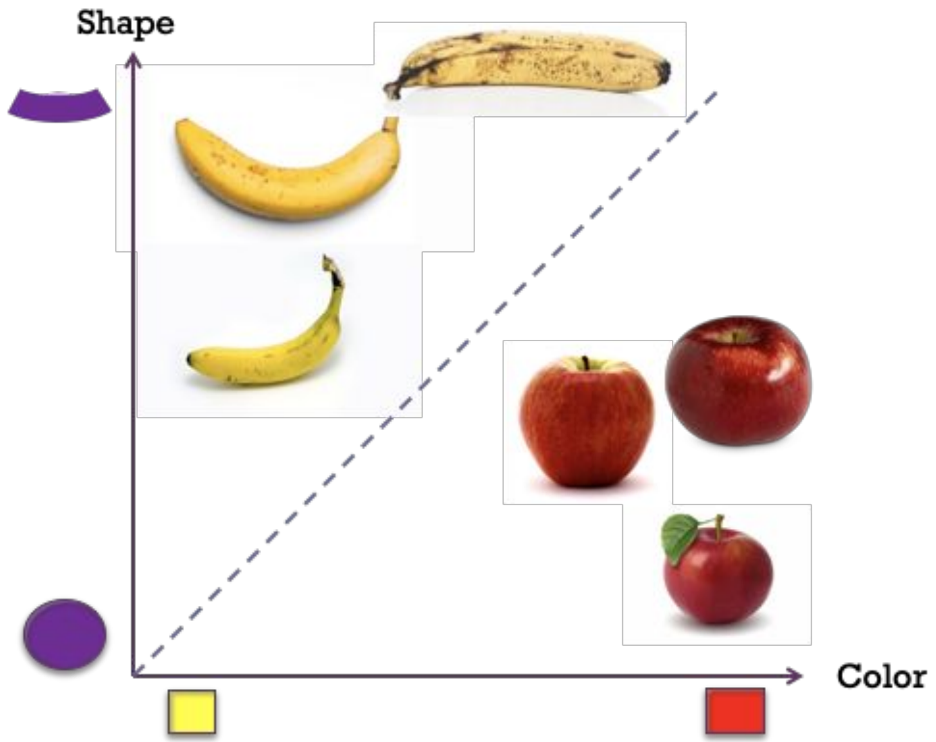
Transfer Learning

Slides courtesy of
Peerapon Vateekul, Ph.D

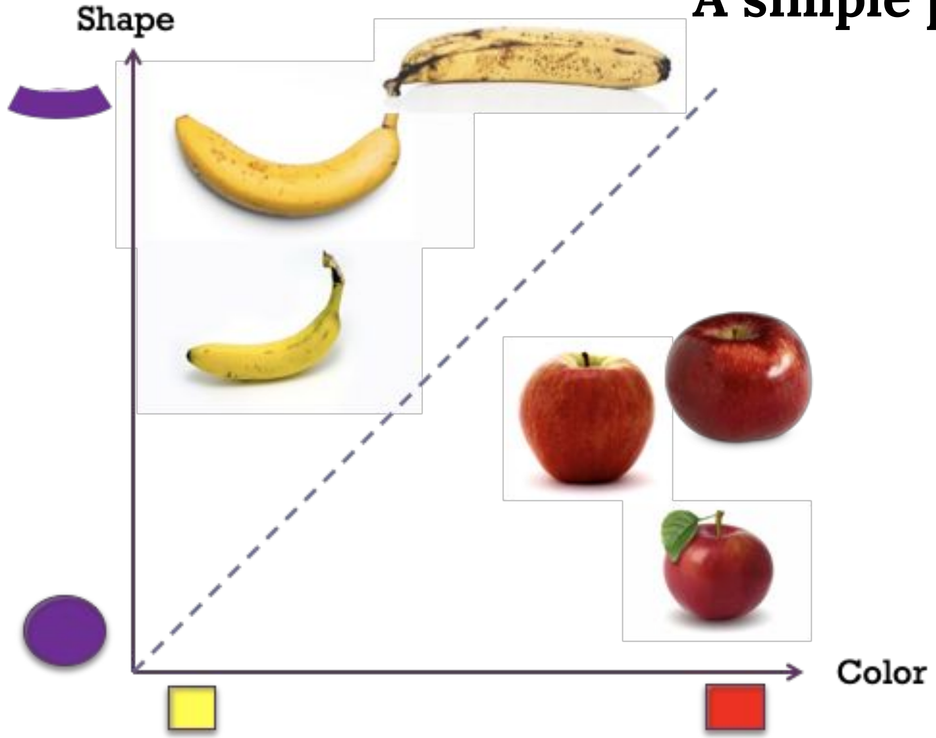
Outline

1 Introduction

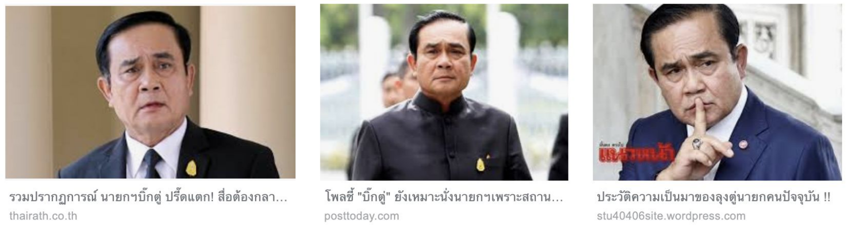
We need enough training data to infer the data pattern and to create model



We need enough training data to infer the data pattern and to create model (cont.)



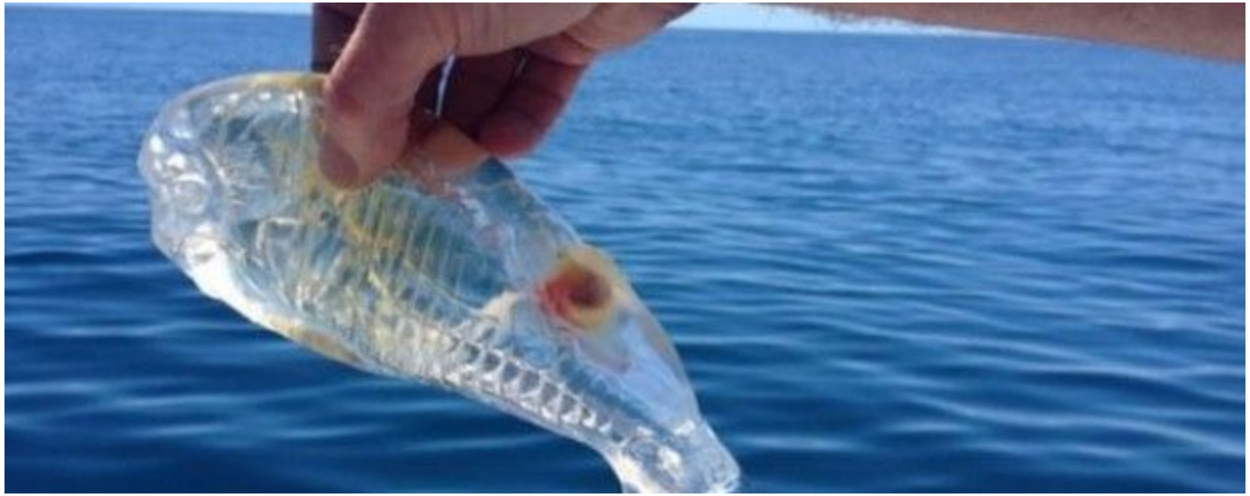
A simple problem requires less training data


รวมปรากฏการณ์ นายกฯบิ๊กตู่ ปรี๊ดแตก! สื่อต้องกลา...
thairath.co.th

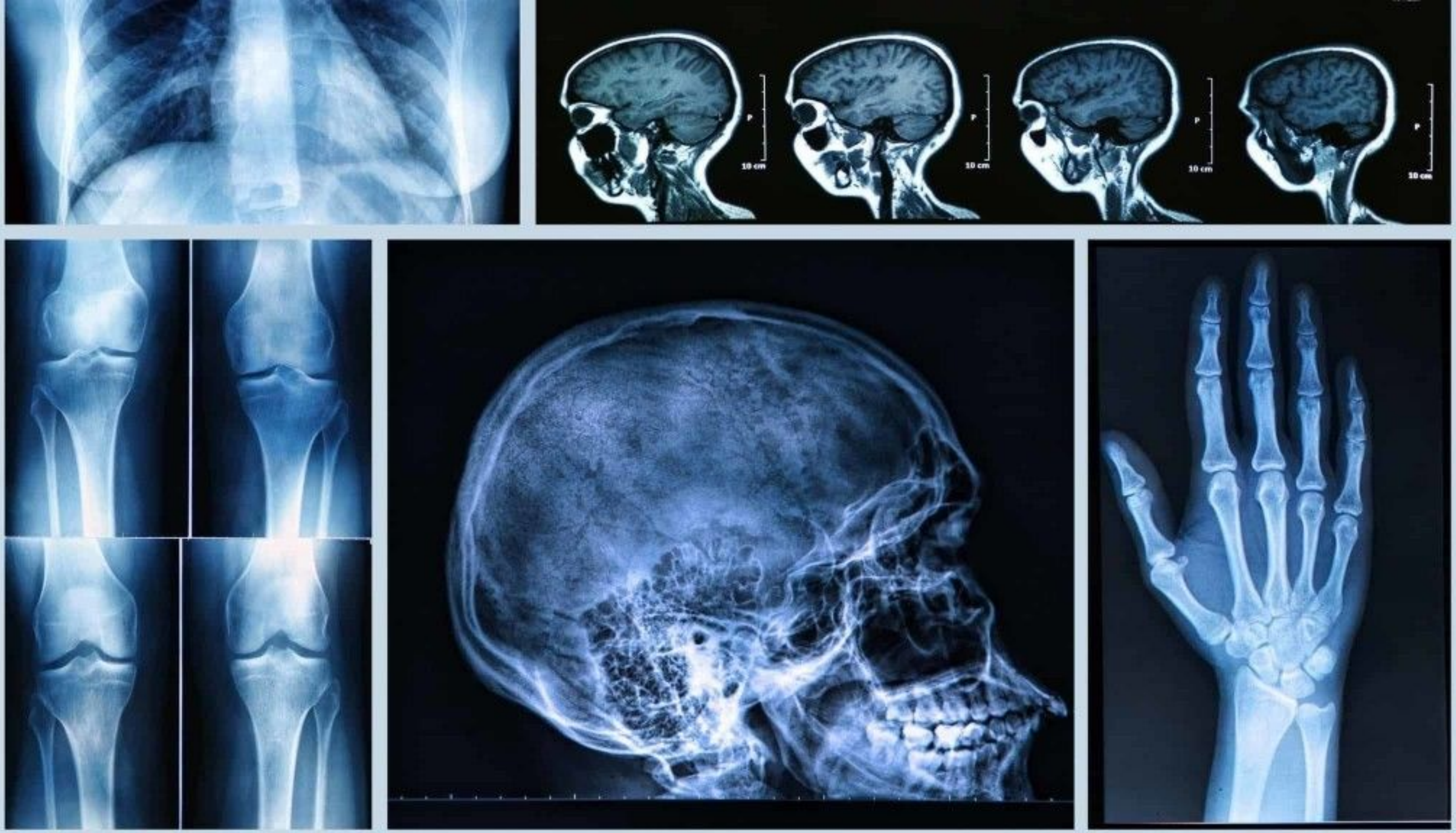
โพลชี้ "บิ๊กตู่" ยังเหมาะนั่งนายกฯเพราะสถาน...
posttoday.com

ประวัติความเป็นมาของลุงตู่นายกคนปัจจุบัน !!
stu40406site.wordpress.com

Have you ever seen this creature before?
Can you guess whether it is land or water animal?



Have you ever seen this creature before?
Can you guess whether it is land or water animal?
You can transfer your knowledge in the past


P
10 cm
P
10 cm
P
10 cm
P
10 cm

Transfer learning

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

Reality:

- *You can **transfer** learned representations from **a related task***
- You can train on a nearby **surrogate objective** for which it is easy to generate labels

Transfer learning: idea

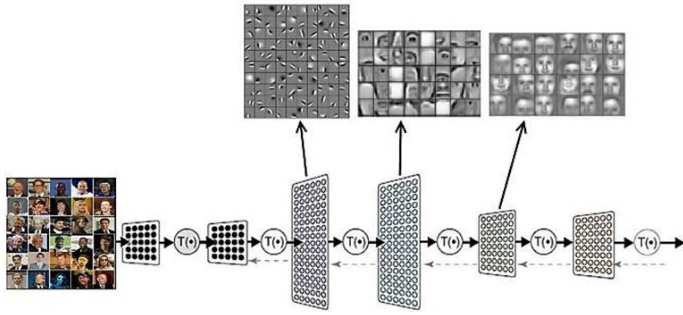
Instead of training a deep network from scratch for your task, you can

- Take a network trained on **a different domain** for a **different source task**
- Adapt it for your domain and your **target task**

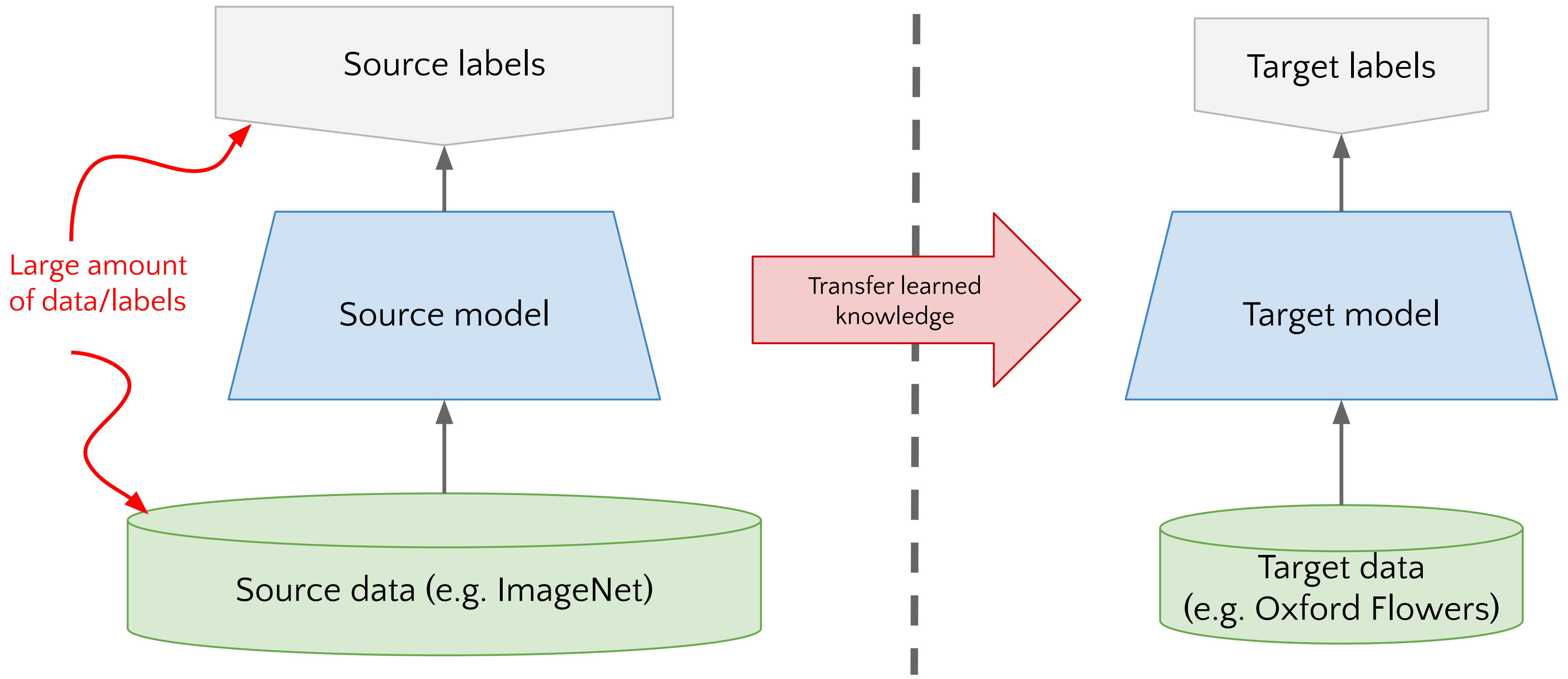
This lecture will talk about how to do this.

Variations:

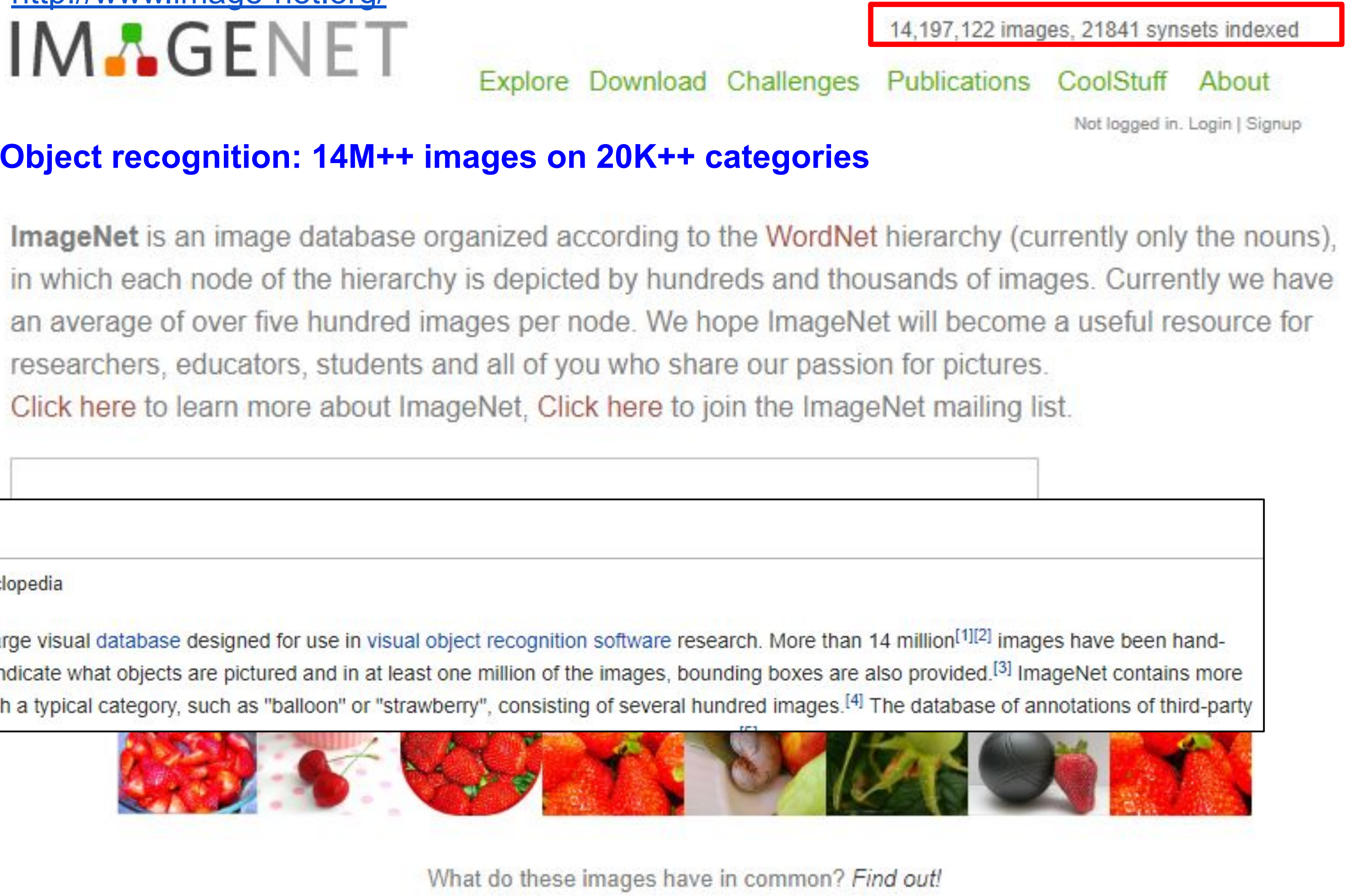
- Different domain (data), same task
- Different domain (data), different task



Transfer learning: idea


Source labels
Large amount of data/labels
Source model
Source data (e.g. ImageNet)
Transfer learned knowledge
Target labels
Target model
Target data (e.g. Oxford Flowers)

http://www.image-net.org/

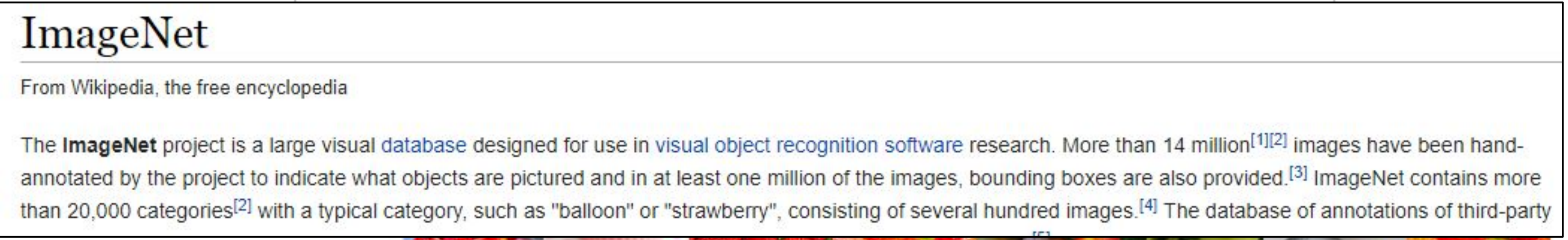


14,197,122 images, 21841 synsets indexed

IMAGENET
Explore Download Challenges Publications CoolStuff About
Not logged in. Login | Signup

Object recognition: 14M++ images on 20K++ categories

ImageNet is an image database organized according to the WordNet hierarchy (currently only the nouns), in which each node of the hierarchy is depicted by hundreds and thousands of images. Currently we have an average of over five hundred images per node. We hope ImageNet will become a useful resource for researchers, educators, students and all of you who share our passion for pictures.
Click here to learn more about ImageNet, Click here to join the ImageNet mailing list.

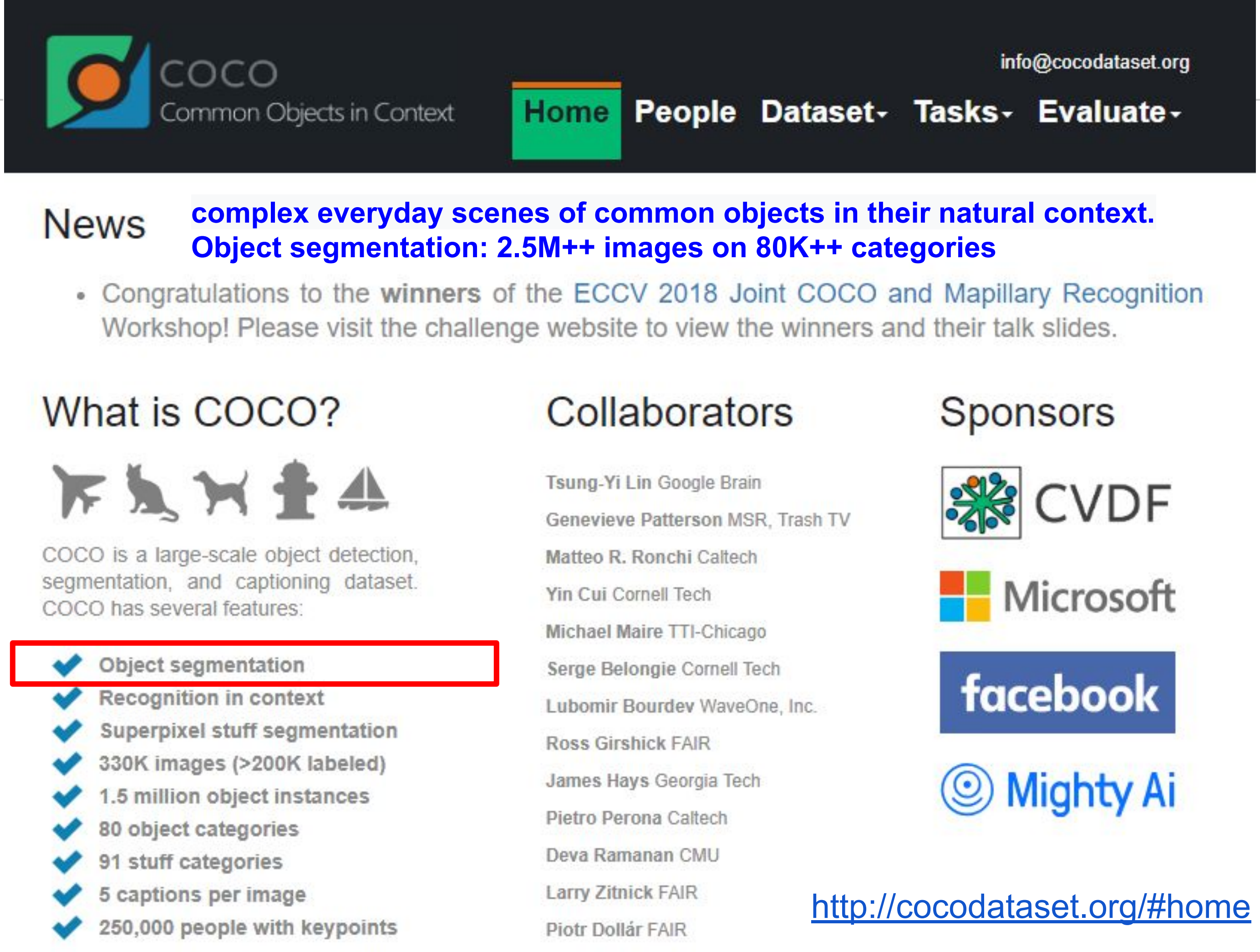


ImageNet

From Wikipedia, the free encyclopedia

The **ImageNet** project is a large visual database designed for use in visual object recognition software research. More than 14 million[1][2] images have been hand-annotated by the project to indicate what objects are pictured and in at least one million of the images, bounding boxes are also provided.[3] ImageNet contains more than 20,000 categories[2] with a typical category, such as "balloon" or "strawberry", consisting of several hundred images.[4] The database of annotations of third-party

What do these images have in common? *Find out!*



News

complex everyday scenes of common objects in their natural context.
Object segmentation: 2.5M++ images on 80K++ categories

- Congratulations to the **winners** of the ECCV 2018 Joint COCO and Mapillary Recognition Workshop! Please visit the challenge website to view the winners and their talk slides.

What is COCO?

COCO is a large-scale object detection, segmentation, and captioning dataset. COCO has several features:

- ✓ Object segmentation
- ✓ Recognition in context
- ✓ Superpixel stuff segmentation
- ✓ 330K images (>200K labeled)
- ✓ 1.5 million object instances
- ✓ 80 object categories
- ✓ 91 stuff categories
- ✓ 5 captions per image
- ✓ 250,000 people with keypoints

Collaborators

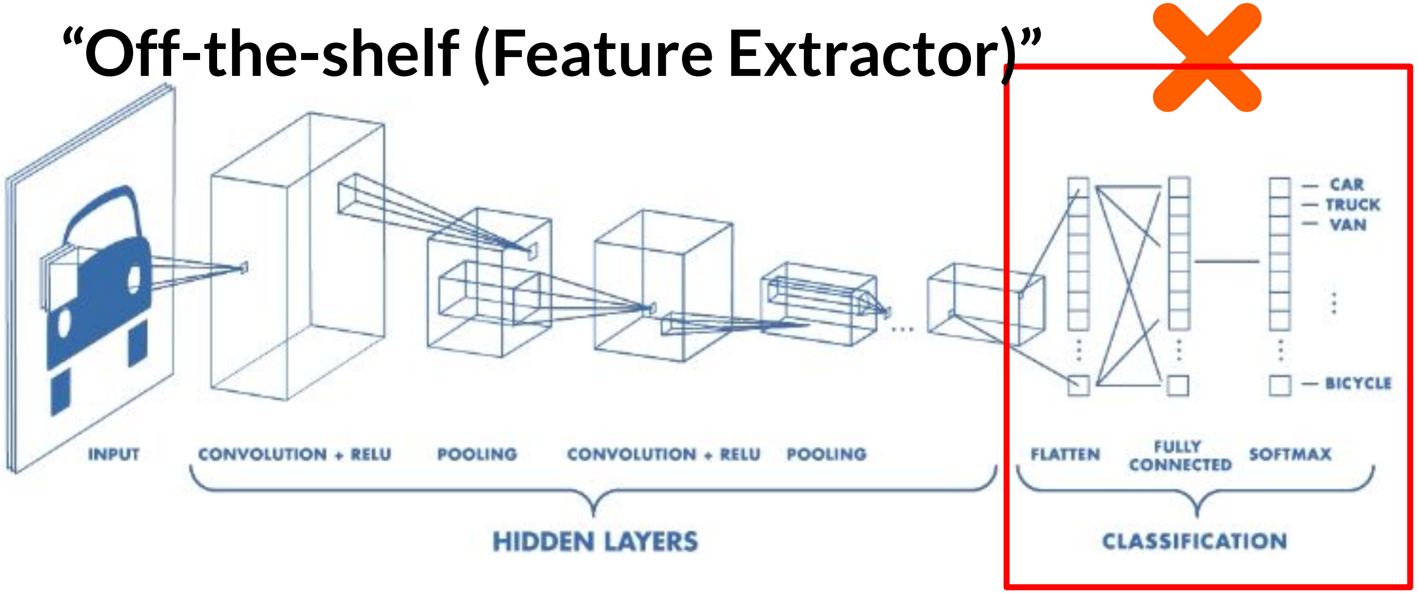
Tsung-Yi Lin Google Brain
Genevieve Patterson MSR, Trash TV
Matteo R. Ronchi Caltech
Yin Cui Cornell Tech
Michael Maire TTI-Chicago
Serge Belongie Cornell Tech
Lubomir Bourdev WaveOne, Inc.
Ross Girshick FAIR
James Hays Georgia Tech
Pietro Perona Caltech
Deva Ramanan CMU
Larry Zitnick FAIR
Piotr Dollár FAIR

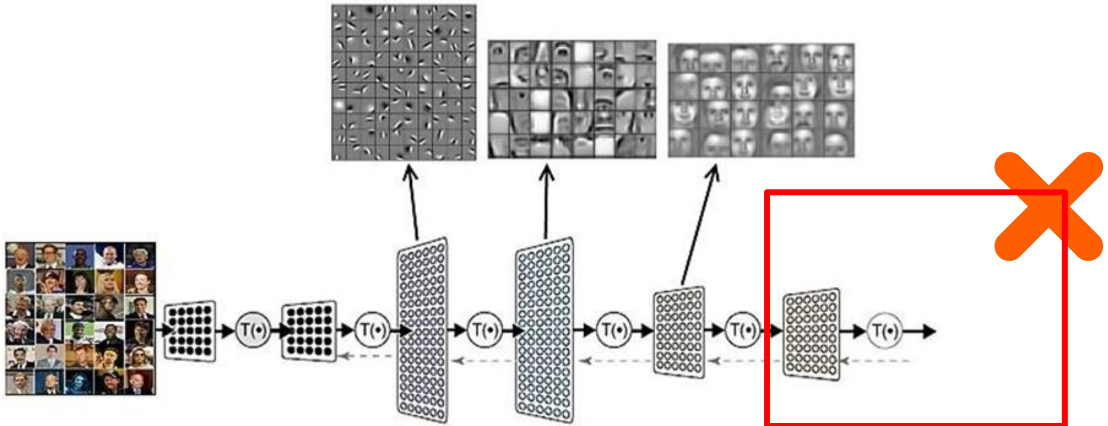
Sponsors

CVDF
Microsoft
facebook
Mighty Ai

http://cocodataset.org/#home

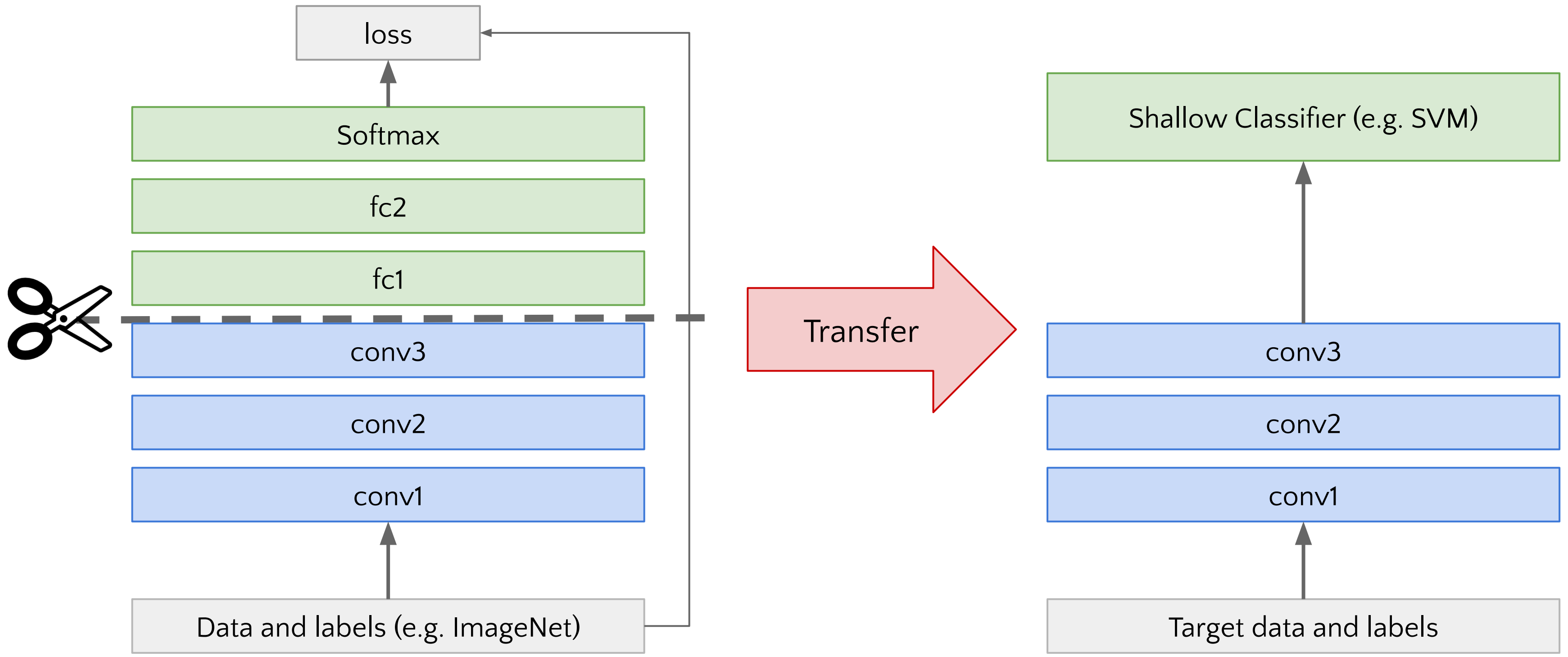
"Off-the-shelf (Feature Extractor)"



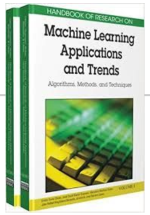


“Off-the-shelf (Feature Extractor)” (cont.)

Idea: use outputs of one or more layers of a network trained on a different task as generic feature detectors. Train a new shallow model on these features.

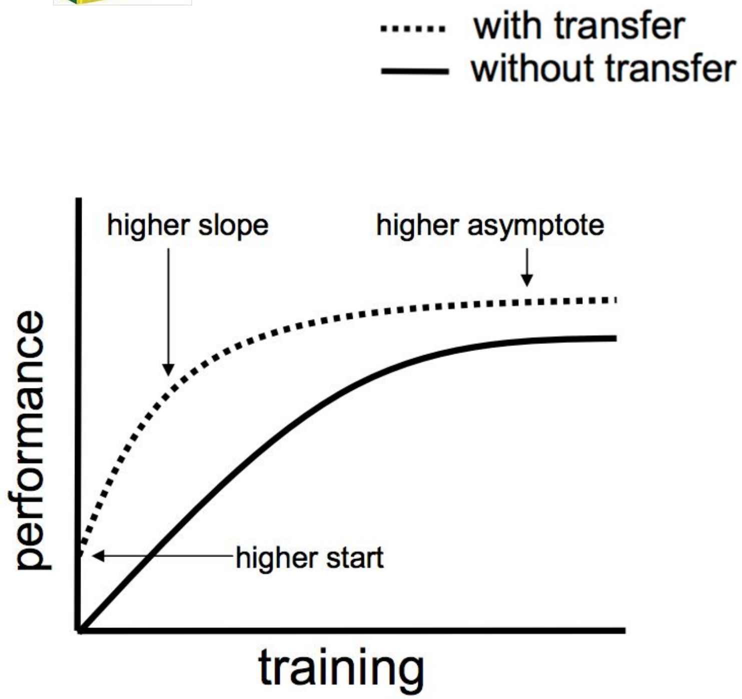


Transfer learning: 3 benefits



Lisa Torrey and Jude Shavlik in their chapter on transfer learning describe **three possible benefits** to look for when using transfer learning:

1. **Higher start.** The initial skill (before refining the model) on the source model is higher than it otherwise would be.
2. **Higher slope.** The rate of improvement of skill during training of the source model is steeper than it otherwise would be.
3. **Higher asymptote.** The converged skill of the trained model is better than it otherwise would be.



Can we do better than off the shelf features ?

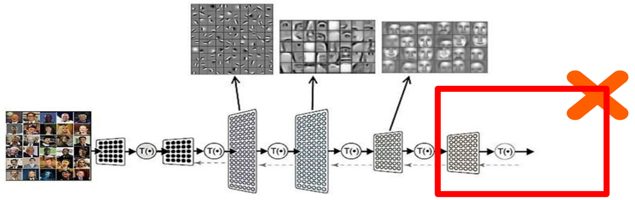
Fine-tuning: supervised task adaptation

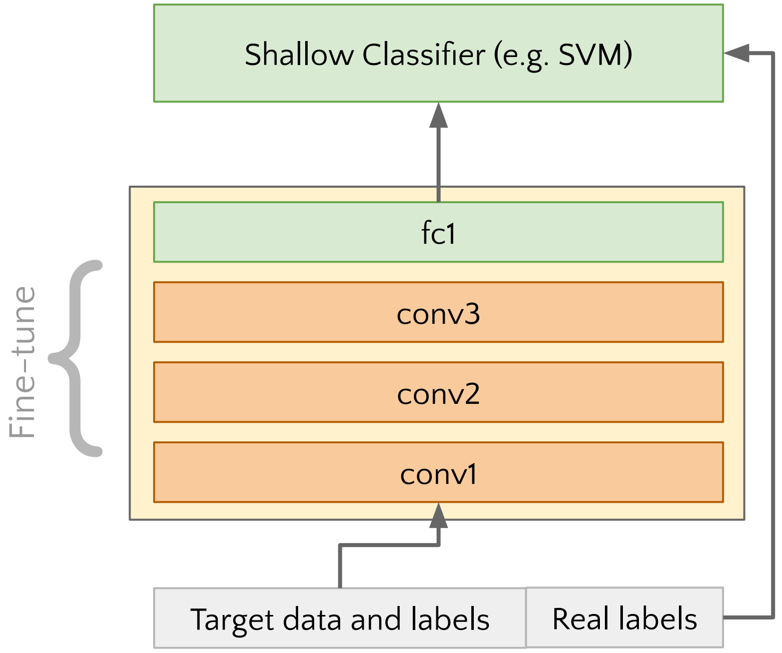
Train deep net on **nearby task** for which it is **easy to get labels** using standard backprop.

- E.g. ImageNet classification
- Pseudo classes from augmented data
- Slow feature learning, ego-motion

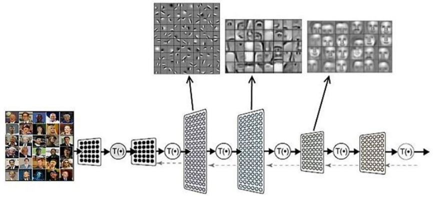
Cut off top layer(s) of network and replace with supervised objective for target domain.

Fine-tune network using backprop with labels for target domain until validation loss starts to increase.





How transferable are features



Lower layers: more general features.

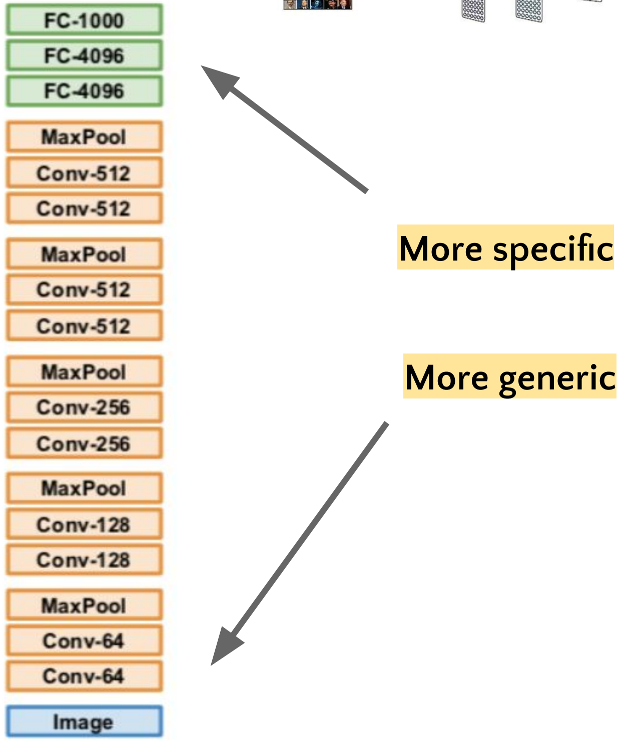
Transfer very well to other tasks.

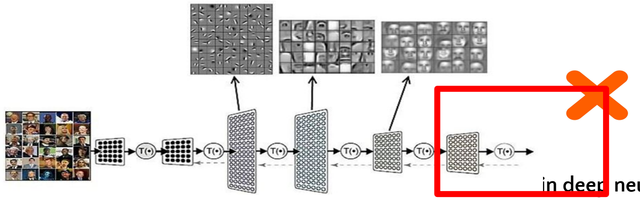
Higher layers: more task specific.

Fine-tuning improves generalization when sufficient examples are available.

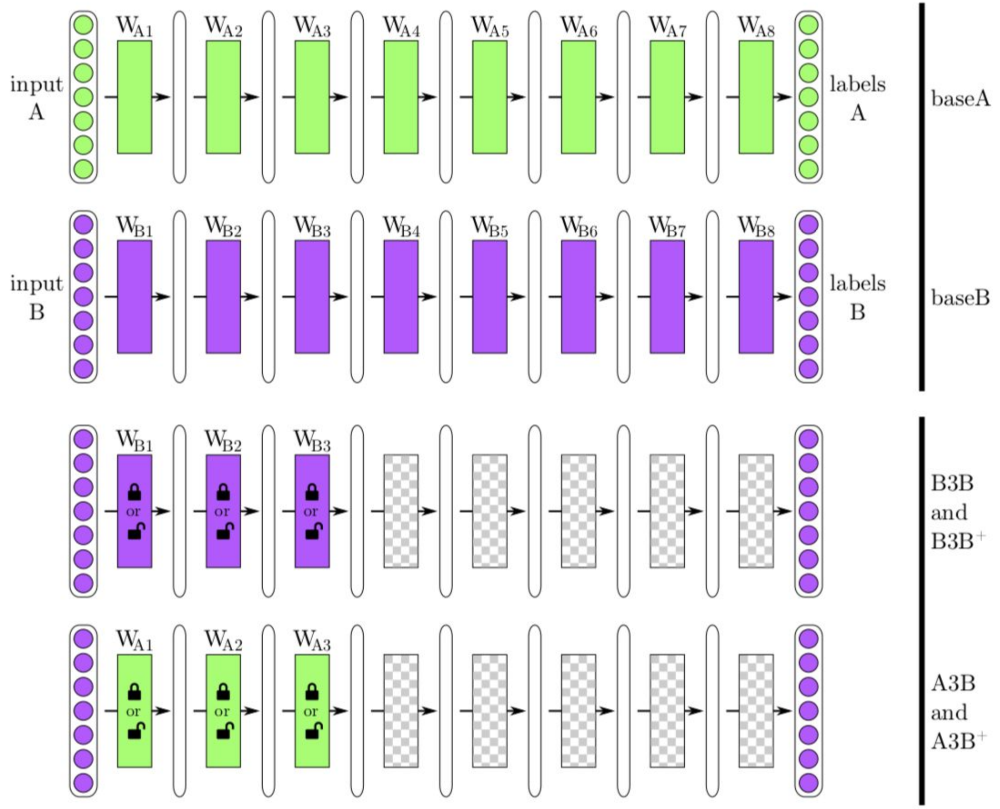
Transfer learning **and** fine tuning often lead to better performance than training from scratch on the target dataset.

Even features transferred from distant tasks are often better than random initial weights!





How transferable are features



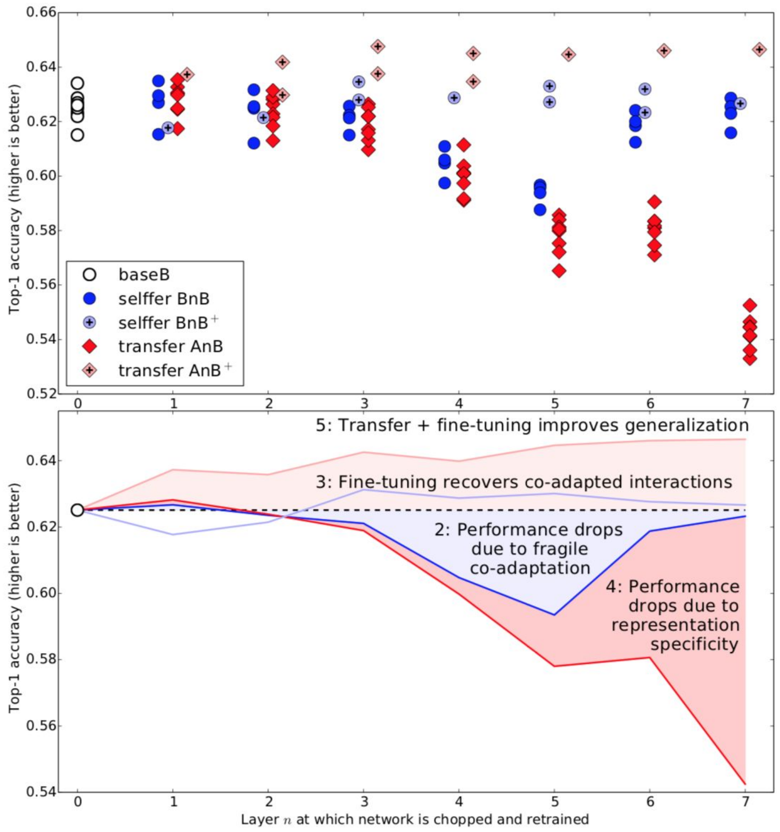
- A ***selffer* network B3B**: the first 3 layers are copied from baseB and frozen. The five higher layers (4–8) are initialized randomly and trained on dataset B. This network is a control for the next transfer network.

- A ***transfer* network A3B**: the first 3 layers are copied from base A and frozen. The five higher layers (4–8) are initialized randomly and trained toward dataset B. Intuitively, here we copy the first 3 layers from a network trained on dataset A and then learn higher layer features on top of them to classify a new target dataset B. If A3B performs as well as baseB, there is evidence that the third-layer features are general, at least with respect to B. If performance suffers, there is evidence that the third-layer features are specific to A.

- **B3B$^+$ and A3B$^+$** are just like B3B and A3B, but where all transferred layers are ***fine-tuned***.

Yosinki et al., **How transferable are features in deep neural networks,** NIPS 2014 https://arxiv.org/abs/1411.1792

How transferable are features



Co-adaptation issue is a problem that layers interact with each other.

2

How to choose backbone model

How to choose backbone (pretrained weights) model; *Trade off accuracy & time*

- Choose model to transfer based on for target application (e.g. smaller network in case you want to run on mobile, and bigger network is required if running on server)
- The graph below shows the backbone models performance measured on the COCO detection task.

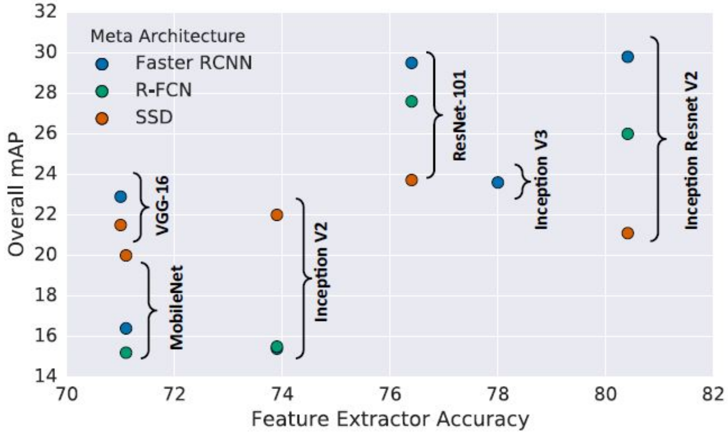
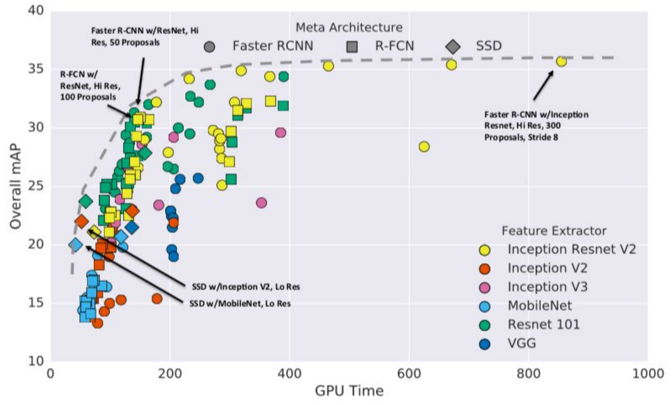


Figure 3: Accuracy of detector (mAP on COCO) vs accuracy of feature extractor (as measured by top-1 accuracy on ImageNet-CLS). To avoid crowding the plot, we show only the low resolution models.



Inception ResNet V2:
Complex model
# higher accuracy
# more time

MobileNet: Small model
# lower accuracy
# less time

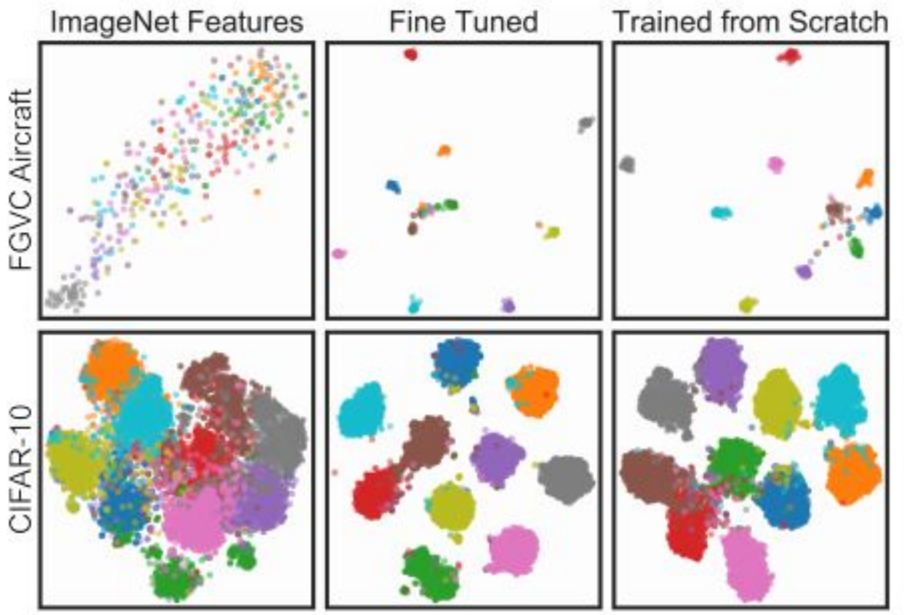
Huang et al., **Speed/accuracy trade-offs for modern convolutional object detectors**, CVPR 2017.

Do Better ImageNet Models Transfer Better?

Similar domain can be transferred easier

Aircraft images
Different from ImageNet

Natural images
Similar to ImageNet



Do Better ImageNet Models Transfer Better? (cont.)

#epochs

>> From ImageNet
Red = With pretrained
Blue = Without pretrained (scratch)

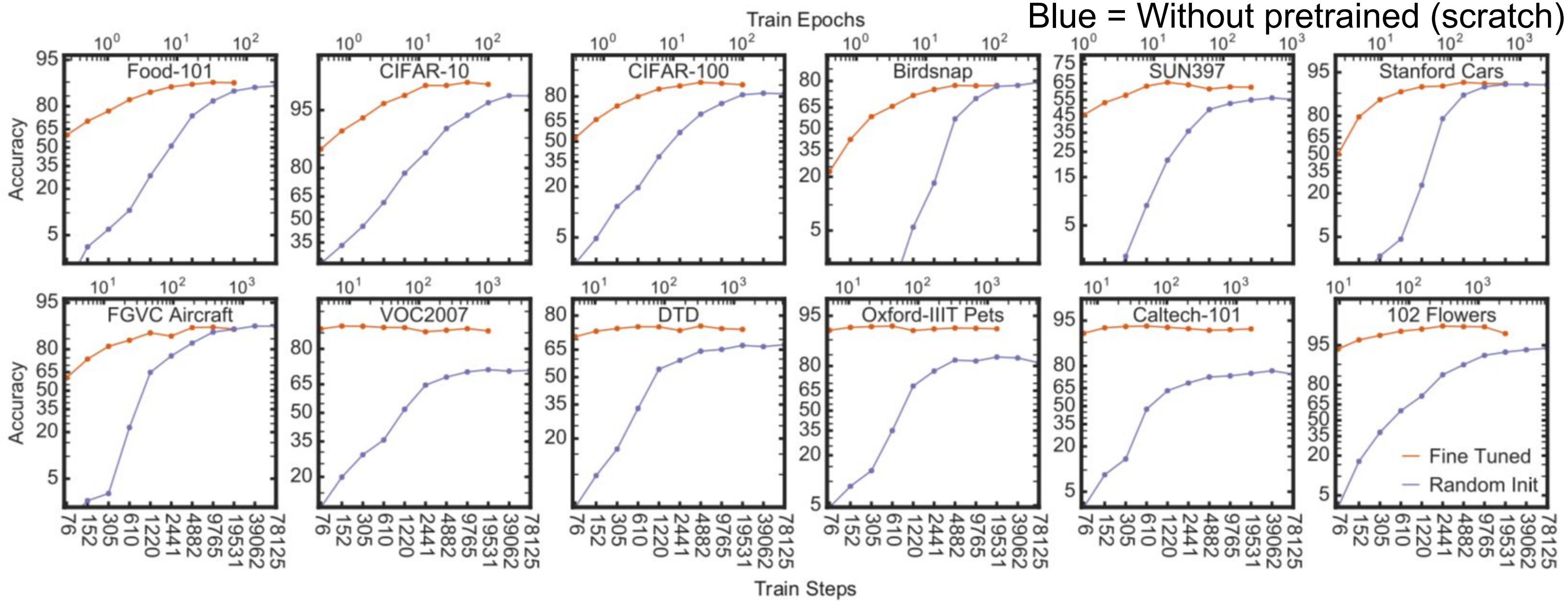


Figure 8. Networks pretrained on ImageNet converge faster, even when final accuracy is the same as training from random initialization. Each point represents an independent Inception v4 model trained with optimized hyperparameters. For fine-tuning, we initialize with the public TensorFlow Inception v4 checkpoint. Axes are logit-scaled.

Do Better ImageNet Models Transfer Better? (cont.)

#examples per class

>> From ImageNet
Red = With pretrained
Violet = Without pretrained (scratch)

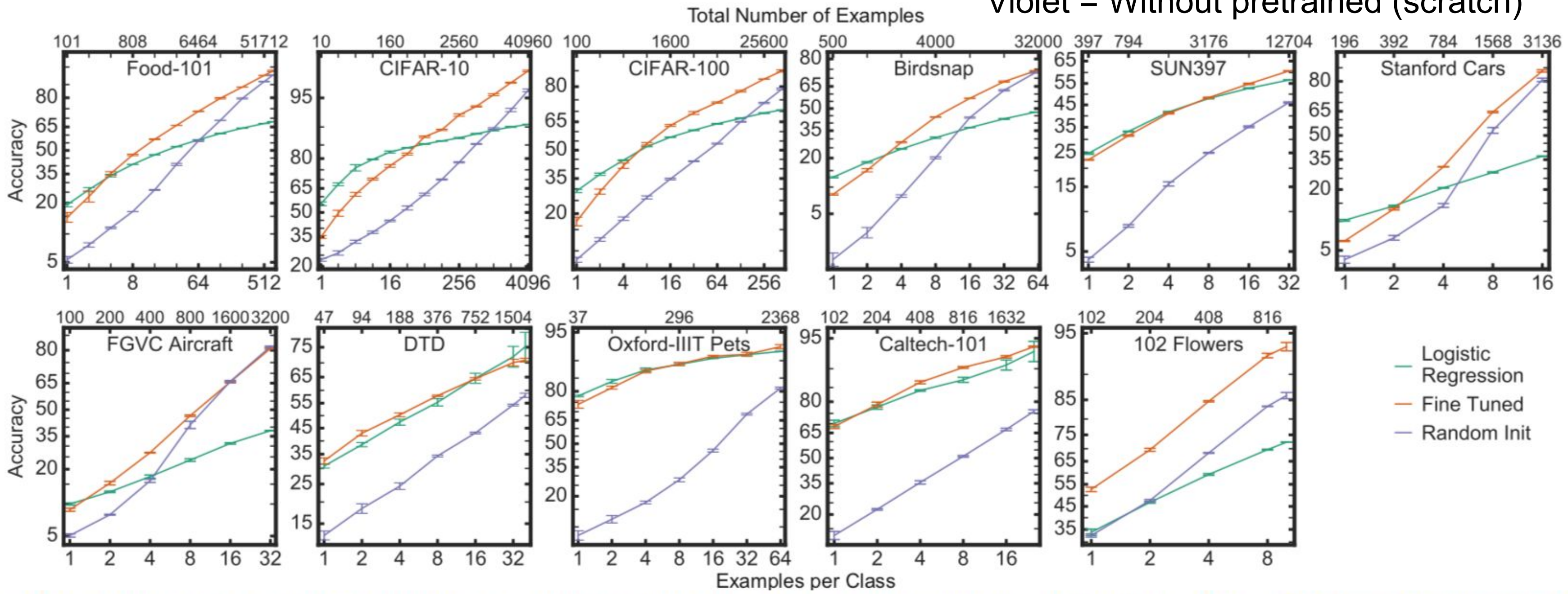


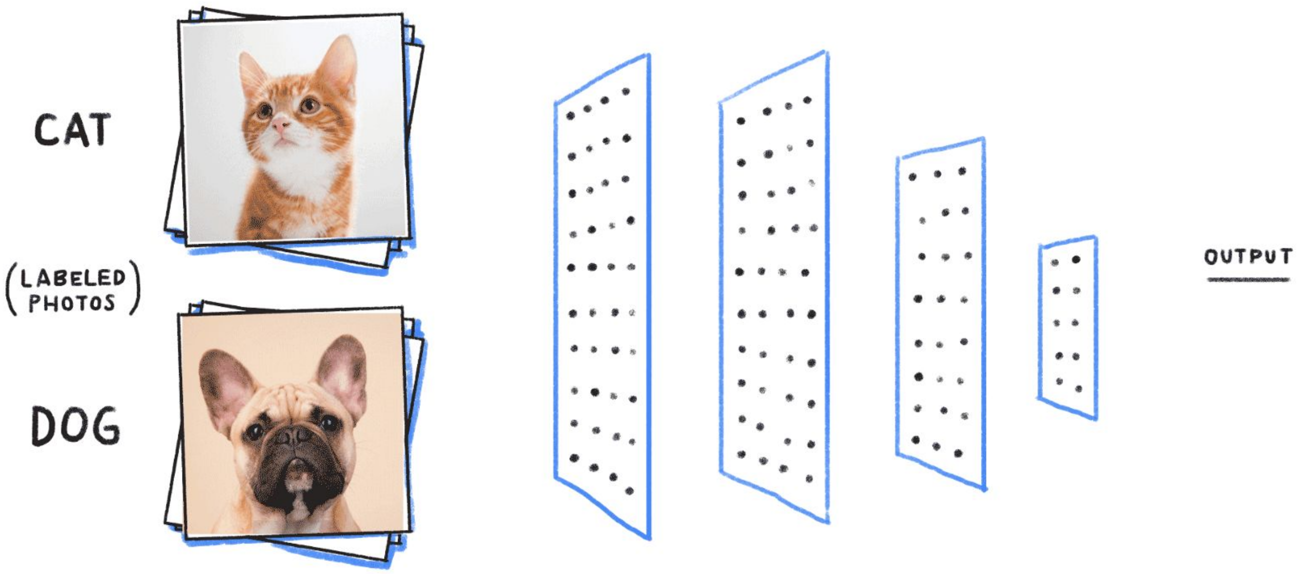
Figure 9. Pretraining on ImageNet improves performance on fine-grained tasks with small amounts of data, but the gap narrows quickly as dataset size increases. Performance of transfer learning with the public Inception v4 model at different dataset sizes. Error bars reflect standard error over 3 subsets. Note that the maximum dataset size shown is not the full dataset. Best viewed in color.

3

Same task different domain

Image classification

We will introduce the Image Classification problem, which is the task of assigning an input image one label from a fixed set of categories.



source : https://becominghuman.ai/building-an-image-classifier-using-deep-learning-in-python-totally-from-a-beginners-perspective-be8dbaf22dd8 source : http://www.robots.ox.ac.uk/~vgg/data/flowers/17/index.html

Example: Oxford 17 Flower Datasets

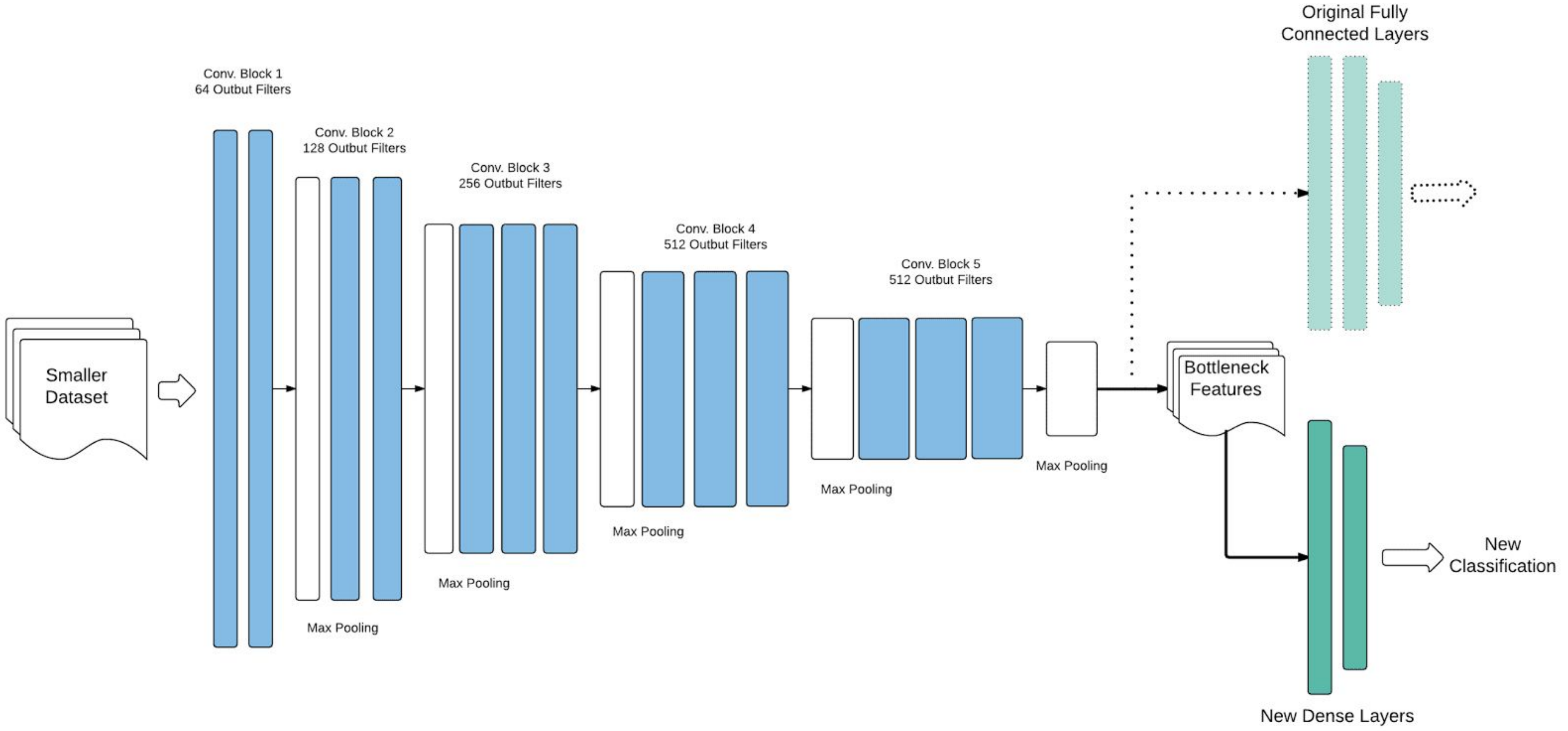
Overview

- 17 category flower dataset with 80 images for each class.
- The flowers chosen are some common flowers in the UK.
- The images have large scale, pose and light variations and there are also classes with large variations of images within the class and close similarity to other classes.
- A subset of the images have been groundtruth labelled for segmentation.

Class Examples

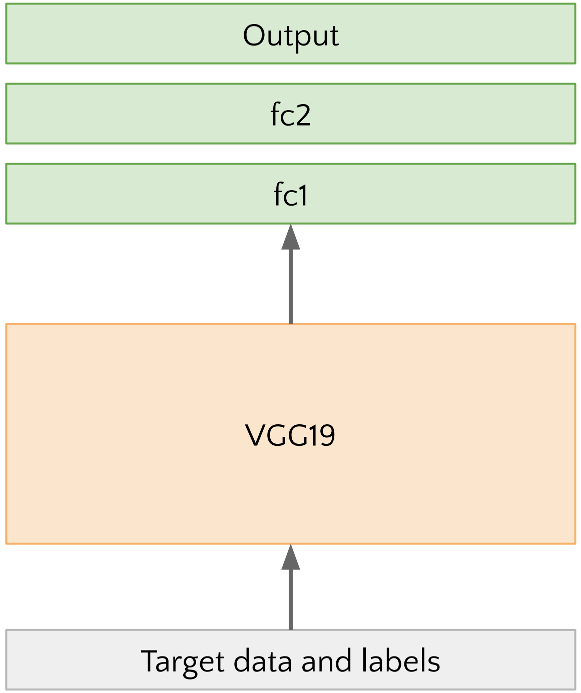
Karen Simonyan, Andrew Zisserman,
Very Deep Convolutional Networks for Large-Scale Image Recognition,
arXiv 2014 https://arxiv.org/abs/1409.1556

VGG19



Model 1: VGG19 (random initialized weights) + 2 Dense layers + Output layer

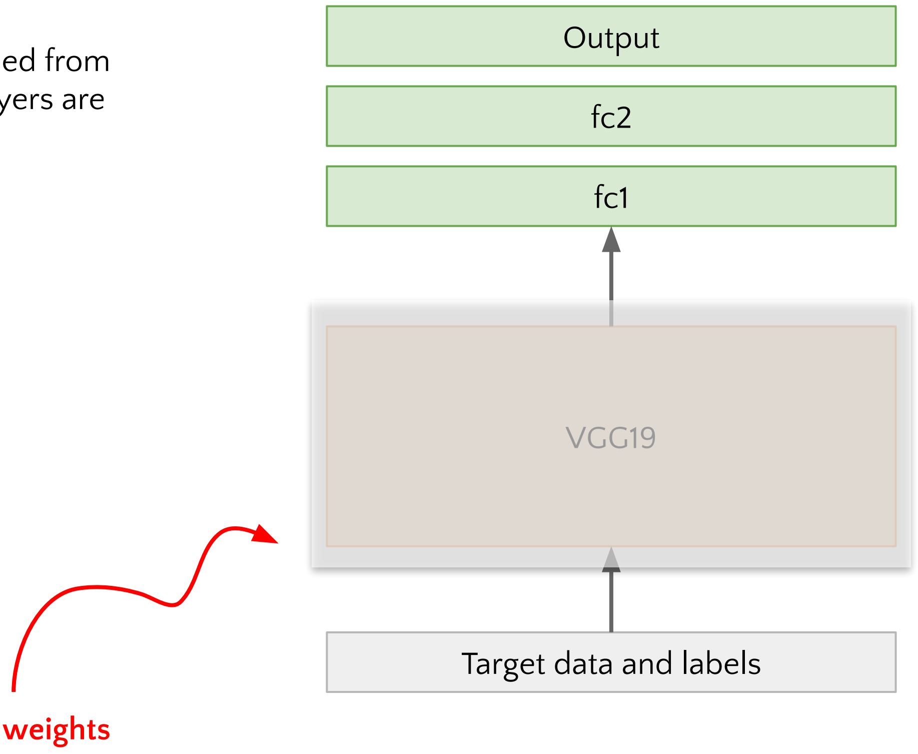
- The model is trained from scratch.
- **no pretrained** weights concept.



Model 2: VGG (pre-trained weights) + 2 Dense layers + Output layer

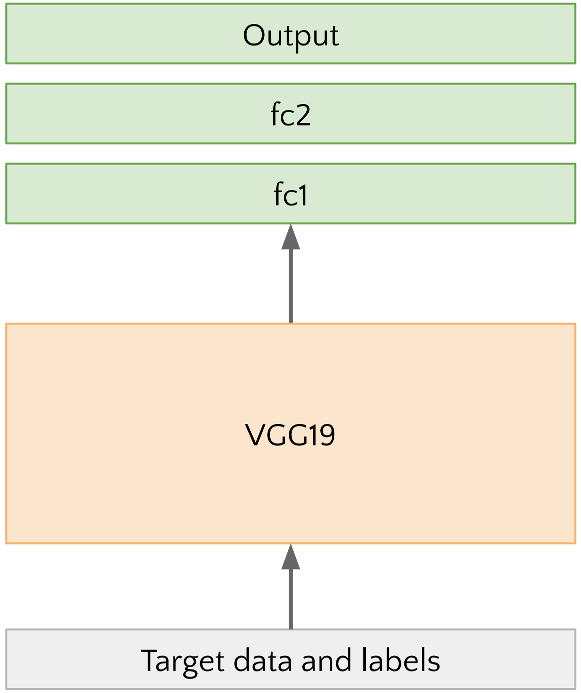
- Use pretrained weights VGG19 trained from Imagenet, new 2 fully connected layers are initialized randomly.

- Feature extractor concept.



Model 3: VGG19 (pre-trained weights) + 2 Dense layers + Output layer

- The whole model, VGG19 and new fully connected layers, is trained to adapt with target task.
- Unfreeze all layers; **fine-tune concept.**



Model 4: VGG (pre-trained weights) + 2 Dense layers + Output layer

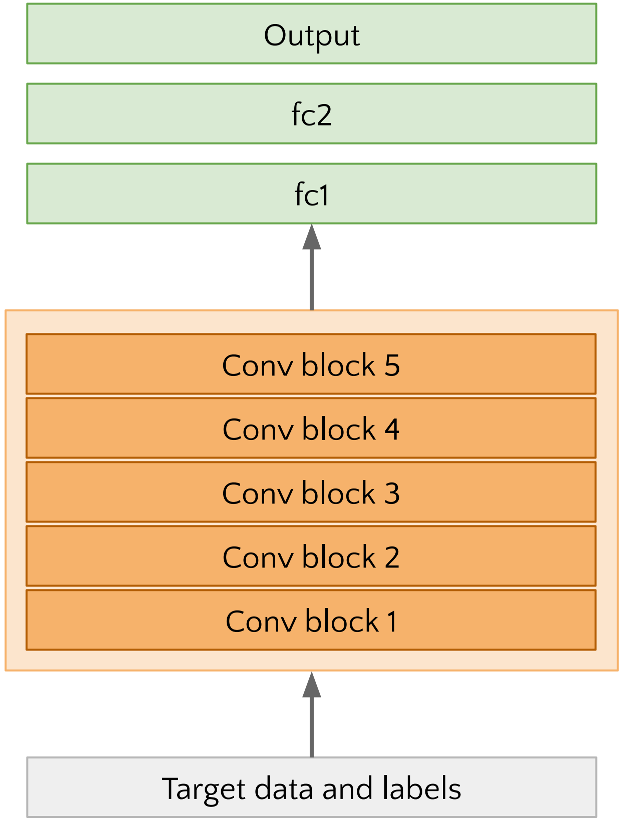
- Use “**chain-thaw**” fine-tune technique that sequentially unfreezes and fine-tune a single layer at a time to increases accuracy on the target task at the expense of extra computational power needed for the fine-tuning and would able to adjust the individual patterns across the network with a reduced risk of overfitting.

- Chain-thaw contains 3 sequential phases

  1. fine-tunes any new layers.

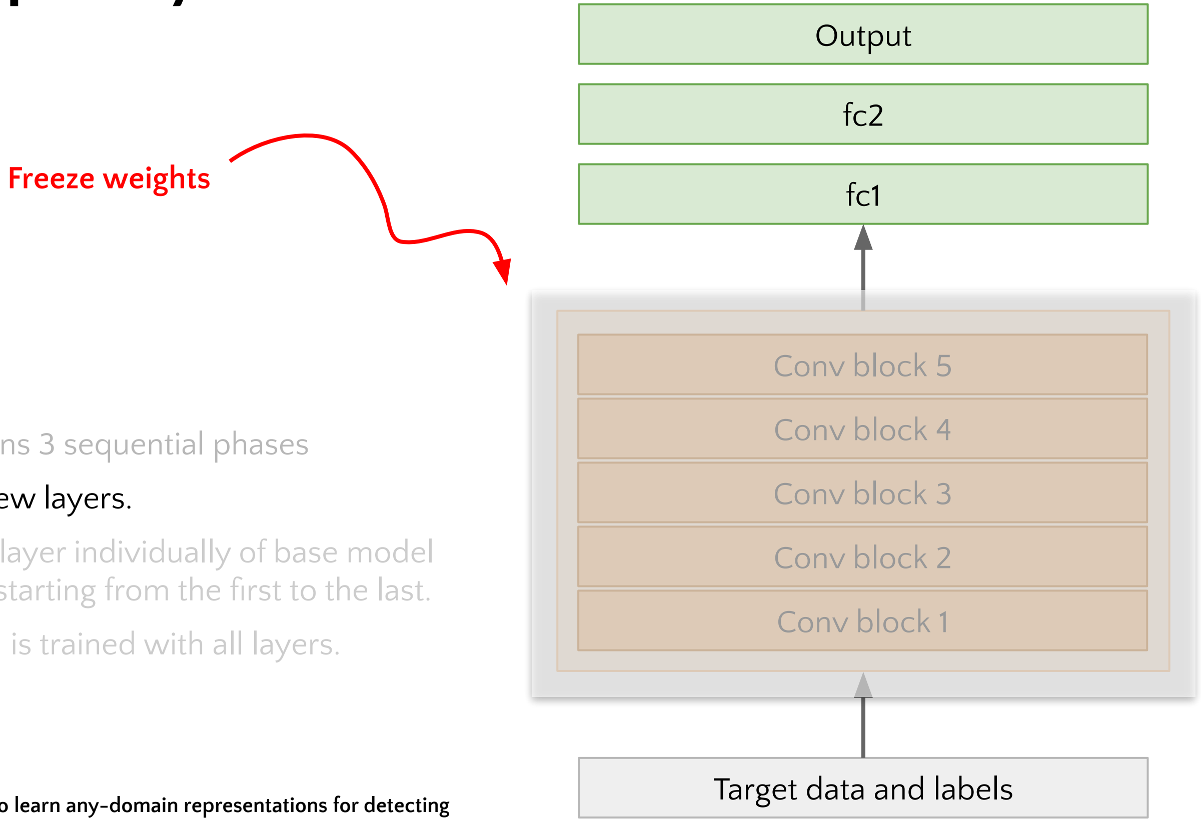
  2. fine-tunes each layer individually of base model (VGG in this case) starting from the first to the last.

  3. the entire model is trained with all layers.



Felbo et al., **Using millions of emoji occurrences to learn any-domain representations for detecting sentiment, emotion and sarcasm,** arxiv 2017, https://arxiv.org/pdf/1708.00524.pdf

Model 4: VGG (pre-trained weights) + 2 Dense layers + Output layer



- Chain-thaw contains 3 sequential phases

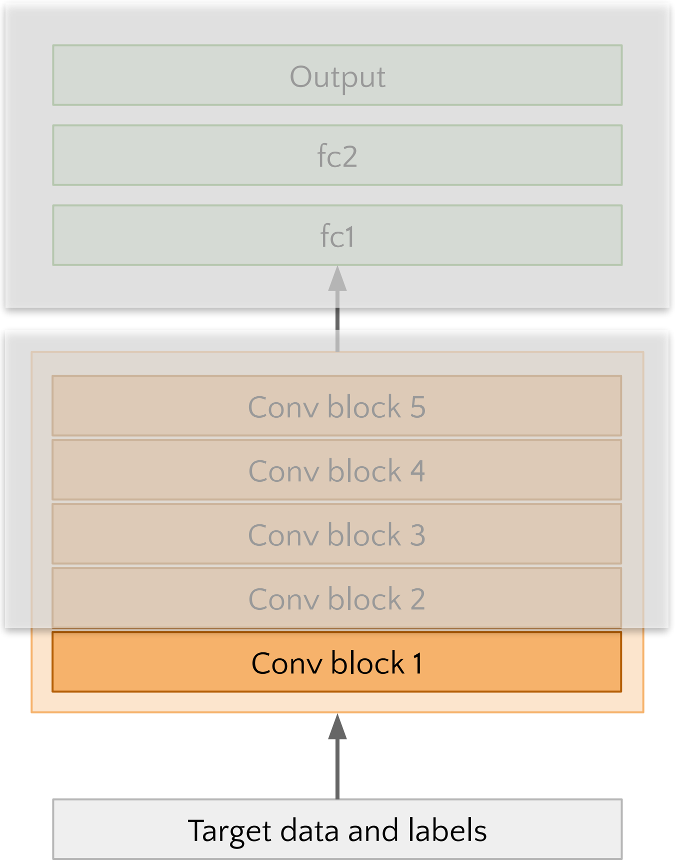
  1. fine-tunes any new layers.

  2. fine-tunes each layer individually of base model (VGG in this case) starting from the first to the last.

  3. the entire model is trained with all layers.

Felbo et al., **Using millions of emoji occurrences to learn any-domain representations for detecting sentiment, emotion and sarcasm,** arxiv 2017, https://arxiv.org/pdf/1708.00524.pdf

Model 4: VGG (pre-trained weights) + 2 Dense layers + Output layer



- Chain-thaw contains 3 sequential phases

  1. fine-tunes any new layers.

  2. fine-tunes each layer individually of base model (VGG in this case) starting from the first to the last.

  3. the entire model is trained with all layers.

Felbo et al., **Using millions of emoji occurrences to learn any-domain representations for detecting sentiment, emotion and sarcasm,** arxiv 2017, https://arxiv.org/pdf/1708.00524.pdf

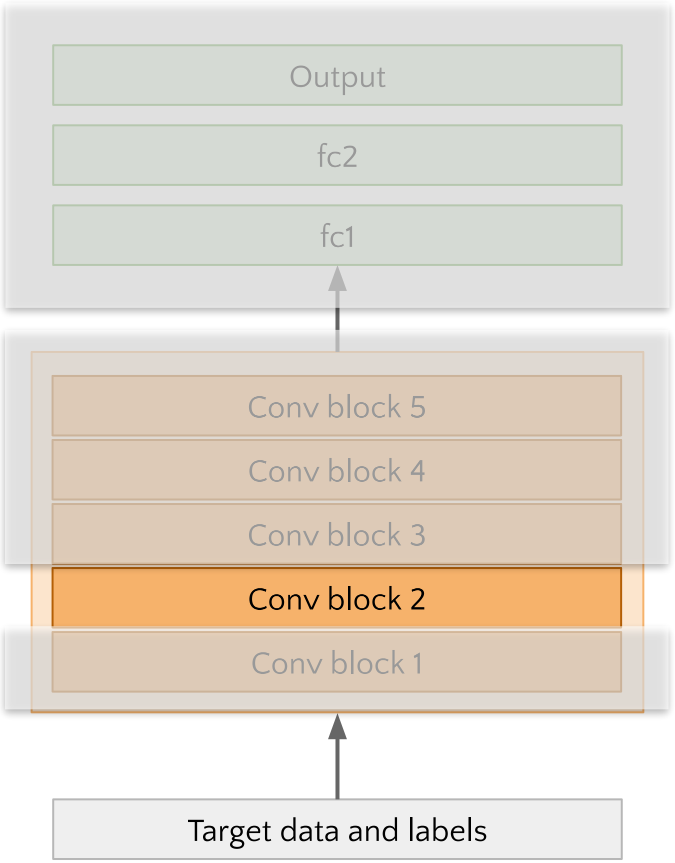
Model 4: VGG (pre-trained weights) + 2 Dense layers + Output layer

- Chain-thaw contains 3 sequential phases

  1. fine-tunes any new layers.

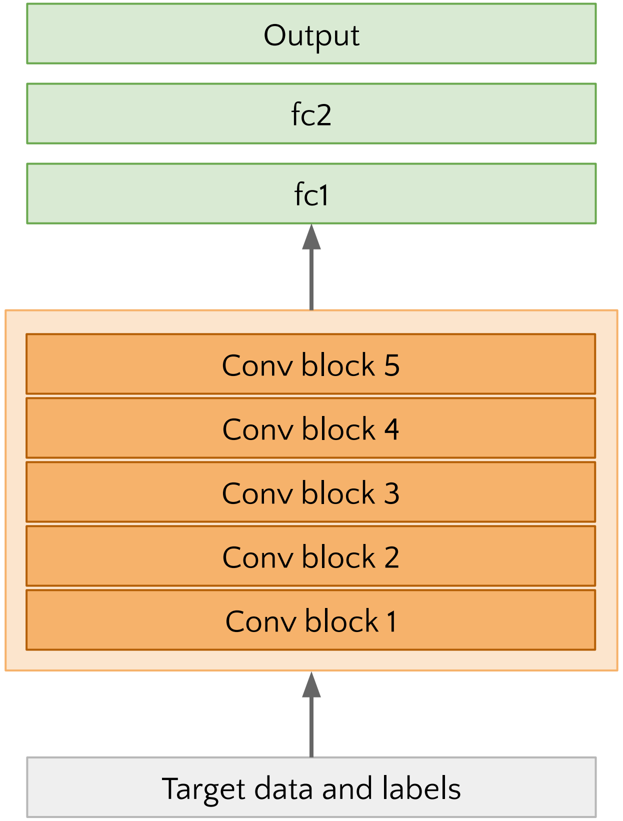
  2. fine-tunes each layer individually of base model (VGG in this case) starting from the first to the last.

  3. the entire model is trained with all layers.



Felbo et al., **Using millions of emoji occurrences to learn any-domain representations for detecting sentiment, emotion and sarcasm,** arxiv 2017, https://arxiv.org/pdf/1708.00524.pdf

Model 4: VGG (pre-trained weights) + 2 Dense layers + Output layer



- Chain-thaw contains 3 sequential phases

  1. fine-tunes any new layers.

  2. fine-tunes each layer individually of base model (VGG in this case) starting from the first to the last.

  3. the entire model is trained with all layers.

Felbo et al., **Using millions of emoji occurrences to learn any-domain representations for detecting sentiment, emotion and sarcasm,** arxiv 2017, https://arxiv.org/pdf/1708.00524.pdf

Conclusion

- Possible to train very large models on small data by using transfer learning and domain adaptation.
- Off the shelf features, as fixed features, work very well in various domains and tasks for smaller samples, but should be adapted to increase performance (fine tune).
- Learned features are hierarchical representations, different layers learn at different level and transfer differently, Lower layers of network contain very generic features, higher layers more task specific features. Tasks that are very different are harder to transfer and required more layer up top.
- **Supervised domain adaptation via fine tuning almost always improves performance.**
- Best backbone for transfer overall is ResNet family.

References

K. McGuinness, Transfer Learning (D2L4 Insight@DCU Machine Learning Workshop 2017)