Gotta Adapt ’Em All: Joint Pixel and Feature-Level Domain Adaptation for Recognition in the Wild

Luan Tran1* Kihyuk Sohn2 Xiang Yu2 Xiaoming Liu1 Manmohan Chandraker2,3
1Michigan State University 2NEC Labs America 3UC San Diego

Abstract

Recent developments in deep domain adaptation have allowed knowledge transfer from a labeled source domain to an unlabeled target domain at the level of intermediate features or input pixels. We propose that advantages may be derived by combining them, in the form of different insights that lead to a novel design and complementary properties that result in better performance. At the feature level, inspired by insights from semi-supervised learning, we propose a classification-aware domain adversarial neural network that brings target examples into more classifiable regions of source domain. Next, we posit that computer vision insights are more amenable to injection at the pixel level. In particular, we use 3D geometry and image synthesis based on a generalized appearance flow to preserve identity across pose transformations, while using an attribute-conditioned CycleGAN to translate a single source into multiple target images that differ in lower-level properties such as lighting. Besides standard UDA benchmark, we validate on a novel and apt problem of car recognition in surveillance images using labeled images from the web, handling explicitly specified, nameable factors of variation through pixel-level and implicit, unspecified factors through feature-level adaptation.

1. Introduction

Deep learning has made an enormous impact on many applications in computer vision such as generic object recognition [22, 44, 47, 17], fine-grained categorization [58, 21, 41], object detection [26, 27, 28, 36, 37], semantic segmentation [6, 42] and 3D reconstruction [52, 51]. Much of its success is attributed to the availability of large-scale labeled training data [8, 15]. However, this is hardly true in many practical scenarios: since annotation is expensive, most data remains unlabeled. Consider car recognition problem from surveillance images, where factors such as camera angle, distance, lighting or weather condition are different across locations. It is not feasible to exhaustively annotate all these images. Meanwhile, there exist abundant labeled data from web domain [21, 61, 12], but with very different image characteristics that precludes direct transfer of discriminative CNN-based classifiers. For instance, web images might be from catalog magazines with professional lighting and ground-level camera poses, while surveillance images can originate from cameras atop traffic lights in challenging lighting and weather conditions.

Unsupervised domain adaptation (UDA) is a promising tool for overcoming the lack of labeled training data problem in target domains. Several approaches aim to match distributions between source and target domains at different levels of representations, such as feature [56, 55, 11, 45, 31] or pixel levels [48, 43, 65, 3]. Certain adaptation challenges are better handled in the feature space, but feature-level DA is a black-box algorithm for which adding domain-specific insights during adaptation is more difficult than in pixel space. On the contrary, pixel space is much higher-dimensional and the optimization problem is under-determined. It has been

Table 1: Our framework for unsupervised domain adaptation at multiple semantic levels: at feature-level, we bring insights from semi-supervised learning to obtain highly discriminative domain-invariant representations; at pixel-level, we leverage complementary domain-specific vision insights e.g., geometry and attributes. Our joint pixel and feature-level DA demonstrates significant improvement over individual adaptation counterparts as well as other competing methods such as CyCADA (CycleGAN+DANN) [18] on car recognition in surveillance domain under UDA setting. Please see Section 5 for complete experimental analysis.

*This work is done when L. Tran was an intern at NEC Labs America.
an open challenge how to effectively combine them.

In this work we address this challenge by leveraging complementary tools that are better-suited at each level (see figure in Table 1). Specifically, we posit that feature-level DA is more amenable to techniques from semi-supervised learning (SSL), while pixel-level DA allows domain-specific insights from computer vision. In Section 3, we present our feature-level DA method called classification-aware domain adversarial neural network (DANN-CA) that jointly parameterizes the classifier and domain discriminator inspired by an instance of SSL algorithm [40]. We show this to be a generalization of DANN [11] to incorporate constraints (Fig. 1) that guide discriminator to easily find major modes corresponding to classes in the feature space, and in turn put target examples into more classifiable regions via adversarial loss.

A challenge for pixel-level DA is to simultaneously transform source image properties at multiple semantic levels. In Section 4, we present pixel-level DA by image transformations that make use of vision concepts to deal with different variation factors, such as photometric or geometric transformations (Fig. 2). For recognition in surveillance domain. To handle low-level transformations, we propose an attribute-conditioned CycleGAN (AC-CGAN) that extends [65] to generate multiple target images with different attributes. To handle high-level identity-preserving pose transformations, we use an appearance flow (AF) [64], an warping-based image synthesis tool. To reduce semantic gaps between synthetic and real images, we propose a generalization of AF with 2D keypoints as a domain bridge.

In Section 5, we evaluate our framework on car recognition in surveillance images from the comprehensive cars (CompCars) dataset [61]. We define an experimental protocol with web images as labeled source domain and surveillance images as unlabeled target domain. We explicitly handle nameable factors of variation such as pose and lighting through pixel-level DA, while other nuisance factors are handled by feature-level DA. As in Table 1, we achieve 84.20% accuracy, reducing error by 64.9% from a model trained only on the source domain. We present ablation studies to demonstrate the importance of each adaptation component by extensively evaluating performances with various mixes of components. We further validate the effectiveness of our proposed feature-level DA methods on standard UDA benchmarks, namely digits and traffic signs [11] and office-31 [38], achieving state-of-the-art recognition performance.

In summary, the contributions of our work are:

- A novel UDA framework that adapts at multiple semantic levels from feature to pixel, with complementary insights for each type of adaptation.
- For feature-level DA, a connection of DANN to a semi-supervised variant, motivating a novel regularization via classification-aware domain adversarial neural network.
- For pixel-level DA, an attribute-conditioned CycleGAN to translate a source image into multiple target images with different attributes, along with an warping-based image synthesis for identity-preserving pose translations via a keypoint-based appearance flow.
- A new experimental protocol on car recognition in surveillance domain, with detailed analysis of various modules and efficacy of our UDA framework.
- State-of-the-art performance on standard UDA benchmarks, such as office-31 and digits, traffic signs adaptation tasks, with our feature-level DA method.

Due to a large volume of our work, we put additional details in Section S1–S6 of the supplementary material and at https://abc.def.ghi.jkl.

2. Related Work

Unsupervised Domain Adaptation. Following theoretical developments of domain adaptation [2, 1], a major challenge is to define a proper metric measuring the domain difference. The maximum mean discrepancy [29, 56, 9, 55, 46], which measures the difference based on kernels, and more recently the domain adversarial neural network [11, 4, 3, 45], which measures the difference using discriminator, have been successful. Noticing a similarity in problem settings between UDA and SSL, there have been attempts to combine ideas from SSL. For example, entropy minimization [14] has been used in addition to domain adversarial loss [30, 31]. Our feature-level DA is built on DANN by resolving issues of discriminator in discovering modes in the feature space. Our formulation also connects tightly to SSL and we explain why entropy minimization is essential for DANN.

Perspective Transformation. Previous works [60, 23, 50] propose encoder-decoder networks to generate output images of target viewpoint. Adversarial learning for perspective transformation [53, 54, 62] has demonstrated good performance on disentangling viewpoint from other appearance factors, but there are still concept (e.g., class label) switches in unpaired settings. Rather than learning the output distribution, [64, 34] propose an warping-based viewpoint synthesis by estimating a pixel-level flow field. We extend it to improve generalization to real images using synthetic-to-real domain invariant representations such as 2D keypoints.

Image-to-image Translation. With the success of GAN on image generation [13, 35], conditional variants of GAN [32] have been successfully adopted to image-to-image translation problems in both paired [19] and unpaired [43, 48, 65] training settings. Our model extends the work of [65] for image translation in unpaired settings using a control variable or visual attribute [59] to generate multiple outputs.

Multi-level UDA. A combination of pixel and feature level
adaptation has been attempted in [18], however, we differ in a few important ways. Specifically, we go further in using insights from SSL that allow novel regularizations for feature-level DA, while exploiting 3D geometry and attribute-based conditioning in GANs to simultaneously handle high-level pose and low-level lighting variations. Our experiments include a detailed study of the complementary benefits, as well as the effectiveness of various adaptation modules. While [18] consider problems such as semantic segmentation, we study a car recognition problem that highlights the need for adaptation at various levels. We also demonstrate state-of-the-art results on standard UDA benchmarks.

3. Domain Adversarial Feature Learning

This section describes a classification-aware domain adversarial neural network (Fig. 1(b)) that improves upon a domain adversarial neural network [11] by joint parameterization of classifier and discriminator.

**Notation.** Let \( X_S, X_T \subset X \) be source and target datasets and \( Y = \{1, \ldots, N\} \) be the set for class label. Let \( f : X \rightarrow \mathbb{R}^K \) be the feature generator, e.g., CNN, with parameters \( \theta_f \) that maps input \( x \in X \) into a \( K \)-dimensional vector.

3.1. Recap: Domain Adversarial Neural Network

Domain adversarial training [11] aims to adapt classifier learned from the labeled source domain to unlabeled target domain by making feature distributions of the two domains indistinguishable. This is achieved through a domain discriminator \( D : \mathbb{R}^K \rightarrow (0, 1) \) that tells whether features from the two domains are still distinguishable. Then, \( f \) is trained to confuse \( D \) while classifying the source data correctly:

\[
\begin{align*}
\max_{\theta_c} \{ & \mathcal{L}_C = \mathbb{E}_{X_S} \log C(f, y) \} \quad (1) \\
\max_{\theta_d} \{ & \mathcal{L}_D = \mathbb{E}_{X_S} \log (1 - D(f)) + \mathbb{E}_{X_T} \log D(f) \} \quad (2) \\
\max_{\theta_f} \{ & \mathcal{L}_f = \mathcal{L}_C + \lambda \mathbb{E}_{X_T} \log (1 - D(f)) \} \quad (3)
\end{align*}
\]

\( C : \mathbb{R}^K \times Y \rightarrow (0, 1) \) is a class score function that outputs the probability of an input \( x \) being a class \( y \) among \( N \) categories, i.e., \( C(f(x), y) = P(y | f(x); \theta_c) \). \( \lambda \) balances classification and domain adversarial losses. The parameters \( \{\theta_c, \theta_d\} \) and \( \{\theta_f\} \) are updated in turn using stochastic gradient descent.

3.2. Classification-Aware Adversarial Learning

We note that the problem setup of unsupervised domain adaptation is not different from that of semi-supervised learning once we remove the notion of domains. Inspired by the semi-supervised learning formulation of GANs [40, 7] we propose a new domain adversarial learning objective that jointly parameterizes classifier and discriminator as follows:

\[
\begin{align*}
\max_{\theta_c} \{ & \mathcal{L}_C = \mathbb{E}_{X_S} \log \mathcal{C}(y) + \mathbb{E}_{X_T} \log \mathcal{C}(N+1) \} \quad (4) \\
\max_{\theta_f} \{ & \mathcal{L}_f = \mathbb{E}_{X_T} \log \mathcal{C}(y|Y) + \lambda \mathbb{E}_{X_T} \log (1 - \mathcal{C}(N+1)) \} \quad (5)
\end{align*}
\]

3.2.1. Discriminator Should Know Classification Boundary

Mode collapse is a critical issue in adversarial learning. To prevent it, discriminator needs to discover as many modes in data distribution as possible. While it is difficult to describe the modes in the input space for generative modeling [13], it is relatively easy to characterize the modes in the feature space: there are \( N \) major modes, each of which corresponds to each output class, and the discriminator is demanded for discovering these modes in the feature space. Unfortunately, the discriminator of DANN is trained with binary supervision, implying that the mode discovery is done unsupervisedly. On the other hand, the modes are already embedded in the discriminator of DANN-CA via joint parameterization and the adversarial learning can be made easier.

We further investigate the gradient of adversarial loss in (3) and (5) with respect to \( f \). For the ease of presentation, we assume linear classifier and discriminator. Complete derivation including non-linear version is in Section S1.

\[
\begin{align*}
\frac{\partial \log (1 - D(f))}{\partial f} & = - D(f)w_d \\
\frac{\partial \log (1 - \mathcal{C}(N+1))}{\partial f} & = - \mathcal{C}(N+1)(w_{N+1} - \sum_{y=1}^{N} w_y \mathcal{C}(y|Y))
\end{align*}
\]
where $w_d, w_g \in \mathbb{R}^K$, $y \in \{1, \ldots, N+1\}$ are discriminator and classifier weights, respectively. As is evident from (7), the adversarial loss of DANN cannot capture multiple modes as all target examples induce the gradient of the same direction. Even if we use MLP discriminator in practice, it still demands to discover modes correspond to classes without supervision. The joint parameterization allows not only to push features away from the target domain, but also guides them to be pulled close to classes based on the conditional score $\bar{C}(y|\mathcal{Y})$ of individual target examples.

**Relation to DANN [11].**

Besides parameterization, the learning objectives are tightly linked to those of DANN [11]. For instance, $\bar{L}_D = L_D$ with $D = \bar{C}(N+1)$ and $C(y) = \bar{C}(y|\mathcal{Y})$. It is also easy to show $\bar{L}_C = L_C + L_D$ by rewriting $\bar{C}(y)$ using (6) as follows:

$$
\bar{L}_C = \mathbb{E}_{X_s} \log \bar{C}(y|\mathcal{Y}) + \mathbb{E}_{X_t} \log(1-\bar{C}(N+1)) + \mathbb{E}_{X_s} \log \bar{C}(N+1)
$$

(8)

**Relation to Maximum Classifier Discrepancy [39].**

We also relate our proposed DANN-CA to recently proposed maximum classifier discrepancy (MCD) learning for UDA. MCD learns shared feature extractor by reducing the prediction discrepancy between two (or more) maximally different classifiers. We show that our DANN-CA can be understood as MCD with choices of classifiers and the divergence. Following [39], we define the two classification distributions:

$$
p_1(y|x) = \bar{C}(y|\mathcal{Y}), \ p_2(y|x) = \bar{C}(y), \ y \leq N+1
$$

(9)

Note that two classifiers $F_1$ and $F_2$ in [39] are both represented as $(N+1)$-way classifier. Using KL divergence, we obtain following discrepancy loss:

$$
-\text{KL}(p_1\|p_2) = \log(1-\bar{C}(N+1))
$$

(10)

which is equivalent to the adversarial loss in (5). This analysis provides a unified view of DANN, MCD and more general class of consistency-based SSL algorithms [24, 49, 10]. A theoretical comparison of UDA algorithms is important as empirical comparison could sometimes be misleading [33]. A full derivation of (10) and analysis are in Section S2.

### 4. Pixel-level Cross-Domain Image Translation

As is common for neural networks, DANN is a black-box algorithm and adding domain-specific insight is non-trivial. On the other hand, certain challenges in DA can be better handled in image space. In this section, we introduce complementary tools to deal with nameable factors of variation, such as photometric or perspective transformations, at the pixel level. To achieve this, we propose extensions to prior works on CycleGAN [65] and appearance flows [64]. We describe with an illustrative application of car recognition in surveillance domain where the only labeled data is from web domain. The pipeline of our system is in Fig. 2.

**4.1. Photometric Transformation by CycleGAN**

As noticed from Fig. 2, images from surveillance domain have disparate color statistics from web images as they might be acquired outdoors at different times with significant lighting variations. CycleGAN [65] is proposed as a promising tool for image translation by disentangling low-level statistics from geometric structure. A limitation, however, is that it generates a single output when there could be multiple output styles. We propose an attribute-conditioned CycleGAN (AC-CGANG) that generates diverse output images with the same geometric structure by incorporating a conditioning variable into generators.

Let $A$ be a set of attributes in the target domain (day or night). We learn a generator $G : \mathcal{X}_S \times A \rightarrow \mathcal{X}_T$ that translates an image with certain style $a \in A$ by fooling an attribute-specific discriminator $D_a$. The learning objectives are:

$$
\max_{\theta_d} \{ \mathcal{L}_{D_a} = \mathbb{E}_{X_s} \log D_a(x) + \mathbb{E}_{X_t} \log(1-D_a(G(x,a))) \}
$$

(11)

$$
\max_{\theta_g} \{ \mathcal{L}_G = \mathbb{E}_{X_s} \mathbb{E}_A \log D_a(G(x,a)) \}
$$

(12)

We use multiple discriminators to prevent competition between different attribute configurations, but it is feasible to have one discriminator with $(|A|+1)$-way domain classification loss [48]. Also, one might afford to have multiple generators per attribute without sharing parameters. Following [65], we add cycle consistency loss as follows:

$$
\mathbb{E}_{X_s} \| F(G(x,a), a) - x \|_1 + \mathbb{E}_{X_t} \| G(F(x,a), a) - x \|_1
$$

(13)

where an inverse generator $F$ maps outputs back to source domain $F(G(x,a), a) = x$. We also use patchGAN [19, 65] for discriminators that makes real or fake decisions from local patches and UNet [19] for generators, each of which contributes to preserve geometric structure of an input image.

\[\text{Empirically, using two separate generators for day and night performs slightly better than a single generator. Please see Section S6 for results.}\]
I propose a 2D keypoint-based AFNet (KFNet) that takes work is trained only on the synthetic data [25]. To this end, we argue that 2D keypoints contain sufficient information (AF-based transformation may have a better chance of pre-

where Appearance Flow.

appearance flow field in horizontal and vertical axes called appearance flow (AF) [64]. In contrast to neural network based image synthesis methods [50], AF-based transformation may have a better chance of preserving object identity since all pixels of an output image are from an input image and no new information, such as learned priors in the decoder network, is introduced.

Appearance Flow.

Zhou et al. [64] propose to estimate a pixel-level dense flow from an input image with target viewpoint and synthesize an output by reorganizing pixels using bilinear sampling [20]:

\[ I_p^{i,j} = \sum_{(h,w) \in N} I_s^{h,w}(1 - |F_s^{i,j} - h|)(1 - |F_s^{i,j} - w|), \] (14)

where \(I_s, I_p\) are input and output, \((F_x, F_y)\) is a pixel-level flow field in horizontal and vertical axes called appearance flow (AF), estimated by an AF estimation network (AFNet). \(N\) denotes 4-pixel neighborhood of \((F_x^{i,j}, F_y^{i,j})\). In contrast to neural network based image synthesis methods [50], AF-based transformation may have a better chance of preserving object identity since all pixels of an output image are from an input image and no new information, such as learned priors in the decoder network, is introduced.

Keypoint-based Robust Estimation of AF.

AFNet requires image pairs \((I_s, I_t)\) with perspective being the only factor of variation for training. Since it is infeasible to collect precisely controlled dataset of real images at large-scale, rendered images from 3D CAD models are used. However, this induces a generalization issue when applied to real images at test time.

To make AFNet generalizable, we propose sparse 2D keypoints in replace of an RGB image as an input to AFNet both at train and test times. Although sparse, for objects like cars, we argue that 2D keypoints contain sufficient information to reconstruct (rough) geometry of an entire object, while being invariant across rendered and real domains. Besides, keypoint estimation can be done robustly across synthetic and real domains even when the keypoint localization network is trained only on the synthetic data [25]. To this end, we propose a 2D keypoint-based AFNet (KFNet) that takes estimated 2D keypoints and the target viewpoint as an input pair to generate flow fields \(F\) for synthesis.

The KFNet is trained using rendered image pairs. Moreover, we leverage pretrained AFNet that produces a robust AF representation for rendered images to train the KFNet by distillation. The learning objective is as follows:

\[
\min \{ L = \|F_{\text{kpt}} - F_{\text{pix}}\|_1 + \lambda \| I_p(F_{\text{kpt}}, I_s) - I_t\|_1 \} \] (15)

where \(F_{\text{kpt}}\) is an estimated appearance flow by KFNet and \(F_{\text{pix}}\) is that by AFNet. Here, \(I_p(F, I_s)\) is the predicted image from \(I_s\) using \(F\) based on (14). The training framework by distillation is visualized in Fig. 3.

5. Experiments

We strive for providing empirical evidence for the effectiveness of individual components of our proposed framework as well as their complementarity by conducting extensive experiments on car recognition in surveillance domain. For feature-level adaptation, we also provide performance comparison on standard benchmarks, namely digits and traffic signs [11] and office-31 [38].

5.1. Car Recognition in Surveillance Domain

Dataset. CompCars dataset [61] offers two datasets, one from the web and the other from the surveillance (SV) domains. It contains 52,083 web images across 431 car models and 44,481 SV images across 181 car models. Samples are in Fig. 2. The SV test set contains 9,630 images across 181 car models, of which 6,404 images are in day condition.3

To train an appearance flow estimation network, based on empirical distribution of web images, we render car images at multiple elevation (0°~30°) and azimuth variations (±15°) from ShapeNet [5]. We apply pixel-level adaptation to 5,508 web images of frontal view.

Training. The task is to train a classifier that works well on SV images using labeled web (source) and unlabeled SV (target) images. We use ResNet-18 [17] fine-tuned on web images as our baseline. Then, we train models with different integration of pixel and feature-level DA components. Note that synthesized images by pixel-level adaptation are considered as labeled training examples. Furthermore, we use data augmentation, such as translation, horizontal flip or chromatic jitter, for all models by default. We refer to Section S4.3 for more training details.

Model Selection. While it is desirable to do a model selection without labeled examples from the target domain, to our knowledge, there does not exist an unsupervised evaluation measure that is highly correlated with the supervised performance [4]. To allow more meaningful and interpretable comparisons across different methods, we report our results based on a supervised model selection [4] using a small validation set containing approximately 5 labeled examples per

3We provide a binary label (day or night) for images from surveillance domain by computing the mean pixel-intensity.
validation accuracy

Table 2: Accuracy on SV test set with different perspective transformation methods: appearance flow (AF), keypoint-based AF (KF) and with mask (MKF).

<table>
<thead>
<tr>
<th>ID</th>
<th>Perspective Transformation</th>
<th>SV</th>
<th>Day</th>
<th>Night</th>
</tr>
</thead>
<tbody>
<tr>
<td>M1</td>
<td>Baseline (web only)</td>
<td>54.98</td>
<td>72.67</td>
<td>19.87</td>
</tr>
<tr>
<td>M3</td>
<td>Appearance Flow (AF)</td>
<td>59.73</td>
<td>75.78</td>
<td>27.87</td>
</tr>
<tr>
<td>M4</td>
<td>Keypoint-based AF (KF)</td>
<td>61.55</td>
<td>77.98</td>
<td>28.92</td>
</tr>
<tr>
<td>M5</td>
<td>MKF with mask (MKF)</td>
<td>64.30</td>
<td>78.62</td>
<td>35.87</td>
</tr>
</tbody>
</table>

Table 3: Accuracy on SV test set with different photometric transformation methods: CycleGAN [65], attribute-conditioned CycleGAN (AC-CGAN), and combinations with MKF.

<table>
<thead>
<tr>
<th>ID</th>
<th>Photometric Transformation</th>
<th>SV</th>
<th>Day</th>
<th>Night</th>
</tr>
</thead>
<tbody>
<tr>
<td>M1</td>
<td>Baseline (web only)</td>
<td>54.98</td>
<td>72.67</td>
<td>19.87</td>
</tr>
<tr>
<td>M6</td>
<td>CycleGAN</td>
<td>64.32</td>
<td>77.01</td>
<td>39.12</td>
</tr>
<tr>
<td>M7</td>
<td>AC-CGAN</td>
<td>67.30</td>
<td>78.20</td>
<td>45.66</td>
</tr>
<tr>
<td>M8</td>
<td>MKF+CycleGAN</td>
<td>71.21</td>
<td>81.54</td>
<td>50.68</td>
</tr>
<tr>
<td>M9</td>
<td>MKF+AC-CGAN</td>
<td>79.71</td>
<td>84.10</td>
<td>70.99</td>
</tr>
</tbody>
</table>

Table 4: Accuracy on SV test set with pixel and feature-level DA components. We consider an MKF for perspective and attribute-conditioned CycleGAN (AC-CGAN) for photometric transformations for pixel-level DA, and DANN-CA for feature-level DA.

<table>
<thead>
<tr>
<th>ID</th>
<th>Pixel</th>
<th>Feature</th>
<th>SV</th>
<th>Day</th>
<th>Night</th>
</tr>
</thead>
<tbody>
<tr>
<td>M1</td>
<td>-</td>
<td></td>
<td>54.98</td>
<td>72.67</td>
<td>19.87</td>
</tr>
<tr>
<td>M2</td>
<td>Supervised (web+SV)</td>
<td>98.63</td>
<td>98.92</td>
<td>98.05</td>
<td></td>
</tr>
<tr>
<td>M10</td>
<td>–</td>
<td>DANN</td>
<td>60.40</td>
<td>75.56</td>
<td>30.31</td>
</tr>
<tr>
<td>[18]</td>
<td>CycleGAN</td>
<td>DANN</td>
<td>64.82</td>
<td>76.35</td>
<td>41.93</td>
</tr>
<tr>
<td>M11</td>
<td>–</td>
<td>DANN-CA</td>
<td>75.83</td>
<td>76.73</td>
<td>74.05</td>
</tr>
<tr>
<td>M12</td>
<td>MKF</td>
<td>DANN-CA</td>
<td>80.40</td>
<td>82.50</td>
<td>76.22</td>
</tr>
<tr>
<td>M13</td>
<td>AC-CGAN</td>
<td>DANN-CA</td>
<td>80.24</td>
<td>82.15</td>
<td>76.44</td>
</tr>
<tr>
<td>M14</td>
<td>MKF+AC-CGAN</td>
<td>DANN-CA</td>
<td>84.20</td>
<td>85.77</td>
<td>81.10</td>
</tr>
</tbody>
</table>

Figure 4: Accuracy of DANN (M10) and DANN-CA (M11) on SV validation set over training. We also visualize t-SNE plots of each model at different training epochs.

Table 5: Evaluation on UDA tasks [11], such as MNIST to MNIST-M (M→MM), Synthetic Digits to SVHN (S→S), SVHN to MNIST (S→M), MNIST to SVHN (M→S), or Synthetic Signs to GTSRB (S→G). Test set accuracy averaged over 10 runs is reported. The best performers and the ones within standard error are bold-faced.

<table>
<thead>
<tr>
<th>Method</th>
<th>A→W</th>
<th>D→W</th>
<th>W→D</th>
<th>A→D</th>
<th>D→A</th>
<th>W→A</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source only</td>
<td>76.42</td>
<td>96.76</td>
<td>97.99</td>
<td>79.81</td>
<td>60.44</td>
<td>59.53</td>
</tr>
<tr>
<td>DANN</td>
<td>85.97</td>
<td>96.87</td>
<td>97.94</td>
<td>84.12</td>
<td>67.63</td>
<td>66.78</td>
</tr>
<tr>
<td>DANN-CA</td>
<td>91.35</td>
<td>98.24</td>
<td>99.48</td>
<td>89.94</td>
<td>98.70</td>
<td>98.00</td>
</tr>
</tbody>
</table>

Table 6: Evaluation on office-31 benchmark [38] between Amazon (A), DSLR (D), and Webcam (W) domains using ResNet-50. Target domain accuracy averaged over 5 runs is reported. The best performers and the ones within standard error are bold-faced.

<table>
<thead>
<tr>
<th>Method</th>
<th>A→W</th>
<th>D→W</th>
<th>W→D</th>
<th>A→D</th>
<th>D→A</th>
<th>W→A</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source only</td>
<td>76.42</td>
<td>96.76</td>
<td>97.99</td>
<td>79.81</td>
<td>60.44</td>
<td>59.53</td>
</tr>
<tr>
<td>DANN</td>
<td>85.97</td>
<td>96.87</td>
<td>97.94</td>
<td>84.12</td>
<td>67.63</td>
<td>66.78</td>
</tr>
<tr>
<td>DANN-CA</td>
<td>91.35</td>
<td>98.24</td>
<td>99.48</td>
<td>89.94</td>
<td>98.70</td>
<td>98.00</td>
</tr>
</tbody>
</table>

5.2. Summary Results

We report the classification accuracy on the surveillance test set in Tables 2 to 4. Noticing a huge accuracy drop on night images, we also report accuracy of individual day and night sets. We present t-SNE [57] plots of web (blue), day (red) and night (green) images in Fig. 4 and Fig. 8.

Firstly, although achieving state-of-the-art accuracy on the web test set (96.4% vs 91.2% [61]), the baseline model trained only on web images suffers from generalization to SV images, resulting in only 54.98% accuracy. Comparing to the performance of the model trained with target domain supervision (98.65%) in Table 4 provides a sense of how different two domains are. While the baseline adaptation model, DANN (M10 in Table 4), achieves only 58.80%, the proposed joint pixel and feature-level adaptation method achieves 84.20%, reducing the error by 64.9% from the baseline M1. While the use of baseline pixel (CycleGAN) and feature-level (DANN) DA methods as in [18] demonstrates moderate improvement (64.82%) over the baseline, this is far below our proposed DA framework. In the following, we present comprehensive studies on the contribution of individual components and their complementarity.

5.3. Analysis on Pixel-level Adaptation

This section contributes to the analysis of our pixel-level DA on dealing with perspective and photometric transformations, typical factors of variation introduced in SV domain.

Perspective Transformation with CycleGAN [65].

The success of CycleGAN on image translation is attributed by few factors, such as cycle consistency loss, patch-based discriminator, or generator with skip connection. However, these constraints may be too strong to translate viewpoint. As is evident from Fig. 6, the output of CycleGAN (second row) maintains the geometric structure of the input (first row) faithfully but fails at adapting to the viewpoint of SV domain. Relaxing constraints, such as removing skip connections of generator and increasing receptive field size of patch-based discriminator, allows perspective adaptation possible (third row), but we lose many details crucial for recognition tasks.

Our approach solves the challenge by translating images
When combined with perspective transformation (M8) and CycleGAN, web to SV domain as in Fig. 5(b). The conclusion from our visual investigation aligns with the recognition performance, where combined perspective transformation and CycleGAN (M8) achieves 71.21%, which improves upon a model without perspective transformation (M6, 64.32%) in Table 3 or a model without CycleGAN (M5, 64.30%) in Table 2.

Disentangling Illumination via AC-CGAN.
The AC-CGAN fixes the unimodal translation nature of CycleGAN with a latent code [59]. This allows learning disentangled representation from an attribute, which in our case the illumination, and as a result, we can synthesize images of the same car with different illumination conditions, as in Fig. 5(b). Moreover, the continuous interpolation of latent code allows to generate continuous change in illumination factor (e.g., color tone, pixel intensity of headlight) without changing the shape and appearance of each car, as in Fig. 7.

Generating images with diverse illumination conditions improves the recognition accuracy as in Table 3, especially on the night images of SV domain. The AC-CGAN (M7) improves by 2.98% upon the CycleGAN (M6). Moreover, when combined with perspective transformation (M8 and M9), we observe a larger increase in improvement of 8.50%.

Comparison between AFNet and KFNet.
KFNet is developed to improve the generalization of AFNet to real images. Before comparing these models on them, we evaluate KFNet on rendered images from 3D CAD models to demonstrate comparable performance to AFNet. We show inputs, output targets and transformed images by AFNet and KFNet in Fig. 5(a). We observe reliable estimation of appearance flow by KFNet. Furthermore, we obtain 0.072 per-pixel L1 reconstruction error between rendered output images and perspective transformed images at four elevations (0° to 30°) using KFNet, which is comparable to 0.071 error of AFNet (pixel values are normalized to [0, 1]).

Now, we show results on real images in Fig. 5(b). AFNet struggles to generalize on real images and generates distorted images with incorrect target elevation. Although sparse, 2D keypoints are more robust to domain shift from synthetic to real and are sufficient to preserve the object geometry and correctly transform to the target perspective. Finally, better recognition performance on SV domain of the network trained with source and the perspective transformed images (59.73%→61.55% from M3 to M4 in Table 2) implies the superiority of the proposed KFNet.

5.4. Analysis on Feature-level Adaptation
We demonstrate the superiority of the proposed DANN-CA to the DANN on car recognition and other UDA tasks.

Evaluation on Car Recognition in SV Domain.
Note that, on top of 512-dim features, the linear classifier (512−431/432) is used for both models, while we use the 3-layer MLP (512→320→320→1) for the discriminator of DANN after trying several discriminator architectures with
different depths. As in Table 4, the improvement of DANN-CA is larger than that of DANN, confirming the superiority of the proposed method. We further investigate the behavior of these methods from training curves in Fig. 4. The DANN starts to drop significantly after few epochs of adversarial training, remaining with a few collapsed modes in the end. While it shows some fluctuations at the beginning of training, DANN-CA shows clear progression over training and finally reaches at convergence.

Evaluation on UDA Benchmarks.
We also evaluate the performance of DANN and our DANN-CA on UDA benchmarks. For digits and traffic signs tasks, we use data augmentation as in [16]. Due to space constraint, we provide more details on experimental setting and comparison to other methods in Section S5. As we see in the summary results of Table 5 and 6, our proposed DANN-CA outperforms the DANN on all tasks and sometimes by a huge margin. We remind that the only difference between the two methods is the parameterization of the classifier and discriminator, and it clearly shows the importance of joint parameterization in adversarial domain adaptation.

5.5. Analysis on Joint PnF Adaptation
Finally, we provide an empirical analysis on the proposed joint pixel and feature-level (PnF) adaptation. In the joint framework, we train models with feature-level adaptation methods using unlabeled target domain and expanded labeled source domain including original source images and synthesized images by pixel-level DA.

Improved Domain Alignment with Feature-level DA. While it allows high-fidelity generation, constraints in the pixel-level DA make it hard to faithfully adapt to the target domain. It is evident from Fig. 8 where t-SNE plot of M9 is less clean than that of M11. This implies that the role of feature-level DA in joint DA framework is to learn remaining factors not yet discovered by the pixel-level DA.

Improved Training Stability with Pixel-level DA. We delve deeper into understanding the interplay between pixel and feature-level DAs. Fig. 8 shows accuracy curves of pixel-level (M9), feature-level (M11) and joint (M14) DA models on day (dotted) and night (solid) of SV validation sets. While the accuracy on days are stable for all models, we observe a large up-and-down for curve on nights of M11. Note that the fluctuation in the night curve of M9 is not as significant. This is due to many constraints (e.g., warp-based viewpoint synthesis, cycle-consistency or UNet architecture) imposed on the training of pixel-level DA, allowing high-fidelity translation of perspective and illumination variations whose outputs are closer to the target domain than the source examples. Consequently, M14 shows significantly less fluctuation during the training than M11.

We further study the training stability from the mode coverage perspective. Assuming modes correspond to classes in the feature space, the number of classes that are not assigned as top-1 prediction by any of SV test set images is used as a proxy to mode coverage. We provide results in Table 7. While M11 has 29.6 classes on average over 5 runs with no assigned SV image, only 2 classes are missing for M9. The pixel-level DA effectively complements the mode collapse of adversarial learning in the feature-level DA, reducing the number of missing modes to 10.4 for M14.

Complementarity of Components. To summarize, each module has its own disadvantage, such as training instability for feature-level DA and the lack of adaptation flexibility for pixel-level DA. Our empirical analysis suggests that these shortages can be complemented when combined in a unified framework, improving the accuracy by 4.49% and 8.37% upon individual modules, respectively.

6. Conclusion
With an observation that certain adaptation challenges are better handled in feature space and others in pixel space, we propose a joint UDA framework by leveraging complementary tools that are better-suited for each type of adaptation challenge. Importance and complementarity of each component are demonstrated through extensive experiments on a novel application of car recognition in surveillance domain. We also demonstrate state-of-the-art performance on UDA benchmarks with our proposed feature-level DA methods.

Acknowledgement
We thank Paul Vernaza, Gaurav Sharma, Wongun Choi, and Wenling Shang for helpful discussion.
References


