SWINBURNE UNIVERSITY OF TECHNOLOGY

COS30082
Applied Machine Learning

Lecture 6

Transfer Learning for Computer Vision

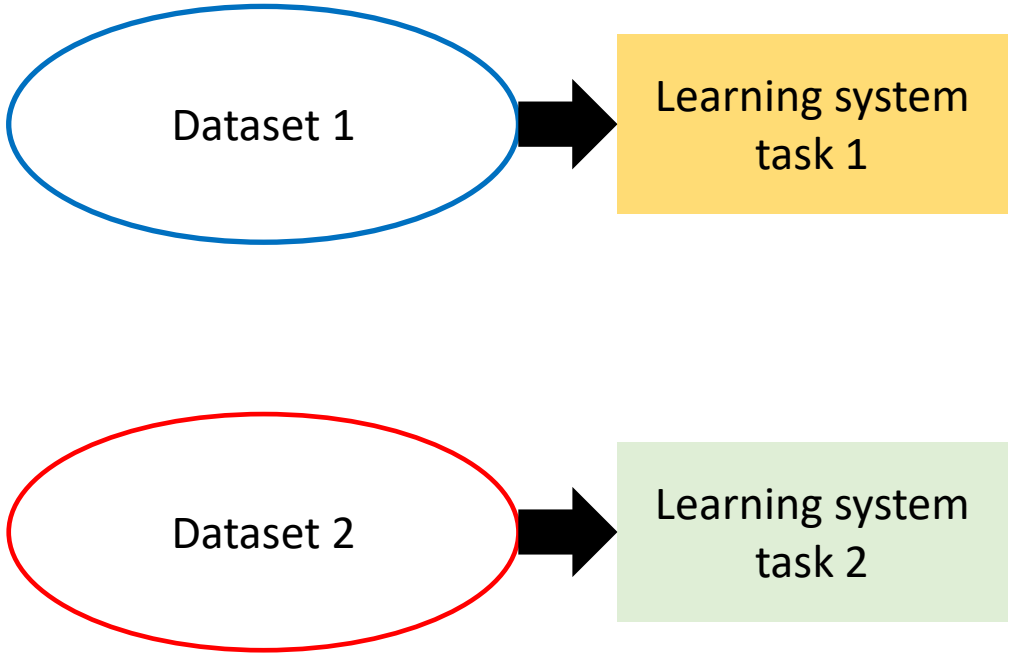
What is Transfer Learning?

- The ability of a system to recognize and apply knowledge and skills learned in previous tasks to novel tasks or new domains, which share some commonality.

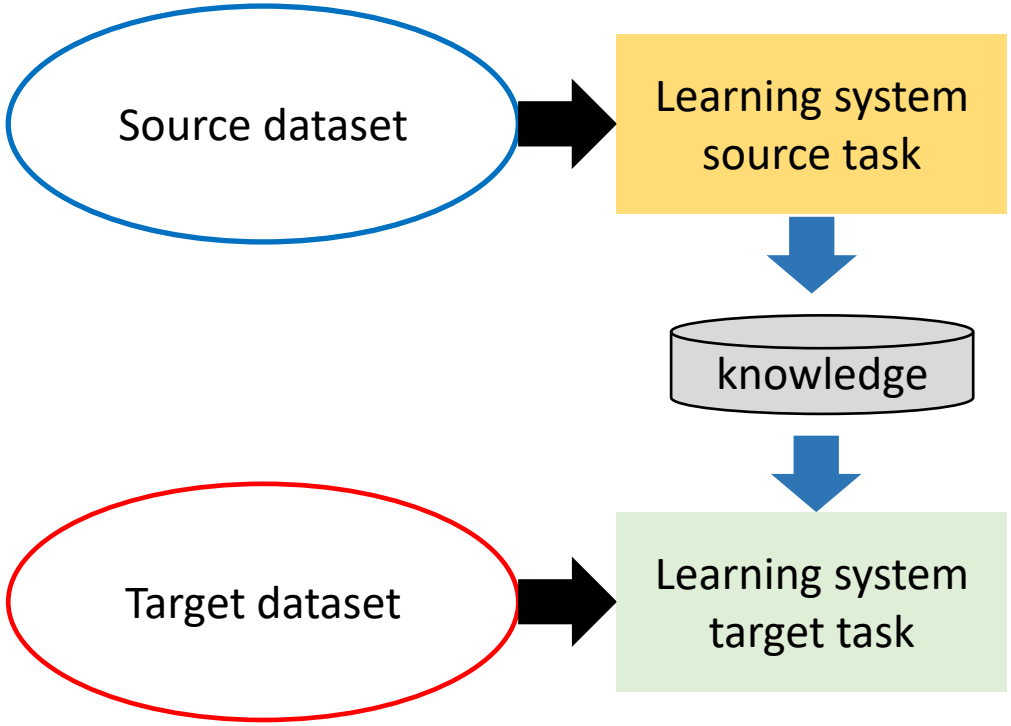
- Given a target task, how to identify the commonality between the task and previous (source) tasks, and transfer knowledge from the previous tasks to the target one?

Traditional ML vs Transfer Learning

- Isolated, single task
  - Knowledge is not retained
  - Learning does not take into account the knowledge acquired in the other task



- The learning of the target task relies on the source task
  - Faster learning process, more accurate and need less training data



Understanding of Transfer Learning

- Case study 1: Dogs classification

Dataset 1

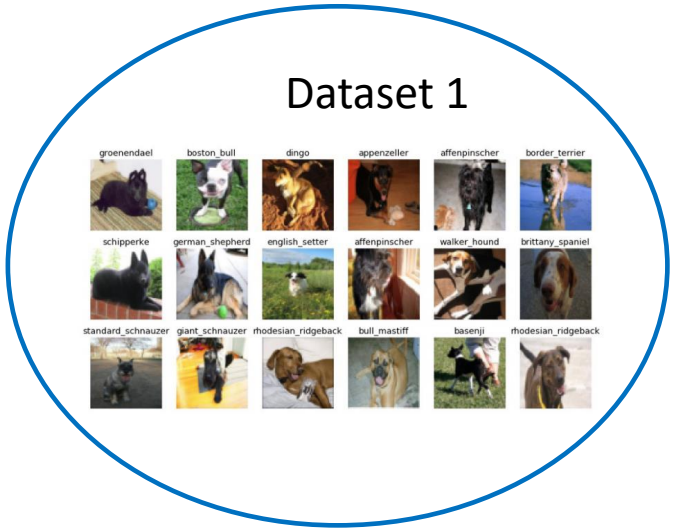
Learning system task 1

Understanding of Transfer Learning

- Case study 1: Dogs classification

Dataset 2

Understanding of Transfer Learning



Learning system task 1

Learning system task 1

- Performance degradation due to model that does not generalize well
  - model's bias towards training data.

Understanding of Transfer Learning

Dataset 1

Learning system task 1

Strategy 1: Retrain new model

Dataset 2

Learning system task 2

Dataset 1

Learning system task 1 (source)

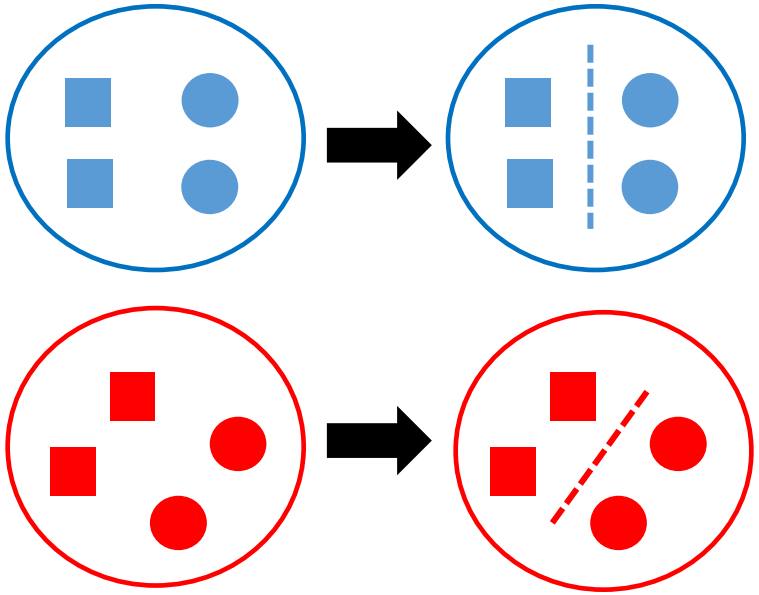
Strategy 2: Transfer learning

Dataset 2

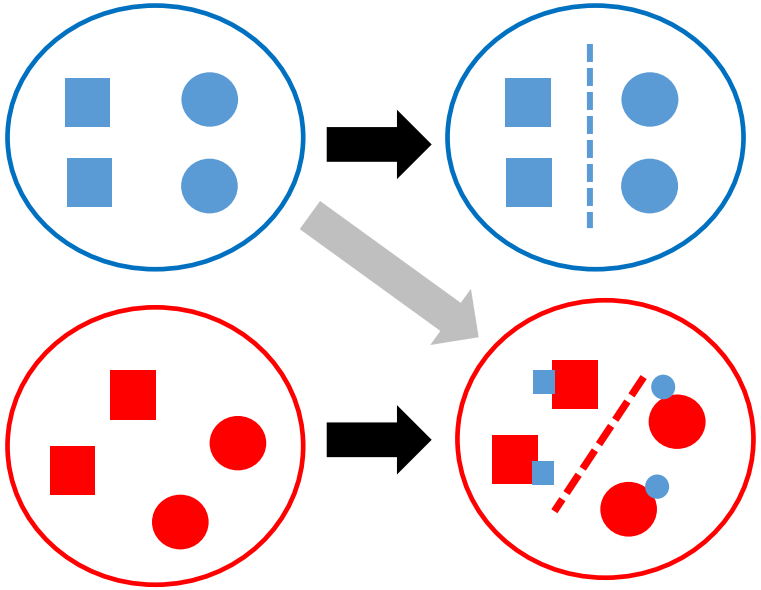
Learning system task 2 (target)

Understanding of Transfer Learning

Traditional ML



Transfer Learning



- The circles and squares are labeled training samples from two classes.
- The size of a circle or square indicates its weight
- The dotted and dashed lines are classification boundaries

The purpose of using Transfer Learning

- To leverage knowledge from previously trained models for training newer models
- To accelerate the learning process
- To improve generalization
- To tackle dataset bottlenecks
  - Reduce the need for data related to the target task

Key takeaways

- What to transfers?
  - Identify which part of the knowledge is source-specific and what is common between the source and the target.
  - Be careful of negative transfer
  - The choice of source data or source model is an open problem and may require domain expertise and/or intuition developed via experience.
- When to transfers?
  - Do not have a lot of target data
- How to transfer?
  - We will talk about transfer learning for Deep learning in the following slides

Transfer Learning for Deep learning

Myth ☹

' You can't do deep learning unless you have a million labelled examples for your problem '

Reality ☺

' You can transfer learned representation from related task'

Transfer Learning for Deep learning

SWINBURNE UNIVERSITY OF TECHNOLOGY SARAWAK CAMPUS

- Key idea:
  - Instead of training a deep network from scratch for the target task, we deploy the network that has already been trained on a different domain for a different **source task**, we adapt it to our domain and **target task**.

- Variation:
  - Same domain different task
  - Different domain same task

What is domain and task?

- If two domains are different, they may have different **feature spaces** or different marginal distributions.

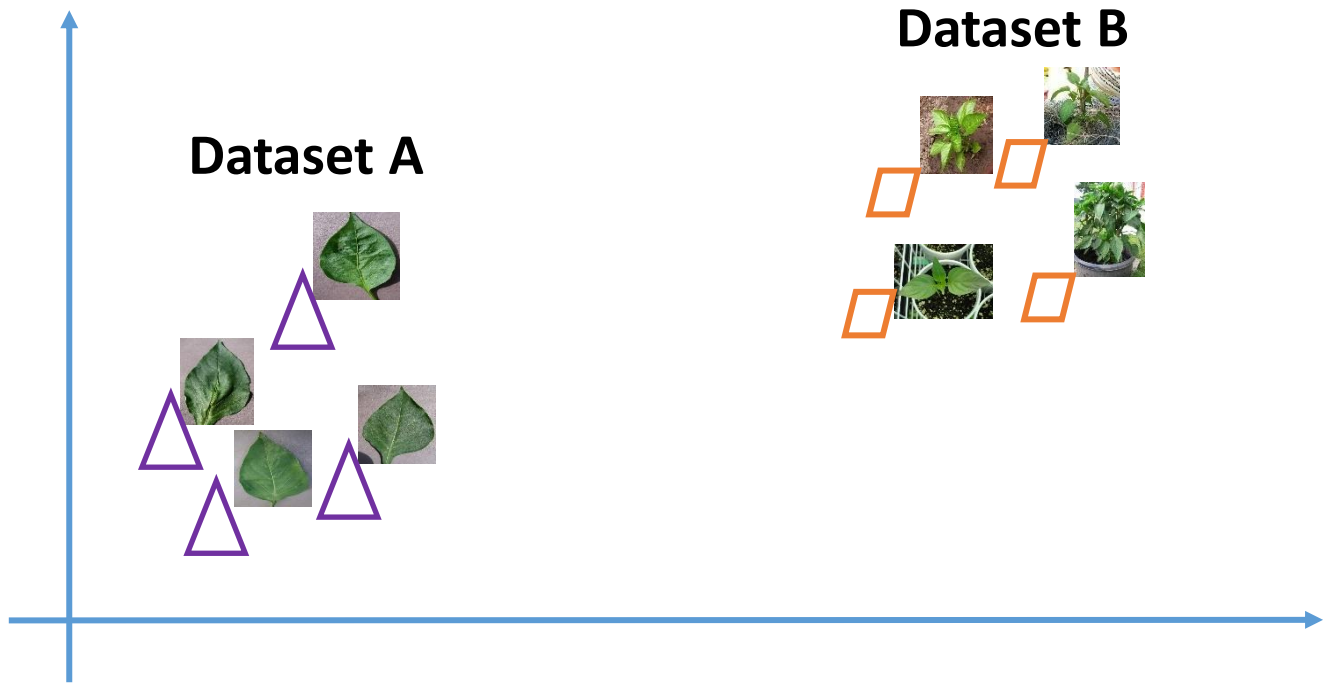
Bell pepper data

Dataset A

Dataset B

What is domain and task?

- If two domains are different, they may have different **feature spaces** or different marginal distributions**.**



What is domain and task?

- If two tasks are different, they may have different **label spaces** or different conditional distributions

Dataset A

Dataset B

Deep Transfer Learning strategies

- Off-the-shelf pre-trained models as feature extractors
- Fine tuning Off-the-shelf pre-trained models

What are pre-trained models?

- Pre-trained models are the details of networks shared by teams/individuals to be made available to all.

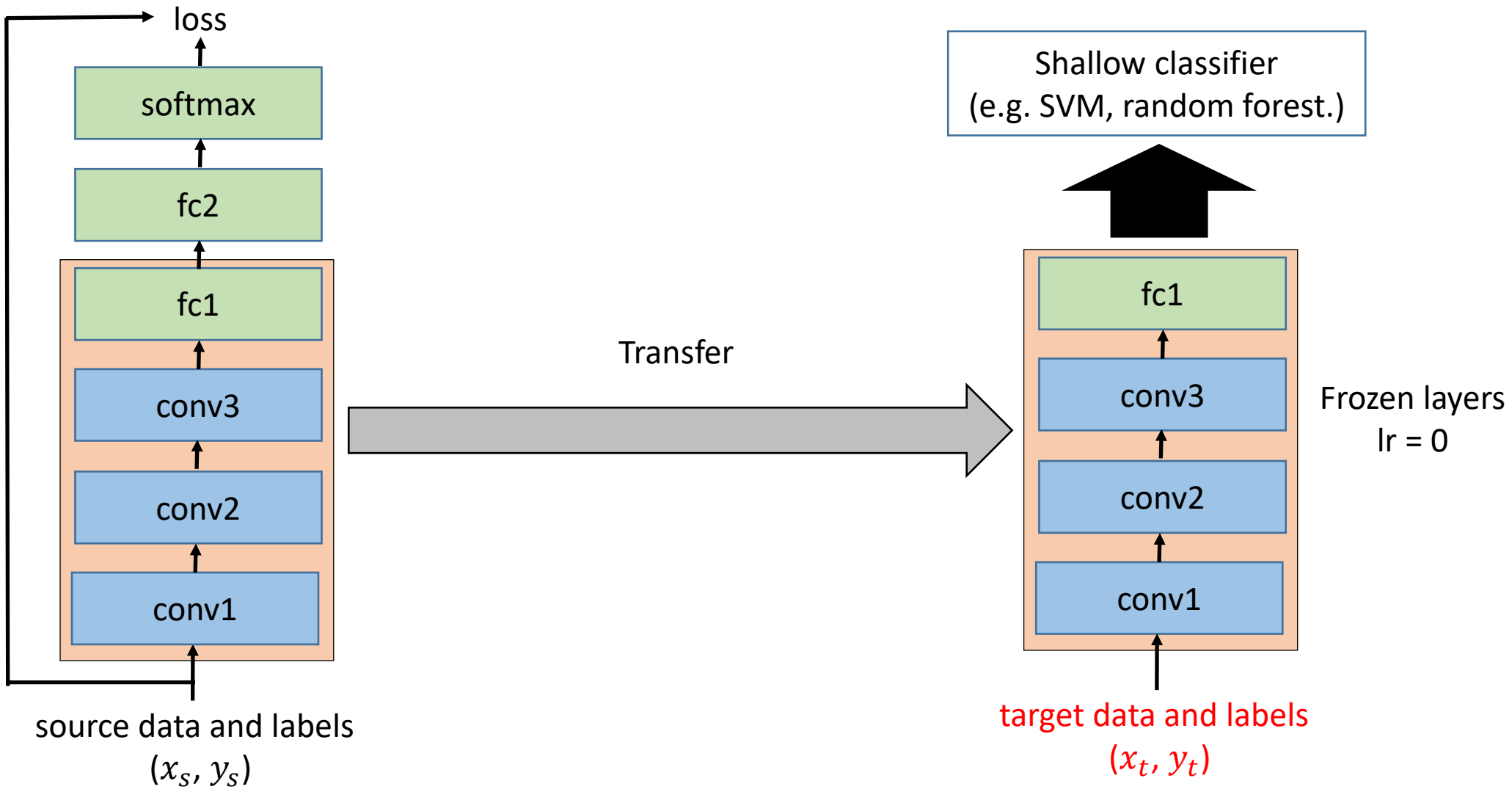
- Pre-trained models are usually shared in the form of the millions of parameters/weights the model achieved while being trained to a stable state.

Off-the-shelf pre-trained models

- The key idea here is to simply leverage the weighted layers of the pre-trained model to **extract features**, but not to update the weights of the model layers during training with new data for the new task.

- Approach: Use the output of one or more layers of a network trained for a different task as **generic feature extractors**. Then, train a new shallow classifier on these features.

Off-the-shelf pre-trained models



Off-the-shelf pre-trained models

Now a question might arise, how well do these pre-trained off-the-shelf features really work in practice with different tasks?

- [1] reported that CNN off-the-shelf features works well for fine-grained classification, but fine-tuning (the second approach) could lead to improvement through better adaptation.
- Off-the-shelf pre-trained models are normally used when the source and target datasets are from the **same domain** and the target dataset is **small.**

[1] Bousetouane, F. and Morris, B., 2015, December. Off-the-shelf CNN features for fine-grained classification of vessels in a maritime environment. In *International Symposium on Visual Computing* (pp. 379-388). Springer, Cham.

Fine Tuning Off-the-shelf Pre-trained Models

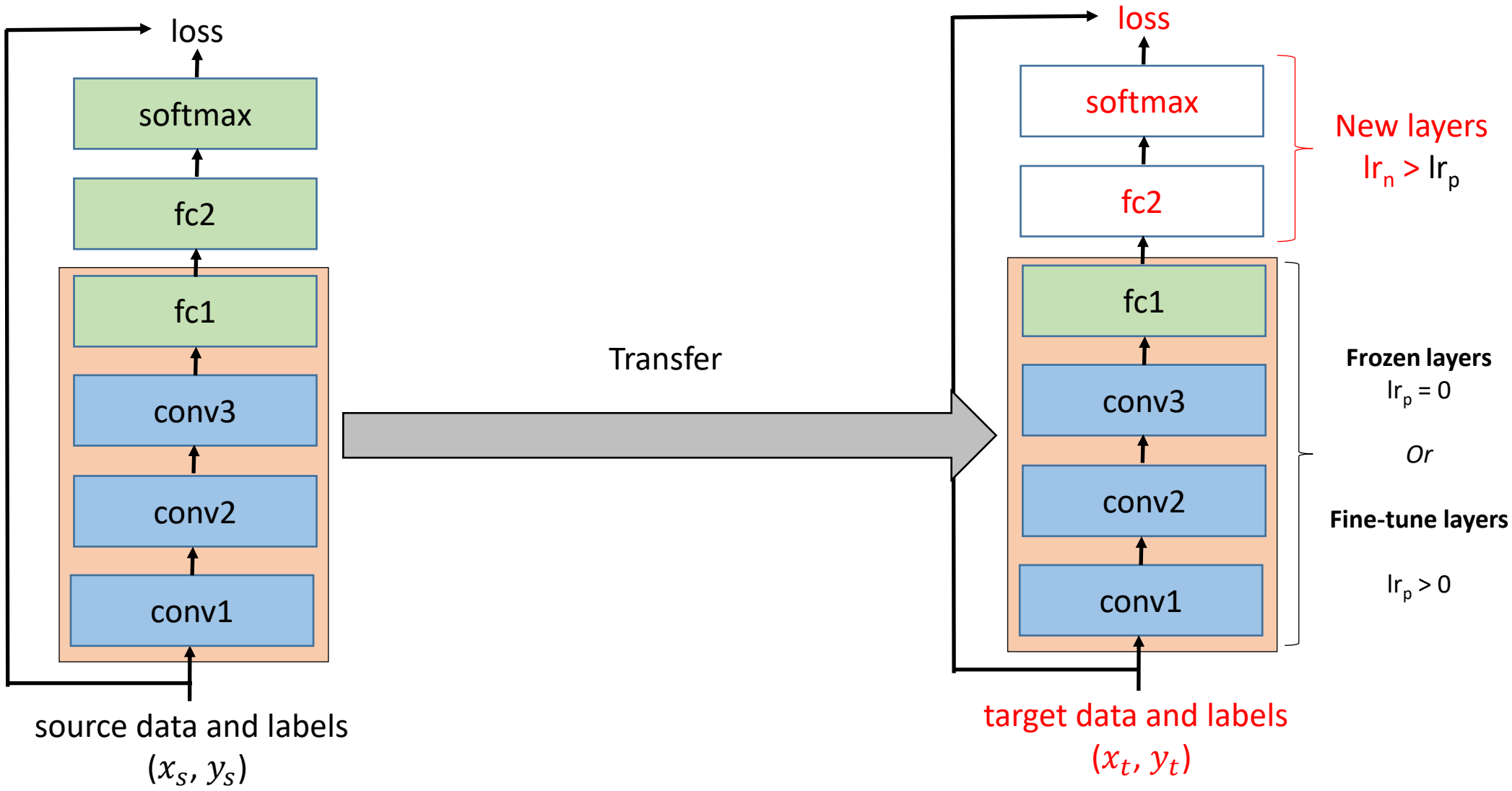
SWINBURNE UNIVERSITY OF TECHNOLOGY SARAWAK CAMPUS

- This is a more complex technique, in which we not only **replace** the last layer (for classification/regression), but we also selectively **retrain** some of the previous layers.

- We use the knowledge in terms of the global architecture of the network and use its states as a starting point for our retraining step.

- Approach: freeze (fix weights) some layers during retraining, or **fine-tune** others according to our needs.

Fine Tuning Off-the-shelf Pre-trained Models



Fine Tuning Off-the-shelf Pre-trained Models

SWINBURNE UNIVERSITY OF TECHNOLOGY SARAWAK CAMPUS

Now the first question arises:

Should we freeze layers in the network to use them as feature extractors or should we also fine-tune layers in the process?

- It depends on the target task. You can:
  - Freeze when target data is scarce. It is to avoid overfitting.
  - Fine-tune when there are more target data.

❖Freeze means weights are not updated during the training (back-propagation)

Pre-trained models and source data

SWINBURNE UNIVERSITY OF TECHNOLOGY SARAWAK CAMPUS

- For computer vision, you can leverage some popular models that we have learn in Lecture 5:
  - VGG (eg. VGG-16)
  - GoogleNet (eg. Inception V3)
  - ResNet

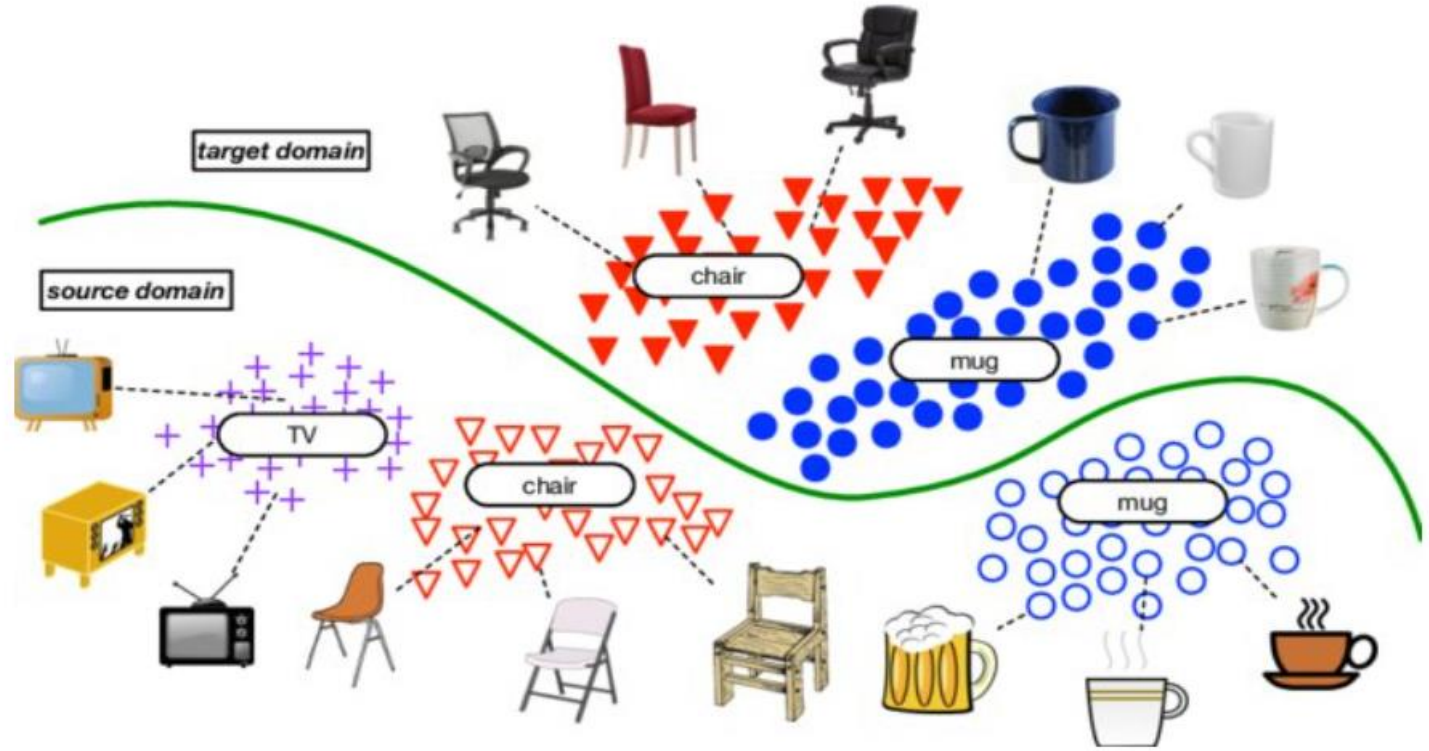
- Source data is depend on the application:
  - Image classification: ImageNet, Places Dataset
  - Video classification: Sport-1M, Kinetics, Activity Net

Types of Deep Transfer Learning

- Transfer learning is a general concept or principle of solving a target task using knowledge from the source task domain.
- Examples of Deep Transfer Learning
  - **Doman adaptation**
  - Multi-task Learning
  - Zero-shot learning
  - One-shot learning

Domain adaptation

- **Challenge**: The distribution of data in the target domain is different than in the source domain. We call this **domain shift**.
- **Questions**: How to overcome the differences between the domains so that a classifier trained on the source domain generalizes well to the target domain.



Domain adaptation

- Divergence based Domain Adaptation
- Adversarial based Domain Adaptation
- Reconstruction based Domain Adaptation

Divergence based Domain Adaptation

- Divergence based domain adaptation works on the principle of **minimizing some divergence** based criterion between **source** and **target** distribution, hence leading to **domain invariant-features.**
- Domain-invariant features refers to features that do not change significantly whether they are observed in the source domain or the target domain.
- The goal is to learn such features that are effective and relevant for the task in both the training (source) and the application (target) environments.
- By identifying and using domain-invariant features, a machine learning model can perform well on data from the target domain even though it was trained on data from the source domain.

Divergence based Domain Adaptation - Example

Imagine you have a dataset of car images taken during the day and you want to classify car types in images taken at night. The lighting conditions are different, but the shape of the cars, their logos, and design features like grills and window shapes do not change between day and night.

When training a model for car classification, domain-invariant features could include **Car Shape, Logos, Wheels,** …

By focusing on these domain-invariant features, a classification model trained on daytime images can still recognize and classify cars in nighttime images effectively. The domain shift due to lighting conditions is mitigated by the model's focus on features that are constant across both domains.

Divergence based Domain Adaptation

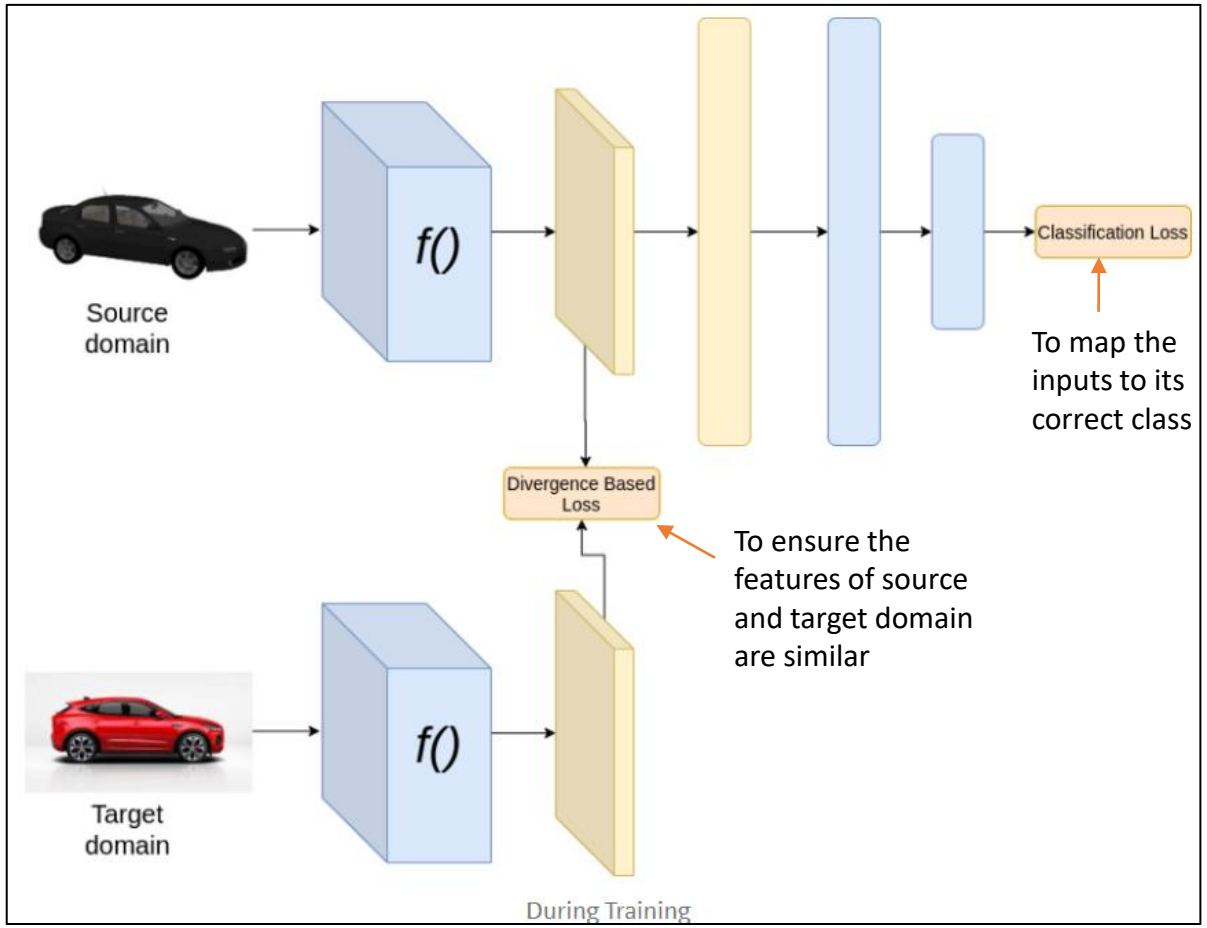
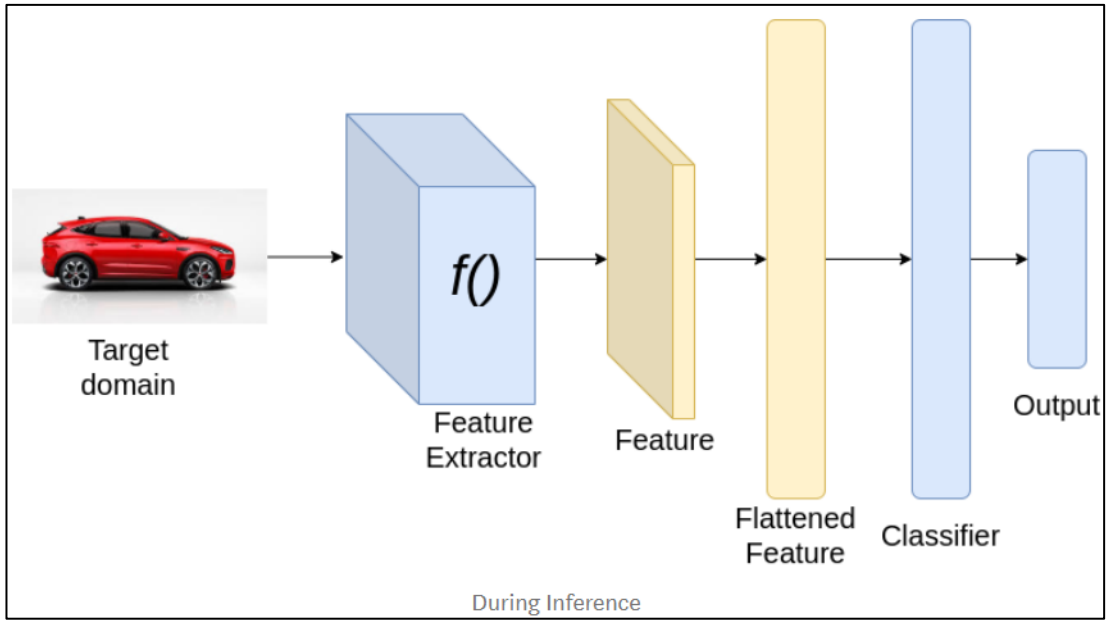


Image From: https://levelup.gitconnected.com/understanding-domain-adaptation-63b3bb89436f



- Divergences are typically non-parametric and consist of handcrafted mathematical formulas that are not tailored to any specific dataset or task, such as classification, object detection, or segmentation.

Adversarial based Domain Adaptation

- **Adversarial learning methods** are a promising approach to training robust deep networks, and can generate complex samples across diverse domains.

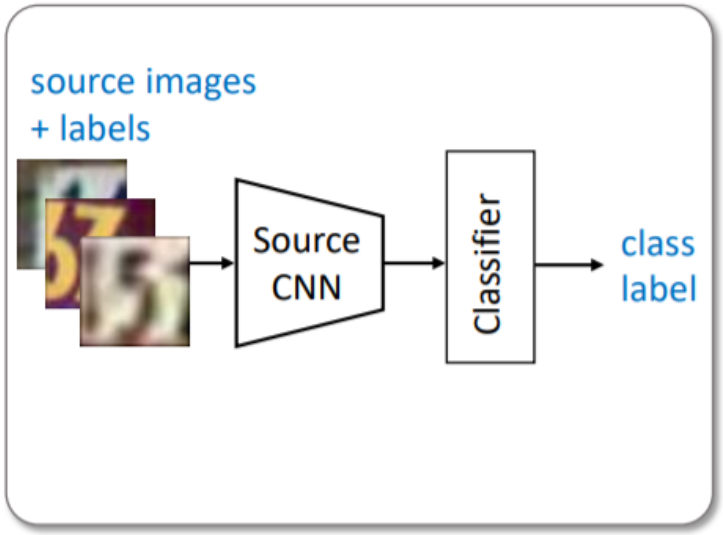
- It is a technique used in machine learning to **fool or misguide** a model with malicious input

- For adversarial based domain adaptation, the model is intended to learn a discriminative mapping of target images to the source feature space (target encoder) by fooling a domain discriminator that tries to distinguish the encoded target images from source examples.

Adversarial based Domain Adaptation

Step 1

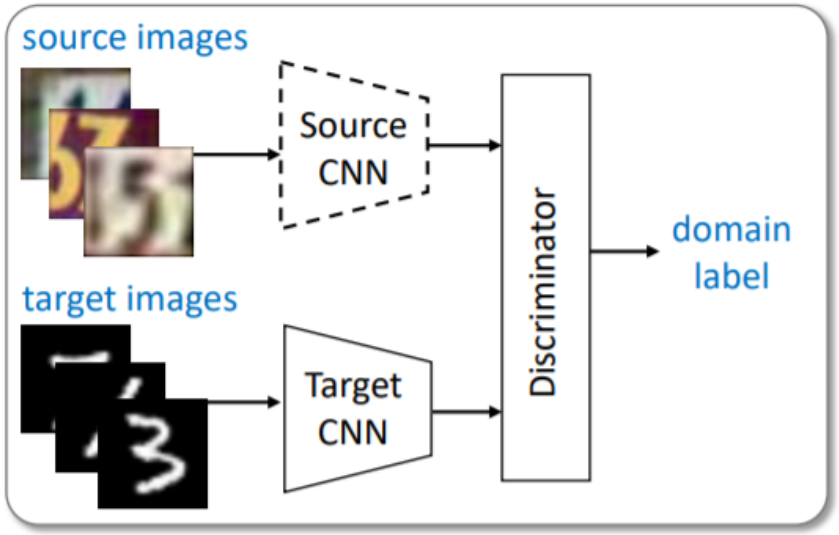
Pre-training



First pre-train a source encoder CNN using labelled source image examples

Step 2

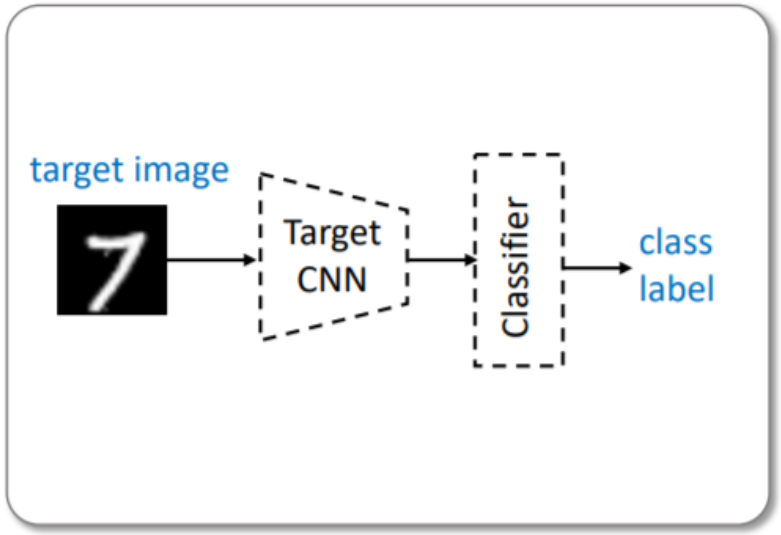
Adversarial Adaptation



Discriminator helps to produce features that are indistinguishable for source and target domain

Step 3

Testing



Target images are mapped with the target encoder to the shared feature space and classified by the source classifier

Tzeng, E., Hoffman, J., Saenko, K. and Darrell, T., 2017. Adversarial discriminative domain adaptation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 7167-7176).

Adversarial Based Domain Adaptation – Brief Overview

SWINBURNE UNIVERSITY OF TECHNOLOGY SARAWAK CAMPUS

1.Training Phase:

1. The feature extractor and the discriminator are both active.
2. The feature extractor learns to generate features (same feature extractor for both source and target data).
3. The discriminator learns to distinguish between the features coming from the source and the target domains.
4. The adversarial training process aims to reach a point where the discriminator can no longer easily tell the difference between features from the two domains, indicating that the feature extractor is generating domain-invariant features.

2.Prediction Phase:

1. Only the feature extractor is used to process new data.
2. The learned domain-invariant features are fed into the rest of the model (e.g., a classifier) to make predictions.
3. The discriminator is discarded because its job of differentiating between domains is not relevant during the prediction stage.

Reconstruction based Domain Adaptation

- This works on the idea of Image-to-Image translation.
- The simplest model for Image-to-Image translation could be an encoder-decoder based network and use a discriminator to enforce encoder-decoder network to produce images that are similar to the source domain.

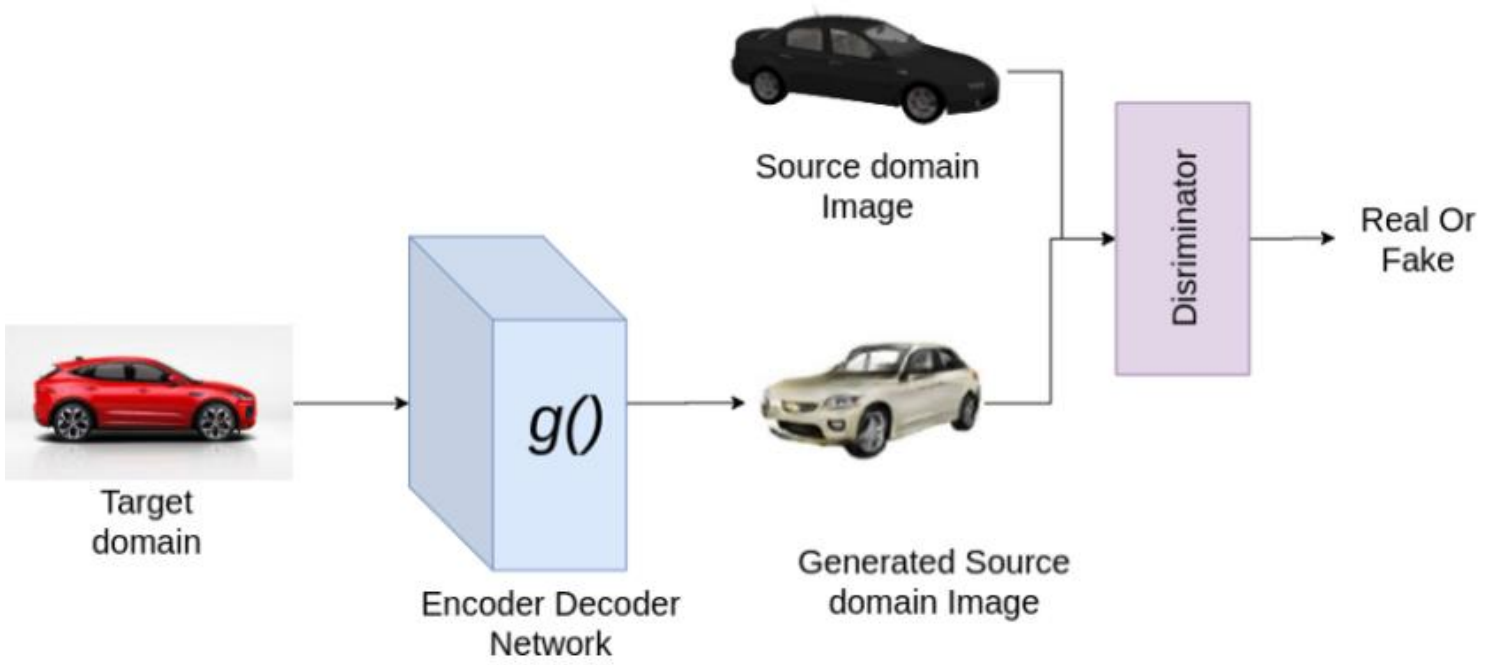


Image From: https://levelup.gitconnected.com/understanding-domain-adaptation-63b3bb89436f

Reconstruction based Domain Adaptation

- Then, train a source domain classifier based on source data

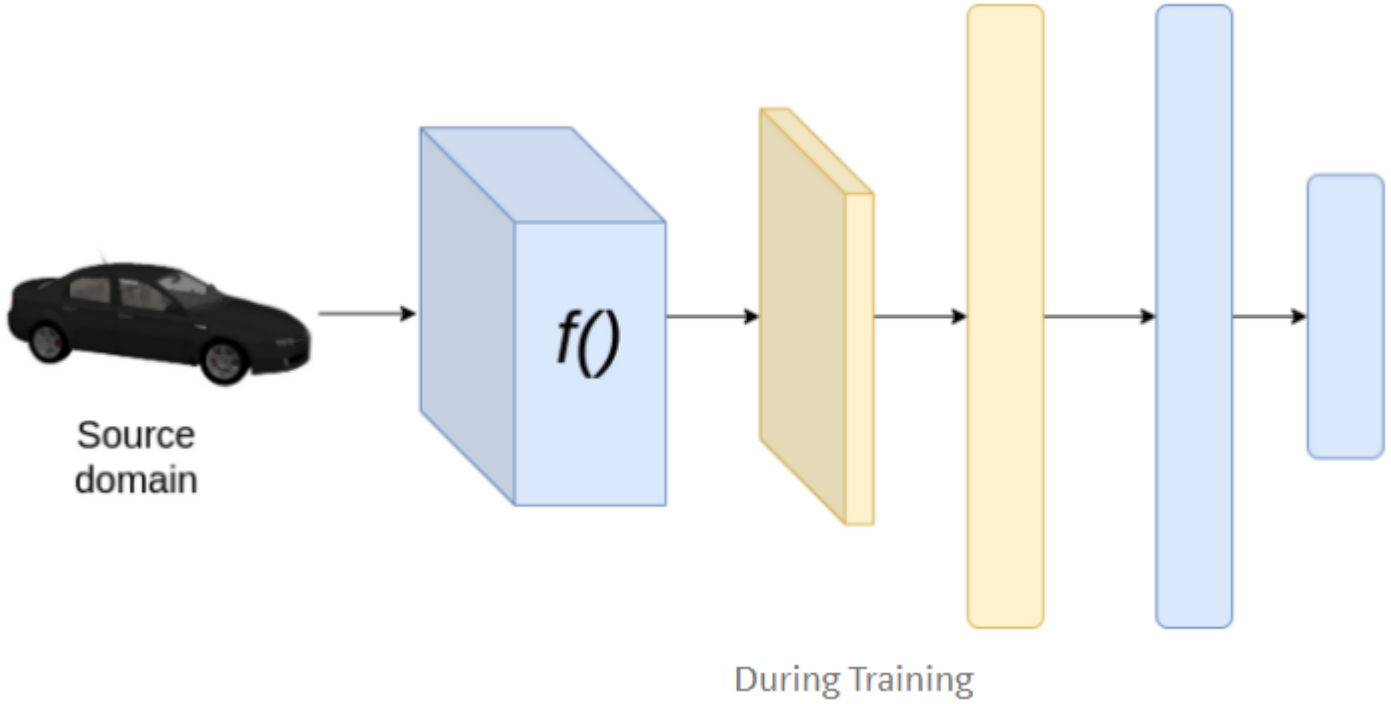
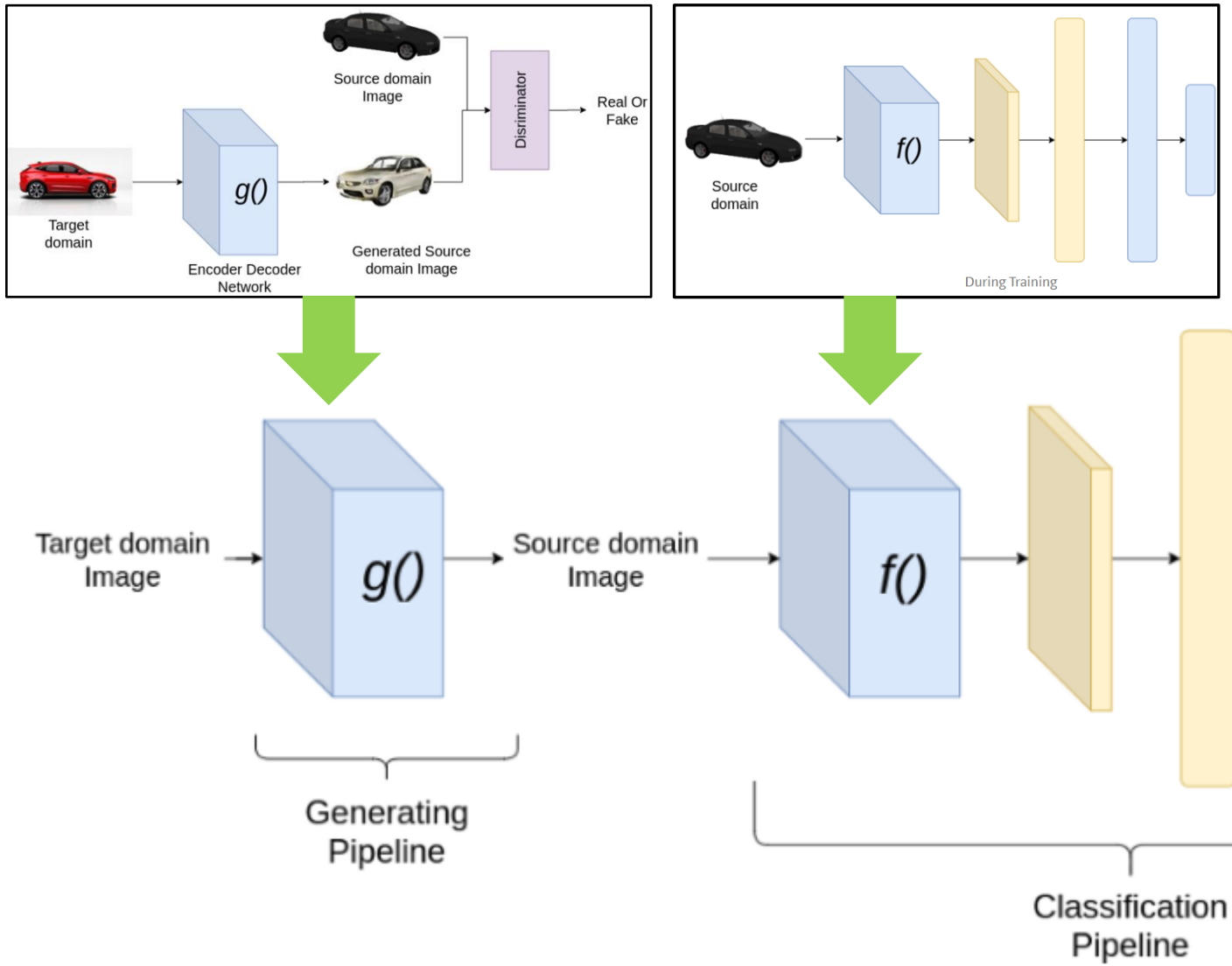


Image From: https://levelup.gitconnected.com/understanding-domain-adaptation-63b3bb89436f

Reconstruction based Domain Adaptation



- During inference, the image of the target domain will be transmitted to the source domain classifier trained for prediction.

Image From: https://levelup.gitconnected.com/understanding-domain-adaptation-63b3bb89436f

Reconstruction based Domain Adaptation – Brief Overview

SWINBURNE UNIVERSITY OF TECHNOLOGY SARAWAK CAMPUS

- **Encoder**: Compresses target image into a feature vector that captures essential details, ignoring domain-specific details.
- **Decoder**: Reconstructs the image to match the source domain's style, ensuring it fits the source domain.
- **Source Classifier**: Classifies the style-matched image using a model trained on source domain data.

Types of Deep Transfer Learning

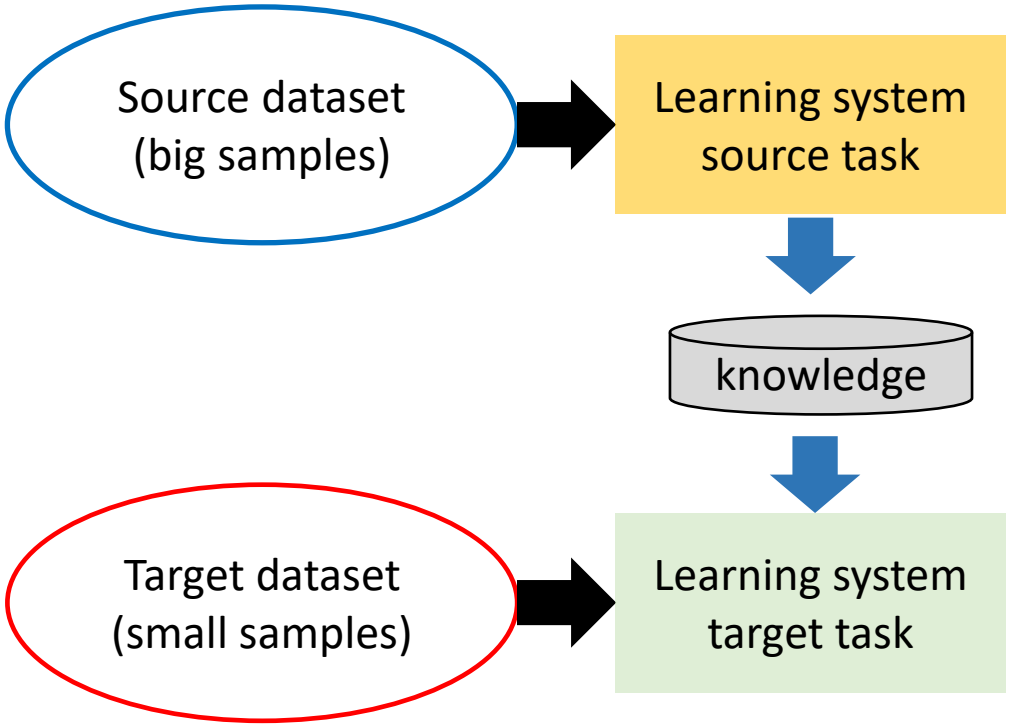
- Transfer learning is a general concept or principle of solving a target task using knowledge from the source task domain.
- Examples of Deep Transfer Learning
  - Doman adaptation
  - **Multi-task Learning**
  - Zero-shot learning
  - One-shot learning

Multi-task Learning (MTL)

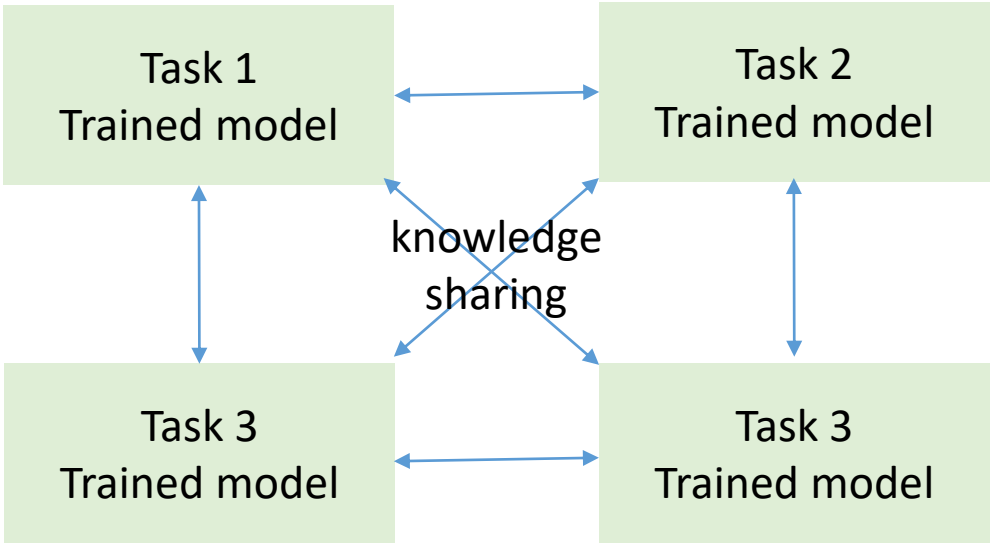
- Multitask learning is a slightly different flavor of the world of transfer learning.
- By giving a set of learning tasks, t1 , t2 , …, t(n), the learner **co-learn all tasks simultaneously.**
  - In other words, the learner optimizes the learning/performance across all of the n tasks **through some shared knowledge.**
- In this case, the learner receives information on multiple tasks at once without distinguishing between the source and targets.

Transfer Learning vs MTL

Transfer learning



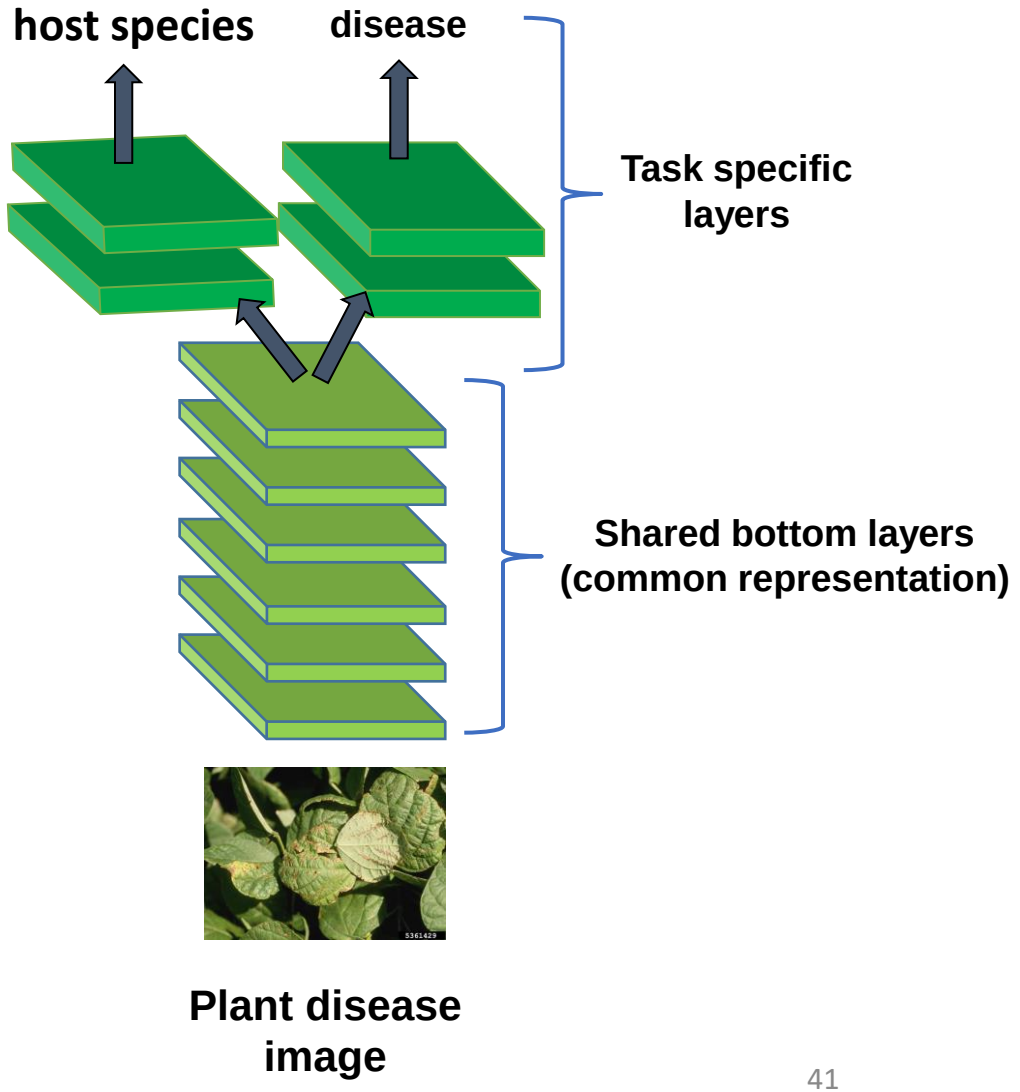
Multitask Learning



Generic Multi-task Learning

Case study: Plant disease identification

- The generic multitask learning consists of two components: the shared layers and the specific layers.

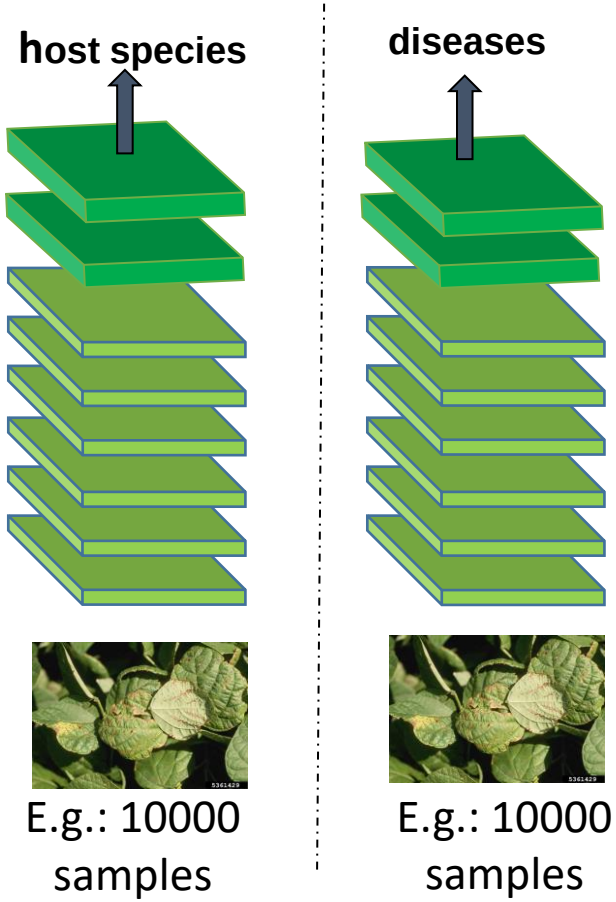


When MTL makes sense?

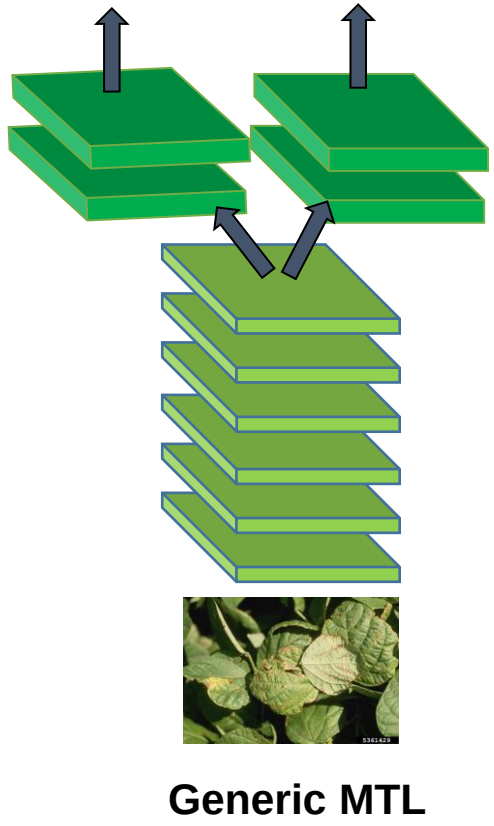
- Training on a set of tasks that could benefit from sharing lower-level features
  - For example: The features of the host species can help identify plant diseases. These features are the ones that pathologists use to infer associated diseases and to establish lists of diseases associated with host species.

When MTL makes sense?

- Usually: Amount of data you have for each task is quite similar



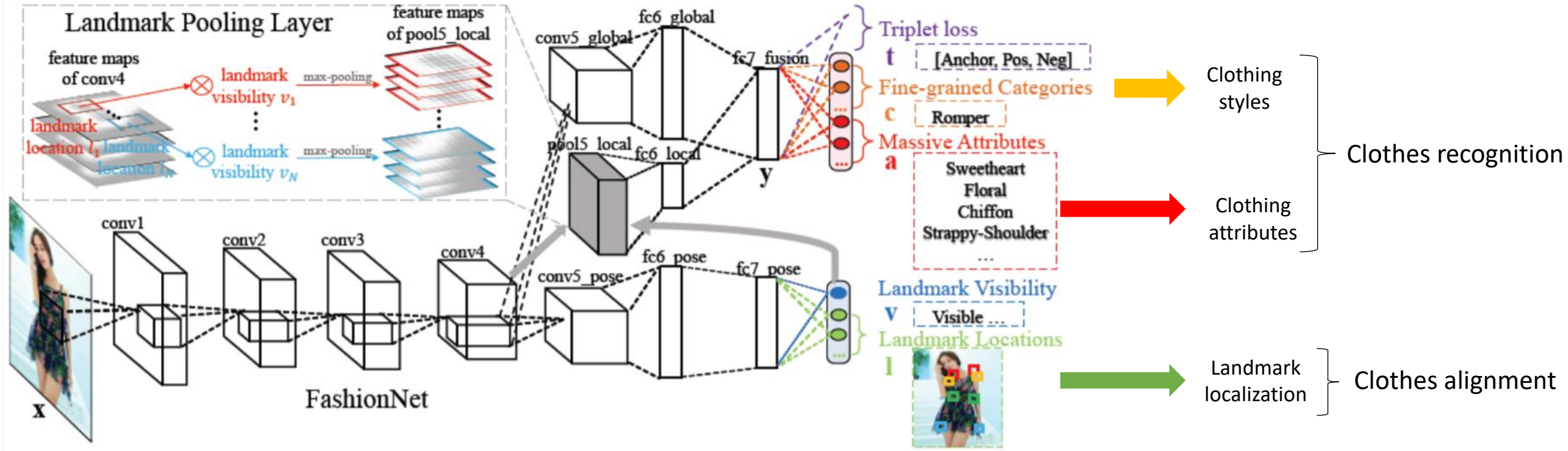
10000 + 10000 = 20000 samples



Having more **relevant** data to train a deep network could help increase performance.

When MTL makes sense?

- Can train a **large** network of neurons to do all tasks well
  - The size of the neural network must be taken into consideration if one wants to train it to adapt to many tasks.



Liu, Ziwei, et al. "Deepfashion: Powering robust clothes recognition and retrieval with rich annotations." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016.

Types of Deep Transfer Learning

- Transfer learning is a general concept or principle of solving a target task using knowledge from the source task domain.
- Examples of Deep Transfer Learning
  - Doman adaptation
  - Multi-task Learning
  - **Zero-shot learning**
  - One-shot learning

Zero-shot learning

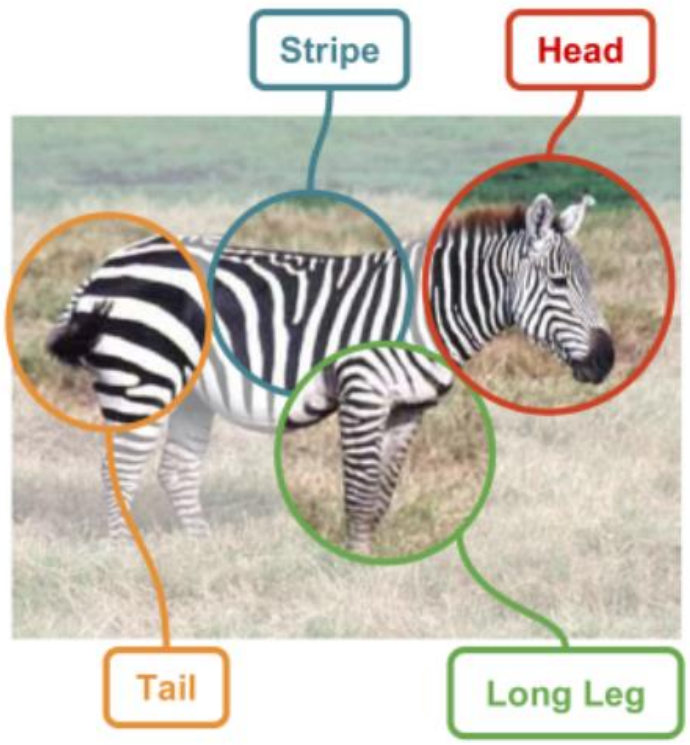
- Zero-shot learning (ZSL) enables identification of classes that are **not seen** before by means of **transferring knowledge** from **seen classes** to **unseen classes**.
- This knowledge transfer is usually done via utilizing **prior information from various auxiliary sources**, such as descriptions/semantic attributes/word embedding's (vector representations of a particular word).
- For example, consider a scenario where an AI model is trained to classify various animals using a collection of images and their descriptive texts. If this model, which knows how to identify horses, is provided with descriptions stating that zebras are essentially striped horses, it can recognize a zebra image accurately, even though it has never explicitly been trained on images of zebras.

Demirel, B., Gokberk Cinbis, R. and Ikizler-Cinbis, N., 2017. Attributes2classname: A discriminative model for attribute-based unsupervised zero-shot learning. In *Proceedings of the IEEE International Conference on Computer Vision* (pp. 1232-1241).

Examples of auxiliary information

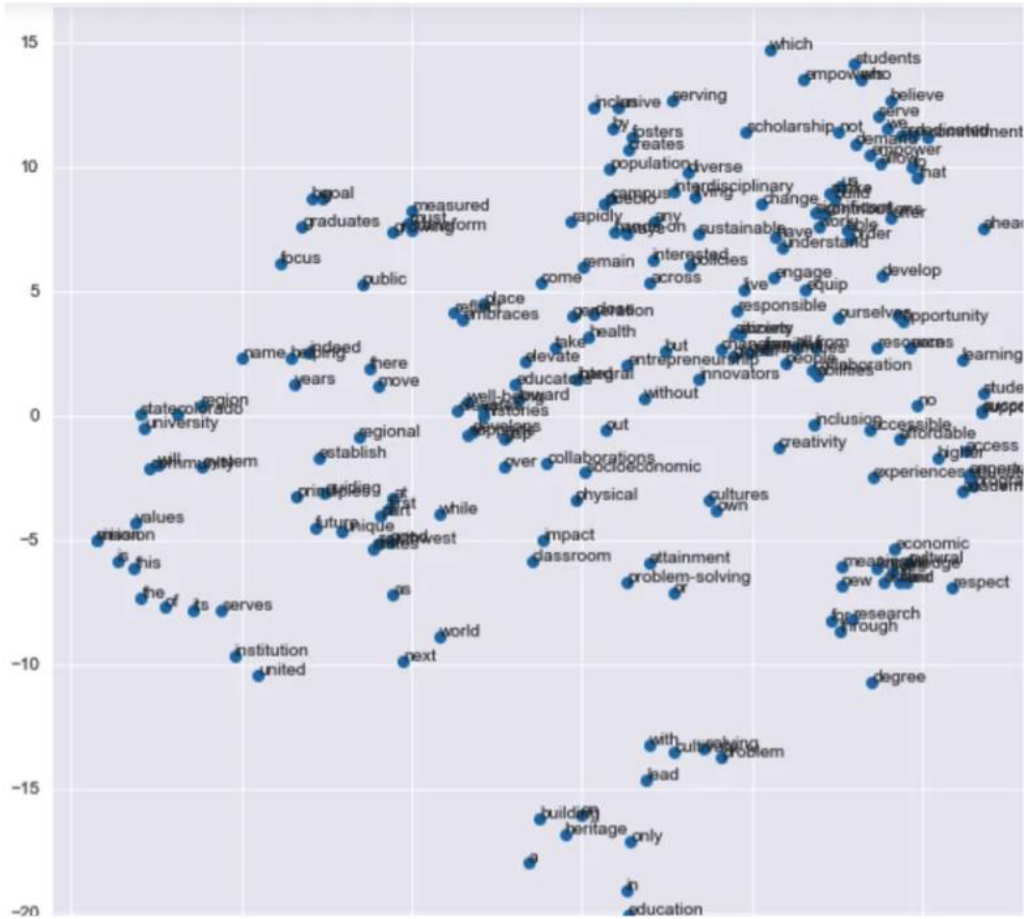
SWINBURNE UNIVERSITY OF TECHNOLOGY SARAWAK CAMPUS

Semantic attributes



Demirel, B., Gokberk Cinbis, R. and Ikizler-Cinbis, N., 2017. Attributes2classname: A discriminative model for attribute-based unsupervised zero-shot learning. In *Proceedings of the IEEE International Conference on Computer Vision* (pp. 1232-1241).

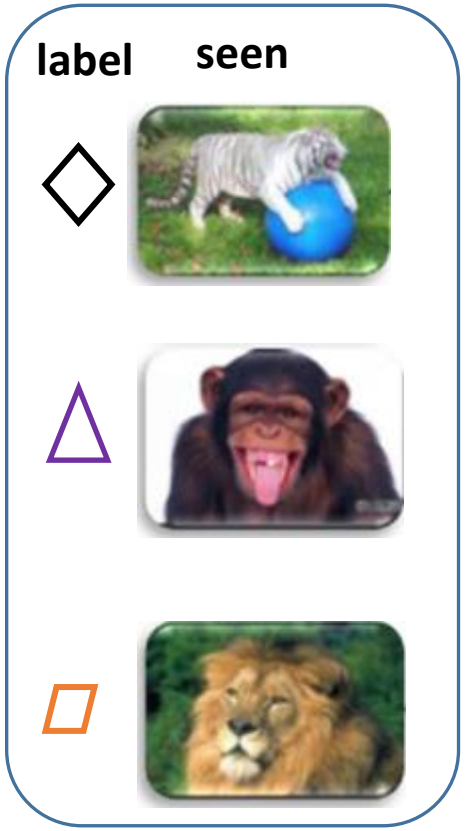
Word embedding

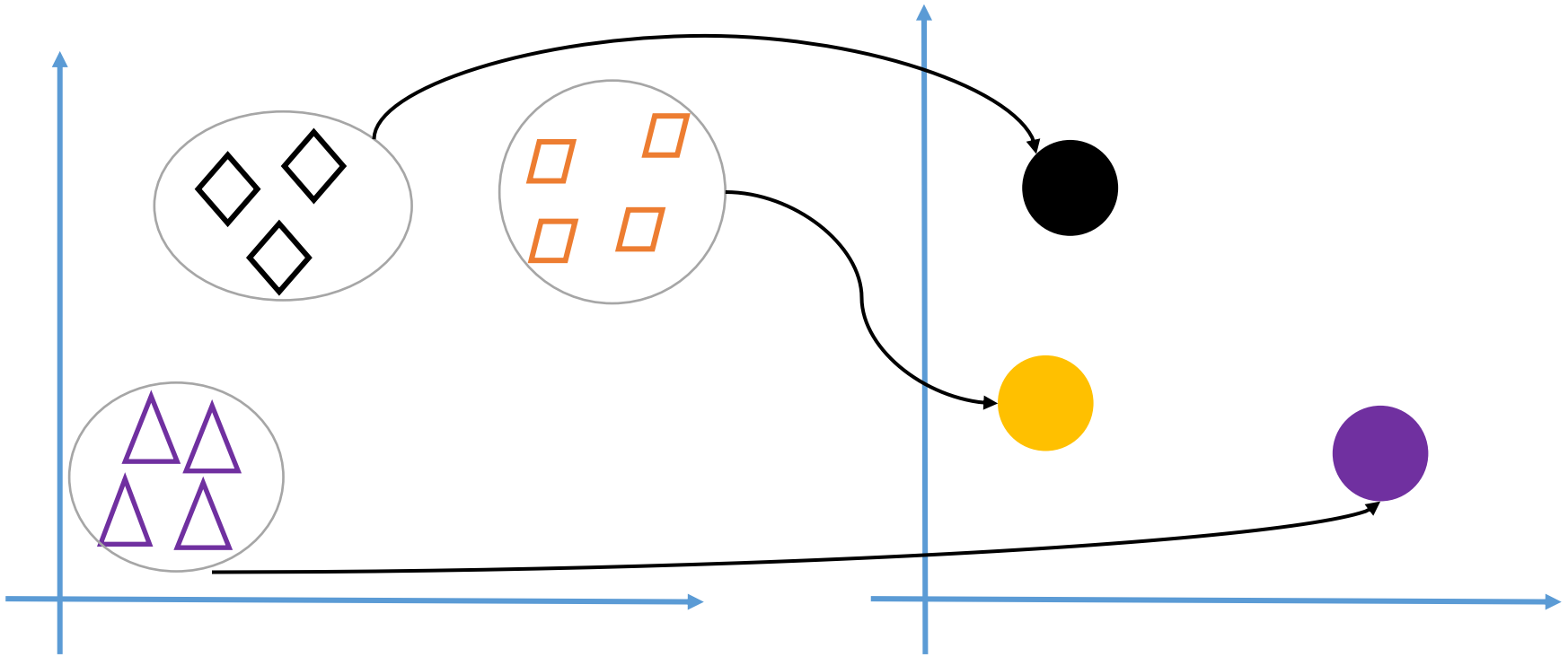


https://we1s.ucsb.edu/research_post/word-embeddings-of-college-and-university-mission-statements-preliminary-findings/

Idea behinds Zero-shot learning

- The aim is learn a projection function from visual space (i.e. image features) to semantic space (i.e. word vectors/semantic embedding) using data from seen classes.



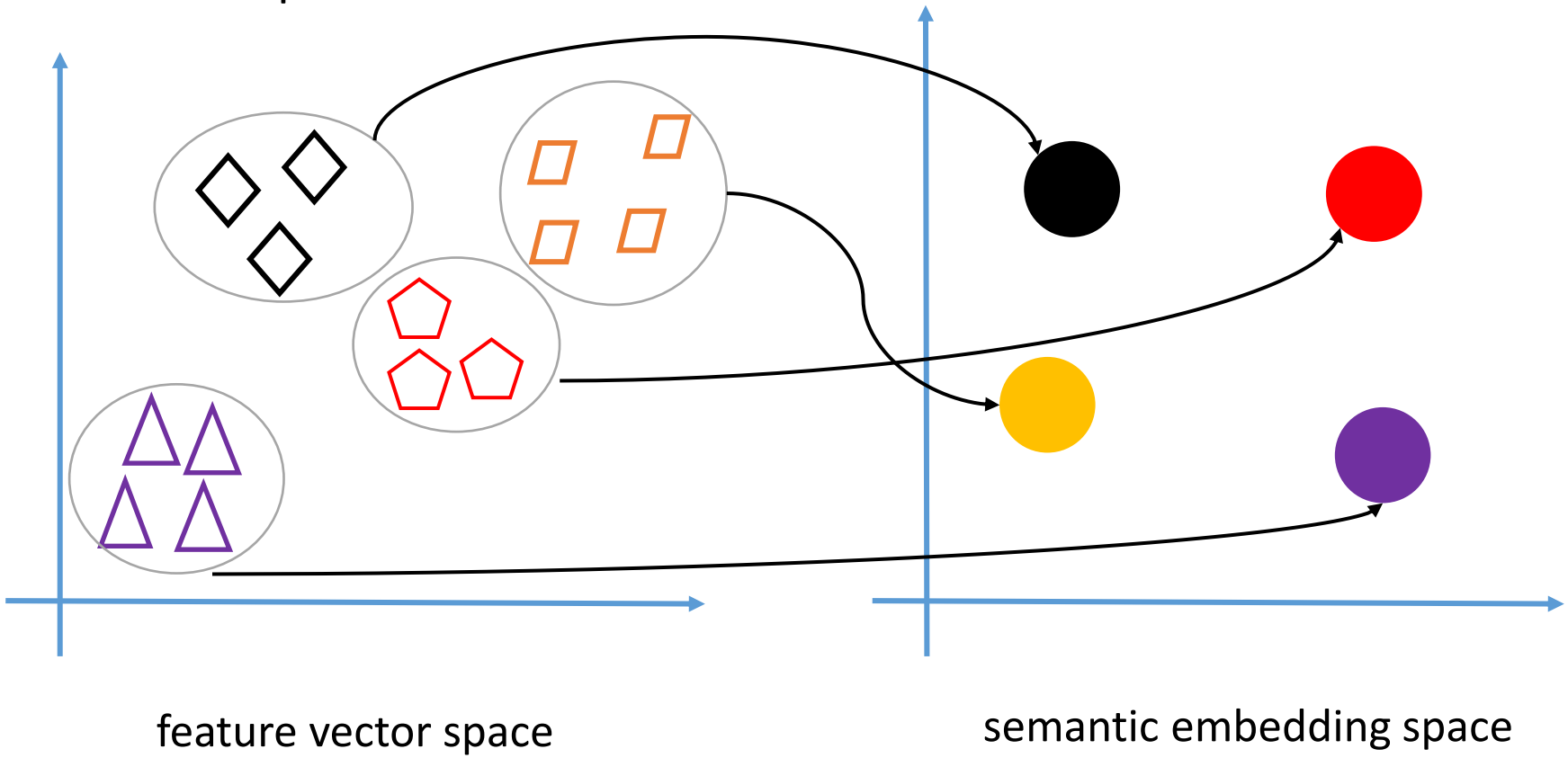


feature vector space

semantic embedding space

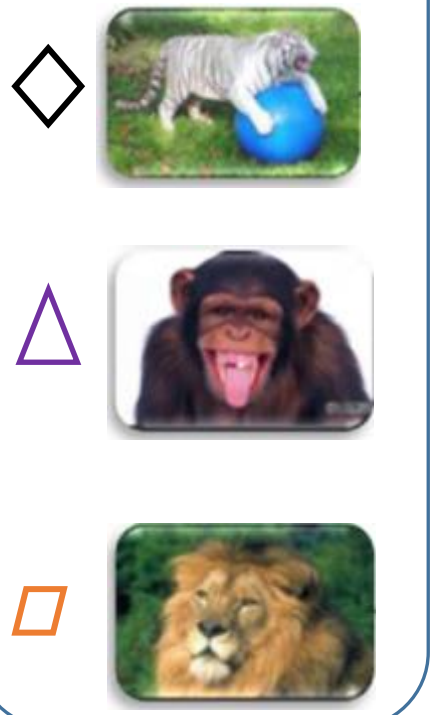
Idea behinds Zero-shot learning

- Then, the unseen class image features are passed as input to the trained network and we get the corresponding semantic embedding as the output.



Idea behinds Zero-shot learning

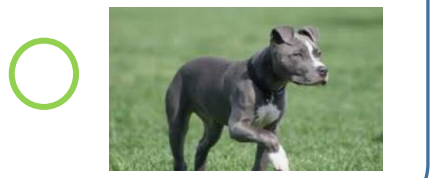
label seen

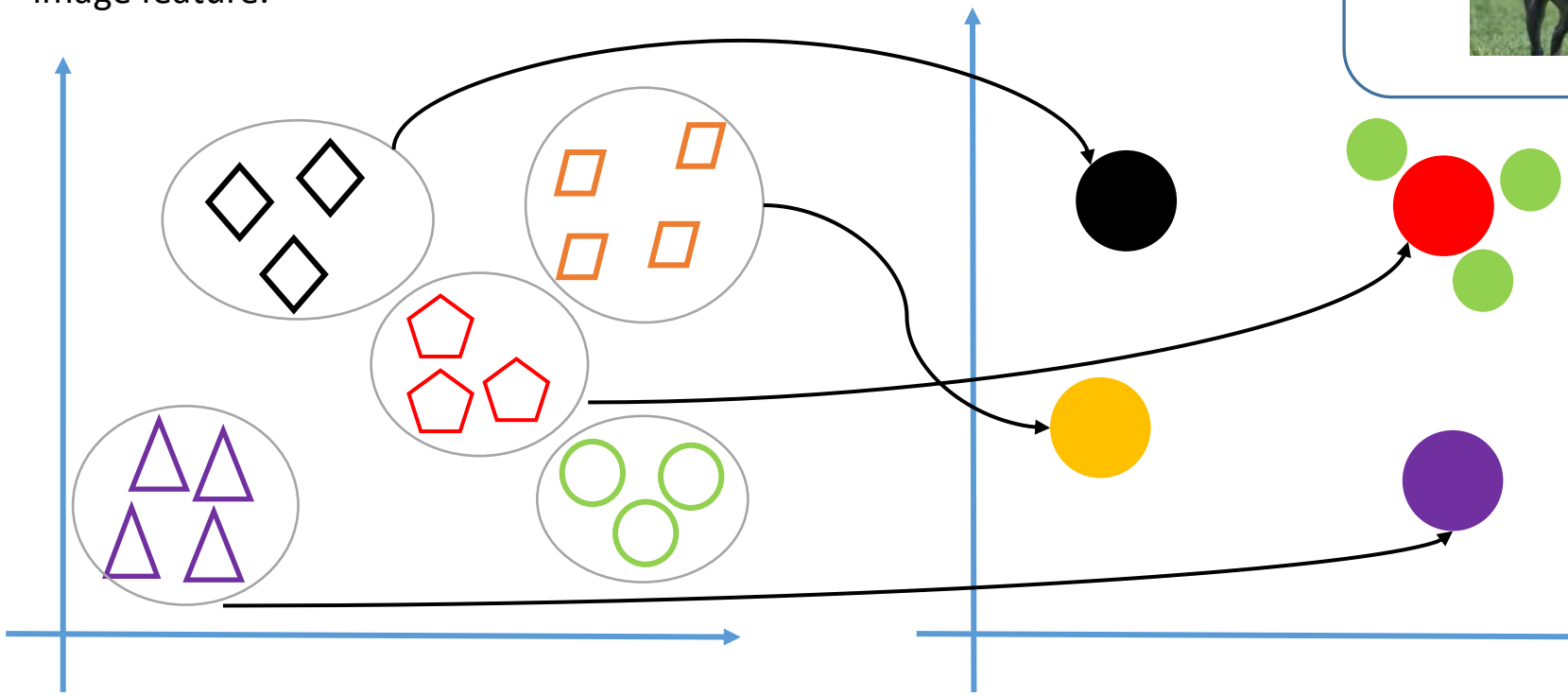


label unseen

- Nearest neighbor search is perform in the semantic embedding space to find the closest match to the output of the network. Finally, the label corresponding to the closest semantic embedding is predicted as the output label of the input image feature.

label test





feature vector space

semantic embedding space

Types of Deep Transfer Learning

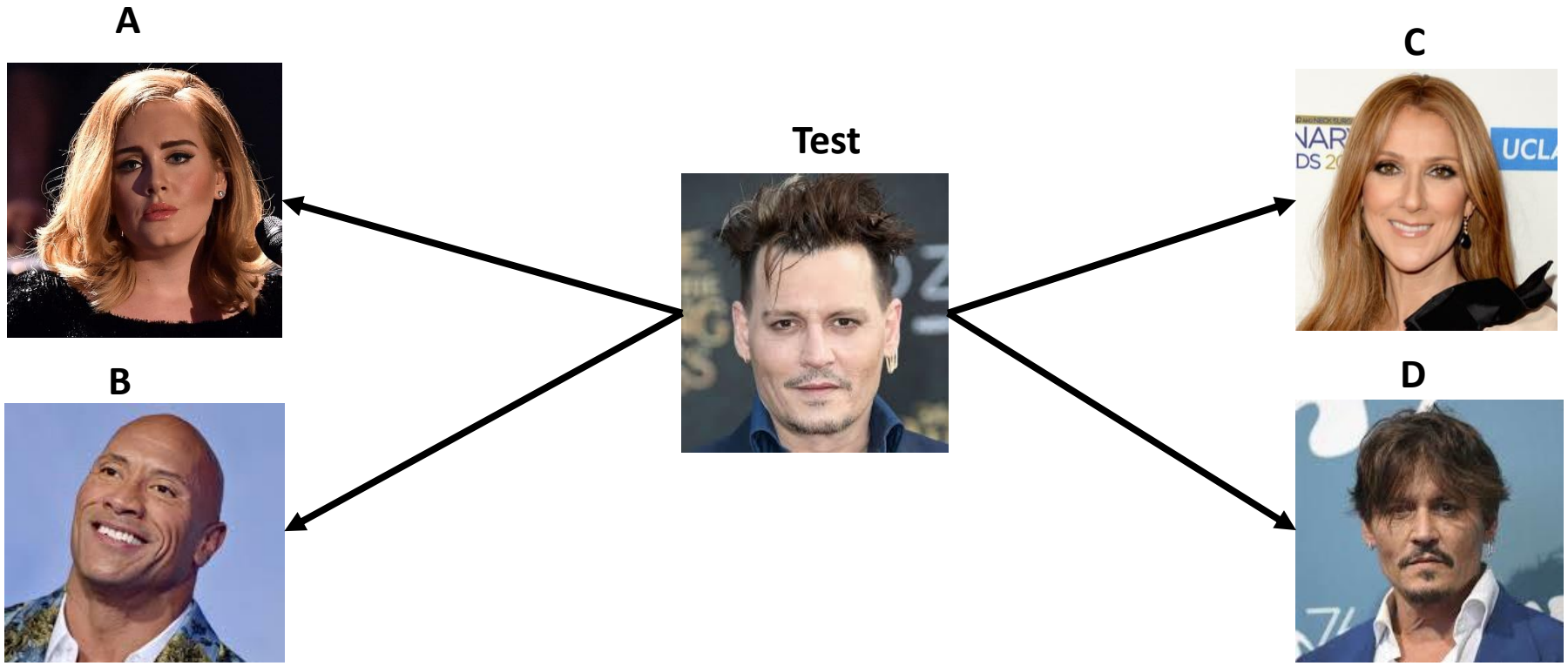
- Transfer learning is a general concept or principle of solving a target task using knowledge from the source task domain.
- Examples of Deep Transfer Learning
  - Doman adaptation
  - Multi-task Learning
  - Zero-shot learning
  - **One-shot learning**

One-shot learning

- **One-shot learning** is a variant of **transfer learning** where we try to infer the required output based on just one or a few training examples
- It emphasize on **knowledge transfer**, which makes use of prior knowledge of learnt categories and allows for learning on minimal training examples.

One-shot learning

- Case study 1: face recognition



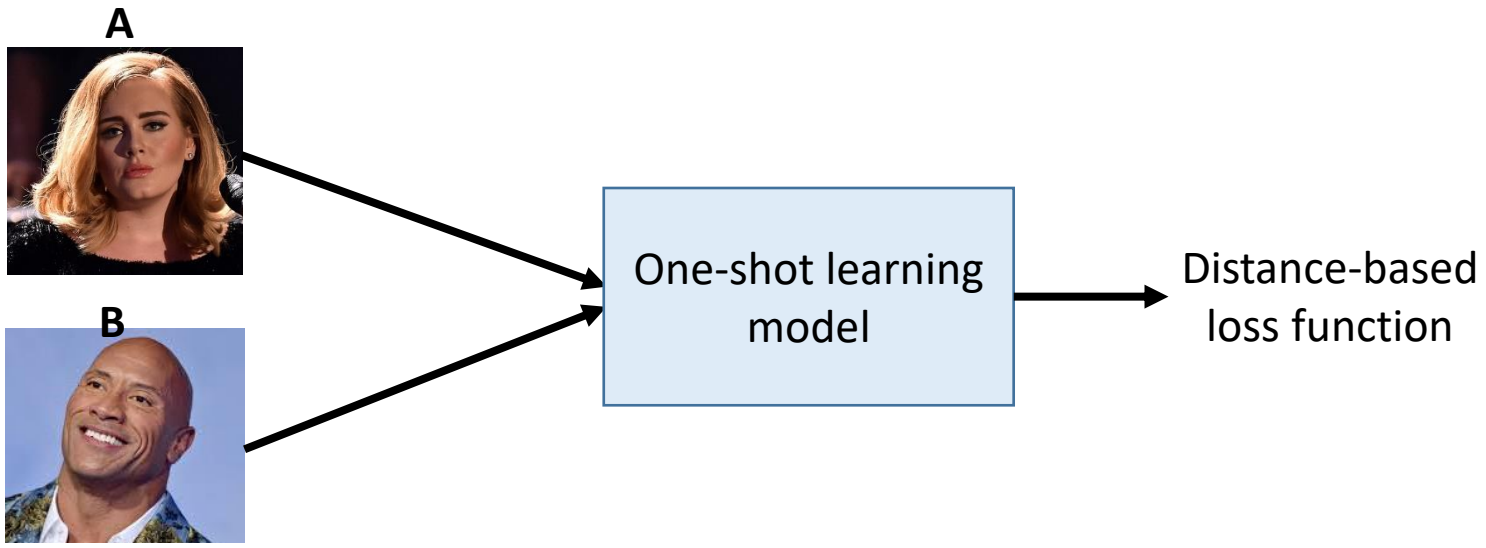
Problem of naïve convolutional network

- A small training set is not enough to train a robust neural network.
  - The trained feature vectors do not contain important information that can be used for future image recognition.

- Retraining the convolutional network every time the number of classes or dataset is increased is way too time-consuming and resource-intensive.

Idea behind one-shot learning

- Train a model to differentiate between the same and different pairs, then generalize these ideas to evaluate new categories.
- Learning a similarity function
  - The a deep learning model takes two images and returns a value that shows the similarity between the two images.



Idea behind one-shot learning

- If the images contain the same object (or the same face), the neural network returns a value below a specific threshold, λ (say zero) and if they are not the same object, it will be above the threshold.