathan P. How. Reinforcement learning with multi-fidelity simulators. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 3888–3895, Hong Kong, June 2014.
- [131] Mark Cutler, Thomas J. Walsh, and Jonathan P. How. Real-world reinforcement learning via multifidelity simulators. *IEEE Transactions on Robotics*, 31(3):655–671, June 2015.
- [132] C. R. d. Souza, A. Gaidon, Y. Cabon, and A. M. López. Procedural generation of videos to train deep action recognition networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2594–2604, July 2017.
- [133] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 2017.
- [134] Jifeng Dai, Yi Li, Kaiming He, and Jian Sun. R-FCN: object detection via region-based fully convolutional networks. *CoRR*, abs/1605.06409, 2016.
- [135] Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc Le, and Ruslan Salakhutdinov. Transformer-XL: Attentive language models beyond a fixed-length context. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2978–2988, Florence, Italy, July 2019. Association for Computational Linguistics.
- [136] Andrea Dal Pozzolo, Olivier Caelen, Yann-Aël Le Borgne, Serge Waterschoot, and Gianluca Bontempi. Learned lessons in credit card fraud detection from a practitioner perspective. *Expert Systems with Applications*, 41:4915–4928, 08 2014.
- [137] Q. Dang, J. Yin, B. Wang, and W. Zheng. Deep learning based 2d human pose estimation: A survey. *Tsinghua Science and Technology*, 24(6):663–676, Dec 2019.
- [138] Salman Ul Hassan Dar, Mahmut Yurt, Levent Karacan, Aykut Erdem, Erkut Erdem, and Tolga Çukur. Image synthesis in multi-contrast MRI with conditional generative adversarial networks. *CoRR*, abs/1802.01221, 2018.
- [139] A. Das, S. Datta, G. Gkioxari, S. Lee, D. Parikh, and D. Batra. Embodied question answering. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 2135–213509, June 2018.
- [140] Mitchell Dawson, Andrew Zisserman, and Christoffer Nellaker. Mining faces from biomedical literature using deep learning. In *Proceedings of the 8th ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics, ACM-BCB ’17*, pages 562–567, New York, NY, USA, 2017. ACM.

- [141] Jiankang Deng, Jia Guo, Yuxiang Zhou, Jinke Yu, Irene Kotsia, and Stefanos Zafeiriou. Retinaface: Single-stage dense face localisation in the wild. *CoRR*, abs/1905.00641, 2019.
- [142] G. N. Desouza and A. C. Kak. Vision for mobile robot navigation: a survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(2):237–267, Feb 2002.
- [143] Jacob Devlin, Rudy R Bunel, Rishabh Singh, Matthew Hausknecht, and Pushmeet Kohli. Neural program meta-induction. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 2080–2088. Curran Associates, Inc., 2017.
- [144] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.
- [145] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186, 2019.
- [146] Jacob Devlin, Jonathan Uesato, Surya Bhupatiraju, Rishabh Singh, Abdel-rahman Mohamed, and Pushmeet Kohli. Robustfill: Neural program learning under noisy I/O. *CoRR*, abs/1703.07469, 2017.
- [147] Giuseppe DiCesare. *Imputation, Estimation and Missing Data in Finance*. PhD thesis, University of Waterloo, 2006.
- [148] Deepak Dilipkumar and Barnabás Póczos. Generative adversarial image refinement for handwriting recognition. 2017.
- [149] Changxing Ding and Dacheng Tao. A comprehensive survey on pose-invariant face recognition. *ACM Trans. Intell. Syst. Technol.*, 7(3):37:1–37:42, February 2016.
- [150] Irit Dinur and Kobbi Nissim. Revealing information while preserving privacy. In *Proceedings of the Twenty-second ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, PODS ’03*, pages 202–210, New York, NY, USA, 2003. ACM.
- [151] Jeff Donahue, Philipp Krähenbühl, and Trevor Darrell. Adversarial feature learning. *CoRR*, abs/1605.09782, 2016.
- [152] A. Dosovitskiy, P. Fischer, E. Ilg, P. Häusser, C. Hazirbas, V. Golkov, P. v. d. Smagt, D. Cremers, and T. Brox. Flownet: Learning optical flow with convolutional networks. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 2758–2766, Dec 2015.
- [153] 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*, pages 1–16, 2017.

- [154] Jeffrey M. Drazen, Stephen Morrissey, Edward W. Campion, and John A. Jarcho. A sprint to the finish. *New England Journal of Medicine*, 373(22):2174–2175, 2015. PMID: 26551058.
- [155] Jörg Drechsler. *Generating Multiply Imputed Synthetic Datasets: Theory and Implementation*. PhD thesis, Otto-Friedrich-Universität Bamberg, 2010.
- [156] Jörg Drechsler. *Synthetic Datasets for Statistical Disclosure Control*, volume 201 of *Lecture Notes in Statistics*. Springer, 2011.
- [157] T. Driemeyer. *Rendering with Mental Ray*. Springer, New York, 2001.
- [158] Nikita Dvornik, Julien Mairal, and Cordelia Schmid. Modeling visual context is key to augmenting object detection datasets. *CoRR*, abs/1807.07428, 2018.
- [159] Debidatta Dwibedi, Ishan Misra, and Martial Hebert. Cut, paste and learn: Surprisingly easy synthesis for instance detection. *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 1310–1319, 2017.
- [160] Cynthia Dwork. Differential privacy: A survey of results. In Manindra Agrawal, Dingzhu Du, Zhenhua Duan, and Angsheng Li, editors, *Theory and Applications of Models of Computation*, pages 1–19, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg.
- [161] Cynthia Dwork, Krishnaram Kenthapadi, Frank McSherry, Ilya Mironov, and Moni Naor. Our data, ourselves: Privacy via distributed noise generation. In Serge Vaudenay, editor, *Advances in Cryptology - EUROCRYPT 2006*, pages 486–503, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.
- [162] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Proceedings of the Third Conference on Theory of Cryptography*, TCC’06, pages 265–284, Berlin, Heidelberg, 2006. Springer-Verlag.
- [163] Cynthia Dwork and Kobbi Nissim. Privacy-preserving datamining on vertically partitioned databases. In Matt Franklin, editor, *Advances in Cryptology – CRYPTO 2004*, pages 528–544, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg.
- [164] Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. *Found. Trends Theor. Comput. Sci.*, 9(3–4):211–407, 2014.
- [165] V.A. Efimova and A.A. Filchenkov. Generative models for placing text on images. In *IFMO Young Researchers Congress*, 2019.
- [166] Christian Eggert, Anton Winschel, and Rainer Lienhart. On the benefit of synthetic data for company logo detection. In *Proceedings of the 23rd ACM International Conference on Multimedia*, MM ’15, pages 1283–1286, New York, NY, USA, 2015. ACM.

- [167] Marwen H Eid, C N De Cecco, John William Nance, Damiano Caruso, M H Albrecht, Adam J. Spandorfer, Domenico De Santis, Akos Varga-Szemes, and Uwe Joseph Schoepf. Cinematic rendering in ct: A novel, lifelike 3d visualization technique. *AJR. American journal of roentgenology*, 209(2):370–379, 2017.
- [168] Hadi Keivan Ekbatani., Oriol Pujol., and Santi Segui. Synthetic data generation for deep learning in counting pedestrians. In *Proceedings of the 6th International Conference on Pattern Recognition Applications and Methods - Volume 1: ICPRAM*, pages 318–323. INSTICC, SciTePress, 2017.
- [169] Teófilo Emídio de Campos, Bodla Rakesh Babu, and Manik Varma. Character recognition in natural images. volume 2, pages 273–280, 01 2009.
- [170] Cristóbal Esteban, Stephanie L. Hyland, and Gunnar Rätsch. Real-valued (Medical) Time Series Generation with Recurrent Conditional GANs. *arXiv e-prints*, page arXiv:1706.02633, Jun 2017.
- [171] Face recognition prize challenge 2017, 2017.
- [172] Marzieh Fadaee, Arianna Bisazza, and Christof Monz. Data augmentation for low-resource neural machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 567–573, Vancouver, Canada, July 2017. Association for Computational Linguistics.
- [173] Jin Fang, Feilong Yan, Tongtong Zhao, Feihu Zhang, Dingfu Zhou, Ruigang Yang, Yongbing Ma, and Liang Wang. Simulating lidar point cloud for autonomous driving using real-world scenes and traffic flows. *ArXiv*, abs/1811.07112, 2018.
- [174] William Fedus, Ian Goodfellow, and Andrew M. Dai. MaskGAN: Better text generation via filling in the _____. In *International Conference on Learning Representations*, 2018.
- [175] C. Fernández, P. Baiget, F.X. Roca, and J. González. Augmenting video surveillance footage with virtual agents for incremental event evaluation. *Pattern Recognition Letters*, 32(6):878 – 889, 2011.
- [176] M. Firman. Rgbd datasets: Past, present and future. In *2016 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 661–673, June 2016.
- [177] Thor I. Fossen, Kristin Y. Pettersen, and Henk Nijmeijer. *Sensing and Control for Autonomous Vehicles: Applications to Land, Water and Air Vehicles*, volume 474 of *Lecture Notes in Control and Information Sciences*. Springer, 2017.
- [178] W. T. Freeman and E. C. Pasztor. Learning low-level vision. In *Proceedings of the Seventh IEEE International Conference on Computer Vision*, volume 2, pages 1182–1189 vol.2, Sep. 1999.

- [179] Karlis Freivalds and Renars Liepins. Improving the neural gpu architecture for algorithm learning. *CoRR*, abs/1702.08727, 2017.
- [180] Maayan Frid-Adar. Gan-based data augmentation for improved liver lesion classification. 2018.
- [181] Maayan Frid-Adar, Idit Diamant, Eyal Klang, Michal Amitai, Jacob Goldberger, and Hayit Greenspan. Gan-based synthetic medical image augmentation for increased cnn performance in liver lesion classification. *Neurocomputing*, 321:321 – 331, 2018.
- [182] Lorenzo Frigerio, Anderson Santana de Oliveira, Laurent Gomez, and Patrick Duverger. Differentially private generative adversarial networks for time series, continuous, and discrete open data. In Gurpreet Dhillon, Fredrik Karlsson, Karin Hedström, and André Zúquete, editors, *ICT Systems Security and Privacy Protection*, pages 151–164, Cham, 2019. Springer International Publishing.
- [183] Fadri Furrer, Michael Burri, Markus Achtelik, and Roland Siegwart. *RotorS—A Modular Gazebo MAV Simulator Framework*, pages 595–625. Springer International Publishing, Cham, 2016.
- [184] A Gaidon, Q Wang, Y Cabon, and E Vig. Virtual worlds as proxy for multi-object tracking analysis. In *CVPR*, 2016.
- [185] Adrien Gaidon, Antonio Lopez, and Florent Perronnin. The reasonable effectiveness of synthetic visual data. *International Journal of Computer Vision*, 126(9):899–901, Sep 2018.
- [186] Ysbrand Galama and Thomas Mensink. Itergans: Iterative gans to learn and control 3d object transformation. *ArXiv*, abs/1804.05651, 2018.
- [187] Ruslan Galinsky, Anton Alekseyev, and Sergey I. Nikolenko. Improving neural network models for natural language processing in russian with synonyms. In *Proc. 5th conference on Artificial Intelligence and Natural Language*, pages 45–51, 2016.
- [188] Stéphane Galland, Nicolas Gaud, Jonathan Demange, and Abderrafiâ Koukam. Environment model for multiagent-based simulation of 3d urban systems. In *Proc. 7th European Workshop on Multiagent Systems (EUMAS09)*, 2009.
- [189] Yaroslav Ganin, Tejas Kulkarni, Igor Babuschkin, S. M. Ali Eslami, and Oriol Vinyals. Synthesizing programs for images using reinforced adversarial learning. *CoRR*, abs/1804.01118, 2018.
- [190] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *Proceedings of the 32Nd International Conference on International Conference on Machine Learning - Volume 37*, ICML’15, pages 1180–1189. JMLR.org, 2015.
- [191] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *J. Mach. Learn. Res.*, 17(1):2096–2030, January 2016.

- [192] Xiaofeng Gao, Ran Gong, Tianmin Shu, Xu Xie, Shu Wang, and Song-Chun Zhu. Vrkitchen: an interactive 3d virtual environment for task-oriented learning. *CoRR*, abs/1903.05757, 2019.
- [193] Richard Garcia and Laura Barnes. Multi-uav simulator utilizing x-plane. *Journal of Intelligent and Robotic Systems*, 57(1):393, Oct 2009.
- [194] Nicolas Gaud, Stéphane Galland, Vincent Hilaire, and Abderrafiâ Koukam. An organisational platform for holonic and multiagent systems. In Koen V. Hindriks, Alexander Pokahr, and Sebastian Sardina, editors, *Programming Multi-Agent Systems*, pages 104–119, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg.
- [195] A Gaulton, L Bellis, J Chambers, M Davies, A Hersey, Y Light, S McGlinchey, Ruth Akhtar, F Atkinson, A.P. Bento, Bissan Al-Lazikani, D Michalovich, and John Overington. Chemb: A large-scale bioactivity database for chemical biology and drug discovery. *Nucleic Acids Res.*, 40:D1100–D1107, 10 2011.
- [196] Erik Gawehn, Jan Hiss, and Gisbert Schneider. Deep learning in drug discovery. *Molecular Informatics*, 35, 12 2015.
- [197] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *International Journal of Robotics Research (IJRR)*, 2013.
- [198] Georgios Georgakis, Arsalan Mousavian, Alexander C. Berg, and Jana Kosecka. Synthesizing training data for object detection in indoor scenes. *ArXiv*, abs/1702.07836, 2017.
- [199] T. Gerig, A. Morel-Forster, C. Blumer, B. Egger, M. Luthi, S. Schoenborn, and T. Vetter. Morphable face models - an open framework. In *2018 13th IEEE International Conference on Automatic Face Gesture Recognition (FG 2018)*, pages 75–82, May 2018.
- [200] F. A. Gers and E. Schmidhuber. Lstm recurrent networks learn simple context-free and context-sensitive languages. *IEEE Transactions on Neural Networks*, 12(6):1333–1340, Nov 2001.
- [201] Mario Valerio Giuffrida, Hanno Scharr, and Sotirios A. Tsaftaris. ARI-GAN: synthetic arabidopsis plants using generative adversarial network. *CoRR*, abs/1709.00938, 2017.
- [202] Yonatan Glassner, Liran Gispan, Ariel Ayash, and Tal Furman Shohet. Closing the gap towards end-to-end autonomous vehicle system. *CoRR*, abs/1901.00114, 2019.
- [203] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Domain adaptation for large-scale sentiment classification: A deep learning approach. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, ICML’11, pages 513–520, USA, 2011. Omnipress.
- [204] Clément Godard, Oisin Mac Aodha, and Gabriel J. Brostow. Unsupervised monocular depth estimation with left-right consistency. In *CVPR*, 2017.

- [205] Rafael Gómez-Bombarelli, David K. Duvenaud, José Miguel Hernández-Lobato, Jorge Aguilera-Iparraguirre, Timothy D. Hirzel, Ryan P. Adams, and Alán Aspuru-Guzik. Automatic chemical design using a data-driven continuous representation of molecules. *CoRR*, abs/1610.02415, 2016.
- [206] Abel Gonzalez-Garcia, Joost van de Weijer, and Yoshua Bengio. Image-to-image translation for cross-domain disentanglement. In *Proceedings of the 32Nd International Conference on Neural Information Processing Systems*, NIPS’18, pages 1294–1305, USA, 2018. Curran Associates Inc.
- [207] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 2672–2680. Curran Associates, Inc., 2014.
- [208] Yash Goyal, Tejas Khot, Aishwarya Agrawal, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. *International Journal of Computer Vision*, 127(4):398–414, Apr 2019.
- [209] Matthieu Grard, Romain Brégier, Florian Sella, Emmanuel Dellandréa, and Liming Chen. Object segmentation in depth maps with one user click and a synthetically trained fully convolutional network. In Fanny Ficuciello, Fabio Ruggiero, and Alberto Finzi, editors, *Human Friendly Robotics*, pages 207–221, Cham, 2019. Springer International Publishing.
- [210] Alex Graves, Greg Wayne, and Ivo Danihelka. Neural turing machines. *CoRR*, abs/1410.5401, 2014.
- [211] Arthur Gretton, Karsten M. Borgwardt, Malte J. Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *J. Mach. Learn. Res.*, 13(1):723–773, March 2012.
- [212] David Griffiths and Jan Boehm. SynthCity: A large scale synthetic point cloud. *arXiv e-prints*, page arXiv:1907.04758, Jul 2019.
- [213] E. Grosicki and H. E. Abed. Icdar 2009 handwriting recognition competition. In *2009 10th International Conference on Document Analysis and Recognition*, pages 1398–1402, July 2009.
- [214] Michael Gschwandtner, Roland Kwitt, Andreas Uhl, and Wolfgang Pree. Blensor: Blender sensor simulation toolbox. In George Bebis, Richard Boyle, Bahram Parvin, Darko Koracin, Song Wang, Kim Kyungnam, Bedrich Benes, Kenneth Moreland, Christoph Borst, Stephen DiVerdi, Chiang Yi-Jen, and Jiang Ming, editors, *Advances in Visual Computing*, pages 199–208, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.
- [215] Xiaodong Gu, Kyunghyun Cho, Jung-Woo Ha, and Sunghun Kim. DialogWAE: Multimodal response generation with conditional wasserstein auto-encoder. In *International Conference on Learning Representations*, 2019.

- [216] J. Guan, R. Li, S. Yu, and X. Zhang. Generation of synthetic electronic medical record text. In *2018 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 374–380, Dec 2018.
- [217] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5767–5777. Curran Associates, Inc., 2017.
- [218] Sumit Gulwani. Automating string processing in spreadsheets using input-output examples. *SIGPLAN Not.*, 46(1):317–330, January 2011.
- [219] Jiaxian Guo, Sidi Lu, Han Cai, Weinan Zhang, Yong Yu, and Jun Wang. Long text generation via adversarial training with leaked information. *CoRR*, abs/1709.08624, 2017.
- [220] Yandong Guo, Lei Zhang, Yuxiao Hu, Xiaodong He, and Jianfeng Gao. Ms-celeb-1m: A dataset and benchmark for large-scale face recognition. *CoRR*, abs/1607.08221, 2016.
- [221] Agrim Gupta, Justin Johnson, Li Fei-Fei, Silvio Savarese, and Alexandre Alahi. Social GAN: socially acceptable trajectories with generative adversarial networks. *CoRR*, abs/1803.10892, 2018.
- [222] Ankush Gupta, Andrea Vedaldi, and Andrew Zisserman. Synthetic data for text localisation in natural images. *CoRR*, abs/1604.06646, 2016.
- [223] Om K. Gupta and Ray A. Jarvis. Using a virtual world to design a simulation platform for vision and robotic systems. In George Bebis, Richard Boyle, Bahram Parvin, Darko Koracin, Yoshinori Kuno, Junxian Wang, Jun-Xuan Wang, Junxian Wang, Renato Pajarola, Peter Lindstrom, André Hinkenjann, Miguel L. Encarnaçāo, Cláudio T. Silva, and Daniel Coming, editors, *Advances in Visual Computing*, pages 233–242, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg.
- [224] S. Gupta, P. Arbeláez, R. Girshick, and J. Malik. Aligning 3d models to rgb-d images of cluttered scenes. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4731–4740, June 2015.
- [225] Saurabh Gupta, Pablo Andrés Arbeláez, Ross B. Girshick, and Jitendra Malik. Inferring 3d object pose in RGB-D images. *CoRR*, abs/1502.04652, 2015.
- [226] Md. Akmal Haidar and Mehdi Rezagholizadeh. Textkd-gan: Text generation using knowledge distillation and generative adversarial networks. *CoRR*, abs/1905.01976, 2019.
- [227] C. Han, H. Hayashi, L. Rundo, R. Araki, W. Shimoda, S. Muramatsu, Y. Furukawa, G. Mauri, and H. Nakayama. Gan-based synthetic brain mr image generation. In *2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018)*, pages 734–738, April 2018.

- [228] Changhee Han, Yoshiro Kitamura, Akira Kudo, Akimichi Ichinose, Leonardo Rundo, Yujiro Furukawa, Kazuki Umemoto, Hideki Nakayama, and Yuanzhong Li. Synthesizing Diverse Lung Nodules Wherever Massively: 3D Multi-Conditional GAN-based CT Image Augmentation for Object Detection. *arXiv e-prints*, page arXiv:1906.04962, Jun 2019.
- [229] Changhee Han, Leonardo Rundo, Ryosuke Araki, Yujiro Furukawa, Giancarlo Mauri, Hideki Nakayama, and Hideaki Hayashi. Infinite brain MR images: Pggan-based data augmentation for tumor detection. *CoRR*, abs/1903.12564, 2019.
- [230] Changhee Han, Leonardo Rundo, Ryosuke Araki, Yudai Nagano, Yujiro Furukawa, Giancarlo Mauri, Hideki Nakayama, and Hideaki Hayashi. Combining Noise-to-Image and Image-to-Image GANs: Brain MR Image Augmentation for Tumor Detection. *arXiv e-prints*, page arXiv:1905.13456, May 2019.
- [231] Xiaoguang Han, Zhaoxuan Zhang, Dong Du, Mingdai Yang, Jingming Yu, Pan Pan, Xin Yang, Ligang Liu, Zixiang Xiong, and Shuguang Cui. Deep reinforcement learning of volume-guided progressive view inpainting for 3d point scene completion from a single depth image. *ArXiv*, abs/1903.04019, 2019.
- [232] A. Handa, V. Pătrăucean, S. Stent, and R. Cipolla. Scenenet: An annotated model generator for indoor scene understanding. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5737–5743, May 2016.
- [233] A. Handa, T. Whelan, J. McDonald, and A. J. Davison. A benchmark for rgb-d visual odometry, 3d reconstruction and slam. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1524–1531, May 2014.
- [234] Ankur Handa, Viorica Patraucean, Vijay Badrinarayanan, Simon Stent, and Roberto Cipolla. Scenenet: Understanding real world indoor scenes with synthetic data. *CoRR*, abs/1511.07041, 2015.
- [235] Markus Hartenfeller and Gisbert Schneider. Enabling future drug discovery by de novo design. *Wiley Interdisciplinary Reviews: Computational Molecular Science*, 1(5):742–759, 2011.
- [236] H. Hattori, V. N. Boddeti, K. Kitani, and T. Kanade. Learning scene-specific pedestrian detectors without real data. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3819–3827, June 2015.
- [237] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask r-cnn. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2980–2988, Oct 2017.
- [238] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.

- [239] J.B. Heaton and Jan Witte. Generating synthetic data to test financial strategies and investment products for regulatory compliance. Technical report, SSRN, March 2019.
- [240] Tobias Heimann, Peter Mountney, Matthias John, and Razvan Ionascu. Learning without labeling: Domain adaptation for ultrasound transducer localization. In Kensaku Mori, Ichiro Sakuma, Yoshinobu Sato, Christian Barillot, and Nassir Navab, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2013*, pages 49–56, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.
- [241] Jeremy Heitz and Daphne Koller. Learning spatial context: Using stuff to find things. In David Forsyth, Philip Torr, and Andrew Zisserman, editors, *Computer Vision – ECCV 2008*, pages 30–43, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg.
- [242] Dirk Helbing and Péter Molnár. Social force model for pedestrian dynamics. *Phys. Rev. E*, 51:4282–4286, May 1995.
- [243] Mark Hendrikx, Sebastiaan Meijer, Joeri Van Der Velden, and Alexandru Iosup. Procedural content generation for games: A survey. *ACM Trans. Multimedia Comput. Commun. Appl.*, 9(1):1:1–1:22, February 2013.
- [244] A. I. Hentati, L. Krichen, M. Fourati, and L. C. Fourati. Simulation tools, environments and frameworks for uav systems performance analysis. In *2018 14th International Wireless Communications Mobile Computing Conference (IWCMC)*, pages 1495–1500, June 2018.
- [245] Daniel Hernandez-Juarez, Lukas Schneider, Antonio Espinosa, David Vazquez, Antonio M. Lopez, Uwe Franke, Marc Pollefeys, and Juan Carlos Moure. Slanted stixels: Representing san francisco’s steepest streets. In *British Machine Vision Conference (BMVC)*, 2017, 2017.
- [246] Stefan Hinterstoisser, Vincent Lepetit, Paul Wohlhart, and Kurt Konolige. On pre-trained image features and synthetic images for deep learning. In Laura Leal-Taixé and Stefan Roth, editors, *Computer Vision – ECCV 2018 Workshops*, pages 682–697, Cham, 2019. Springer International Publishing.
- [247] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, November 1997.
- [248] Tomas Hodan, Pavel Haluza, Stepán Obdrzálek, Jiri Matas, Manolis I. A. Lourakis, and Xenophon Zabulis. T-LESS: an RGB-D dataset for 6d pose estimation of texture-less objects. *CoRR*, abs/1701.05498, 2017.
- [249] Judy Hoffman, Eric Tzeng, Taesung Park, Jun-Yan Zhu, Phillip Isola, Kate Saenko, Alexei A. Efros, and Trevor Darrell. Cycada: Cycle-consistent adversarial domain adaptation. *CoRR*, abs/1711.03213, 2017.
- [250] Judy Hoffman, Dequan Wang, Fisher Yu, and Trevor Darrell. Fcns in the wild: Pixel-level adversarial and constraint-based adaptation. *CoRR*, abs/1612.02649, 2016.

- [251] Steffen Hölldobler, Yvonne Kalinke, and Helko Lehmann. Designing a counter: Another case study of dynamics and activation landscapes in recurrent networks. In Gerhard Brewka, Christopher Habel, and Bernhard Nebel, editors, *KI-97: Advances in Artificial Intelligence*, pages 313–324, Berlin, Heidelberg, 1997. Springer Berlin Heidelberg.
- [252] W. Hong, Z. Wang, M. Yang, and J. Yuan. Conditional generative adversarial network for structured domain adaptation. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1335–1344, June 2018.
- [253] Le Hou, Ayush Agarwal, Dimitris Samaras, Tahsin M. Kurç, Rajarsi R. Gupta, and Joel H. Saltz. Unsupervised histopathology image synthesis. *CoRR*, abs/1712.05021, 2017.
- [254] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *CoRR*, abs/1704.04861, 2017.
- [255] Jeremy Howard and Sebastian Ruder. Fine-tuned language models for text classification. *CoRR*, abs/1801.06146, 2018.
- [256] G. Hu, X. Peng, Y. Yang, T. M. Hospedales, and J. Verbeek. Frankenstein: Learning deep face representations using small data. *IEEE Transactions on Image Processing*, 27(1):293–303, Jan 2018.
- [257] Guosheng Hu, Fei Yan, Chi-Ho Chan, Weihong Deng, William Christmas, Josef Kittler, and Neil Robertson. Face recognition using a unified 3d morphable model. In *Computer Vision – ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part VIII*, volume 9912 of *Lecture Notes in Computer Science*, pages 73–89, 10 2016.
- [258] Ronghang Hu, Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Kate Saenko. Learning to reason: End-to-end module networks for visual question answering. *CoRR*, abs/1704.05526, 2017.
- [259] Ronghang Hu, Marcus Rohrbach, Jacob Andreas, Trevor Darrell, and Kate Saenko. Modeling relationships in referential expressions with compositional modular networks. *CoRR*, abs/1611.09978, 2016.
- [260] Baichuan Huang, Deniz Bayazit, Daniel Ullman, Nakul Gopalan, and Stefanie Tellex. Flight, camera, action! using natural language and mixed reality to control a drone. In *Proceedings of the IEEE International Conference on Robotics and Automation*, 2019.
- [261] Gary B. Huang, Marwan Mattar, Tamara L. Berg, and Erik G. Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. 2008.
- [262] H. Huang, H. Guo, Z. Ding, Y. Chen, and X. Wu. 3d virtual modeling for crowd analysis. In *2015 IEEE International Conference on Information and Automation*, pages 2949–2954, Aug 2015.

- [263] Haoshuo Huang, Qi-Xing Huang, and Philipp Krähenbühl. Domain transfer through deep activation matching. In *ECCV*, 2018.
- [264] R. Huang, S. Zhang, T. Li, and R. He. Beyond face rotation: Global and local perception gan for photorealistic and identity preserving frontal view synthesis. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2458–2467, Oct 2017.
- [265] Xinyu Huang, Xinjing Cheng, Qichuan Geng, Binbin Cao, Dingfu Zhou, Peng Wang, Yuanqing Lin, and Ruigang Yang. The apolloscape dataset for autonomous driving. *CoRR*, abs/1803.06184, 2018.
- [266] Xun Huang, Ming-Yu Liu, Serge J. Belongie, and Jan Kautz. Multimodal unsupervised image-to-image translation. *CoRR*, abs/1804.04732, 2018.
- [267] Zhewei Huang, Wen Heng, and Shuchang Zhou. Learning to paint with model-based deep reinforcement learning. *ArXiv*, abs/1903.04411, 2019.
- [268] P. Huber, Z. Feng, W. Christmas, J. Kittler, and M. Rätsch. Fitting 3d morphable face models using local features. In *2015 IEEE International Conference on Image Processing (ICIP)*, pages 1195–1199, Sep. 2015.
- [269] Braden Hurl, Krzysztof Czarnecki, and Steven Lake Waslander. Precise synthetic image and lidar (presil) dataset for autonomous vehicle perception. *ArXiv*, abs/1905.00160, 2019.
- [270] Haroon Idrees, Muhammad Tayyab, Kishan Athrey, Dong Zhang, Somaya Al-Máadeed, Nasir M. Rajpoot, and Mubarak Shah. Composition loss for counting, density map estimation and localization in dense crowds. *CoRR*, abs/1808.01050, 2018.
- [271] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox. Flownet 2.0: Evolution of optical flow estimation with deep networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jul 2017.
- [272] E. Ilg, T. Saikia, M. Keuper, and T. Brox. Occlusions, motion and depth boundaries with a generic network for disparity, optical flow or scene flow estimation. In *European Conference on Computer Vision (ECCV)*, 2018.
- [273] T. Inoue, S. Choudhury, G. De Magistris, and S. Dasgupta. Transfer learning from synthetic to real images using variational autoencoders for precise position detection. In *2018 25th IEEE International Conference on Image Processing (ICIP)*, pages 2725–2729, Oct 2018.
- [274] C. Ionescu, D. Papava, V. Olaru, and C. Sminchisescu. Human3.6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(7):1325–1339, July 2014.
- [275] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-image translation with conditional adversarial networks. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5967–5976, 2017.

- [276] Max Jaderberg, Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Synthetic data and artificial neural networks for natural scene text recognition. *CoRR*, abs/1406.2227, 2014.
- [277] Max Jaderberg, Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Reading text in the wild with convolutional neural networks. *Int. J. Comput. Vision*, 116(1):1–20, 2016.
- [278] Nick Jakobi, Phil Husbands, and Inman Harvey. Noise and the reality gap: The use of simulation in evolutionary robotics. In Federico Morán, Alvaro Moreno, Juan Julián Merelo, and Pablo Chacón, editors, *Advances in Artificial Life*, pages 704–720, Berlin, Heidelberg, 1995. Springer Berlin Heidelberg.
- [279] Stephen James, Paul Wohlhart, Mrinal Kalakrishnan, Dmitry Kalashnikov, Alex Irpan, Julian Ibarz, Sergey Levine, Raia Hadsell, and Konstantinos Bousmalis. Sim-to-real via sim-to-sim: Data-efficient robotic grasping via randomized-to-canonical adaptation networks. *ArXiv*, abs/1812.07252, 2018.
- [280] Natasha Jaques, Shixiang Gu, Richard E. Turner, and Douglas Eck. Tuning recurrent neural networks with reinforcement learning. *CoRR*, abs/1611.02796, 2016.
- [281] Bo Ji and Tianyi Chen. Generative adversarial network for handwritten text. *CoRR*, abs/1907.11845, 2019.
- [282] Cheng-Bin Jin, Wonmo Jung, Seongsu Joo, Eunsik Park, Young Saem Ahn, In Ho Han, Jae-Il Lee, and Xuenan Cui. Deep CT to MR synthesis using paired and unpaired data. *CoRR*, abs/1805.10790, 2018.
- [283] Wengong Jin, Regina Barzilay, and Tommi Jaakkola. Junction Tree Variational Autoencoder for Molecular Graph Generation. *arXiv e-prints*, page arXiv:1802.04364, Feb 2018.
- [284] Junho Jo, Hyung Il Koo, Jae Woong Soh, and Nam Ik Cho. Handwritten text segmentation via end-to-end learning of convolutional neural network. *CoRR*, abs/1906.05229, 2019.
- [285] Vineet John, Lili Mou, Hareesh Bahuleyan, and Olga Vechtomova. Disentangled representation learning for non-parallel text style transfer. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 424–434, Florence, Italy, July 2019. Association for Computational Linguistics.
- [286] Gregory R. Johnson, Rory M. Donovan-Maiye, and Mary M. Maleckar. Generative Modeling with Conditional Autoencoders: Building an Integrated Cell. *arXiv e-prints*, page arXiv:1705.00092, Apr 2017.
- [287] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *ECCV*, 2016.

- [288] Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C. Lawrence Zitnick, and Ross B. Girshick. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1988–1997, 2017.
- [289] Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Judy Hoffman, Fei Fei Li, C Lawrence Zitnick, and Ross Girshick. Inferring and executing programs for visual reasoning. pages 3008–3017, 10 2017.
- [290] Justin Johnson, Ranjay Krishna, Michael Stark, Li-Jia Li, David A. Shamma, Michael S. Bernstein, and Li Fei-Fei. Image retrieval using scene graphs. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3668–3678, 2015.
- [291] Sam Johnson and Mark Everingham. Clustered pose and nonlinear appearance models for human pose estimation. In *Proceedings of the British Machine Vision Conference*, pages 12.1–12.11. BMVA Press, 2010. doi:10.5244/C.24.12.
- [292] 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)*, pages 746–753, May 2017.
- [293] D. Joo, D. Kim, and J. Kim. Generating a fusion image: One’s identity and another’s shape. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1635–1643, June 2018.
- [294] Armand Joulin and Tomas Mikolov. Inferring algorithmic patterns with stack-augmented recurrent nets. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1, NIPS’15*, pages 190–198, Cambridge, MA, USA, 2015. MIT Press.
- [295] Armand Joulin and Tomas Mikolov. Inferring algorithmic patterns with stack-augmented recurrent nets. *CoRR*, abs/1503.01007, 2015.
- [296] N. Justesen, P. Bontrager, J. Togelius, and S. Risi. Deep learning for video game playing. *IEEE Transactions on Games*, pages 1–1, 2019.
- [297] Niels Justesen, Ruben Rodriguez Torrado, Philip Bontrager, Ahmed Khalifa, Julian Togelius, and Sebastian Risi. Procedural level generation improves generality of deep reinforcement learning. *CoRR*, abs/1806.10729, 2018.
- [298] Artur Kadurin, Alexander Aliper, Andrey Kazennov, Polina Mamoshina, Quentin Vanhaelen, Kuzma Khrabrov, and Alexander Zhavoronkov. The cornucopia of meaningful leads: Applying deep adversarial autoencoders for new molecule development in oncology. *Oncotarget*, 8, 12 2016.
- [299] Artur Kadurin, Sergey I. Nikolenko, Kuzma Khrabrov, Alexander Aliper, and Alex Zhavoronkov. drugan: An advanced generative adversarial autoencoder model for de novo generation of new molecules with desired

molecular properties in silico. *Molecular Pharmaceutics*, 14(9):3098–3104, 2017.

- [300] Lukasz Kaiser and Ilya Sutskever. Neural gpus learn algorithms. *CoRR*, abs/1511.08228, 2016.
- [301] Dmitry Kalashnikov, Alex Irpan, Peter Pastor, Julian Ibarz, Alexander Herzog, Eric Jang, Deirdre Quillen, Ethan Holly, Mrinal Kalakrishnan, Vincent Vanhoucke, and Sergey Levine. Scalable deep reinforcement learning for vision-based robotic manipulation. In Aude Billard, Anca Dragan, Jan Peters, and Jun Morimoto, editors, *Proceedings of The 2nd Conference on Robot Learning*, volume 87 of *Proceedings of Machine Learning Research*, pages 651–673. PMLR, 29–31 Oct 2018.
- [302] Konstantinos Kamnitsas, Christian Baumgartner, Christian Ledig, Virginia Newcombe, Joanna Simpson, Andrew Kane, David Menon, Aditya Nori, Antonio Criminisi, Daniel Rueckert, and Ben Glocker. Unsupervised domain adaptation in brain lesion segmentation with adversarial networks. In Marc Niethammer, Martin Styner, Stephen Aylward, Hongtu Zhu, Ipek Oguz, Pew-Thian Yap, and Dinggang Shen, editors, *Information Processing in Medical Imaging*, pages 597–609, Cham, 2017. Springer International Publishing.
- [303] Konstantinos Kamnitsas, Christian Ledig, Virginia F.J. Newcombe, Joanna P. Simpson, Andrew D. Kane, David K. Menon, Daniel Rueckert, and Ben Glocker. Efficient multi-scale 3d cnn with fully connected crf for accurate brain lesion segmentation. *Medical Image Analysis*, 36:61 – 78, 2017.
- [304] Naoyuki Kan, Nagisa Kondo, Warapon Chinsatit, and Takeshi Saitoh. Effectiveness of data augmentation for cnn-based pupil center point detection. *2018 57th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE)*, pages 41–46, 2018.
- [305] Christoforos Kanellakis and George Nikolakopoulos. Survey on computer vision for uavs: Current developments and trends. *Journal of Intelligent & Robotic Systems*, 01 2017.
- [306] B. Kaneva, A. Torralba, and W. T. Freeman. Evaluation of image features using a photorealistic virtual world. In *2011 International Conference on Computer Vision*, pages 2282–2289, Nov 2011.
- [307] Yue Kang, Hang Yin, and Christian Berger. Test your self-driving algorithm: An overview of publicly available driving datasets and virtual testing environments. *IEEE Transactions on Intelligent Vehicles*, 4:171–185, 2019.
- [308] Neel Kant. Recent advances in neural program synthesis. *CoRR*, abs/1802.02353, 2018.
- [309] Amlan Kar, Aayush Prakash, Ming-Yu Liu, Eric Cameracci, Justin Yuan, Matt Rusiniak, David Acuna, Antonio Torralba, and Sanja Fidler. Metasim: Learning to generate synthetic datasets. *CoRR*, abs/1904.11621, 2019.

- [310] Brian Karis. Real shading in unreal engine 4. Technical report, Epic Games, 2013.
- [311] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *CoRR*, abs/1710.10196, 2018.
- [312] Uri Kartoun. A methodology to generate virtual patient repositories. *CoRR*, abs/1608.00570, 2016.
- [313] Janne Karttunen, Anssi Kanervisto, Ville Hautamäki, and Ville Kytki. From video game to real robot: The transfer between action spaces. *ArXiv*, abs/1905.00741, 2019.
- [314] Michal Kempka, Marek Wydmuch, Grzegorz Runc, Jakub Toczek, and Wojciech Jaskowski. Vizdoom: A doom-based AI research platform for visual reinforcement learning. *CoRR*, abs/1605.02097, 2016.
- [315] Alex Kendall, Yarin Gal, and Roberto Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. *CoRR*, abs/1705.07115, 2017.
- [316] A. R. Khadka, M. M. Oghaz, W. Matta, M. Cosentino, P. Remagnino, and V. Argyriou. Learning how to analyse crowd behaviour using synthetic data. In *Proceedings of the 32Nd International Conference on Computer Animation and Social Agents*, CASA '19, pages 11–14, New York, NY, USA, 2019. ACM.
- [317] Samin Khan, Buu Phan, Rick Salay, and Krzysztof Czarnecki. Procsy: Procedural synthetic dataset generation towards influence factor studies of semantic segmentation networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2019.
- [318] Rawal Khirodkar, Donghyun Yoo, and Kris M. Kitani. Vadra: Visual adversarial domain randomization and augmentation. *ArXiv*, abs/1812.00491, 2018.
- [319] M. Khodabandeh, H. R. V. Joze, I. Zharkov, and V. Pradeep. Diy human action dataset generation. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1529–152910, June 2018.
- [320] Kye-Hyeon Kim, Yeongjae Cheon, Sanghoon Hong, Byung-Seok Roh, and Minje Park. PVANET: deep but lightweight neural networks for real-time object detection. *CoRR*, abs/1608.08021, 2016.
- [321] Mariam Kiran, Paul Richmond, Mike Holcombe, Lee Shawn Chin, David Worth, and Chris Greenough. Flame: Simulating large populations of agents on parallel hardware architectures. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems: Volume 1 - Volume 1*, AAMAS '10, pages 1633–1636, Richland, SC, 2010. International Foundation for Autonomous Agents and Multiagent Systems.

- [322] Sosuke Kobayashi. Contextual augmentation: Data augmentation by words with paradigmatic relations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 452–457, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.
- [323] Levente Kocsis and Csaba Szepesvári. Bandit based monte-carlo planning. In Johannes Fürnkranz, Tobias Scheffer, and Myra Spiliopoulou, editors, *Machine Learning: ECML 2006*, pages 282–293, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.
- [324] N. Koenig and A. Howard. Design and use paradigms for gazebo, an open-source multi-robot simulator. In *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*, volume 3, pages 2149–2154 vol.3, Sep. 2004.
- [325] Eric Kolve, Roozbeh Mottaghi, Daniel Gordon, Yuke Zhu, Abhinav Gupta, and Ali Farhadi. AI2-THOR: an interactive 3d environment for visual AI. *CoRR*, abs/1712.05474, 2017.
- [326] Dimitrios Korkinof, Tobias Rijken, Michael O’Neill, Joseph Yearsley, Hugh Harvey, and Ben Glocker. High-resolution mammogram synthesis using progressive generative adversarial networks. *CoRR*, abs/1807.03401, 2018.
- [327] Adam Kortylewski, Bernhard Egger, Andreas Schneider, Thomas Gerig, Andreas Morel-Forster, and Thomas Vetter. Empirically analyzing the effect of dataset biases on deep face recognition systems. *CoRR*, abs/1712.01619, 2017.
- [328] Adam Kortylewski, Bernhard Egger, Andreas Schneider, Thomas Gerig, Andreas Morel-Forster, and Thomas Vetter. Analyzing and reducing the damage of dataset bias to face recognition with synthetic data. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2019.
- [329] Adam Kortylewski, Andreas Schneider, Thomas Gerig, Clemens Blumer, Bernhard Egger, Corius Reyneke, Andreas Morel-Forster, and Thomas Vetter. Priming deep neural networks with synthetic faces for enhanced performance. *CoRR*, abs/1811.08565, 2018.
- [330] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A. Shamma, Michael S. Bernstein, and Li Fei-Fei. Visual genome: Connecting language and vision using crowdsourced dense image annotations. *International Journal of Computer Vision*, 123(1):32–73, May 2017.
- [331] Praveen Krishnan and C. V. Jawahar. Hwnet v2: an efficient word image representation for handwritten documents. *International Journal on Document Analysis and Recognition (IJDAR)*, Jul 2019.

- [332] Srivatsan Krishnan, Behzad Boroujerdian, William Fu, Aleksandra Faust, and Vijay Janapa Reddi. Air learning: An AI research platform for algorithm-hardware benchmarking of autonomous aerial robots. *CoRR*, abs/1906.00421, 2019.
- [333] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'12, pages 1097–1105, USA, 2012. Curran Associates Inc.
- [334] Abhishek Kumar, Prasanna Sattigeri, Kahini Wadhawan, Leonid Karlinsky, Rogerio Feris, William T. Freeman, and Gregory Wornell. Co-regularized alignment for unsupervised domain adaptation. In *Proceedings of the 32Nd International Conference on Neural Information Processing Systems*, NIPS'18, pages 9367–9378, USA, 2018. Curran Associates Inc.
- [335] Karol Kurach, Marcin Andrychowicz, and Ilya Sutskever. Neural random access machines. *ICLR*, 2016.
- [336] Sefer Kurnaz, Okyay Kaynak, and Ekrem Konakoğlu. Adaptive neuro-fuzzy inference system based autonomous flight control of unmanned air vehicles. In *Proceedings of the 4th International Symposium on Neural Networks: Advances in Neural Networks*, ISNN '07, pages 14–21, Berlin, Heidelberg, 2007. Springer-Verlag.
- [337] Matt J. Kusner and José Miguel Hernández-Lobato. GANS for Sequences of Discrete Elements with the Gumbel-softmax Distribution. *arXiv e-prints*, page arXiv:1611.04051, Nov 2016.
- [338] Matt J. Kusner, Brooks Paige, and José Miguel Hernández-Lobato. Grammar Variational Autoencoder. *arXiv e-prints*, page arXiv:1703.01925, Mar 2017.
- [339] Denis Kuzminykh, Daniil Polykovskiy, Artur Kadurin, Alexander Zhebrak, Ivan Baskov, Sergey I. Nikolenko, Rim Shayakhmetov, and Alex Zhavoronkov. 3d molecular representations based on the wave transform for convolutional neural networks. *Molecular Pharmaceutics*, 15(10):4378–4385, 2018.
- [340] Dmitry Kuznichov, Alon Zvirin, Yaron Honen, and Ron Kimmel. Data augmentation for leaf segmentation and counting tasks in rosette plants. *ArXiv*, abs/1903.08583, 2019.
- [341] Kuan-Ting Lai, Chia-Chih Lin, Chun-Yao Kang, Mei-ENN Liao, and Ming-Syan Chen. Vivid: Virtual environment for visual deep learning. In *Proceedings of the 26th ACM International Conference on Multimedia*, MM '18, pages 1356–1359, New York, NY, USA, 2018. ACM.
- [342] Laminar Research. X-plane flight simulator, 2018.
- [343] Guillaume Lample and Devendra Singh Chaplot. Playing fps games with deep reinforcement learning. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, AAAI'17, pages 2140–2146. AAAI Press, 2017.

- [344] Greg Landrum. Rdkit: Open-source cheminformatics software. 2016.
- [345] Anders Boesen Lindbo Larsen, Søren Kaae Sønderby, Hugo Larochelle, and Ole Winther. Autoencoding beyond pixels using a learned similarity metric. In Maria Florina Balcan and Kilian Q. Weinberger, editors, *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 1558–1566, New York, New York, USA, 20–22 Jun 2016. PMLR.
- [346] Henning Lategahn, Andreas Geiger, and Bernd Kitt. Visual slam for autonomous ground vehicles. pages 1732–1737, 05 2011.
- [347] Beng Yong Lee, Lee Hung Liew, Wai Shiang Cheah, and Yin Chai Wang. Simulation videos for understanding occlusion effects on kernel based object tracking. In Sang-Soo Yeo, Yi Pan, Yang Sun Lee, and Hang Bae Chang, editors, *Computer Science and its Applications*, pages 139–147, Dordrecht, 2012. Springer Netherlands.
- [348] D. Lee, T. Ryan, and H. J. Kim. Autonomous landing of a vtol uav on a moving platform using image-based visual servoing. In *2012 IEEE International Conference on Robotics and Automation*, pages 971–976, May 2012.
- [349] Kevin Lee and David Moloney. An evaluation of synthetic data for deep learning stereo depth algorithms. In *Proceedings of the International Conference on Watermarking and Image Processing*, ICWIP 2017, pages 42–45, New York, NY, USA, 2017. ACM.
- [350] Scott Lee. Natural language generation for electronic health records. In *npj Digital Medicine*, 2018.
- [351] Mike Lewis and Angela Fan. Generative question answering: Learning to answer the whole question. In *International Conference on Learning Representations*, 2019.
- [352] Dong Li, Dongbin Zhao, Qichao Zhang, and Yaran Chen. Reinforcement learning and deep learning based lateral control for autonomous driving. *CoRR*, abs/1810.12778, 2018.
- [353] Jason Li, Ravi Gadde, Boris Ginsburg, and Vitaly Lavrukhin. Training neural speech recognition systems with synthetic speech augmentation. *CoRR*, abs/1811.00707, 2018.
- [354] K. Li, Y. Li, S. You, and N. Barnes. Photo-realistic simulation of road scene for data-driven methods in bad weather. In *2017 IEEE International Conference on Computer Vision Workshops (ICCVW)*, pages 491–500, Oct 2017.
- [355] Stan Z. Li and Anil K. Jain. *Handbook of Face Recognition*. Springer Publishing Company, Incorporated, 2nd edition, 2011.
- [356] Wei Li, Chengwei Pan, Rong Zhang, Jiaping Ren, Yuexin Ma, Jin Fang, Feilong Yan, Qichuan Geng, Xinyu Huang, Huajun Gong, Weiwei Xu, Guoping Wang, Dinesh Manocha, and Ruigang Yang. Aads: Augmented

- autonomous driving simulation using data-driven algorithms. *CoRR*, abs/1901.07849, 2019.
- [357] Qing Lian, Fengmao Lv, Lixin Duan, and Boqing Gong. Constructing self-motivated pyramid curriculums for cross-domain semantic segmentation: A non-adversarial approach. *ArXiv*, abs/1908.09547, 2019.
 - [358] Xiaodan Liang, Hao Zhang, and Eric P. Xing. Generative semantic manipulation with contrasting gan. *CoRR*, abs/1708.00315, 2017.
 - [359] J. Liebelt and C. Schmid. Multi-view object class detection with a 3d geometric model. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1688–1695, June 2010.
 - [360] J. Liebelt, C. Schmid, and K. Schertler. Viewpoint-independent object class detection using 3d feature maps. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, June 2008.
 - [361] Katrina Ligett, Seth Neel, Aaron Roth, Bo Waggoner, and Steven Z. Wu. Accuracy first: Selecting a differential privacy level for accuracy constrained erm. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 2566–2576. Curran Associates, Inc., 2017.
 - [362] T. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie. Feature pyramid networks for object detection. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 936–944, July 2017.
 - [363] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, Lubomir D. Bourdev, Ross B. Girshick, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. *CoRR*, abs/1405.0312, 2014.
 - [364] Zeming Lin, Jonas Gehring, Vasil Khalidov, and Gabriel Synnaeve. STAR-DATA: A starcraft AI research dataset. *CoRR*, abs/1708.02139, 2017.
 - [365] J. J. Little and A. Verri. Analysis of differential and matching methods for optical flow. In *[1989] Proceedings. Workshop on Visual Motion*, pages 173–180, March 1989.
 - [366] R.J.A. Little. Statistical analysis of masked data. *Journal of Official Statistics*, 9:407–426, 1993.
 - [367] Alexander H. Liu, Yen-Cheng Liu, Yu-Ying Yeh, and Yu-Chiang Frank Wang. A unified feature disentangler for multi-domain image translation and manipulation. In *Proceedings of the 32Nd International Conference on Neural Information Processing Systems*, NIPS’18, pages 2595–2604, USA, 2018. Curran Associates Inc.
 - [368] Fuchang Liu, Shuangjian Wang, Dandan Ding, Qingshu Yuan, Zhengwei Yao, Zhigeng Pan, and Haisheng Li. Retrieving indoor objects: 2d-3d alignment using single image and interactive roi-based refinement. *Computers & Graphics*, 70:108 – 117, 2018. CAD/Graphics 2017.

- [369] Guan-Horng Liu, Avinash Siravuru, Sai Prabhakar, Manuela M. Veloso, and George Kantor. Learning end-to-end multimodal sensor policies for autonomous navigation. *CoRR*, abs/1705.10422, 2017.
- [370] Ming-Yu Liu, Thomas Breuel, and Jan Kautz. Unsupervised image-to-image translation networks. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 700–708. Curran Associates, Inc., 2017.
- [371] Mingming Liu, Yanwen Guo, and Jun Wang. Indoor scene modeling from a single image using normal inference and edge features. *Vis. Comput.*, 33(10):1227–1240, October 2017.
- [372] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C. Berg. Ssd: Single shot multibox detector. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *Computer Vision – ECCV 2016*, pages 21–37, Cham, 2016. Springer International Publishing.
- [373] Zhao Liu, Jianke Zhu, Jiajun Bu, and Chun Chen. A survey of human pose estimation. *J. Vis. Comun. Image Represent.*, 32(C):10–19, October 2015.
- [374] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, December 2015.
- [375] Daniele Loiacono, Luigi Cardamone, and Pier Luca Lanzi. Simulated car racing championship: Competition software manual. *CoRR*, abs/1304.1672, 2013.
- [376] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael I. Jordan. Learning transferable features with deep adaptation networks. In *Proceedings of the 32Nd International Conference on International Conference on Machine Learning - Volume 37*, ICML’15, pages 97–105. JMLR.org, 2015.
- [377] Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I. Jordan. Unsupervised domain adaptation with residual transfer networks. In *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS’16, pages 136–144, USA, 2016. Curran Associates Inc.
- [378] Antonio M. López, Jiaolong Xu, José L. Gómez, David Vázquez, and Germán Ros. *From Virtual to Real World Visual Perception Using Domain Adaptation—The DPM as Example*, pages 243–258. Springer International Publishing, Cham, 2017.
- [379] Gilles Louppe, Joeri Hermans, and Kyle Cranmer. Adversarial Variational Optimization of Non-Differentiable Simulators. *arXiv e-prints*, page arXiv:1707.07113, Jul 2017.

- [380] W. Lu, G. Miklau, and V. Gupta. Generating private synthetic databases for untrusted system evaluation. In *2014 IEEE 30th International Conference on Data Engineering*, pages 652–663, March 2014.
- [381] Ihor Lubashevsky and Hiromasa Ando. Intermittent Control Properties of Car Following: Theory and Driving Simulator Experiments. *arXiv e-prints*, page arXiv:1609.01812, Sep 2016.
- [382] D. Ludl, T. Gulde, S. Thalji, and C. Curio. Using simulation to improve human pose estimation for corner cases. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 3575–3582, Nov 2018.
- [383] Mincong Luo, Yin Tong, and Jiachi Liu. Orthogonal policy gradient and autonomous driving application. *CoRR*, abs/1811.06151, 2018.
- [384] Liqian Ma, Qianru Sun, Bernt Schiele, and Luc Van Gool. A novel bilevel paradigm for image-to-image translation. *ArXiv*, abs/1904.09028, 2019.
- [385] Rui Ma, Akshay Gadi Patil, Matthew Fisher, Manyi Li, Sören Pirk, Binh-Son Hua, Sai-Kit Yeung, Xin Tong, Leonidas J. Guibas, and Hao Zhang. Language-driven synthesis of 3d scenes from scene databases. *ACM Trans. Graph.*, 37:212:1–212:16, 2018.
- [386] R. Madaan, D. Maturana, and S. Scherer. Wire detection using synthetic data and dilated convolutional networks for unmanned aerial vehicles. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3487–3494, Sep. 2017.
- [387] Faisal Mahmood, Richard Chen, and Nicholas J. Durr. Unsupervised reverse domain adaptation for synthetic medical images via adversarial training. *CoRR*, abs/1711.06606, 2017.
- [388] Faisal Mahmood, Richard J. Chen, Sandra Sudarsky, Daphne Yu, and Nicholas J. Durr. Deep learning with cinematic rendering - fine-tuning deep neural networks using photorealistic medical images. *CoRR*, abs/1805.08400, 2018.
- [389] Aakif Mairaj, Asif I. Baba, and Ahmad Y. Javaid. Application specific drone simulators: Recent advances and challenges. *Simulation Modelling Practice and Theory*, 94:100 – 117, 2019.
- [390] Alireza Makhzani, Jonathon Shlens, Navdeep Jaity, and Ian Goodfellow. Adversarial autoencoders. In *International Conference on Learning Representations*, 2016.
- [391] J. Malik, A. Elhayek, F. Nunnari, K. Varanasi, K. Tamaddon, A. Heloir, and D. Stricker. Deephtps: End-to-end estimation of 3d hand pose and shape by learning from synthetic depth. In *2018 International Conference on 3D Vision (3DV)*, pages 110–119, Sep. 2018.
- [392] Zohar Manna and Richard J. Waldinger. Toward automatic program synthesis. *Commun. ACM*, 14(3):151–165, March 1971.

- [393] Manolis Savva*, Abhishek Kadian*, Oleksandr Maksymets*, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, Vladlen Koltun, Jitendra Malik, Devi Parikh, and Dhruv Batra. Habitat: A Platform for Embodied AI Research. *arXiv preprint arXiv:1904.01201*, 2019.
- [394] X. Mao, Q. Li, H. Xie, R. Y. K. Lau, Z. Wang, and S. P. Smolley. Least squares generative adversarial networks. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2813–2821, Oct 2017.
- [395] Alina Marcu, Dragoș Costea, Vlad Licăreț, Mihai Pîrvu, Emil Slușanschi, and Marius Leordeanu. Safeuav: Learning to estimate depth and safe landing areas for uavs from synthetic data. In Laura Leal-Taixé and Stefan Roth, editors, *Computer Vision – ECCV 2018 Workshops*, pages 43–58, Cham, 2019. Springer International Publishing.
- [396] J. Marín, D. Vázquez, D. Gerónimo, and A. M. López. Learning appearance in virtual scenarios for pedestrian detection. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 137–144, June 2010.
- [397] Karl Mason, Sadegh Vejdan, and Santiago Grijalva. An ”on the fly” framework for efficiently generating synthetic big data sets. *ArXiv*, abs/1903.06798, 2019.
- [398] Rainer Mautz and Sebastian Tilch. Survey of optical indoor positioning systems. pages 1–7, 09 2011.
- [399] Nikolaus Mayer, Eddy Ilg, Philipp Fischer, Caner Hazirbas, Daniel Cremers, Alexey Dosovitskiy, and Thomas Brox. What makes good synthetic training data for learning disparity and optical flow estimation? *Int. J. Comput. Vision*, 126(9):942–960, September 2018.
- [400] Nikolaus Mayer, Eddy Ilg, Philip Häusser, Philipp Fischer, Daniel Cremers, Alexey Dosovitskiy, and Thomas Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. *CoRR*, abs/1512.02134, 2015.
- [401] J. McCormac, A. Handa, S. Leutenegger, and A. J. Davison. Scenenet rgbd: Can 5m synthetic images beat generic imagenet pre-training on indoor segmentation? In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2697–2706, Oct 2017.
- [402] S. McLachlan, K. Dube, and T. Gallagher. Using the caremap with health incidents statistics for generating the realistic synthetic electronic healthcare record. In *2016 IEEE International Conference on Healthcare Informatics (ICHI)*, pages 439–448, Oct 2016.
- [403] Scott McLachlan. *Realism in synthetic data generation*. PhD thesis, Massey University, Palmerston North, New Zealand, 2017.
- [404] H. Brendan McMahan, Daniel Ramage, Kunal Talwar, and Li Zhang. Learning differentially private recurrent language models. In *International Conference on Learning Representations*, 2018.

- [405] Mdclone: Introducing a new clinical data paradigm, 2016.
- [406] Bhairav Mehta, Manfred Diaz, Florian Golemo, Christopher Joseph Pal, and Liam Paull. Active domain randomization. *ArXiv*, abs/1904.04762, 2019.
- [407] S. Meister and D. Kondermann. Real versus realistically rendered scenes for optical flow evaluation. In *2011 14th ITG Conference on Electronic Media Technology*, pages 1–6, March 2011.
- [408] Oren Melamud and Chaitanya Shivade. Towards automatic generation of shareable synthetic clinical notes using neural language models. *ArXiv*, abs/1905.07002, 2019.
- [409] Moritz Menze and Andreas Geiger. Object scene flow for autonomous vehicles. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.
- [410] Juan Julián Merelo Guervós, Antonio Fernández-Ares, Antonio Álvarez-Caballero, Pablo García-Sánchez, and Víctor M. Rivas. Reddwarfdata: a simplified dataset of starcraft matches. *CoRR*, abs/1712.10179, 2017.
- [411] Stefan Milz, Georg Arbeiter, Christian Witt, Bassam Abdallah, and Senthil Yogamani. Visual slam for automated driving: Exploring the applications of deep learning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2018.
- [412] P. Mitra, A. Choudhury, V. R. Aparow, G. Kulandaivelu, and J. Dauwels. Towards modeling of perception errors in autonomous vehicles. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 3024–3029, Nov 2018.
- [413] Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In Maria Florina Balcan and Kilian Q. Weinberger, editors, *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 1928–1937, New York, New York, USA, 20–22 Jun 2016. PMLR.
- [414] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, February 2015.
- [415] Kaichun Mo, Shilin Zhu, Angel X. Chang, Li Yi, Subarna Tripathi, Leonidas J. Guibas, and Hao Su. Partnet: A large-scale benchmark for fine-grained and hierarchical part-level 3d object understanding. *CoRR*, abs/1812.02713, 2018.

- [416] Boris Moiseev, Artem Konev, Alexander Chigorin, and Anton Konushin. Evaluation of traffic sign recognition methods trained on synthetically generated data. In Jacques Blanc-Talon, Andrzej Kasinski, Wilfried Philips, Dan Popescu, and Paul Scheunders, editors, *Advanced Concepts for Intelligent Vision Systems*, pages 576–583, Cham, 2013. Springer International Publishing.
- [417] Linda J. Moniz, Anna L. Buczak, Lang M. Hung, Steven Babin, Michael Dorko, and Joseph M. Lombardo. Construction and validation of synthetic electronic medical records. In *Online journal of public health informatics*, 2009.
- [418] Pau Bramon Mora. Deep 3d pose regression of real objects trained with synthetic data. M.Sc. Thesis, *Universitat Polit cnica de Catalunya (UPC)*, 2019.
- [419] Yair Movshovitz-Attias, Takeo Kanade, and Yaser Sheikh. How useful is photo-realistic rendering for visual learning? In Gang Hua and Herv  J gou, editors, *Computer Vision – ECCV 2016 Workshops*, pages 202–217, Cham, 2016. Springer International Publishing.
- [420] Yazan Mualla, Wenshuai Bai, St phane Galland, and Christophe Nicolle. Comparison of agent-based simulation frameworks for unmanned aerial transportation applications. *Procedia Computer Science*, 130:791 – 796, 2018. The 9th International Conference on Ambient Systems, Networks and Technologies (ANT 2018) / The 8th International Conference on Sustainable Energy Information Technology (SEIT-2018) / Affiliated Workshops.
- [421] Reiichiro Nakano. Neural painters: A learned differentiable constraint for generating brushstroke paintings. *ArXiv*, abs/1904.08410, 2019.
- [422] Thomas Neff. *Data Augmentation in Deep Learning using Generative Adversarial Networks*. PhD thesis, Graz University of Technology, 2018.
- [423] D. Nie, R. Trullo, J. Lian, L. Wang, C. Petitjean, S. Ruan, Q. Wang, and D. Shen. Medical image synthesis with deep convolutional adversarial networks. *IEEE Transactions on Biomedical Engineering*, 65(12):2720–2730, Dec 2018.
- [424] Dong Nie, Roger Trullo, Caroline Petitjean, Su Ruan, and Dinggang Shen. Medical image synthesis with context-aware generative adversarial networks. *CoRR*, abs/1612.05362, 2016.
- [425] Michael J. North, Nicholson T. Collier, Jonathan Ozik, Eric R. Tatara, Charles M. Macal, Mark Bragen, and Pam Sydelko. Complex adaptive systems modeling with repast simphony. *Complex Adaptive Systems Modeling*, 1(1):3, Mar 2013.
- [426] Farzan Erlik Nowruzi, Prince Kapoor, Dhanvin Kolhatkar, Fahed Al Has-sanan, Robert Lagani re, and Julien Rebut. How much real data do we actually need: Analyzing object detection performance using synthetic and real data. *ArXiv*, abs/1907.07061, 2019.

- [427] Michael O’Byrne, Vikram Pakrashi, Franck Schoefs, and Bidisha Ghosh. Semantic segmentation of underwater imagery using deep networks trained on synthetic imagery. *Journal of Marine Science and Engineering*, 6(3), 2018.
- [428] Augustus Odena, Christopher Olah, and Jonathon Shlens. Conditional image synthesis with auxiliary classifier gans. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML’17, pages 2642–2651. JMLR.org, 2017.
- [429] Junhyuk Oh, Valliappa Chockalingam, Satinder Singh, and Honglak Lee. Control of memory, active perception, and action in minecraft. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, ICML’16, pages 2790–2799. JMLR.org, 2016.
- [430] Marcus Olivecrona, Thomas Blaschke, Ola Engkvist, and Hongming Chen. Molecular de-novo design through deep reinforcement learning. *Journal of Cheminformatics*, 9(1):48, Sep 2017.
- [431] Elizabeth A. Olson, Corina Barbalata, Junming Zhang, Katherine A. Skinner, and Matthew Johnson-Roberson. Synthetic data generation for deep learning of underwater disparity estimation. *OCEANS 2018 MTS/IEEE Charleston*, pages 1–6, 2018.
- [432] OpenAI, Marcin Andrychowicz, Bowen Baker, Maciek Chociej, Rafal Józefowicz, Bob McGrew, Jakub Pachocki, Arthur Petron, Matthias Plappert, Glenn Powell, Alex Ray, Jonas Schneider, Szymon Sidor, Josh Tobin, Peter Welinder, Lilian Weng, and Wojciech Zaremba. Learning dexterous in-hand manipulation. *CoRR*, 2018.
- [433] Takayuki Osa, Joni Pajarinen, Gerhard Neumann, J. Andrew Bagnell, Pieter Abbeel, and Jan Peters. An algorithmic perspective on imitation learning. *CoRR*, abs/1811.06711, 2018.
- [434] Pavel Ostyakov, Roman Suvorov, Elizaveta Logacheva, Oleg Khomenko, and Sergey I. Nikolenko. SEIGAN: towards compositional image generation by simultaneously learning to segment, enhance, and inpaint. *CoRR*, abs/1811.07630, 2018.
- [435] Brian Paden, Michal Čáp, Sze Zheng Yong, Dmitry Yershov, and Emilio Frazzoli. A survey of motion planning and control techniques for self-driving urban vehicles. *IEEE Transactions on Intelligent Vehicles*, 1, 04 2016.
- [436] Nicolas Papernot, Martín Abadi, Úlfar Erlingsson, Ian J. Goodfellow, and Kunal Talwar. Semi-supervised knowledge transfer for deep learning from private training data. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*, 2017.
- [437] J. Papon and M. Schoeler. Semantic pose using deep networks trained on synthetic rgb-d. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 774–782, Dec 2015.

- [438] Yoav I. H. Parish and Pascal Müller. Procedural modeling of cities. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '01, pages 301–308, New York, NY, USA, 2001. ACM.
- [439] Alexander Pashevich, Robin A. M. Strudel, Igor Kalevatykh, Ivan Laptev, and Cordelia Schmid. Learning to augment synthetic images for sim2real policy transfer. *CoRR*, abs/1903.07740, 2019.
- [440] Shreyas Patel, Ashutosh Kakadiya, Maitrey Mehta, Raj Derasari, Rahul Patel, and Ratnik Gandhi. Correlated discrete data generation using adversarial training. *CoRR*, abs/1804.00925, 2018.
- [441] V. M. Patel, R. Gopalan, R. Li, and R. Chellappa. Visual domain adaptation: A survey of recent advances. *IEEE Signal Processing Magazine*, 32(3):53–69, May 2015.
- [442] Martin Pecka, Karel Zimmermann, Matx011Bj Petrlx00EDk, and Tomx00E1x0161 Svoboda. Data-driven policy transfer with imprecise perception simulation. *IEEE Robotics and Automation Letters*, 3:3916–3921, 2018.
- [443] Scott Drew Pendleton, Hans Andersen, Xinxin Du, Xiaotong Shen, Malika Meghjani, You Hong Eng, Daniela Rus, and Marcelo H. Ang. Perception, planning, control, and coordination for autonomous vehicles. *Machines*, 5(1), 2017.
- [444] Xingchao Peng, Baochen Sun, Karim Ali, and Kate Saenko. Exploring invariances in deep convolutional neural networks using synthetic images. *CoRR*, abs/1412.7122, 2014.
- [445] Xingchao Peng, Baochen Sun, Karim Ali, and Kate Saenko. Learning deep object detectors from 3d models. In *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, ICCV '15, pages 1278–1286, Washington, DC, USA, 2015. IEEE Computer Society.
- [446] Peng Wang, Xiaohui Shen, Zhe Lin, S. Cohen, B. Price, and A. Yuille. Towards unified depth and semantic prediction from a single image. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2800–2809, June 2015.
- [447] Bojan Pepik, Rodrigo Benenson, Tobias Ritschel, and Bernt Schiele. What is holding back convnets for detection? *CoRR*, abs/1508.02844, 2015.
- [448] Gabriel Pereyra, George Tucker, Jan Chorowski, Lukasz Kaiser, and Geoffrey E. Hinton. Regularizing neural networks by penalizing confident output distributions. *CoRR*, abs/1701.06548, 2017.
- [449] Patrick Pérez, Michel Gangnet, and Andrew Blake. Poisson image editing. *ACM Trans. Graph.*, 22(3):313–318, July 2003.
- [450] M. Peris, S. Martull, A. Maki, Y. Ohkawa, and K. Fukui. Towards a simulation driven stereo vision system. In *Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012)*, pages 1038–1042, Nov 2012.

- [451] E. Perot, M. Jaritz, M. Toromanoff, and R. d. Charette. End-to-end driving in a realistic racing game with deep reinforcement learning. In *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 474–475, July 2017.
- [452] Haoyue Ping, Julia Stoyanovich, and Bill Howe. Datasynthesizer: Privacy-preserving synthetic datasets. In *Proceedings of the 29th International Conference on Scientific and Statistical Database Management*, SSDBM ’17, pages 42:1–42:5, New York, NY, USA, 2017. ACM.
- [453] N. Pinto, Y. Barhomi, D. D. Cox, and J. J. DiCarlo. Comparing state-of-the-art visual features on invariant object recognition tasks. In *2011 IEEE Workshop on Applications of Computer Vision (WACV)*, pages 463–470, Jan 2011.
- [454] H. Pirsiavash, D. Ramanan, and C. C. Fowlkes. Globally-optimal greedy algorithms for tracking a variable number of objects. In *CVPR 2011*, pages 1201–1208, June 2011.
- [455] B. Planche, Z. Wu, K. Ma, S. Sun, S. Kluckner, O. Lehmann, T. Chen, A. Hutter, S. Zakharov, H. Kosch, and J. Ernst. Depthsynth: Real-time realistic synthetic data generation from cad models for 2.5d recognition. In *2017 International Conference on 3D Vision (3DV)*, pages 1–10, Oct 2017.
- [456] Daniil Polykovskiy, Alexander Zhebrak, Benjamin Sanchez-Lengeling, Sergey Golovanov, Oktai Tatanov, Stanislav Belyaev, Rauf Kurbanov, Aleksey Artamonov, Vladimir Aladinsky, Mark Veselov, Artur Kadurin, Sergey I. Nikolenko, Alan Aspuru-Guzlik, and Alex Zhavoronkov. Molecular sets (moses): A benchmarking platform for molecular generation models, 2018.
- [457] Daniil Polykovskiy, Alexander Zhebrak, Dmitry Vetrov, Yan Ivanenkov, Vladimir Aladinsky, Polina Mamoshina, Marine Bozdaganyan, Alexander Aliper, Alex Zhavoronkov, and Artur Kadurin. Entangled conditional adversarial autoencoder for de novo drug discovery. *Molecular Pharmaceutics*, 15(10):4398–4405, 2018. PMID: 30180591.
- [458] Dean A. Pomerleau. Advances in neural information processing systems 1. chapter ALVINN: An Autonomous Land Vehicle in a Neural Network, pages 305–313. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1989.
- [459] Mariya Popova, Olexandr Isayev, and Alexander Tropsha. Deep reinforcement learning for de novo drug design. *Science Advances*, 4(7), 2018.
- [460] Ronald Poppe. A survey on vision-based human action recognition. *Image Vision Comput.*, 28(6):976–990, June 2010.
- [461] Y. A. Prabowo, B. R. Trilaksono, and F. R. Triputra. Hardware in-the-loop simulation for visual servoing of fixed wing uav. In *2015 International Conference on Electrical Engineering and Informatics (ICEEI)*, pages 247–252, Aug 2015.

- [462] Aayush Prakash, Shaad Boochoon, Mark Brophy, David Acuna, Eric Cameracci, Gavriel State, Omer Shapira, and Stanley T. Birchfield. Structured domain randomization: Bridging the reality gap by context-aware synthetic data. *CoRR*, abs/1810.10093, 2018.
- [463] P. Prusinkiewicz and J. Hanan. *Lindenmayer systems, fractals, and plants*, volume 79 of *Springer Science & Business Media*. Springer, 2013.
- [464] S. Qi, Y. Zhu, S. Huang, C. Jiang, and S. Zhu. Human-centric indoor scene synthesis using stochastic grammar. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5899–5908, June 2018.
- [465] Xiaojuan Qi, Renjie Liao, Jiaya Jia, Sanja Fidler, and Raquel Urtasun. 3d graph neural networks for rgbd semantic segmentation. *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 5209–5218, 2017.
- [466] Weichao Qiu and Alan L. Yuille. Unrealcv: Connecting computer vision to unreal engine. *CoRR*, abs/1609.01326, 2016.
- [467] Weichao Qiu, Fangwei Zhong, Yi Zhang, Zihao Xiao Siyuan Qiao, Tae Soo Kim, Yizhou Wang, and Alan Yuille. Unrealcv: Virtual worlds for computer vision. *ACM Multimedia Open Source Software Competition*, 2017.
- [468] R. Queiroz, M. Cohen, J. L. Moreira, A. Braun, J. C. Jacques Junior, and S. R. Musse. Generating facial ground truth with synthetic faces. In *2010 23rd SIBGRAPI Conference on Graphics, Patterns and Images*, pages 25–31, Aug 2010.
- [469] Craig Quiter and Maik Ernst. deepdrive/deepdrive: 2.0, March 2018.
- [470] F. Z. Qureshi and D. Terzopoulos. Surveillance in virtual reality: System design and multi-camera control. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, June 2007.
- [471] Mahdi Rad, Markus Oberweger, and Vincent Lepetit. Feature mapping for learning fast and accurate 3d pose inference from synthetic images. *CoRR*, abs/1712.03904, 2017.
- [472] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, 2016.
- [473] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. 2018.
- [474] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2018.

- [475] H. Ragheb, S. Velastin, P. Remagnino, and T. Ellis. Vihasi: Virtual human action silhouette data for the performance evaluation of silhouette-based action recognition methods. In *2008 Second ACM/IEEE International Conference on Distributed Smart Cameras*, pages 1–10, Sep. 2008.
- [476] T.E. Raghunathan, J.P. Reiter, and D.B. Rubin. Multiple imputation for statistical disclosure limitation. *Journal of Official Statistics*, 19(1):1–16, 2003.
- [477] Param S. Rajpura, Ravi S. Hegde, and Hristo Bojinov. Object detection using deep cnns trained on synthetic images. *CoRR*, abs/1706.06782, 2017.
- [478] R. Ranjan, V. M. Patel, and R. Chellappa. Hyperface: A deep multi-task learning framework for face detection, landmark localization, pose estimation, and gender recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(1):121–135, Jan 2019.
- [479] Ahmad Rashid, Alan Do-Omri, Md. Akmal Haidar, Qun Liu, and Mehdi Rezagholizadeh. Bilingual-GAN: A step towards parallel text generation. In *Proceedings of the Workshop on Methods for Optimizing and Evaluating Neural Language Generation*, pages 55–64, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.
- [480] Dino Stephen Ratcliffe, Sam Devlin, Udo Kruschwitz, and Luca Citi. Clyde: A deep reinforcement learning doom playing agent. In *AAAI Workshops*, 2017.
- [481] Alexander J Ratner, Henry Ehrenberg, Zeshan Hussain, Jared Dunnmon, and Christopher Ré. Learning to compose domain-specific transformations for data augmentation. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 3236–3246. Curran Associates, Inc., 2017.
- [482] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *CoRR*, abs/1804.02767, 2018.
- [483] Jerome P. Reiter. Releasing multiply imputed, synthetic public use micro-data: An illustration and empirical study. *Journal of the Royal Statistical Society Series A*, 168:185–205, 01 2005.
- [484] Jerome P Reiter and Jörg Drechsler. Releasing multiply-imputed synthetic data generated in two stages to protect confidentiality. *Statistica Sinica*, 20, 07 2007.
- [485] Jerome P. Reiter and Trivellore E. Raghunathan. The multiple adaptations of multiple imputation. *Journal of the American Statistical Association*, 102:1462–1471, 2007.
- [486] K. Rematas, I. Kemelmacher-Shlizerman, B. Curless, and S. Seitz. Soccer on your tabletop. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4738–4747, June 2018.

- [487] Tal Remez, Jonathan Huang, and Matthew R. Brown. Learning to segment via cut-and-paste. In *ECCV*, 2018.
- [488] Jian Ren, Ilker Hacihaliloglu, Eric A. Singer, David J. Foran, and Xin Qi. Adversarial domain adaptation for classification of prostate histopathology whole-slide images. In *Medical Image Computing and Computer Assisted Intervention - MICCAI 2018 - 21st International Conference, Granada, Spain, September 16-20, 2018, Proceedings, Part II*, pages 201–209, 2018.
- [489] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 39(06):1137–1149, jun 2017.
- [490] Zhongzheng Ren and Yong Jae Lee. Cross-domain self-supervised multi-task feature learning using synthetic imagery. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 762–771, 2018.
- [491] Zhongzheng Ren, Yong Jae Lee, and Michael S. Ryoo. Learning to anonymize faces for privacy preserving action detection. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, *Computer Vision – ECCV 2018*, pages 639–655, Cham, 2018. Springer International Publishing.
- [492] Jean-Louis Reymond, Lars Ruddigkeit, Lorenz Blum, and Ruud van Deursen. The enumeration of chemical space. *Wiley Interdisciplinary Reviews: Computational Molecular Science*, 2(5):717–733, 2012.
- [493] Stephan R. Richter, Zeeshan Hayder, and Vladlen Koltun. Playing for benchmarks. *CoRR*, abs/1709.07322, 2017.
- [494] Stephan R. Richter, Vibhav Vineet, Stefan Roth, and Vladlen Koltun. Playing for data: Ground truth from computer games. *CoRR*, abs/1608.02192, 2016.
- [495] Alexandre Robicquet, Amir Sadeghian, Alexandre Alahi, and Silvio Savarese. Learning social etiquette: Human trajectory understanding in crowded scenes. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *Computer Vision – ECCV 2016*, pages 549–565, Cham, 2016. Springer International Publishing.
- [496] E. Rohmer, S. P. N. Singh, and M. Freese. V-rep: A versatile and scalable robot simulation framework. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1321–1326, Nov 2013.
- [497] L. Oyuki Rojas-Perez, R. Munguia-Silva, and J. Martinez-Carranza. Real-time landing zone detection for uavs using single aerial images. In *IMAV2018, 10th International Micro Air Vehicle Conference (IMAV)*, Nov 2018.
- [498] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, volume 9351 of *LNCS*, pages 234–241. Springer, 2015. (available on arXiv:1505.04597 [cs.CV]).

- [499] 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 *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3234–3243, June 2016.
- [500] German Ros, Laura Sellart, Gabriel Villalonga, Elias Maidanik, Francisco Molero, Marc Garcia, Adriana Cedeño, Francisco Perez, Didier Ramirez, Eduardo Escobar, Jose Luis Gomez, David Vazquez, and Antonio M. Lopez. *Semantic Segmentation of Urban Scenes via Domain Adaptation of SYNTHIA*, pages 227–241. Springer International Publishing, Cham, 2017.
- [501] Germán Ros, Simon Stent, Pablo F. Alcantarilla, and Tomoki Watanabe. Training constrained deconvolutional networks for road scene semantic segmentation. *CoRR*, abs/1604.01545, 2016.
- [502] Alexander B. Rosenberg, Rupali P. Patwardhan, Jay Shendure, and Georg Seelig. Learning the sequence determinants of alternative splicing from millions of random sequences. *Cell*, 163(3):698 – 711, 2015.
- [503] Francisca Rosique, Pedro J. Navarro, Carlos Fernández, and Antonio Padilla. A systematic review of perception system and simulators for autonomous vehicles research. *Sensors*, 19(3), 2019.
- [504] A. Rozantsev, M. Salzmann, and P. Fua. Beyond sharing weights for deep domain adaptation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(4):801–814, April 2019.
- [505] Artem Rozantsev, Vincent Lepetit, and Pascal Fua. On rendering synthetic images for training an object detector. *Computer Vision and Image Understanding*, 137:24 – 37, 2015.
- [506] D.B. Rubin. Discussion: Statistical disclosure limitation. *Journal of Official Statistics*, 9:462–468, 1993.
- [507] Nataniel Ruiz, Samuel Schulter, and Manmohan Chandraker. Learning to simulate. In *International Conference on Learning Representations*, 2019.
- [508] Andrei A. Rusu, Neil C. Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. Progressive neural networks. *CoRR*, abs/1606.04671, 2016.
- [509] Andrei A. Rusu, Matej Vecerik, Thomas Rothörl, Nicolas Heess, Razvan Pascanu, and Raia Hadsell. Sim-to-real robot learning from pixels with progressive nets. In *CoRL*, 2017.
- [510] Luise Valentin Rygaard. Using synthesized speech to improve speech recognition for low-resource languages. In *Grace Hopper Celebration 2015, Poster Session*, 2015.
- [511] Joseph S. Lombardo and Linda Moniz. A method for generation and distribution of synthetic medical record data for evaluation of disease-monitoring systems. *Johns Hopkins APL Technical Digest (Applied Physics Laboratory)*, 27, 01 2008.

- [512] Yunus Saatci and Andrew G Wilson. Bayesian gan. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 3622–3631. Curran Associates, Inc., 2017.
- [513] Fereshteh Sadeghi and Sergey Levine. Cad2rl: Real single-image flight without a single real image. *ArXiv*, abs/1611.04201, 2017.
- [514] Kuniaki Saito, Yoshitaka Ushiku, and Tatsuya Harada. Asymmetric tri-training for unsupervised domain adaptation. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML’17, pages 2988–2997. JMLR.org, 2017.
- [515] Fatemeh Sadat Saleh, Mohammad Sadegh Aliakbarian, Mathieu Salzmann, Lars Petersson, and Jose M. Alvarez. Effective use of synthetic data for urban scene semantic segmentation. *CoRR*, abs/1807.06132, 2018.
- [516] Ahmad El Sallab, Mohammed Abdou, Etienne Perot, and Senthil Yogamani. End-to-end deep reinforcement learning for lane keeping assist. *ArXiv*, abs/1612.04340, 2016.
- [517] Ahmad El Sallab, Mohammed Abdou, Etienne Perot, and Senthil Yogamani. Deep reinforcement learning framework for autonomous driving. *ArXiv*, abs/1704.02532, 2017.
- [518] Eder Santana and George Hotz. Learning a driving simulator. *ArXiv*, abs/1608.01230, 2016.
- [519] Adam Santoro, David Raposo, David G. T. Barrett, Mateusz Malinowski, Razvan Pascanu, Peter W. Battaglia, and Timothy P. Lillicrap. A simple neural network module for relational reasoning. In *NIPS*, 2017.
- [520] Kevin Santoso and Gede Putra Kusuma. Face recognition using modified openface. *Procedia Computer Science*, 135:510 – 517, 2018. The 3rd International Conference on Computer Science and Computational Intelligence (ICCS 2018) : Empowering Smart Technology in Digital Era for a Better Life.
- [521] Manolis Savva, Angel X. Chang, Alexey Dosovitskiy, Thomas Funkhouser, and Vladlen Koltun. MINOS: Multimodal indoor simulator for navigation in complex environments. *arXiv:1712.03931*, 2017.
- [522] Ashutosh Saxena, Justin Driemeyer, and Andrew Y. Ng. Robotic grasping of novel objects using vision. *The International Journal of Robotics Research*, 27(2):157–173, 2008.
- [523] T. Sayre-McCord, W. Guerra, A. Antonini, J. Arneberg, A. Brown, G. Cavalheiro, Y. Fang, A. Gorodetsky, D. McCoy, S. Quilter, F. Riether, E. Tal, Y. Terzioglu, L. Carbone, and S. Karaman. Visual-inertial navigation algorithm development using photorealistic camera simulation in the loop. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2566–2573, May 2018.

- [524] D. Scharstein, R. Szeliski, and R. Zabih. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. In *Proceedings IEEE Workshop on Stereo and Multi-Baseline Vision (SMBV 2001)*, pages 131–140, Dec 2001.
- [525] Daniel Scharstein, Heiko Hirschmüller, York Kitajima, Greg Krathwohl, Nera Nesic, Xi Wang, and Porter Westling. High-resolution stereo datasets with subpixel-accurate ground truth. In Xiaoyi Jiang, Joachim Hornegger, and Reinhard Koch, editors, *GCPR*, volume 8753 of *Lecture Notes in Computer Science*, pages 31–42. Springer, 2014.
- [526] Tom Schaul, Julian Togelius, and Jürgen Schmidhuber. Measuring intelligence through games. *CoRR*, abs/1109.1314, 2011.
- [527] Matthew J. Schneider and John M. Abowd. A new method for protecting interrelated time series with bayesian prior distributions and synthetic data. *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, 178(4):963–975, 2015.
- [528] Petra Schneider and Gisbert Schneider. De novo design at the edge of chaos. *Journal of Medicinal Chemistry*, 59(9):4077–4086, 2016. PMID: 26881908.
- [529] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 815–823, 2015.
- [530] Marwin H. S. Segler, Thierry Kogej, Christian Tyrchan, and Mark P. Waller. Generating focussed molecule libraries for drug discovery with recurrent neural networks. *CoRR*, abs/1701.01329, 2017.
- [531] Marwin H. S. Segler, Thierry Kogej, Christian Tyrchan, and Mark P. Waller. Generating focused molecule libraries for drug discovery with recurrent neural networks. *ACS Central Science*, 4(1):120–131, 2018. PMID: 29392184.
- [532] Steven M. Seitz, Brian Curless, James Diebel, Daniel Scharstein, and Richard Szeliski. A comparison and evaluation of multi-view stereo reconstruction algorithms. In *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 1*, CVPR ’06, pages 519–528, Washington, DC, USA, 2006. IEEE Computer Society.
- [533] Gabriela Sejnova, Michael Tesar, and Michal Vavrecka. Compositional models for vqa: Can neural module networks really count? *Procedia Computer Science*, 145:481 – 487, 2018. Postproceedings of the 9th Annual International Conference on Biologically Inspired Cognitive Architectures, BICA 2018 (Ninth Annual Meeting of the BICA Society), held August 22–24, 2018 in Prague, Czech Republic.
- [534] Matan Sela, Pingmei Xu, Junfeng He, Vidhya Navalpakkam, and Dmitry Lagun. Gazegan - unpaired adversarial image generation for gaze estimation. *CoRR*, abs/1711.09767, 2017.

- [535] Rico Sennrich, Barry Haddow, and Alexandra Birch. Edinburgh neural machine translation systems for wmt 16. In *Proceedings of the First Conference on Machine Translation*, pages 371–376, Berlin, Germany, August 2016. Association for Computational Linguistics.
- [536] Sensefly datasets, 2019.
- [537] G. Sepulveda, J. C. Niebles, and A. Soto. A deep learning based behavioral approach to indoor autonomous navigation. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4646–4653, May 2018.
- [538] M. Tarek Shaban, Christoph Baur, Nassir Navab, and Shadi Albarqouni. Staingan: Stain style transfer for digital histological images. *CoRR*, abs/1804.01601, 2018.
- [539] Shital Shah, Debadatta Dey, Chris Lovett, and Ashish Kapoor. Airsim: High-fidelity visual and physical simulation for autonomous vehicles. In *Field and Service Robotics*, 2017.
- [540] Noor Shaker, Julian Togelius, and Mark J. Nelson. *Procedural Content Generation in Games*. Springer Publishing Company, Incorporated, 1st edition, 2016.
- [541] Ariel Shamir. A survey on mesh segmentation techniques. *Computer Graphics Forum*, 27(6):1539–1556, 2008.
- [542] Richard Shin, Neel Kant, Kavi Gupta, Chris Bender, Brandon Trabucco, Rishabh Singh, and Dawn Song. Synthetic datasets for neural program synthesis. In *International Conference on Learning Representations*, 2019.
- [543] S. Shoman, T. Mashita, A. Plopski, P. Ratsamee, Y. Uranishi, and H. Takemura. Illumination invariant camera localization using synthetic images. In *2018 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct)*, pages 143–144, Oct 2018.
- [544] Connor Shorten and Taghi M. Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of Big Data*, 6(1):60, Jul 2019.
- [545] Ashish Shrivastava, Tomas Pfister, Oncel Tuzel, Josh Susskind, Wenda Wang, and Russell Webb. Learning from simulated and unsupervised images through adversarial training. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2242–2251, 2017.
- [546] Xujie Si, Yuan Yang, Hanjun Dai, Mayur Naik, and Le Song. Learning a meta-solver for syntax-guided program synthesis. In *International Conference on Learning Representations*, 2019.
- [547] Aliaksandr Siarohin, Stéphane Lathuilière, Sergey Tulyakov, Elisa Ricci, and Nicu Sebe. Animating arbitrary objects via deep motion transfer. *ArXiv*, abs/1812.08861, 2018.

- [548] Nathan Silberman, Derek Hoiem, Pushmeet Kohli, and Rob Fergus. Indoor segmentation and support inference from rgbd images. In Andrew Fitzgibbon, Svetlana Lazebnik, Pietro Perona, Yoichi Sato, and Cordelia Schmid, editors, *Computer Vision – ECCV 2012*, pages 746–760, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.
- [549] J. C. Silveira Jacques Junior, S. R. Musse, and C. R. Jung. Crowd analysis using computer vision techniques. *IEEE Signal Processing Magazine*, 27(5):66–77, Sep. 2010.
- [550] Bharat Singh, Mahyar Najibi, and Larry S Davis. Sniper: Efficient multi-scale training. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*, pages 9310–9320. Curran Associates, Inc., 2018.
- [551] Dianna M Smith, Graham P Clarke, and Kirk Harland. Improving the synthetic data generation process in spatial microsimulation models. *Environment and Planning A: Economy and Space*, 41(5):1251–1268, 2009.
- [552] D. L. Smyth, F. G. Glavin, and M. G. Madden. Using a game engine to simulate critical incidents and data collection by autonomous drones. In *2018 IEEE Games, Entertainment, Media Conference (GEM)*, pages 1–9, Aug 2018.
- [553] David L. Smyth, James Fennell, Sai Abinesh, Nazli B. Karimi, Frank G. Glavin, Ihsan Ullah, Brett Drury, and Michael G. Madden. A virtual environment with multi-robot navigation, analytics, and decision support for critical incident investigation. *ArXiv*, abs/1806.04497, 2018.
- [554] Kihyuk Sohn, Sifei Liu, Guangyu Zhong, Xiang Yu, Ming-Hsuan Yang, and Manmohan Chandraker. Unsupervised domain adaptation for face recognition in unlabeled videos. *CoRR*, abs/1708.02191, 2017.
- [555] Pavel Solovev, Vladimir Aliev, Pavel Ostyakov, Gleb Sterkin, Elizaveta Logacheva, Stepan Troeshestov, Roman Suvorov, Anton Mashikhin, Oleg Khomenko, and Sergey I. Nikolenko. Learning state representations in complex systems with multimodal data. *CoRR*, abs/1811.11067, 2018.
- [556] S. Song, S. P. Lichtenberg, and J. Xiao. Sun rgb-d: A rgb-d scene understanding benchmark suite. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 567–576, June 2015.
- [557] Shuran Song, Fisher Yu, Andy Zeng, Angel X Chang, Manolis Savva, and Thomas Funkhouser. Semantic scene completion from a single depth image. *Proceedings of 30th IEEE Conference on Computer Vision and Pattern Recognition*, 2017.
- [558] Michael Stark, Michael Goesele, and Bernt Schiele. Back to the future: Learning shape models from 3d cad data. In *BMVC*, 2010.
- [559] Gregory J. Stein and Nicholas G. Roy. Genesis-rt: Generating synthetic images for training secondary real-world tasks. *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 7151–7158, 2018.

- [560] Oliver Struckmeier. Leagueai: Improving object detector performance and flexibility through automatically generated training data and domain randomization. *ArXiv*, abs/1905.13546, 2019.
- [561] Jürgen Sturm, Nikolas Engelhard, Felix Endres, Wolfram Burgard, and Daniel Cremers. A benchmark for the evaluation of rgbd slam systems. *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 573–580, 2012.
- [562] H. Su, S. Maji, E. Kalogerakis, and E. Learned-Miller. Multi-view convolutional neural networks for 3d shape recognition. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 945–953, Dec 2015.
- [563] H. Su, C. R. Qi, Y. Li, and L. J. Guibas. Render for cnn: Viewpoint estimation in images using cnns trained with rendered 3d model views. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 2686–2694, Dec 2015.
- [564] Zhihui Su, Ming Ye, Guohui Zhang, Lei Dai, and Jianda Sheng. Improvement multi-stage model for human pose estimation. *CoRR*, abs/1902.07837, 2019.
- [565] T. Sulkowski, P. Bugiel, and J. Izydorczyk. In search of the ultimate autonomous driving simulator. In *2018 International Conference on Signals and Electronic Systems (ICSES)*, pages 252–256, Sep. 2018.
- [566] The 2019 sumo workshop 360° indoor scene understanding and modeling, 2019.
- [567] Baochen Sun and Kate Saenko. From virtual to reality: Fast adaptation of virtual object detectors to real domains. In *Proceedings of the British Machine Vision Conference*. BMVA Press, 2014.
- [568] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang. Deep high-resolution representation learning for human pose estimation. *CoRR*, abs/1902.09212, 2019.
- [569] Gabriel Synnaeve, Nantas Nardelli, Alex Auvolat, Soumith Chintala, Timothée Lacroix, Zeming Lin, Florian Richoux, and Nicolas Usunier. Torchcraft: a library for machine learning research on real-time strategy games. *CoRR*, abs/1611.00625, 2016.
- [570] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2818–2826, June 2016.
- [571] C. Szegedy, Wei Liu, Yangqing Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–9, June 2015.

- [572] M. Sánchez, J. L. Martínez, J. Morales, A. Robles, and M. Morán. Automatic generation of labeled 3d point clouds of natural environments with gazebo. In *2019 IEEE International Conference on Mechatronics (ICM)*, volume 1, pages 161–166, March 2019.
- [573] L. Tai, G. Paolo, and M. Liu. Virtual-to-real deep reinforcement learning: Continuous control of mobile robots for mapless navigation. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 31–36, Sep. 2017.
- [574] Lei Tai and Ming Liu. Deep-learning in mobile robotics - from perception to control systems: A survey on why and why not. *CoRR*, abs/1612.07139, 2016.
- [575] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf. Deepface: Closing the gap to human-level performance in face verification. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1701–1708, June 2014.
- [576] Bowen Tan, Nayun Xu, and Bingyu Kong. Autonomous driving in reality with reinforcement learning and image translation. *CoRR*, abs/1801.05299, 2018.
- [577] Bowen Tan, Nayun Xu, and Bingyu Kong. Autonomous driving in reality with reinforcement learning and image translation. *ArXiv*, abs/1801.05299, 2018.
- [578] You-Bao Tang, Sooyoun Oh, Yu-Xing Tang, Jing Xiao, and Ronald M. Summers. Ct-realistic data augmentation using generative adversarial network for robust lymph node segmentation. In *SPIE Medical Imaging 2019: Computer-Aided Diagnosis*, volume 10950, 2019.
- [579] Youbao Tang, Yuxing Tang, Jing Xiao, and Ronald M. Summers. Xlsor: A robust and accurate lung segmentor on chest x-rays using criss-cross attention and customized radiorealistic abnormalities generation. *CoRR*, abs/1904.09229, 2019.
- [580] Zhentao Tang, Kun Shao, Yuanheng Zhu, Dong Mei Li, Dongbin Zhao, and Tingwen Huang. A review of computational intelligence for starcraft ai. *2018 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 1167–1173, 2018.
- [581] Camillo J. Taylor and Anthony Cowley. Parsing indoor scenes using rgb-d imagery. In *Robotics: Science and Systems*, 2012.
- [582] G. R. Taylor, A. J. Chosak, and P. C. Brewer. Ovvv: Using virtual worlds to design and evaluate surveillance systems. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, June 2007.
- [583] The SPRINT Research Group. A randomized trial of intensive versus standard blood-pressure control. *New England Journal of Medicine*, 373(22):2103–2116, 2015. PMID: 26551272.

- [584] J. Thieling and J. Roßmann. Highly-scalable and generalized sensor structures for efficient physically-based simulation of multi-modal sensor networks. In *2018 12th International Conference on Sensing Technology (ICST)*, pages 202–207, Dec 2018.
- [585] Yonglin Tian, Xuan Li, Kunfeng Wang, and Fei-Yue Wang. Training and testing object detectors with virtual images. *IEEE/CAA Journal of Automatica Sinica*, 5:539–546, 02 2018.
- [586] Yuandong Tian, Qucheng Gong, Wenling Shang, Yuxin Wu, and Larry Zitnick. ELF: an extensive, lightweight and flexible research platform for real-time strategy games. *CoRR*, abs/1707.01067, 2017.
- [587] J. Tobin, L. Biewald, R. Duan, M. Andrychowicz, A. Handa, V. Kumar, B. McGrew, A. Ray, J. Schneider, P. Welinder, W. Zaremba, and P. Abbeel. Domain randomization and generative models for robotic grasping. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3482–3489, Oct 2018.
- [588] 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)*, pages 23–30, Sep. 2017.
- [589] Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5026–5033, 2012.
- [590] Julian Togelius, Emil Kastbjerg, David Schedl, and Georgios N. Yannakakis. What is procedural content generation?: Mario on the borderline. In *Proceedings of the 2Nd International Workshop on Procedural Content Generation in Games*, PCGames ’11, pages 3:1–3:6, New York, NY, USA, 2011. ACM.
- [591] Tatiana Tommasi, Novi Patricia, Barbara Caputo, and Tinne Tuytelaars. *A Deeper Look at Dataset Bias*, pages 37–55. Springer International Publishing, Cham, 2017.
- [592] Anh Tuan Tran, Tal Hassner, Iacopo Masi, and Gérard G. Medioni. Regressing robust and discriminative 3d morphable models with a very deep neural network. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1493–1502, 2017.
- [593] L. Tran, X. Yin, and X. Liu. Disentangled representation learning gan for pose-invariant face recognition. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1283–1292, July 2017.
- [594] J. Tremblay, T. To, and S. Birchfield. Falling things: A synthetic dataset for 3d object detection and pose estimation. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 2119–21193, June 2018.

- [595] Jonathan Tremblay, Aayush Prakash, David Acuna, Mark Brophy, Varun Jampani, Cem Anil, Thang To, Eric Cameracci, Shaad Boochoon, and Stanley T. Birchfield. Training deep networks with synthetic data: Bridging the reality gap by domain randomization. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1082–10828, 2018.
- [596] Jonathan Tremblay, Thang To, Balakumar Sundaralingam, Yu Xiang, Dieter Fox, and Stanley T. Birchfield. Deep object pose estimation for semantic robotic grasping of household objects. In *CoRL*, 2018.
- [597] Aleksei Triastcyn and Boi Faltings. Generating Artificial Data for Private Deep Learning. *arXiv e-prints*, page arXiv:1803.03148, Mar 2018.
- [598] Aleksei Triastcyn and Boi Faltings. Generating differentially private datasets using gans. *CoRR*, abs/1803.03148, 2018.
- [599] Joshua C. Triyonoputro, Weiwei Wan, and Kensuke Harada. Quickly inserting pegs into uncertain holes using multi-view images and deep network trained on synthetic data. *ArXiv*, abs/1902.09157, 2019.
- [600] Matthew Trumble, Andrew Gilbert, Charles Malleson, Adrian Hilton, and John Collomosse. Total capture: 3d human pose estimation fusing video and inertial sensors. In *BMVC*, 09 2017.
- [601] Apostolia Tsirikoglou, Joel Kronander, Magnus Wrenninge, and Jonas Unger. Procedural modeling and physically based rendering for synthetic data generation in automotive applications. *CoRR*, abs/1710.06270, 2017.
- [602] Eric Tzeng, Coline Devin, Judy Hoffman, Chelsea Finn, Xingchao Peng, Sergey Levine, Kate Saenko, and Trevor Darrell. Towards adapting deep visuomotor representations from simulated to real environments. *CoRR*, abs/1511.07111, 2015.
- [603] Eric Tzeng, Judy Hoffman, Trevor Darrell, and Kate Saenko. Simultaneous deep transfer across domains and tasks. In *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, ICCV ’15, pages 4068–4076, Washington, DC, USA, 2015. IEEE Computer Society.
- [604] Eric Tzeng, Judy Hoffman, Ning Zhang, Kate Saenko, and Trevor Darrell. Deep domain confusion: Maximizing for domain invariance. *CoRR*, abs/1412.3474, 2014.
- [605] Jordan Ubbens, Mikolaj Cieslak, Przemyslaw Prusinkiewicz, and Ian Stavness. The use of plant models in deep learning: an application to leaf counting in rosette plants. *Plant Methods*, 14(1):6, Jan 2018.
- [606] Ihsan Ullah, Sai Abinesh, David L. Smyth, Nazli B. Karimi, Brett Drury, Frank G. Glavin, and Michael G. Madden. A virtual testbed for critical incident investigation with autonomous remote aerial vehicle surveying, artificial intelligence, and decision support. In Carlos Alzate, Anna Monreale, Haytham Assem, Albert Bifet, Teodora Sandra Buda, Bora Caglayan, Brett Drury, Eva García-Martín, Ricard Gavaldà, Irena Koprinska, Stefan Kramer, Niklas Lavesson, Michael Madden, Ian Molloy, Maria-Irina

Nicolae, and Mathieu Sinn, editors, *ECML PKDD 2018 Workshops*, pages 216–221, Cham, 2019. Springer International Publishing.

- [607] Dmitry Ulyanov, Vadim Lebedev, Andrea Vedaldi, and Victor Lempitsky. Texture networks: Feed-forward synthesis of textures and stylized images. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, ICML’16, pages 1349–1357. JMLR.org, 2016.
- [608] Tim Van den Bulcke, Koenraad Van Leemput, Bart Naudts, Piet van Remortel, Hongwu Ma, Alain Verschoren, Bart De Moor, and Kathleen Marchal. Syntren: a generator of synthetic gene expression data for design and analysis of structure learning algorithms. *BMC Bioinformatics*, 7(1):43, Jan 2006.
- [609] GÜl Varol, Javier Romero, Xavier Martin, Naureen Mahmood, Michael J. Black, Ivan Laptev, and Cordelia Schmid. Learning from synthetic humans. *CoRR*, abs/1701.01370, 2017.
- [610] Ashish Vaswani, Samy Bengio, Eugene Brevdo, Francois Chollet, Aidan N. Gomez, Stephan Gouws, Llion Jones, Lukasz Kaiser, Nal Kalchbrenner, Niki Parmar, Ryan Sepassi, Noam Shazeer, and Jakob Uszkoreit. Tensor2tensor for neural machine translation. *CoRR*, abs/1803.07416, 2018.
- [611] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc., 2017.
- [612] D. Vázquez, A. M. López, J. Marín, D. Ponsa, and D. Gerónimo. Virtual and real world adaptation for pedestrian detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(4):797–809, April 2014.
- [613] Eric Veach and Leonidas J. Guibas. Metropolis light transport. In *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH ’97, pages 65–76, New York, NY, USA, 1997. ACM Press/Addison-Wesley Publishing Co.
- [614] Ashwin J. Vijayakumar, Abhishek Mohta, Oleksandr Polozov, Dhruv Batra, Prateek Jain, and Sumit Gulwani. Neural-guided deductive search for real-time program synthesis from examples. *CoRR*, abs/1804.01186, 2018.
- [615] Oriol Vinyals, Timo Ewalds, Sergey Bartunov, Petko Georgiev, Alexander Sasha Vezhnevets, Michelle Yeo, Alireza Makhzani, Heinrich Küttler, John Agapiou, Julian Schrittwieser, John Quan, Stephen Gaffney, Stig Petersen, Karen Simonyan, Tom Schaul, Hado van Hasselt, David Silver, Timothy P. Lillicrap, Kevin Calderone, Paul Keet, Anthony Brunasso, David Lawrence, Anders Ekermo, Jacob Repp, and Rodney Tsing. Starcraft II: A new challenge for reinforcement learning. *CoRR*, abs/1708.04782, 2017.

- [616] Quan Van Vuong, Sharad Vikram, Hao Su, Sicun Gao, and Henrik I. Christensen. How to pick the domain randomization parameters for sim-to-real transfer of reinforcement learning policies? *ArXiv*, abs/1903.11774, 2019.
- [617] Jason Walonoski, Mark Kramer, Joseph Nichols, Andre Quina, Chris Moesel, Dylan Hall, Carlton Duffett, Kudakwashe Dube, Thomas Gallagher, and Scott McLachlan. Synthea: An approach, method, and software mechanism for generating synthetic patients and the synthetic electronic health care record. *Journal of the American Medical Informatics Association*, 25(3):230–238, 08 2017.
- [618] R. Wandarosanza, B. R. Trilaksono, and E. Hidayat. Hardware-in-the-loop simulation of uav hexacopter for chemical hazard monitoring mission. In *2016 6th International Conference on System Engineering and Technology (ICSET)*, pages 189–193, Oct 2016.
- [619] F. Wang, Y. Zhuang, H. Gu, and H. Hu. Automatic generation of synthetic lidar point clouds for 3-d data analysis. *IEEE Transactions on Instrumentation and Measurement*, 68(7):2671–2673, July 2019.
- [620] H. Wang, B. Kang, and D. Kim. Pfw: A face database in the wild for studying face identification and verification in uncontrolled environment. In *2013 2nd IAPR Asian Conference on Pattern Recognition*, pages 356–360, Nov 2013.
- [621] Hao Wang, Qilong Wang, Fan Yang, Weiqi Zhang, and Wangmeng Zuo. Data augmentation for object detection via progressive and selective instance-switching. *ArXiv*, abs/1906.00358, 2019.
- [622] K. Wang, R. Zhao, and Q. Ji. A hierarchical generative model for eye image synthesis and eye gaze estimation. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 440–448, June 2018.
- [623] Kai Wang, Fuyuan Shi, Wenqi Wang, Yibing Nan, and Shiguo Lian. Synthetic data generation and adaption for object detection in smart vending machines. *CoRR*, abs/1904.12294, 2019.
- [624] Kai Wang, Jianmin Zheng, Hock-Soon Seah, and Yuewen Ma. Triangular mesh deformation via edge-based graph. *Computer-Aided Design and Applications*, 9(3):345–359, 2012.
- [625] Qi Wang, Junyu Gao, Wei Lin, and Yuan Yuan. Learning from synthetic data for crowd counting in the wild. *CoRR*, abs/1903.03303, 2019.
- [626] Sen Wang, Daoyuan Jia, and Xinshuo Weng. Deep reinforcement learning for autonomous driving. *CoRR*, abs/1811.11329, 2018.
- [627] 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*, pages 8798–8807, June 2018.

- [628] T. Wang, D. J. Wu, A. Coates, and A. Y. Ng. End-to-end text recognition with convolutional neural networks. In *Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012)*, pages 3304–3308, Nov 2012.
- [629] William Yang Wang and Diyi Yang. That’s so annoying!!!: A lexical and frame-semantic embedding based data augmentation approach to automatic categorization of annoying behaviors using #petpeeve tweets. In *EMNLP*, 2015.
- [630] Xiang Wang, Kai Wang, and Shiguo Lian. A survey on face data augmentation. *CoRR*, abs/1904.11685, 2019.
- [631] Xinyi Wang, Hieu Pham, Zihang Dai, and Graham Neubig. Switchout: an efficient data augmentation algorithm for neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 856–861, Brussels, Belgium, October-November 2018. Association for Computational Linguistics.
- [632] Yu-Xiang Wang, Stephen E. Fienberg, and Alexander J. Smola. Privacy for free: Posterior sampling and stochastic gradient monte carlo. In *Proceedings of the 32Nd International Conference on International Conference on Machine Learning - Volume 37*, ICML’15, pages 2493–2502. JMLR.org, 2015.
- [633] Zhiyong Wang, Jinxiang Chai, and Shihong Xia. Combining recurrent neural networks and adversarial training for human motion synthesis and control. *CoRR*, abs/1806.08666, 2018.
- [634] Daniel Ward, Peyman Moghadam, and Nicolas Hudson. Deep leaf segmentation using synthetic data. In *BMVC*, 2018.
- [635] M. Warren and B. Upcroft. Robust scale initialization for long-range stereo visual odometry. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2115–2121, Nov 2013.
- [636] David Weininger. Smiles, a chemical language and information system. 1. introduction to methodology and encoding rules. *Journal of Chemical Information and Computer Sciences*, 28(1):31–36, 1988.
- [637] Janet Wiles and Jeffrey L. Elman. Learning to count without a counter: A case study of dynamics and activation landscapes in recurrent networks. In *ACCSS*, 1995.
- [638] Paul Wohlhart and Vincent Lepetit. Learning descriptors for object recognition and 3d pose estimation. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3109–3118, 2015.
- [639] Jelmer M. Wolterink, Anna M. Dinkla, Mark H. F. Savenije, Peter R. Seevinck, Cornelis A. T. van den Berg, and Ivana Isgum. Deep MR to CT synthesis using unpaired data. *CoRR*, abs/1708.01155, 2017.

- [640] E. Wood, T. Baltruaitis, X. Zhang, Y. Sugano, P. Robinson, and A. Bulling. Rendering of eyes for eye-shape registration and gaze estimation. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 3756–3764, Dec 2015.
- [641] Erroll Wood, Tadas Baltrušaitis, Louis-Philippe Morency, Peter Robinson, and Andreas Bulling. Learning an appearance-based gaze estimator from one million synthesised images. In *Proceedings of the Ninth Biennial ACM Symposium on Eye Tracking Research & Applications*, ETRA ’16, pages 131–138, New York, NY, USA, 2016. ACM.
- [642] Magnus Wrenninge and Jonas Unger. Synscapes: A photorealistic synthetic dataset for street scene parsing. *CoRR*, abs/1810.08705, 2018.
- [643] Bichen Wu, Alvin Wan, Xiangyu Yue, and Kurt Keutzer. Squeezeseg: Convolutional neural nets with recurrent CRF for real-time road-object segmentation from 3d lidar point cloud. *CoRR*, abs/1710.07368, 2017.
- [644] Bichen Wu, Xuanyu Zhou, Sicheng Zhao, Xiangyu Yue, and Kurt Keutzer. Squeezesegv2: Improved model structure and unsupervised domain adaptation for road-object segmentation from a lidar point cloud. *CoRR*, abs/1809.08495, 2018.
- [645] Huikai Wu, Junge Zhang, and Kaiqi Huang. Msc: A dataset for macro-management in starcraft ii. *ArXiv*, abs/1710.03131, 2017.
- [646] Xiang Wu, Ran He, Zhenan Sun, and Tieniu Tan. A light cnn for deep face representation with noisy labels. *IEEE Transactions on Information Forensics and Security*, 13:2884–2896, 2018.
- [647] Yi Wu, Yuxin Wu, Georgia Gkioxari, and Yuandong Tian. Building generalizable agents with a realistic and rich 3d environment. *CoRR*, abs/1801.02209, 2018.
- [648] Yuxin Wu and Yuandong Tian. Training agent for first-person shooter game with actor-critic curriculum learning. In *ICLR*, 2017.
- [649] Zhirong Wu, Shuran Song, Aditya Khosla, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets for 2.5d object recognition and next-best-view prediction. *ArXiv*, abs/1406.5670, 2014.
- [650] Markus Wulfmeier, Alex Bewley, and Ingmar Posner. Addressing appearance change in outdoor robotics with adversarial domain adaptation. *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1551–1558, 2017.
- [651] Bernhard Wymann, Eric Espié, Christophe Guionneau, Christos Dimitsakakis, Rémi Coulom, and Andrew Sumner. TORCS, The Open Racing Car Simulator. <http://www.torcs.org>, 2014.
- [652] Fei Xia, Amir Roshan Zamir, Zhi-Yang He, Alexander Sax, Jitendra Malik, and Silvio Savarese. Gibson env: Real-world perception for embodied agents. *CoRR*, abs/1808.10654, 2018.

- [653] Y. Xiang, A. Alahi, and S. Savarese. Learning to track: Online multi-object tracking by decision making. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 4705–4713, Dec 2015.
- [654] J. Xiao, A. Owens, and A. Torralba. Sun3d: A database of big spaces reconstructed using sfm and object labels. In *2013 IEEE International Conference on Computer Vision*, pages 1625–1632, Dec 2013.
- [655] Shengtao Xiao, Jiashi Feng, Junliang Xing, Hanjiang Lai, Shuicheng Yan, and Ashraf Kassim. Robust facial landmark detection via recurrent attentive-refinement networks. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *Computer Vision – ECCV 2016*, pages 57–72, Cham, 2016. Springer International Publishing.
- [656] Xiaoxia Hu, Xuefeng Liu, Zhenming He, and Jiahua Zhang. Batch modeling of 3d city based on esri cityengine. In *IET International Conference on Smart and Sustainable City 2013 (ICSSC 2013)*, pages 69–73, Aug 2013.
- [657] Liyang Xie, Kaixiang Lin, Shu Wang, Fei Wang, and Jiayu Zhou. Differentially private generative adversarial network. *CoRR*, abs/1802.06739, 2018.
- [658] Xu Xie, Hangxin Liu, Zhenliang Zhang, Yuxing Qiu, Feng Gao, Siyuan Qi, Yixin Zhu, and Song-Chun Zhu. Vrgym: A virtual testbed for physical and interactive ai. In *Proceedings of the ACM Turing Celebration Conference - China*, ACM TURC ’19, pages 100:1–100:6, New York, NY, USA, 2019. ACM.
- [659] Ziang Xie, Sida I. Wang, Jiwei Li, Daniel Lévy, Aiming Nie, Daniel Jurafsky, and Andrew Y. Ng. Data noising as smoothing in neural network language models. *CoRR*, abs/1703.02573, 2017.
- [660] Hui Y. Xiong, Babak Alipanahi, Leo J. Lee, Hannes Bretschneider, Daniele Merico, Ryan K. C. Yuen, Yimin Hua, Serge Gueroussov, Hamed S. Najafabadi, Timothy R. Hughes, Quaid Morris, Yoseph Barash, Adrian R. Krainer, Nebojsa Jojic, Stephen W. Scherer, Benjamin J. Blencowe, and Brendan J. Frey. The human splicing code reveals new insights into the genetic determinants of disease. *Science*, 347(6218), 2015.
- [661] Xi Xiong, Jianqiang Wang, Fang Zhang, and Keqiang Li. Combining deep reinforcement learning and safety based control for autonomous driving. *CoRR*, abs/1612.00147, 2016.
- [662] Dan Xu, Wanli Ouyang, Xiaogang Wang, and Nicu Sebe. Pad-net: Multi-tasks guided prediction-and-distillation network for simultaneous depth estimation and scene parsing. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 675–684, 2018.
- [663] Gao Yan Xu, Qixing Zhang, Dongcai Liu, Gaohua Lin, Jinjun Wang, and Yongming Zhang. Adversarial adaptation from synthesis to reality in fast detector for smoke detection. *IEEE Access*, 7:29471–29483, 2019.

- [664] J. Xu, D. Vázquez, S. Ramos, A. M. López, and D. Ponsa. Adapting a pedestrian detector by boosting lda exemplar classifiers. In *2013 IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 688–693, June 2013.
- [665] Xiang Xu, Pengfei Dou, Ha A. Le, and Ioannis A. Kakadiaris. When 3d-aided 2d face recognition meets deep learning: An extended UR2D for pose-invariant face recognition. *CoRR*, abs/1709.06532, 2017.
- [666] Pingkun Yan, Sheng Xu, Ardeshir R. Rastinehad, and Bradford J. Wood. Adversarial image registration with application for MR and TRUS image fusion. *CoRR*, abs/1804.11024, 2018.
- [667] Guangyu Robert Yang, Igor Ganichev, Xiao-Jing Wang, Jonathon Shlens, and David Sussillo. A dataset and architecture for visual reasoning with a working memory. In *ECCV*, 2018.
- [668] Luona Yang, Xiaodan Liang, and Eric P. Xing. Unsupervised real-to-virtual domain unification for end-to-end highway driving. *CoRR*, abs/1801.03458, 2018.
- [669] Wei Yang, Shuang Li, Wanli Ouyang, Hongsheng Li, and Xiaogang Wang. Learning feature pyramids for human pose estimation. *CoRR*, abs/1708.01101, 2017.
- [670] Li Yi, Leonidas Guibas, Aaron Hertzmann, Vladimir G. Kim, Hao Su, and Ersin Yumer. Learning hierarchical shape segmentation and labeling from online repositories. *ACM Trans. Graph.*, 36(4):70:1–70:12, July 2017.
- [671] Li Yi, Vladimir G. Kim, Duygu Ceylan, I-Chao Shen, Mengyan Yan, Hao Su, Cewu Lu, Qixing Huang, Alla Sheffer, and Leonidas Guibas. A scalable active framework for region annotation in 3d shape collections. *ACM Trans. Graph.*, 35(6):210:1–210:12, 2016.
- [672] Li Yi, Lin Shao, Manolis Savva, Haibin Huang, Yang Zhou, Qirui Wang, Benjamin Graham, Martin Engelcke, Roman Klokov, Victor S. Lempitsky, Yuan Gan, Pengyu Wang, Kun Liu, Fenggen Yu, Panpan Shui, Bingyang Hu, Yan Zhang, Yangyan Li, Rui Bu, Mingchao Sun, Wei Wu, Minki Jeong, Jaehoon Choi, Changick Kim, Angom Geetchandra, Narasimha Murthy, Bhargava Ramu, Bharadwaj Manda, M. Ramanathan, Gautam Kumar, P. Preetham, Siddharth Srivastava, Swati Bhugra, Brejesh Lall, Christian Häne, Shubham Tulsiani, Jitendra Malik, Jared Lafer, Ramsey Jones, Siyuan Li, Jie Lu, Shi Jin, Jingyi Yu, Qixing Huang, Evangelos Kalogerakis, Silvio Savarese, Pat Hanrahan, Thomas A. Funkhouser, Hao Su, and Leonidas J. Guibas. Large-scale 3d shape reconstruction and segmentation from shapenet core55. *CoRR*, abs/1710.06104, 2017.
- [673] Xin Yi, Ekta Walia, and Paul Babyn. Generative adversarial network in medical imaging: A review. *CoRR*, abs/1809.07294, 2018.
- [674] Senthil Yogamani, Ciarán Hughes, Jonathan Horgan, Ganesh Sistu, Padraig Varley, Derek O’Dea, Michal Uricár, Stefan Milz, Martin Simon,

Karl Amende, Christian Witt, Hazem Rashed, Sumanth Chennupati, Sanjaya Nayak, Saquib Mansoor, Xavier Perrotin, and Patrick Perez. Woodscape: A multi-task, multi-camera fisheye dataset for autonomous driving. *CoRR*, abs/1905.01489, 2019.

- [675] Jinsung Yoon, James Jordon, and Mihaela van der Schaar. PATE-GAN: Generating synthetic data with differential privacy guarantees. In *International Conference on Learning Representations*, 2019.
- [676] Yurong You, Xinlei Pan, Ziyan Wang, and Cewu Lu. Virtual to real reinforcement learning for autonomous driving. *CoRR*, abs/1704.03952, 2017.
- [677] B. Yu, L. Zhou, L. Wang, J. Fripp, and P. Bourgeat. 3d cgan based cross-modality mr image synthesis for brain tumor segmentation. In *2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018)*, pages 626–630, April 2018.
- [678] Lantao Yu, Weinan Zhang, Jun Wang, and Yong Yu. Seqgan: Sequence generative adversarial nets with policy gradient. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, AAAI’17, pages 2852–2858. AAAI Press, 2017.
- [679] Xiangyu Yue, Bichen Wu, Sanjit A. Seshia, Kurt Keutzer, and Alberto L. Sangiovanni-Vincentelli. A lidar point cloud generator: from a virtual world to autonomous driving. In *ICMR*, 2018.
- [680] Xiangyu Yue, Yang Zhang, Sicheng Zhao, Alberto Sangiovanni-Vincentelli, Kurt Keutzer, and Boqing Gong. Domain randomization and pyramid consistency: Simulation-to-real generalization without accessing target domain data. *ArXiv*, abs/1909.00889, 2019.
- [681] B. Yilmaz, M. F. Amasyali, M. Balciar, E. Uslu, and S. Yavuz. Impact of artificial dataset enlargement on performance of deformable part models. In *2016 24th Signal Processing and Communication Application Conference (SIU)*, pages 193–196, May 2016.
- [682] Sergey Zakharov, Wadim Kehl, and Slobodan Ilic. Deceptionnet: Network-driven domain randomization. *CoRR*, abs/1904.02750, 2019.
- [683] Wojciech Zaremba and Ilya Sutskever. Learning to execute. *CoRR*, abs/1410.4615, 2014.
- [684] Maksym Zavershynskyi, Alexander Skidanov, and Illia Polosukhin. NAPS: natural program synthesis dataset. *CoRR*, abs/1807.03168, 2018.
- [685] Rowan Zellers, Ari Holtzman, Hannah Rashkin, Yonatan Bisk, Ali Farhadi, Franziska Roesner, and Yejin Choi. Defending against neural fake news. *CoRR*, abs/1905.12616, 2019.
- [686] Boyu Zhang, Wei Zhao, JiaFeng Liu, Rui Wu, and XiangLong Tang. Character recognition in natural scene images using local description. In Yan-ning Zhang, Zhi-Hua Zhou, Changshui Zhang, and Ying Li, editors, *Intelligent Science and Intelligent Data Engineering*, pages 193–200, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.

- [687] Hong-Bo Zhang, Qing Lei, Bi-Neng Zhong, Ji-Xiang Du, and JiaLin Peng. A survey on human pose estimation. *Intelligent Automation & Soft Computing*, 22(3):483–489, 2016.
- [688] Hong-Bo Zhang, Yi-Xiang Zhang, Bineng Zhong, Qing Lei, Lijie Yang, Ji-Xiang Du, and Duan-Sheng Chen. A comprehensive survey of vision-based human action recognition methods. *Sensors*, 19:1005, 02 2019.
- [689] Jing Zhang, Wanqing Li, Philip Ogunbona, and Dong Xu. Recent advances in transfer learning for cross-dataset visual recognition: A problem-oriented perspective. *ACM Comput. Surv.*, 52(1):7:1–7:38, February 2019.
- [690] Jingwei Zhang, Lei Tai, Yufeng Xiong, Ming Liu, Joschka Boedecker, and Wolfram Burgard. VR goggles for robots: Real-to-sim domain adaptation for visual control. *CoRR*, abs/1802.00265, 2018.
- [691] Jun Zhang, Graham Cormode, Cecilia M. Procopiuc, Divesh Srivastava, and Xiaokui Xiao. Privbayes: Private data release via bayesian networks. *ACM Trans. Database Syst.*, 42(4):25:1–25:41, October 2017.
- [692] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao. Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE Signal Processing Letters*, 23(10):1499–1503, Oct 2016.
- [693] Song-Hai Zhang, S. J. Zhang, Yuan Liang, and Peter Hall. A survey of 3d indoor scene synthesis. *Journal of Computer Science and Technology*, 34:594–608, 2019.
- [694] X. Zhang, Y. Sugano, M. Fritz, and A. Bulling. Appearance-based gaze estimation in the wild. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4511–4520, June 2015.
- [695] X. Zhang, J. Zhao, and Y. LeCun. Character-level convolutional networks for text classification. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 649–657. Curran Associates, Inc., 2015.
- [696] Xinyang Zhang, Shouling Ji, and Ting Wang. Differentially private releasing via deep generative model. *CoRR*, abs/1801.01594, 2018.
- [697] Y. Zhang, D. Zhou, S. Chen, S. Gao, and Y. Ma. Single-image crowd counting via multi-column convolutional neural network. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 589–597, June 2016.
- [698] Yang Zhang, Philip David, Hassan Foroosh, and Boqing Gong. A curriculum domain adaptation approach to the semantic segmentation of urban scenes. *CoRR*, abs/1812.09953, 2018.
- [699] Yang Zhang, Philip David, and Boqing Gong. Curriculum domain adaptation for semantic segmentation of urban scenes. *CoRR*, abs/1707.09465, 2017.

- [700] Yi Zhang, Weichao Qiu, Qing Chen, Xiaolin C. Hu, and Alan Loddon Yuille. Unrealstereo: A synthetic dataset for analyzing stereo vision. *ArXiv*, abs/1612.04647, 2016.
- [701] Yinda Zhang, Shuran Song, Ersin Yumer, Manolis Savva, Joon-Young Lee, Hailin Jin, and Thomas A. Funkhouser. Physically-based rendering for indoor scene understanding using convolutional neural networks. *CoRR*, abs/1612.07429, 2016.
- [702] Yizhe Zhang, Zhe Gan, Kai Fan, Zhi Chen, Ricardo Henao, Dinghan Shen, and Lawrence Carin. Adversarial Feature Matching for Text Generation. *arXiv e-prints*, page arXiv:1706.03850, Jun 2017.
- [703] Yizhe Zhang, Lin Yang, Jianxu Chen, Maridel Fredericksen, David P. Hughes, and Danny Z. Chen. Deep adversarial networks for biomedical image segmentation utilizing unannotated images. In Maxime Descoteaux, Lena Maier-Hein, Alfred Franz, Pierre Jannin, D. Louis Collins, and Simon Duchesne, editors, *Medical Image Computing and Computer Assisted Intervention – MICCAI 2017*, pages 408–416, Cham, 2017. Springer International Publishing.
- [704] Zizhao Zhang, Lin Yang, and Yefeng Zheng. Translating and segmenting multimodal medical volumes with cycle- and shape-consistency generative adversarial network. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9242–9251, 2018.
- [705] Han Zhao, Shanghang Zhang, Guanhong Wu, João P. Costeira, José M. F. Moura, and Geoffrey J. Gordon. Adversarial multiple source domain adaptation. In *Proceedings of the 32Nd International Conference on Neural Information Processing Systems*, NIPS’18, pages 8568–8579, USA, 2018. Curran Associates Inc.
- [706] He Zhao, Huiqi Li, and Li Cheng. Synthesizing filamentary structured images with gans. *CoRR*, abs/1706.02185, 2017.
- [707] Jian Zhao, Lin Xiong, Yu Cheng, Yi Cheng, Jianshu Li, Li Zhou, Yan Xu, Jayashree Karlekar, Sugiri Pranata, Shengmei Shen, Junliang Xing, Shuicheng Yan, and Jiashi Feng. 3d-aided deep pose-invariant face recognition. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, pages 1184–1190. International Joint Conferences on Artificial Intelligence Organization, 7 2018.
- [708] Jian Zhao, Lin Xiong, Jianshu Li, Junliang Xing, Shuicheng Yan, and Jiashi Feng. 3d-aided dual-agent gans for unconstrained face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PP:1–1, 07 2018.
- [709] Junbo Jake Zhao, Yoon Kim, Kelly Zhang, Alexander M. Rush, and Yann LeCun. Adversarially regularized autoencoders for generating discrete structures. *CoRR*, abs/1706.04223, 2017.
- [710] Chuanxia Zheng, Tat-Jen Cham, and Jianfei Cai. T2net: Synthetic-to-realistic translation for solving single-image depth estimation tasks. In

Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, *Computer Vision – ECCV 2018*, pages 798–814, Cham, 2018. Springer International Publishing.

- [711] Ningyuan Zheng, Yifan Jiang, and Ding jiang Huang. Strokenet: A neural painting environment. In *ICLR*, 2019.
- [712] Fangzheng Zhou. A survey of mainstream indoor positioning systems. *Journal of Physics: Conference Series*, 910:012069, 10 2017.
- [713] Zhou Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, April 2004.
- [714] 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)*, pages 2242–2251, Oct 2017.
- [715] Song-Chun Zhu and David Mumford. A stochastic grammar of images. *Found. Trends. Comput. Graph. Vis.*, 2(4):259–362, January 2006.
- [716] X. Zhu, J. Yan, D. Yi, Z. Lei, and S. Z. Li. Discriminative 3d morphable model fitting. In *2015 11th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG)*, volume 1, pages 1–8, May 2015.
- [717] Yezi Zhu, Marc Aoun, Marcel Krijn, and Joaquin Vanschoren. Data augmentation using conditional generative adversarial networks for leaf counting in arabidopsis plants. In *BMVC*, 2018.
- [718] Barret Zoph and Quoc V. Le. Neural architecture search with reinforcement learning. *CoRR*, abs/1611.01578, 2016.
- [719] Barret Zoph, Vijay Vasudevan, Jonathon Shlens, and Quoc V. Le. Learning transferable architectures for scalable image recognition. *CoRR*, abs/1707.07012, 2017.