AdaDepth: Unsupervised Content Congruent Adaptation for Depth Estimation

Jogendra Nath Kundu* Phani Krishna Uppala* Anuj Pahuja R. Venkatesh Babu
Video Analytics Lab, Department of Computational and Data Sciences
Indian Institute of Science, Bangalore, India
jogendrak@iisc.ac.in, {krishnaphaniiitg, anujpahuja13}@gmail.com, venky@iisc.ac.in

Supervised deep learning methods have shown promising results for the task of monocular depth estimation; but acquiring ground truth is costly, and prone to noise as well as inaccuracies. While synthetic datasets have been used to circumvent above problems, the resultant models do not generalize well to natural scenes due to the inherent domain shift. Recent adversarial approaches for domain adaption have performed well in mitigating the differences between the source and target domains. But these methods are mostly limited to a classification setup and do not scale well for fully-convolutional architectures. In this work, we propose AdaDepth - an unsupervised domain adaptation strategy for the pixel-wise regression task of monocular depth estimation. The proposed approach is devoid of above limitations through a) adversarial learning and b) explicit imposition of content consistency on the adapted target representation. Our unsupervised approach performs competitively with other established approaches on depth estimation tasks and achieves state-of-the-art results in a semi-supervised setting.

1. Introduction

Deep neural networks have brought a sudden sense of optimism for solving challenging computer vision tasks, especially in a data-hungry supervised setup. However, the generalizability of such models relies heavily on the availability of accurate annotations for massive amount of diverse training samples. To disentangle this dependency, researchers have started focusing towards the effectiveness of easily obtainable synthetic datasets in training deep neural models. For problem domains like semantic scene understanding, which face difficulty due to insufficient ground-truth for supervision, use of graphically rendered images has been a primary alternative. Even though synthetic images look visually appealing, deep models trained on them often perform sub-optimally when tested on real scenes, showing lack of generalization [19, 35]. From a probabilistic perspective, considering input samples for a network being drawn from a certain source distribution, the network can perform sufficiently well on test set only if the test data is also sampled from the same distribution. Hence, the general approach has been to transfer learned representations from synthetic to real datasets by fine-tuning the model on a mixed set of samples [42].

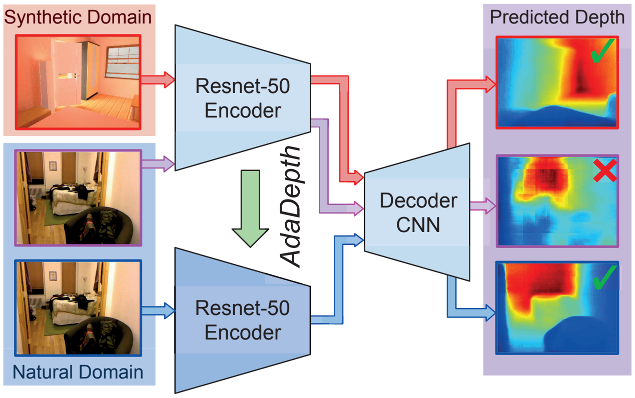

Figure 1. Illustration of the proposed domain adaptation method with input image domain discrepancy (red and blue background) followed by depth-map prediction. Color coded arrows represent corresponding RGB image and depth predictions for the synthetic-trained encoder (red and pink bordered) and for the adapted encoder (blue bordered); indicating that synthetic-trained model shows sub-optimal performance on natural images.

For depth estimation tasks, the ground-truth acquired using devices like Kinect or other depth sensors exhibits noisy artifacts [40] and hence severely limits the performance of a supervised depth prediction network. In the widely used NYU Depth Dataset [34], such cases are addressed by manually inpainting the depth values in the distorted regions. But the dataset has only a handful of such crafted samples, mainly because the process is laborious and prone to pixel-level annotation errors. These shortcomings show the need for a framework that is minimally dependent on scarce clean

*Equal contribution

ground truth data. *AdaDepth* addresses this need by adapting representations learned from graphically rendered synthetic image and depth pairs to real natural scenes.

Monocular depth estimation is an ill-posed problem; yet it has many applications in graphics [21], computational photography [2] and robotics [26, 41]. To overcome the lack of multi-view information, depth prediction models need to exploit global semantic information to regress accurate pixel-level depth. It is observed that an end-to-end Fully Convolutional Network (FCN) [25] can extract useful objectness features for efficient depth prediction without explicit enforcement. Such objectness information is exhibited by both synthetic and natural scenes as synthetic scenes also adhere to the natural distribution of relative object placement.

Previous works on domain adaptation techniques either attempt to learn an extra mapping layer to reduce domain representation gap [33] or learn domain invariant representations by simultaneously adapting for both source and target domains [44]. In contrast to classification-based approaches, there are very few works focusing on spatially structured prediction tasks [17]. Zhang *et al.* [50] show the inefficiency of classification-based approaches on such tasks, mostly because of the higher dimensional feature space. To the best of our knowledge, we are the first to explore unsupervised adversarial domain adaptation for a spatially structured regression task of depth estimation. In general, *Mode collapse* [37] is a common phenomenon observed during adversarial training in absence of paired supervision. Because of the complex embedded representation of FCN, preservation of spatial input structure in an unsupervised adaptation process becomes challenging during adversarial learning. Considering no access to target depth-maps, we address this challenge using the proposed *content congruent regularization* methods that preserve the input structural content during adaptation. The proposed adaptation paradigm results in improved depth-map estimation when tested on the target natural scenes.

Our contributions in this paper are as follows:

- We propose an unsupervised adversarial adaptation setup *AdaDepth*, that works on the high-dimensional structured encoder representation in contrast to adaptation at task-specific output layer.
- We address the problem of *mode collapse* by enforcing content consistency on the adapted representation using a novel feature reconstruction regularization framework.
- We demonstrate *AdaDepth's* effectiveness on the task of monocular depth estimation by empirically evaluating on NYU Depth and KITTI datasets. With minimal supervision, we also show state-of-the-art performance on depth estimation for natural target scenes.

2. Related work

Supervised Monocular Depth Estimation There is a cluster of previous works on the use of hand-crafted features and probabilistic models to address the problem of depth estimation from single image. Liu *et al.* [28] use predicted labels from semantic segmentation to explicitly use the objectness cues for the depth estimation task. Ladicky *et al.* [24] instead carry out a joint prediction of pixel-level semantic class and depth. Recent spurt in deep learning based methods has motivated researchers to use rich CNN features for this task. Eigen *et al.* [6] were the first to use CNNs for depth regression by integrating coarse and fine scale features using a two-scale architecture. They also combined the prediction of surface normals and semantic labels with a deeper VGG inspired architecture with three-scale refinement [5]. To further improve the prediction quality, hierarchical graphical models like CRF have been combined with the CNN based super-pixel depth estimation [27]. For continuous depth prediction, Liu *et al.* [29] use deep convolutional neural fields to learn the end-to-end unary and pairwise potentials of CRF to facilitate the training process. Laina *et al.* [25] proposed a ResNet [16] based encoder-decoder architecture with improved depth prediction results.

Unsupervised/Semi-supervised Depth Estimation Another line of related work on depth estimation focuses on unsupervised/semi-supervised approaches using geometry-based cues. Garg *et al.* [10] proposed an encoder-decoder architecture to predict depth maps from stereo pair images using an image alignment loss. Extending this, Godard *et al.* [13] proposed to minimize the left-right consistency of estimated disparities in stereo image pair for the unsupervised depth prediction task. On the other hand, Yevhen *et al.* [23] follow a semi-supervised approach using sparse ground-truth depth-map along with the image alignment loss in a stereo matching setup. Zhou *et al.* [52] used video sequences for depth prediction with view synthesis as a supervisory signal.

Transfer learning using Synthetic Scenes Lately, graphically rendered datasets are being used for various computer vision tasks such as pose prediction of human and objects [42, 47], optical flow prediction [4] and semantic segmentation [35]. Zhang *et al.* [51] proposed a large-scale physically-based rendering dataset for indoor scenes to bridge the gap between the real and synthetic scene with improved lighting setup. But training deep CNN models on such diverse synthetic images does not generalize directly for natural RGB scenes.

Domain adaptation Csurka [3] published a comprehensive survey on domain adaptation techniques for visual applications. Our work falls in the subarea of DeepDA (Deep Domain Adaptation) architectures. Several such architectures incorporate a classification loss and a discrepancy loss [12, 46, 31, 43], with Maximum Mean Discrepancy (MMD) [15] being the commonly used discrepancy loss. Long *et al.* [31] use MMD for the layers embedded in a kernel Hilbert space to effectively learn the higher order statistics between the source and target distribution. Sun and Saenko [43] proposed a deep correlation alignment algorithm (CORAL) which matches the mean and covariance of the two distributions at the final feature level to align their second-order statistics for adaptation.

Another line of work uses adversarial loss in conjunction with classification loss, with an objective to diminish domain confusion [44, 8, 9, 45]. As opposed to prior works that usually use a fully-connected layer at the end for class adaptation, we employ a DeepDA architecture for a more challenging pixel-wise regression task of depth estimation. Our proposed method uses the concept of Generative Adversarial Networks (GANs) [14] to address the domain discrepancy at an intermediate feature level. In GAN framework, the objective of generator is to produce data which can fool the discriminator, whereas the discriminator improves itself by discriminating the generated samples from the given target distribution. Following this, Isola *et al.* [18] proposed *pix2pix*, that uses a conditional discriminator to enforce consistency in generated image for a given representation. Without such conditioning, the generator can produce random samples that are inconsistent with the given input representation, while minimizing the adversarial loss. As an extension, Zhu *et al.* [53] introduced *CycleGAN*, a cycle consistency framework to enforce consistency of input representation at the generator output for unpaired image-to-image translation task.

3. Approach

Consider synthetic images $x_s \in X_s$ and the corresponding depth maps $y_s \in Y_s$ as samples from a source distribution, $p_s(x, y)$. Similarly, the real images $x_t \in X_t$ are considered to be drawn from a target distribution $p_t(x, y)$, where $p_s \neq p_t$. Under the assumption of unsupervised adaptation, we do not have access to the real depth samples $y_t \in Y_t$.

Considering a deep CNN model as a transfer function from an input image to the corresponding depth, the base model can be divided into two transformations: M_s, that transforms an image to latent representation, and T_s, that transforms latent representation to the final depth prediction. The base CNN model is first trained with full supervision from the available synthetic image-depth pairs i.e. $\bar{y_s} = T_s(M_s(x_s))$. A separate depth prediction model for the real images drawn from target distribution can be written as $\bar{y_t} = T_t(M_t(x_t))$. Due to *domain shift*, direct inference on target samples x_t through the network trained on X_s results in conflicting latent representation and predictions, i.e. $M_s(x_t) \neq M_t(x_t)$ and $T_s(M_s(x_t)) \neq T_t(M_t(x_t))$. For effective domain adaptation, ideally both M_s and T_s have to be adapted to get better performance for the target samples. Considering that X_s and X_t only exhibit perceptual differences caused by the graphical rendering process, both domains have many similarities in terms of objectness information and relative object placement. Therefore, we only adapt M_t for the target distribution $p_t(x)$. To generalize the learned features for the new domain, we plan to match the latent distributions of $M_s(X_s)$ and $M_t(X_t)$ so that the subsequent transformation T_s can be used independent of the domain as $T_s = T_t = T$.

We start the adaptation process by initializing M_t and T_t with the supervisely trained weights from M_s and T_s respectively. To adapt the parameters of M_t for the target samples x_t, we introduce two different discriminators D_F and D_Y. The objective of D_F is to discriminate between the source and target latent representations $M_s(x_s)$ and $M_t(x_t)$, whereas the objective of D_Y is to discriminate between Y_s and $T(M_t(X_t))$. Assuming similar depth map distribution for both synthetic and real scenes ($p(Y_s = y_s) \approx p(Y_t = y_t)$), inferences through the corresponding transformation functions $T(M_s(x_s))$ and $T(M_t(x_t))$ are directed towards the same output density function.

We use a ResNet-50 [16] based encoder-decoder architecture [25] for demonstrating our approach. Existing literature [49] reveals that in hierarchical deep networks, the lower layers learn generic features related to the given data distribution whereas the consequent layers learn more task specific features. This implies that the transferability of learned features for different data distributions (source and target) decreases as we move from lower to higher layers with an increase in domain discrimination capability. We experimentally evaluated this by varying the number of shared layers between M_s and M_t, starting from the initial layers to the final layers. From Figure 3, it is clear that towards higher layers of M_s, features are more discriminable for synthetic versus natural input distribution. Therefore, we deduce that adaptation using only *Res-5* blocks of M_t (*Res-5a*, *Res-5b* and *Res-5c*) and fixed shared parameters of other layers (Figure 2) is optimal for adversarial adaptation as it requires minimal number of parameters to update.

In rest of this section, we describe the adversarial objectives along with the proposed content consistent loss formulations to update the parameters of M_t for depth estimation.

3.1. Adversarial Objectives

We define an adversarial objective L_{advD} at the prediction level for D_Y and an adversarial objective L_{advF} at the latent space feature level for D_F. They can be defined as:

$$\begin{aligned}\mathcal{L}_{advD} = &\mathbb{E}_{y_s \sim Y_s}[\log D_Y(y_s)] \\ &+ \mathbb{E}_{x_t \sim X_t}[\log (1 - (D_Y(T(M_t(x_t)))))] \quad (1)\end{aligned}$$

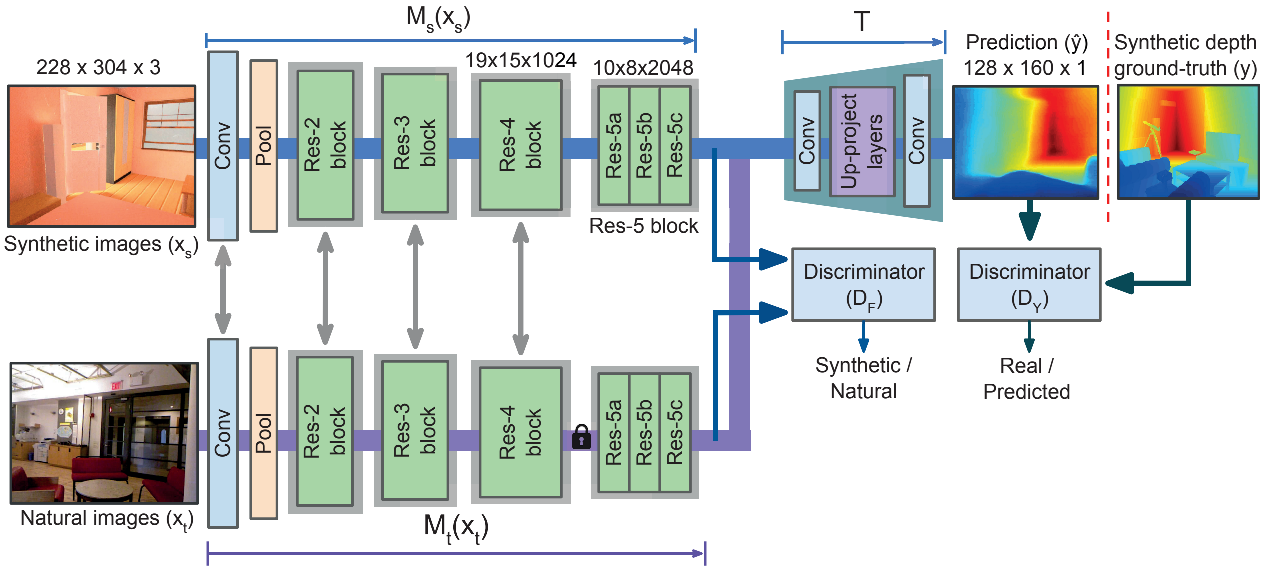

Figure 2. *AdaDepth*: Our deep residual encoder-decoder base architecture with adversarial setup illustrating different transformation functions as described in Section 3. The source (synthetic) and target (real) branch are specified by blue and purple channel respectively. The double-headed arrows between res-blocks indicate parameter sharing. Note that during adaptation of the synthetic-trained $T(M_t(x_t))$, only the layers in purple branch are updated (i.e. *Res-5* block) until the location of *lock icon*.

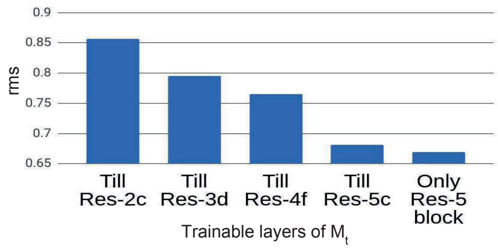

Figure 3. Effect of various weight sharing strategies on adversarial adaptation process with domain consistency regularization (Section 3.2.1).

$$\mathcal{L}_{advF} = \mathbb{E}_{x_s \sim X_s}[\log D_F(M_s(x_s))] + \mathbb{E}_{x_t \sim X_t}[\log(1 - (D_F(M_t(x_t))))] \quad (2)$$

M_t parameters are updated to minimize both the adversarial losses, whereas the discriminators D_Y and D_F are updated to maximize the respective objective functions. The final objective to update the parameters of M_t, D_Y and D_F can be expressed as $\min_{M_t} \max_{D_Y} \mathcal{L}_{advD}$ and $\min_{M_t} \max_{D_F} \mathcal{L}_{advF}$.

3.2. Content Congruency

In practice, a deep CNN exhibits complex output and latent feature distribution with multiple modes. Relying only on adversarial objective for parameter update leads to *mode collapse*. Theoretically, adversarial objective should work for a stochastic transfer function. However, since we do not use any randomness in our depth prediction model, it is highly susceptible to this problem. At times, the output prediction becomes inconsistent with the corresponding input image even at optimum adversarial objective. To tackle this, we enforce content congruent regularization methods as discussed below.

3.2.1 Domain Consistency Regularization (DCR)

Since we start the adversarial learning after training on synthetic images, the resultant adaptation via adversarial objective should not distort the rich learned representations from the source domain. It is then reasonable to assume that M_s and M_t differ by a small perturbation. We do so by enforcing a constraint on the learned representation while adapting the parameters for the new target domain. As per the proposed constraint, the latent representation for the samples from the target domain $M_t(x_t)$ must be regularized during the adaptation process with respect to $M_s(x_t)$ and can be represented as:

$$\mathcal{L}_{domain} = \mathbb{E}_{x_t \sim X_t}[\|M_s(x_t) - M_t(x_t)\|_1] \quad (3)$$

3.2.2 Residual Transfer Framework (RTF)

Considering the adaptation process from M_s to M_t as a feature perturbation, Long *et al.* [32] proposed a residual transfer network to model M_t as $M_s + \Delta M$. On similar lines, we implement an additional skip multi-layer CNN block with additive feature fusion to model ΔM such that $M_t = M_s + \Delta M$ (Figure 4a). To maintain content consistency, ΔM is constrained to be of low value so as to avoid distortion of the base M_s activations. Also note that in this