Unsupervised Pixel-Level Domain Adaptation with Generative Adversarial Networks

Konstantinos Bousmalis, Nathan Silberman, David Dohan, Dumitru Erhan, and Dilip Krishnan. CVPR (2017)

Edemir Ferreira de Andrade Junior
Department of Computer Science
Federal University of Minas Gerais

June 23, 2017

DCC
DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO

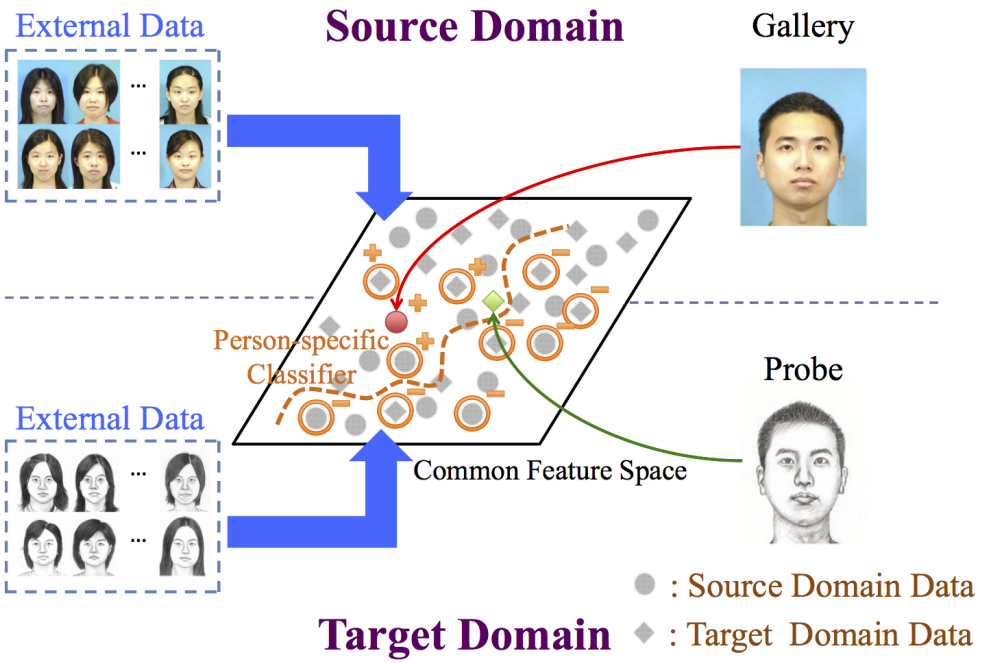
UFMG

Introduction

- In unsupervised domain adaptations, we would like to **transfer knowledge learned** from:
- a **source domain** (which we have labeled data)
- to a **target domain** (which we have no ground truth labels)

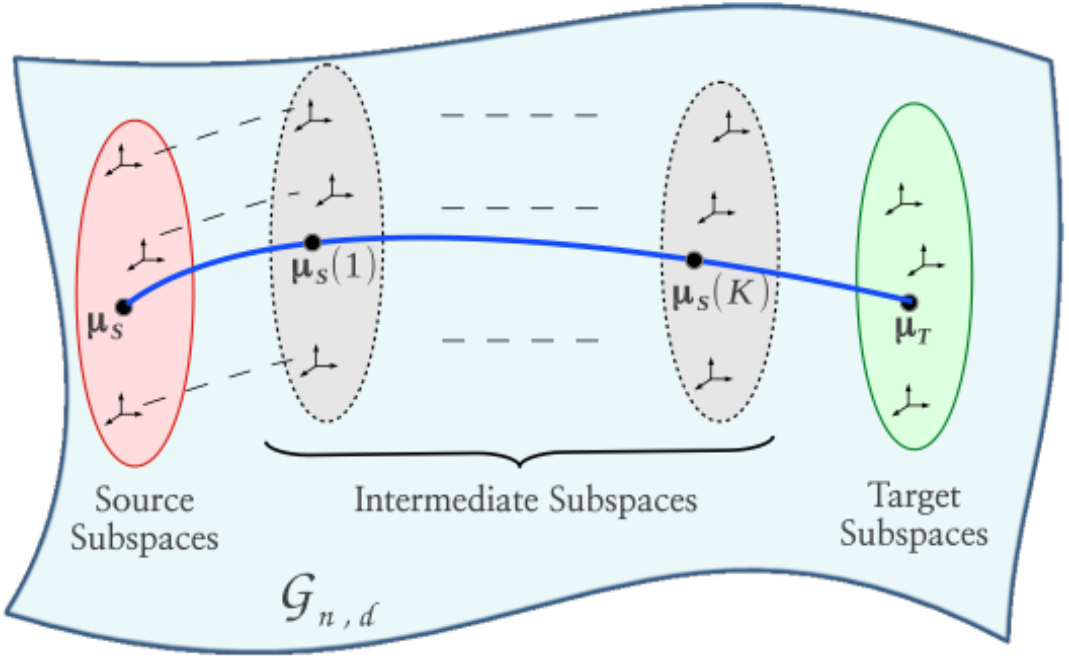
Introduction

- In general, previous work either attempts to: find a mapping from representations of the source domain to those of the target [1]



Introduction

- Seeks to find representations that are shared between the two domains [2], such that the features are invariant to the domain from which they are extracted.



While such approaches have shown good progress, they are still not on par with purely supervised approaches trained only on the target domain.

Introduction

- While such approaches have shown good progress, they are still not on par with purely supervised approaches trained only on the target domain.
- The authors propose a novel Generative Adversarial Network (GAN)-based architecture, which:
- change the images from the source domain to appear as if they were sampled from the target domain while maintaining their original content

Proposed method

- The method offers a number of advantages over existing domain adaptation approaches:
  - Decoupling from the Task-Specific Architecture
  - Generalization Across Label Spaces
  - Training Stability
  - Data Augmentation
  - Interpretability

Model

- Formally definition:
- Let $\mathbf{X}^s = \{\mathbf{x}_i^s, \mathbf{y}_i^s\}_{i=0}^{N^s}$, a labeled dataset of N^s samples from the source domain
- Let $\mathbf{X}^t = \{\mathbf{x}_i^t\}_{i=0}^{N^t}$, an unlabeled dataset of N^t samples from the target domain.
- The pixel adaptation model consists of:
- A generator function $G(\mathbf{x}^s, \mathbf{z}; \theta_G) \rightarrow \mathbf{x}^f$
  - parameterized by θ_G
  - that maps a source domain image $\mathbf{x}_i^s \in \mathbf{X}^s$ and a noise vector $\mathbf{z} \sim p_z$ to an adapted, or fake, image $\mathbf{x}^f$
- Given the generator function G, it is possible to create a new dataset $\mathbf{X}^f = \{G(\mathbf{x}^s, \mathbf{z}), \mathbf{y}^s\}$ of any size.

Learning

- Furthermore, the model is augmented by a discriminator function $D(\mathbf{x}; \theta_D)$ that outputs the likelihood 'd' that a given image $\mathbf{x}$ has been sampled from the target domain.
- The discriminator tries to distinguish between 'fake' images $\mathbf{X}^f$ produced by the generator, and 'real' images from the target domain $\mathbf{X}^t$
- Note that in contrast to the standard GAN formulation, the model's generator is conditioned on both a noise vector and an image from the source domain.
- In addition to the discriminator, the model is also augmented with a classifier $T(\mathbf{x}; \theta_T) \rightarrow$ **y'** which assigns task-specific labels **?** to images $\mathbf{x} \in \{\mathbf{X}^f, \mathbf{X}^t\}$

Learning

- Our goal is to optimize the following minimax objective:

$$\min_{\theta_G, \theta_T} \max_{\theta_D} = \alpha \mathbb{L}_d(\mathbf{D}, \mathbf{G}) + \beta \mathbb{L}_t(\mathbf{G}, \mathbf{T})$$

- where:
  $L_d(D, G)$ represents the domain loss
  $L_T(G, T)$ is a task-specific loss (softmax cross-entropy loss)
  α and β control the interaction of losses

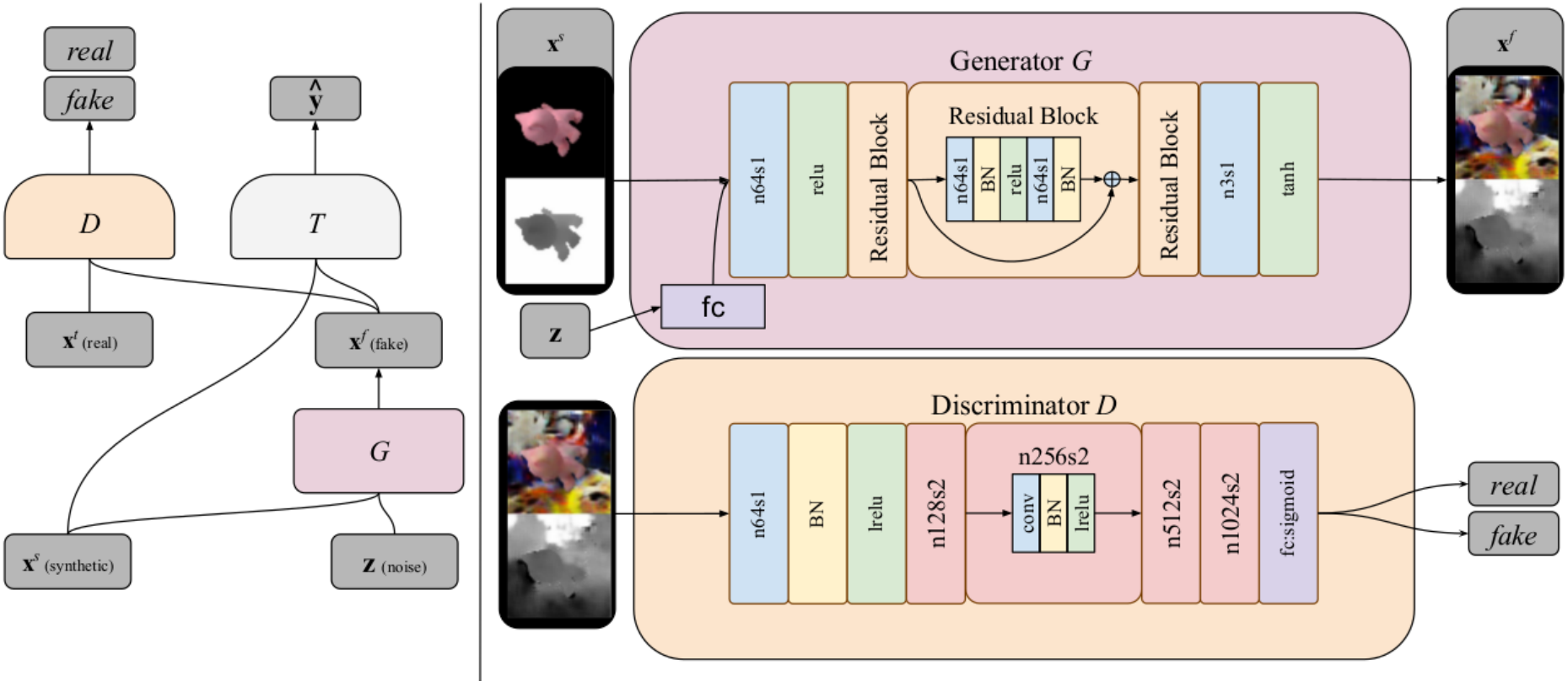
Learning

- Notice that the model train *T* with both adapted and non-adapted source images
- When training **T** only on adapted images, it's possible to achieve similar performance but doing so may require many runs with different initializations due to the instability of the model.
- Found that training classifier T on both source and adapted images avoids this scenario and greatly stabilizes training.
- Finally, it's important to reiterate that once trained, free to adapt other images from the source domain which might use a different label space.

Learning

- **G** is a convolutional neural network with residual connections that maintains the resolution of the original image.
- Our discriminator **D** is also a convolutional neural network. The minimax optimization is achieved by alternating between two steps.
- 1) During the first step, we update the discriminator and task-specific parameters θ_D, θ_T, while keeping the generator parameters θ_G fixed.
- 2) During the second step we fix θ_D, θ_T and update θ_G

Overview of the model architecture



Evaluation

- The method was evaluated in the following datasets:
  - MNIST, MNIST-M, USPS, and a variation of LINEMOD
- Qualitative and quantitative evaulation components, using a number of unsupervised domain adaptation scenarios
  - Qualitative: Visually inspecting the generated images.
  - Quantitative: comparison of the performance of previous models work and to "Source Only" and "Target Only" baselines that do not use any domain adaptation.
- MNIST to USPS
- MNIST to MNIST-M
- Synthetic Cropped LineMod to Cropped LineMod

Quantitative Results

Model	MNIST to USPS	MNIST to MNIST-M
Source Only	78.9	63.6 (56.6)
CORAL [41]	81.7	57.7
MMD [45, 31]	81.1	76.9
DANN [14]	85.1	77.4
DSN [5]	91.3	83.2
CoGAN [30]	91.2	62.0
Our model	**95.9**	**98.2**
Target-only	96.5	96.4 (95.9)

Quantitative Results

Cropped Linemod to Cropped Linemod" scenario.

Model	**Classification Accuracy**	**Mean Angle Error**
Source-only	47.33%	89.2°
MMD [45, 31]	72.35%	70.62°
DANN [14]	99.90%	56.58°
DSN [5]	**100.00%**	53.27°
Our model	99.98%	**23.5°**
Target-only	100.00%	6.47°

Qualitative Results

(a) Image examples from the Linemod dataset.

(b) Examples generated by our model, trained on Linemod.

Quantitative Results

Figure 3. Visualization of our model's ability to generate samples when trained to adapt MNIST to MNIST-M. (a) Source images $\mathbf{x}^s$ from MNIST; (b) The samples adapted with our model $G(\mathbf{x}^s, \mathbf{z})$ with random noise $\mathbf{z}$; (c) The nearest neighbors in the MNIST-M training set of the generated samples in the middle row. Differences between the middle and bottom rows suggest that the model

Quantitative Results

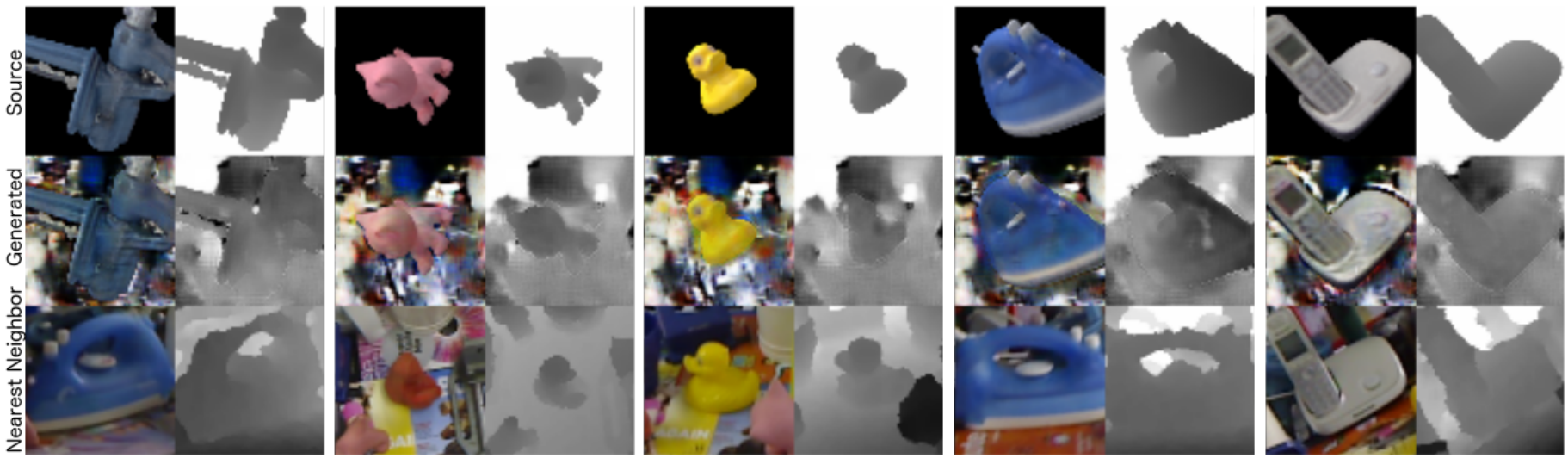


Figure 4. Visualization of our model's ability to generate samples when trained to adapt Synth Cropped Linemod to Cropped Linemod. *Top Row:* Source RGB and Depth image pairs from Synth Cropped LineMod $\mathbf{x}^s$; *Middle Row:* The samples adapted with our model $G(\mathbf{x}^s, \mathbf{z})$ with random noise $\mathbf{z}$; *Bottom Row:* The nearest neighbors between the generated samples in the middle row and images from the target training set. Differences between the generated and target images suggest that the model is not memorizing the target dataset.

Quantitative Results: Sensitivity to Used Backgrounds

Model–RGB-only	Classification Accuracy	Mean Angle Error
Source-Only–Black	47.33%	89.2°
Source-Only–INet	91.15%	50.18°
Our Model–Black	94.16%	55.74°
Our Model–INet	96.95%	36.79°

Quantitative Results: Generalization of the Model

Table 4. Performance of our model trained on only 6 out of 11 Linemod objects. The first row, 'Unseen Classes,' displays the performance on all the samples of the remaining 5 Linemod objects not seen during training. The second row, 'Full test set,' displays the performance on the target domain test set for all 11 objects.

Test Set	**Classification Accuracy**	**Mean Angle Error**
Unseen Classes	98.98%	31.69°
Full test set	99.28%	32.37°

Quantitative Results: Stability Study

Table 5. The effect of using the task and content losses L_t, L_c on the standard deviation (std) of the performance of our model on the "Synth Cropped Linemod to Linemod" scenario. L_t^{source} means we use source data to train T; $L_t^{adapted}$ means we use generated data to train T; L_c means we use our content–similarity loss. A lower std on the performance metrics means that the results are more easily reproducible.

L_t^{source}	$L_t^{adapted}$	L_c	**Classification Accuracy std**	**Mean Angle Error std**
-	-	-	23.26	16.33
-	✓	-	22.32	17.48
✓	✓	-	2.04	**3.24**
✓	✓	✓	**1.60**	6.97

Quantitative Results: Stability Study

Table 6. Semi-supervised experiments for the "Synthetic Cropped Linemod to Cropped Linemod" scenario. When a small set of 1,000 target data is available to our model, it is able to improve upon baselines trained on either just these 1,000 samples or the synthetic training set augmented with these labeled target samples.

Method	**Classification Accuracy**	**Mean Angle Error**
1000-only	99.51%	25.26°
Synth+1000	99.89%	23.50°
Our model	**99.93%**	**13.31°**

Conclusions

- They are able to do so by using a GAN?based technique, stabilized by both a task-specific loss and a novel content?similarity loss.
- decouples the process of domain adaptation from the task-specific architecture
- provides the added benefit of being easy to understand via the visualization of the adapted image outputs of the model.