Transfer Learning [TL] for Deep CNN

- Modern architectures are really deep and need a lot of data to train from scratch
- Data and labeling is expensive and time consuming
- e.g. Healthcare, Retail, Surveillance etc

Why Transfer Learning?

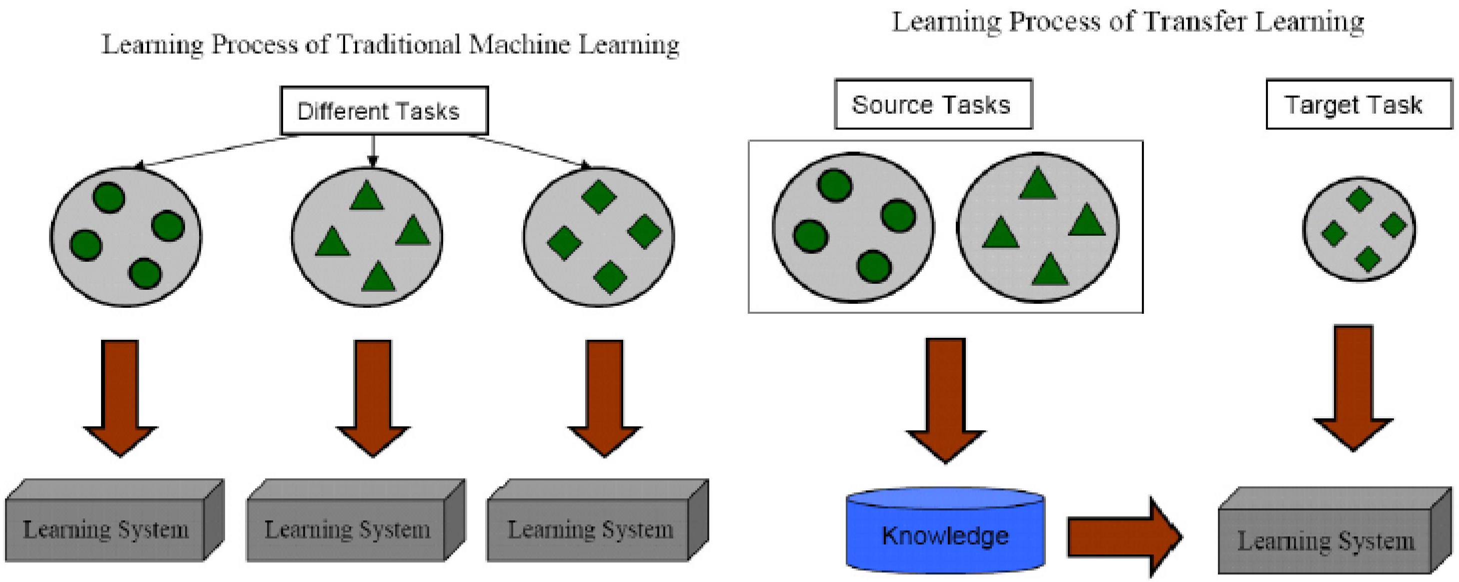
Given a new application, one looks at opportunities for re-using knowledge (e.g. architectures and weights) from similar learning problems which were trained with large amounts of data

Transfer Learning!

Humans are great at transfer learning

(e.g. Bicycle → Bike, Tennis → Badminton, Language skills)

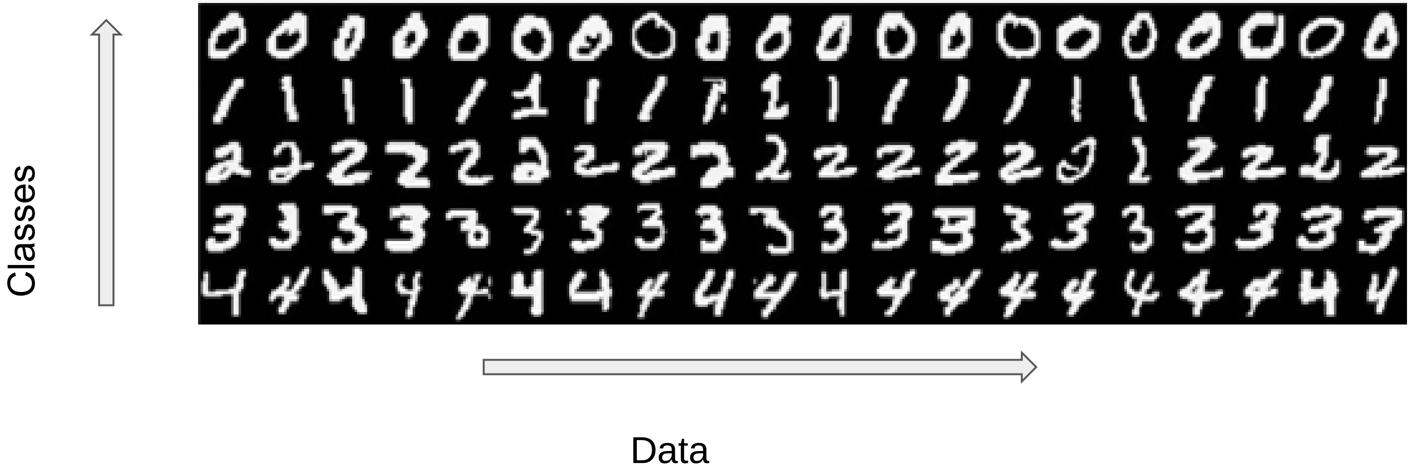
Transfer Learning - How it works?



(a) Traditional Machine Learning (b) Transfer Learning

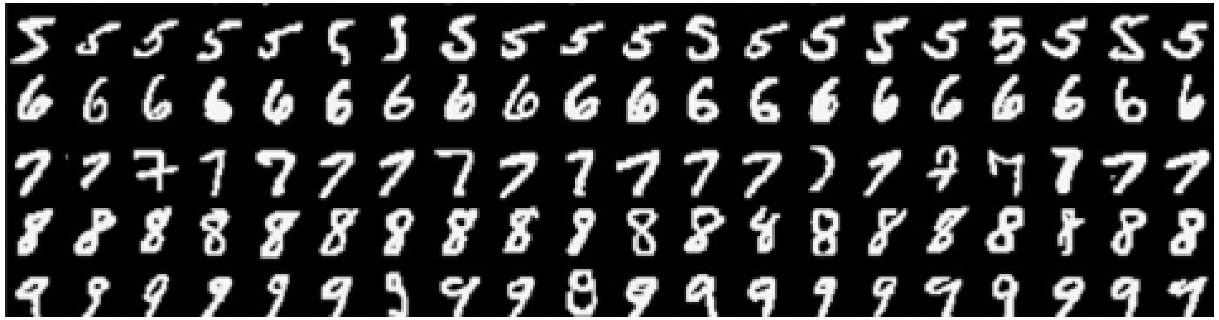
Transfer learning scenario

Assume that we have a ‘source domain’ classifier we have built on large amounts of data (e.g. Digit 0-4 classifier)



Target 1: Similar domain, different task

With the knowledge of how to build a classifier model which can classify digits 0 to 4, can we build a model which can classify digits 5 to 9?



Target 2: Different domain, same tasks

Want to build a number plate digit

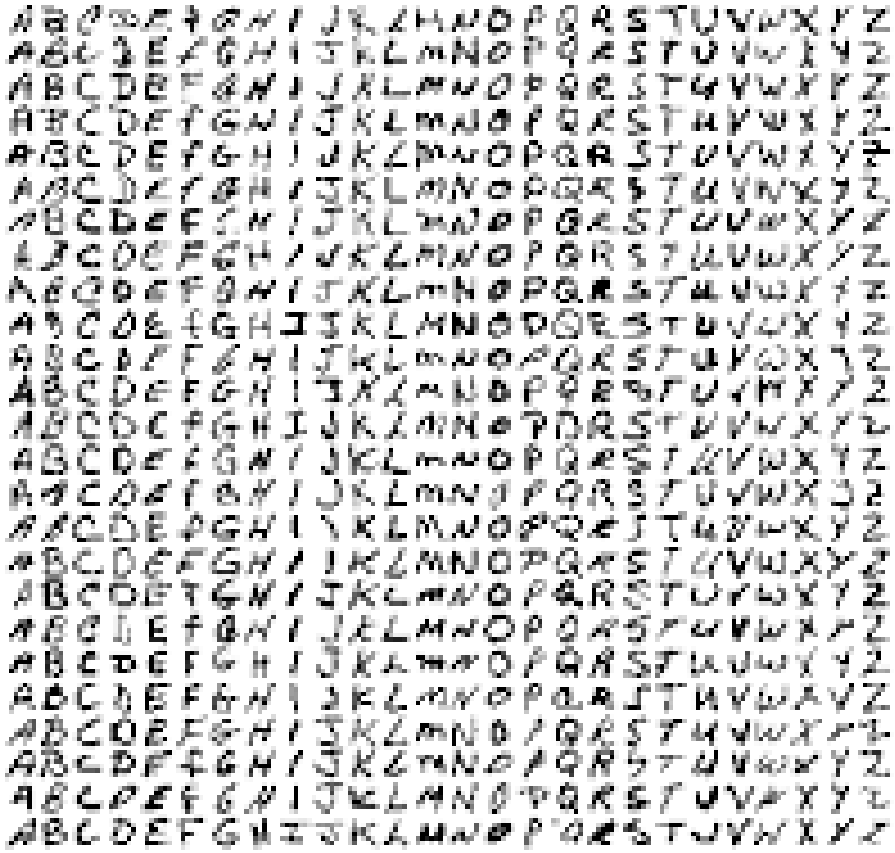
classifier

Domain Adaptation

With the knowledge of how to build a classifier model which can classify digits 0 to 4, can we build a model which can classify digits (0 to 4 only) in a number plate?

Target 3: Different Domain, Different tasks

Want to build an alphabet classifier

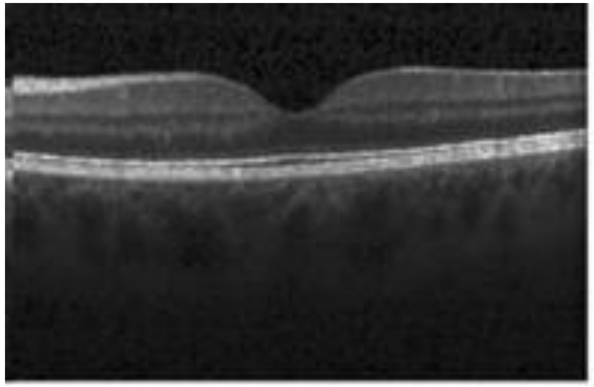


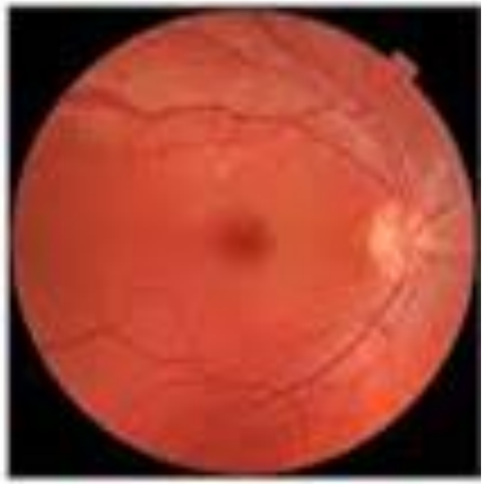
With the knowledge of how to build a classifier model which can classify digits 0 to 4, can we build a model which can classify alphabets?

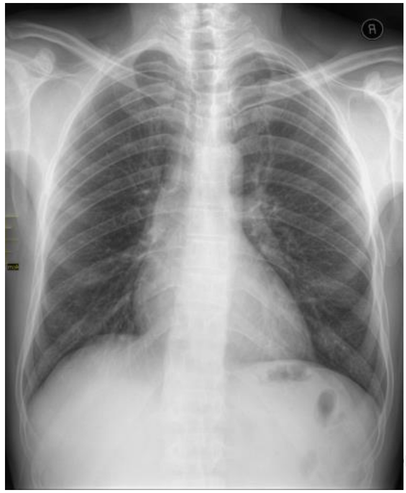
Applications of Transfer Learning

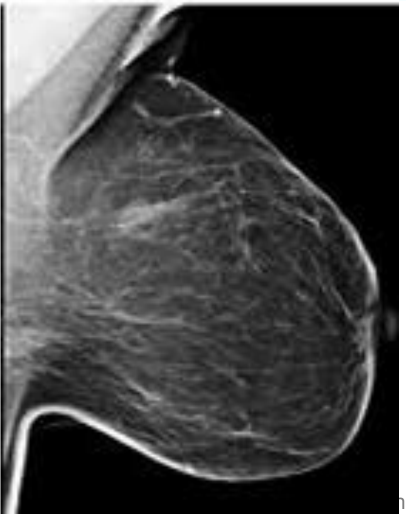
Transfer learning from ImageNet examples

Medical Data disease classif







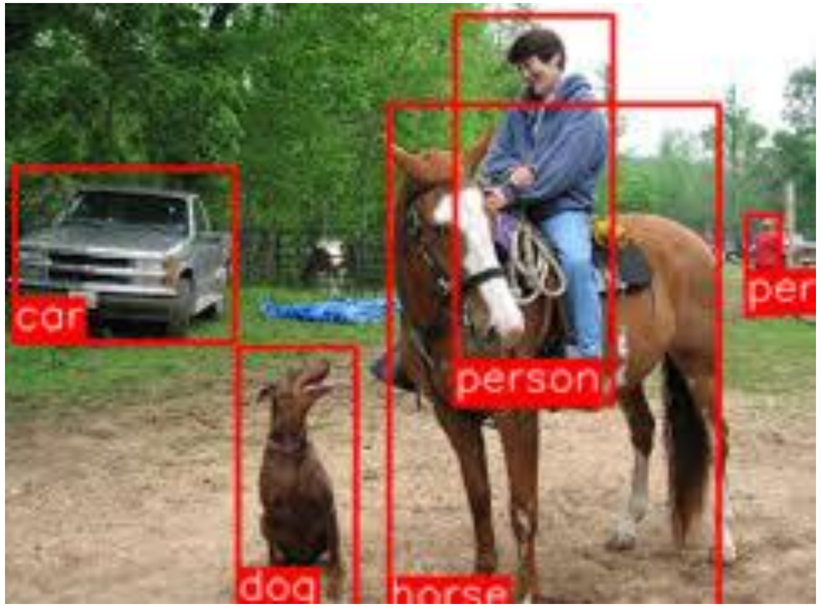


Data in the order of thousands, pixel resolution in the order of couple of millions

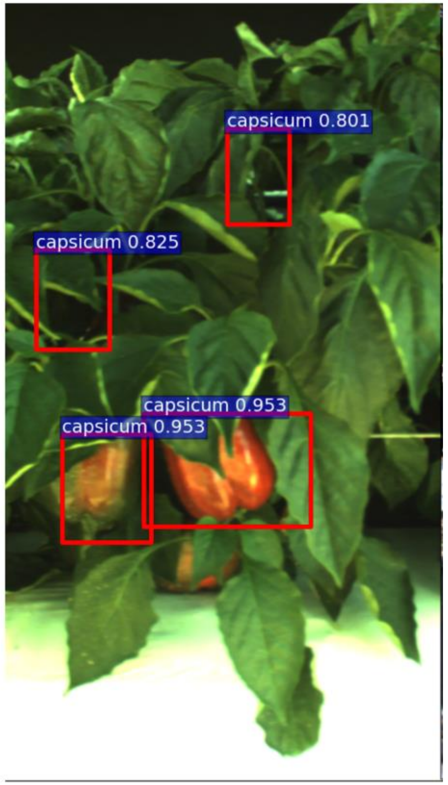
e.g. Adapt a VGG network trained on ImageNet to classify above data

Applications of Transfer Learning

Object Detection/ Recognition



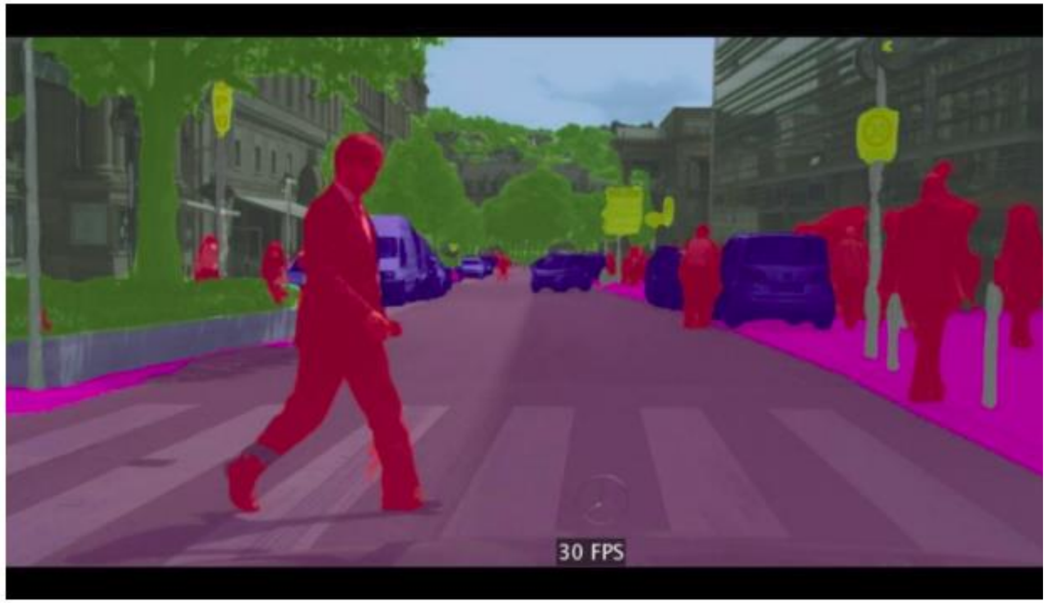
Less Data with bounding boxes and labels

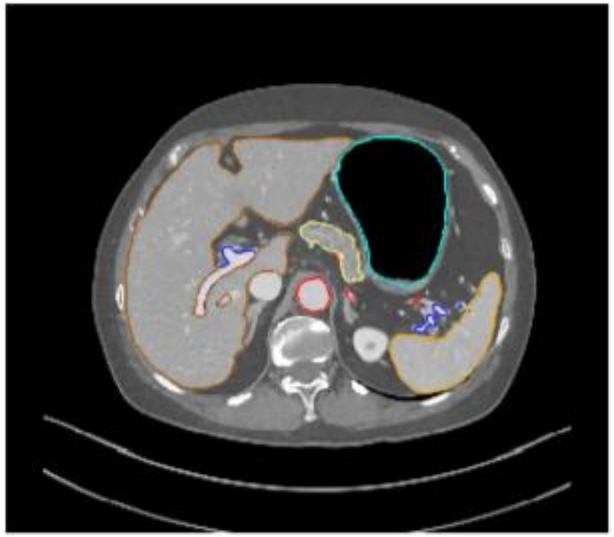


e.g. Adapt a VGG network trained on ImageNet to 'put' a Bounding box around objects and recognize/classify them

Applications of Transfer Learning

Segmentation - Difficult to get segmented training data



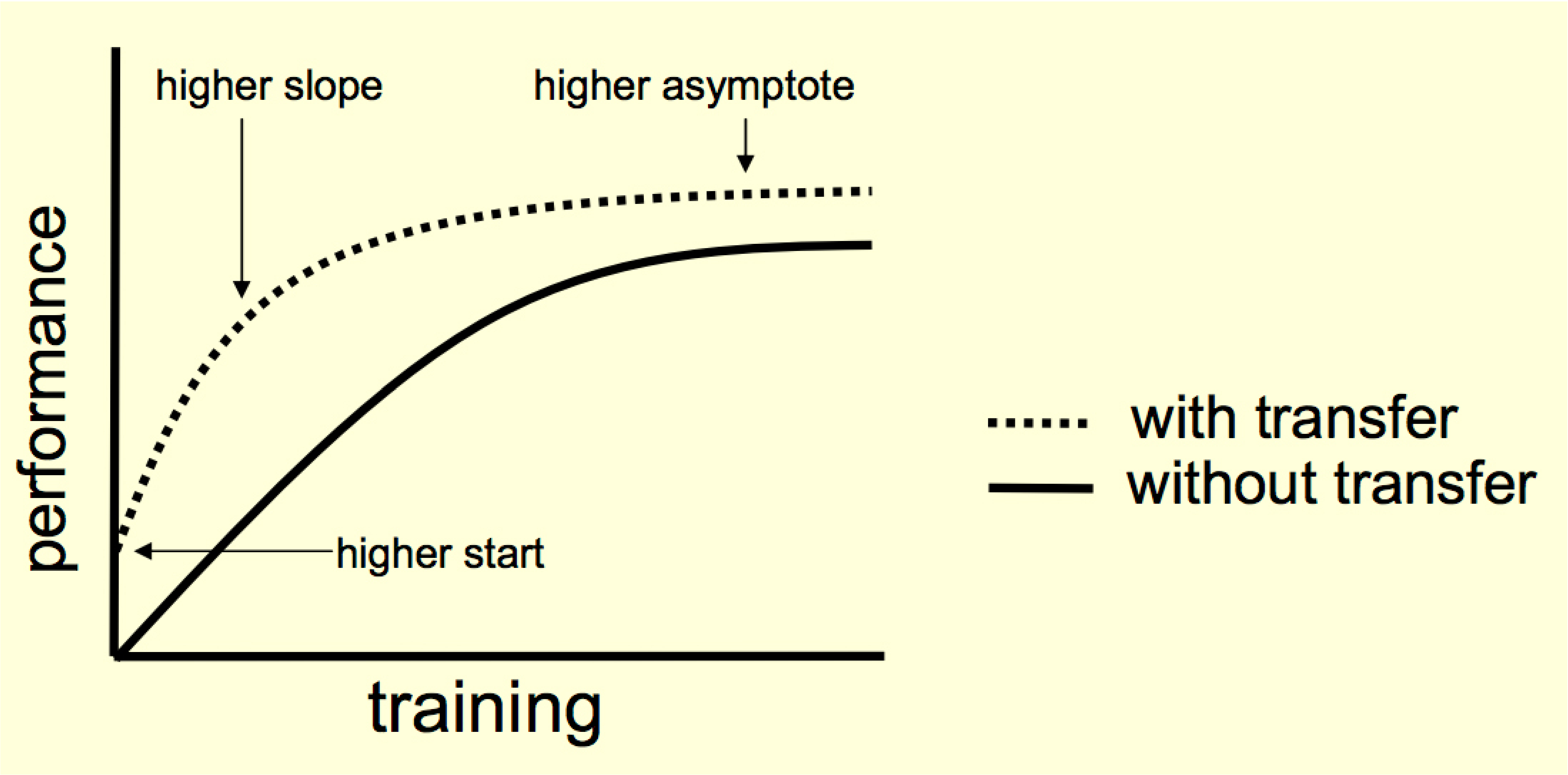


(c) Segmentation (axial)

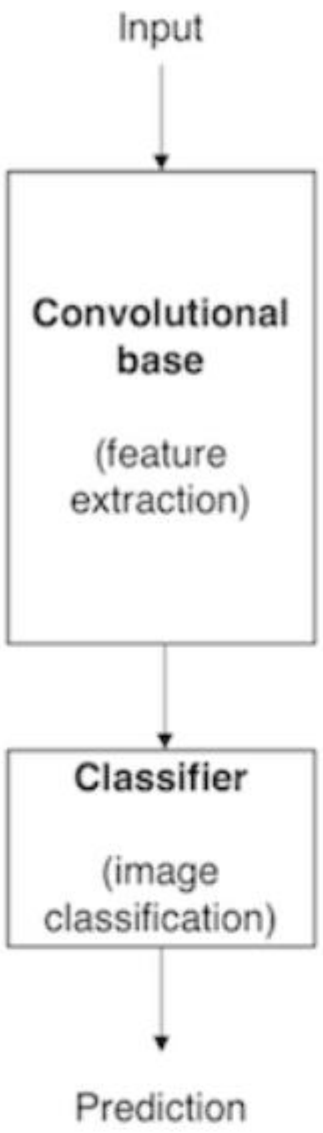
(d) Segmentation (3D)

e.g. Adapt a VGG network trained on ImageNet to classify each pixel

Advantage of TL

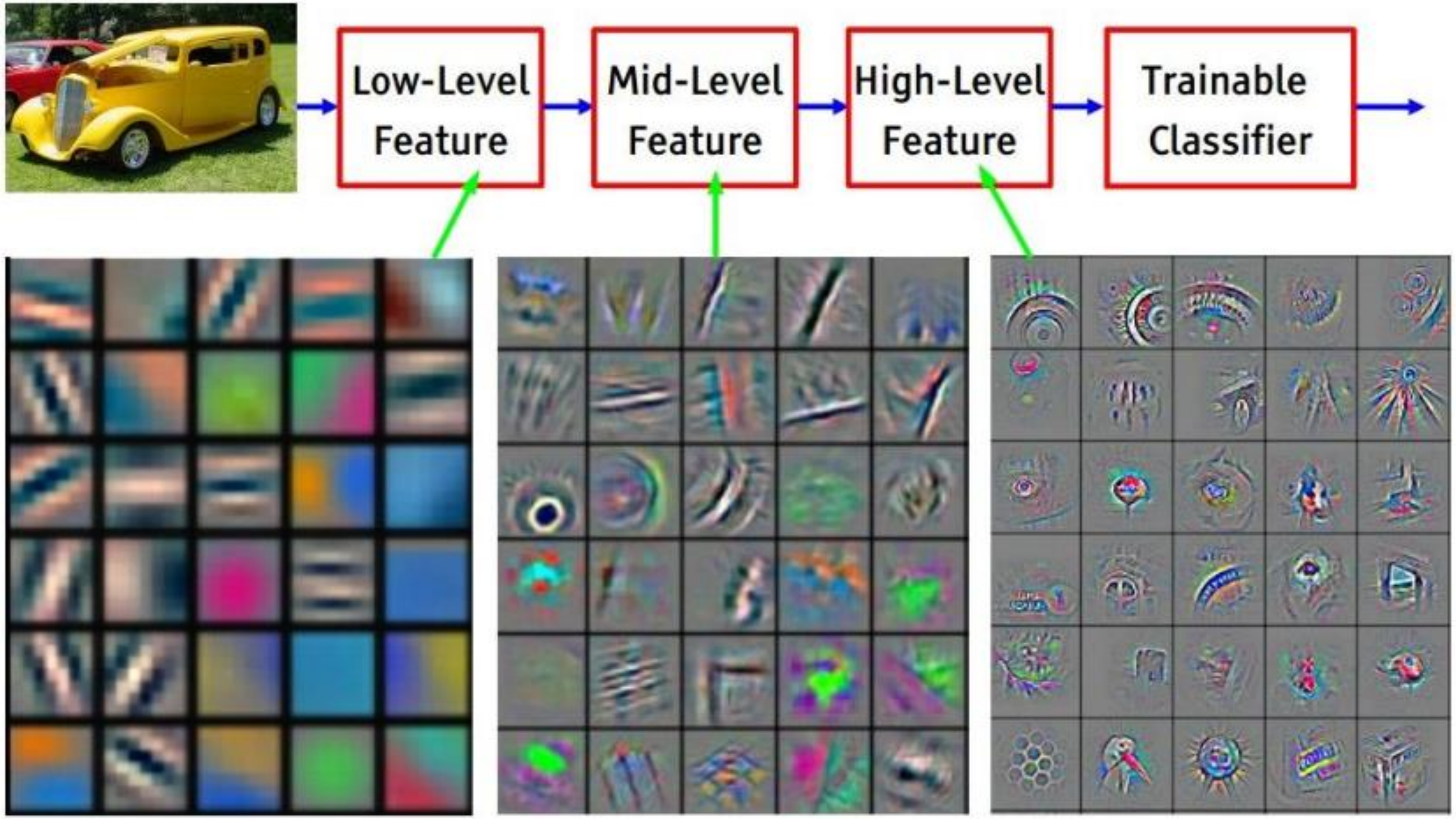


Way to do TL (common!)



Given a network trained on lots of data

Why this way?

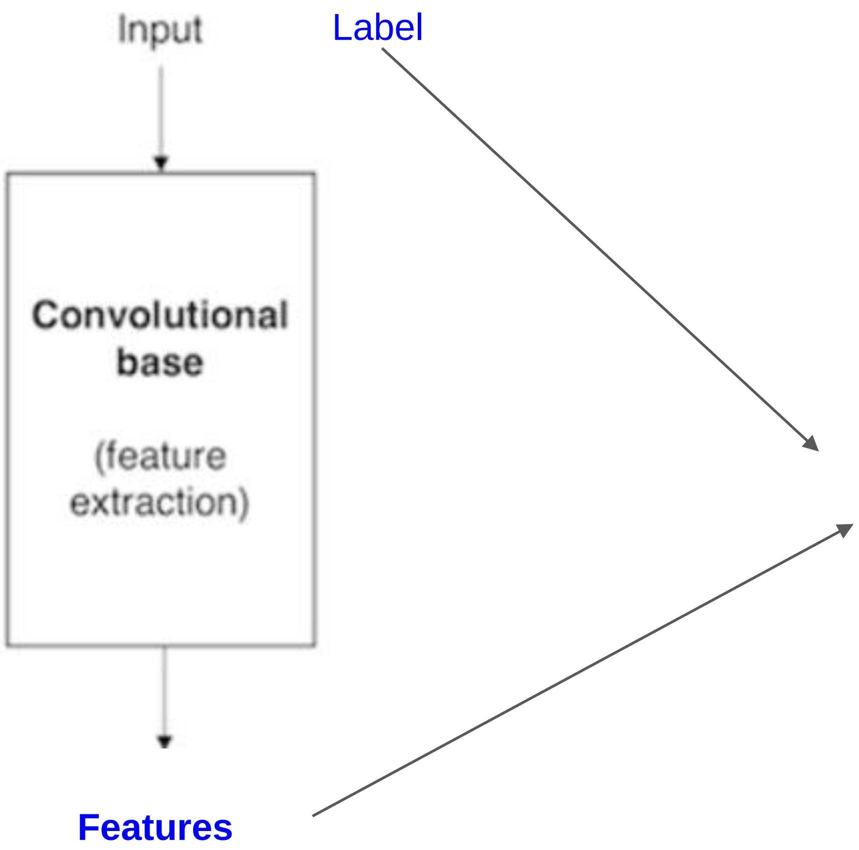


Highly Transferable: Bottom layers capture low-level features which are likely to be re-usable across applications

Less Transferable: Top layers capture high-level features which are likely to be specific to application

Feature Extraction

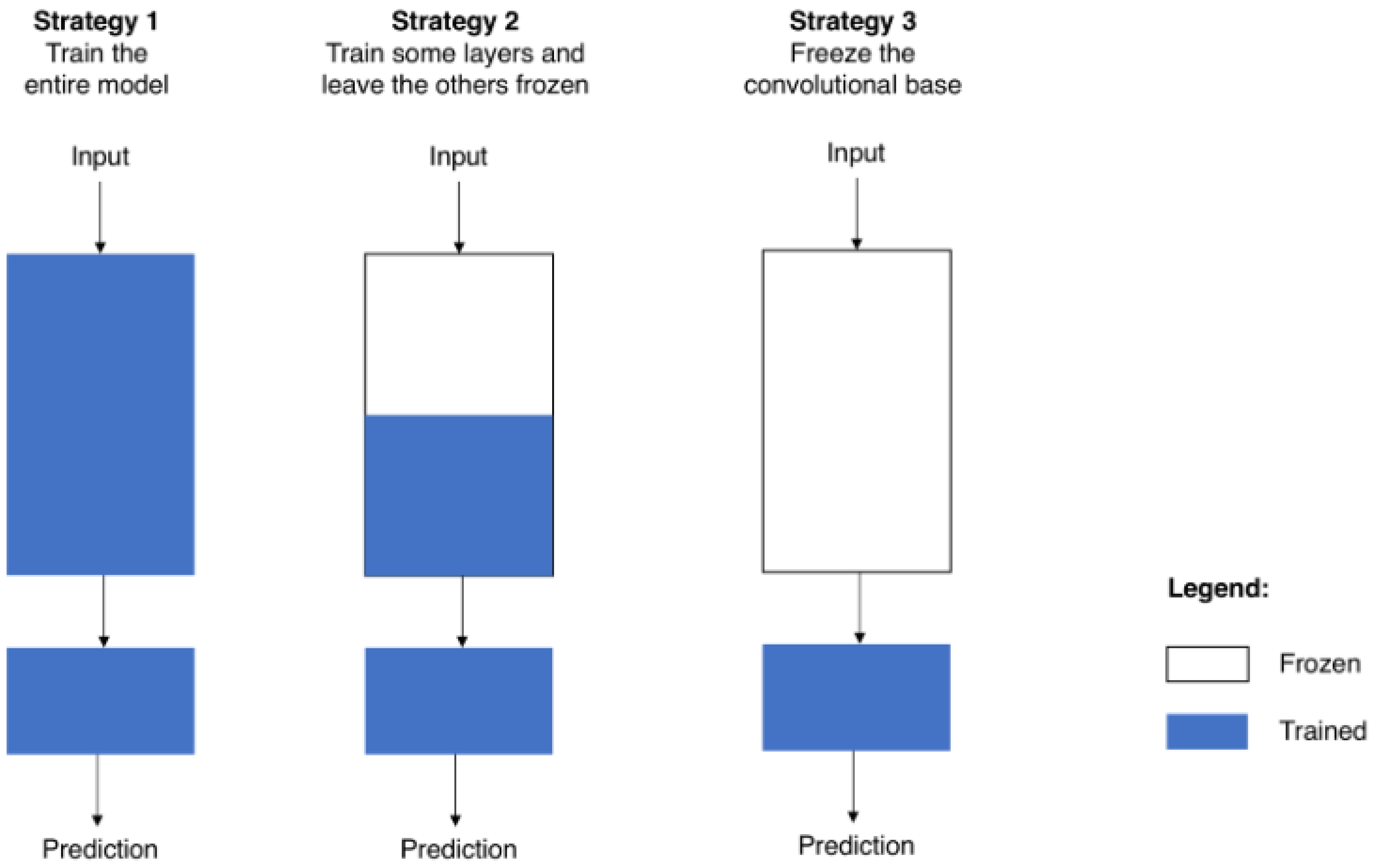
For a new supervised problem



Train ML classifier
(e.g. RF, DT, SVM, DNN etc.)

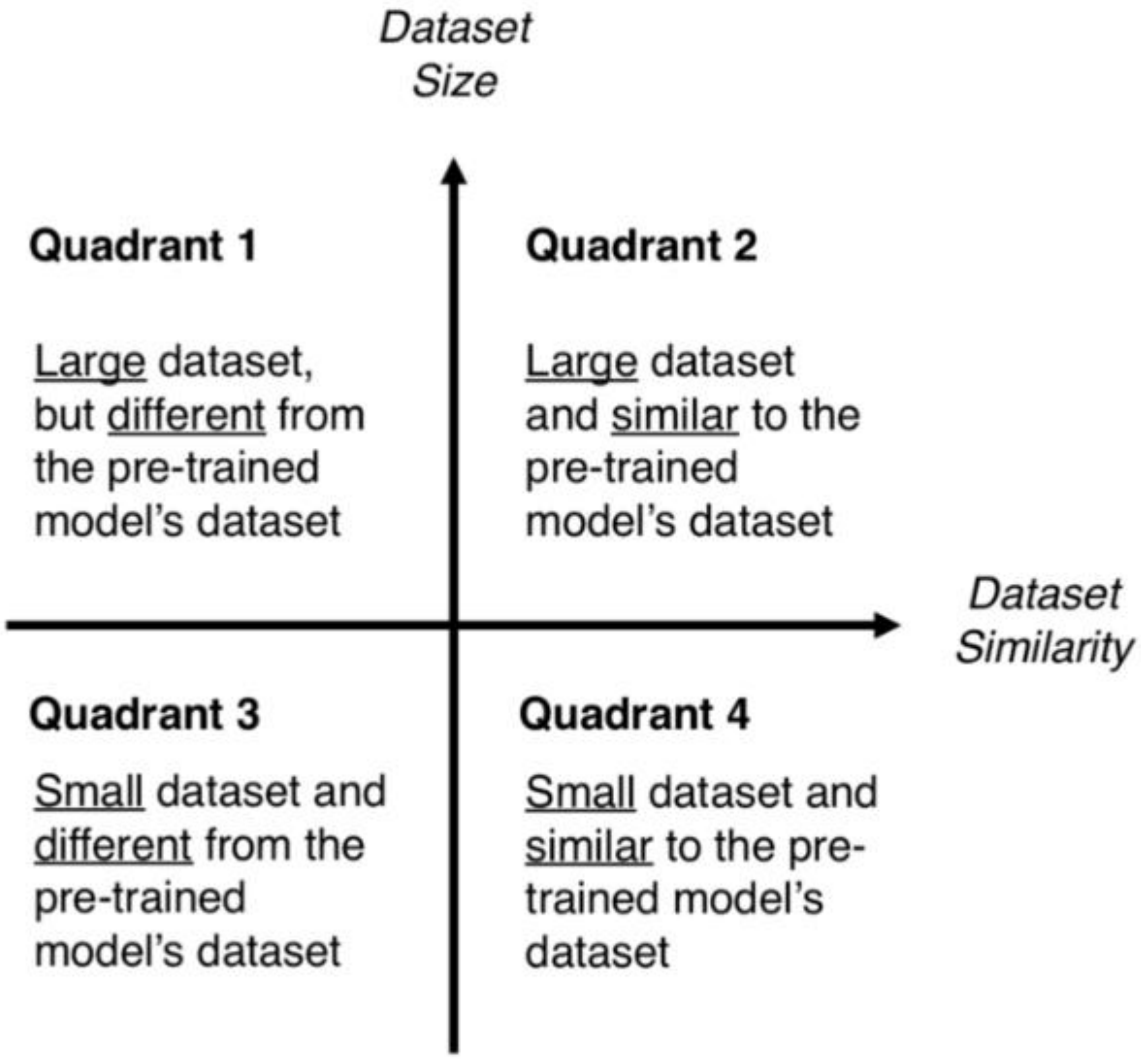
Typically for less data

DL Network route



Can you guess the TL scenario for above settings in terms of Domain, Tasks, amount of data?

Different cases: TL



Dataset Size

Dataset Similarity

Quadrant 1

Large dataset, but different from the pre-trained model's dataset

Quadrant 2

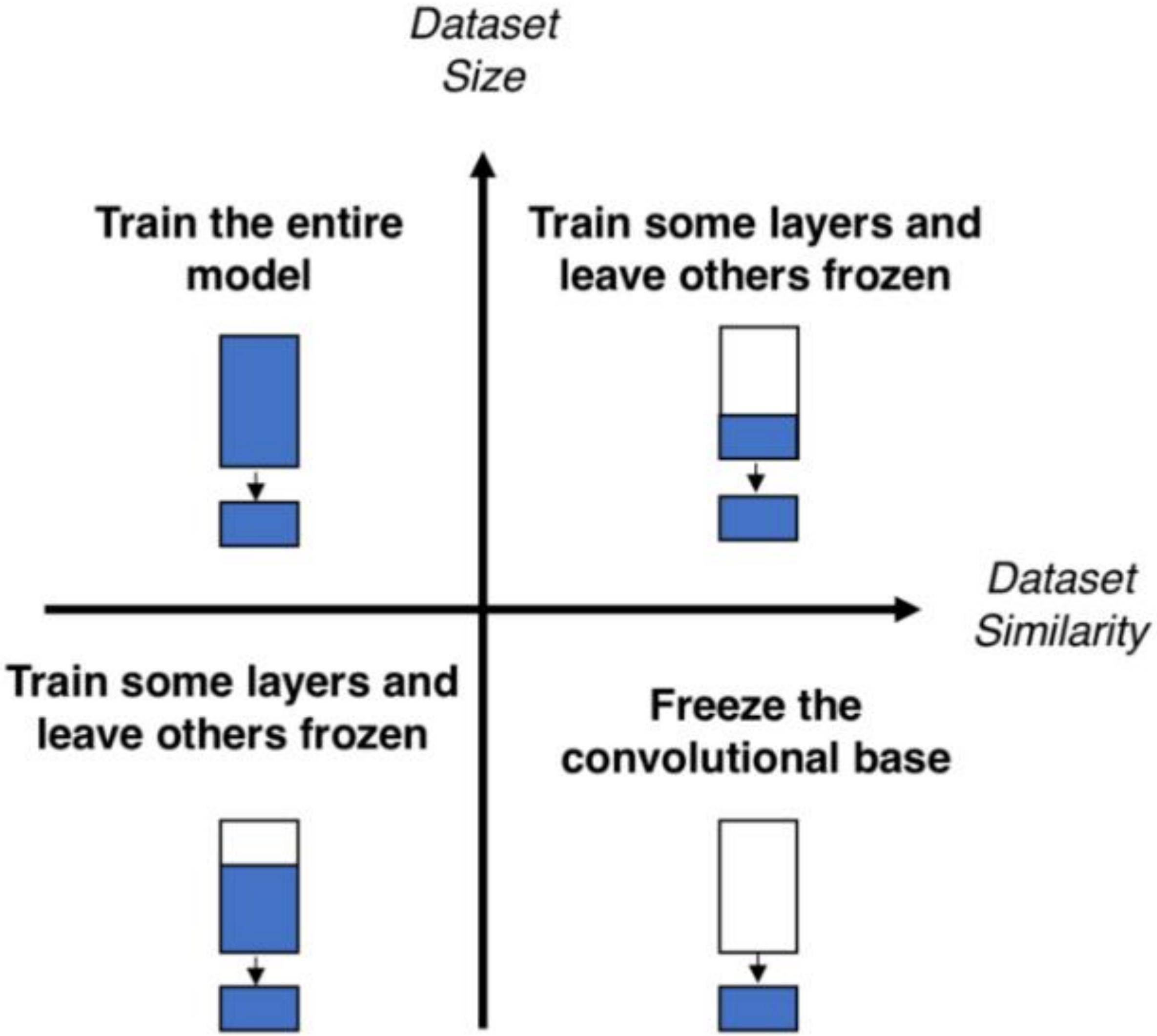
Large dataset and similar to the pre-trained model's dataset

Quadrant 3

Small dataset and different from the pre-trained model's dataset

Quadrant 4

Small dataset and similar to the pre-trained model's dataset



Dataset Size

Dataset Similarity

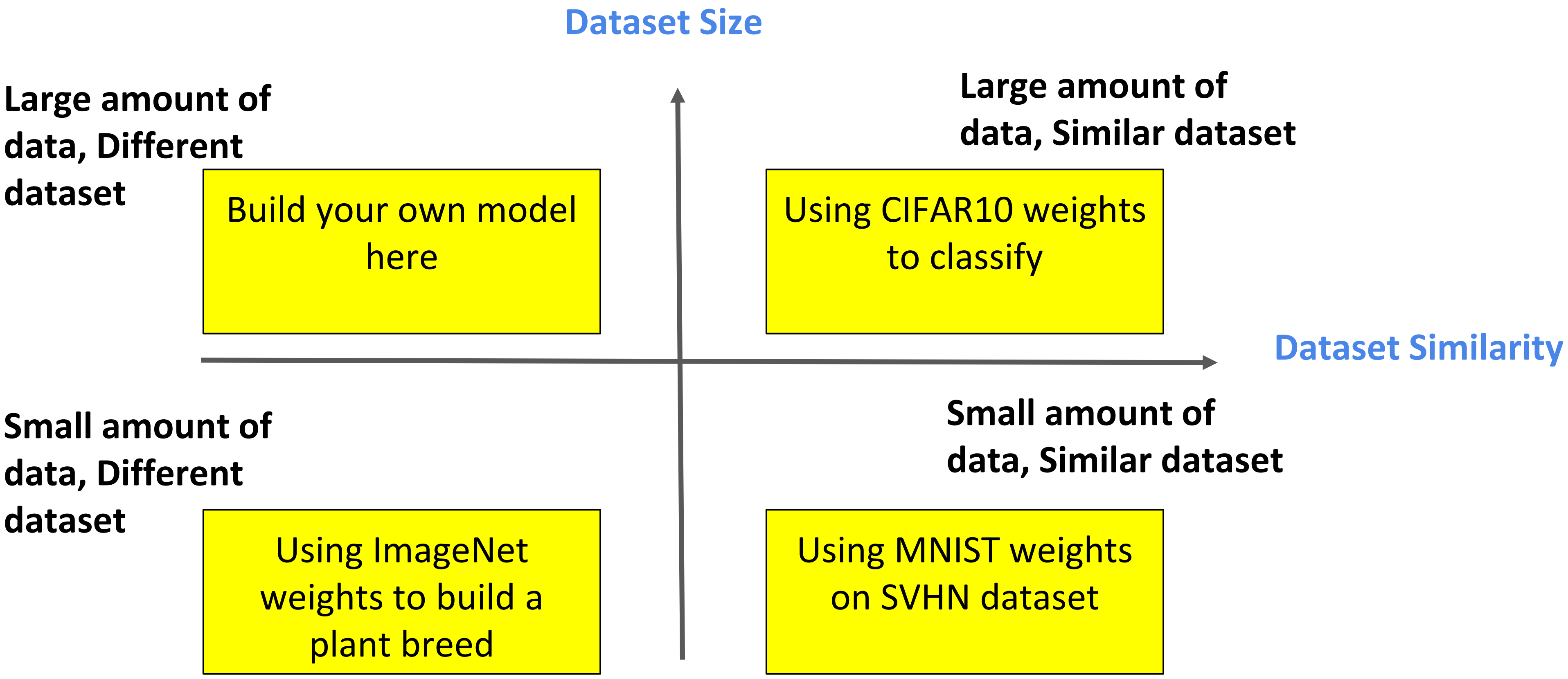
Train the entire model

Train some layers and leave others frozen

Train some layers and leave others frozen

Freeze the convolutional base

Different cases: TL



Dataset Size

Dataset Similarity

Large amount of data, Different dataset

Build your own model here

Large amount of data, Similar dataset

Using CIFAR10 weights to classify

Small amount of data, Different dataset

Using ImageNet weights to build a plant breed

Small amount of data, Similar dataset

Using MNIST weights on SVHN dataset

TL process summary

1. Extract Features Using Conv networks and learn using ML classifier
2. Transfer full architecture from standard Conv networks
3. Transfer from related domains

Practices

1. Less data, Similar task
   a. Fine tune only classification layer

1. Small data, different task
   a. Fine tune last few Conv layers

1. Large data, Different task
   a. Fine tune entire network

1. Large data, same task
   a. Tune Dense first, then Conv, then full
   b. Pay attention to LR (typically small for Bottom layers)
   c. Differential LR across layers