DS-UA 301
Advanced Topics in Data Science
Advanced Techniques in ML and Deep Learning

LECTURE 7

Parijat Dube

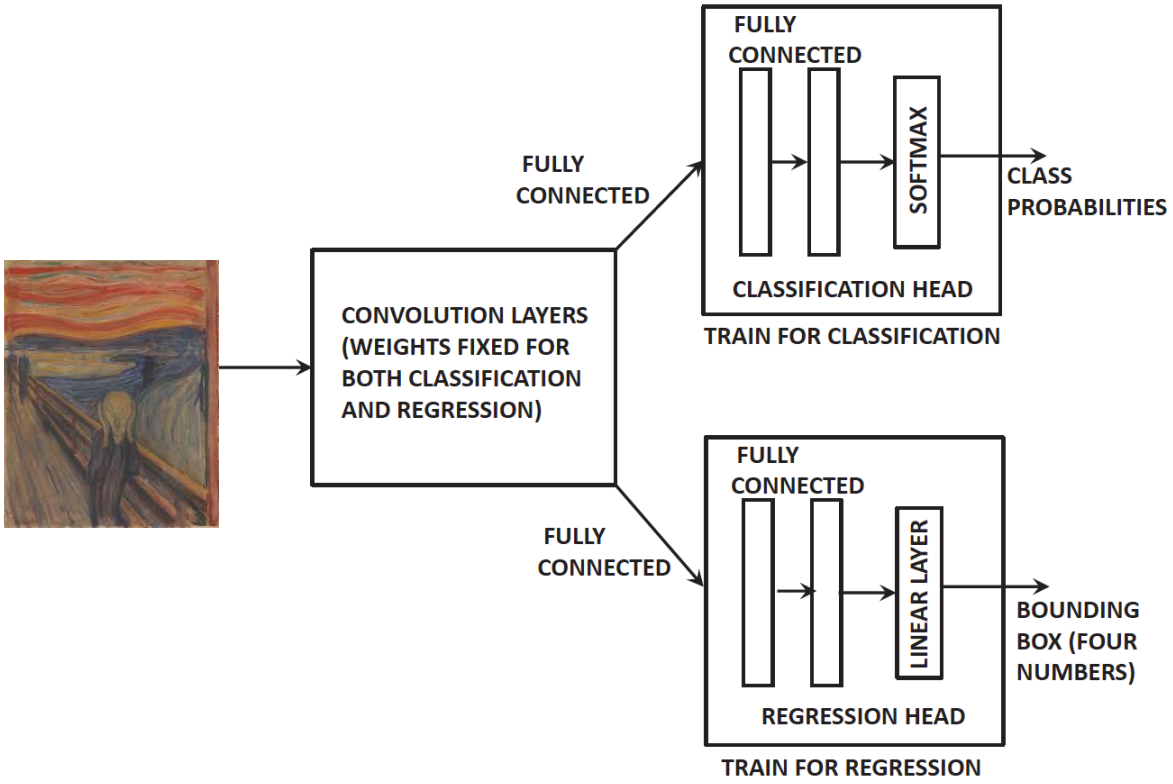
CNN Applications

- Image classification
- Object localization
- Object detection

Object Localization

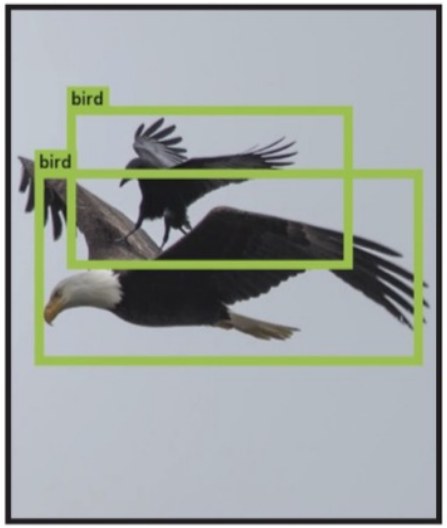
- Given a fixed set of objects in an image, identify rectangular regions in the image in which the object occurs
- Four numbers to identify a bounding box: coordinates of the top-left corner and two dimensions (length and width)
- Often integrated with classification of objects

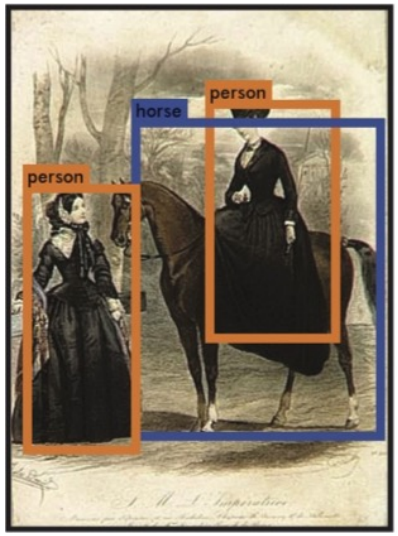
Object Localization

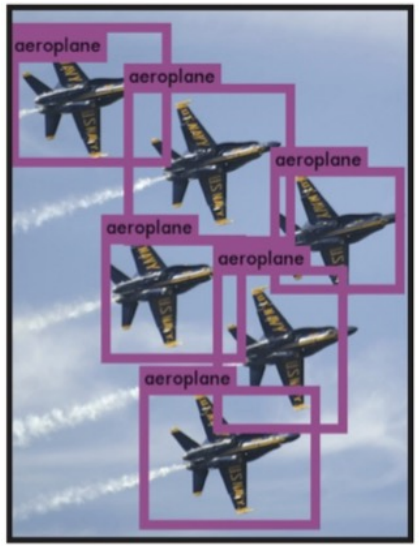


Object Detection

- Variable number of objects of different classes in an image
- Identify all the objects and their class

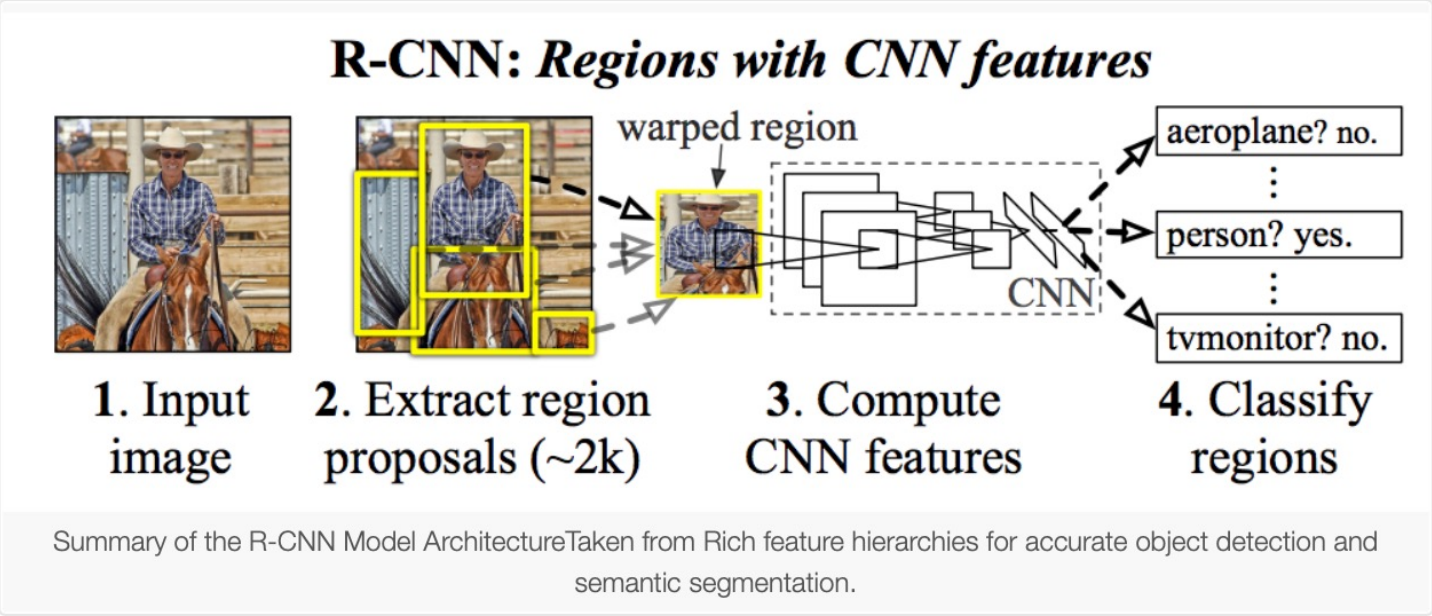






Cannot use object localization architecture as the number of classification and regression heads needed is not known

R-CNN (Region Based Convolutional Neural Network)



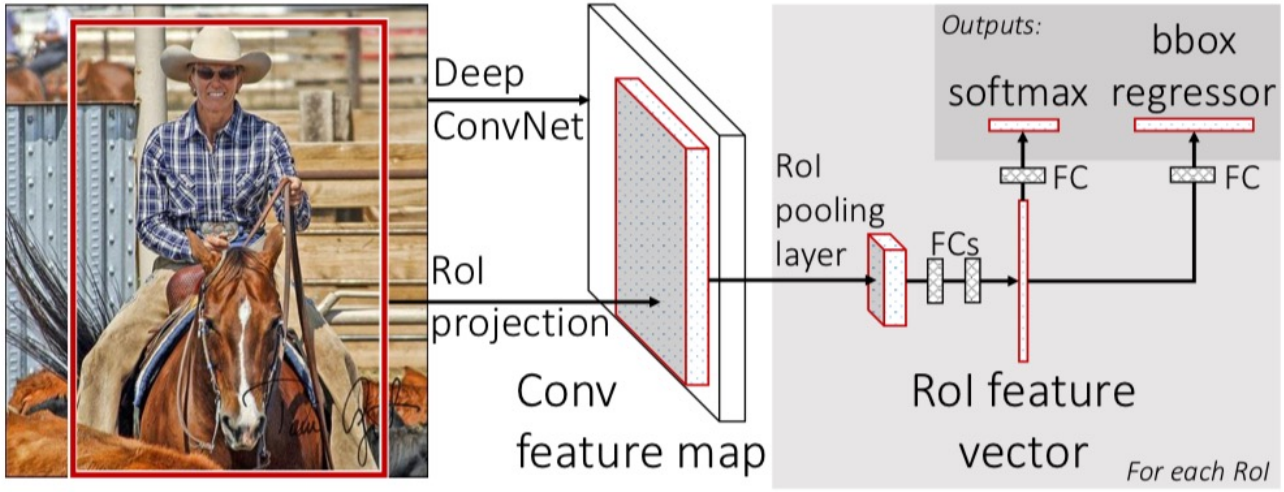
Summary of the R-CNN Model ArchitectureTaken from Rich feature hierarchies for accurate object detection and semantic segmentation.

Three main modules:

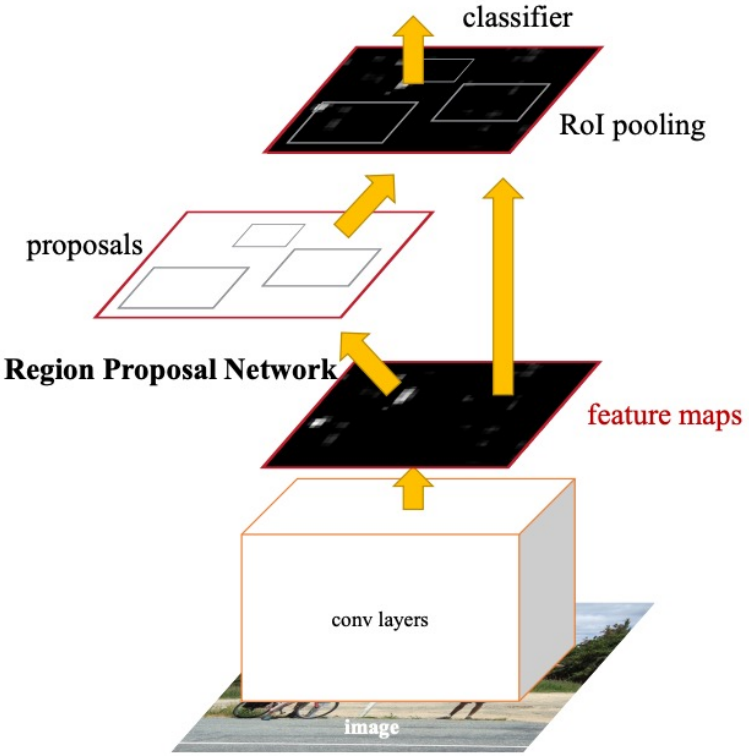
1. **Region Proposal**. Generate and extract category independent region proposals, e.g. candidate bounding boxes.
2. **Feature Extractor**. Extract feature from each candidate region, e.g. using a deep convolutional neural network.
3. **Classifier**. Classify features as one of the known class, e.g. linear SVM classifier model.

Fast R-CNN

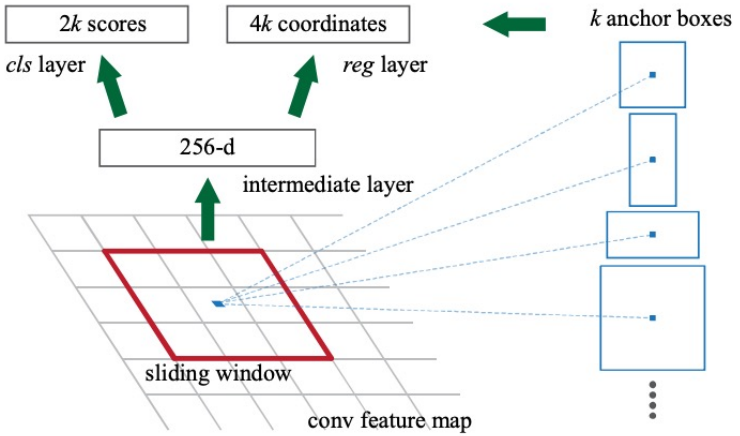
- R-CNN is slow
  - CNN-based feature extraction pass on each of the candidate regions, no sharing of computation
  - ~2,000 proposed regions per image at test-time



Faster R-CNN

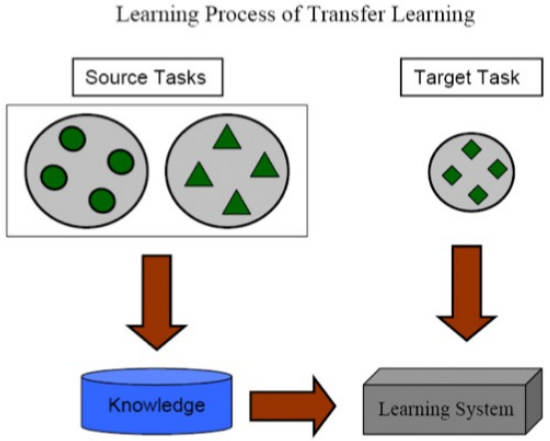


Region Proposal Network



Transfer Learning

- Building good machine learning models require lot of training data
  - To capture robust representation of unknown input distribution
- Small training jobs are common and labeled data is scarce in many domains
  - In commercial VR service (Bhatta et al. 2019), average number of images submitted is 250 and average number of classes are 5; ~50 images per class
- Can we leverage knowledge learnt from related tasks for target task ?
- Transfer learning is a class of techniques to reuse knowledge gathered from “source” tasks (with sufficiently rich set of labeled data) for a “target task (with few labeled data)



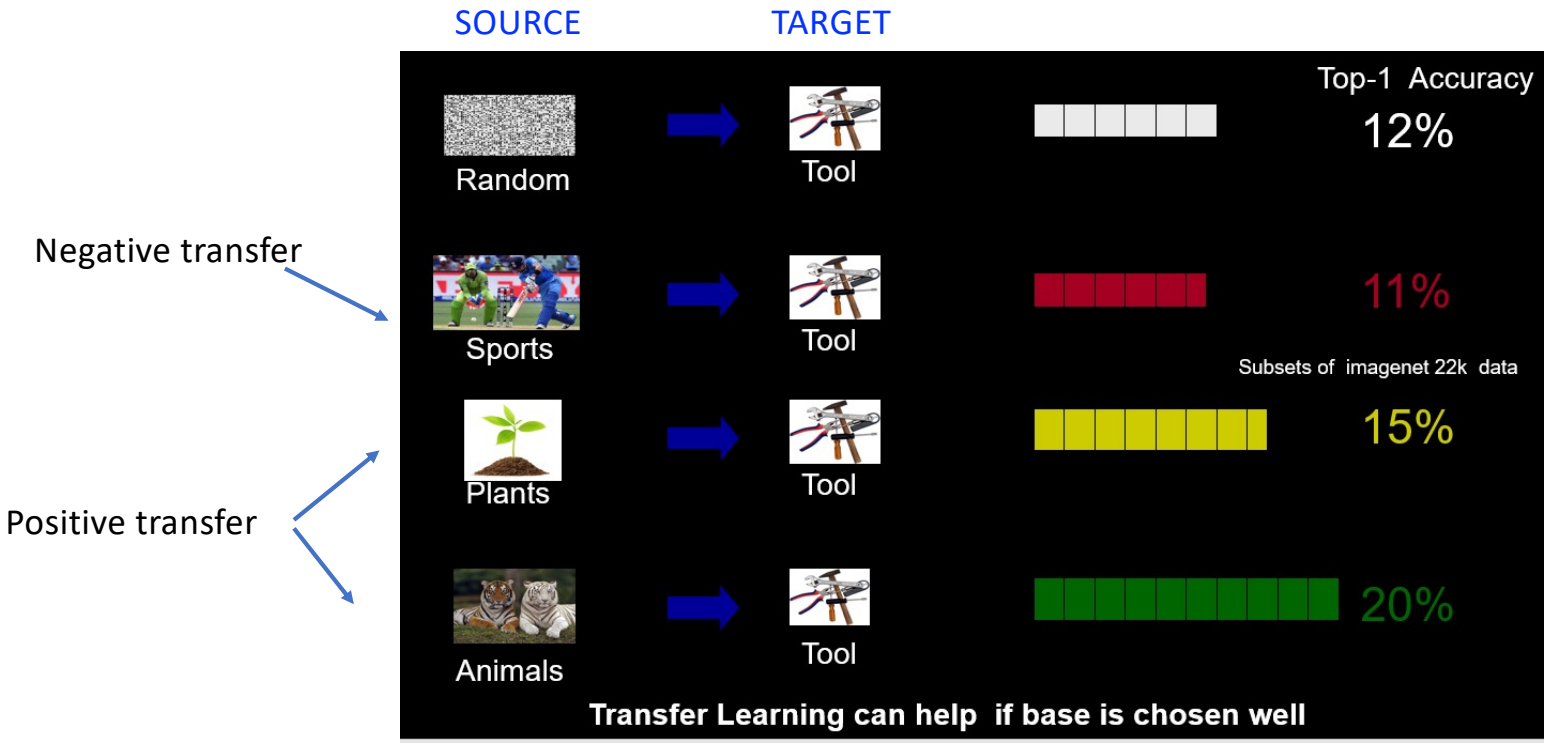
Transfer Learning Approaches

- *Common intuition*: Networks which have learned compact representations of a "source" task, can reuse these representations to achieve higher performance on a related "target" task.
- *Instance based* approaches attempt to identify appropriate data used in the source task to supplement target task training
- *Feature representation based* approaches attempt to leverage source task weight matrices
  - Trained weights in source network have captured a representation of the input that can be transferred by fine-tuning the weights or retraining the final dense layer of the network on the new task.

Improving Transferability in Transfer Learning

- **Selection of source model:** What is the best ”source” task/model to transfer knowledge for a “target” task ?
  - How to measure “similarity” between source and target task ?
    - Develop similarity measures between source and target datasets in feature space
  - Does size always matter ?
    - Will a source model trained on huge datasets will always outperform source models trained on smaller but more related datasets ?
  - Improper choice of a base dataset/model for a target could result in degraded performance compared to not using transfer learning (negative transfer)
- **Degree of finetuning**
  - Which layers to finetune and which to freeze ?
  - What should be the learning rate of layers to be finetuned ?
    - Higher learning rate ➔ loose information from source task

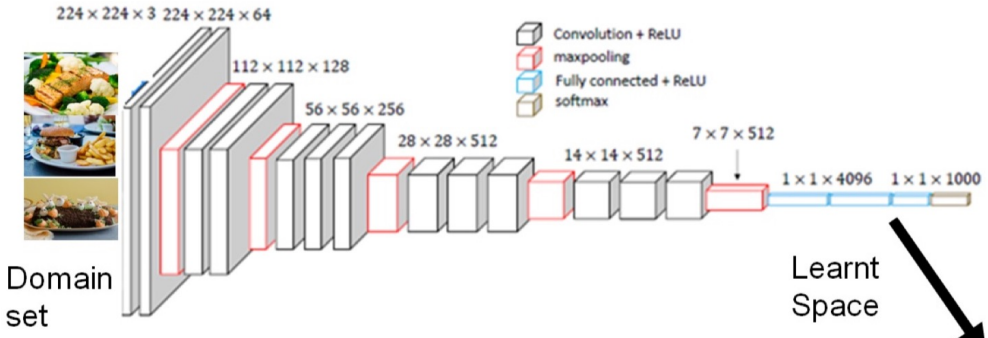
Good and bad sources

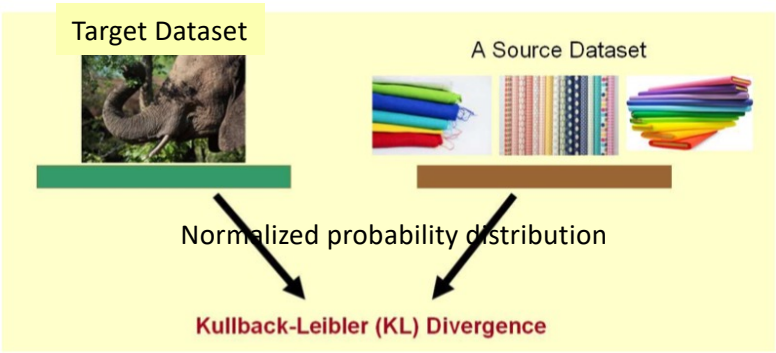


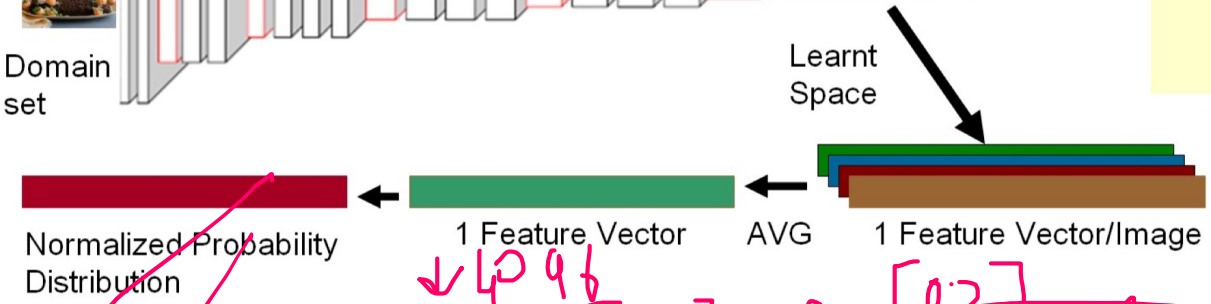
Efficient Selection of Source Models

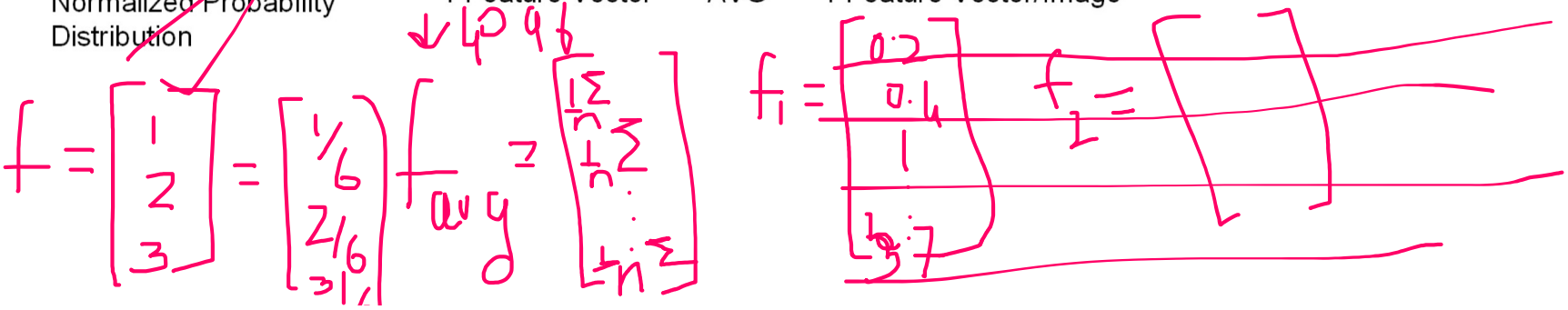
- Training consumes valuable time / compute resources, we do not want to do a lengthy optimization process or train many candidate models per task
- Predict to Learn (P2L) approach (Bhatta et al 2019)
  - An efficient method to estimate the appropriateness of a previously trained model for use with a new learning task
  - Precludes the exhaustive approach of fine-tuning all existing source models to search for the best fit.
  - P2L only requires a single forward pass of the target data set through a single reference model to identify the most likely candidate for fine-tuning.
- P2L accounts for two attributes:
  - Similarity between source and target dataset: source dataset which is similar to a target dataset in feature space is most likely to have a higher degree of transferability and hence a better candidate for fine-tuning
  - Size of source dataset

P2L Approach: dataset similarity in feature space





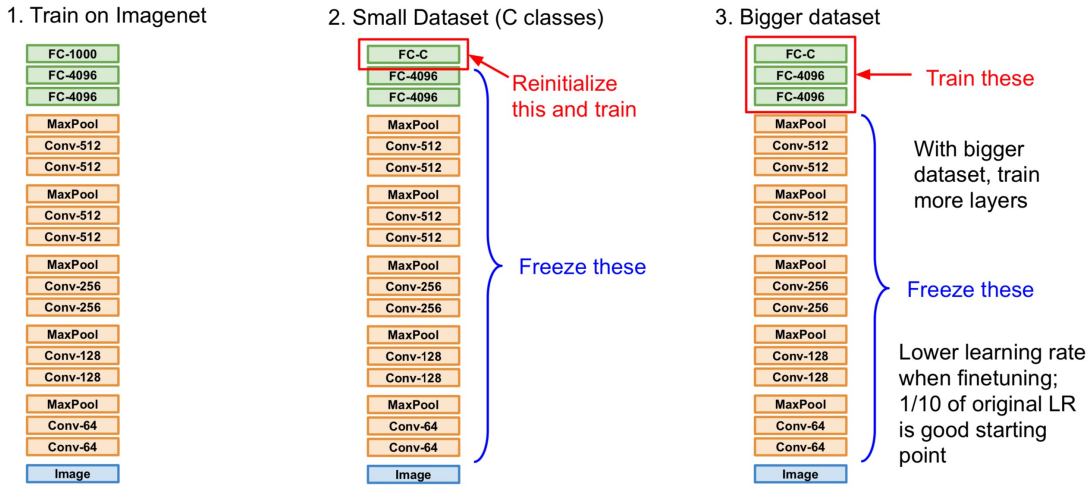




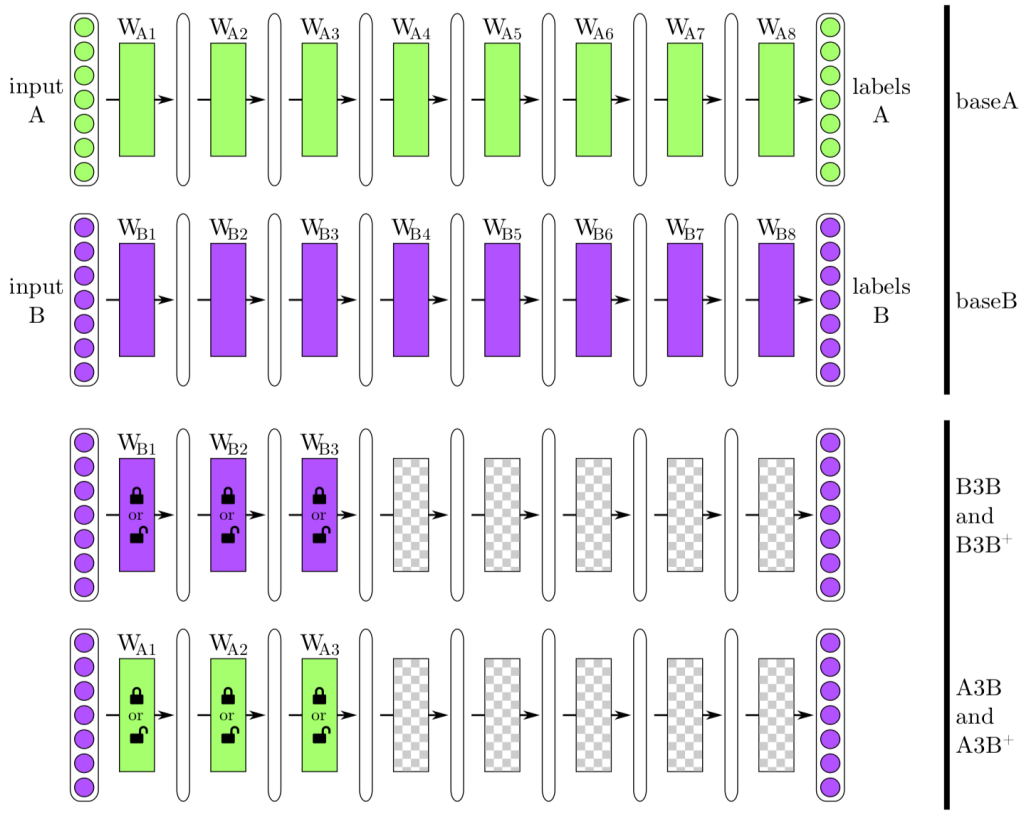
Transfer Learning: Basic Finetuning

- Take almost any deep network pre-trained on a large dataset of your choice
- Model zoo of pretrained models
  - TensorFlow: https://github.com/tensorflow/models
  - PyTorch: https://github.com/pytorch/vision
- Replace the last (classification) layer with a randomly initialized one
- Train only the new layer's weights, using the "frozen" embedding

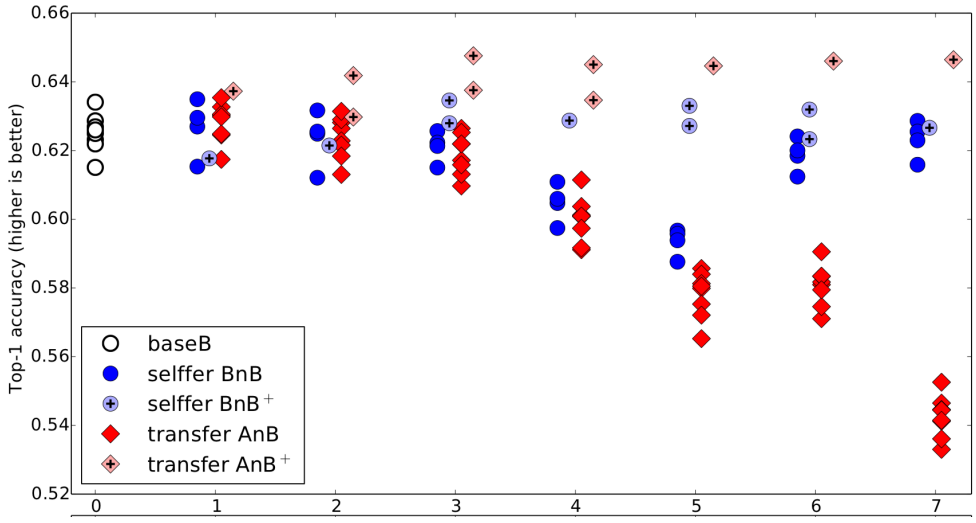
  - This baseline method works well in many settings…
  - …. Can we improve on it?



Transferability



Transferability



Layer n at which network is chopped and retrained

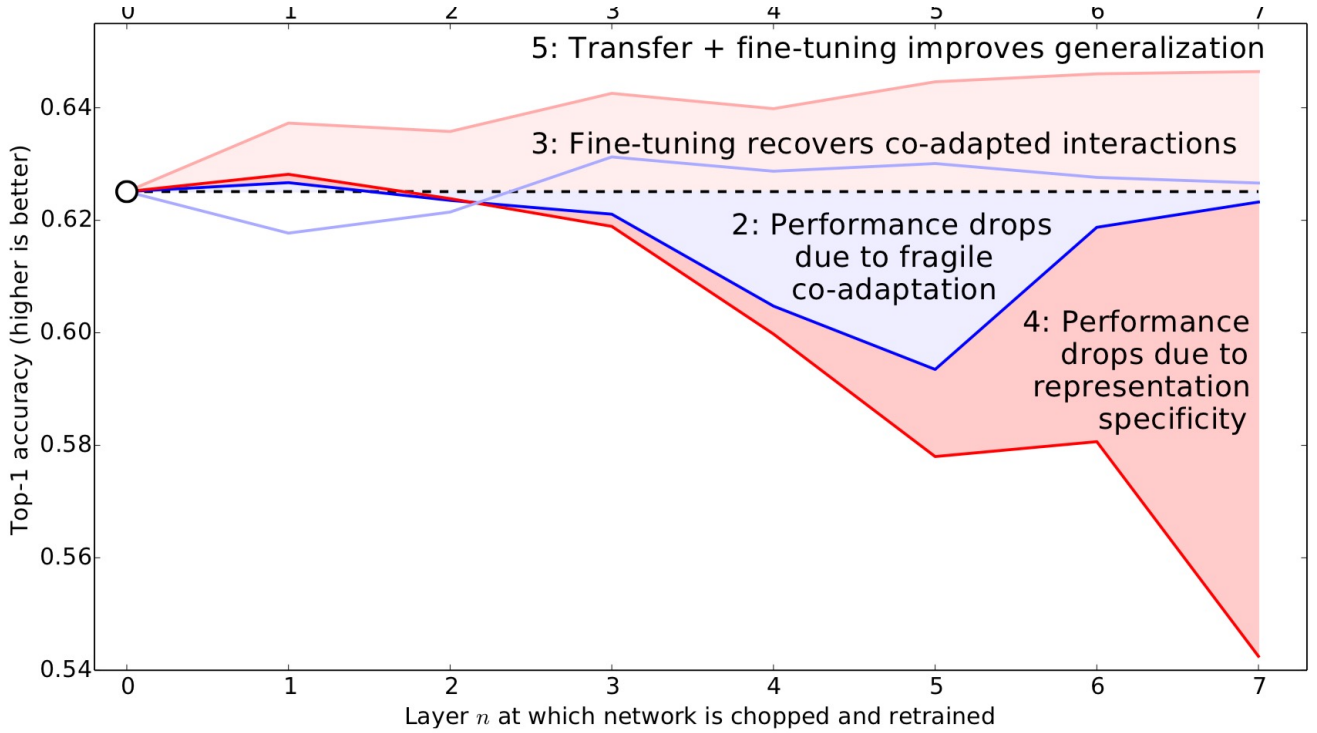
Each marker in the figure represents the average accuracy over the validation set for a trained network. The white circles above n = 0 represent the accuracy of baseB. There are eight points, because we tested on four separate random A/B splits. Each dark blue dot represents a BnB network. Light blue points represent BnB+ networks, or fine-tuned versions of BnB. Dark red diamonds are AnB networks, and light red diamonds are the fine-tuned AnB+ versions.

How transferable are features in deep neural networks?

Questions

- The white baseB circles show that a network trained to classify a random subset of 500 classes attains a top-1 accuracy of 0.625, or 37.5% error. This error is lower than the 42.5% top-1 error attained on the 1000-class network. Why ?

Transferability

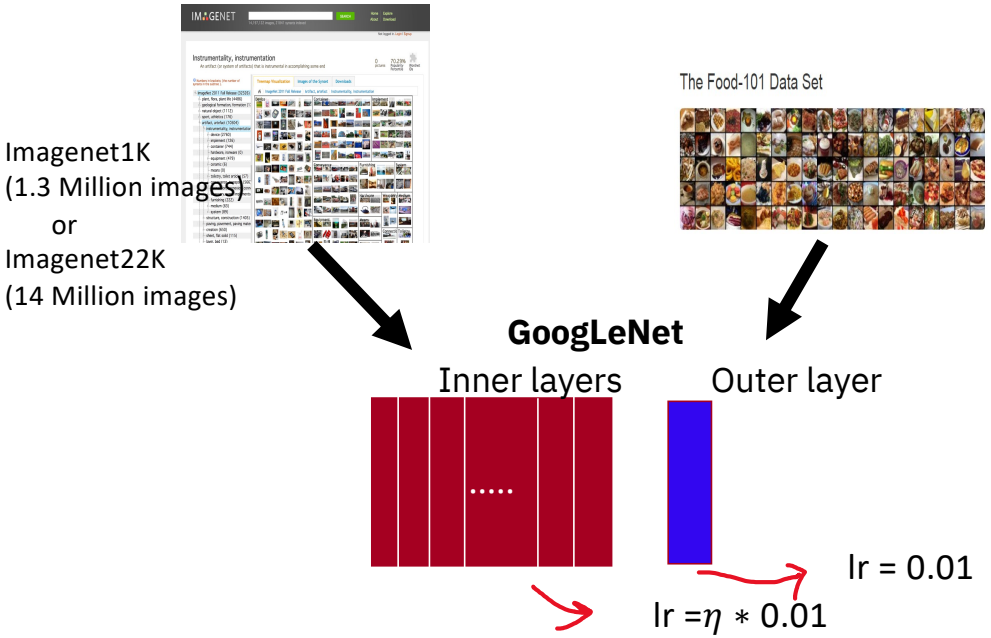


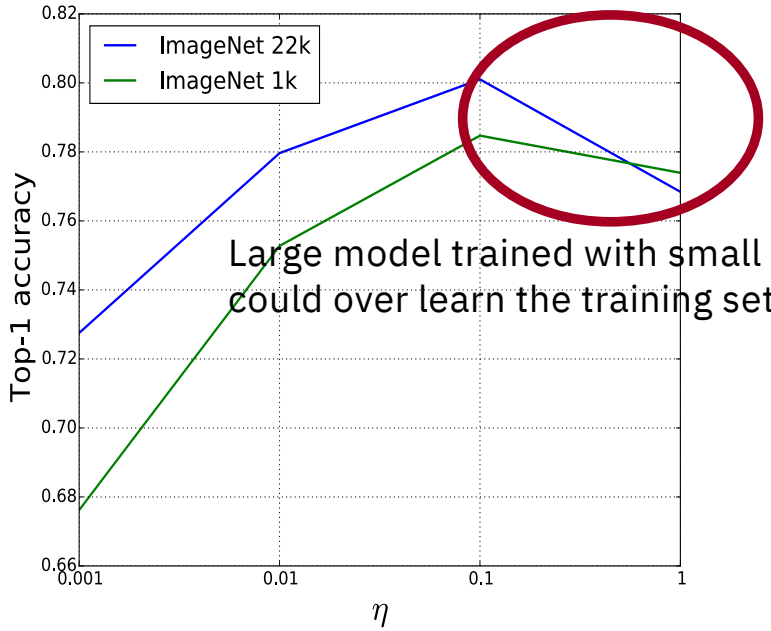
Lines connecting the means of each treatment.

Learn to Transfer

- Which source model to use ?
  - Caffe Model Zoo has several pretrained models for different tasks
  - TensorFlow 2 Detection Model Zoo has several models pretrained on COCO 2017 dataset
- Which method to use ?
  - Shallow learning driving SVMs
  - Fine tuning
  - Single source or ensemble
- Which layers to freeze and finetune ?
  - ResNet101 has 101 layers
  - GoogLeNet has 22 layers
  - VGG16 has 16 layers
- What is the performance requirement?
  - Latency
  - Memory Footprint
  - Target domain accuracy
- What curriculum should I follow?
  - Imagenet1K (1000 classes ,~1.3M images,) -> Food101 (101 classes, 101000 images) -> Greek Food
  - Imagenet22K (22K classes ,~15M images,) -> Greek Food

Impact of base model data size on transfer learning





Fine-tuning results on Food-101 dataset with varying numbers of learning rate multipliers for layers with pre-trained weights

Bhattacharjee et al, "Distributed learning of deep feature embedding for visual recognition tasks", IBM J of R & D, 2017

IBM Visual Recognition Service

- IBM Watson Visual Recognition Service
- Cloud based service
- Supports both image classification and object detection
  - Use of pre-trained models or developing custom models
- Video: Visual Recognition in Watson Studio
- Custom object detection
- Visual recognition with CoreML
  - Source code and instructions for yourself to try: https://github.com/bourdakos1/visual-recognition-with-coreml
  - Tutorial http://ibm.biz/coremlworkshop

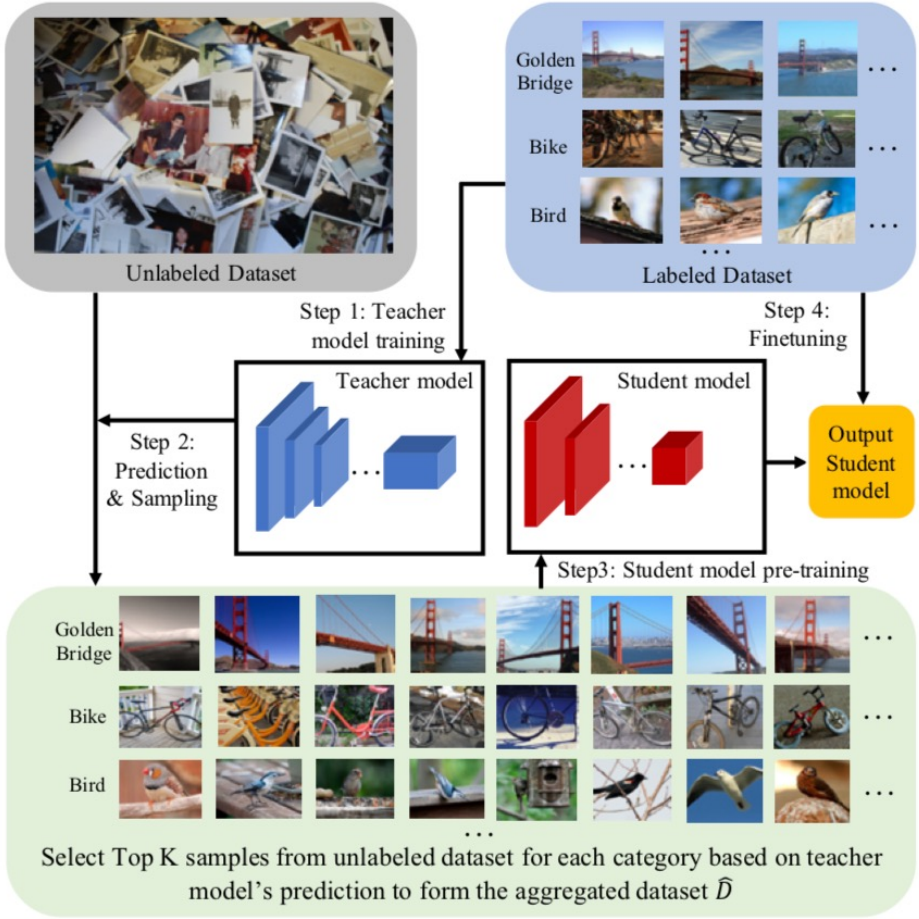
Exploiting Web Data for Training

- Web and social media are most important sources of data for vision research
- Vision datasets such as ImageNet (classification), PASCAL VOC (object detection), and MS COCO (object detection) are created from Google or Flickr **by harnessing human intelligence** to filter out the noisy images and label object locations
- Human labeling is costly ! AI Platform Data Labeling Service
- Can we learn visual representations and object detectors from the web data directly ?
  - Without the need for any human intervention

Automated Labeling of Images

- Pseudo labeling of unlabeled Data for **Transfer Learning**
- Goal: Exploit data from the wild to capture rich representations in source models
- Can we automatically label data and build base models for transfer learning ?
  - Semi-supervised learning: using supervised learning to train teacher model and use it to classify unlabeled images
  - Weakly supervised learning: using hashtags, captions of images from social media sites without any supervised learning

Semi-supervised Learning



(1) Train a teacher model on labeled data $\mathcal{D}$;

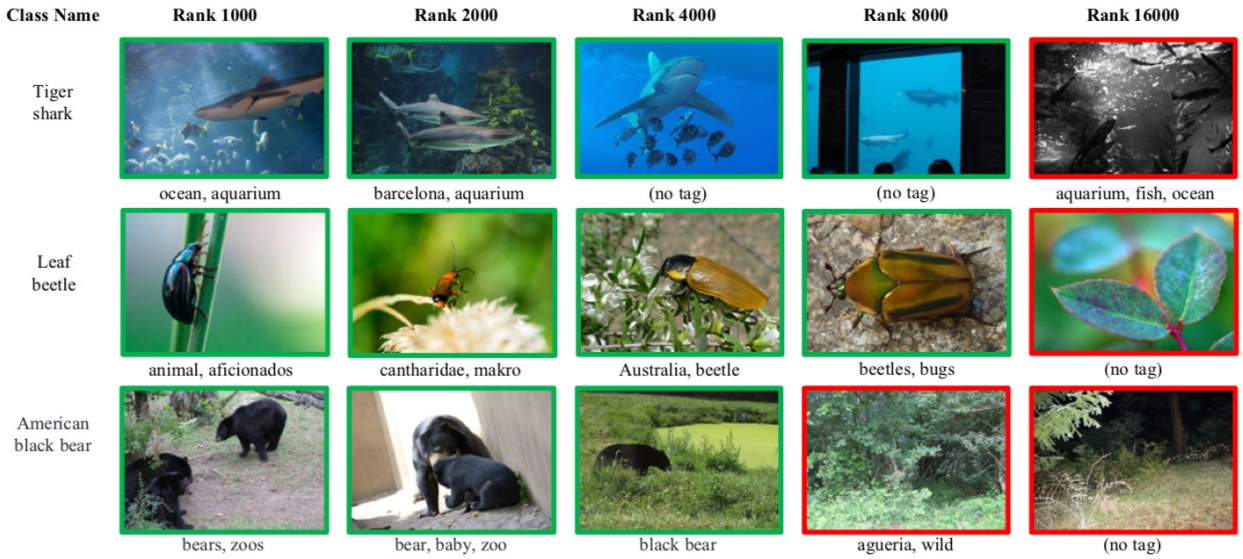
(2) Run the trained model on unlabeled data $\mathcal{U}$ and select relevant examples for each label to construct a new labeled dataset $\hat{\mathcal{D}}$;

(3) Train a new student model on $\hat{\mathcal{D}}$;

(4) Fine-tune the trained student on the labeled set $\mathcal{D}$.

Yalniz et al. Billion-scale semi-supervised learning for image classification. 2019

Labeling unlabeled images



- Teacher model is run on each example in unlabeled dataset to obtain the softmax prediction vector
- For each image, the classes associated with the **P** highest scores are retained
- For each class rank the images based on the corresponding classification scores
- Top-**K** images from each class is taken to create a new dataset

- K: Number of top samples from unlabeled dataset to keep for each class in the new dataset
- Probability of introducing false positive becomes higher as K increases

Parameters P and K

- Why P > 1 ?
  - It is difficult to identify accurately under-represented concepts
  - Some concepts occulted by more prominent co-occurring concepts
- $\hat{\mathcal{D}}_l$: Top K images of each class l

$$\hat{\mathcal{D}} = \bigcup_{l=1}^{L} \hat{\mathcal{D}}_l$$

- Can an image belong to multiple classes in $\hat{\mathcal{D}}$?
- Student model is still a multi-class classification problem
  - Allowing the image to be replicated for each class it belongs to

Performance of teacher-student learning

fine-tuning the model on clean labeled data is crucial to achieve good performance

Teacher model: ResNext-101-32x48

Supervised baseline

Student Model	Ours: Semi-supervised Pre-training	Ours: Semi-supervised Fine-tuned	Fully Supervised
ResNet-18	68.7	**72.6**	70.6
ResNet-50	75.9	**79.1**	76.4
ResNext-50-32x4	76.7	**79.9**	77.6
ResNext-101-32x4	77.5	**80.8**	78.5
ResNext-101-32x8	78.1	**81.2**	79.1
ResNext-101-32x16	78.5	**81.2**	79.6

Table 2: ImageNet1k-val top-1 accuracy for students models of varying capacity before and after fine-tuning compared to corresponding fully-supervised baseline models.

Teacher Model	Teacher # Params	Teacher top-1	Student top-1	Gain (%)
ResNet-18	8.6M	70.6	75.7	-0.7
ResNet-50	25M	76.4	77.6	+1.2
ResNext-50-32x4	25M	77.6	78.2	+1.8
ResNext-101-32x4	43M	78.5	78.7	+2.3
ResNext-101-32x8	88M	79.1	78.7	+2.3
ResNext-101-32x16	193M	79.6	79.1	+2.7
ResNext-101-32x48	829M	79.8	79.1	+2.7

Table 3: Varying the teacher capacity for training a ResNet-50 student model with our approach. The gain is the absolute accuracy improvement over the supervised baseline.

teacher model brings a significant improvement over the supervised baseline for various capacity target models (1.6-2.6%)

Performance of teacher-student learning

Sensitivity to size of unlabeled dataset

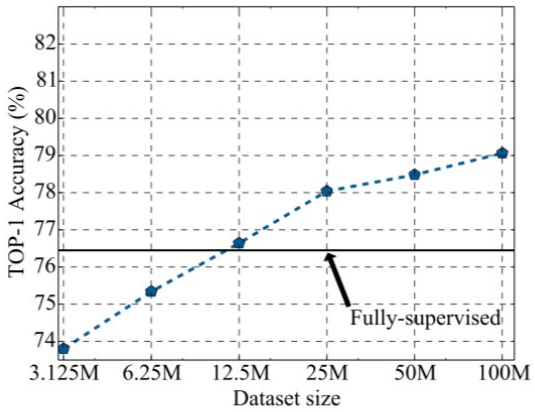


Figure 3: ResNet-50 student model accuracy as a function of the size of the unlabeled dataset $\mathcal{U}$.

Sensitivity to training iterations

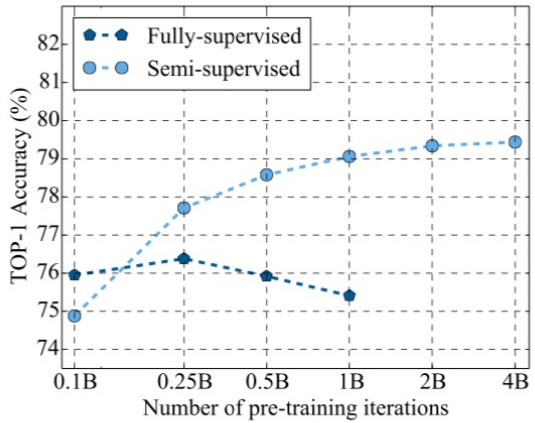


Figure 4: Effect of number of training iterations on the accuracy of fully-supervised and semi-supervised ResNet-50 student models.

Sensitivity to K

Increased labeling noise

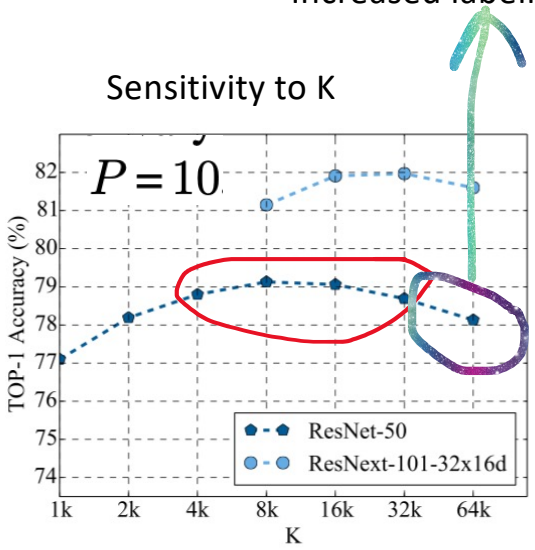


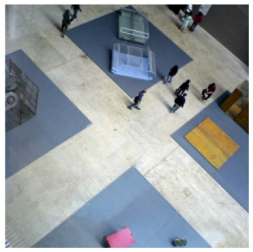
Figure 5: Student model accuracies as a function of the sampling hyper-parameter K.

Learning from weak labels

- Facebook contains tons of data like this:

the veranda hotel portixol palma

plane approaching zrh avro regional jet rj



not as impressive as embankment that s for sure

student housing by lungaard tranberg architects in copenhagen click here to see where this photo was taken

article in the local paper about all the unusual things found at otto s home

this was another one with my old digital camera i like the way it looks for some things though slow and lower resolution than new cameras another problem is that it s a bit of a brick to carry and is a pain unless you re carrying a bag with some room it s nearly x x and weighs ounces new one is x x and weighs ounces i underexposed this one a bit did exposure bracketing script underexposure on that camera looks melty yummy gold kodak film like

https://cs.nyu.edu/~fergus/teaching/vision/9_detection_pt2.pdf

Weakly Supervised Learning Architecture

- Train convolutional network to predict words that co-occur with an image
  - Flickr 100M dataset contains~100M photos with associated "captions"
- Treat each individual word in a photo's caption as a target for that photo
  - A multi-label learning problem with extremely noisy labels
- Train convolutional networks to predict the words from the images
  - Using standard convnet architectures
- Train using multi-class logistic loss function with 100K classes

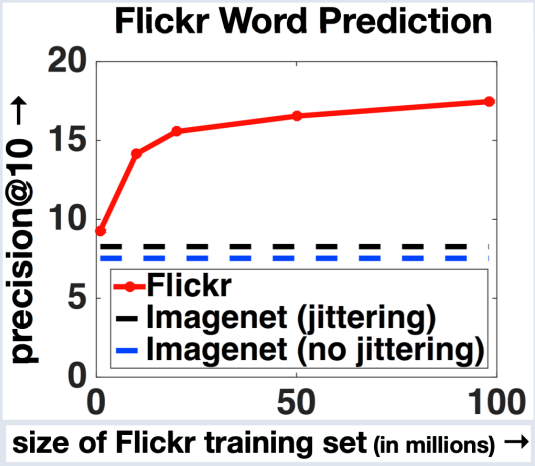
$$\ell(\theta, \mathbf{W}; \mathcal{D}) = \frac{-1}{N} \sum_{n=1}^{N} \sum_{k=1}^{K} y_{nk} \log \left[\frac{\exp(\mathbf{w}_k^\top f(\mathbf{x}_n; \theta))}{\sum_{k'=1}^{K} \exp(\mathbf{w}_{k'}^\top f(\mathbf{x}_n; \theta))} \right]$$

Experimental evaluation

- **Experiment 1:** Given a photo, predict the words

Word prediction: Learning curves

- How much data do we need to train good word prediction models?

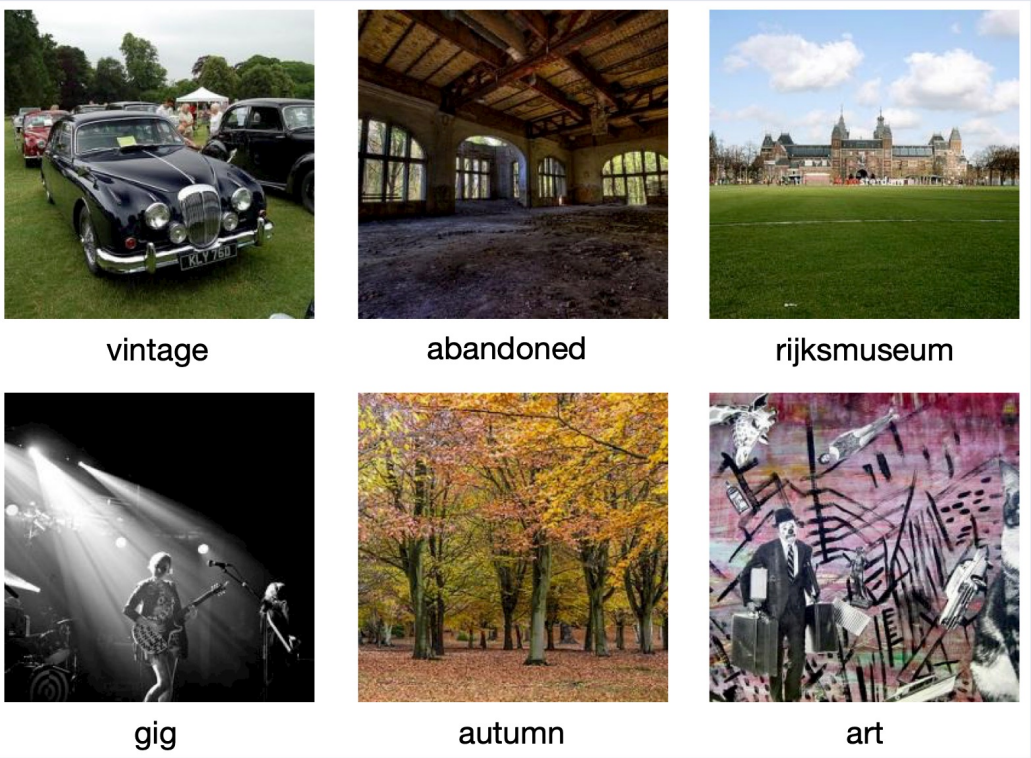


- Having tens of millions of weakly supervised images helps!

https://cs.nyu.edu/~fergus/teaching/vision/9_detection_pt2.pdf

Word prediction

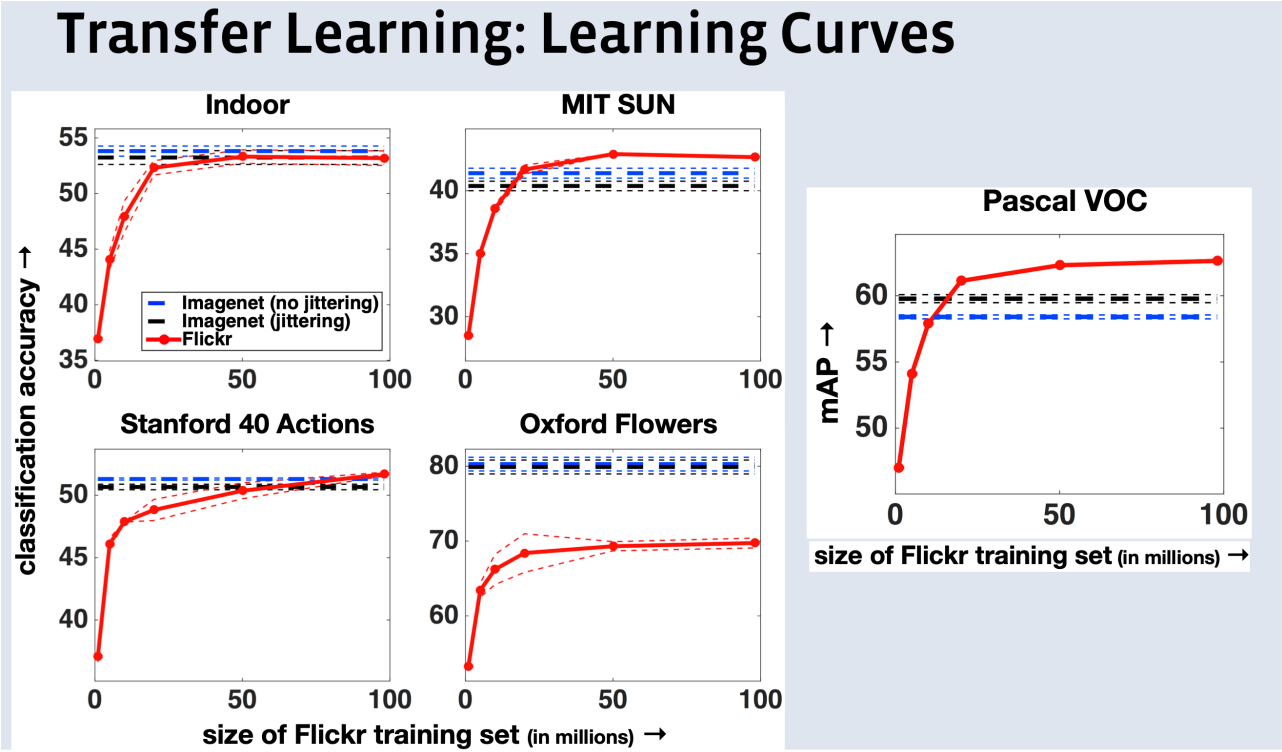
- Six images with high scores for arbitrary words:



https://cs.nyu.edu/~fergus/teaching/vision/9_detection_pt2.pdf

Experimental evaluation

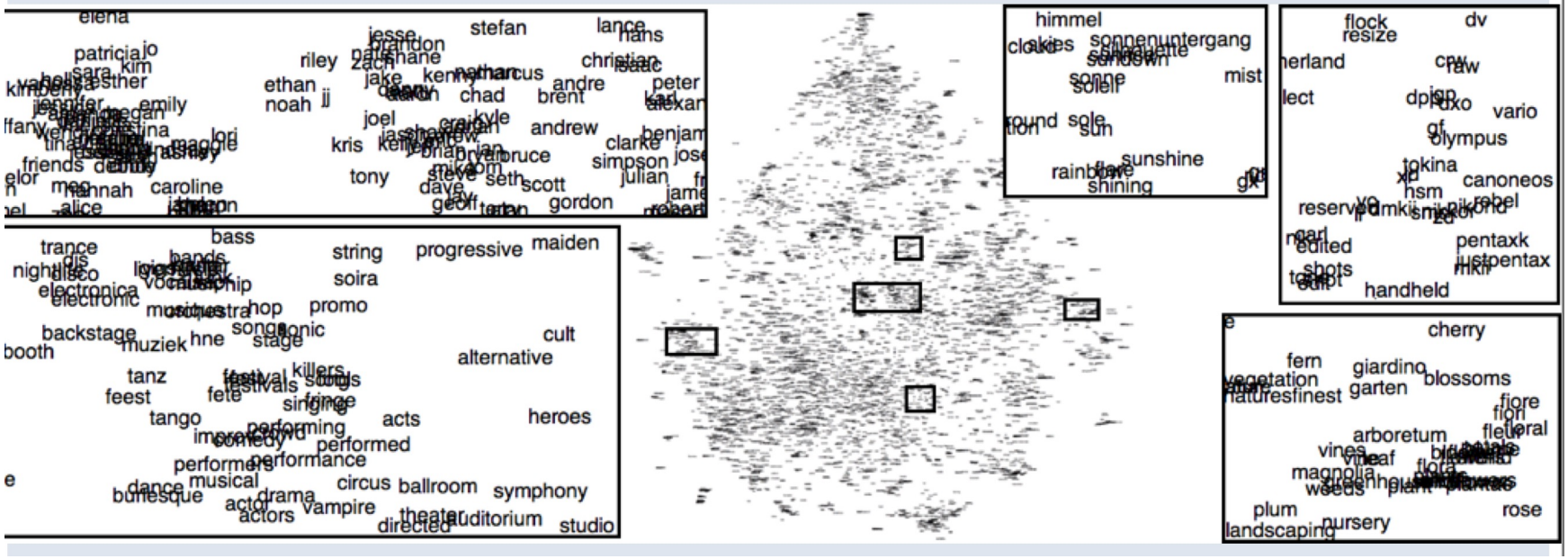
Experiment 2: Use the features learned by the CNNs for transfer learning to other vision tasks



https://cs.nyu.edu/~fergus/teaching/vision/9_detection_pt2.pdf

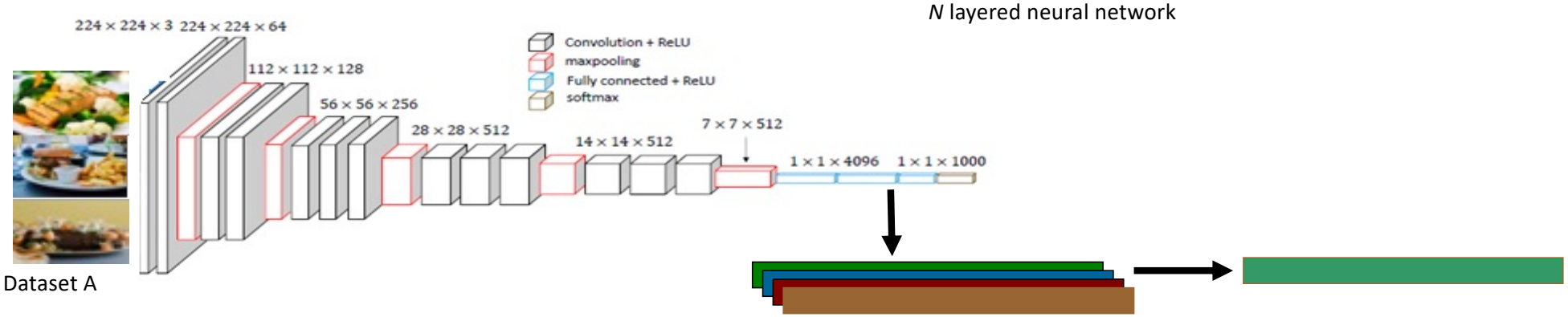
Analyzing the word embeddings

- Output layer of our convnets is essentially a word embedding
- This embedding has captured semantic information:



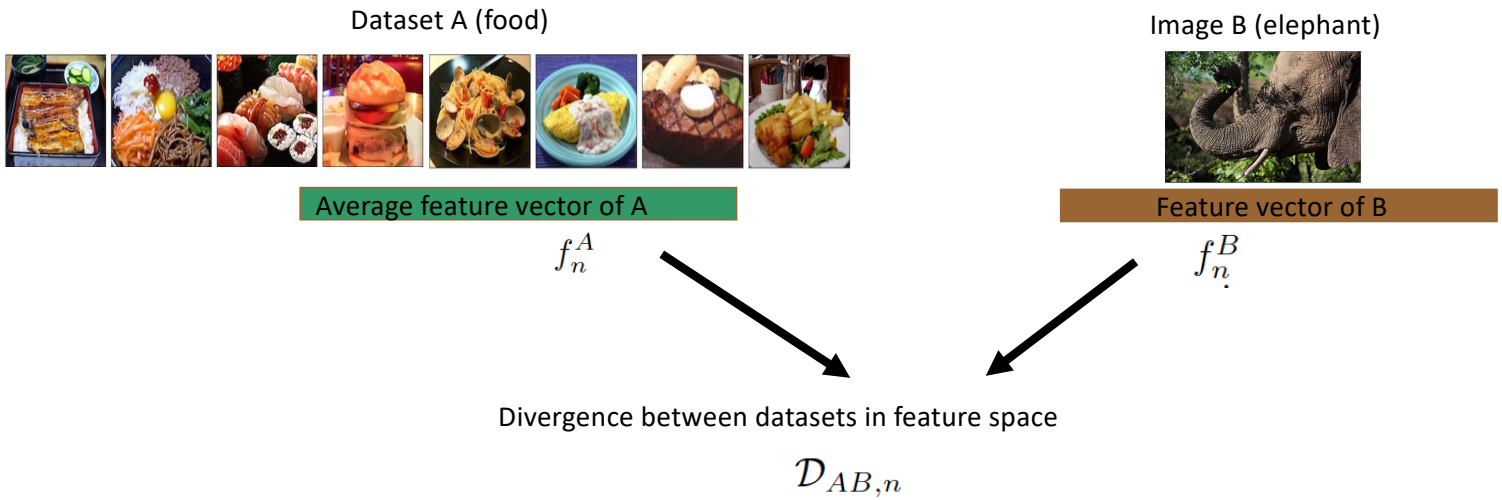
https://cs.nyu.edu/~fergus/teaching/vision/9_detection_pt2.pdf

Feature Vector Embeddings



- Average over the activations from layer n
- For $n = N - 1$, vector of 4096 dimension

Divergence in Feature Space



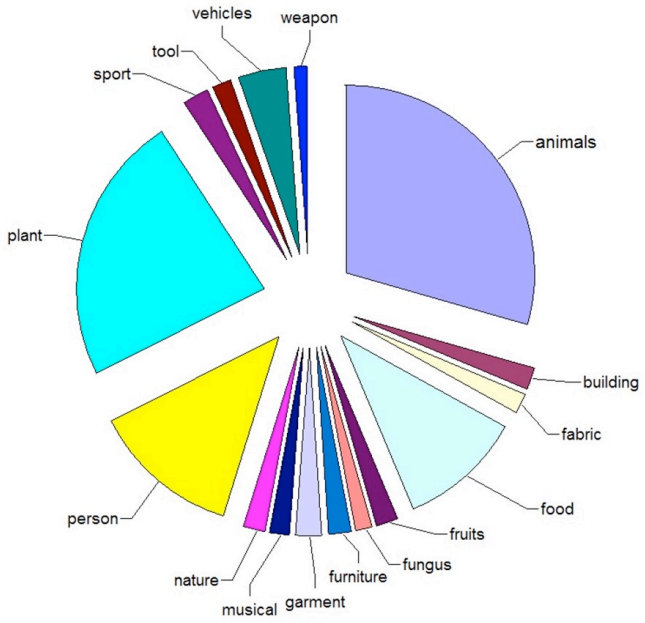
- Different metrics to measure divergence
  - Cosine similarity
  - Kullback-Leibler divergence

Divergence based Pseudo Labeling to Improve Transfer Learning

- For each unlabeled sample, calculate its distance (in feature space) from a set of named anchor points representing known and labeled categories, like animal, plant, tool
- Construct pseudo-labels for unlabeled samples based on these distances
  - Pseudo-labels are a sequence of semantically descriptive names: e.g., ⟨tool, plant⟩
- Train a source model using these automatically generate pseudo labels
- Finetune the source model on target dataset

Anchor Points Using Imagenet22K

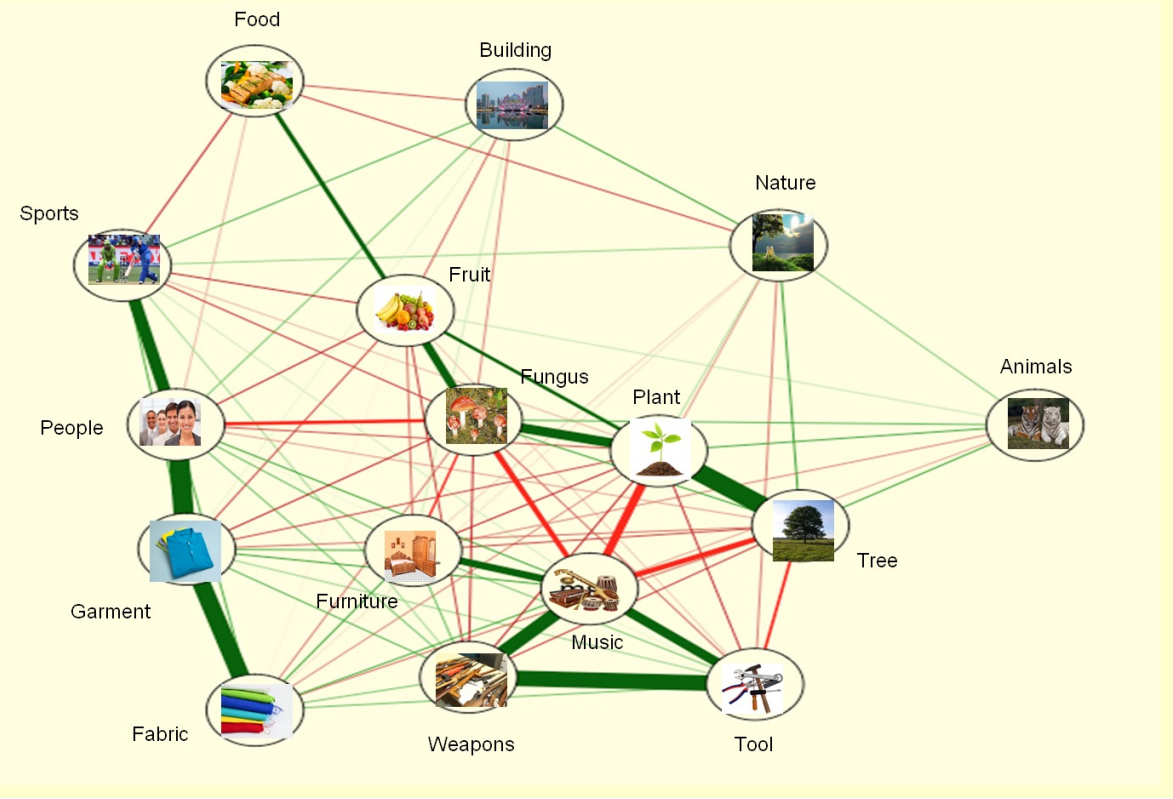
16 source datasets



Vertical Partitions of ImageNet22K

Dataset	Images	Classes
sport	200692	170
tool	175095	317
fruit	185091	307
fabric	159110	232
building	193928	240
furniture	195052	187
garment	214172	252
music	137770	156
nature	192255	241
weapon	102946	138
person	1203512	2880
plant	2224817	4040
tree	562555	995
fungus	135919	299
food	999470	1500
animal	2783256	3796

Divergence in Feature Space between datasets



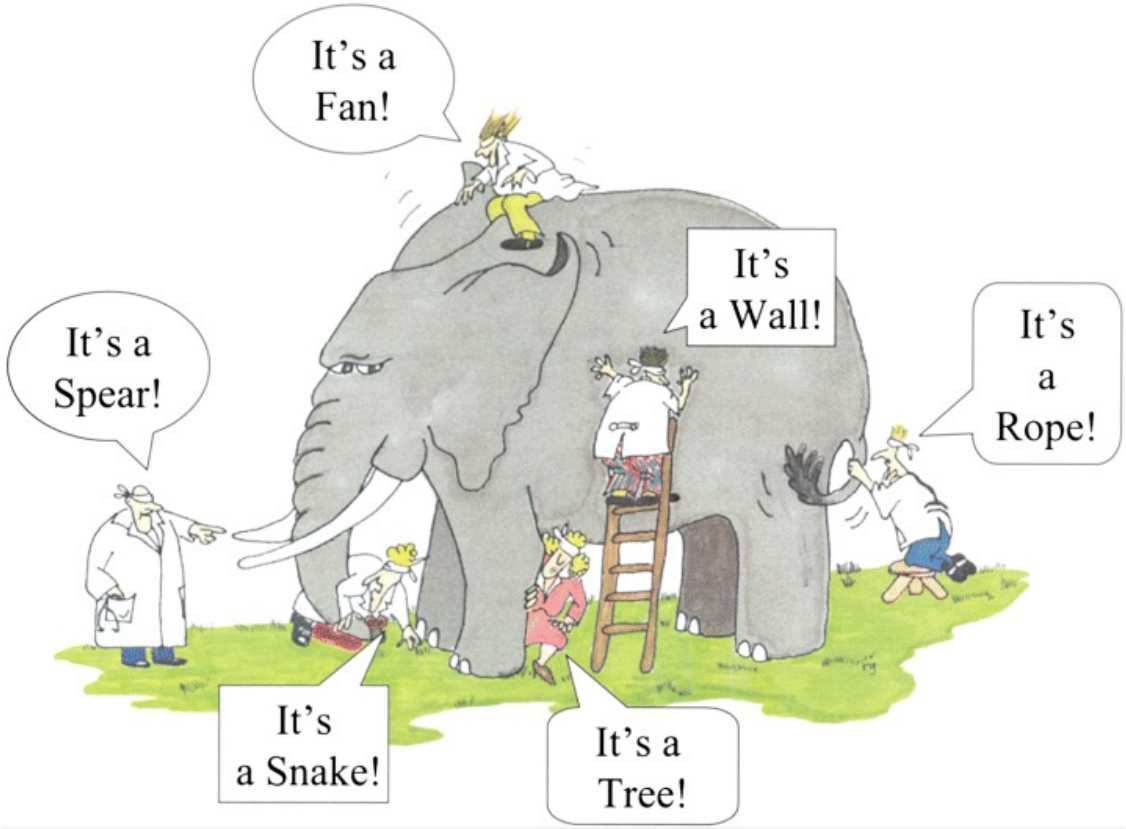
KL divergence between dataset distributions

$$D_{KL}(p,q) = \sum_i p(i) log\left(\frac{p(i)}{q(i)}\right)$$

Each of the 16 anchor points is a point in 4K dimension space

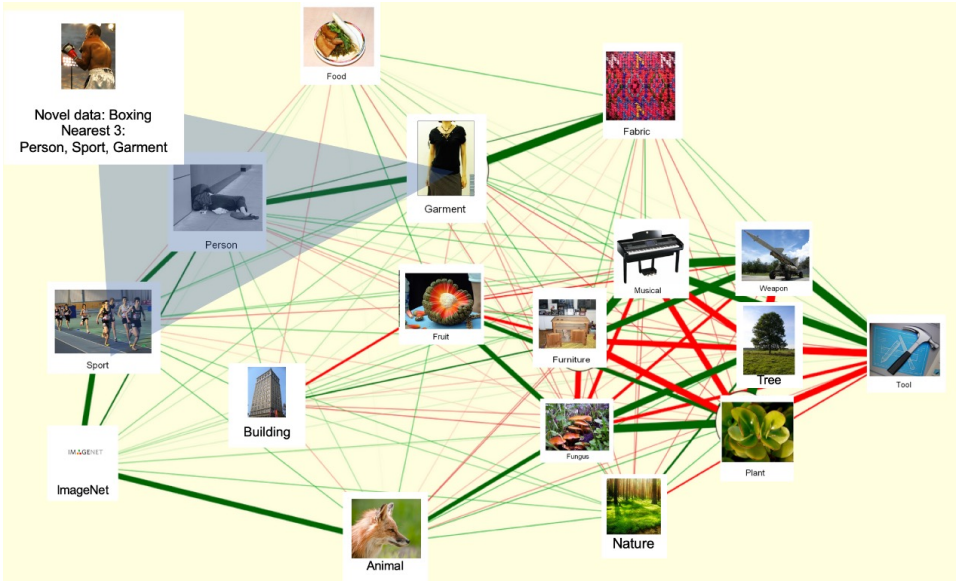
Divergence in feature space between an image with existing datasets
Measure of how close the image is to the dataset in feature space

An Analogy

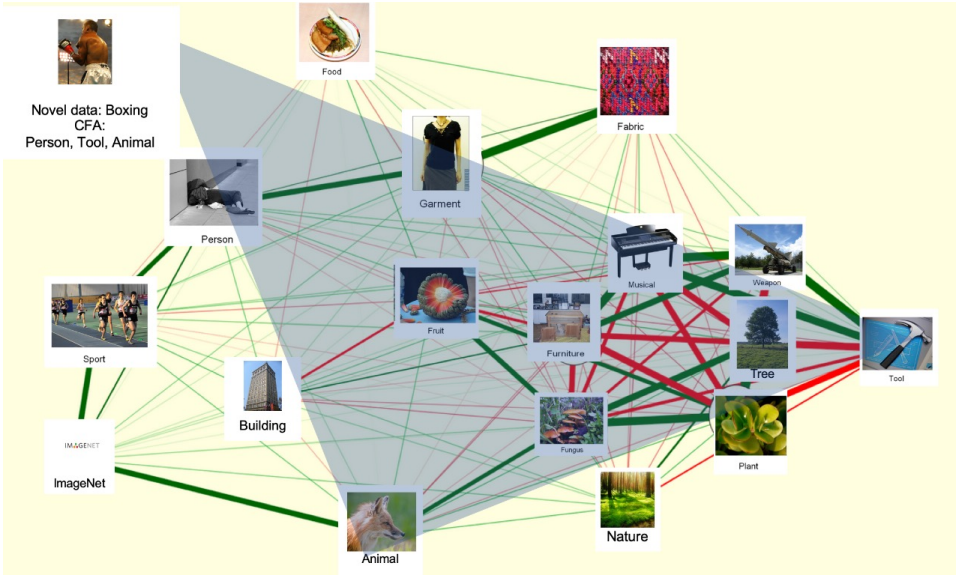


Divergence Based Pseudo Labeling

Nearest-N
(Nearset-1: closest , Nearest-2: closest two)



Closest Farthest (CFA)

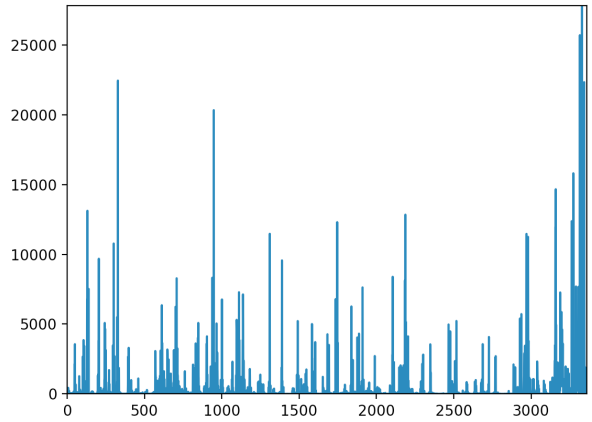


Visual Example: Nearest-3 and CFA

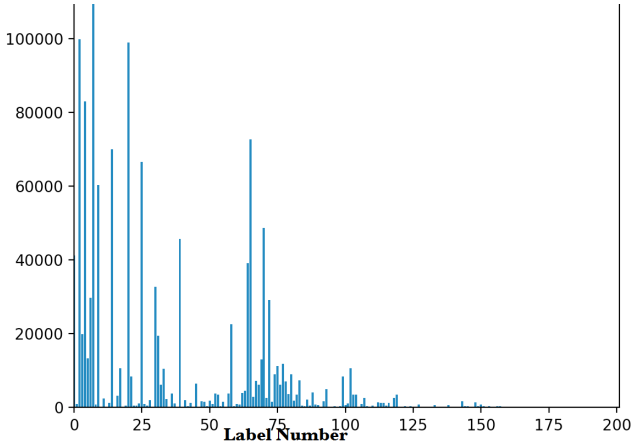
Pseudo Label Datasets

dataset	labels	images/label		accuracy
		mean	std-dev	(base model)
Nearest-1	16	80073	79325	78.74%
Nearest-2	240	5338	10146	55.75%
Nearest-3	3360	381	1503	37.07%
Uniform	1144	1119	193	33.49%
CFA	201	6373	17648	81.01%
Random	1000	1200	0	0.08 %
Vanilla	1000	1200	0	67.14%

Label distribution in Nearest-3



Label distribution in CFA

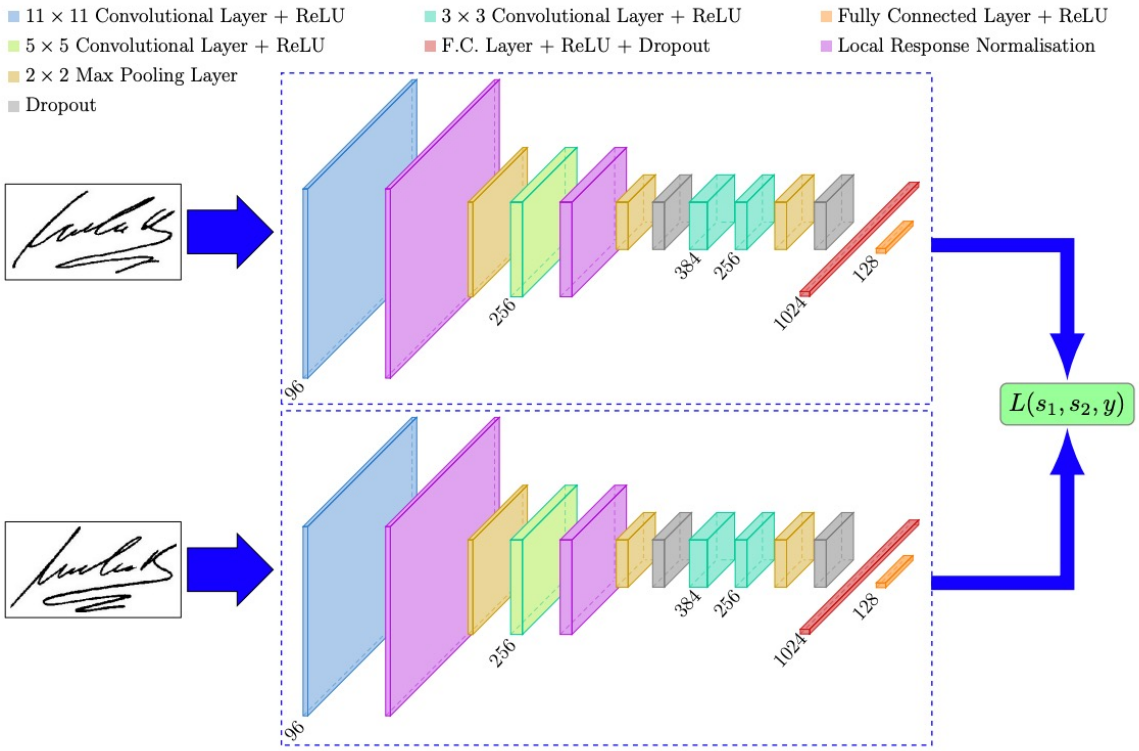


Experimental Evaluation

Base →	Pseudo-labeled						Imagenet1K
Target ↓	Nearest-1	Nearest-2	Nearest-3	CFA	Uniform	Random	Vanilla
music	42.98%	43.60 %	**43.86%**	42.87 %	43.71%	1.57%	47.19%
tool	38.79%	39.12%	**39.44%**	39.40%	39.39%	1.24%	42.65%
weapon	29.51%	**30.24%**	30.21%	29.46%	29.92%	2.09%	32.25%
fungus	21.28%	21.96%	**22.16%**	21.78%	21.88%	1.60%	23.59%
flowers	75.94%	74.90%	72.88%	**76.64%**	72.36%	0.43%	85.13%
sport	28.68%	30.46%	**30.76%**	30.01%	30.74%	0.98%	37.37%
person	6.87%	7.25%	7.89%	7.29%	**8.05%**	0.12%	10.12%
food	8.58%	9.21%	**9.62%**	9.19%	9.36%	0.13%	12.52%
fruit	18.53%	**19.53%**	19.12%	19.05%	18.54%	0.82%	25.95%
garment	16.84%	17.30%	**18.05%**	17.29%	17.61%	0.68%	24.48%
animal	15.40%	17.54%	**18.46%**	17.31%	18.40%	0.10%	24.87%
plant	10.46%	11.27%	**11.69%**	11.14%	11.47%	0.08%	15.34%

Table 4: Transfer learning accuracy for different target datasets; best accuracy shown in bold.

Siamese Networks



Contrastive Loss

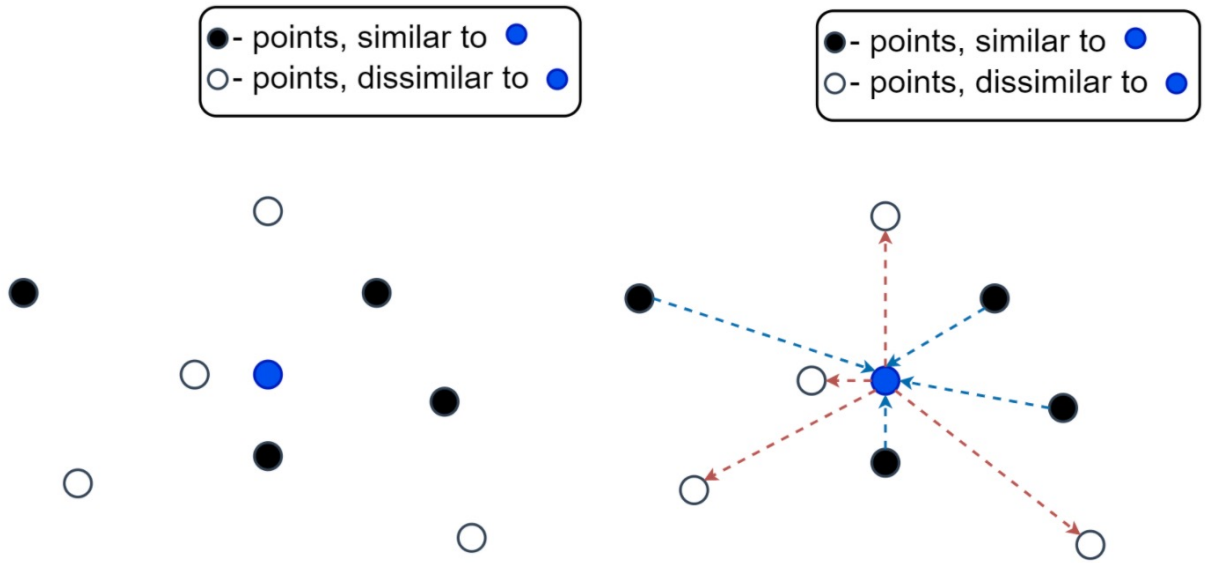
Introduced in [the paper](#) by Le Cunn et al in 2006

$$L(W,(Y,\vec{X_1},\vec{X_2})^i) = (1-Y)L_S\left(D_W^i\right) + YL_D\left(D_W^i\right)$$

The **Dw** term in parenthesis is the **similarity** (or, rather, **dissimilarity**) between 2 transformed data points, given by Le Cunn like so:

$$D_W(\vec{X_1},\vec{X_2}) = \|G_W(\vec{X_1}) - G_W(\vec{X_2})\|_2$$

Explaining Contrastive Loss Function



We want to minimize intra-class distance and maximize inter-class distance

An example Contrastive Loss

$$L(W, Y, \vec{X}_1, \vec{X}_2) =$$
$$(1 - Y)\frac{1}{2}(D_W)^2 + (Y)\frac{1}{2}\{max(0, m - D_W)\}^2$$

Why we need margin m in contrastive loss?