Introduction to (Unsupervised) Domain Adaptation

Peeyush Singhal
March 13, 2023

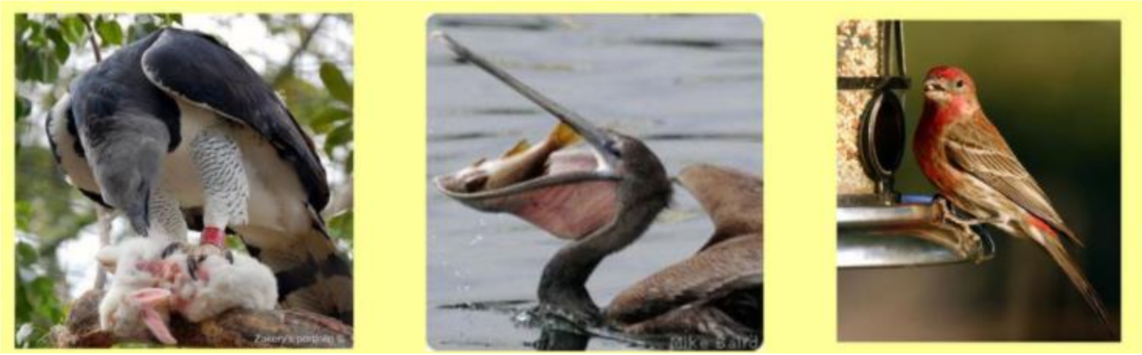
Excepts from

P. Singhal, R. Walambe, S. Ramanna and K. Kotecha, "Domain Adaptation: Challenges, Methods, Datasets, and Applications," in *IEEE Access*, vol. 11, pp. 6973-7020, 2023, doi: 10.1109/ACCESS.2023.3237025.

Outline

- Motivation and Introduction
  - Transfer Learning and Domain Adaptation
  - Examples
  - Important Aspects
- Toy Example
- Domain Adaptation in Map making : Discussion
  - Reduction of Models
  - Reduction in labeling

Introduction



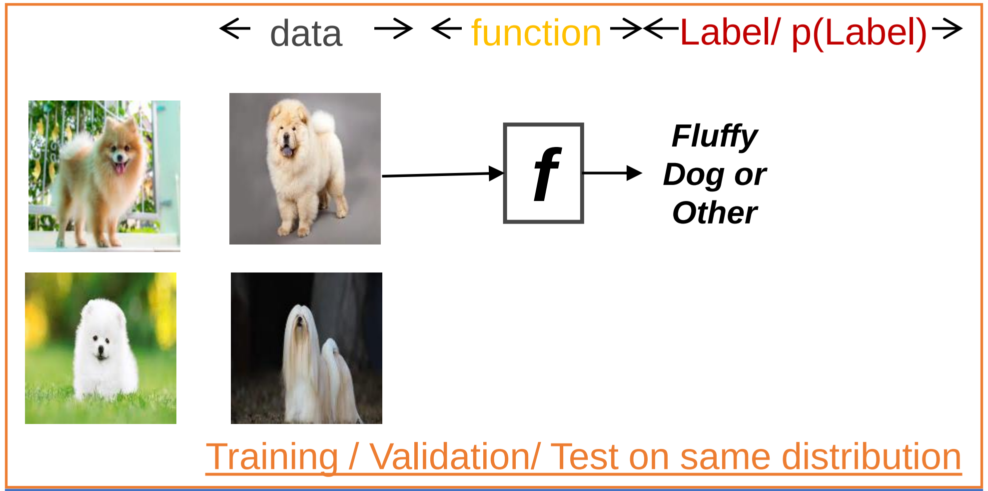
- Leon C. Megginson summed up (Darwin, 1859) by saying, “It is not the strongest of the species that survives, not the most intelligent that survives. It is the one that is most adaptable to change”
- Adaptation is everywhere in nature
- Transformation to do a particular task is adaptation
- Typically adaptation yields better results than no adaptation
- Domain adaptation is adaptation to different domain (“similar”) data for same task

Pictures: https://slideplayer.com/slide/14898230/

Motivation – Deviation from i.i.d.

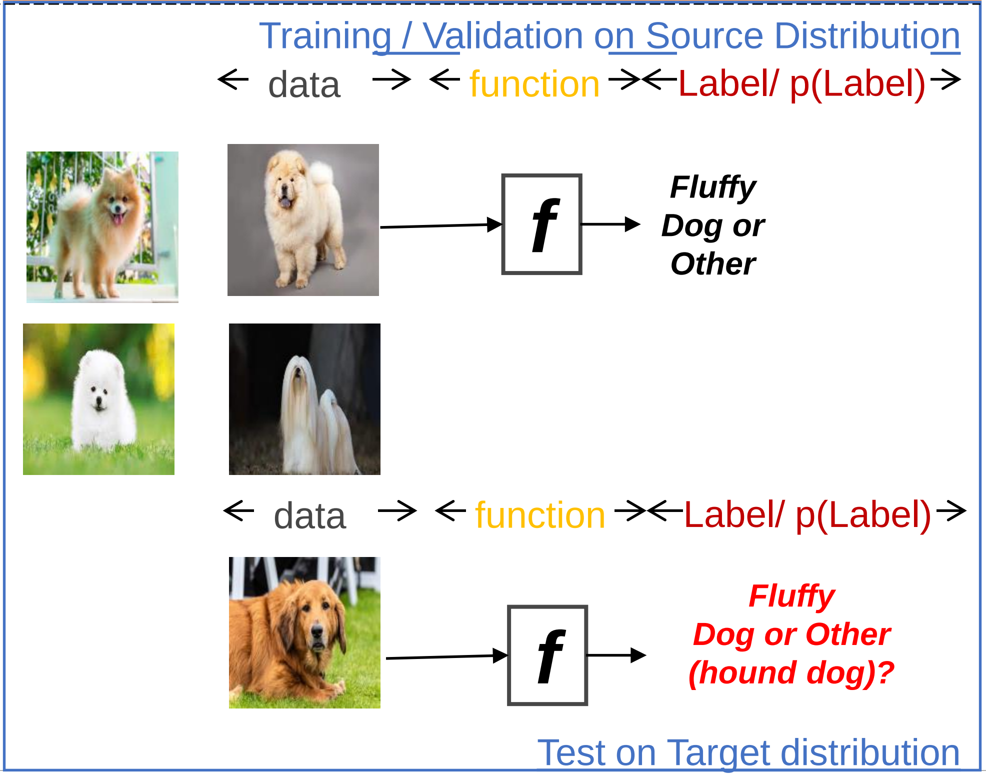
Typical Supervised Learning

- Find deterministic function $f : y = f(x)$, *where x : data, y : label*
- $x,y \sim D^{training}, D^{testing}$ i.e., $D^{training} = D^{testing}$
- $\mathcal{T}^{training} = \mathcal{T}^{testing}$
- Training, Validation and Test data are from the same distribution and typically i.i.d.



In Real Life / Operational scenarios

- Testing data does not come from the same distribution on which model was trained on.
- $D^{training} \neq D^{testing}$ and $\mathcal{T}^{training} = \mathcal{T}^{testing}$
- If the model is tested on out of distribution data, it suffers lower metrics (accuracy, precision, recall)
- There is a need for models to work on “similar” data
  - Data collection is expensive
  - Labelled / Annotated is time consuming
  - Sometimes, it is infeasible too e.g., Structural Health Monitoring of building – failure data for the same “type” of building is in feasible / Privacy issues



Examples of Domain Adaptation

- Spam filtering
  - Public email collection / one user → personal inboxes / another user
- Intrusion detection
  - Existing types of intrusions → unknown types of intrusions
- Sentiment analysis
  - Expert review articles→ blog review articles
- Healthcare equipment difference
  - Reducing variation between analysis of data sets of CT, MRI scanners associated with different vendor, calibration
- COVID-19
  - Applying AI diagnostic algorithms, trained on labeled data associated with previous diseases, to new unlabeled data associated with the COVID-19 pandemic.
  - A sudden societal change, such as a pandemic outbreak, can constitute domain shift and cause machine learning algorithms trained on now-obsolete consumer data to fail and require intervention.

Computer Vision Example - Office Home Dataset: Similar objects but differ in background, style

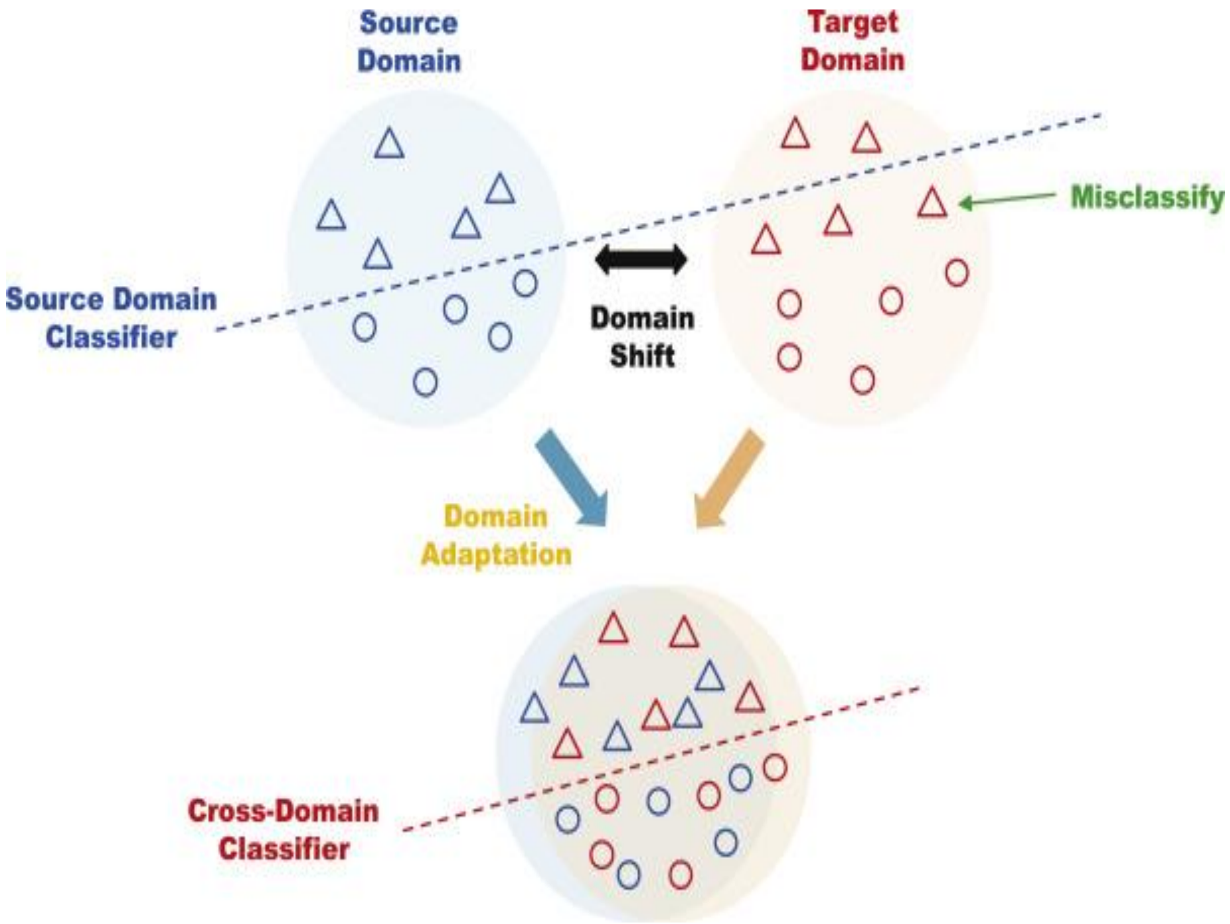
Important Aspects of Domain Adaptation

1. Domain adaptation is the ability to apply an algorithm trained in one or more "source domains" to a different (but ***related***) "target domain“

    1. It <u>cannot be applied to totally different datasets</u> – i.e. one with cats and one with horses

    2. The source and target domains all have the same feature space (but different distributions); in contrast, transfer learning includes cases where the target domain's feature space is different from the source feature space or spaces.

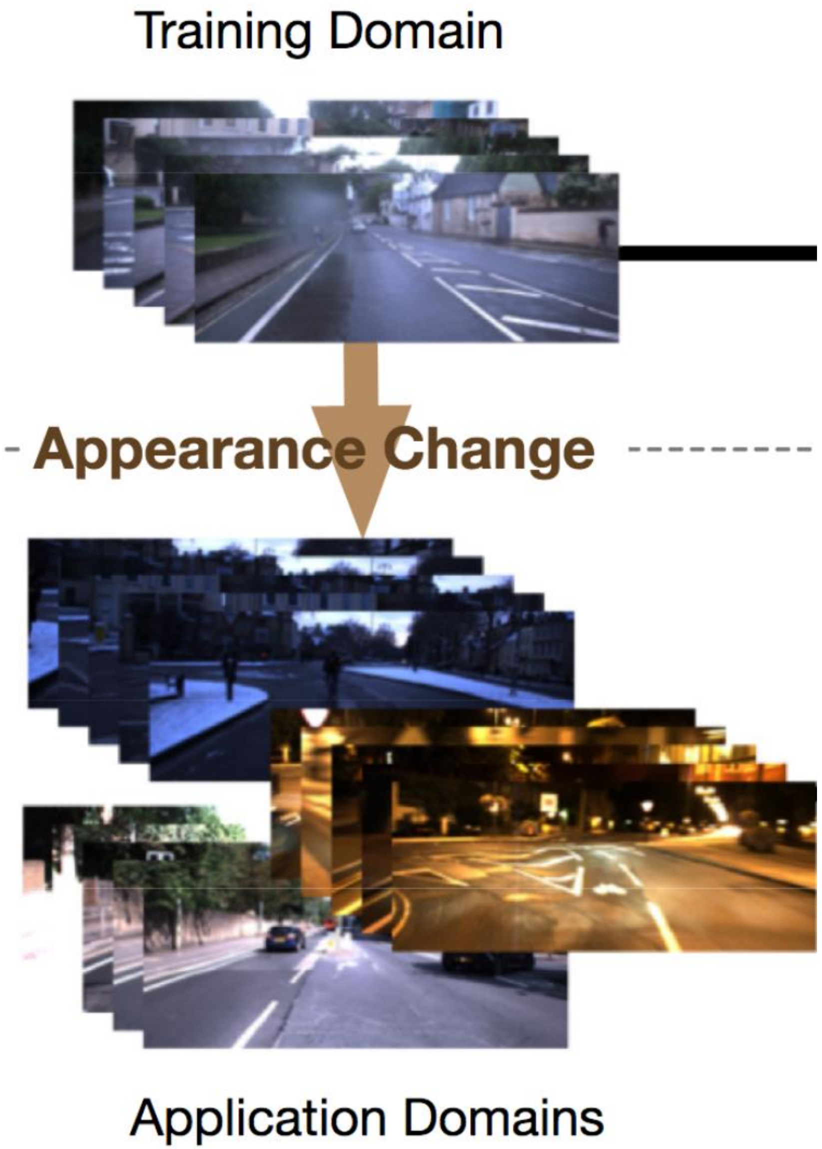
2. **Helpful in areas of domain shift / distributional shift** (change in the data distribution between an algorithm's training dataset, and a dataset it encounters when deployed)
3. Readily labeled data sets are biased / inclined for the task/field they perform best
4. Given, deep networks provide high accuracy (metric) and can model complex functions, it is essential to understand Domain Adaptation in Deep Networks
5. **Focus to understand features which are relevant for the task and not relevant for the domain**. These are also called domain invariant features



Source: Xu, Nanxi & Li, Xiang. (2021). Intelligent fault diagnosis methodology under varying operating conditions using multi-layer domain adversarial learning strategy. International Journal of Dynamics and Control. 9. 1-11. 10.1007/s40435-021-00760-0.

Domain Adaptation Application - Same Task, Different Domain

- A learning Domain $\mathcal{D}$, is defined as $\mathcal{D} = \{\chi, P(X)\}$ there can be two cases
  1. $\chi^s \neq \chi^t$
  2. $P(X^s) \neq P(X^t)$



Case 1 : $\chi^s \neq \chi^t$

This may be the case,
- Computer Vision : where χ^s is a space of grayscale images while χ^t is a space of colour images.
- NLP : χ^s, χ^t are two documents in different languages

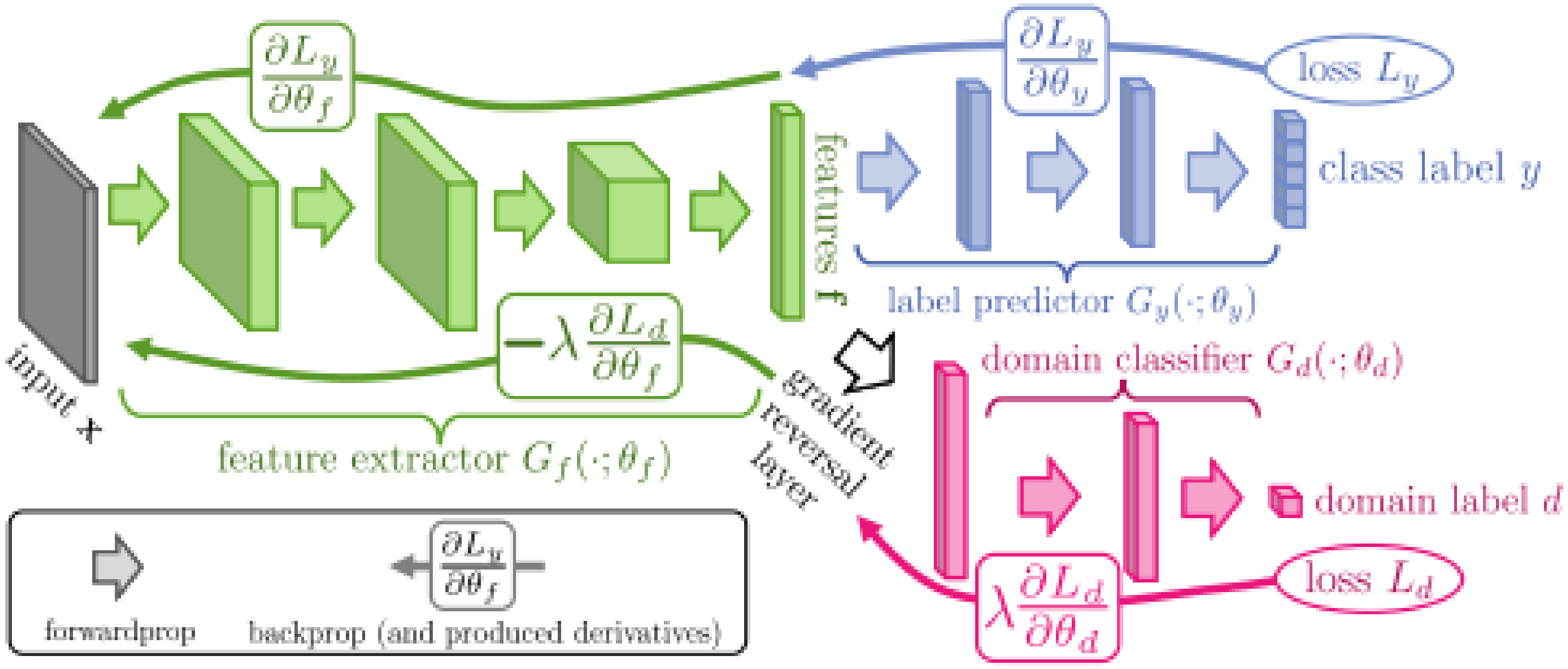
Case 2 :$P(X^s) \neq P(X^t)$

- Computer Vision : source domain contains hand-drawn images, while target domain contains photographs;
- NLP: documents in the same language about different topics

This is a very common scenario, and usually called **domain adaptation**.

Toy Example

Domain-Adversarial Training of Neural Networks (DANN) - Overview



- Trains feature extractor (green network) and class/label predictor (blue network) on source data.
- Trains feature extractor (green network) and domain classifier (pink network) on the source and target data.
- Use of Gradient Reversal layer to maximize domain confusion.

- Seminal work in domain adaptation
- Ability to support nearly all tasks – NLP, Computer Vision etc.
- First instance of using adversarial methods in domain adaptation

Domain-Adversarial Neural Network or DANN (Ganin, et al., 2016)
[https://arxiv.org/pdf/1505.07818.pdf]

Problem Statement – Unsupervised domain adaptation in Computer Vision

Let's consider a problem with different domains but an identical task:

- Source domain: MNIST
- Target domain: MNIST-M, a colored and textured version of MNIST

Task in both cases is the usual 10-class digit classification.

MNIST

SOURCE

TARGET

MNIST-M

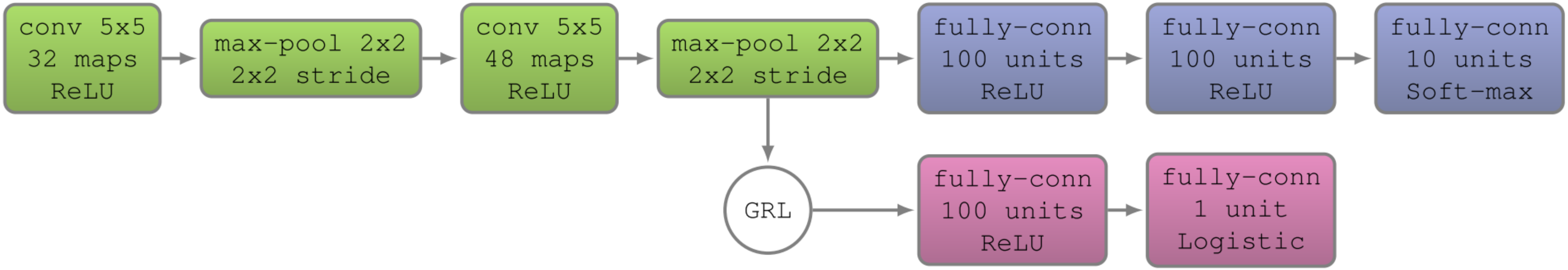
Unsupervised DA setting: We assume that there are **no available labels** for the target domain.
We need to force our CNN to learn features of the digits shapes only, not color distributions.

Our approach, (based on Ganin et al. 2015):
•Train a classifier for the **domain** of an image based on deep convolutional features.
•Try to maximize the loss of this classifier when training the CNN (**confusion loss**).
•Simultaneously, minimize the classification loss on the source domain using the same convolutional features.
•Train the digit classifier with source domain data, and the domain classifier with both domains' data.

Data Download for MNIST-M :
https://drive.google.com/open?id=0B_tExHiYS-0veklUZHFYT19KYjg

Data Download for MNIST :
Available in pytorch torchvision datasets

Implementation Overview



Our model will consist of three parts, as in the figure:

•A "deep" CNN for image feature extraction (2x Conv, ReLU, MaxPool)
•A digit-classification head (3x FC, ReLU)
•A domain classification head (2x FC, ReLU), with **gradient reversal layer** (GRL).

Thank You