greatlearning
Learning for Life

Transfer Learning for Computer Vision

Girish Gore
www.linkedin.com/in/datascientistgirishgore

REOMNIFY
Location Intelligence Simplified

About the Speaker !

THINKBIGGER ANALYTICS
Data | Discovery | Decisions

Professional Experience

- ➔ 10+ Years of Industry Experience (5+ Y: SAS Analytics ,3+ Y : Cognizant , 2+: Start Ups)
- ➔ Artificial Intelligence,Algorithms , Analytics & Data Products
- ➔ Data Scientist by passion

Training Experience

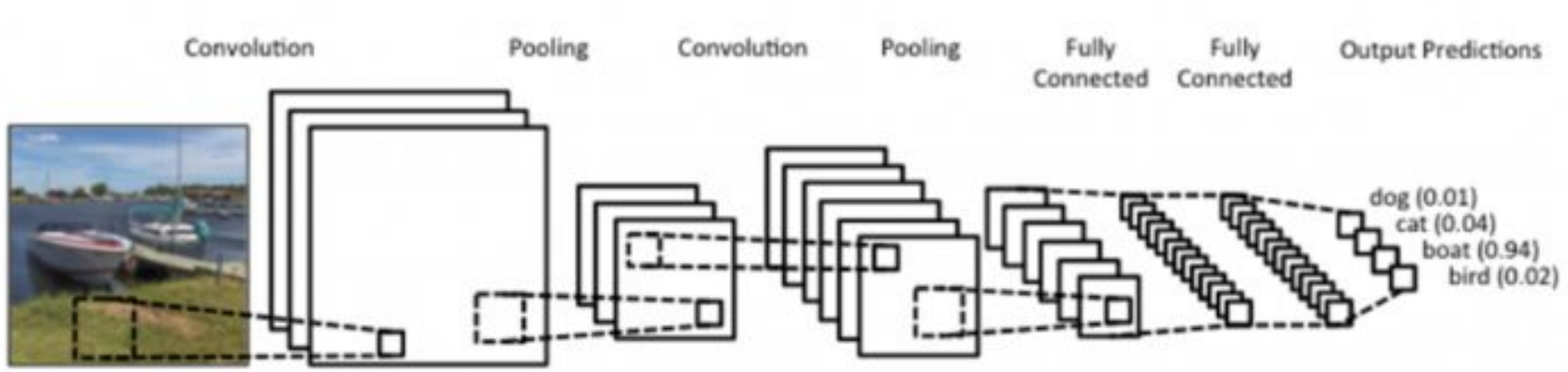
- ➔ Mentor in Data Science at SpringBoard, SF
- ➔ Adjunct Faculty : Machine & Deep Learning Using Python at Aegis School Of Data Science
- ➔ Adjunct Faculty : Advanced Deep Learning at Great Learning
- ➔ 4+Y Training Corporates & Executives with experience ranging from 5 years to 25+ years

Hobbies

- ➔ Playing Tennis , Foodie (btw a pure Vegetarian :))
- ➔ Meditation & Vedic Philosophy

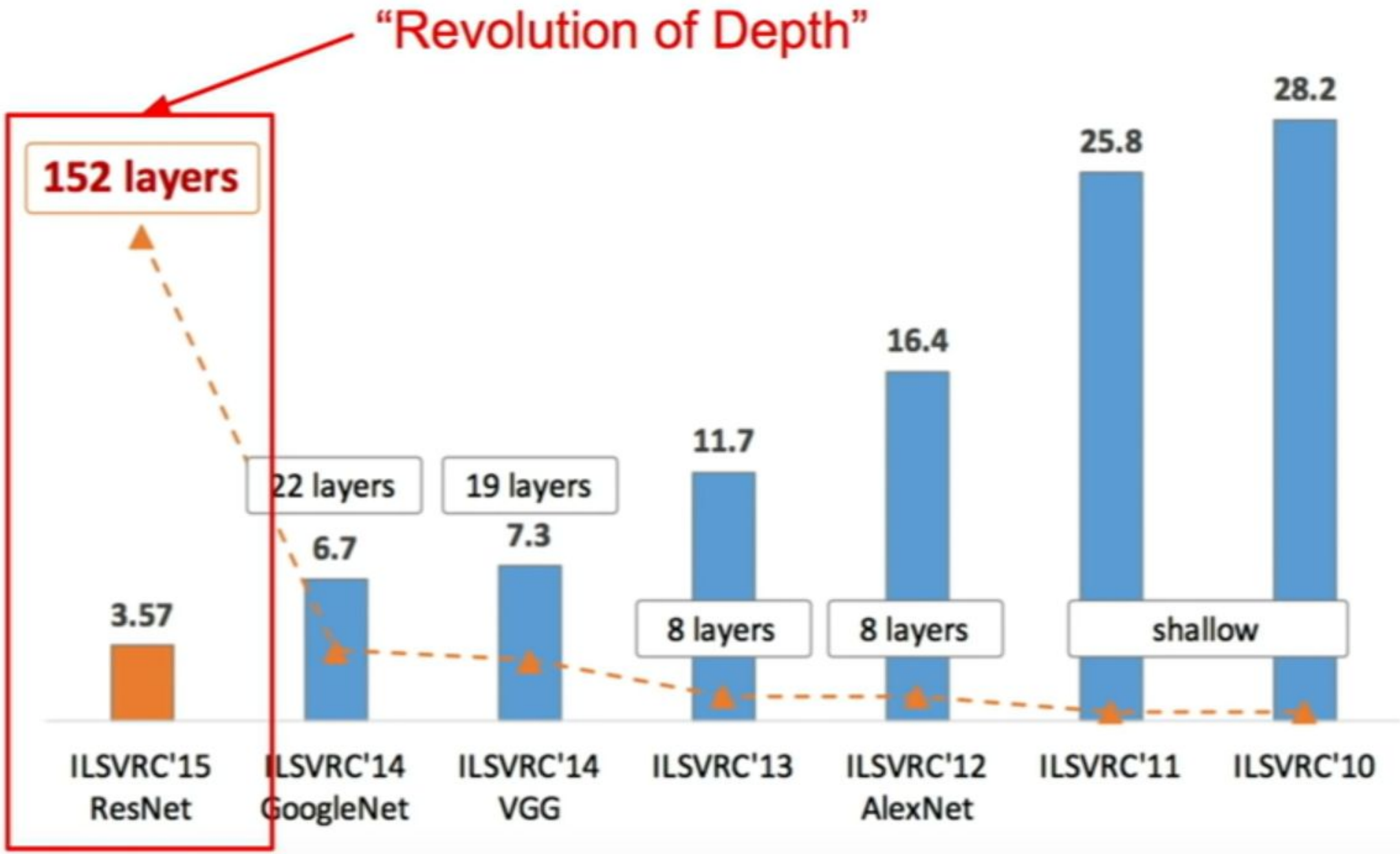
Revising CNN's for Object Classification

Typical ConvNet



$$n_{out} = \left\lfloor \frac{n_{in} + 2p - k}{s} \right\rfloor + 1$$

n_{in}: number of input features
n_{out}: number of output features
k: convolution kernel size
p: convolution padding size
s: convolution stride size


"Revolution of Depth"
152 layers
22 layers
19 layers
8 layers
8 layers
shallow
3.57
6.7
7.3
11.7
16.4
25.8
28.2
ILSVRC'15 ResNet
ILSVRC'14 GoogleNet
ILSVRC'14 VGG
ILSVRC'13
ILSVRC'12 AlexNet
ILSVRC'11
ILSVRC'10

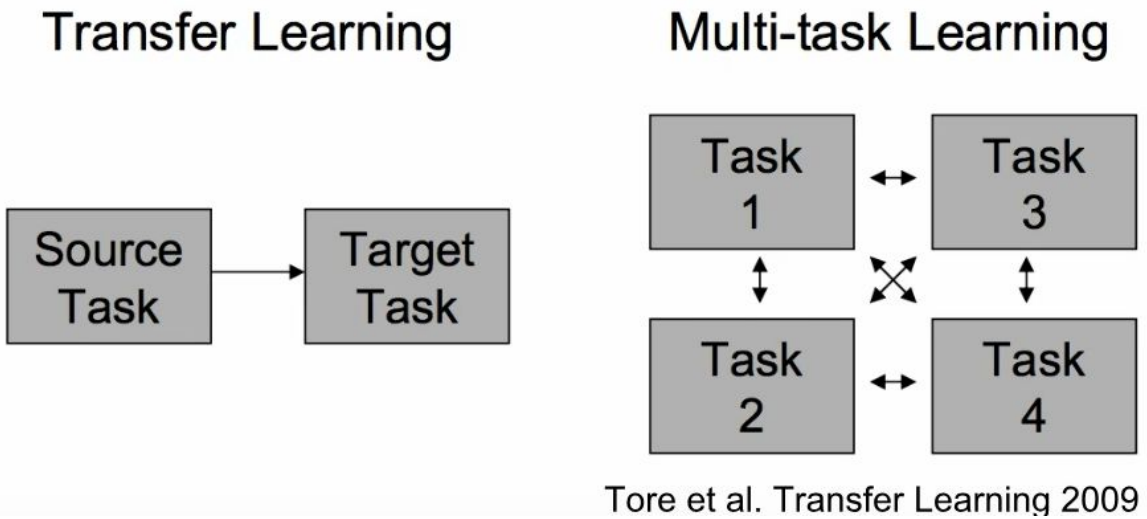
Module objectives

- Transfer Learning vs Multi Task Learning
- What is Transfer Learning in DNN
- Different Use Cases and Approaches
- Actual Applications

Multi Task Learning vs Transfer Learning

Multi-task Learning:
Learning different tasks in a single model.

Transfer Learning:
Learning target task by transferring knowledge from source task(s).



Tore et al. Transfer Learning 2009

Transfer Learning [TL] for Deep CNN

- Modern architectures are really deep and need a lot of data to train from scratch
- Training from scratch is time Consuming & expensive
  - Data Collection
  - Data Labeling
  - Training Models
- Applied in variety of fields / domains e.g. Healthcare, Retail, Surveillance etc. Its more practical to use knowledge from one domain / task and reuse

Why Transfer Learning?

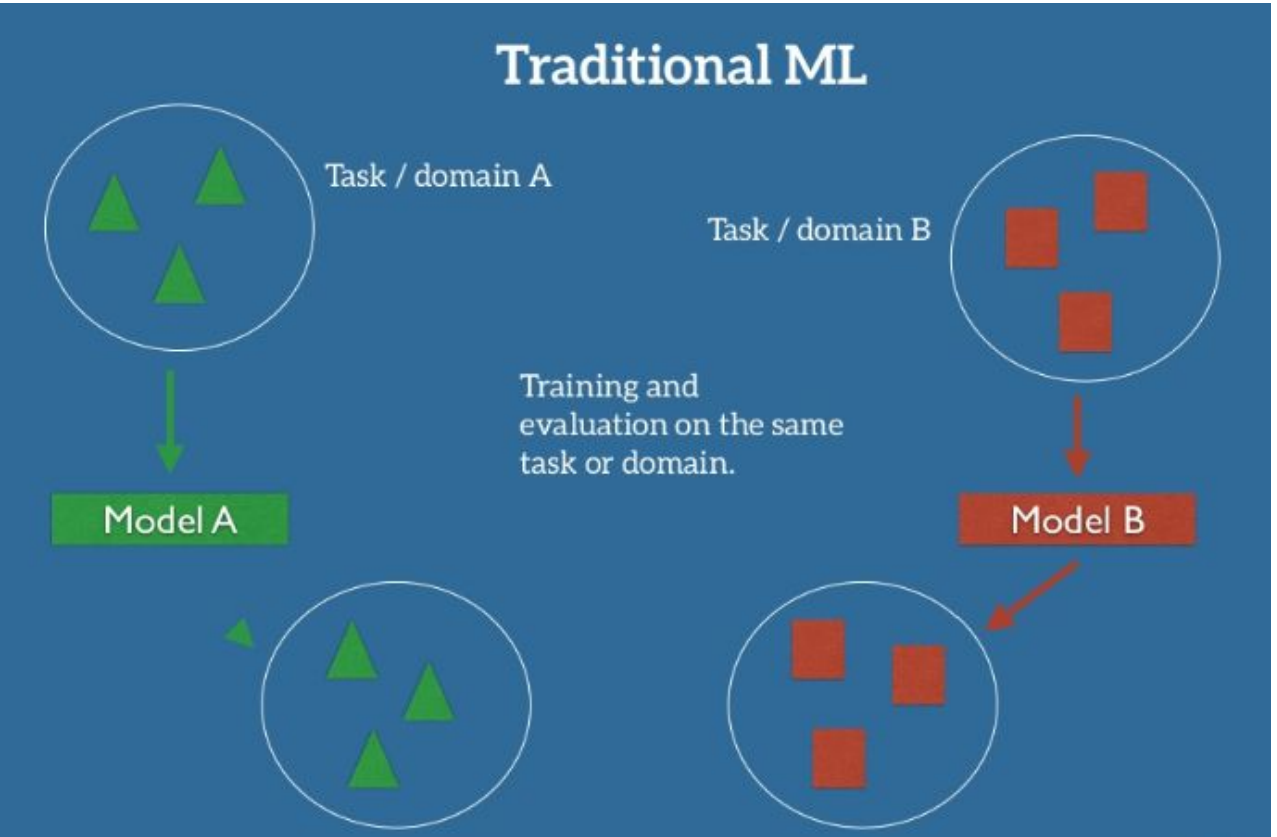
- Given a new application, one looks at opportunities for re-using knowledge from similar learning problems which were trained with large amounts of data

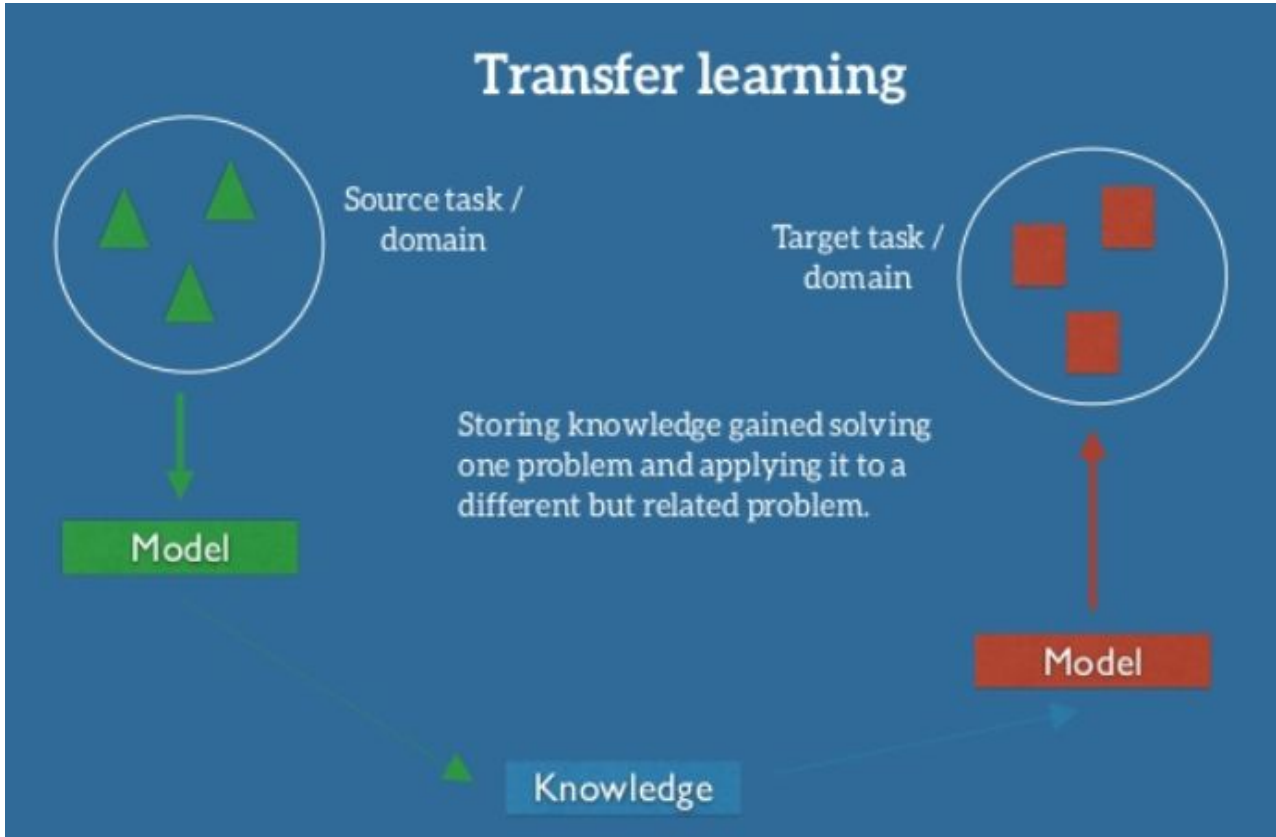
  Transfer Learning!

- Key things transferred are
  - Architecture
  - Weights (learning parameters)

- Humans are great at transfer learning
  - Bicycle → Bike,
  - Tennis → Badminton,
  - Language → skills

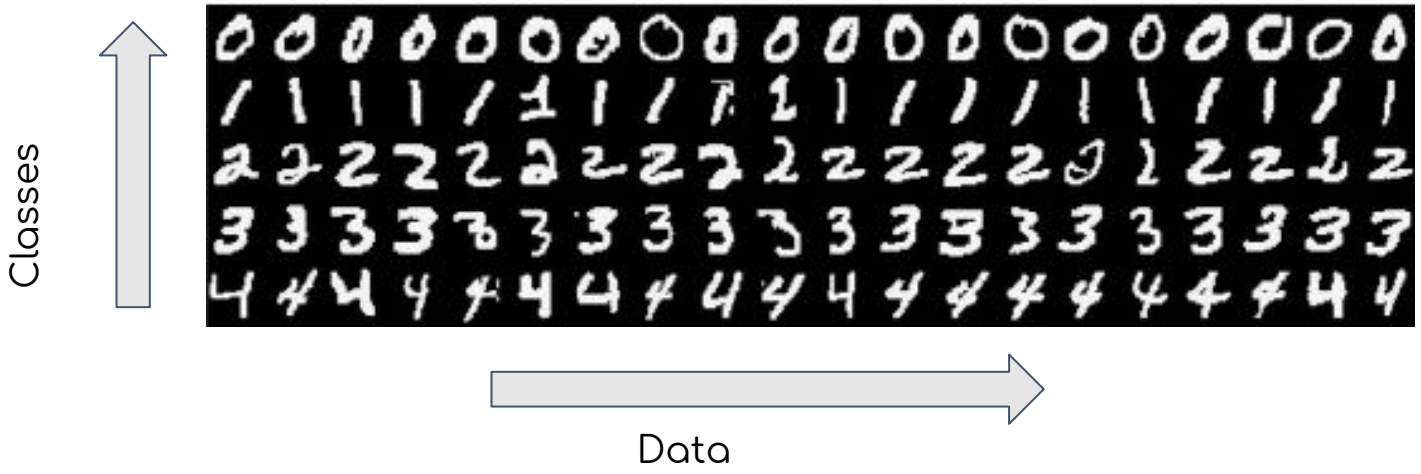
Transfer Learning - How it works?





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

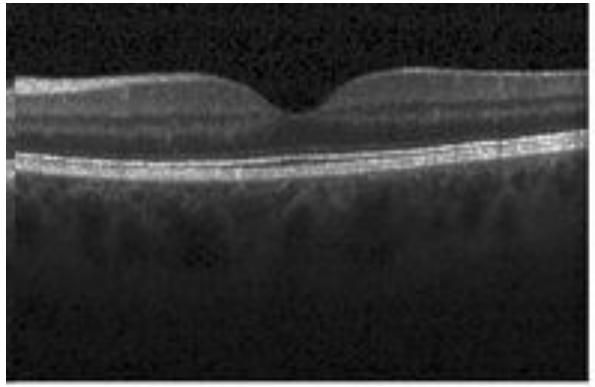
Want to build an alphabet classifier

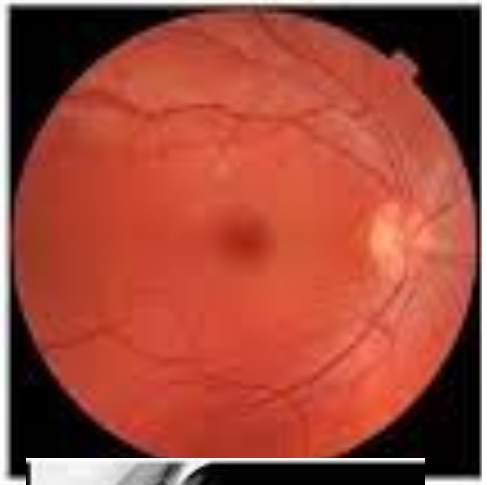
With the knowledge of how to build a classifier model which can classify digits 0 to 4, can we build a model which can classify alphabets?

Applications of Transfer Learning

Transfer learning from ImageNet examples

Medical Data disease classification





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

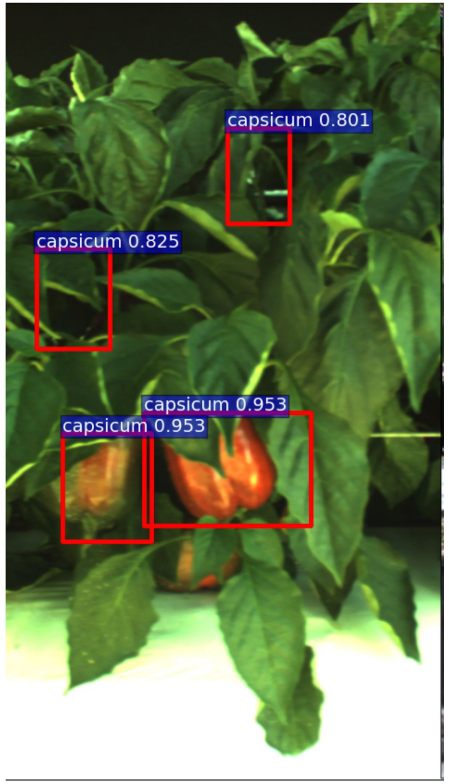
Applications of Transfer Learning

greatlearning
Learning for Life

Object Detection/ Recognition

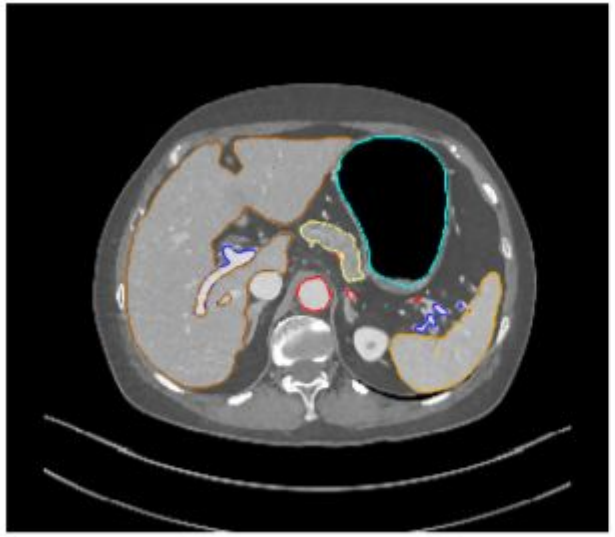
Less Data with bounding boxes and labels

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

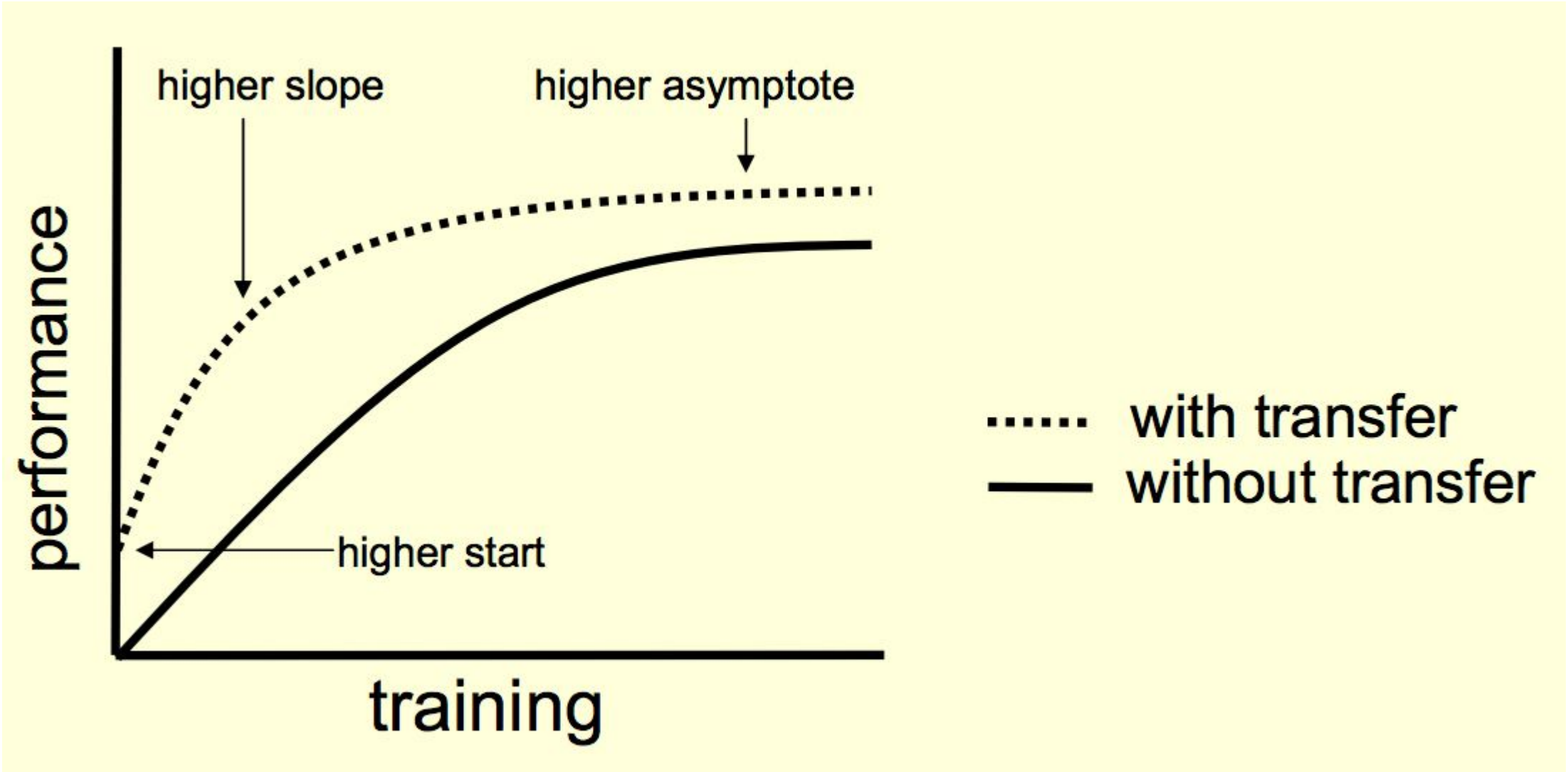


(c) Segmentation (axial)

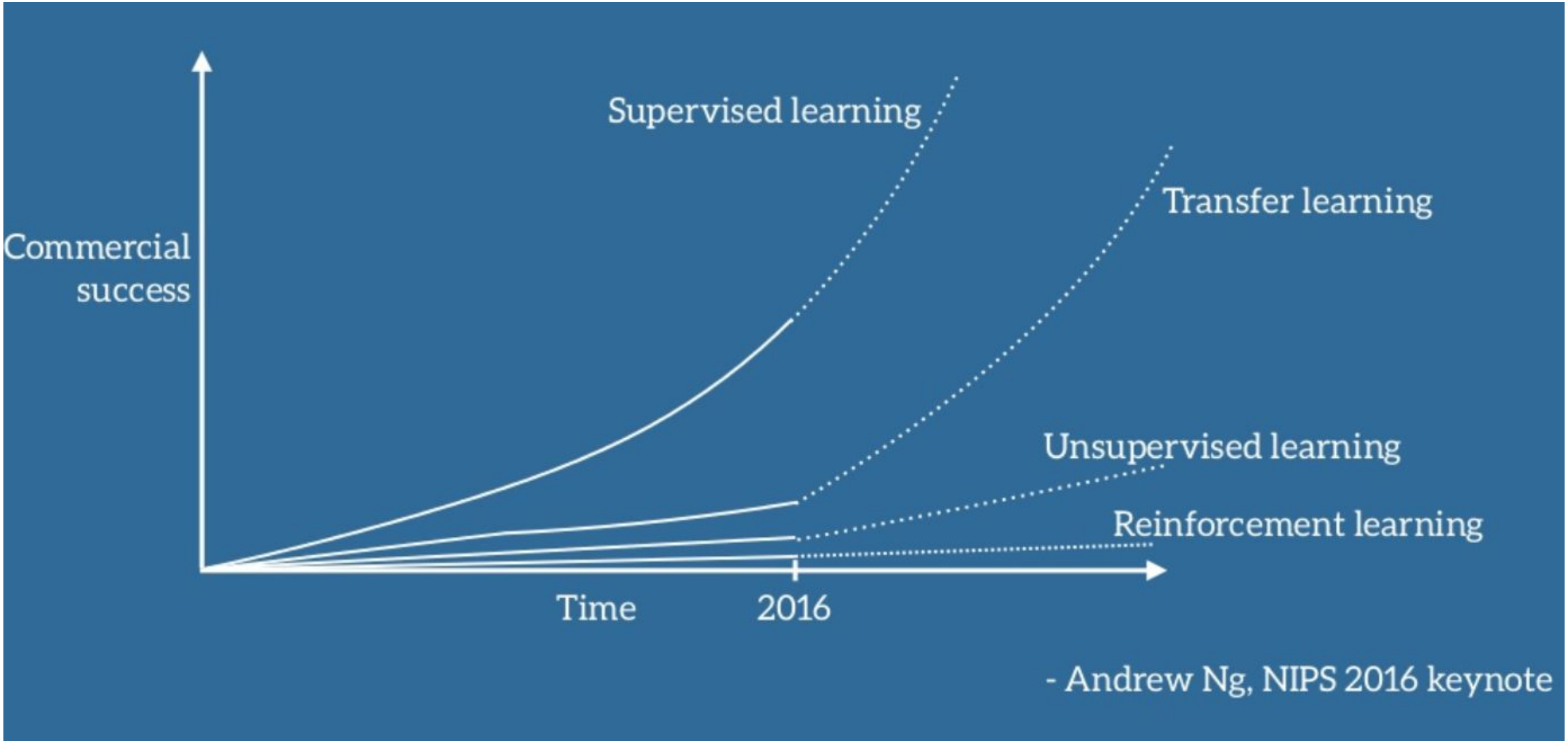
(d) Segmentation (3D)

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

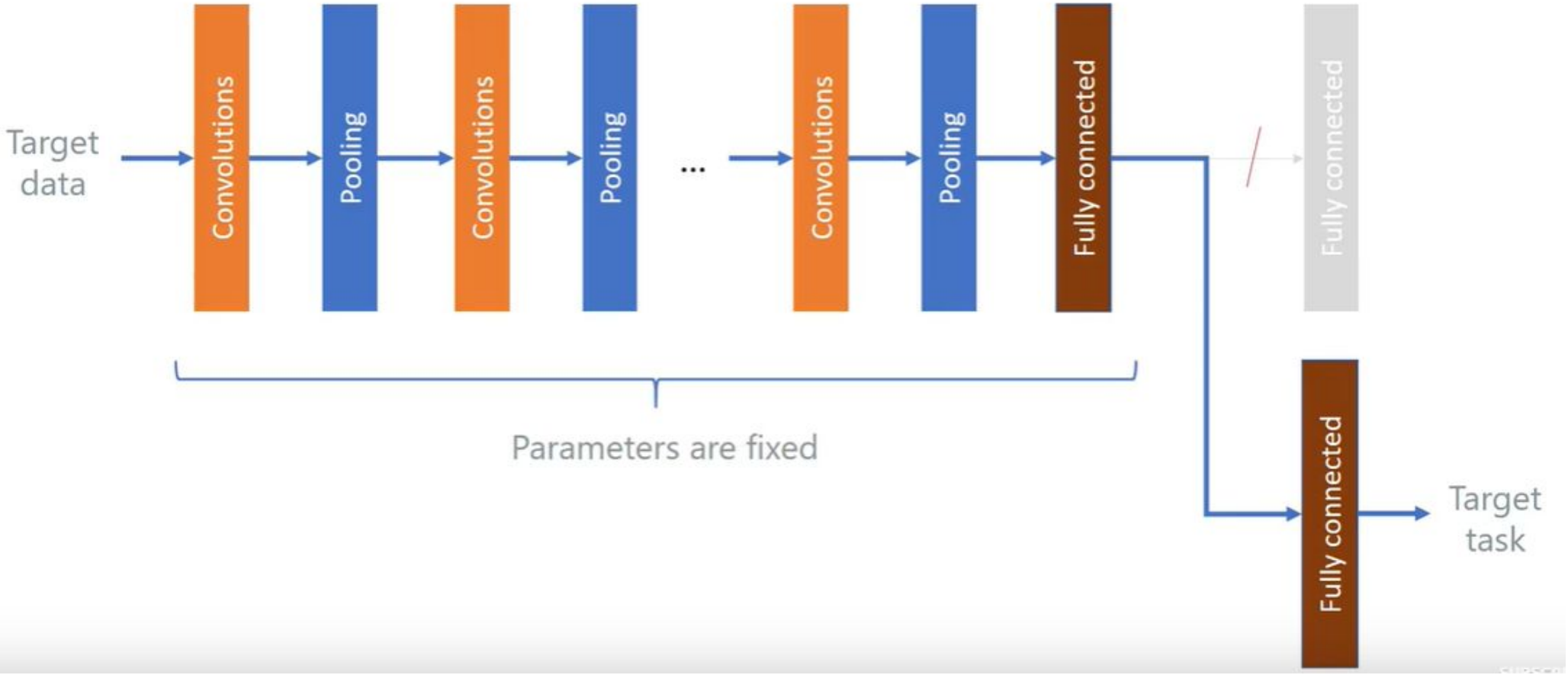
Advantage of TL



Why Now ?

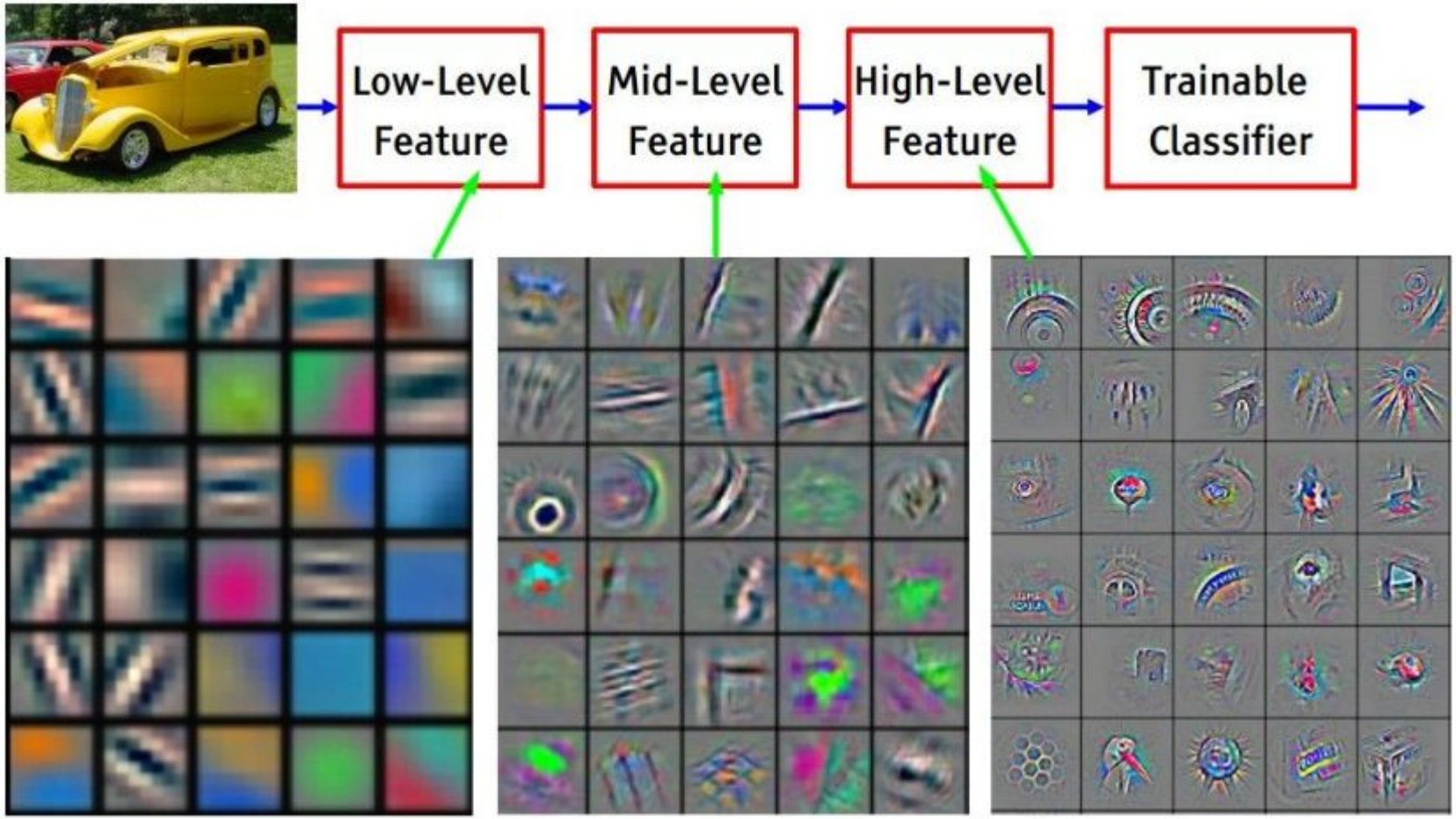

Supervised learning
Transfer learning
Commercial success
Unsupervised learning
Reinforcement learning
Time
2016
- Andrew Ng, NIPS 2016 keynote

Way to do TL (most common!)



Given a network trained on lots of data

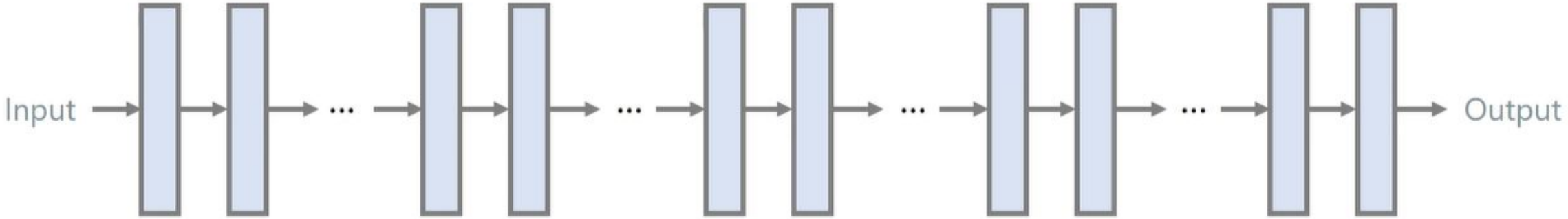
Why this way?

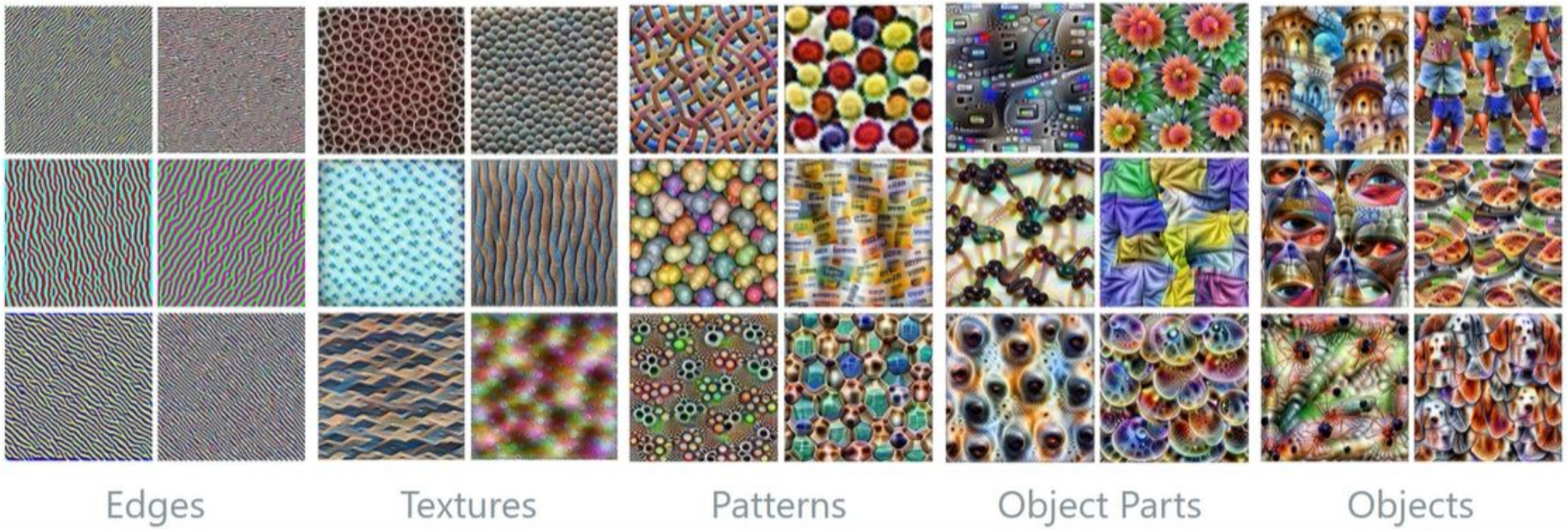


Highly Transferable: Bottom layers capture low-level features which are likely to be re-usable across applications

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

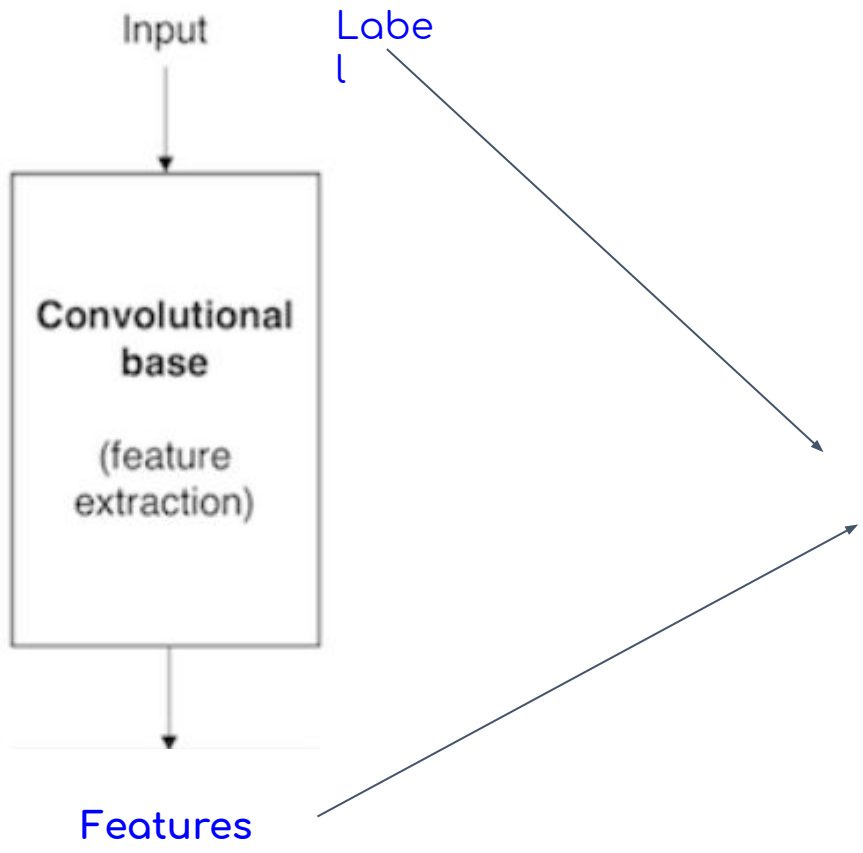
How Conv Networks learn ...





Feature Extraction

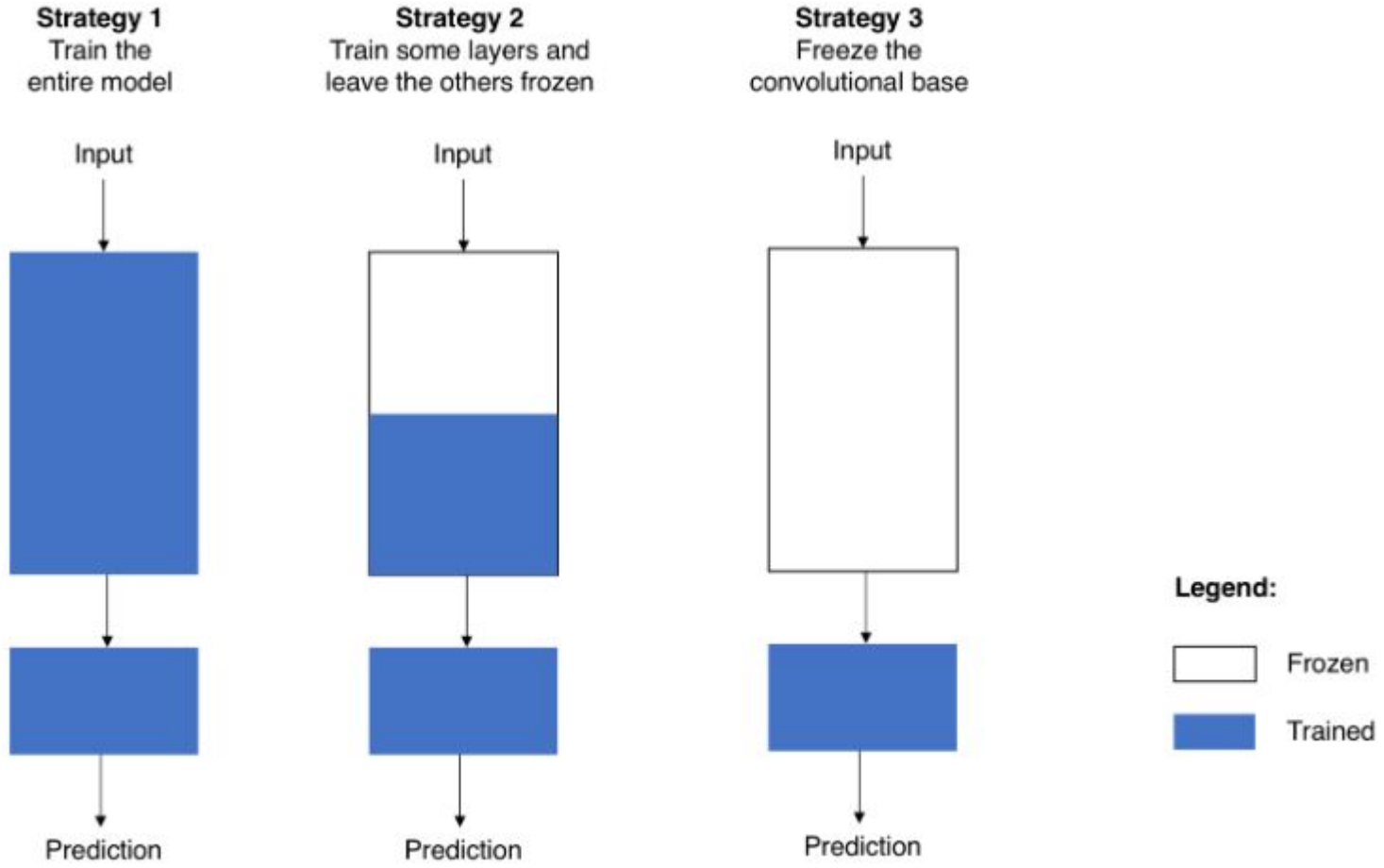
For a new supervised problem



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

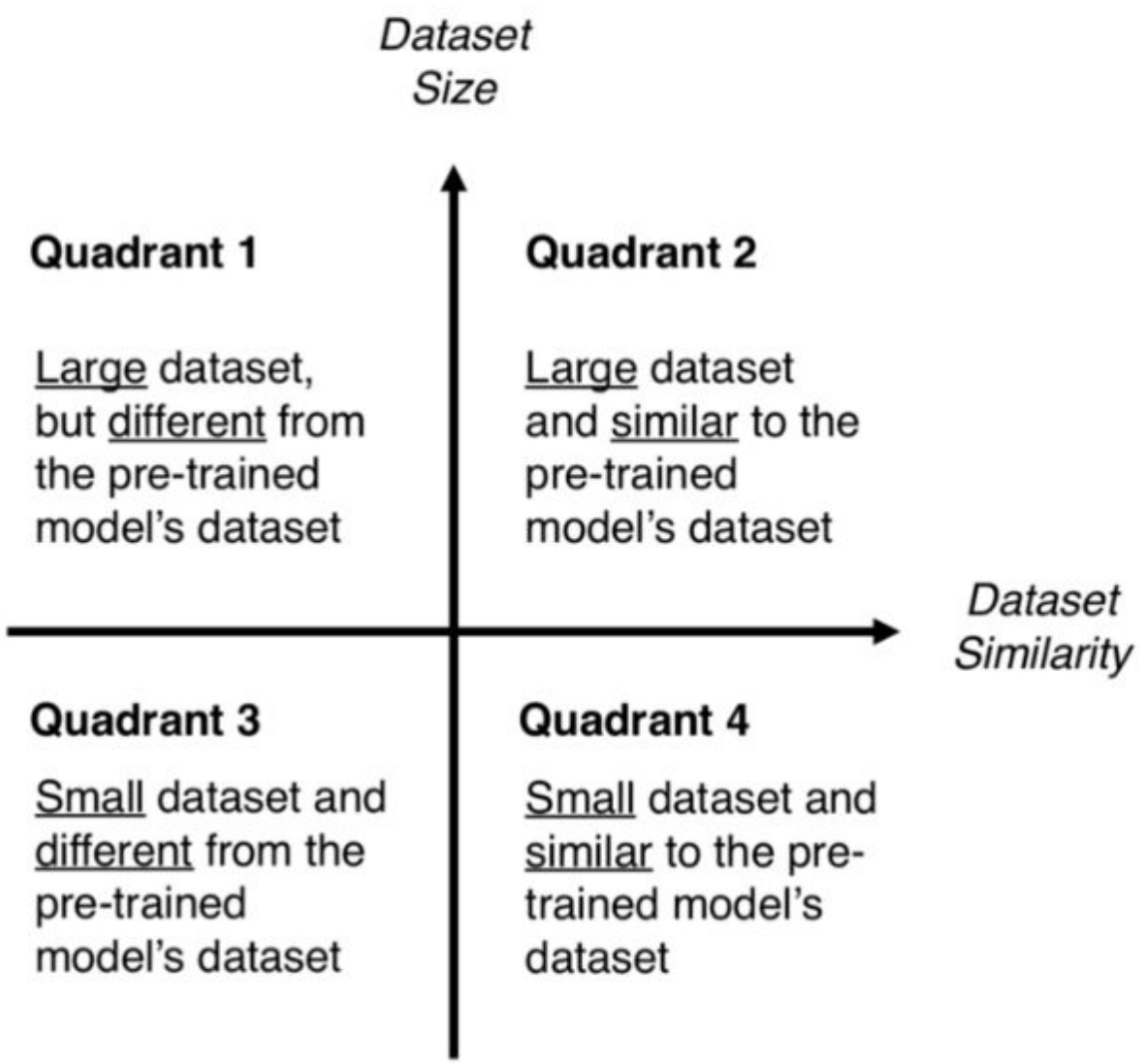
Typically for less data

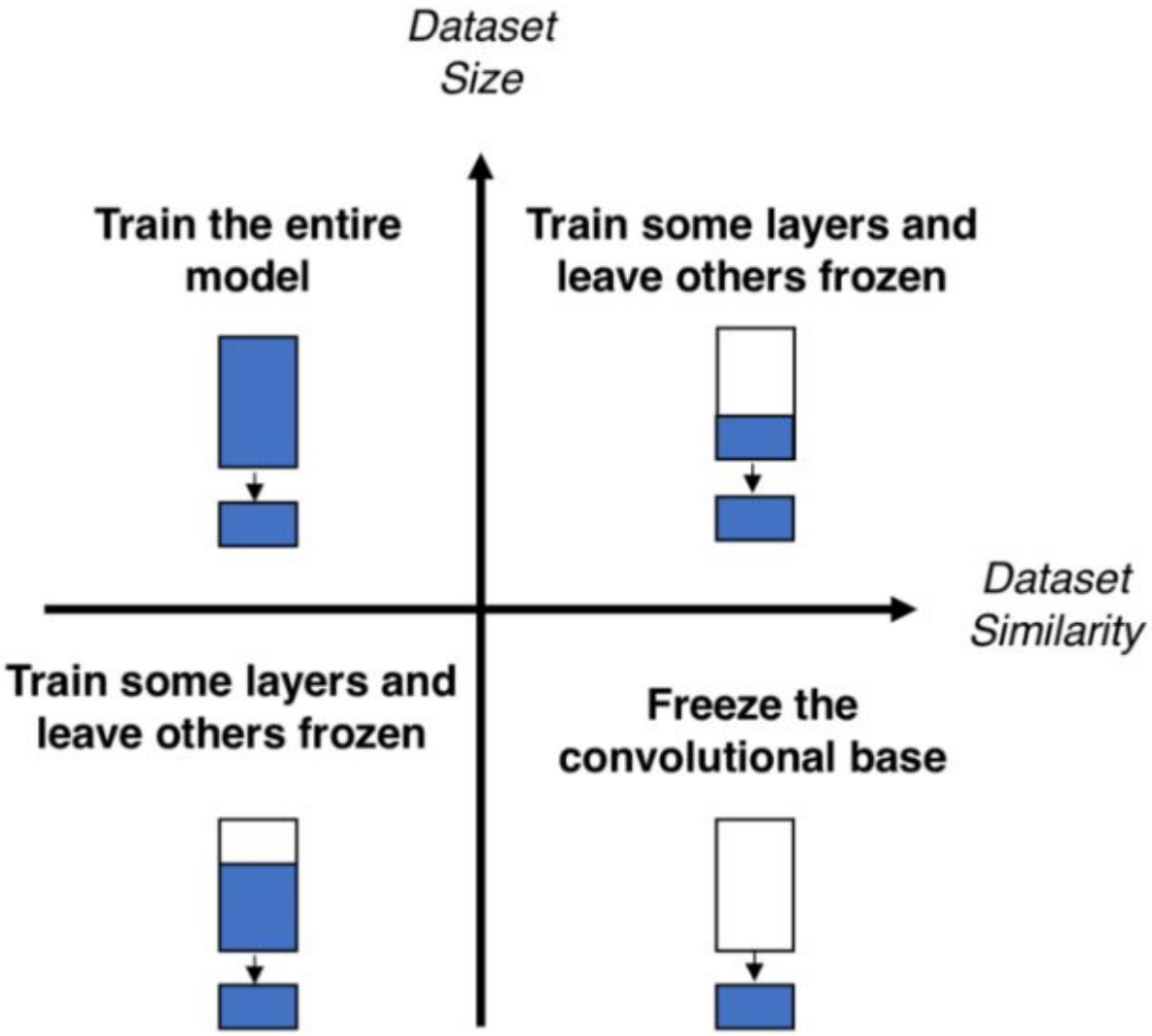
DL Network route



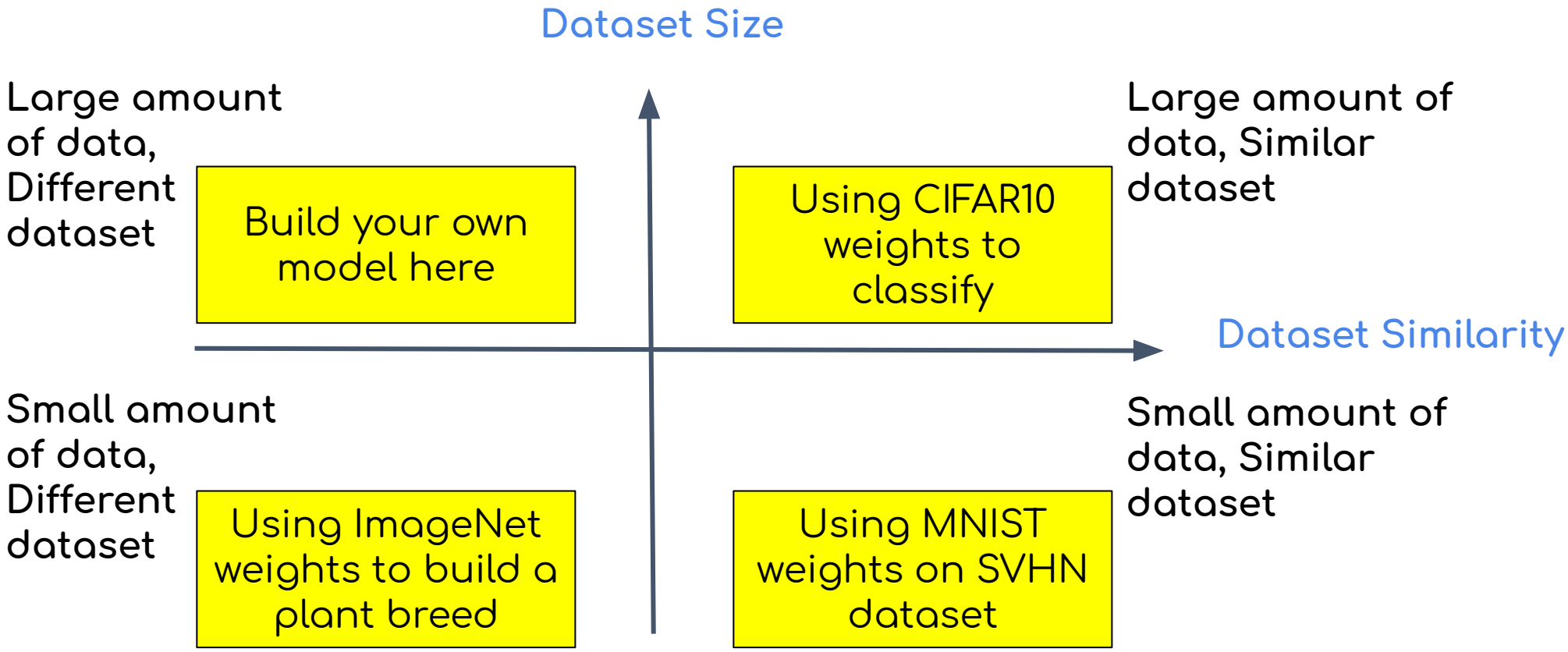
Can you guess the TL scenario for above settings in terms of Domain, Tasks, amount of data?

Different cases: TL





Different cases: TL



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

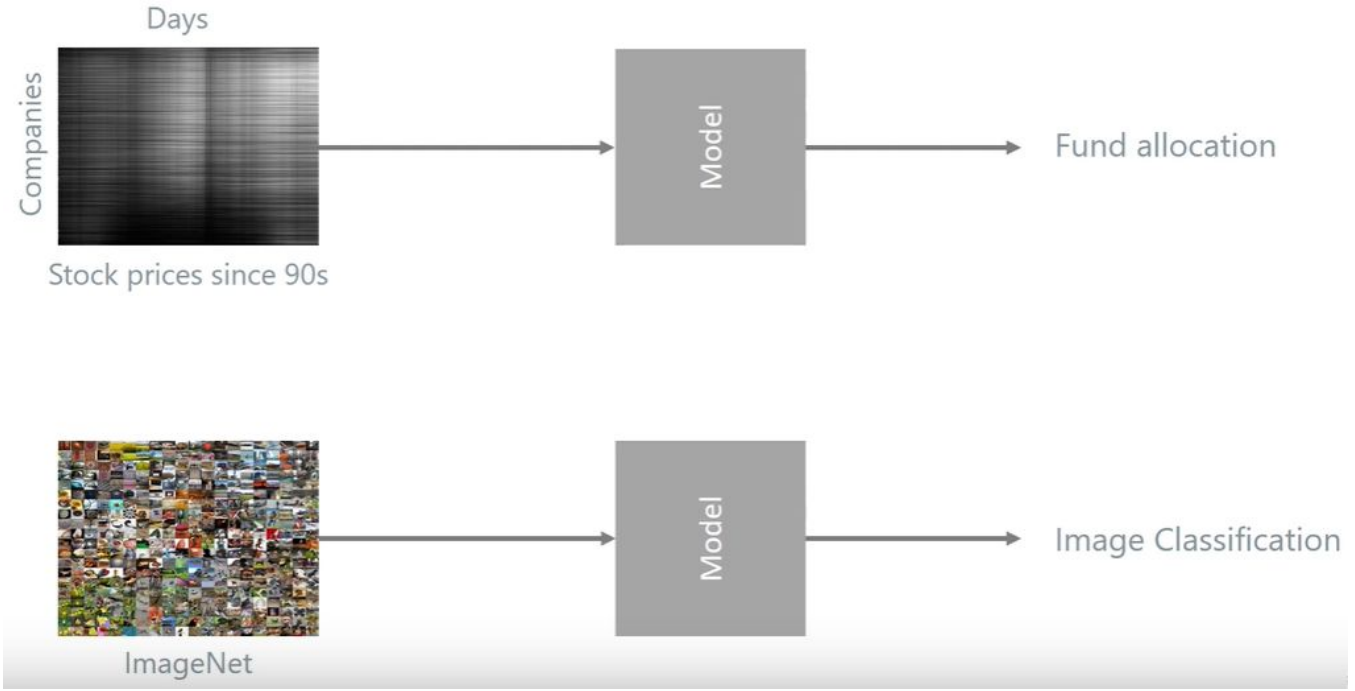
4. Key Terms
   - Pre Training
   - Fine Tuning

Practices

- Less data, Similar task
  - Fine tune only classification layer
- Small data, different task
  - Fine tune last few Conv layers
- Large data, Different task
  - Fine tune entire network
- Large data, same task
  - Tune Dense first, then Conv, then full
  - Pay attention to LR (typically small for Bottom layers)
  - Differential LR across layers

Limitations

- Both types of data & tasks are vastly different. Cannot be used.



- Architectures vastly vary