Domain Adaptation

Domain Adaptation (DA) allows machine learning methods trained on data sampled from one distribution to be applied to data sampled from another.

One-Step DA

The source and target domains are directly related, allowing for knowledge transfer in one step.

Multi-Step DA

Build a series of intermediate bridges to connect two seemingly unrelated domains, then perform one-step DA via this bridge.

One-Step Domain Adaptation

One-step domain adaptation can be categorised into three approaches, each of which can be further split into subcategories as shown in table I.

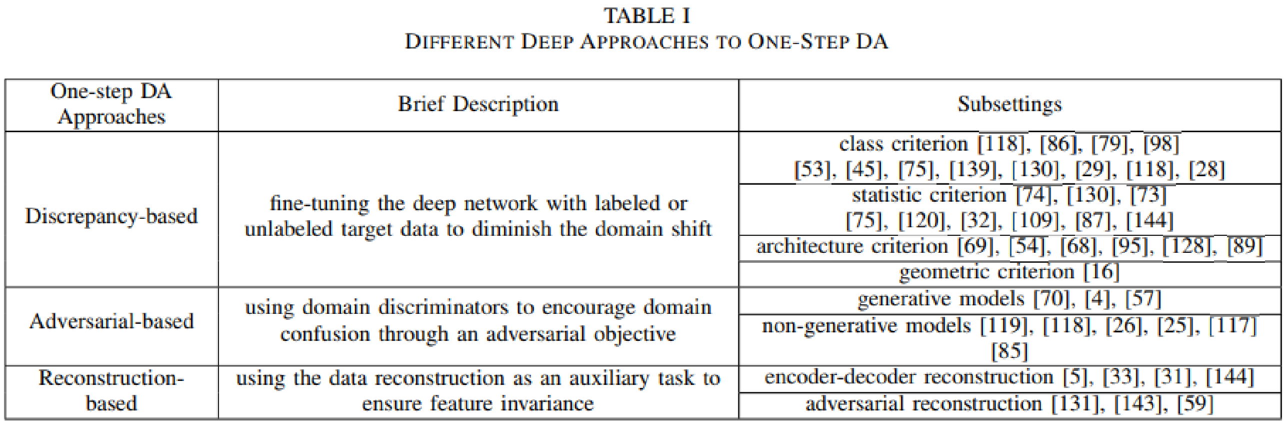

TABLE I
DIFFERENT DEEP APPROACHES TO ONE-STEP DA

One-step DA Approaches	Brief Description	Subsettings
Discrepancy-based	fine-tuning the deep network with labeled or unlabeled target data to diminish the domain shift	class criterion [118], [86], [79], [98] [53], [45], [75], [139], [130], [29], [118], [28]
		statistic criterion [74], [130], [73] [75], [120], [32], [109], [87], [144]
		architecture criterion [69], [54], [68], [95], [128], [89]
		geometric criterion [16]
Adversarial-based	using domain discriminators to encourage domain confusion through an adversarial objective	generative models [70], [4], [57]
		non-generative models [119], [118], [26], [25], [117] [85]
Reconstruction-based	using the data reconstruction as an auxiliary task to ensure feature invariance	encoder-decoder reconstruction [5], [33], [31], [144]
		adversarial reconstruction [131], [143], [59]

1.1 Discrepancy-Based DA

Assumes that fine-tuning the deep network model with labelled or unlabelled target data can diminish the shift between the two domains.

1.1.1 Class Criterion

Uses the class label information as a guide for transferring knowledge between different domains. When such samples are unavailable, some other techniques can be adopted to substitute for class labelled data, such as pseudo labels and attribute representation.

Soft Label Loss

Introduced by Geoff Hinton (of course), a "soft" softmax can be used when training on the new domain as shown in equation 1.1, where T is the temperature that's normally set to 1 in standard softmax. Larger values of T produce a softer probability distribution over classes.

$$q_i = \frac{exp(z_j/T)}{\sum_j exp(z_j/T)} \tag{1.1}$$

Originally used in a paper about distilling knowledge in other networks, using soft labels rather than hard labels can preserve relationships between classes across domains. It is recommended that this is used in conjunction with the domain confusion loss (discussed later).

From [11]:

> *The bottle soft label will have a higher weight on mug than on keyboard, since bottles and mugs are more visually similar. Thus, soft label training with this particular soft label directly enforces the relationship that bottles and mugs should be closer in feature space than bottles and keyboards.*
>
> *... we ensure that the parameters for categories without any labelled target data are still updated to output non-zero probabilities.*

Embedded Metric Learning

Unified Deep Supervised Domain Adaptation and Generalization[7] uses a siamese network to process source and target domain examples simultaneously, and applies a *Contrastive Semantic Alignment Loss* to the embedded examples, minimising the dissimilarity across domains.

Deep Transfer Metric Learning[4] demonstrates an unsupervised technique for transforming samples into a new subspace. Their training objectives are that:

1. the inter-class variations are maximised and the intra-class variations are minimised.
2. the distribution divergence between the source domain and the target domain at the top layer of the network is minimised.

1.1.2 Statistic Criterion

> *Aligns the statistical distribution shift between the source and target domains using some mechanisms.*

A frequently used metric is **Maximum Mean Discrepancy (MMD)**, which estimates the distance between different distributions.

Deep Domain Confusion: Maximizing for Domain Invariance[10] applies an *adaptation layer* (bottleneck) at some point in the network, then jointly minimises classification error while maximising MMD at the adaptation layer. The minimisation of MMD between source and target domains is known as **domain confusion**.

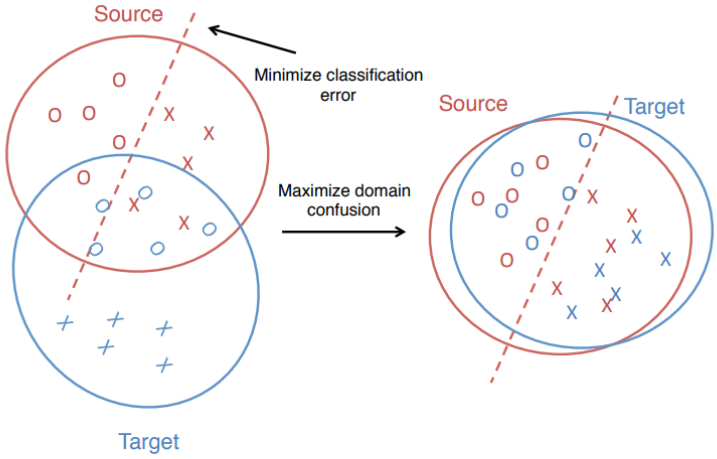

Figure 1.1: By jointly minimising classification error while maximising domain confusion, we learn representations that are discriminative and domain invariant

Learning Transferable Features with Deep Adaptation Networks[6] takes it one step further by applying domain confusion to several layers of a network.

Deep CORAL[9] is similar to the previous systems, but instead of minimising MMD, minimises the difference in second-order statistics between the source and target feature activations (the learned feature covariances). Similarly, this can be applied to any layer of a network.

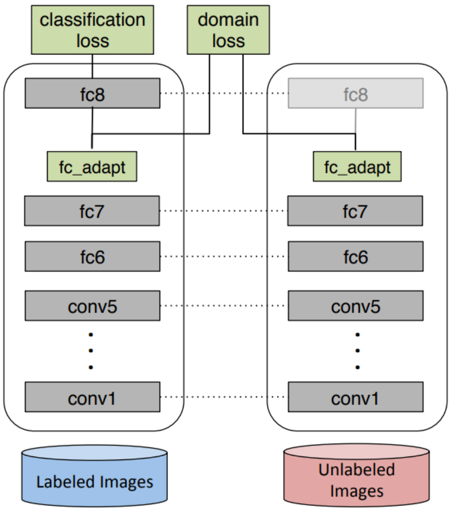

Figure 1.2: Architecture of Deep Domain Confusion[10]

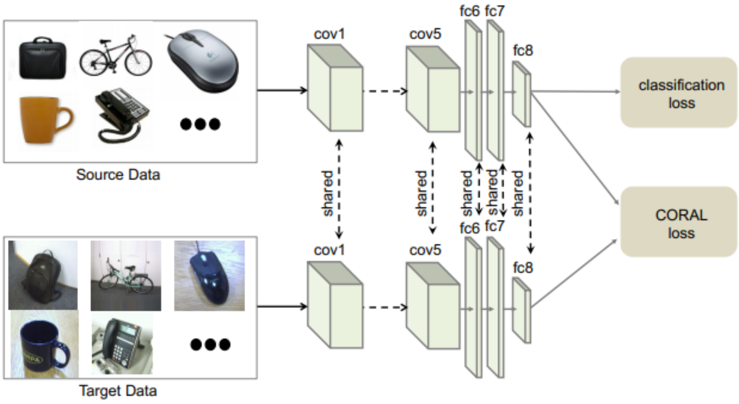

Figure 1.3: Deep CORAL architecture[9]

1.1.3 Architecture Criterion

> *Aims at improving the ability of learning more transferable features by adjusting the architectures of deep networks. The techniques that are proven to be cost effective include adaptive batch normalisation (BN), weak-related weight and domain-guided dropout.*

Instead of transfer-learning by fine-tuning weights to a target domain, *Beyond Sharing Weights for Deep Domain Adaptation*[8] adopts a dual-stream network architecture where select layers in the target classifier has distinct weights, which are regularised such that they have a similar distribution as those in the source domain model. They also say *"...the class-related knowledge is stored in the weight matrix, whereas domain-related knowledge is represented by the statistics of the batch normalization (BN) layer"*

Revisiting Batch Normalization For Practical Domain Adaptation[5] hypothesizes that "the label related knowledge is stored in the weight matrix of each layer, whereas domain related knowledge is represented by the statistics of the Batch Normalization layer". They recommend the usage of **Adaptive Batch Normalization** to normalise features between domains.

AutoDIAL: Automatic DomaIn Alignment Layers[1] feeds source/target domain images into a network in parallel and inserts **Domain Alignment (DA) layers** (similar to batch norm) to enforce feature similarity across domains. A learned parameter controls this "degree of domain alignment" as features pass through each of the DA layers.

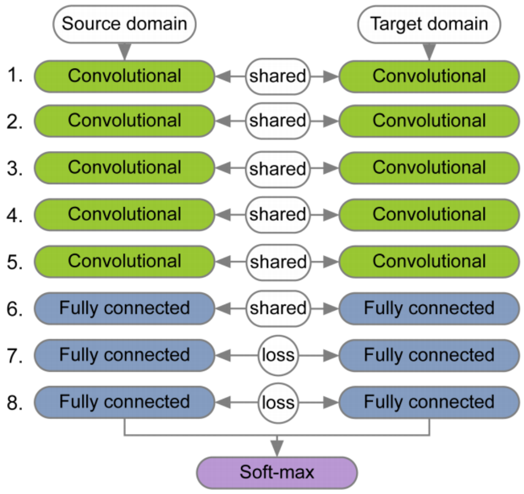

Figure 1.4:

The survey[13] states that *Improved Texture Networks*[12] shows **Instance Normalisation** performs better at DA than standard Batch Normalisation... but I couldn't find that in the paper.

Learning Deep Feature Representations with Domain Guided Dropout for Person Re-identification[14] introduces **Domain Guided Dropout**, which is essentially a method of applying dropout to irrelevant/unhelpful neurons per-domain. It's used like so:

1. Train on all domains
2. For each entry in a feature vector for an image:
3. The impact score of this neuron is $\mathcal{L}(g(x)_{\backslash i}) - \mathcal{L}(g(x))$, where $g(x)_{\backslash i}$ is the feature vector after zeroing-out the i-th neuron response to zero, averaged over all images in each domain
4. Continue training, but use the scores to guide dropout for each domain

This comes in two flavours: deterministic and stochastic. Deterministic applies the resultant mask when the score is < 0, stochastic uses the score to sample from a Bernoulli distribution.

1.1.4 Geometric Criterion

> *Bridges the source and target domains according to their geometrical properties. This criterion assumes that the relationship of geometric structures can reduce the domain shift.*

1.2 Adversarial-Based DA

1.2.1 Generative Models

> *The typical case is to use source images, noise vectors or both to generate simulated samples that are similar to the target samples and preserve the annotation information of the source domain*

Cross-Domain Weakly-Supervised Object Detection through Progressive Domain Adaptation[3] uses two techniques to generate target domain examples. Perform image-to-image translation from the source to target domain using CycleGAN = accurate bounding boxes, inaccurate visual features. Perform object detection to build "pseudo-labels" = inaccurate bounding boxes, accurate visual features.

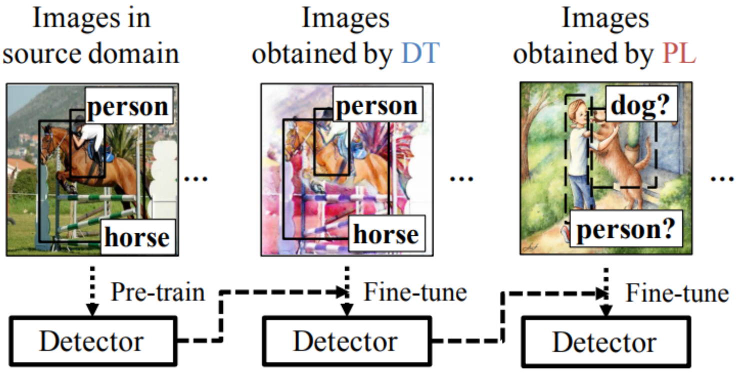

Figure 1.5: A three-step training approach to go from source to target images. DT = Domain Transfer GAN; PL = Pseudo-Labels (from detector)

1.2.2 Non-Generative Models

> *The feature extractor learns a discriminative representation using the labels in the source domain and maps the target data to the same space through a domain-confusion loss.*

Domain Adaptive Faster R-CNN for Object Detection in the Wild[2] adds an **Image-Level Domain Classifier** which is trained to predict the domain from which an image is sampled. They reverse gradients at the boundary, to apply adversarial training.

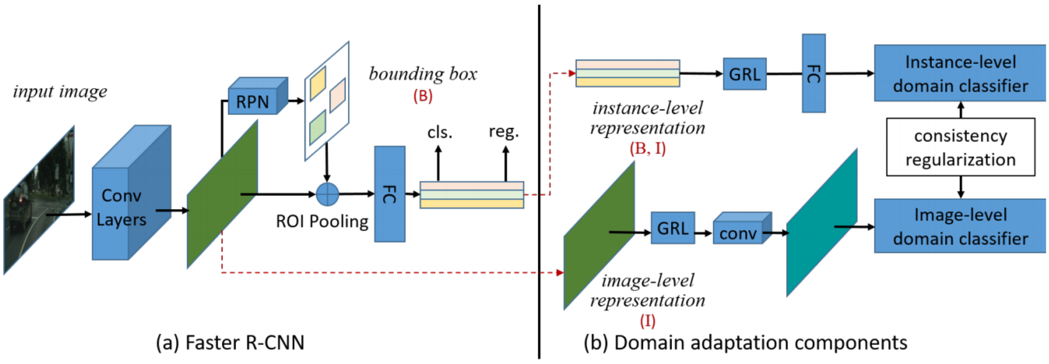
input image
Conv Layers
RPN
bounding box
(B)
cls.
reg.
ROI Pooling
FC
(a) Faster R-CNN
instance-level representation
(B, I)
GRL
FC
Instance-level domain classifier
consistency regularization
Image-level domain classifier
GRL
conv
image-level representation
(I)
(b) Domain adaptation components

Figure 1.6: (a): Faster R-CNN; (b, upper): a Faster R-CNN-specific DA solution; (b, lower): conv-net used to predict image domain, with gradient-reversal-layer (GRL) at the boundary

1.3 Reconstruction-Based DA

> *Assumes that the data reconstruction of the source or target samples can be helpful for improving the performance of DA*

1.3.1 Encoder-Decoder Reconstruction

> *By using stacked autoencoders (SAEs), encoder-decoder reconstruction methods combine the encoder network for representation learning with a decoder network for data reconstruction*

1.3.2 Adversarial Reconstruction

> *The reconstruction error is measured as the difference between the reconstructed and original images within each image domain by a cyclic mapping obtained via a GAN discriminator*

Multi-Step Domain Adaptation

Multi-step domain adaptation can be categorised into three approaches as shown in table II.

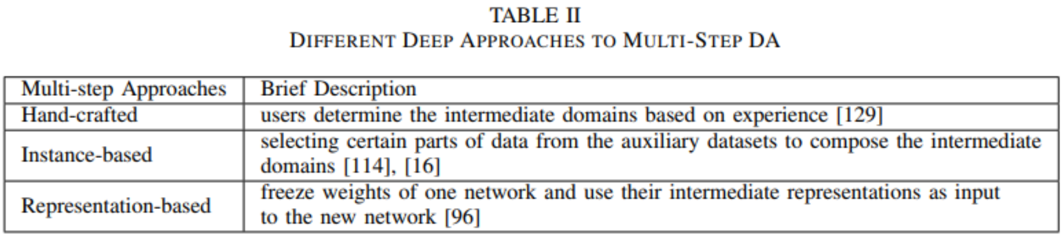

TABLE II
Different Deep Approaches to Multi-Step DA

Multi-step Approaches	Brief Description
Hand-crafted	users determine the intermediate domains based on experience [129]
Instance-based	selecting certain parts of data from the auxiliary datasets to compose the intermediate domains [114], [16]
Representation-based	freeze weights of one network and use their intermediate representations as input to the new network [96]

2.3.3 Hand-Crafted

Users determine the intermediate domains based on experience.

2.3.4 Instance-Based

Selecting certain parts of data from the auxiliary datasets to compose the intermediate domains to train the deep network

2.3.5 Representation-Based

Transfer is enabled via freezing the previously trained network and using their intermediate representations as input to the new one.

Application to Swimming Project