EBERHARD KARLS
UNIVERSITÄT
TÜBINGEN

Mathematisch-Naturwissenschaftliche Fakultät

Lernbasierte Computer Vision

Bachelorarbeit

A Comparison of Synthetic-to-Real Domain Adaptation Techniques

Eberhard Karls Universität Tübingen
Mathematisch-Naturwissenschaftliche Fakultät
Wilhelm-Schickard-Institut für Informatik
Lernbasierte Computer Vision
Peter Trost, peter.trost@student.uni-tuebingen.de, 2019

Bearbeitungszeitraum: 24.05.2019-23.09.2019

Betreuer/Gutachter: Prof. Dr. Andreas Geiger, Universität Tübingen

Selbstständigkeitserklärung

Hiermit versichere ich, dass ich die vorliegende Bachelorarbeit selbständig und nur mit den angegebenen Hilfsmitteln angefertigt habe und dass alle Stellen, die dem Wortlaut oder dem Sinne nach anderen Werken entnommen sind, durch Angaben von Quellen als Entlehnung kenntlich gemacht worden sind. Diese Bachelorarbeit wurde in gleicher oder ähnlicher Form in keinem anderen Studiengang als Prüfungsleistung vorgelegt.

Peter Trost (Matrikelnummer 4039682), June 19, 2019

Abstract

Template

Acknowledgments

If you have someone to Acknowledge ;)

Contents

1. Introduction

What is this all about?

Cite like this: [GPAM+14]

1.1. Problem Statement

There are many techniques for adapting images from a synthetic to a realistic domain. Each having different approaches and using different methods. In this work I am comparing three current approaches on synthetic-to-real domain adaptation and try to show their strengths and weaknesses.

2. Background Knowledge

2.1. Domain Adaptation

Domain Adaptation is the task of transfering a model that is working well on a source data distribution to a related target data distribution. In this work we will focus on the adaptation from synthetic to real images. Synthetic meaning that the image was rendered from a virtual scene and real meaning an image taken from a real-world scene. **TODO: insert example images**

2.2. CNNs and GANs

Convolutional Neural Networks are Deep Neural Networks consisting of convolution layers, that extract features from input data, pooling layers, that **TODO: ??** and a fully connected network for classification. **TODO: add more description and example images**

General Adversarial Networks implement a two-player-game: A Discriminator learns from a given data distribution what's "real". The Generator generates data. The goal of the generator is to fool the discriminator into believing the generated data is "real". The discriminator will label anything as "fake" that doesn't resemble the learned "real" data distribution. This way GANs can learn to generate realistic looking images of faces, translate image art styles from one to another and improve semantic segmentation.

3. Related Work

3.1. Domain Adaptation for Structured Output via Discriminative Patch Representation

see [TSSC19]

3.1.1. Abstract

- labeling data is expensive
- therefore propose domain adaptation method to adapt labeled source data to unlabeled target domain (e.g. GTA5 (playing for data) to city-scapes)
- learn discriminative feature representations of patches based on label histograms in the source domain, through construction of clustered space
- then use adversarial learning sscheme to push feature representations in target patches to the closer distributions in source ones
- can integrate a global alignment process with this patch-level alignment and achieve state-of-the-art performance on semantic segmentation
- extensive ablation studies on numerous benchmark datasets with various settings (e.g. synth-to-real, cross-city)

3.1.2. Introduction

- pixel-level annotation of ground truth expensive. e.g. road-scene iamges of different cities may have various appearance distributions, differences over time and weather
- existing state-of-the-art methods use feature-level or output space adaptation, exploiut global distribution alignment, such as spatial layout, but these might differ significantly between two domains due to differences in camera poses or field of view
- authours instead match patches that are more likely to be shared across domains regardless of where they are located

- consider label histograms as a factor (Kulkarni et al., 2015; Odena et al., 2017) and learn discriminative representations for patches to relax high-variation problem among them
- use this to better align patches between source and target domains
- utilize two adversarial modules to align global/patch-level distributions
- global one based on output space adaptation (Tsai et al. 2018)
- take source domain labels and extract label histogram as a patch-level representation
- then apply K-means clustering to group extracted patch representations into K clusters **TODO: read this part again for better understanding (page 2)**

3.1.3. domain adaptation for structured output

- given source and target images $I_s, I_t \in \mathbb{R}^{H\times W\times 3}$ and source labels Y_s, the goal is to align predicted output distribution O_t of target data with source distribution O_s
- use loss function for supervised learning on source data to predict the structured output, adversarial loss is adopted to align the global distribution
- further incorporate classification loss in a clustered space to learn patch-level discriminative representations F_s from source output distribution O_s. For target data another adversarial loss is used to align patch-level distributions between F_s and F_t, where the goal is to push F_t to be closer to distributon of F_s.
- objective function :

$$\mathcal{L}_{\text{total}}(I_s, I_t, Y_s, \Gamma(Y_s)) = \mathcal{L}_s + \lambda_d \mathcal{L}_d + \lambda^g_{\text{adv}} \mathcal{L}^g_{\text{adv}} + \lambda^l_{\text{adv}} \mathcal{L}^l_{\text{adv}} \tag{3.1}$$

  where $\mathcal{L}_s$ and $\mathcal{L}_d$ are supervised loss function for learning structured prediction and discriminative representation on source data. Γ denotes clustering process on ground truth label distribution. $\mathcal{L}^g_{\text{adv}}, \mathcal{L}^l_{\text{adv}}$ denote global and patch-level adversarial loss. λ's are weights for the different loss function
- $\mathcal{L}_s$ can be optimized by fully-convolutional network $\mathbf{G}$ that predicts the structured output with the loss summed over the spatial map indexed with h, w and number of categories C:

$$\mathcal{L}_s(I_s, Y_s; \mathbf{G}) = -\sum_{h,w} \sum_{c\in C} Y_s^{(h,w,c)} \log(O_s^{(h,w,c)}) \tag{3.2}$$

  where $O_s = \mathbf{G}(I_s) \in (0,1)$ is the predicted output distribution through softmax function and is up-sampled to the size of the input image.

- with discriminator $\mathbf{D}_g$:

$$\mathcal{L}_{\text{adv}}^{g}(I_s, I_t; \mathbf{G}, \mathbf{D}_g) = \sum_{h,w} \mathbb{E}[\log \mathbf{D}_g(O_s)^{(h,w,1)}] + \mathbb{E}[\log(1 - \mathbf{D}_g(O_t)^{(h,w,1)})] \tag{3.3}$$

- optimize following min-max problem with inputs dropped for simplicity:

$$\min_{\mathbf{G}} \max_{\mathbf{D}_g} \mathcal{L}_s(\mathbf{G}) + \lambda_{\text{adv}}^{g} \mathcal{L}_{\text{adv}}^{g}(\mathbf{G}, \mathbf{D}_g) \tag{3.4}$$

- label histograms for patches: first randomly sample patches from source images, using a 2-by-2 grid on patches to extract spatial label histograms, and concatenate them into a vector, each histogram is a $2 \cdot 2 \cdot C$ dimensional vector. Second apply K-means clustering on these histograms, whereby the label for any patch can be assigned as the cluster center with the closest distance on the histogram
- add classification module $\mathbf{H}$ after the predicted output O_s, to simulate the procedure of constructin the label histogram and learn a discriminative representation
  learned representation: $F_s = \mathbf{H}(\mathbf{G}(I_s)) \in (0,1)^{U \times V \times K}$ (softmax function, K is number of clusters)
- learning process to construct clustered space formulated as cross-entropy loss:

$$\mathcal{L}_d(I_s, \Gamma(Y_s); \mathbf{G}, \mathbf{H}) = -\sum_{u,v} \sum_{k \in K} \Gamma(Y_s)^{(u,v,k)} \log(F_s^{(u,v,k)}) \tag{3.5}$$

- goal is now to align patches regardless of where they are located in the image (without spatial and neighborhood support)
- reshape F by concatenating the K-dimensional vectors along the spatial map, results in $U \cdot V$ independent data points
- this reshaped data is denoted as $\hat{F}$, adversarial objective:

$$\mathcal{L}_{\text{adv}}^{l}(I_s, I_t; \mathbf{G}, \mathbf{H}, \mathbf{D}_l) = \sum_{u,v} \mathbb{E}[\log \mathbf{D}_l(\hat{F}_s)^{(u,v,1)}] + \mathbb{E}[\log(1 - \mathbf{D}_l(\hat{F}_t)^{(u,v,1)})] \tag{3.6}$$

  where $\mathbf{D}_l$ is the discriminator to classify whether the feature representation $\hat{F}$ is from source or target domain
- integrate (3.5) and (3.6) into min-max problem in 3.4:

$$\min_{\mathbf{G},\mathbf{H}} \max_{\mathbf{D}_g, \mathbf{D}_l} \mathcal{L}_s(\mathbf{G}) + \lambda_d \mathcal{L}_d(\mathbf{G}, \mathbf{H}) + \lambda_{\text{adv}}^{g}, \mathcal{L}_{\text{adv}}^{g}(\mathbf{G}, \mathbf{D}_g) + \lambda_{\text{adv}}^{l} \mathcal{L}_{\text{adv}}^{l}(\mathbf{G}, \mathbf{H}, \mathbf{D}_l) \tag{3.7}$$

3.2. Effective Use of Synthetic Data for Urban Scene Semantic Segmentation

see [SAS+18]

3.3. Exemplar Guided Unsupervised Image-to-Image Translation with Semantic Consistency

see [MJG+18]

3.4. Exploiting Semantics in Adversarial Training for Image-Level Domain Adaptation

see [RTdS18]

3.5. FCNs in the Wild: Pixel-level Adversarial and Constraint-based Adaptation

see [HWYD16]

3.6. From Virtual to Real World Visual Perception using Domain Adaptation - The DPM Example

see [LXG+16]

3.7. Semantic-aware Grad-GAN for Virtual-to-Real Urban Scene Adaption

see [LLJX18]

3.8. Syn2Real: A New Benchmark for Synthetic-to-Real Visual Domain Adaptation

see [PUS+18]

3.9. VisDA: The Visual Domain Adaptation Challenge

see [PUK+17]

4. Datasets

4.1. Playing for Data: Ground Truth from Computer Games

see [RVRK16]

- contains 24966 images taken from a street view of Grand Theft Auto V (GTA5) in 1914×1052 pixels
- two orders of magnitude larger thatn CamVid and three orders of magnitude larger than semantic segmentation created for KITTI dataset
- highly realistic with moving cars, objects, pedestrians, bikes, day/night, changing lighting and weather conditions
- includes labels for these images
- labeling process took 49 hours, 3 magnitudes faster than comparable real datasets (normal annotation would've approximately taken 12 person-years)
- annotation took 7 seconds per image on average (514 times faster tahn for CamVid, 771 times faster than for Cityscapes)
- achieved by detouring: injecting a wrapper between game and graphics hardware to log functioncalls and reconstruct 3D scene
- objects in that scene can be assigned an object ID
- labeling an object in one image will then propagate that label to that object in every image that contains it

TODO: add some image samples TODO: add diversity of collected data graph

5. Domain Adaptation Techniques

5.1. Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks

see [ZPIE17]

- image-to-image translation: extracting characteristics of an image and translating it to another style while preserving the characteristics (rgb to greyscale, painting to photo,..)
- special in this approach: no paired images necessary (datasets with paired images are far more expensive)
- create mapping $G : X \rightarrow Y$ from source domain X to target domain Y
- The Generator has to trick the discriminator into believing $G(x), x \in X$ is actually a real sample y from the target domain Y (matches distribution $p_{\text{data}}(y)$)
- problem of mode collapse: any inputimage will be translated to the same output image
- add cycle-consistency constraint: create mapping $F : Y \rightarrow X$ and add contraint $F(G(x)) \overset{!}{\approx} x$
- objective for mapping/generator G and discriminator D_Y:
  $\mathcal{L}_{\text{GAN}}(G, D_Y, X, Y) = \mathbb{E}_{y \sim p_{\text{data}}(y)}[\log D_Y(y)] + \mathbb{E}_{x \sim p_{\text{data}}(x)}[1 - \log D_Y(G(x))]$
- analogous for mapping/generator F and discriminator D_X
- generators try to minimize the objective, discriminators try to maximize it
- cycle consistency loss:
  $\mathcal{L}_{\text{cyc}}(G, F) = \mathbb{E}_{x \sim p_{\text{data}}(x)}[||F(G(x)) - x||_1] + \mathbb{E}_{y \sim p_{\text{data}}(y)}[||G(F(y)) - y||_1]$
- full objective:
  $\mathcal{L}(G, F, D_X, D_Y) = \mathcal{L}_{\text{GAN}}(G, D_Y, X, Y) + \mathcal{L}_{\text{GAN}}(F, D_X, Y, X) + \lambda \mathcal{L}_{\text{cyc}}(G, F)$
- solve: $G^*, F^* = \arg\min_{G,F} \max_{D_X, D_Y} \mathcal{L}(G, F, D_X, D_Y)$

5.1.1. Training Details

- for $\mathcal{L}_{\text{GAN}}$ replaced negative log-likelihood objective by least-squares loss → more stable during training and generates higher quality results
- for GAN loss $\mathcal{L}_{\text{GAN}}(G, D, X, Y)$ they train
  G to minimize $\mathbb{E}_{x \sim p_{\text{data}}(x)}[(D(G(x)) - 1)^2]$
  and D to minimize $\mathbb{E}_{y \sim p_{\text{data}}(y)}[(D(y) - 1)^2] + \mathbb{E}_{x \sim p_{\text{data}}(x)}[D(G(x))^2]$
- reduce model oscillation **TODO: add Shrivasta et al.**s method update discriminator using history f generated images instead of the latest ones generated. Use a buffer of 50 images
- set $\lambda = 10$ in **TODO: Equation 3**, Adam solver, batchsize 1, trained from scratch with learning rate 0.0002 for the first 100 epochs then decay linearly to 0 in the following 100.

5.2. CyCADA: Cycle Consistent Adversarial Domain Adaptation

see [HTP+17]

5.2.1. Introduction

- synthetic datasets cheaper and more accurate in classification than real ones
- per-pixel label accuracy drops from 93%(real) to 54%(synthetic)
- while translating from synth to real semantic information might be lost (e.g translating line-drawing of a cat to a picture of a dog)
- CyCADA uses cycle consistency and semantic losses
- apply model to digit recognition and semantic segmentation of urban scenes across domains.
- improves per-pixel accuracy from 54% to 82% on synth-to-real. (**TODO: compared to what?**)
- shows that domain adaptation benefits greatily from cycle-consistent pixel transformations
- adaptation at both pixel and representation level can offer complementary improvements with joint pixel-space and feature adaptation leading to the highest performing model for digit classification tasks

5.2.2. Related Work

- **TODO: cite everything**
- visual domain adaptation introduced along with a pairwise metric transform solution by Seanko et al. 2010
- further popularized by broad study of visual dataset bias (Torralba & Efros, 2011)
- early deep adaptive works focused on feature space alignment through minimizing distance between first or second order feature space statistics of source and target (Tzeng et al., 2014; Long & Wand, 2015)
- further improved thorugh use of domain adversarial objectives whereby a domain classifier is trained to distinguish between source and target representations while domain representation is learned so as to maximize error of domain classifier

- representation optimized by using standard minimax objective (Ganin & Lempitsky, 2015)
- symmetric confusion objective (Tzeng et al., 2015)
- inverted label objective (Tzeng et al., 2017)
- each related to GAN (Goodfellow et al., 2014) and followup trianing procedures for these networks (Salimans et al., 2016b; Arjovsky et al., 2017)
- these feature-space adaptation methods focus on modifications to the discriminative representation space. Other recent methods hav sought adaptation in the pixel-space using various generative approaches
- one advantage of pixel-space adaptation: result may be more human interpretable, since an image from on domain can now be visualized in a new domain
- CoGANs (Liu & Tuzel, 2016b) jointly learn source and target representation through explicit weight sharing of certain layers, source and target have unique gen adv objective
- Ghifary et al. 2016 use an additional reconstruction objective in target domain to envourage alignment in the unsupervised adaptation setting
- another approach: directly convert target image into a source style image (or vise versa), largely based on GANs (cite Goodfellow..)
- successfully applied GANs to various applications such as image generation (Denton et al., 2015; Radford et al., 2015; Zhao et al., 2016), image editing (Zhu et al., 2016) and feature learning (Salimans et al., 2016a; Donahue et al., 2017). Recent work (Isola et al., 2016; Sangkloy et al., 2016; Karacan et al., 2016) adopt conditional GANs (Mirza & Osindero, 2014) for these image-to-image translation problems (Isola et al., 2016), but require input-output image pairs for training, which is in general not available in domain adaptation problems
- no training pairs: Yoo et al. 2016 learns source to target encoder-decoder along with a generative adversarial objective on reconstruction which is applied for predicting clothing people are wearing
- Domain Transfer Network (Taigman et al. 2017b) trains generator to transform a source image into a target image by enforccing consistency in embedding space
- Shrivastava et al. 2017 instead use L1 reconstruction loss to force generated target images to be similar to original source images. works well for limited domain shifts where domains are similar in pixel-space, but can be too limiting for setting with larger domain shifts

- Bousmalis et al. 2017b use a content similarity loss to ensure the generated target image is similar to original source image; however this requires prior knowledge about which parts of the image stay the same across domains (e.g. foreground)
- cycada method does not require pre-defining what content is shared between domains and instead simply translates images back to their original domains while ensuring that the remain identical to their original version
- BiGAN (Donahue et al., 2017) and ALI (Dumoulin et al., 2016) take an approach of simultaneously learning the transofmrations between pixel and latent space.
- CycleGAN (Zhu et al., 2017) produced compelling image translation results such as generating photorealistic images from impressionism paintings or transfomring horses into zebras at high resolution using cycle-consistency loss
- this loss was simultaneously proposed by Yi et al. 2017 and Kim et al. 2017 to great effect as well
- adaptation across weather conditions in simple road scenes was first studied by Levinkov & and Fritz 2013
- convolutional domain adversarial based approach was proposed for more general drive cam scenes and for adaptation from simulated to real environments (Hoffmann et al., 2016)
- Ros et al. 2016b learns a multi-source model through concatenating all available labeled data and learning a single large model and then transfers to a sparsely labeled target domain through distillation (Hinton et al., 2015)
- Chen et al. 2017 use an adversarial objective to align both global and class-specific statistics, while mining additional temporal data from street view datasets to learn static object prior
- Zhang et al. 2017 instead perform segmentation adaptation by aligning label distributions both globally and across superpixels in an image

5.2.3. Cycle-consistent adversarial domain adaptation

- consider problem of unsupervised adaptation
- provided source data X_S, source labels Y_S, and target data X_T, but no target labels
- goal: learn a model f that can correctly predict label for target data X_T
- begin by learning source model f_S that can perform the task on the source data

- for K-way classification with cross-entropy loss:

$$\mathcal{L}_{\text{task}}(f_S, X_S, Y_S) = -\mathbb{E}_{(x_s,y_s)\sim(X_S,Y_S)} \sum_{k=1}^{K} \mathbb{1}_{[k=y_s]} \log\Big(\sigma(f_S^{(k)}(x_s))\Big) \tag{5.1}$$

  where σ denotes softmax function

- learned model f_S will perform well on source data but typically domain shift between source and target domain leads to reduced performance when evaluating on target data
- by mapping samples into common space, the model can learn on source data while still generalizing to target data
- mapping from source to target $G_{S\to T}$ trained to produce target samples that fool adversarial discriminator D_T
- discriminator attempts to classify real target data from source target data. loss function:

$$\mathcal{L}_{\text{GAN}}(G_{S\to T}, D_T, X_T, X_S) = \mathbb{E}_{x_t\sim X_T}\Big[\log D_T(x_t)\Big] \tag{5.2}$$

$$+ \mathbb{E}_{x_s\sim X_S}\Big[\log(1 - D_T(G_{S\to T}(x_s)))\Big] \tag{5.3}$$

- this objective ensures that $G_{S\to T}$, given source samples, produces convincing target samples
- this ability to directly map samples between domains allows to learn target model f_T by minimizing $\mathcal{L}_{\text{task}}(f_T, G_{S\to T}(X_S), Y_S)$
- previous approaches that optimized similar objectives have shown effetive results but in practice can often be unstable and prone to failure
- although GAN loss ensures $G_{S\to T}(x_s)$ for some x_s will resemble data drawn from X_T but there is no way to guarantee $G_{S\to T}(x_s)$ preserves structure or content of original sample x_s
- to encourage source content to be preserved during conversion: cycle-consistency constraint (**TODO: cite the 3 works that proposed it**). Mapping $G_{T\to S}$ trained according to GAN loss $\mathcal{L}_{\text{GAN}}(G_{T\to S}, D_S, X_S, X_T)$
- want $G_{T\to S}(G_{S\to T}(x_s)) \approx x_s$ and $G_{S\to T}(G_{T\to S}(x_t)) \approx x_t$
- done by imposing L1 penalty on reconstruction error (reffered to as cycle-consistency loss):

$$\mathcal{L}_{\text{cyc}}(G_{S\to T}, G_{T\to S}, X_S, X_T) = \mathbb{E}_{x_s\sim X_S}\Big[\|G_{T\to S}(G_{S\to T}(x_s)) - x_s\|_1\Big] \tag{5.4}$$

$$+ \mathbb{E}_{x_t\sim X_T}\Big[\|G_{S\to T}(G_{T\to S}(x_t)) - x_t\|_1\Big] \tag{5.5}$$

- also explicitly encourage high semantic consistency before and after image translation
- pretrain source task model f_S, fixing weights and using it as a noisy labeler by which an image to be classified in the same way after translation as it was before translation according to this classifier is encouraged
- define fixed classifier f, predicted label for given input X: $p(f, X) = \arg\max(f(X))$
- semantic consistency before and after image translation:

$$\mathcal{L}_{\text{sem}}(G_{S\to T}, G_{T\to S}, X_S, X_T, f_S) = \mathcal{L}_{\text{task}}(f_S, G_{T\to S}(X_T), p(f_S, X_T)) \tag{5.6}$$
$$+ \mathcal{L}_{\text{task}}(f_S, G_{S\to T}(X_S), p(f_S, X_S)) \tag{5.7}$$

- can be viewed analogously to content loss in style transfer (Gatys et al., 2016) or in pixel adaptation (Taigman et al., 2017a), where shared content to preserve is dictated by the source task model f_S
- could also consider a feature-level method which discriminates between the features or semantics from two image sets as viewed under a task network → additional feature level GAN loss:

$$\mathcal{L}_{\text{GAN}}(f_T, D_{\text{feat}}, f_S(G_{S\to T}(X_S)), X_T) \tag{5.8}$$

- together form complete objective:

$$\mathcal{L}_{\text{CyCADA}}(f_T, X_S, X_T, Y_S, G_{S\to T}, G_{T\to S}, D_S, D_T) \tag{5.9}$$
$$= \mathcal{L}_{\text{task}}(f_T, G_{S\to T}(X_S), Y_S) \tag{5.10}$$
$$+ \mathcal{L}_{\text{GAN}}(G_{S\to T}, D_T, X_T, X_S) + \mathcal{L}_{\text{GAN}}(G_{T\to S}, D_S, X_S, X_T) \tag{5.11}$$
$$+ \mathcal{L}_{\text{GAN}}(f_T, D_{\text{feat}}, f_S(G_{S\to T}(X_S)), X_T) \tag{5.12}$$
$$+ \mathcal{L}_{\text{cyc}}(G_{S\to T}, G_{T\to S}, X_S, X_T) + \mathcal{L}_{\text{sem}}(G_{S\to T}, G_{T\to S}, X_S, X_T, f_S) \tag{5.13}$$

- ultimately corresponds to solving for a target model f_T according to the optimization problem

$$f_T^* = \arg\min_{f_T} \min_{\substack{G_{T\to S}\\ G_{S\to T}}} \max_{D_S, D_T} \mathcal{L}_{\text{CyCADA}}(f_T, X_S, X_T, Y_S, G_{S\to T}, G_{T\to S}, D_S, D_T) \tag{5.14}$$

5.3. Technique 3

6. Comparison

6.1. training the nets on tcml cluster

6.2. comparison Benchmark(s)

7. Conclusion

To conclude...

A. Blub

Bibliography

[GPAM+14] 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.

[HTP+17] 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.

[HWYD16] 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.

[LLJX18] Peilun Li, Xiaodan Liang, Daoyuan Jia, and Eric P. Xing. Semantic-aware grad-gan for virtual-to-real urban scene adaption. *CoRR*, abs/1801.01726, 2018.

[LXG+16] Antonio M. López, Jiaolong Xu, Jose Luis Gomez, David Vázquez, and Germán Ros. From virtual to real world visual perception using domain adaptation - the DPM as example. *CoRR*, abs/1612.09134, 2016.

[MJG+18] Liqian Ma, Xu Jia, Stamatios Georgoulis, Tinne Tuytelaars, and Luc Van Gool. Exemplar guided unsupervised image-to-image translation. *CoRR*, abs/1805.11145, 2018.

[PUK+17] Xingchao Peng, Ben Usman, Neela Kaushik, Judy Hoffman, Dequan Wang, and Kate Saenko. Visda: The visual domain adaptation challenge. *CoRR*, abs/1710.06924, 2017.

[PUS+18] Xingchao Peng, Ben Usman, Kuniaki Saito, Neela Kaushik, Judy Hoffman, and Kate Saenko. Syn2real: A new benchmark forsynthetic-to-real visual domain adaptation. *CoRR*, abs/1806.09755, 2018.

[RTdS18] Pierluigi Zama Ramirez, Alessio Tonioni, and Luigi di Stefano. Exploiting semantics in adversarial training for image-level domain adaptation. *CoRR*, abs/1810.05852, 2018.

[RVRK16] Stephan R. Richter, Vibhav Vineet, Stefan Roth, and Vladlen Koltun. Playing for data: Ground truth from computer games. In Bastian Leibe,

Jiri Matas, Nicu Sebe, and Max Welling, editors, *European Conference on Computer Vision (ECCV)*, volume 9906 of *LNCS*, pages 102–118. Springer International Publishing, 2016.

[SAS+18] 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.

[TSSC19] Yi-Hsuan Tsai, Kihyuk Sohn, Samuel Schulter, and Manmohan Krishna Chandraker. Domain adaptation for structured output via discriminative patch representations. *CoRR*, abs/1901.05427, 2019.

[ZPIE17] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. *CoRR*, abs/1703.10593, 2017.