Learning from Auxiliary Supervision

a dissertation presented by

Ishan Nigam

to

The Robotics Institute

in partial fulfillment of the requirements
for the degree of

Master of Science in Robotics

Carnegie Mellon University
May 2018

Thesis advisor: Deva Ramanan Ishan Nigam

Learning from Auxiliary Supervision

Abstract

Supervised learning for high-level vision tasks has advanced significantly over the last decade. One of the primary driving forces for these improvements has been the availability of vast amounts of labeled data. However, annotating data is an expensive and time-consuming process. For example, densely segmenting a natural scene image takes approximately 30 minutes. This mode of supervised learning becomes a hurdle as we generalize to new tasks. In this thesis, we explore a few techniques to improve learning by using alternate modes of supervision. First, we explore how richly annotated ground view segmentation benchmarks may be used to improve the performance of aerial semantic segmentation. Next, we explore how image retrieval may be performed by learning universal representations that generalize well to new tasks. Lastly, we propose to model user behavior as implicit supervision for discovering the latent factors of variation in data to improve image retrieval. Our research suggests that, apart from improving learning algorithms and collecting more data, it is possible to learn better representations from alternative modes of supervision.

Contents

1 Introduction **1**

2 Ensemble Knowledge Transfer for Semantic Segmentation **3**

2.1 Abstract 4

2.2 Introduction 4

2.3 Related Work 8

2.4 AeroScapes Semantic Segmentation Dataset 10

2.5 Ensemble Knowledge Transfer 12

2.6 Experimental Analysis 17

2.7 Conclusion 21

3 Learning Universal Embeddings from Attributes **23**

3.1 Abstract 24

3.2 Introduction 24

3.3 Related Work 27

3.4 Methodology 29

3.5 Experiments 36

3.6 Conclusion 41

4 Discovering Latent Factors of Variation for Context-based Image Understanding **44**

4.1 Abstract 45

4.2 Introduction 45
4.3 Related Work 48
4.4 Methodology 50
4.5 Experiments 57
4.6 Conclusion 62

References **73**

Listing of figures

2.2.1 Contemporary recognition systems make use of multi-*target* knowledge transfer (**top**), where knowledge from a single source domain is transferred to multiple target domains. We explore multi-*source* knowledge transfer (**bottom**), where knowledge from multiple source domains is transferred to a single target domain. We propose an ensemble progressively fine-tuned via diverse source domains. We explore such issues through the illustrative task of semantic segmentation on aerial drone images and introduce *AeroScapes* - the aerial counterpart to autonomous vehicle segmentation benchmarks. 6

2.3.1 The AeroScapes Semantic Segmentation Dataset captures aerial outdoor scenes using a drone. The dataset comprises of 3269 images and ground truth segmentation maps for both *stuff* and *thing* categories. 8

2.4.1 Pixel distribution of the AeroScapes Dataset. The distribution is dominated by *stuff* classes. *Thing* classes constitute 1.51% of the pixel distribution. 11

2.5.1 Appearance of person class: **(a)** ILSVRC [59], **(b)** ADE20k [18], and **(c)** AeroScapes. While ILSVRC comprises of a million images representing a thousand classes, the visual appearance of the classes may be extremely different from scene parsing benchmarks such as ADE20k and AeroScapes. A network trained on ILSVRC will likely not associate the representations it learns for the person class with those in AeroScapes. However, ADE20k appears to be visually similar to AeroScapes. 13

2.5.2 Similarity in visual structure and symmetry may occur in absence of semantic similarity - **(a)** a potted plant in PASCAL VOC appears visually similar to vegetation in Aeroscapes, and **(b)** a shower in ADE20k appears similar to an outdoor streetlight in AeroScapes. While potted plants are expected to be visually similar to vegetation, it is surprising to observe structural similarity between indoor showers and outdoor street lights. Since we only transfer class agnostic knowledge, qualitiative similarities may translate into improvements in quantitative performance. . 15

2.5.3 The AeroScapes dataset is used to finetune the higher (task-specific) layers of convolutional networks trained on independent source domains. Lower layers in the networks are not modified to preserve complementary information derived from diverse source domains. Finetuned representations are concatenated and a regressor is learned on the combined representations for the final prediction. 16

2.6.1 Comparison of methods. FCN 4-stride (**Imagenet-4s**) and FCN 8-stride (**ImageNet-8s**) networks are pre-trained ImageNet baselines. Single-source models are pre-trained on PASCAL Context (**PASCAL**), Cityscapes (**Cityscapes**), and ADE20k (**ADE20k**). Ensembles are created by several different strategies: winner-take-all **Ensemble-Winner**, sparse-selection **Ensemble-Select**, simple averaging **Ensemble-Average**, and weighted averaging **Ensemble-Linear**. All models are FCN 8-stride networks, unless otherwise mentioned. The legend indicates mean IOU for each method. Please see text for more discussion. 19

2.6.2 Each row shows an image, ground-truth label, proposed model (**Ensemble-Linear**), and the best single-source model (**PASCAL**). Row 1: the ensemble segments human, but the single-source model fails. Row 2: the ensemble segments both the humans and part of the obstacle, but single-source model does not. Row 3: single-source model does not detect the drone but proposed model segments it. 20

2.6.3 Comparisons of single-source models and multi-source models. Each pair of bars represent a single source domain, where the first bar is a model obtained with standard fine-tuning, and the second bar is obtained with an ensemble of models learned from that single source domain. While ensembling within a single-domain helps, ensembling across multiple source domains provides a considerable 4% boost. 22

3.2.1 Learning a universal embedding space that maps to various subspaces with different notions of similarity in attributes. Different attributes often require opposite invariances (for example, "young" vs. "male"), which can be captured by different subspaces but not by a single embedding. The joint learning of a universal embedding and different subspaces would encourage automatic feature sharing and disentangling in the universal space, leading to reduced feature redundancy and boosted generalization as well. 26
3.3.1 The network architecture to learn the Weighted Triplet-induced Adversarial Universal Embedding (WT-AUE). The mini-batch contains one (or several) positive pair for each of all N_a attributes (denoted by different colors). Then the exhaustively sampled triplet $t = (i, j, k)$ passes through a CNN and adversarial network to obtain the perturbed universal embeddings for robustness. The universal embeddings are finally mapped to different attribute-specific subspaces, where triplet t can have contradictory notions of similarity. The weighted triplet loss is proposed to aggregate the triplet losses in all subspaces, which simultaneously encourages hard triplet mining and feature interactions across concepts. 30
3.4.1 Adversarial perturbations of the universal embeddings of a triplet $t = (i, j, k)$, and the resulting perturbations in two example subspaces of "heel height" and "gender" of shoes. The triplet is made harder with margin-violating examples (color denotes class label) in each subspace, which improves the embedding quality and convergence speed. 43

4.2.1 Learning a universal embedding space that maps to various subspaces with different notions of similarity in attributes. Different attributes often require opposite invariances (for example, "young" vs. "male"), which can be captured by different subspaces but not by a single embedding. The joint learning of a universal embedding and different subspaces would encourage automatic feature sharing and disentangling in the universal space, leading to reduced feature redundancy and boosted generalization as well. 47

4.3.1 The network architecture to learn the Weighted Triplet-induced Adversarial Universal Embedding (WT-AUE). The mini-batch contains one (or several) positive pair for each of all N_a attributes (denoted by different colors). Then the exhaustively sampled triplet $t = (i, j, k)$ passes through a CNN and adversarial network to obtain the perturbed universal embeddings for robustness. The universal embeddings are finally mapped to different attribute-specific subspaces, where triplet t can have contradictory notions of similarity. The weighted triplet loss is proposed to aggregate the triplet losses in all subspaces, which simultaneously encourages hard triplet mining and feature interactions across concepts. 51

4.4.1 Adversarial perturbations of the universal embeddings of a triplet $t = (i, j, k)$, and the resulting perturbations in two example subspaces of "heel height" and "gender" of shoes. The triplet is made harder with margin-violating examples (color denotes class label) in each subspace, which improves the embedding quality and convergence speed. 64

This is the dedication.

Acknowledgments

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Morbi commodo, ipsum sed pharetra gravida, orci magna rhoncus neque, id pulvinar odio lorem non turpis. Nullam sit amet enim. Suspendisse id velit vitae ligula volutpat condimentum. Aliquam erat volutpat. Sed quis velit. Nulla facilisi. Nulla libero. Vivamus pharetra posuere sapien. Nam consectetuer. Sed aliquam, nunc eget euismod ullamcorper, lectus nunc ullamcorper orci, fermentum bibendum enim nibh eget ipsum. Donec porttitor ligula eu dolor. Maecenas vitae nulla consequat libero cursus venenatis. Nam magna enim, accumsan eu, blandit sed, blandit a, eros.

1

Introduction

There's something to be said for having a good opening line. Morbi commodo, ipsum sed pharetra gravida, orci $x = 1/\alpha$ magna rhoncus neque, id pulvinar odio lorem non turpis [17, 34]. Nullam sit amet enim. Suspendisse id velit vitae ligula volutpat condimentum. Aliquam erat volutpat. Sed quis velit. Nulla facilisi. Nulla libero. Vivamus pharetra posuere sapien. Nam consectetuer. Sed aliquam, nunc eget euismod ullamcorper, lectus nunc ullamcorper orci, fermentum bibendum enim nibh eget ipsum. Donec porttitor ligula eu dolor. Maecenas vitae nulla consequat libero cursus venenatis. Nam magna enim, accumsan eu, blandit sed, blandit a, eros.

$$\zeta = \frac{1039}{\pi}$$

Suspendisse vestibulum dignissim quam. Integer vel augue. Phasellus nulla purus, interdum ac, venenatis non, varius rutrum, leo. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Duis a eros. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Fusce magna mi, porttitor quis, convallis eget, sodales ac, urna. Phasellus luctus venenatis magna. Vivamus eget lacus. Nunc tincidunt convallis tortor. Duis eros mi, dictum vel, fringilla sit amet, fermentum id, sem. Phasellus nunc enim, faucibus ut, laoreet in, consequat id, metus. Vivamus dignissim. Cras lobortis tempor velit. Phasellus nec diam ac nisl lacinia tristique. Nullam nec metus id mi dictum dignissim. Nullam quis wisi non sem lobortis condimentum. Phasellus pulvinar, nulla non aliquam eleifend, tortor wisi scelerisque felis, in sollicitudin arcu ante lacinia leo.

2

Ensemble Knowledge Transfer for Semantic Segmentation

Abstract

Semantic segmentation networks are usually learned in a strictly supervised manner, i.e., they are trained and tested on similar data distributions. Performance drops drastically in the presence of domain shifts. In this paper, we explore methods for learning across train and test distributions that dramatically differ in scene structure, viewpoints, and objects statistics. Motivated by the proliferation of aerial drone robotics, we consider the target task of semantic segmentation from aerial viewpoints. Inspired by the impact of Cityscapes [12], we introduce AeroScapes, a new dataset of 3269 images of aerial scenes (captured with a fleet of drones) annotated with dense semantic segmentations. Our dataset differs from existing segmentation datasets (that focus on ground-view or indoor-scene domains) in terms of viewpoint, scene composition, and object scales. We propose a simple but effective approach for transferring knowledge from such diverse domains (for which considerable annotated training data exists) to our target task. To do so, we train multiple models for aerial segmentation via progressive fine-tuning through each source domain. We then treat these collections of models as an ensemble that can be aggregated to significantly improve performance. We demonstrate large absolute improvements (8.12%) over widely-used standard baselines.

Introduction

Pixel-level semantic segmentation of natural scenes is a fundamental visual recognition task. Recent history has shown significant progress on standard segmentation benchmarks, e.g., PASCAL VOC and Microsoft COCO [18, 41]. This success is largely owed to convolutional networks [9, 40, 81]. The community has also explored segmentation tasks that incorporate both *stuff* (amorphous background regions like grass and sky) and *things* (objects like car and person) categories [12, 84]. Other applications are found in domains such as biomedical imaging [1, 11, 65], and satellite imaging [30, 32, 46]. In

particular, autonomous driving has witnessed significant development [3, 54, 67] together with an increasing number of available benchmarks [12, 49, 57].

Segmentation benchmarks: Classic semantic segmentation benchmarks have focused on general scenes, including indoor and outdoor settings [18, 41, 48, 84]. Spurred by the introduction of novel sensors, many segmentation benchmarks have focused on limited viewpoints of specialized scenes such as ground-views of urban environments (for autonomous vehicles) [12, 48, 84], and direct overhead views (for orbital satellites) [32, 47, 56]. However, recent advances in aerial robotics allow for significantly more ease in capturing diverse viewpoints and scenes. These represent a considerable departure in statistics compared to previously-studied domains, which is the focus of our work.

Domain shift: Most deep segmentation models are deliberately trained and tested on similar data domains to attain high accuracy. Drastic performance drop is often observed in the presence of domain shifts. Indeed, domain shifts across dataset distributions pose a major challenge for learning good representations that can generalize well to all domains. Interestingly, another perspective is that *multi-source learning* of representations from such diverse source domains may, in fact, *help* generalization because each domain provides complementary information for the target task. In our work, we introduce a simple approach for transferring appropriate information from a diverse set of source domains for a particular target task.

Knowledge transfer: We turn to transfer learning techniques that allow us to transfer knowledge from existing domains (for which ample annotated data exists) to the aerial setting (for which limited annotated data exists). While transfer learning from a source to target task is a well studied problem [51, 75], by far the most common approach is fine-tuning a model pre-trained on the source task [24]. Indeed, virtually *every* contemporary visual recognition system transfers knowledge from ImageNet [59] to the

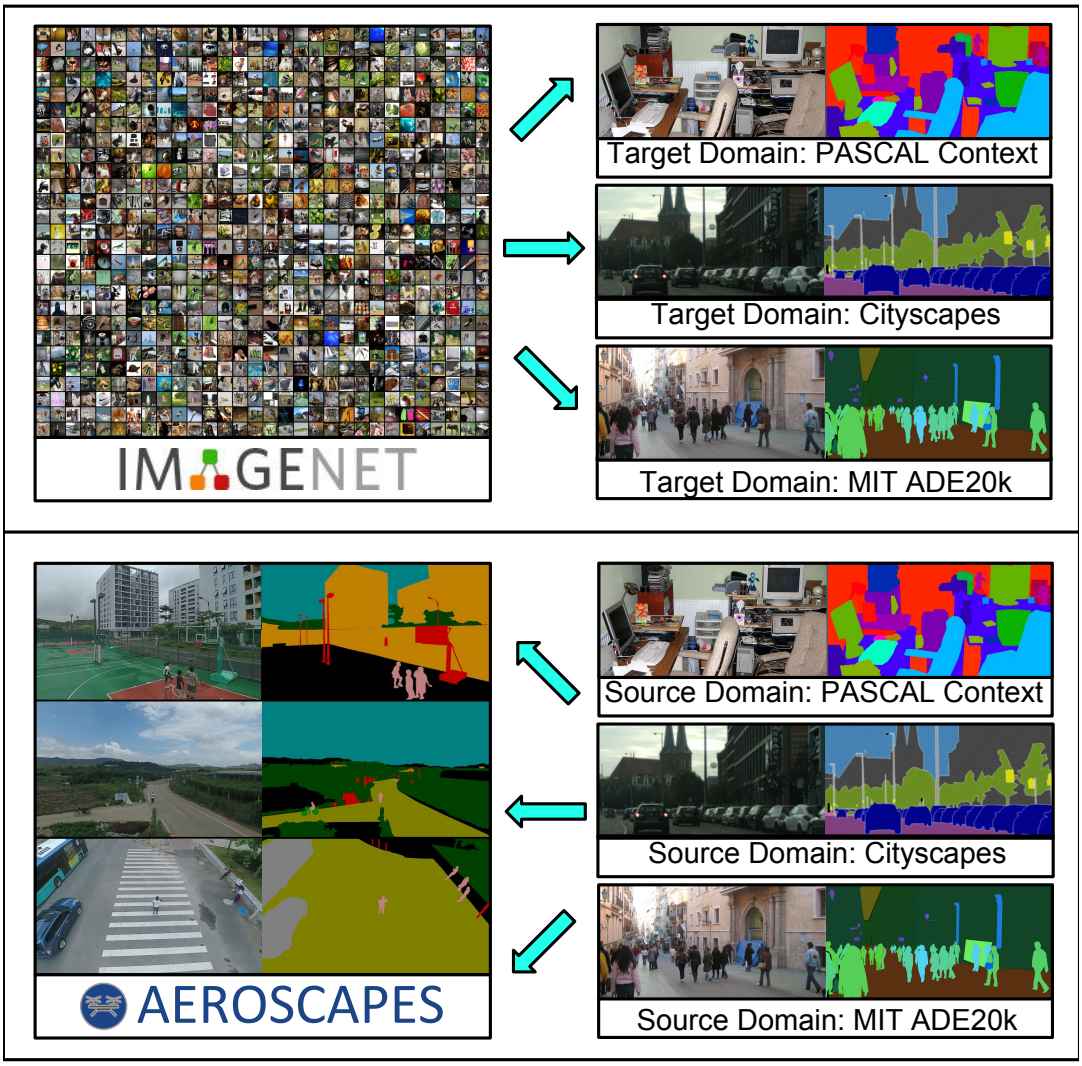

Figure 2.2.1: Contemporary recognition systems make use of multi-*target* knowledge transfer (**top**), where knowledge from a single source domain is transferred to multiple target domains. We explore multi-*source* knowledge transfer (**bottom**), where knowledge from multiple source domains is transferred to a single target domain. We propose an ensemble progressively fine-tuned via diverse source domains. We explore such issues through the illustrative task of semantic segmentation on aerial drone images and introduce *AeroScapes* - the aerial counterpart to autonomous vehicle segmentation benchmarks.

target task of interest. We use this methodology to produce a ully-convolutional network (FCN) as an initial baseline, by fine-tuning on a modest set of aerial training images (e.g., ImageNet → AeroScapes). However, we would like to transfer knowledge from *multiple* domains, including indoor scenes and ground-view images of urban environments (see Fig. 2.2.1). Such source domains with richly annotated datasets represent a rich knowledge source that we would like to exploit. But the precise *manner* in which this knowledge should be transferred can be distinct and subtle - some indoor objects (such as people) can appear outdoors, and perhaps some outdoor objects look similar under aerial viewpoints (such as bicycles and motorcyles).
Ensemble transfer: Our key insight is to combine knowledge from multiple sources by learning an *ensemble* of models that are trained with *progressive fine-tuning* (ImageNet → PASCAL → AeroScapes, ImageNet → Cityscapes → AeroScapes, etc.). Intuitively, each model in the ensemble makes use of different source knowledge and so will likely make different errors (e.g., PASCAL models may be more accurate on people because they occur often in PASCAL, while Cityscapes models may be more accurate on vehicles). We then optimally combine these ensembles so as to obtain a final prediction. Our ensemble model improves over strong baselines by 8.12%. In summary, the contributions of this research is as follows:

- We propose a novel architecture-agnostic method to transfer knowledge present in diverse data sources, as encoded by richly-labeled source datasets tailored for domains *other* than the target domain of interest.
- We release the AeroScapes aerial semantic segmentation dataset, captured to study transferability of knowledge from multiple segmentation benchmarks.
- We experimentally validate our proposed benchmark using Fully Convolutional Networks, and report significant improvements over strong baselines trained with widely-adopted best-practices.

Related Work

Semantic segmentation: Start-of-the-art semantic segmentation methods use the convolutional networks to learn a pixel-to-pixel mapping from the image space to semantic label space [10, 13, 39, 40, 43, 81]. The success of these deep neural networks can be attributed to the availability of a large amount of pixel-level annotations and the ability of deep nets to learn from large data in an end-to-end manner. One of the most successful deep models is the Fully Convolutional Network (FCN) [43] that can directly generate the spatial label map as output.

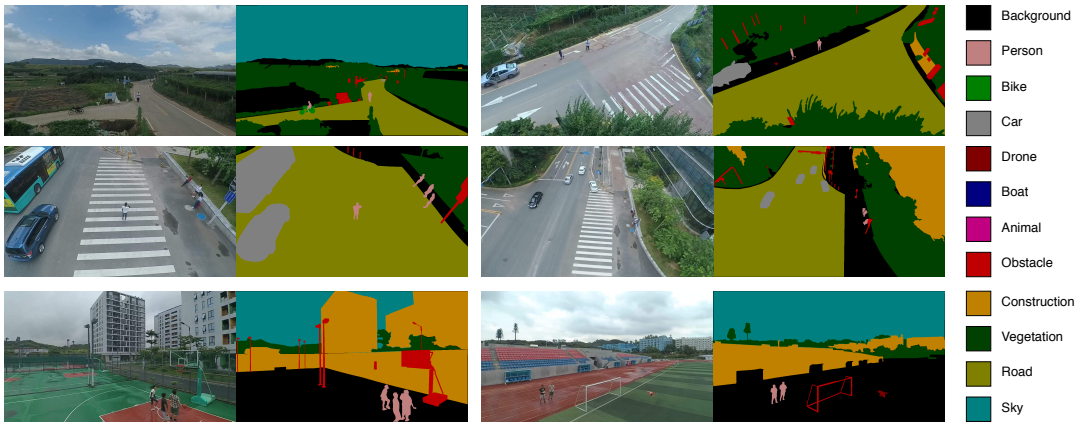

Figure 2.3.1: The AeroScapes Semantic Segmentation Dataset captures aerial outdoor scenes using a drone. The dataset comprises of 3269 images and ground truth segmentation maps for both *stuff* and *thing* categories.

Multi-task learning: Multi-task learning improves model generalization by combining domain-specific information learned through complementary tasks on each domain [7]. These methods usually learn a generalizable representation by learning representations across domains. Inspired by the *multi-task learning* paradigm, we present a *multi-source learning* framework, which learns a representation for a single target domain from multiple source representations. Theoretically, it is possible to learn a single representation

from different domains under a multi-task framework [36]. However, in practice, this requires appropriate weighting among different tasks and a large memory budget to deal with multi-domain data simultaneously. Our proposed multi-source learning framework proves to achieves competitive results in a simple but effective way.

Knowledge Transfer: Pixel-level annotation of semantic categories is a time consuming endeavor. A rich literature employs semi-supervised and weakly-supervised learning methods to aid such tedious labelling efforts, which can be regarded as knowledge transfer in the label space. Weak supervision is generally provided as class-level labels [33], specific point annotations [4], object localizations [72], or saliency mechanisms [29].The authors in [52] develop an Expectation-Maximization framework for image segmentation under both weakly-supervised and semi-supervised settings. Recently, Chaudhry et al. [8] combined the saliency and attention maps to obtain reliable cues to boost segmentation performance and effectively explore knowledge from class labels.

Domain Adaptation: Domain adaptation methods aim to address the gap between the distributions across different data domains [37]. Recent deep learning-based methods align the domain features by maximizing the confusion [19, 20, 68] or explicitly minimizing the distances [44, 45] between their distributions across domain. To our knowledge, [25] is the only deep domain adaptation method applied to semantic segmentation. It involves image domain adversarial training and class distribution alignment, which renders learning difficult. Many domain adaptation methods focus on scenarios where little or no labeled data is available for the target domain. In our case, we have put forth considerable effort to collect and annotate the AeroScapes dataset, and so use the well-established paradigm of fine-tuning to transfer knowledge from multiple source domains to our target AeroScapes domain.

AeroScapes Semantic Segmentation Dataset

Most classical localization benchmarks focus on understanding objects in images, disregarding the setting in which the objects occur. Background elements provide semantic and geometric context for objects in the foreground [5, 48]. For example, an autonomous car may navigate based on roads it identifies in its line of sight, or the path planner may require that the car never attempts to park on sky or water. Thus, it is imperative that terrain-based or aerial autonomous agents are taught to identify both foreground as well as background elements.

The ability to foresee events in the future is a critical attribute of real-time autonomous systems, which rely on scene understanding for decision making. An appropriate test bed for such systems must incorporate labeled image sequences [12, 57]. Agents that rely on visual scene understanding for decision making must also learn to incorporate temporal information into their representations. Thus, it is necessary that evaluation benchmarks for navigation systems incorporate video data.

Aerial robots allow us capture previously unexplored viewpoints and diverse environments. While autonomous cars are constrained to move on the ground, aerial robots have the freedom to navigate in three-dimensions, allowing us to capture visual scales and view-points that are richer and more varied than prior benchmarks. The above constraints motivate us to collect the AeroScapes Dataset [1], which contains images captured from a drone operating at an altitude of 5-50 meters. The segmentation maps associated with these images are labeled with both *stuff* classes - vegetation, roads, sky, construction - and *thing* classes - person, bikes, cars, drones, boats, obstacles, animals (Fig.2.3.1).

The AeroScapes dataset comprises of 3269 images acquired from 141 video sequences, and contains several video sequences that are temporally downsampled. The class distribution in AeroScapes reflects the data imbalance observed in typical outdoor images comprising of both stuff and

[1] AeroScapes Dataset: http://www.github.com/ishann/aeroscapes

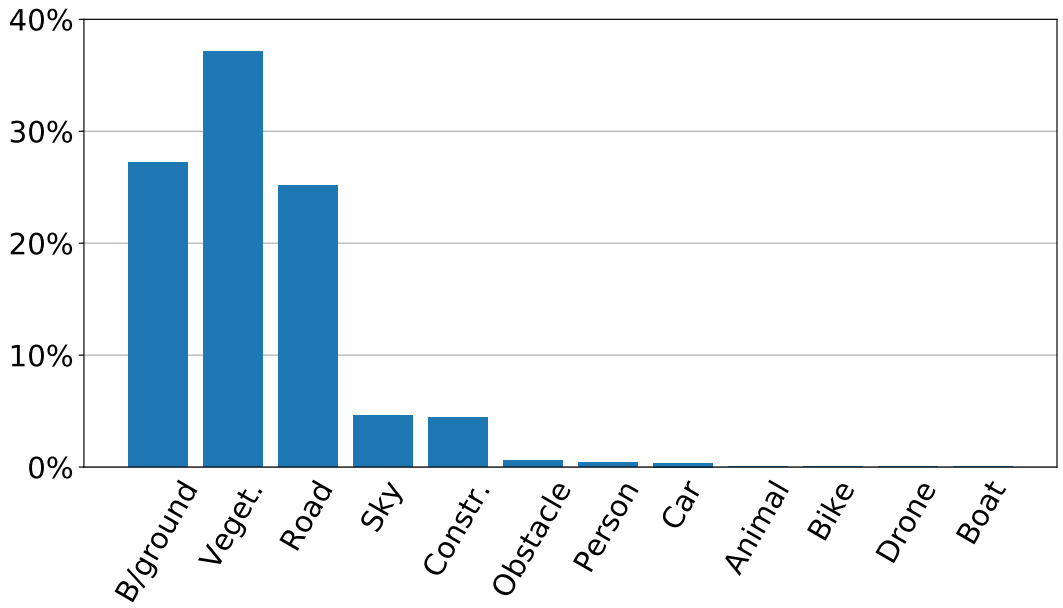

Figure 2.4.1: Pixel distribution of the AeroScapes Dataset. The distribution is dominated by *stuff* classes. *Thing* classes constitute 1.51% of the pixel distribution.

things annotations. The cumulative weight of the things classes is approximately 1.51% of the data (Fig. 2.4.1).

Numbers only tell a partial story (about the statistical distribution) of the dataset. Fig. 2.5.1 shows representative samples for the person class from (a) ILSVRC dataset [59], (b) ADE20k dataset [84], and (c) AeroScapes dataset. A deep convolutional network trained on ILSVRC (source domain) is likely to not associate representations it learns for the person class with those for AeroScapes (target domain). However, ADE20k appears visually similar to AeroScapes for the person class. In Section 2.6, we observe that the visual appearance of object categories affects the performance of the system on the particular class.

Ensemble Knowledge Transfer

Our primary thesis is that the collective set of segmentation benchmarks represents a "meta" knowledge source that can be applied to a related, but different task. Importantly, each source encodes a considerable amount of curated human knowledge, manifested through the images and labels. We propose to *extract* this knowledge by training deep networks on each data source and *transfer* the knowledge to the target domain through fine-tuning. The above procedure yields an ensemble of models, one for each data source, that can be applied to the target domain. We then *aggregate* information across the ensemble. We begin by discussing the intuitions which motivate this approach, particularly for the AeroScapes semantic segmentation setting.

Motivation

Here, we review some types of "knowledge" from our source domains [18, 48, 84] that may transfer to our target (AeroScapes), and challenges associated with transferring each type of knowledge.

Within-class transfer: The most simplest form of transfer may be within-class knowledge transfer. Many of our source domains contain the `people` class, and one might expect additional training examples from such domains are still helpful to train `people` classifiers for our target domain. However, a notable challenge of aerial drone footage is that it is dramatically different distributions of camera viewpoints compared to existing datasets - objects tend to be smaller and viewed from overhead viewpoints (Fig. 2.5.1). Many classes also have inherently different types of appearance in outdoor settings versus indoor ones (e.g., the types of clothing worn by people). Hence one challenge is devising a knowledge transfer procedure that can generalizes across viewpoints, scales, and sub-categorical appearance variations.

Between-class transfer: One might also expect between-class transfer. A potted plant from an indoor scene may resemble a tree in an outdoor scene, and shower heads may appear similar to urban obstacles such as streetlights

Figure 2.5.1: Appearance of person class: **(a)** ILSVRC [59], **(b)** ADE20k [18], and **(c)** AeroScapes. While ILSVRC comprises of a million images representing a thousand classes, the visual appearance of the classes may be extremely different from scene parsing benchmarks such as ADE20k and AeroScapes. A network trained on ILSVRC will likely not associate the representations it learns for the person class with those in AeroScapes. However, ADE20k appears to be visually similar to AeroScapes.

(Fig. 2.5.2a). Here, a central challenge is uncovering what types of object class knowledge can be shared *across* different classes.

Source selection: A final challenge is that different transfer strategies may be appropriate for different sources. For example, `people` in ground-views of urban scenes (e.g., from CityScapes) may transfer better to `people` in Aeroscapes, while indoor images of people (e.g., from ADE20K) should not be applied to aerial footage. This suggests that a remaining challenge is determining *what* kinds of information should be extracted from *which* sources, and finally, how to combine such information from disparate knowledge sources.

Single-Source Knowledge Transfer

We advocate a simple approach to knowledge transfer that addresses the first two motivations (and associated challenges) above: *fine-tune* a model trained on a source domain to the target. This produces a set of classifiers for the target domain, where each one is pretrained on a different source domain. While rather commonplace in today's recognition pipelines, we discuss some

subtle issues that arise when addressing the third challenge (source selection and combination). Given an input image x, we write each classifier for pixel y as follows:

$$p_s(y|x), \quad \text{where} \quad x \in \mathcal{X}_{\text{target}}, \tag{2.1}$$
$$y \in \mathcal{Y}_{\text{target}} = \{1, 2, \ldots T\}, s \in \text{Sources} = \{1, 2, \ldots S\}.$$

We assume classifiers are trained with softmax objective functions that report distributions overall target classes y for each pixel x, for each pre-trained source domain s.

Knowledge diversity: There exists widely-used heuristics and best practices for fine-tuning a pre-trained network on a target dataset. A common heuristic is to run a fixed number of iterations of gradient descent, with an annealed stepsize schedule, until error on validation data (also from the target dataset) no longer decreases. However, in our case, we will combine the knowledge encoded across the S predictors through ensembling. Our intuition is that this aggregation will prove more effective if the predictors encode different types of knowledge and make different types of errors. Hence, we intentionally *limit* the amount of fine-tuning, so as to ensure the classifiers retain more of the knowledge from their source domain. We explored various strategies for doing so, including early stopping and small step sizes. We found that *fixing* all but the upper two layers of the network allowed for sufficient source information to be retained.

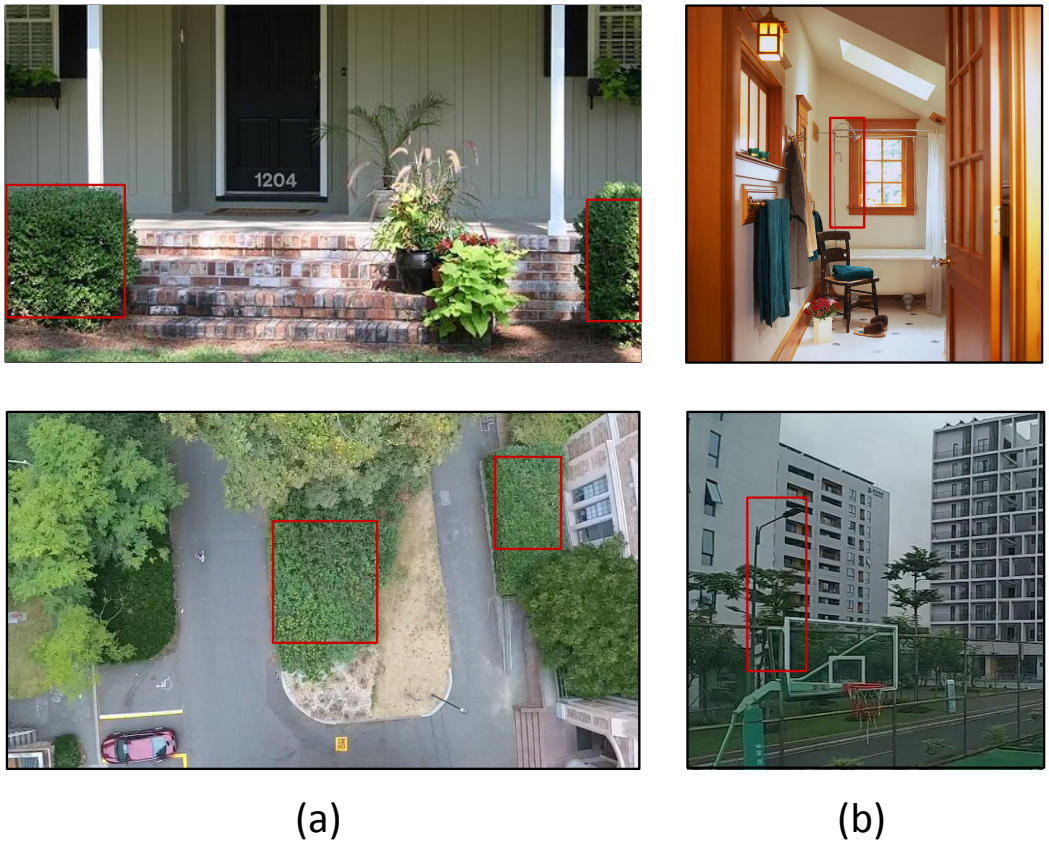

Figure 2.5.2: Similarity in visual structure and symmetry may occur in absence of semantic similarity - **(a)** a potted plant in PASCAL VOC appears visually similar to vegetation in Aeroscapes, and **(b)** a shower in ADE20k appears similar to an outdoor streetlight in AeroScapes. While potted plants are expected to be visually similar to vegetation, it is surprising to observe structural similarity between indoor showers and outdoor street lights. Since we only transfer class agnostic knowledge, qualitiative similarities may translate into improvements in quantitative performance.

Multi-source Knowledge Transfer

We follow a simple procedure for *linearly* integrating knowledge encoded across S classifiers pre-trained on each source domain:

$$p(y = t|x) \propto \sum_{s=1}^{S} w_{st} p_s(y = t|x), \quad \forall t \in \{1 \dots T\} \tag{2.2}$$

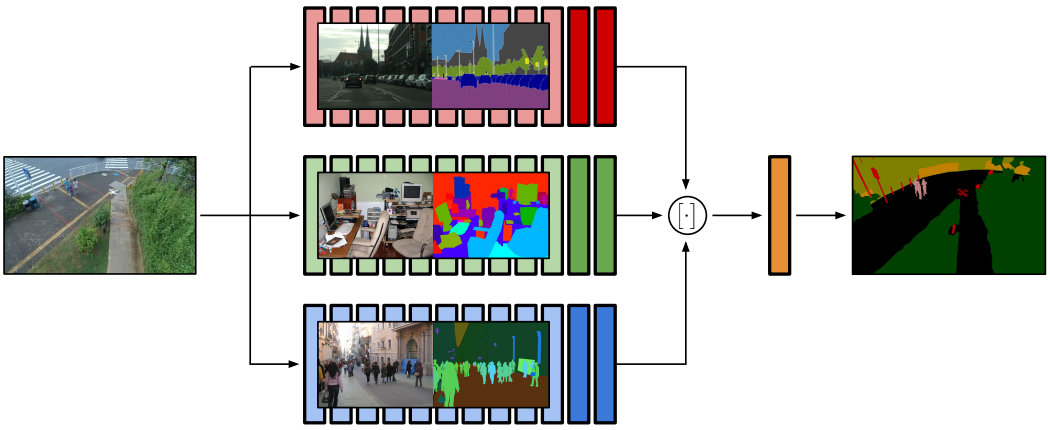

Figure 2.5.3: The AeroScapes dataset is used to finetune the higher (task-specific) layers of convolutional networks trained on independent source domains. Lower layers in the networks are not modified to preserve complementary information derived from diverse source domains. Finetuned representations are concatenated and a regressor is learned on the combined representations for the final prediction.

where w_{st} refers to a $T \times S$ matrix of weights. Note that the righthand-side of the equation must be normalized to produce a proper distribution over class labels, hence the proportionality sign.

We now describe different strategies for producing weights w_{st}. The most straightforward is to train the weights on the same target dataset used for fine-tuning

$$w_{st} \quad \text{trained on} \quad \mathcal{X}_{\text{target}}, \mathcal{Y}_{\text{target}} \quad \textbf{[Ensemble-Linear]} \tag{2.3}$$

This can be implemented as a sparse linear layer with a standard loss-function (e.g., cross-entropy). However, this might result in overfitting since weights are trained with the same data used for fine-tuning. We also explored simpler strategies for defining the weights, including a simple average:

$$w_{st} = 1, \quad \forall st \qquad \textbf{[Ensemble-Average]} \tag{2.4}$$

as well as a sparse selection approach where a *single* source classifier is used

for each target class t:

$$w_{st} = \begin{cases} 1, s = \text{select}(t) \\ 0, \text{otherwise} \end{cases} \qquad \textbf{[Ensemble-Select]} \tag{2.5}$$

The above approach selects a particular source classifier for each target class, allowing one to "mix and match" classifiers from different source domains. A natural strategy for selection is to use the source that performs the best for each target class.

Experimental Analysis

In this section, we explore the proposed ensemble knowledge transfer method for improving the performance of semantic segmentation tasks. The analysis is performed using the Cityscapes [12], PASCAL Context [48], and ADE20k [84] scene parsing segmentation benchmarks serving as the *source* domains and the AeroScapes dataset (Section 2.4) serving as the *target* domain.

We begin with a brief description of the methodology we follow for learning models for the AeroScapes dataset on the independent source domains, and the ensemble knowledge transfer network design for combining these single-source models. These descriptions are accompanied by analyses for the performance of these models. We conclude with analysis which demonstrates that complementary information from diverse source domains improves the performance of the multi-source ensemble.

Implementation Details: A number of architectures have recently been proposed for semantic segmentation [9, 39, 40]. However, we choose to use the simple and effective Fully Convolutional Networks [43](FCNs) for all experiments. We train the deep networks (Sec. 2.6.1) via Stochastic Gradient Descent using a minibatch size of one, $1e-10$ fixed learning rate, 0.99 momentum, and $5e-4$ weight decay. For each source domain, we freeze the first nine convolutional layers of the network and finetune the successive

layers. The AeroScapes Dataset is divided into a $80\% - 20\%$ train-test split. We ensure that image frames from a video sequence are only included in either training or testing. Throughout our experiments, the mean Intersection Over Union (mIOU) metric is used to report segmentation performance. The regression networks (Sec. 2.6.2) are trained with fixed $1e-2$ learning rate, 0.9 momentum, and $5e-4$ weight decay. The Caffe toolbox [28] is used to implement the networks. When learning linear weights for ensembling (2.3), we found that balanced sampling of pixels was needed to ensure that the network is not biased towards *stuff* classes.

Baseline: To obtain a baseline, we first fine-tune the VGG-16 convolutional network [64] pre-trained on Imagenet (ILSVRC) [59] towards AeroScapes. Because AeroScapes contains many small objects, we explored high-resolution FCN models that produced outputs at 2-pixel, 4-pixel, and 8-pixel strides. We found 4-pixel networks were easy to train while producing good results (a mIOU of 48.96% - see Fig. 2.6.1).

Learning from Single Sources

For each source, we search over hyperparameters to find the best settings for fine-tuning from that source. This produces three different AeroScape models that score 52.02%, 51.62%, and 49.55% corresponding to PASCAL Context, ADE20K, and Citiscapes. The class-wise performance for each of these methods is detailed in Fig. 2.6.1.

Analysis: First, we note that the baseline pre-trained ImageNet network exhibits some interesting structure when examined at the class-level. A high-resolution network (4s) performs better than a low resolution network (8s), except for for people and bicycles. We posit that a certain degree of "blurring" by operating at coarser resolutions helps knowledge transfer for such classes. Overall PASCAL produces the best single-source model, with a 3% improvement over the baseline. However, when examining performance per-class, we see that different sources do well for different classes. PASCAL

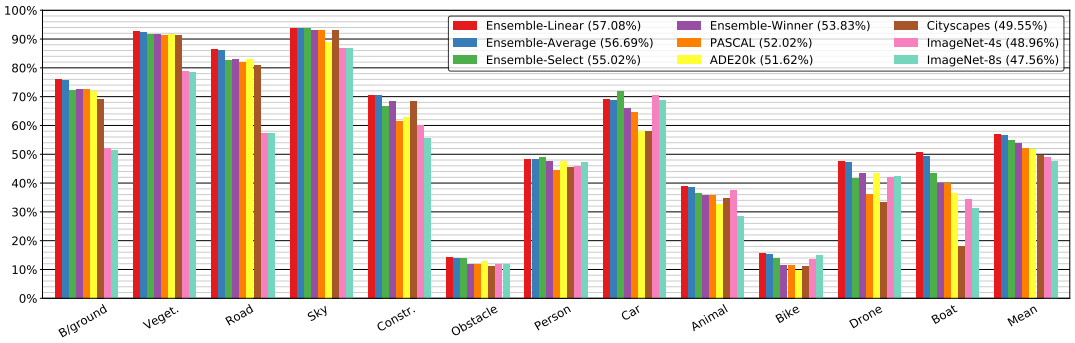

Figure 2.6.1: Comparison of methods. FCN 4-stride (**Imagenet-4s**) and FCN 8-stride (**ImageNet-8s**) networks are pre-trained ImageNet baselines. Single-source models are pre-trained on PASCAL Context (**PASCAL**), Cityscapes (**Cityscapes**), and ADE20k (**ADE20k**). Ensembles are created by several different strategies: winner-take-all **Ensemble-Winner**, sparse-selection **Ensemble-Select**, simple averaging **Ensemble-Average**, and weighted averaging **Ensemble-Linear**. All models are FCN 8-stride networks, unless otherwise mentioned. The legend indicates mean IOU for each method. Please see text for more discussion.

people tend to be large and Citiscapes people tend to be upright pedestrians or drivers, while ADE20K people look most visually similar to those found in AeroScapes (Fig. 2.5.1). Cityscapes performs better for construction but does worse on boats. This may not be all that surprising since there are not boats in urban street scenes! Perhaps surprisingly, all cars, including Cityscapes cars, do worse that the Imagenet baseline. We posit that dramatic viewpoint differences makes Cityscapes cars less helpful than one might think. In summary, these results suggest that combining source-specific classifiers will likely improve performance.

Learning from Multiple Sources

Since certain pre-trained models do better on specific classes, it is natural to explore a *winner-take-all* approach: for each class, select the best performance across all sources, and average across all classes. This strategy improves performance by 1.8% (over the best single-source model) to 53.83% (**Ensemble-Winner** in Fig. 2.6.1). While this suggests that combining

sources is helpful, this is not a realizeable model.

The above thought-experiment can be turned into a realizeable model by applying (2.5) and selecting the class-wise winners from above. Interestingly, this further improves accuracy by 1.2% to 55.02% mIOU (**Ensemble-Select** in Fig. 2.6.1). We posit this improvement comes from competition between the selected classifiers arising from the normalization in (2.2).

However, the above discards potentially useful information from other classes that were not selected. A naive strategy for combining information from all S source classifiers is to average their prediction (2.4). This further improves accuracy by 1.6% to 56.69% (**Ensemble-Average** in Fig. 2.6.1).

Finally, it is naturally to take weighted combination, where weights can be tuned to behave like an average or a selection (or anything inbetween). The resulting linearly-combined ensemble from (2.3) slightly improves performance by 0.4% to 57.08% mIOU (**Ensemble-Linear** in Fig. 2.6.1). We show qualitative results in Fig. 2.6.2. Please see the figure caption for more discussion.

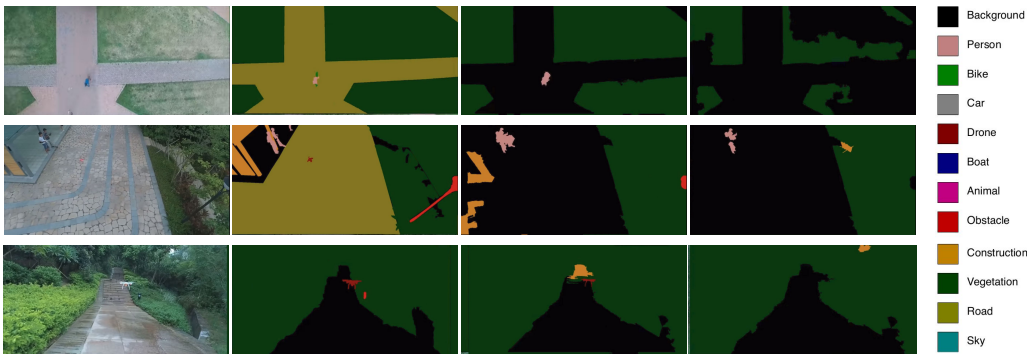

Figure 2.6.2: Each row shows an image, ground-truth label, proposed model (**Ensemble-Linear**), and the best single-source model (**PASCAL**). Row 1: the ensemble segments human, but the single-source model fails. Row 2: the ensemble segments both the humans and part of the obstacle, but single-source model does not. Row 3: single-source model does not detect the drone but proposed model segments it.

Analysis: Overall, **Ensemble-Average** performs surprisingly well, suggesting

that each pre-trained source network naturally contains complementary information. The sole category where **Ensemble-Linear** noticeably outperforms a simple average is `boat`. Here, the Cityscapes single-source model performs quite poorly, likely degrading the average computation.

Single-source ensembles: At this point, it is natural to ask if the improvement from the multi-source ensemble is due to complementary knowledge from *multiple sources* or simply a function of increased capacity due to *ensembling*? To answer this, we train ensemble networks of equivalent capacity on *singular* source domains. Since we wish to examine the effect of homogeneous data training in contrast with heterogenous data training, each network is trained on the entire source domain. Fig. 2.6.3 shows that single-source ensembling helps to an extent. However, single-source ensembles (53.05% mIOU) do not do as well as our proposed multi-source approach (57.08% mIOU).

Conclusion

Fully Convolutional Networks (FCNs) have established state-of-the-art performance on existing semantic segmentation benchmarks. Data-driven methods trained in supervised settings usually suffer from performance drop in the presence of domain shifts. In this research, we explore FCNs for semantic segmentation across data distributions that dramatically differ in scene structure, viewpoints, and objects statistics. We consider semantic segmentation on images with aerial viewpoints and study the transferability of knowledge from ground-view segmentation benchmarks. To this end, we prepare and release the AeroScapes dataset - a collection of 3269 aerial images (and associated semantic segmentation maps) captured using a fleet of drones.

We train multiple models for aerial segmentation via progressive fine-tuning from multiple source domains. The precise knowledge to be transferred from each domain is distinct and subtle - indoor objects can appear outdoors and

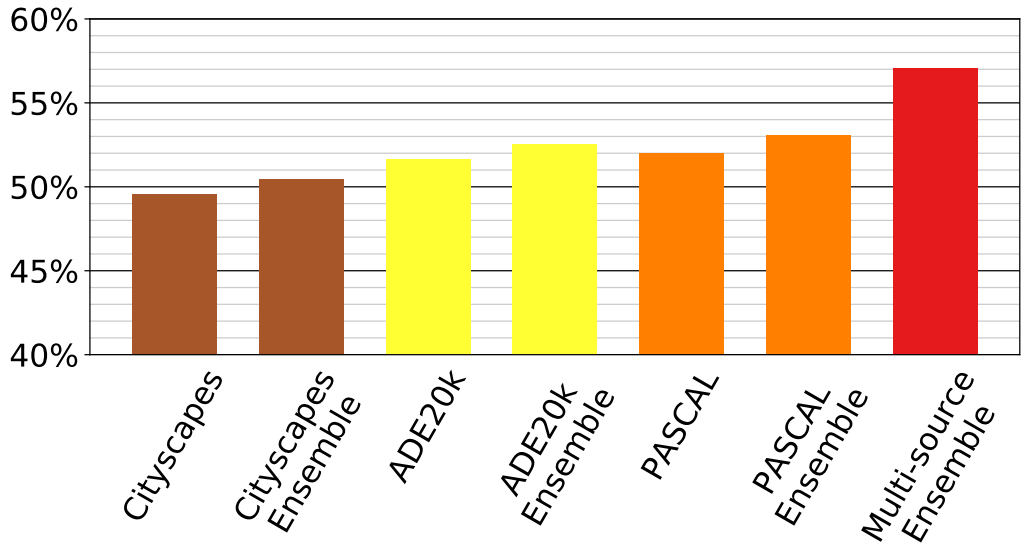

Figure 2.6.3: Comparisons of single-source models and multi-source models. Each pair of bars represent a single source domain, where the first bar is a model obtained with standard fine-tuning, and the second bar is obtained with an ensemble of models learned from that single source domain. While ensembling within a single-domain helps, ensembling across multiple source domains provides a considerable 4% boost.

outdoor objects may appear to be similar under aerial viewpoints. Thus, we treat the models tuned from different domains as an ensemble and aggregate them to improve performance. We successfully learn important components from each source domain through a regression network, resulting in an overall improvement of 8.12% compared to a standard baseline.

The proposed framework is agnostic of the underlying network architecture and allows us to leverage small segmentation datasets that may comprise of critical complementary information. As future work, network finetuning and ensembling may be collaboratively learned to leverage information from diverse data sources.

3

Learning Universal Embeddings from Attributes

Abstract

We address the problem of learning a universal embedding space from which different semantic concepts or notions of similarity can originate. Contemporary attribute datasets provide rich multi-label annotations to achieve this goal.Universal embeddings learned from the multi-label attributes would naturally encourage feature sharing, leading to reduced feature redundancy and boosted generalization ability. This paper presents a multi-task framework to learn universal embeddings by mapping them to different subspaces, each corresponds to one particular attribute and is supervised by the triplet similarity. A weighted triplet loss is proposed to combine all the possible triplets for all attributes in a small batch, with weights designed to favor triplets with hard examples by computing their attribute set similarity. This not only eliminates the need for hard triplet mining in batch, and makes the most of available data in a simple way, but also helps to learn the structure of the private and shared features in universal space to encode different triplet similarities across attributes. The learning is made more robust by competing against an adversarial network that perturbs the universal embeddings to increase the loss. The resulting Weighted Triplet-induced Adversarial Universal Embedding (WT-AUE) achieves strong results for attribute prediction, low-shot generalization as well as off-task recognition.

Introduction

Deep networks have revolutionized visual understanding through the ability to learn data-driven representations. Such representations are typically trained for a particular task with massive amounts of labeled data [14].

Learning embeddings: A particularly influential variant of representation learning is the problem of learning embedding spaces, which transform input images into a finite-dimensional vector space where images with similar labels

are closeby according to some distance [61]. Directly learning embeddings has many advantages. Firstly, they are not be tied to category labels, and so can be trained on pairs (or triples) of examples with same/different tags [22]. Secondly, they can be used for zero-shot or one-shot learning by applying the learned distance functions to novel class exemplars [62]. However, embeddings do have limitations. They assume that a *single* universal notion of distance always holds. But this is not always true. Consider the problem of learning an embedding of faces that is *attribute-specific* (Fig. 4.2.1). A gender-specific embedding will likely need to be invariant to the age attribute, while an age-specific embedding will likely need to be invariant to gender. This suggests it may be hopeless to learn a single "one-size-fits-all" embedding, since an embedding that is invariant to both age and gender seems useless for either task.

Multi-task learning: To reconcile the apparent contradiction above, we appeal to the large literature on multi-task learning [6]. Recent work has shown that a shared set of nonlinear features can be jointly learned to serve a variety of different tasks. Examples include joint learning for facial landmark detection, head pose and facial attribute prediction [80], training a universal Convolutional Neural Network (CNN) for low-, mid-, and high-level vision tasks [35], etc. Multi-task networks have also been shown to transfer better to novel tasks because the shared features appear to be more generic [53]. From our perspective, the shared feature representation can be thought as as an embedding that is *linearly projected* into (one or more) one-dimensional spaces representing classifiers for one or more tasks.

Multi-task embeddings: Our formulation can be seen as the natural integration of multi-task learning and nonlinear embeddings: we learn a single universal embedding space that is *linearly projected* into attribute-specific subspaces. Our overall philosophy is similar to multi-view embeddings [2] and the recent work of conditional similarity networks (CSNs) [69]. However, there

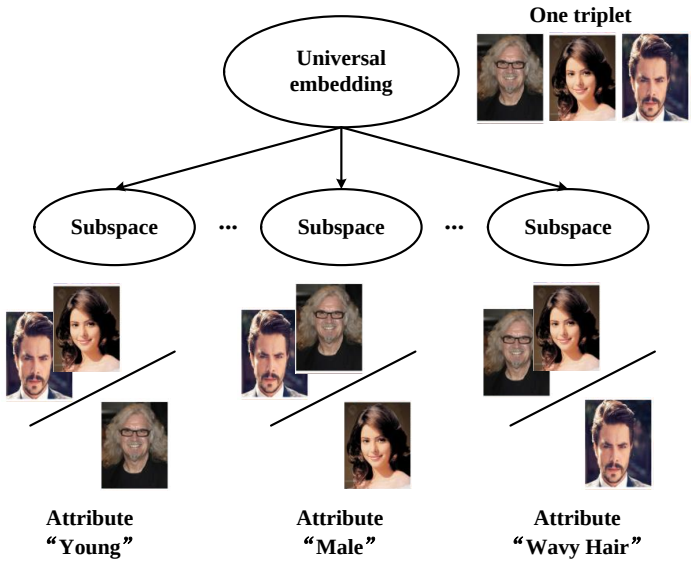

Figure 3.2.1: Learning a universal embedding space that maps to various subspaces with different notions of similarity in attributes. Different attributes often require opposite invariances (for example, "young" vs. "male"), which can be captured by different subspaces but not by a single embedding. The joint learning of a universal embedding and different subspaces would encourage automatic feature sharing and disentangling in the universal space, leading to reduced feature redundancy and boosted generalization as well.

are several important differences. The former learns separate embeddings without any multi-task sharing, while the latter learns a single universal embedding that is element-wise masked out to produce different embeddings capturing different notions of similarity. In some sense, CSNs learn axis-aligned embedding subspaces, while we learn arbitrary linear projections (consistent with common best-practices in multi-task learning). We demonstrate that our universal embedding space can be used for few-shot learning of attribute-specific embeddings. This has the practical advantage that data (e.g., faces) can be stored using a fixed universal embedding representation that will likely apply to *any* future attribute of interest.

Training: We also contribute several algorithmic innovations to the problem of learning embeddings specific to our multi-task setting. As in much

recent work, we learn with triples of examples, but use multiple attribute labels to speed up convergence and increase accuracy in three distinct ways: (1) When learning on mini-batches, we project each triplet to every attribute-specific subspace, effectively learning from *multiple* supervisory signals per batch. (2) Unlike the sampling techniques discussed in recent work [74], we exhaustively use all triplets in batch. We also find that *softly weighting* triplets improves convergence. We use multi-view attribute labels to define a notion of semantic similarity between examples, which is in turn used to efficiently construct a meaningful weighting. The weights particularly favor those hard examples, acting as a soft tool of hard triplet mining. (3) We introduce an *adversarial* network that perturbs the universal embeddings so as to generate harder triplets in each subspace.

Related Work

Multi-Task Embedding Learning

Our universal embedding is learned in a multi-task framework. Example multi-task learning methods learn CNN features for joint facial landmark detection, head pose and facial attribute prediction [80], or for low-, mid-, and high-level vision tasks [35]. Another line of multi-task embedding methods learn from multi-label attribute annotations. For example, [55] proposed a higher-order Boltzmann machine to disentangle attributes from a manifold with multiplicative interactions. Others use CNN to either learn a single feature embedding with independent attribute classifiers [42], or directly learn multiple attribute-specific subspaces [2]. However, a single feature embedding cannot capture the contradictory notions of similarity in different attributes. Direct subspace learning will result in high feature redundancy without feature sharing mechanisms. Our method overcomes these issues by jointly learning a universal embedding and the associated subspaces.

Recent methods improve by combining the attribute constraints with a

classification loss [79] or even human expertise [73]. One closely-related method to ours, called conditional similarity network [69], learns masks on top of a shared embedding to disentangle latent subspaces for different attributes. However their subspaces are independently learned in a axis-aligned manner, which will be shown to underperform our joint learning of arbitrary projections.

Deep Embedding Learning and Data Sampling

Deep embedding methods typically map images into an embedding space, where their distances preserve the relative similarity. The contrastive loss [22] and triplet loss [62] are two popular embedding methods, both striving for a similar objective: to bring the same-class images close together while pushing away inter-class images. Contrastive loss chooses to enforce an absolute distance margin to separate the positive image pairs from negative pairs. It is thus not as flexible as triplet loss that only constrains the relative distance relation. Recent improvements are obtained by using more examples within a batch [50, 66] and using an adaptive distance function to evaluate similarities [26]. However, these embedding methods are only designed to characterize a single notion of semantic similarity. While we learn universal embeddings in a multi-attribute context.

[74] claimed that the data sampling strategy in batch plays an equal or more important role than the loss function. The authors proposed a distance weighted sampling strategy that samples data uniformly according to their relative distance. This leads to batch data uniformly spread over the whole distance range, and provides stabler gradients than random sampling, semi-hard negative mining [62], and hard negative mining [63]. In [77], an ensemble model is used to mine hard examples at multiple levels. Despite achieved gains, these sampling strategies suffer from expensive extra costs of distance computations. Our method uses all the $\mathcal{O}(n^3)$ triplet data in batch for the first time, and eliminates the need for hard triplet mining by a simple

yet effective weighting scheme at negligible cost.

Adversarial Network Training

Adversarial training of neural networks has been shown effective for image generation [15] and data augmentation with synthetic rare images [27]. With similar ideas of generating *adversarial* images, [21] proposed to perturb image pixels to produce erroneous class labels, and many other works followed, e.g., [82]. While [60] proposed to manipulate deep feature embeddings instead to generate hard-to-classify examples, which can be easier than image space perturbations. In line with such adversarial embedding learning, [71] proposed two adversarial networks to generate adversarial features that mimic object occlusions and deformations respectively and are hard for a detector to classify. A related attributed-guided augmentation method [16] synthesizes new features at a desired attribute strength via a deep encoder-decoder. In comparison, we only employ a two-layered regression network to generate adversarial universal feature embeddings, aimed for violating triplet similarity constraints in different subspaces. This simple adversarial network already works well with our proposed loss function.

Methodology

Our goal is to learn a universal embedding function $f(x;\Theta)$ via a CNN with parameters Θ, from image x into a common feature space $\mathbb{R}^d$, such that the embedded features can be shared or disentangled to encode different notions of similarity. For brevity, in the following we will omit Θ and use $f(x)$ and $f(x;\Theta)$ interchangeably.

Multi-Task Learning with Attributes

To achieve the above goal, we append to the universal embedding $f(x)$ one fully-connected layer with weights $W_a \in \mathbb{R}^{b\times d}$ to map to a small subspace (of

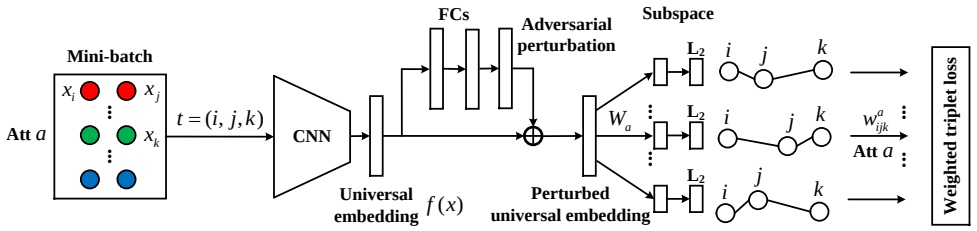

Figure 3.3.1: The network architecture to learn the Weighted Triplet-induced Adversarial Universal Embedding (WT-AUE). The mini-batch contains one (or several) positive pair for each of all N_a attributes (denoted by different colors). Then the exhaustively sampled triplet $t = (i, j, k)$ passes through a CNN and adversarial network to obtain the perturbed universal embeddings for robustness. The universal embeddings are finally mapped to different attribute-specific subspaces, where triplet t can have contradictory notions of similarity. The weighted triplet loss is proposed to aggregate the triplet losses in all subspaces, which simultaneously encourages hard triplet mining and feature interactions across concepts.

dimension $b < d$) for each attribute a (see Fig. 4.3.1). In each subspace, the attribute-specific similarity is used to supervise the learning of subspace features $W_a f(x)$, and in turn, that of universal embedding $f(x)$. We characterize similarities by the Euclidean distance between features in the corresponding subspace:

$$D^a_{ij} = \|W_a f(x_i) - W_a f(x_j)\|_2, \tag{3.1}$$

where the feature vector $W_a f(x_i)$ is normalized to have unit length for training stability.

The triplet loss [62] is used to learn such Euclidean distances in each subspace. We sample triplets $T_a = \{t_a = (i, j, k)\}$ with the anchor i, positive j and negative k examples defined by their labels of attribute a. We wish to enforce the relative distance relation $D^a_{ij} < D^a_{ik}$, and arrive to the following triplet loss for multi-task learning:

$$L_{tri} = \frac{1}{N_a |T_a|} \sum_a \sum_{t_a} \left[D^a_{ij} - D^a_{ik} + m \right]_+, \tag{3.2}$$

where $[\cdot]_+ = max(0, \cdot)$ is the hinge function and m is the enforced margin. This loss function computes the loss for separately sampled triplets for every attribute in one mini-batch, and sums over all N_a attributes. It can thus jointly train all the subspaces and universal embeddings based on back-propagation. However, two important problems are left unaddressed: how to construct the batch and how to sample triplets in it. We will introduce our specific strategies next, and show that they significantly impact both the convergence rate and performance as found in [74].

Triplet Sampling and Weighted Triplet Loss

To construct compact but rich mini-batches that have sufficient data for multi-task learning, we follow the method in N-pair loss [66] that permits information recycling. Specifically, our batch is constructed with one (or several) positive pair for each of all N_a attributes, with batch size $n \propto 2N_a$ (see Fig. 4.3.1). To form triplets for each positive pair under attribute a, it is likely to retrieve the negative example of this attribute from the remaining examples in batch. If we similarly compute one loss for each triplet per attribute, as is done in [66] and many other works, we argue this is a huge waste of batch data in two ways: 1) Only one triplet is used to learn for each attribute, i.e., $|T_a| = 1$ in Eq. (4.2). 2) A triplet is just considered under one attribute at a time, not under all attribute labels which may capture contradictory similarity notions. It hence loses the chance of leveraging the competition between subspaces over the same data source to feedback to and improve the shared universal embeddings in terms of feature expressiveness. Indeed, Eq. (4.2) penalizes for various similarity aspects using separately sampled triplets $\{t_a\}$ that are not necessarily overlapped.

To avoid such data waste, we go to the other extreme and enumerate all the $\mathcal{O}(n^3)$ triplets $\{t = (i, j, k)\}$ from a n-sized batch, where each t is considered under all N_a attributes. The total complexity is $\mathcal{O}(n^3 N_a)$. Note our batch with size n is organized much more efficiently than a naive one constructed

with $\mathcal{O}(n^3 N_a)$ distinct triplets.

However, by making the most of batch data in this way, we are immediately faced with several other issues: 1) It quickly becomes intractable since the number of gradient computations per batch grows quadruply with respect to the batch size n and attribute number N_a. 2) Many triplets will not be valid with exactly one positive pair (i, j) and a negative example k. In the binary attribute case, the probability of composing a valid triplet t is $p_t = \frac{2\times1}{2^3} = \frac{1}{4}$. Although our batch already guarantees for each attribute, there will be at least one positive pair to form valid triplets, the invalid ones may still appear frequently and shall not contribute to the loss. 3) The optimization will suffer from slow convergence and poor local optima due to many non-informative triplets that induce (near-) zero loss. Existing works often alleviate this issue via sampling approaches, including semi-hard negative mining [62] and distance weighted sampling [74]. One drawback is that they incur expensive extra costs to evaluate the feature distances for sampling.

We propose a simple yet effective weighting scheme for the $\mathcal{O}(n^3)$ triplets, and come to a weighted triplet loss for multi-task learning:

$$L_{wt} = \frac{1}{n^3 N_a} \sum_t \sum_a w_{ijk}^a \left[D_{ij}^a - D_{ik}^a + m \right]_+ , \tag{3.3}$$

where w_{ijk}^a is the weight for triplet t under attribute a, and is normalized in batch. It is defined using the Jaccard index $J(A_i, A_j) = \frac{|A_i \cap A_j|}{|A_i \cup A_j|}$ that measures similarity between the attribute label sets A_i and A_j of images x_i and x_j. We have:

$$\begin{aligned} w_{ijk}^a &= J(a_i, a_j) \cdot (1 - J(a_i, a_k)) \\ &\cdot (1 - J(A_i \backslash \{a_i\}, A_j \backslash \{a_j\})) \cdot J(A_i \backslash \{a_i\}, A_k \backslash \{a_k\}), \end{aligned} \tag{3.4}$$

where the first two terms actually act as the triplet validity check under the current attribute a. They become zero for an invalid triplet and directly rule it out from the weighted loss. This significantly reduces the number of necessary

computations of gradients during a backward pass, making its speed more reasonable. The last two terms help to favor those valid triplets with hard positive and negative examples, which respectively have small and large similarities of attribute labels with respect to anchor i. Such weighting scheme can be regarded as a soft way of hard triplet mining, but has a negligible cost without expensive example search based on deep feature distance.

Ablation study: Table 4.4.1 compares our Weighted Triplet loss (WT) in Eq. (4.3) with various baselines to quantify the efficacy of each component: 1) triplet weighting by w^a_{ijk}, 2) reusing each triplet across attributes $\sum_a$, 3) and another dimension of data utilization by exhaustive search of triplets within batch $\sum_t$. We have the following observations:

- Triplet weighting as a soft hard mining scheme improves convergence quality and speed. We found equal weighting with the last two terms in Eq. (4.4) set to one, hurts performance on the Zappos50k [76] and CelebA [42] datasets. In comparison, weighting with w^a_{ijk} encourages hard triplets while suppressing low-quality ones to prevent poor local optima. It also converges about 3-5 times faster than equal weighting. When compared with traditional hard mining techniques, our soft weighting scheme eliminates the need for the expensive distance evaluations.

- If we do not penalize the same triplet under different attributes (used data size reduced to $\mathcal{O}(n^3)$), performance deteriorates too. We argue that by reusing each triplet across attributes, it helps to learn the structure of the private and shared features in universal embedding to encode different triplet similarities. In other words, feature sharing and disentangling are automatically learned in such process.

- Sufficient batch data usage matters. To stress its importance for universal embedding learning, we follow the common practice in literature by only sampling N_a rather than all $\mathcal{O}(n^3)$ triplets from batch,

Table 3.4.1: Average triplet prediction error (%) for 4 attributes on Zappos50k shoe dataset, and classification error (%) of 40 face attributes on CelebA dataset (cropped images). Our Weighted Triplet loss (WT) is compared against the variants of its three components: equal weighting, triplets not considered across attributes ($\mathcal{O}(n^3)$), sampling a triplet subset from batch with different strategies ($\mathcal{O}(N_a \cdot N_a)$).

Method	Zappos50k	CelebA
WT	**9.62**	**9.19**
Equal Weighting	9.91	9.88
$\mathcal{O}(n^3)$ not across attributes	10.05	9.75
$\mathcal{O}(N_a^2)$ random negative sampling	10.87	10.72
$\mathcal{O}(N_a^2)$ semi-hard sampling	10.71	10.54
$\mathcal{O}(N_a^2)$ distance weighted sampling	10.43	10.58
$\mathcal{O}(N_a^2)$ N-pair	10.29	10.24

but still keep our weighting and triplet reusing schemes. The complexity is thus reduced from $\mathcal{O}(n^3 N_a)$ to $\mathcal{O}(N_a \cdot N_a)$. We first sampled triplets using each of the N_a positive pairs in batch and a random negative example from the rest. This leads to a significant performance drop in both accuracy and convergence speed. We also experimented with semi-hard sampling [62] and distance weighted sampling [74] for the negative example, and found slightly better results but at a larger searching cost. The N-pair tuplets [66], extending triplets to include all the negative examples in batch rather than one, achieve larger gains verifying again the importance of using sufficient data.

Adversarial Learning of Universal Embedding

The above proposed weighted triplet loss is able to jointly learn universal embeddings and the associated subspaces in a data-driven way, which requires a large amount of multi-label annotations of attributes. However, it is often

difficult to collect attribute datasets that cover large enough data variance or label contrast to learn expressive universal embeddings. When such data are lacking on existing small- or medium-sized attribute datasets, we found the embedding quality is hindered indeed.

This motivates us to to enrich the data by generating adversarial examples in the *feature* space. Inspired by recent adversarial training works, we propose a simple adversarial regression sub-network to generate hard features for improving embedding quality. As shown in Fig. 4.3.1, the adversarial sub-network takes the universal embedding $f(x) \in \mathbb{R}^d$ as input, and generates via two equal-sized fully-connected layers (with parameters Θ_{ad}) a d-dimensional perturbation vector, which is then added to $f(x)$ to obtain a perturbed universal embedding $f_{ad}(x)$ with parameters Θ and Θ_{ad}.

Ideally, perturbations introduced in the universal embedding space should be passed to the following subspaces to make the triplets therein hard, i.e., triplet features may violate the similarity constraint with high loss. Our objective remains the same: to penalize such violation in all subspaces. This makes us come to our final loss for Weighted Triplet-induced Adversarial Universal Embedding (WT-AUE):

$$L_{wtaue} = L_{wt}(f(X)) - \alpha L_{wt}(f_{ad}(X)) + \beta \|f_{ad}(X) - f(X)\|_2, \qquad (3.5)$$

where X denotes the batch data $\{x_i\}$, the last term penalizes the L_2 norm of adversarial perturbations to prevent embedding inflation, and α and β are weighting parameters.

Such WT-AUE loss lets the original CNN and its adversarial sub-network compete against each other: if the perturbation of universal embeddings results in violations of triplet constraints in some subspaces, the adversarial loss (second term in Eq. (4.5)) will be low but the original WT loss (first term) is high. By overcoming the obstacles created by the adversary, our universal embedding will be more noise-resistant. In other words, this helps impose safe margins in the local neighborhoods of both universal space and

different subspaces. Fig. 4.4.1 shows an example where a triplet is perturbed to be hard with closer positive and negative examples that will violate the margin constraint in two subspaces. Separating these examples with our loss can effectively maintain local margins in all embedding spaces. By walking through the entire embedding spaces along repeated training epochs, we finally achieve consistent discrimination in any local neighborhood.

Overall training procedure: We first train on one dataset without adversaries for about 20k iterations, in order to learn the universal embeddings with parameters Θ and attribute-specific subspaces with $\{W_a\}$. Then we pre-train the adversarial sub-network with Θ_{ad} for about 10k iterations while fixing Θ. Finally, both Θ_{ad} and Θ are jointly learned until convergence. We found the adversarial pre-training is important. It teaches the adversarial sub-network how to generate meaningful perturbations in the initial stage with random initialization. Otherwise, with a well-learned WT embedding, the adversarial learning is likely to be stuck with meaningless local minima having zero adversarial loss almost all time. To this end, we pre-train using the N_a triplets sampled from batch, each adversarially trained only for one attribute rather than N_a, with total complexity N_a. This generates universal perturbations aimed for one attribute at a time, and starts to learn how to maximally violate the corresponding triplet constraint with high loss. Once sufficient adversary is learned for all attributes, we move to the joint training and the adversarial sub-network improves as the original CNN becomes better and better and vice versa.

Experiments

Datasets and Implementation Details

Zappos50k shoe dataset. The first attribute dataset we use is Zappos50k [76] with 50k images. The image size 136×102 is resized to 136×102 by us. We consider the 4 attribute labels: the shoe type (shoes,

boots, sandals or slippers), shoe gender (women, men, girls or boys), the height of the shoes' heels (numeric values from 0 to 5 inches) and the closing mechanism of the shoes (buckle, pull on, slip on, hook and loop or laced up). For the numeric "heel height" attribute labels, we use their closeness to define the positive and negative examples in triplets. Following [69], we split the dataset into three parts: 70% for training, 10% for validation and 20% for testing. We similarly have 40k testing triplets $\{t_a = (i, j, k)\}$ for each attribute a. The testing criteria is to evaluate the validity of each t_a by the corresponding subspace features: whether the feature distance between the positive pair (i, j) is smaller than that between the negative pair (i, k) of attribute a. The error rate is 50% by chance. The extra shoes' brand information is used to perform off-task classification to show the generalization ability of our universal embedding.

CelebA dataset. The face attribute dataset CelebA [42] contains about 202k face images of 10k identities. Each image is annotated with 40 binary attributes like "male", "oval face" and "wavy hair". The dataset is split into 162k training, 20k validation and 20k testing images. There are no identity overlap between these splits. The 5 key points for each face image are used to align and crop image to 55×47 pixels. Both the original and cropped images are used to report our results, in terms of the average classification error across attributes. To classify each attribute for a testing image, we follow [42] to train a SVM classifier in the corresponding attribute subspace on validation set.

Implementation details. For fair comparisons with recent works, we similarly use the 18 layer deep residual network [23] and the 12-layer deep fully convolutional neural network in [31] on Zappos50k and CelebA datasets, respectively. We use the last global average pooling layer of the two networks as our universal embedding $f(x)$, with dimensions $d = 64$ and $d = 1024$. The respective subspace dimensions are $b = 5$ and $b = 20$. To set a proper batch size $n \propto 2N_a$ for two datasets with attribute numbers $N_a = 4$ and $N_a = 40$, we balance the computational cost against data adequacy in batch. Since Zappos50k has much fewer attributes to consider than CelebA, we sample

more positive pairs per attribute in batch for Zappos50k than for CelebA. In practice, we sample 5 and 1 positive pairs per attribute for the two datasets, having batch size $n = 40$ and $n = 80$. For all our experiments, the initial learning rate is 0.001, and $\alpha = 1$, $\beta = 0.0005$ in Eq. (4.5).

Attribute and Subspace Evaluations

We start with the evaluations of attribute prediction performance in difference subspaces. Note on Zappos50k dataset, the attribute is directly evaluated by checking the subspace features under triplet constraints; on CelebA dataset, we classify attributes based on subspace features. These experiments are to test how well our universal embedding can factorize subspaces with different attribute concepts.

Table 4.5.1 compares our method with several important baselines on Zappos50k. The single triplet embedding is learned from all available triplets as if they all come from a common space. It leads to a high error rate of 23.72%, showing that multiple attribute notions cannot be captured in a single space. Training a set of attribute-specialized triplet embeddings can be an immediate remedy, reducing the error rate to 11.35%. However, this comes at the cost of N_a times more network parameters. Conditional Similarity Network (CSN) [69] achieves efficiency by sharing a common embedding and learning masks to attend to relevant dimensions for various attributes. CSN even outperforms training a set of embeddings in accuracy. Our Weighted Triplet loss (WT) achieves a drastic improvement (9.62%) by mapping from a universal embedding to attribute-specific subspaces, and performing a weighted combination of the exhaustively sampled triplets across all subspaces. This indicates the importance of maximal and discriminatory data usage for a collaborative subspace learning. With adversarial perturbations that robustify embedding learning, our WT-AUE achieves the lowest error rate of 9.34%.

Table 4.5.2 compares our method with the previous FaceTracer [38] and PANDA [78] as well as state-of-the-art face attribute prediction methods like

Table 3.5.1: Average triplet prediction error (%) of 4 attributes on Zappos50k dataset.

Method	Error rate
Single triplet embedding	23.72
Set of specialized triplet embeddings	11.35
Conditional similarity network	10.73
WT	9.62
WT-AUE	**9.34**

Table 3.5.2: Average classification error (%) of 40 attributes on the CelebA original and cropped image sets.

Method	Original	Cropped
FaceTracer	18.88	-
PANDA	15.00	-
[42]	12.70	-
[70]	12.00	-
[83]	10.20	-
[58]	-	9.06
SSP+SSG	**8.84**	**8.20**
WT	9.58	9.19
WT-AUE	9.02	8.46

SSP+SSG [31]. Our WT-AUE outperforms most prior arts by a large margin, and is comparable to the SSP+SSG method that employs another semantic segmentation network to improve attribute prediction. Unlike most of these single-embedding-based methods, our WT-AUE excels by joint learning of a universal embedding and different subspaces, where adversarial learning helps a lot too (in comparison to WT).

Generalization to Unseen Attributes

One benefit of our joint training of universal embedding and attribute-specific subspaces is that, this fosters automatic feature interactions (e.g., sharing or disentangling) in the universal embedding space to generate different attribute notions or even new ones. This suggests the potential of feature generalization

Table 3.5.3: Average triplet prediction error (%) for 4 attributes on Zappos50k dataset, in a generalization setting: leave-one-attribute-out training of universal embedding which is then frozen, followed by training the new subspace for the left out attribute with its different amounts of training set. We report the average result from such training and testing for each of the 4 attributes.

Method	Error rate
WT-AUE	**9.34**
80% train	9.54
40% train	10.12
10% train	10.94
Conditional similarity network	10.73

from reduced feature redundancy. To prove this hypothesis empirically, we conduct a generalization experiment in the leave-one-attribute-out style: each time we learn the universal embedding supervised by only $N_a - 1$ attributes, and then freeze the universal embedding parameters to learn the subspace W_a for the left out (thus new) attribute. Note both learning stages use the WT-AUE loss. Our goal is to test how well such learned universal embedding can generalize to unseen attribute concepts.

Considering the Zappos50k dataset (50k images) is smaller than CelebA (202k images), we use Zappos50k to better examine our embedding's generalization ability with a small data size. Moreover, we further mimic low-shot learning for the left-out attribute with a decreasing amount of training data for it (80%, 40%, 10%). Table 4.5.3 lists the triplet prediction error rate averaged across $N_a = 4$ attributes under such generalization test. We observe a graceful performance degradation with fewer and fewer training data of unseen attributes. More interestingly, using only 10% training data of them, we achieve comparable results to the state-of-the-art conditional similarity network [69]. This indeed validates the superior generalization ability of our universal embedding.

Table 3.5.4: Top 1 accuracy (%) of the off-task brand classification on Zappos50k shoe dataset.

Method	Top 1 accuracy
ImageNet pre-trained	54.00
Finetuned single triplet embedding	49.08
Conditional similarity network	53.67
WT-AUE	**56.89**

Transfer Learning for Off-Task Recognition

Another way to evaluate the generalization ability of our universal embedding is to use it for transfer learning under a different task. We use Zappos50k dataset again for its small size. Following [69], we select the 30 shoe brands with the most examples in Zappos50k for standard multi-way classification. During training, we first learn the universal embedding supervised by 4 attributes with the WT-AUE loss, then fix the embedding and learn one fully connected layer with Softmax loss for the 30 brand classes.

Table 4.5.4 shows our method attains the highest classification accuracy. Single triplet embedding hurts the performance since it learns contradictory similarity notions and hinders the embedding quality. Our method outperforms the conditional similarity network [69] and achieves gains over the ImageNet pre-trained model. This indicates the strong supervision provided by our WT-AUE loss boosts embedding generalization.

Conclusion

We present a multi-task learning method to learn universal feature embeddings from multi-label attributes. By mapping universal embeddings to various attribute-specific subspaces, we propose a weighted triplet loss to learn from all the triplets, especially those hard ones in all subspaces during each mini-batch training. Adversary is also introduced to robustify the learning by competing against the universal embedding perturbations. We show this effectively encourages feature sharing and disentangling in universal space,

which leads to reduced feature redundancy and boosted generalization ability. Experiments not only demonstrate the superior attribute prediction results by our learned subspaces, but also validate the generalization ability of our universal embedding in the tasks of low-shot learning for unseen attributes and off-task recognition.

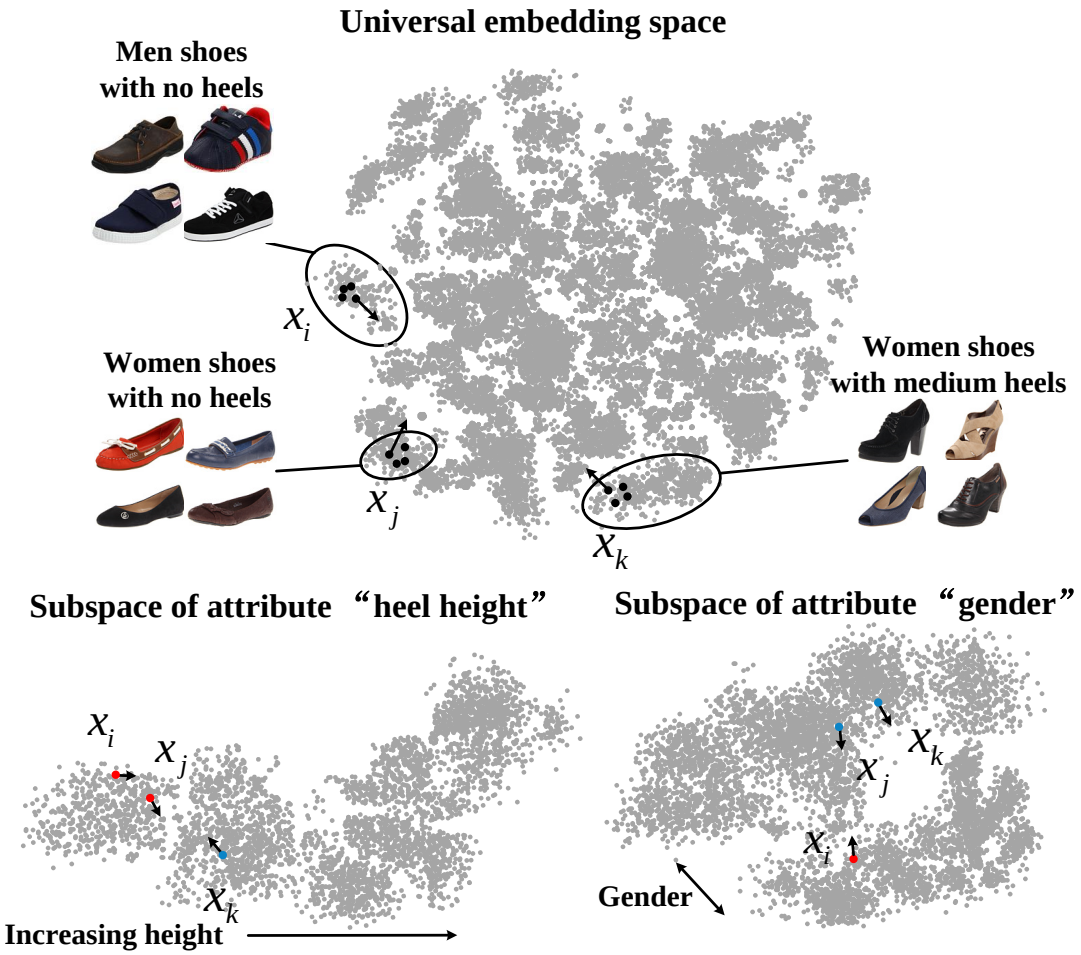

Figure 3.4.1: Adversarial perturbations of the universal embeddings of a triplet $t = (i, j, k)$, and the resulting perturbations in two example subspaces of "heel height" and "gender" of shoes. The triplet is made harder with margin-violating examples (color denotes class label) in each subspace, which improves the embedding quality and convergence speed.

4

Discovering Latent Factors of Variation for Context-based Image Understanding

Abstract

We address the problem of learning a universal embedding space from which different semantic concepts or notions of similarity can originate. Contemporary attribute datasets provide rich multi-label annotations to achieve this goal.Universal embeddings learned from the multi-label attributes would naturally encourage feature sharing, leading to reduced feature redundancy and boosted generalization ability. This paper presents a multi-task framework to learn universal embeddings by mapping them to different subspaces, each corresponds to one particular attribute and is supervised by the triplet similarity. A weighted triplet loss is proposed to combine all the possible triplets for all attributes in a small batch, with weights designed to favor triplets with hard examples by computing their attribute set similarity. This not only eliminates the need for hard triplet mining in batch, and makes the most of available data in a simple way, but also helps to learn the structure of the private and shared features in universal space to encode different triplet similarities across attributes. The learning is made more robust by competing against an adversarial network that perturbs the universal embeddings to increase the loss. The resulting Weighted Triplet-induced Adversarial Universal Embedding (WT-AUE) achieves strong results for attribute prediction, low-shot generalization as well as off-task recognition.

Introduction

Deep networks have revolutionized visual understanding through the ability to learn data-driven representations. Such representations are typically trained for a particular task with massive amounts of labeled data [14].

Learning embeddings: A particularly influential variant of representation learning is the problem of learning embedding spaces, which transform input images into a finite-dimensional vector space where images with similar labels

are closeby according to some distance [61]. Directly learning embeddings has many advantages. Firstly, they are not be tied to category labels, and so can be trained on pairs (or triples) of examples with same/different tags [22]. Secondly, they can be used for zero-shot or one-shot learning by applying the learned distance functions to novel class exemplars [62]. However, embeddings do have limitations. They assume that a *single* universal notion of distance always holds. But this is not always true. Consider the problem of learning an embedding of faces that is *attribute-specific* (Fig. 4.2.1). A gender-specific embedding will likely need to be invariant to the age attribute, while an age-specific embedding will likely need to be invariant to gender. This suggests it may be hopeless to learn a single "one-size-fits-all" embedding, since an embedding that is invariant to both age and gender seems useless for either task.

Multi-task learning: To reconcile the apparent contradiction above, we appeal to the large literature on multi-task learning [6]. Recent work has shown that a shared set of nonlinear features can be jointly learned to serve a variety of different tasks. Examples include joint learning for facial landmark detection, head pose and facial attribute prediction [80], training a universal Convolutional Neural Network (CNN) for low-, mid-, and high-level vision tasks [35], etc. Multi-task networks have also been shown to transfer better to novel tasks because the shared features appear to be more generic [53]. From our perspective, the shared feature representation can be thought as as an embedding that is *linearly projected* into (one or more) one-dimensional spaces representing classifiers for one or more tasks.

Multi-task embeddings: Our formulation can be seen as the natural integration of multi-task learning and nonlinear embeddings: we learn a single universal embedding space that is *linearly projected* into attribute-specific subspaces. Our overall philosophy is similar to multi-view embeddings [2] and the recent work of conditional similarity networks (CSNs) [69]. However, there

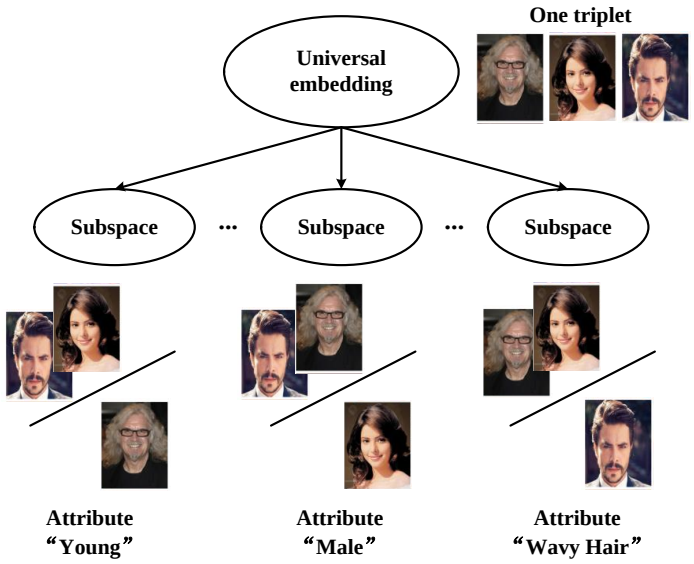

Figure 4.2.1: Learning a universal embedding space that maps to various subspaces with different notions of similarity in attributes. Different attributes often require opposite invariances (for example, "young" vs. "male"), which can be captured by different subspaces but not by a single embedding. The joint learning of a universal embedding and different subspaces would encourage automatic feature sharing and disentangling in the universal space, leading to reduced feature redundancy and boosted generalization as well.

are several important differences. The former learns separate embeddings without any multi-task sharing, while the latter learns a single universal embedding that is element-wise masked out to produce different embeddings capturing different notions of similarity. In some sense, CSNs learn axis-aligned embedding subspaces, while we learn arbitrary linear projections (consistent with common best-practices in multi-task learning). We demonstrate that our universal embedding space can be used for few-shot learning of attribute-specific embeddings. This has the practical advantage that data (e.g., faces) can be stored using a fixed universal embedding representation that will likely apply to *any* future attribute of interest.

Training: We also contribute several algorithmic innovations to the problem of learning embeddings specific to our multi-task setting. As in much

recent work, we learn with triples of examples, but use multiple attribute labels to speed up convergence and increase accuracy in three distinct ways: (1) When learning on mini-batches, we project each triplet to every attribute-specific subspace, effectively learning from *multiple* supervisory signals per batch. (2) Unlike the sampling techniques discussed in recent work [74], we exhaustively use all triplets in batch. We also find that *softly weighting* triplets improves convergence. We use multi-view attribute labels to define a notion of semantic similarity between examples, which is in turn used to efficiently construct a meaningful weighting. The weights particularly favor those hard examples, acting as a soft tool of hard triplet mining. (3) We introduce an *adversarial* network that perturbs the universal embeddings so as to generate harder triplets in each subspace.

Related Work

Multi-Task Embedding Learning

Our universal embedding is learned in a multi-task framework. Example multi-task learning methods learn CNN features for joint facial landmark detection, head pose and facial attribute prediction [80], or for low-, mid-, and high-level vision tasks [35]. Another line of multi-task embedding methods learn from multi-label attribute annotations. For example, [55] proposed a higher-order Boltzmann machine to disentangle attributes from a manifold with multiplicative interactions. Others use CNN to either learn a single feature embedding with independent attribute classifiers [42], or directly learn multiple attribute-specific subspaces [2]. However, a single feature embedding cannot capture the contradictory notions of similarity in different attributes. Direct subspace learning will result in high feature redundancy without feature sharing mechanisms. Our method overcomes these issues by jointly learning a universal embedding and the associated subspaces.

Recent methods improve by combining the attribute constraints with a

classification loss [79] or even human expertise [73]. One closely-related method to ours, called conditional similarity network [69], learns masks on top of a shared embedding to disentangle latent subspaces for different attributes. However their subspaces are independently learned in a axis-aligned manner, which will be shown to underperform our joint learning of arbitrary projections.

Deep Embedding Learning and Data Sampling

Deep embedding methods typically map images into an embedding space, where their distances preserve the relative similarity. The contrastive loss [22] and triplet loss [62] are two popular embedding methods, both striving for a similar objective: to bring the same-class images close together while pushing away inter-class images. Contrastive loss chooses to enforce an absolute distance margin to separate the positive image pairs from negative pairs. It is thus not as flexible as triplet loss that only constrains the relative distance relation. Recent improvements are obtained by using more examples within a batch [50, 66] and using an adaptive distance function to evaluate similarities [26]. However, these embedding methods are only designed to characterize a single notion of semantic similarity. While we learn universal embeddings in a multi-attribute context.

[74] claimed that the data sampling strategy in batch plays an equal or more important role than the loss function. The authors proposed a distance weighted sampling strategy that samples data uniformly according to their relative distance. This leads to batch data uniformly spread over the whole distance range, and provides stabler gradients than random sampling, semi-hard negative mining [62], and hard negative mining [63]. In [77], an ensemble model is used to mine hard examples at multiple levels. Despite achieved gains, these sampling strategies suffer from expensive extra costs of distance computations. Our method uses all the $\mathcal{O}(n^3)$ triplet data in batch for the first time, and eliminates the need for hard triplet mining by a simple

yet effective weighting scheme at negligible cost.

Adversarial Network Training

Adversarial training of neural networks has been shown effective for image generation [15] and data augmentation with synthetic rare images [27]. With similar ideas of generating *adversarial* images, [21] proposed to perturb image pixels to produce erroneous class labels, and many other works followed, e.g., [82]. While [60] proposed to manipulate deep feature embeddings instead to generate hard-to-classify examples, which can be easier than image space perturbations. In line with such adversarial embedding learning, [71] proposed two adversarial networks to generate adversarial features that mimic object occlusions and deformations respectively and are hard for a detector to classify. A related attributed-guided augmentation method [16] synthesizes new features at a desired attribute strength via a deep encoder-decoder. In comparison, we only employ a two-layered regression network to generate adversarial universal feature embeddings, aimed for violating triplet similarity constraints in different subspaces. This simple adversarial network already works well with our proposed loss function.

Methodology

Our goal is to learn a universal embedding function $f(x;\Theta)$ via a CNN with parameters Θ, from image x into a common feature space $\mathbb{R}^d$, such that the embedded features can be shared or disentangled to encode different notions of similarity. For brevity, in the following we will omit Θ and use $f(x)$ and $f(x;\Theta)$ interchangeably.

Multi-Task Learning with Attributes

To achieve the above goal, we append to the universal embedding $f(x)$ one fully-connected layer with weights $W_a \in \mathbb{R}^{b\times d}$ to map to a small subspace (of

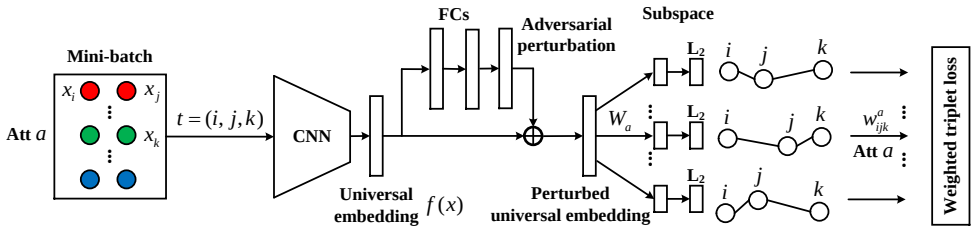

Figure 4.3.1: The network architecture to learn the Weighted Triplet-induced Adversarial Universal Embedding (WT-AUE). The mini-batch contains one (or several) positive pair for each of all N_a attributes (denoted by different colors). Then the exhaustively sampled triplet $t = (i, j, k)$ passes through a CNN and adversarial network to obtain the perturbed universal embeddings for robustness. The universal embeddings are finally mapped to different attribute-specific subspaces, where triplet t can have contradictory notions of similarity. The weighted triplet loss is proposed to aggregate the triplet losses in all subspaces, which simultaneously encourages hard triplet mining and feature interactions across concepts.

dimension $b < d$) for each attribute a (see Fig. 4.3.1). In each subspace, the attribute-specific similarity is used to supervise the learning of subspace features $W_a f(x)$, and in turn, that of universal embedding $f(x)$. We characterize similarities by the Euclidean distance between features in the corresponding subspace:

$$D^a_{ij} = \|W_a f(x_i) - W_a f(x_j)\|_2, \tag{4.1}$$

where the feature vector $W_a f(x_i)$ is normalized to have unit length for training stability.

The triplet loss [62] is used to learn such Euclidean distances in each subspace. We sample triplets $T_a = \{t_a = (i, j, k)\}$ with the anchor i, positive j and negative k examples defined by their labels of attribute a. We wish to enforce the relative distance relation $D^a_{ij} < D^a_{ik}$, and arrive to the following triplet loss for multi-task learning:

$$L_{tri} = \frac{1}{N_a |T_a|} \sum_a \sum_{t_a} \left[D^a_{ij} - D^a_{ik} + m \right]_+, \tag{4.2}$$

where $[\cdot]_+ = max(0, \cdot)$ is the hinge function and m is the enforced margin. This loss function computes the loss for separately sampled triplets for every attribute in one mini-batch, and sums over all N_a attributes. It can thus jointly train all the subspaces and universal embeddings based on back-propagation. However, two important problems are left unaddressed: how to construct the batch and how to sample triplets in it. We will introduce our specific strategies next, and show that they significantly impact both the convergence rate and performance as found in [74].

Triplet Sampling and Weighted Triplet Loss

To construct compact but rich mini-batches that have sufficient data for multi-task learning, we follow the method in N-pair loss [66] that permits information recycling. Specifically, our batch is constructed with one (or several) positive pair for each of all N_a attributes, with batch size $n \propto 2N_a$ (see Fig. 4.3.1). To form triplets for each positive pair under attribute a, it is likely to retrieve the negative example of this attribute from the remaining examples in batch. If we similarly compute one loss for each triplet per attribute, as is done in [66] and many other works, we argue this is a huge waste of batch data in two ways: 1) Only one triplet is used to learn for each attribute, i.e., $|T_a| = 1$ in Eq. (4.2). 2) A triplet is just considered under one attribute at a time, not under all attribute labels which may capture contradictory similarity notions. It hence loses the chance of leveraging the competition between subspaces over the same data source to feedback to and improve the shared universal embeddings in terms of feature expressiveness. Indeed, Eq. (4.2) penalizes for various similarity aspects using separately sampled triplets $\{t_a\}$ that are not necessarily overlapped.

To avoid such data waste, we go to the other extreme and enumerate all the $\mathcal{O}(n^3)$ triplets $\{t = (i, j, k)\}$ from a n-sized batch, where each t is considered under all N_a attributes. The total complexity is $\mathcal{O}(n^3 N_a)$. Note our batch with size n is organized much more efficiently than a naive one constructed

with $\mathcal{O}(n^3 N_a)$ distinct triplets.

However, by making the most of batch data in this way, we are immediately faced with several other issues: 1) It quickly becomes intractable since the number of gradient computations per batch grows quadruply with respect to the batch size n and attribute number N_a. 2) Many triplets will not be valid with exactly one positive pair (i, j) and a negative example k. In the binary attribute case, the probability of composing a valid triplet t is $p_t = \frac{2\times1}{2^3} = \frac{1}{4}$. Although our batch already guarantees for each attribute, there will be at least one positive pair to form valid triplets, the invalid ones may still appear frequently and shall not contribute to the loss. 3) The optimization will suffer from slow convergence and poor local optima due to many non-informative triplets that induce (near-) zero loss. Existing works often alleviate this issue via sampling approaches, including semi-hard negative mining [62] and distance weighted sampling [74]. One drawback is that they incur expensive extra costs to evaluate the feature distances for sampling.

We propose a simple yet effective weighting scheme for the $\mathcal{O}(n^3)$ triplets, and come to a weighted triplet loss for multi-task learning:

$$L_{wt} = \frac{1}{n^3 N_a} \sum_t \sum_a w_{ijk}^a \left[D_{ij}^a - D_{ik}^a + m \right]_+ , \tag{4.3}$$

where w_{ijk}^a is the weight for triplet t under attribute a, and is normalized in batch. It is defined using the Jaccard index $J(A_i, A_j) = \frac{|A_i \cap A_j|}{|A_i \cup A_j|}$ that measures similarity between the attribute label sets A_i and A_j of images x_i and x_j. We have:

$$\begin{aligned} w_{ijk}^a &= J(a_i, a_j) \cdot (1 - J(a_i, a_k)) \\ &\cdot (1 - J(A_i \backslash \{a_i\}, A_j \backslash \{a_j\})) \cdot J(A_i \backslash \{a_i\}, A_k \backslash \{a_k\}), \end{aligned} \tag{4.4}$$

where the first two terms actually act as the triplet validity check under the current attribute a. They become zero for an invalid triplet and directly rule it out from the weighted loss. This significantly reduces the number of necessary

computations of gradients during a backward pass, making its speed more reasonable. The last two terms help to favor those valid triplets with hard positive and negative examples, which respectively have small and large similarities of attribute labels with respect to anchor i. Such weighting scheme can be regarded as a soft way of hard triplet mining, but has a negligible cost without expensive example search based on deep feature distance.

Ablation study: Table 4.4.1 compares our Weighted Triplet loss (WT) in Eq. (4.3) with various baselines to quantify the efficacy of each component: 1) triplet weighting by w^a_{ijk}, 2) reusing each triplet across attributes $\sum_a$, 3) and another dimension of data utilization by exhaustive search of triplets within batch $\sum_t$. We have the following observations:

- Triplet weighting as a soft hard mining scheme improves convergence quality and speed. We found equal weighting with the last two terms in Eq. (4.4) set to one, hurts performance on the Zappos50k [76] and CelebA [42] datasets. In comparison, weighting with w^a_{ijk} encourages hard triplets while suppressing low-quality ones to prevent poor local optima. It also converges about 3-5 times faster than equal weighting. When compared with traditional hard mining techniques, our soft weighting scheme eliminates the need for the expensive distance evaluations.

- If we do not penalize the same triplet under different attributes (used data size reduced to $\mathcal{O}(n^3)$), performance deteriorates too. We argue that by reusing each triplet across attributes, it helps to learn the structure of the private and shared features in universal embedding to encode different triplet similarities. In other words, feature sharing and disentangling are automatically learned in such process.

- Sufficient batch data usage matters. To stress its importance for universal embedding learning, we follow the common practice in literature by only sampling N_a rather than all $\mathcal{O}(n^3)$ triplets from batch,

Table 4.4.1: Average triplet prediction error (%) for 4 attributes on Zappos50k shoe dataset, and classification error (%) of 40 face attributes on CelebA dataset (cropped images). Our Weighted Triplet loss (WT) is compared against the variants of its three components: equal weighting, triplets not considered across attributes ($\mathcal{O}(n^3)$), sampling a triplet subset from batch with different strategies ($\mathcal{O}(N_a \cdot N_a)$).

Method	Zappos50k	CelebA
WT	**9.62**	**9.19**
Equal Weighting	9.91	9.88
$\mathcal{O}(n^3)$ not across attributes	10.05	9.75
$\mathcal{O}(N_a^2)$ random negative sampling	10.87	10.72
$\mathcal{O}(N_a^2)$ semi-hard sampling	10.71	10.54
$\mathcal{O}(N_a^2)$ distance weighted sampling	10.43	10.58
$\mathcal{O}(N_a^2)$ N-pair	10.29	10.24

but still keep our weighting and triplet reusing schemes. The complexity is thus reduced from $\mathcal{O}(n^3 N_a)$ to $\mathcal{O}(N_a \cdot N_a)$. We first sampled triplets using each of the N_a positive pairs in batch and a random negative example from the rest. This leads to a significant performance drop in both accuracy and convergence speed. We also experimented with semi-hard sampling [62] and distance weighted sampling [74] for the negative example, and found slightly better results but at a larger searching cost. The N-pair tuplets [66], extending triplets to include all the negative examples in batch rather than one, achieve larger gains verifying again the importance of using sufficient data.

Adversarial Learning of Universal Embedding

The above proposed weighted triplet loss is able to jointly learn universal embeddings and the associated subspaces in a data-driven way, which requires a large amount of multi-label annotations of attributes. However, it is often

difficult to collect attribute datasets that cover large enough data variance or label contrast to learn expressive universal embeddings. When such data are lacking on existing small- or medium-sized attribute datasets, we found the embedding quality is hindered indeed.

This motivates us to to enrich the data by generating adversarial examples in the *feature* space. Inspired by recent adversarial training works, we propose a simple adversarial regression sub-network to generate hard features for improving embedding quality. As shown in Fig. 4.3.1, the adversarial sub-network takes the universal embedding $f(x) \in \mathbb{R}^d$ as input, and generates via two equal-sized fully-connected layers (with parameters Θ_{ad}) a d-dimensional perturbation vector, which is then added to $f(x)$ to obtain a perturbed universal embedding $f_{ad}(x)$ with parameters Θ and Θ_{ad}.

Ideally, perturbations introduced in the universal embedding space should be passed to the following subspaces to make the triplets therein hard, i.e., triplet features may violate the similarity constraint with high loss. Our objective remains the same: to penalize such violation in all subspaces. This makes us come to our final loss for Weighted Triplet-induced Adversarial Universal Embedding (WT-AUE):

$$L_{wtaue} = L_{wt}(f(X)) - \alpha L_{wt}(f_{ad}(X)) + \beta \|f_{ad}(X) - f(X)\|_2, \qquad (4.5)$$

where X denotes the batch data $\{x_i\}$, the last term penalizes the L_2 norm of adversarial perturbations to prevent embedding inflation, and α and β are weighting parameters.

Such WT-AUE loss lets the original CNN and its adversarial sub-network compete against each other: if the perturbation of universal embeddings results in violations of triplet constraints in some subspaces, the adversarial loss (second term in Eq. (4.5)) will be low but the original WT loss (first term) is high. By overcoming the obstacles created by the adversary, our universal embedding will be more noise-resistant. In other words, this helps impose safe margins in the local neighborhoods of both universal space and

different subspaces. Fig. 4.4.1 shows an example where a triplet is perturbed to be hard with closer positive and negative examples that will violate the margin constraint in two subspaces. Separating these examples with our loss can effectively maintain local margins in all embedding spaces. By walking through the entire embedding spaces along repeated training epochs, we finally achieve consistent discrimination in any local neighborhood.

Overall training procedure: We first train on one dataset without adversaries for about 20k iterations, in order to learn the universal embeddings with parameters Θ and attribute-specific subspaces with $\{W_a\}$. Then we pre-train the adversarial sub-network with Θ_{ad} for about 10k iterations while fixing Θ. Finally, both Θ_{ad} and Θ are jointly learned until convergence. We found the adversarial pre-training is important. It teaches the adversarial sub-network how to generate meaningful perturbations in the initial stage with random initialization. Otherwise, with a well-learned WT embedding, the adversarial learning is likely to be stuck with meaningless local minima having zero adversarial loss almost all time. To this end, we pre-train using the N_a triplets sampled from batch, each adversarially trained only for one attribute rather than N_a, with total complexity N_a. This generates universal perturbations aimed for one attribute at a time, and starts to learn how to maximally violate the corresponding triplet constraint with high loss. Once sufficient adversary is learned for all attributes, we move to the joint training and the adversarial sub-network improves as the original CNN becomes better and better and vice versa.

Experiments

Datasets and Implementation Details

Zappos50k shoe dataset. The first attribute dataset we use is Zappos50k [76] with 50k images. The image size 136×102 is resized to 136×102 by us. We consider the 4 attribute labels: the shoe type (shoes,

boots, sandals or slippers), shoe gender (women, men, girls or boys), the height of the shoes' heels (numeric values from 0 to 5 inches) and the closing mechanism of the shoes (buckle, pull on, slip on, hook and loop or laced up). For the numeric "heel height" attribute labels, we use their closeness to define the positive and negative examples in triplets. Following [69], we split the dataset into three parts: 70% for training, 10% for validation and 20% for testing. We similarly have 40k testing triplets $\{t_a = (i, j, k)\}$ for each attribute a. The testing criteria is to evaluate the validity of each t_a by the corresponding subspace features: whether the feature distance between the positive pair (i, j) is smaller than that between the negative pair (i, k) of attribute a. The error rate is 50% by chance. The extra shoes' brand information is used to perform off-task classification to show the generalization ability of our universal embedding.

CelebA dataset. The face attribute dataset CelebA [42] contains about 202k face images of 10k identities. Each image is annotated with 40 binary attributes like "male", "oval face" and "wavy hair". The dataset is split into 162k training, 20k validation and 20k testing images. There are no identity overlap between these splits. The 5 key points for each face image are used to align and crop image to 55×47 pixels. Both the original and cropped images are used to report our results, in terms of the average classification error across attributes. To classify each attribute for a testing image, we follow [42] to train a SVM classifier in the corresponding attribute subspace on validation set.

Implementation details. For fair comparisons with recent works, we similarly use the 18 layer deep residual network [23] and the 12-layer deep fully convolutional neural network in [31] on Zappos50k and CelebA datasets, respectively. We use the last global average pooling layer of the two networks as our universal embedding $f(x)$, with dimensions $d = 64$ and $d = 1024$. The respective subspace dimensions are $b = 5$ and $b = 20$. To set a proper batch size $n \propto 2N_a$ for two datasets with attribute numbers $N_a = 4$ and $N_a = 40$, we balance the computational cost against data adequacy in batch. Since Zappos50k has much fewer attributes to consider than CelebA, we sample

more positive pairs per attribute in batch for Zappos50k than for CelebA. In practice, we sample 5 and 1 positive pairs per attribute for the two datasets, having batch size $n = 40$ and $n = 80$. For all our experiments, the initial learning rate is 0.001, and $\alpha = 1$, $\beta = 0.0005$ in Eq. (4.5).

Attribute and Subspace Evaluations

We start with the evaluations of attribute prediction performance in difference subspaces. Note on Zappos50k dataset, the attribute is directly evaluated by checking the subspace features under triplet constraints; on CelebA dataset, we classify attributes based on subspace features. These experiments are to test how well our universal embedding can factorize subspaces with different attribute concepts.

Table 4.5.1 compares our method with several important baselines on Zappos50k. The single triplet embedding is learned from all available triplets as if they all come from a common space. It leads to a high error rate of 23.72%, showing that multiple attribute notions cannot be captured in a single space. Training a set of attribute-specialized triplet embeddings can be an immediate remedy, reducing the error rate to 11.35%. However, this comes at the cost of N_a times more network parameters. Conditional Similarity Network (CSN) [69] achieves efficiency by sharing a common embedding and learning masks to attend to relevant dimensions for various attributes. CSN even outperforms training a set of embeddings in accuracy. Our Weighted Triplet loss (WT) achieves a drastic improvement (9.62%) by mapping from a universal embedding to attribute-specific subspaces, and performing a weighted combination of the exhaustively sampled triplets across all subspaces. This indicates the importance of maximal and discriminatory data usage for a collaborative subspace learning. With adversarial perturbations that robustify embedding learning, our WT-AUE achieves the lowest error rate of 9.34%.

Table 4.5.2 compares our method with the previous FaceTracer [38] and PANDA [78] as well as state-of-the-art face attribute prediction methods like

Table 4.5.1: Average triplet prediction error (%) of 4 attributes on Zappos50k dataset.

Method	Error rate
Single triplet embedding	23.72
Set of specialized triplet embeddings	11.35
Conditional similarity network	10.73
WT	9.62
WT-AUE	**9.34**

Table 4.5.2: Average classification error (%) of 40 attributes on the CelebA original and cropped image sets.

Method	Original	Cropped
FaceTracer	18.88	-
PANDA	15.00	-
[42]	12.70	-
[70]	12.00	-
[83]	10.20	-
[58]	-	9.06
SSP+SSG	**8.84**	**8.20**
WT	9.58	9.19
WT-AUE	9.02	8.46

SSP+SSG [31]. Our WT-AUE outperforms most prior arts by a large margin, and is comparable to the SSP+SSG method that employs another semantic segmentation network to improve attribute prediction. Unlike most of these single-embedding-based methods, our WT-AUE excels by joint learning of a universal embedding and different subspaces, where adversarial learning helps a lot too (in comparison to WT).

Generalization to Unseen Attributes

One benefit of our joint training of universal embedding and attribute-specific subspaces is that, this fosters automatic feature interactions (e.g., sharing or disentangling) in the universal embedding space to generate different attribute notions or even new ones. This suggests the potential of feature generalization

Table 4.5.3: Average triplet prediction error (%) for 4 attributes on Zappos50k dataset, in a generalization setting: leave-one-attribute-out training of universal embedding which is then frozen, followed by training the new subspace for the left out attribute with its different amounts of training set. We report the average result from such training and testing for each of the 4 attributes.

Method	Error rate
WT-AUE	**9.34**
80% train	9.54
40% train	10.12
10% train	10.94
Conditional similarity network	10.73

from reduced feature redundancy. To prove this hypothesis empirically, we conduct a generalization experiment in the leave-one-attribute-out style: each time we learn the universal embedding supervised by only $N_a - 1$ attributes, and then freeze the universal embedding parameters to learn the subspace W_a for the left out (thus new) attribute. Note both learning stages use the WT-AUE loss. Our goal is to test how well such learned universal embedding can generalize to unseen attribute concepts.

Considering the Zappos50k dataset (50k images) is smaller than CelebA (202k images), we use Zappos50k to better examine our embedding's generalization ability with a small data size. Moreover, we further mimic low-shot learning for the left-out attribute with a decreasing amount of training data for it (80%, 40%, 10%). Table 4.5.3 lists the triplet prediction error rate averaged across $N_a = 4$ attributes under such generalization test. We observe a graceful performance degradation with fewer and fewer training data of unseen attributes. More interestingly, using only 10% training data of them, we achieve comparable results to the state-of-the-art conditional similarity network [69]. This indeed validates the superior generalization ability of our universal embedding.

Table 4.5.4: Top 1 accuracy (%) of the off-task brand classification on Zappos50k shoe dataset.

Method	Top 1 accuracy
ImageNet pre-trained	54.00
Finetuned single triplet embedding	49.08
Conditional similarity network	53.67
WT-AUE	**56.89**

Transfer Learning for Off-Task Recognition

Another way to evaluate the generalization ability of our universal embedding is to use it for transfer learning under a different task. We use Zappos50k dataset again for its small size. Following [69], we select the 30 shoe brands with the most examples in Zappos50k for standard multi-way classification. During training, we first learn the universal embedding supervised by 4 attributes with the WT-AUE loss, then fix the embedding and learn one fully connected layer with Softmax loss for the 30 brand classes.

Table 4.5.4 shows our method attains the highest classification accuracy. Single triplet embedding hurts the performance since it learns contradictory similarity notions and hinders the embedding quality. Our method outperforms the conditional similarity network [69] and achieves gains over the ImageNet pre-trained model. This indicates the strong supervision provided by our WT-AUE loss boosts embedding generalization.

Conclusion

We present a multi-task learning method to learn universal feature embeddings from multi-label attributes. By mapping universal embeddings to various attribute-specific subspaces, we propose a weighted triplet loss to learn from all the triplets, especially those hard ones in all subspaces during each mini-batch training. Adversary is also introduced to robustify the learning by competing against the universal embedding perturbations. We show this effectively encourages feature sharing and disentangling in universal space,

which leads to reduced feature redundancy and boosted generalization ability. Experiments not only demonstrate the superior attribute prediction results by our learned subspaces, but also validate the generalization ability of our universal embedding in the tasks of low-shot learning for unseen attributes and off-task recognition.

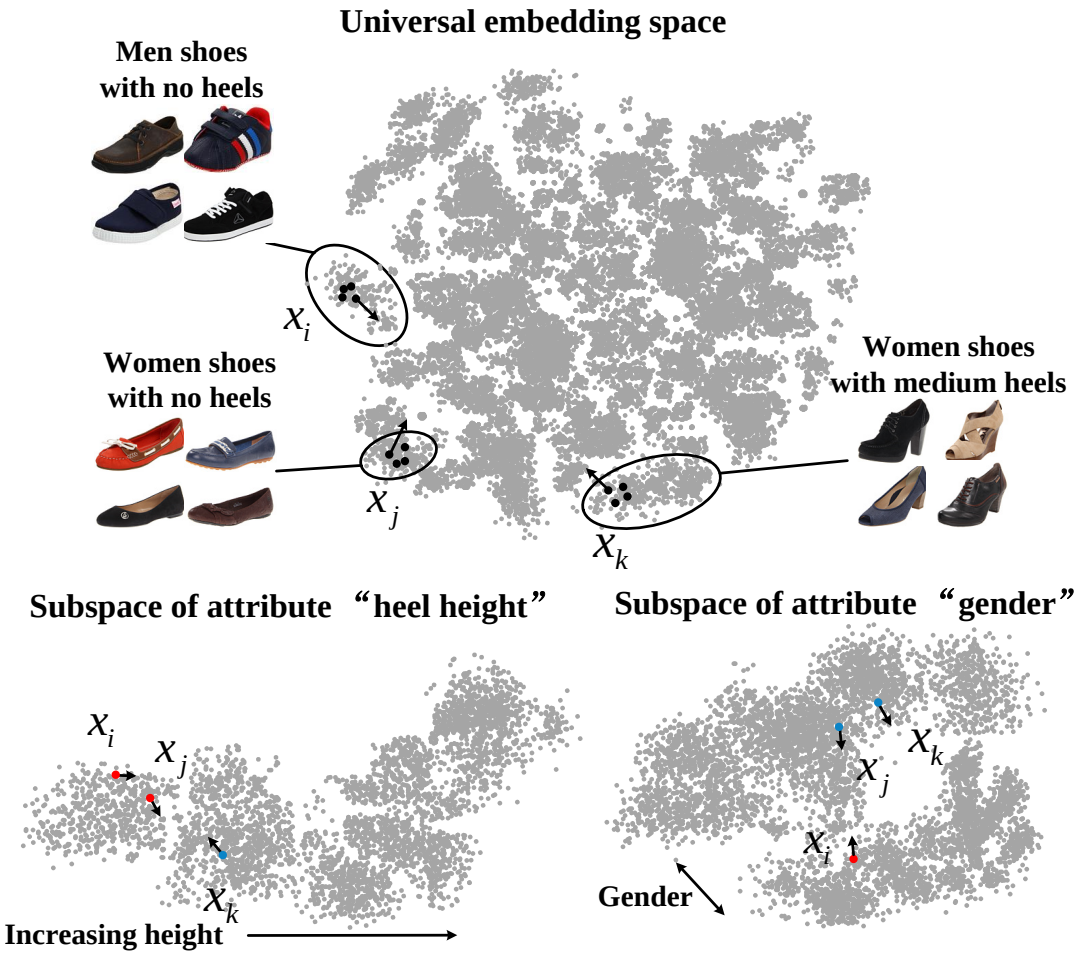

Figure 4.4.1: Adversarial perturbations of the universal embeddings of a triplet $t = (i, j, k)$, and the resulting perturbations in two example subspaces of "heel height" and "gender" of shoes. The triplet is made harder with margin-violating examples (color denotes class label) in each subspace, which improves the embedding quality and convergence speed.

References

[1] S. Albarqouni, C. Baur, F. Achilles, V. Belagiannis, S. Demirci, and N. Navab. Aggnet: Deep learning from crowds for mitosis detection in breast cancer histology images. *IEEE Transactions on Medical Imaging*, 35(5):1313–1321, 2016.

[2] Ehsan Amid and Antti Ukkonen. Multiview triplet embedding: Learning attributes in multiple maps. In *ICML*, 2015.

[3] V. Badrinarayanan, A. Kendall, and R. Cipolla. SegNet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(12): 2481–2495, 2017.

[4] Amy Bearman, Olga Russakovsky, Vittorio Ferrari, and Li Fei-Fei. What's the point: Semantic segmentation with point supervision. In *Proceedings of the European Conference on Computer Vision*, 2016.

[5] H. Caesar, J. Uijlings, and V. Ferrari. COCO-Stuff: Thing and stuff classes in context. In *arXiv:cs-CV/1612.03716*, 2016.

[6] Rich Caruana. Multitask learning. In *Learning to learn*, pages 95–133. Springer, 1998.

[7] Rich Caruana. Multitask learning. In *Learning to learn*, pages 95–133. Springer, 1998.

[8] Arslan Chaudhry, Puneet Kumar Dokania, and Philip H.S. Torr. Discovering class-specific pixels for weakly-supervised semantic segmentation. *Proceedings of the British Machine Vision Conference*, 2017.

[9] L. C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PP(99), 2017.

[10] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. In *arXiv:cs-CV/170i7.05821*, 2017.

[11] Jie-Zhi Cheng, Dong Ni, Yi-Hong Chou, Jing Qin, Chui-Mei Tiu, Yeun-Chung Chang, Chiun-Sheng Huang, Dinggang Shen, and Chung-Ming Chen. Computer-aided diagnosis with deep learning architecture: Applications to breast lesions in us images and pulmonary nodules in ct scans. *Nature Scientific Reports*, 2016.

[12] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The Cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[13] Jifeng Dai, Kaiming He, and Jian Sun. Instance-aware semantic segmentation via multi-task network cascades. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[14] J. Deng, W. Dong, R. Socher, L. J. Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009.

[15] Emily L Denton, Soumith Chintala, arthur szlam, and Rob Fergus. Deep generative image models using a Laplacian pyramid of adversarial networks. In *NIPS*, 2015.

[16] Mandar Dixit, Roland Kwitt, Marc Niethammer, and Nuno Vasconcelos. AGA: Attribute guided augmentation. In *CVPR*, 2017.

[17] Manfred Eigen. Selforganization of matter and the evolution of biological macromolecules. *Naturwissenschaften*, 58(10):465–523, 1971.

[18] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2012 (VOC2012), 2010.

[19] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *Proceedings of the International Conference on Machine Learning*, 2015.

[20] Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *Journal of Machine Learning Research*, 17(59):1–35, 2016.

[21] Ian Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *ICLR*, 2015.

[22] Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *CVPR*, 2006.

[23] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[24] G. E. Hinton and R. Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006.

[25] Judy Hoffman, Dequan Wang, Fisher Yu, and Trevor Darrell. FCNs in the Wild: Pixel-level adversarial and constraint-based adaptation. *arXiv:cs-CV/1612.02649*, 2016.

[26] Chen Huang, Chen Change Loy, and Xiaoou Tang. Local similarity-aware deep feature embedding. In *NIPS*, 2016.

[27] Shiyu Huang and Deva Ramanan. Expecting the unexpected: Training detectors for unusual pedestrians with adversarial imposters. In *CVPR*, 2017.

[28] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the 22nd ACM International Conference on Multimedia*, pages 675–678, 2014.

[29] Seong Joon Oh, Rodrigo Benenson, Anna Khoreva, Zeynep Akata, Mario Fritz, and Bernt Schiele. Exploiting saliency for object segmentation from image level labels. In *Proceedings of the IEEE Conferene on Computer Vision and Pattern Recognition*, 2017.

[30] P. Kaiser, J. D. Wegner, A. Lucchi, M. Jaggi, T. Hofmann, and K. Schindler. Learning aerial image segmentation from online maps. *IEEE Transactions on Geoscience and Remote Sensing*, 55(11):6054–6068, 2017.

[31] Mahdi M. Kalayeh, Boqing Gong, and Mubarak Shah. Improving facial attribute prediction using semantic segmentation. In *CVPR*, 2017.

[32] Ronald Kemker and Christopher Kanan. Algorithms for semantic segmentation of multispectral remote sensing imagery using deep learning. In *arXiv:cs-CV/1703.06452*, 2017.

[33] Anna Khoreva, Rodrigo Benenson, Jan Hosang, Matthias Hein, and Bernt Schiele. Simple does it: Weakly supervised instance and semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[34] Donald E Knuth. Semantics of context-free languages. *Mathematical Systems Theory*, 2(2):127–145, 1968.

[35] Iasonas Kokkinos. UberNet: Training a universal convolutional neural network for low-, mid-, and high-level vision using diverse datasets and limited memory. In *CVPR*, 2017.

[36] Iasonas Kokkinos. Ubernet: Training a Universal Convolutional Neural Network for Low-, Mid-, and High-Level Vision Using Diverse Datasets and Limited Memory. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[37] B. Kulis, K. Saenko, and T. Darrell. What you saw is not what you get: Domain adaptation using asymmetric kernel transforms. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2011.

[38] Neeraj Kumar, Peter Belhumeur, and Shree Nayar. Facetracer: A search engine for large collections of images with faces. In *ECCV*, 2008.

[39] Yi Li, Haozhi Qi, Jifeng Dai, Xiangyang Ji, and Yichen Wei. Fully convolutional instance-aware semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[40] Guosheng Lin, Anton Milan, Chunhua Shen, and Ian Reid. RefineNet: Multi-path refinement networks for high-resolution semantic

segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[41] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. COCO: Common objects in context. In *Proceedings of the European Conference on Computer Vision*, 2014.

[42] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *ICCV*, 2015.

[43] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[44] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael I. Jordan. Learning transferable features with deep adaptation networks. In *Proceedings of the International Conference on Machine Learning*, 2015.

[45] Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I. Jordan. Unsupervised domain adaptation with residual transfer networks. In *Advances in Neural Information Processing Systems*, 2016.

[46] Y. Long, Y. Gong, Z. Xiao, and Q. Liu. Accurate object localization in remote sensing images based on convolutional neural networks. *IEEE Transactions on Geoscience and Remote Sensing*, 55(5):2486–2498, 2017.

[47] Gellert Mattyus, Shenlong Wang, Sanja Fidler, and Raquel Urtasun. Enhancing road maps by parsing aerial images around the world. In *Proceedings of the IEEE International Conference on Computer Vision*, 2017.

[48] R. Mottaghi, X. Chen, X. Liu, N.-G. Cho, S.-W. Lee, S. Fidler, R. Urtasun, and A. Yuille. The role of context for object detection and semantic segmentation in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014.

[49] Gerhard Neuhold, Tobias Ollmann, Samuel Rota Bulo, and Peter Kontschieder. The mapillary vistas dataset for semantic understanding of street scenes. In *Proceedings of the IEEE International Conference on Computer Vision*, 2017.

[50] Hyun Oh Song, Yu Xiang, Stefanie Jegelka, and Silvio Savarese. Deep metric learning via lifted structured feature embedding. In *CVPR*, 2016.

[51] S. J. Pan and Q. Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359, 2010.

[52] G. Papandreou, L.-C. Chen, K. P. Murphy, and A. L. Yuille. Weakly-and semi-supervised learning of a deep convolutional network for semantic image segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*, 2015.

[53] Emilio Parisotto, Jimmy Lei Ba, and Ruslan Salakhutdinov. Actor-mimic: Deep multitask and transfer reinforcement learning. *arXiv preprint arXiv:1511.06342*, 2015.

[54] Adam Paszke, Abhishek Chaurasia, Sangpil Kim, and Eugenio Culurciello. Enet: A deep neural network architecture for real-time semantic segmentation. *arXiv:cs-CV/1606.02147*, 2016.

[55] Scott Reed, Kihyuk Sohn, Yuting Zhang, and Honglak Lee. Learning to disentangle factors of variation with manifold interaction. In *ICML*, 2014.

[56] Alexandre Robicquet, Amir Sadeghian, Alexandre Alahi, and Silvio Savarese. Learning social etiquette: Human trajectory understanding in crowded scenes. In *Proceedings of the European Conference on Computer Vision*, 2016.

[57] German Ros, Laura Sellart, Joanna Materzynska, David Vazquez, and Antonio Lopez. The SYNTHIA Dataset: A large collection of synthetic images for semantic segmentation of urban scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[58] Ethan M. Rudd, Manuel Günther, and Terrance E. Boult. MOON: A mixed objective optimization network for the recognition of facial attributes. In *ECCV*, 2016.

[59] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

[60] Sara Sabour, Yanshuai Cao, Fartash Faghri, and David J Fleet. Adversarial manipulation of deep representations. In *ICLR*, 2016.

[61] Ruslan Salakhutdinov and Geoffrey E Hinton. Learning a nonlinear embedding by preserving class neighbourhood structure. In *AISTATS*, 2007.

[62] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *CVPR*, 2015.

[63] Edgar Simo-Serra, Eduard Trulls, Luis Ferraz, Iasonas Kokkinos, Pascal Fua, and Francesc Moreno-Noguer. Discriminative learning of deep convolutional feature point descriptors. In *ICCV*, 2015.

[64] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *Proceedings of the International Conference on Learning Representations*, 2015.

[65] K. Sirinukunwattana, S. E. A. Raza, Y. W. Tsang, D. R. J. Snead, I. A. Cree, and N. M. Rajpoot. Locality sensitive deep learning for detection and classification of nuclei in routine colon cancer histology images. *IEEE Transactions on Medical Imaging*, 35(5):1196–1206, 2016.

[66] Kihyuk Sohn. Improved deep metric learning with multi-class N-pair loss objective. In *NIPS*, 2016.

[67] Marvin Teichmann, Michael Weber, Marius Zoellner, Roberto Cipolla, and Raquel Urtasun. MultiNet: Real-time joint semantic reasoning for autonomous driving. *arXiv:cs-CV/1612.07695*, 2016.

[68] E. Tzeng, J. Hoffman, T. Darrell, and K. Saenko. Simultaneous deep transfer across domains and tasks. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2015.

[69] Andreas Veit, Serge Belongie, and Theofanis Karaletsos. Conditional similarity networks. In *CVPR*, 2017.

[70] Jing Wang, Yu Cheng, and Rogério Schmidt Feris. Walk and learn: Facial attribute representation learning from egocentric video and contextual data. In *CVPR*, 2016.

[71] Xiaolong Wang, Abhinav Shrivastava, and Abhinav Gupta. A-Fast-RCNN: Hard positive generation via adversary for object detection. In *CVPR*, 2017.

[72] Yunchao Wei, Jiashi Feng, Xiaodan Liang, Ming-Ming Cheng, Yao Zhao, and Shuicheng Yan. Object region mining with adversarial erasing: A simple classification to semantic segmentation approach. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[73] Michael J. Wilber, Iljung S. Kwak, David Kriegman, and Serge Belongie. Learning concept embeddings with combined human-machine expertise. In *ICCV*, 2015.

[74] Chao-Yuan Wu, R. Manmatha, Alexander J. Smola, and Philipp Krähenbühl. Sampling matters in deep embedding learning. *arXiv preprint*, arXiv:1706.07567, 2017.

[75] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson. How transferable are features in deep neural networks? In *Advances in Neural Information Processing Systems*, 2014.

[76] A. Yu and K. Grauman. Fine-grained visual comparisons with local learning. In *CVPR*, 2014.

[77] Yuhui Yuan, Kuiyuan Yang, and Chao Zhang. Hard-aware deeply cascaded embedding. In *ICCV*, 2017.

[78] Ning Zhang, M. Paluri, M. Ranzato, T. Darrell, and L. Bourdev. PANDA: Pose aligned networks for deep attribute modeling. In *CVPR*, 2014.

[79] Xiaofan Zhang, Feng Zhou, Yuanqing Lin, and Shaoting Zhang. Embedding label structures for fine-grained feature representation. In *CVPR*, 2016.

[80] Zhanpeng Zhang, Ping Luo, Chen Change Loy, and Xiaoou Tang. Facial landmark detection by deep multi-task learning. In *ECCV*, 2014.

[81] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[82] Stephan Zheng, Yang Song, Thomas Leung, and Ian Goodfellow. Improving the robustness of deep neural networks via stability training. In *CVPR*, 2016.

[83] Yang Zhong, Josephine Sullivan, and Haibo Li. Face attribute prediction with classification CNN. In *ICIP*, 2016.

[84] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ADE20k dataset. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.