Lecture 9:

Generating Supervision

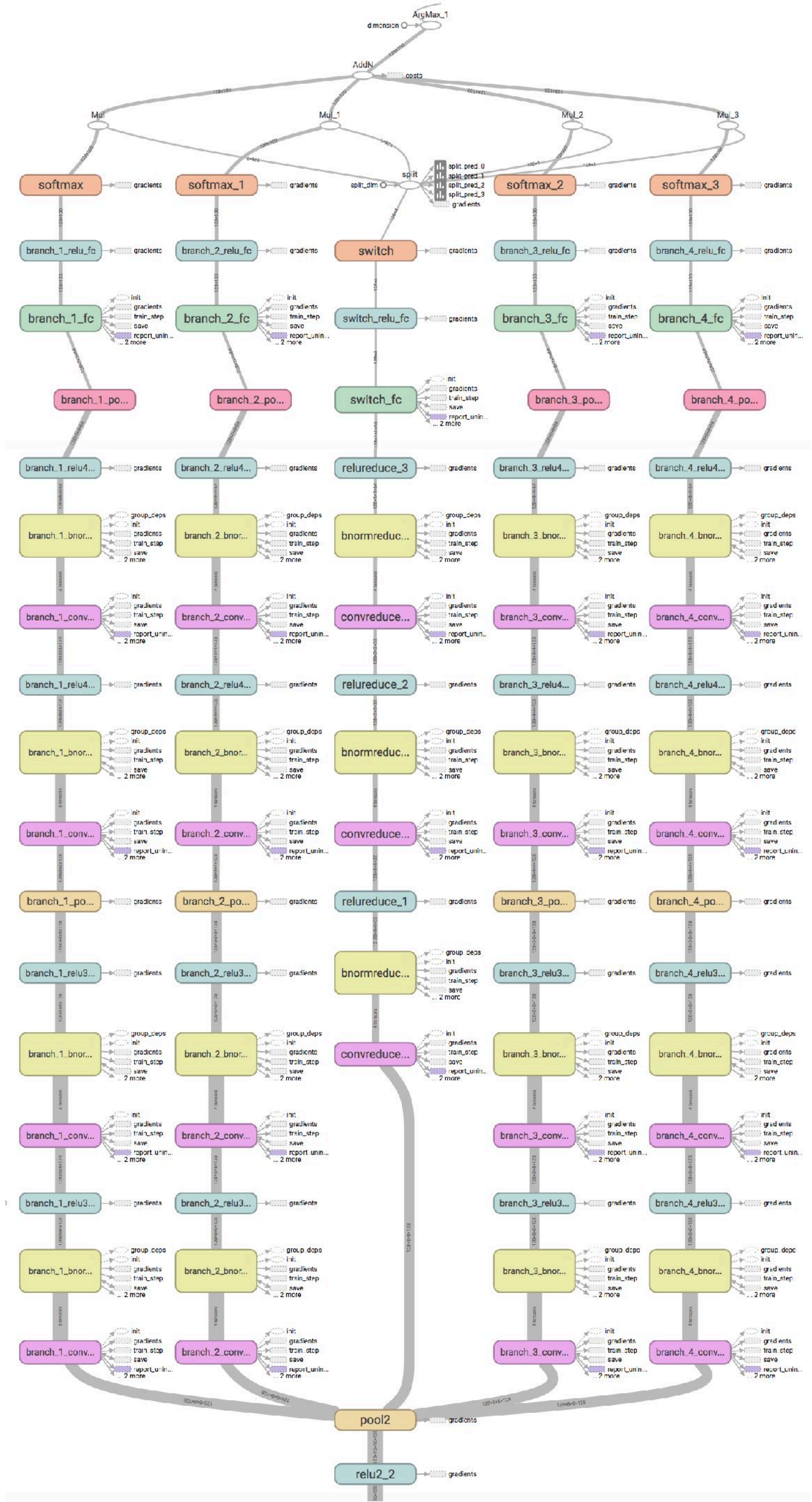
Visual Computing Systems
Stanford CS348K, Spring 2023

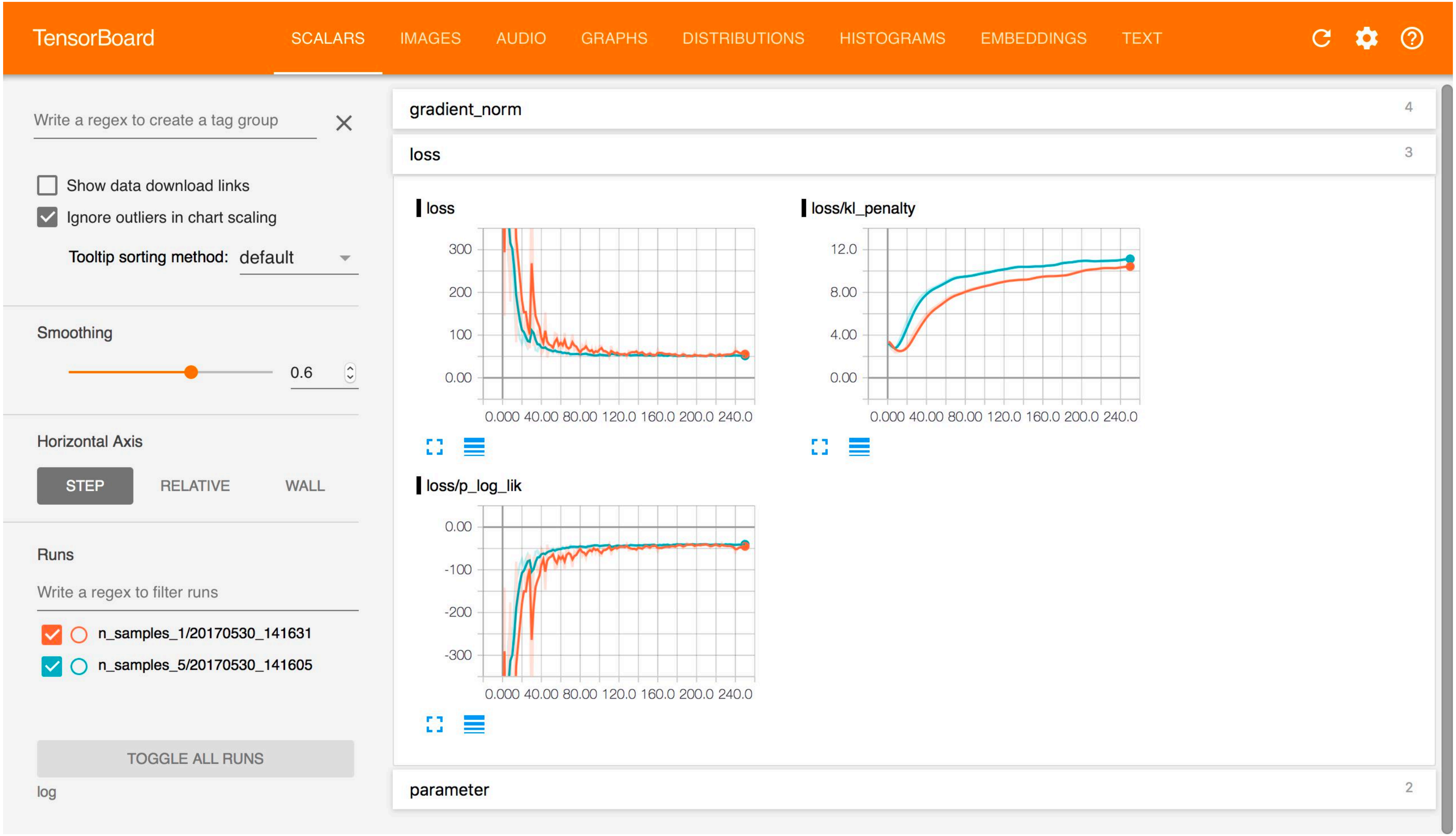
Today's agenda

- Much of this class involved discussing the Snorkel paper(s) from the reading list

PyTorch/TensorFlow/MX.Net data-flow graphs

- Key abstraction: a program is a DAG of (large granularity) operations that consume and produce N-D tensors





TensorBoard
SCALARS
IMAGES
AUDIO
GRAPHS
DISTRIBUTIONS
HISTOGRAMS
EMBEDDINGS
TEXT
Write a regex to create a tag group
Show data download links
Ignore outliers in chart scaling
Tooltip sorting method: default
Smoothing
0.6
Horizontal Axis
STEP
RELATIVE
WALL
Runs
Write a regex to filter runs
n_samples_1/20170530_141631
n_samples_5/20170530_141605
TOGGLE ALL RUNS
log
gradient_norm
4
loss
3
loss
300
200
100
0.00
0.000 40.00 80.00 120.0 160.0 200.0 240.0
loss/kl_penalty
12.0
8.00
4.00
0.00
0.000 40.00 80.00 120.0 160.0 200.0 240.0
loss/p_log_lik
0.00
-100
-200
-300
0.000 40.00 80.00 120.0 160.0 200.0 240.0
parameter
2

Services provided by ML "frameworks"

- Functionality:
  - Implementations of wide range of useful operators
    - Conv, dilated conv, relu, softmax, pooling, separable conv, etc.
    - Implementations of various optimizers:
      - Basic SGD, with momentum, Adagrad, etc.
  - Ability to compose operators into large graphs to create models
  - Carry out back-propagation
- Performance:
  - High-performance implementation of operators (layer types)
  - Scheduling onto multiple GPUs, parallel CPUs (and sometimes multiple machines)
  - Automatic sparsification and pruning
- Meta-optimization:
  - Hyper-parameter search
  - More recently: neural architecture search

How to improve system support for ML?

Hardware/software for…
faster inference?
faster training?

Compilers for fusing layers,
performing code optimizations?

List of papers at
MLSys 2020 Conference

Mon Mar 02, 2020

Time	Ballroom A
07:00 AM (Breaks)	
07:45 AM (Breaks)	Opening Remarks
08:00 AM (Orals)	Distributed and Parallel Learning Algorithms
	A System for Massively Parallel Hyperparameter Tuning
08:25 AM (Orals)	PLink: Discovering and Exploiting Locality for Accelerated Distributed Training on the public Cloud
08:50 AM (Orals)	Federated Optimization in Heterogeneous Networks
09:15 AM (Orals)	BPPSA: Scaling Back-propagation by Parallel Scan Algorithm
09:40 AM (Orals)	Distributed Hierarchical GPU Parameter Server for Massive Scale Deep Learning Ads Systems
10:30 AM (Orals)	Efficient Model Training
	Resource Elasticity in Distributed Deep Learning
10:55 AM (Orals)	SLIDE : In Defense of Smart Algorithms over Hardware Acceleration for Large-Scale Deep Learning Systems
11:20 AM (Orals)	FLEET: Flexible Efficient Ensemble Training for Heterogeneous Deep Neural Networks
11:45 AM (Orals)	Breaking the Memory Wall with Optimal Tensor Rematerialization
01:30 PM (Invited Talks)	Theory and Systems for Weak Supervision
02:30 PM (Orals)	Efficient Inference and Model Serving
	What is the State of Neural Network Pruning?
02:55 PM (Orals)	SkyNet: a Hardware-Efficient Method for Object Detection and Tracking on Embedded Systems
03:20 PM (Orals)	MNN: A Universal and Efficient Inference Engine
03:45 PM (Orals)	Willump: A Statistically-Aware End-to-end Optimizer for Machine Learning Inference
04:30 PM (Orals)	Model / Data Quality and Privacy
	Attention-based Learning for Missing Data Imputation in HoloClean
04:55 PM (Orals)	Privacy-Preserving Bandits
05:20 PM (Orals)	Understanding the Downstream Instability of Word Embeddings
05:45 PM (Orals)	Model Assertions for Monitoring and Improving ML Models
06:00 PM (Demonstrations)	

Tue Mar 03, 2020

Time	Ballroom A
07:00 AM (Breaks)	
08:00 AM (Orals)	ML programming models and abstractions & ML applied to systems
	AutoPhase: Juggling HLS Phase Orderings in Random Forests with Deep Reinforcement Learning
08:25 AM (Orals)	Automatically batching control-intensive programs for modern accelerators
08:50 AM (Orals)	Predictive Precompute with Recurrent Neural Networks
09:15 AM (Orals)	Sense & Sensitivities: The Path to General-Purpose Algorithmic Differentiation
09:40 AM (Orals)	Ordering Chaos: Memory-Aware Scheduling of Irregularly Wired Neural Networks for Edge Devices
10:30 AM (Orals)	Efficient inference and model serving
	Fine-Grained GPU Sharing Primitives for Deep Learning Applications
10:55 AM (Orals)	Improving the Accuracy, Scalability, and Performance of Graph Neural Networks with Roc
11:20 AM (Orals)	OPTIMUS: OPTImized matrix MUltiplication Structure for Transformer neural network accelerator
11:45 AM (Orals)	PoET-BiN: Power Efficient Tiny Binary Neurons
01:30 PM (Invited Talks)	The Emerging Role of Cryptography in Trustworthy AI
02:30 PM (Orals)	Quantization of deep neural networks
	Memory-Driven Mixed Low Precision Quantization for Enabling Deep Network Inference on Microcontrollers
02:55 PM (Orals)	Trained Quantization Thresholds for Accurate and Efficient Fixed-Point Inference of Deep Neural Networks
03:20 PM (Orals)	Riptide: Fast End-to-End Binarized Neural Networks
03:45 PM (Orals)	Searching for Winograd-aware Quantized Networks
04:30 PM (Orals)	Efficient Model Training 2
	Blink: Fast and Generic Collectives for Distributed ML
04:55 PM (Orals)	A Systematic Methodology for Analysis of Deep Learning Hardware and Software Platforms
05:20 PM (Orals)	MotherNets: Rapid Deep Ensemble Learning
05:45 PM (Orals)	MLPerf Training Benchmark

Today's theme

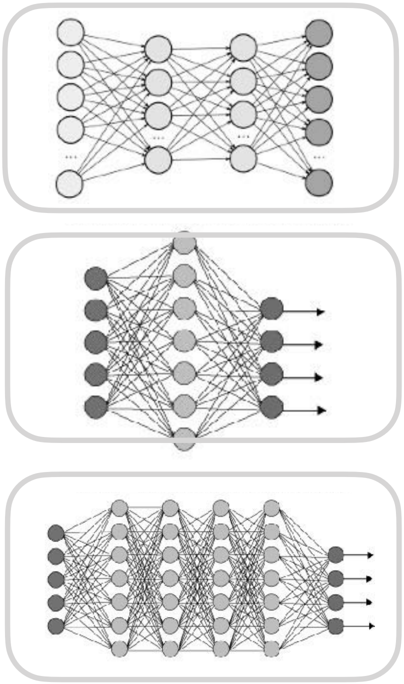
- Today, in many domains large collections of *unlabeled data* are readily accessible

- But labels for the data (supervision) is extremely precious

- Implication: ML engineers are interested in using any means necessary to acquire sources of supervision

Today's problem setup

Given:

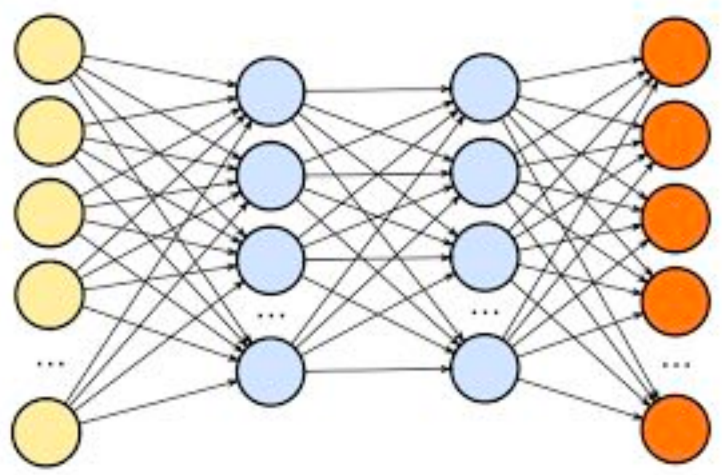


Pre-trained models
(for other tasks)

Huge corpus of unlabeled data
Perhaps with a sparse set of human labels

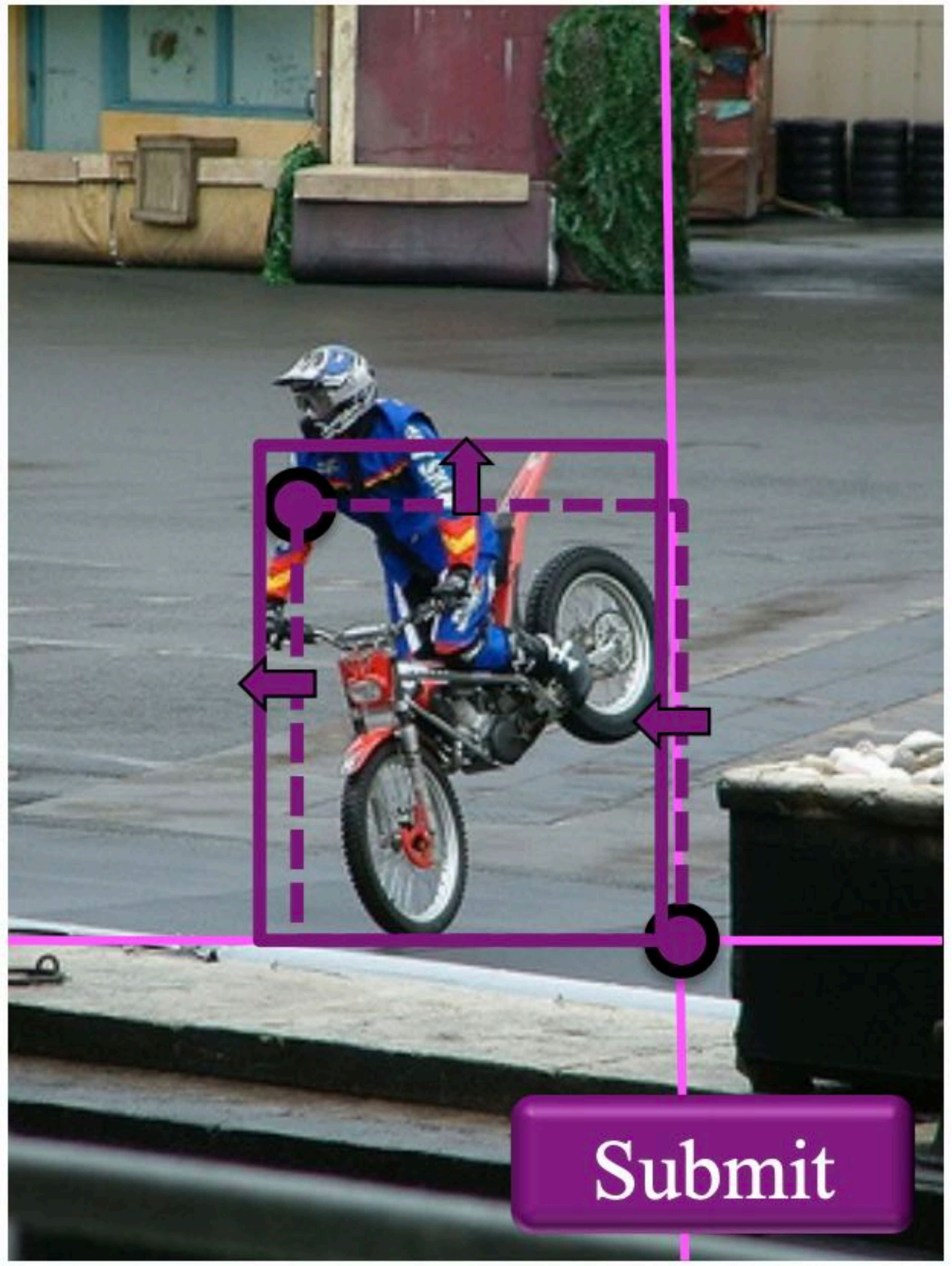
Abundant
Compute

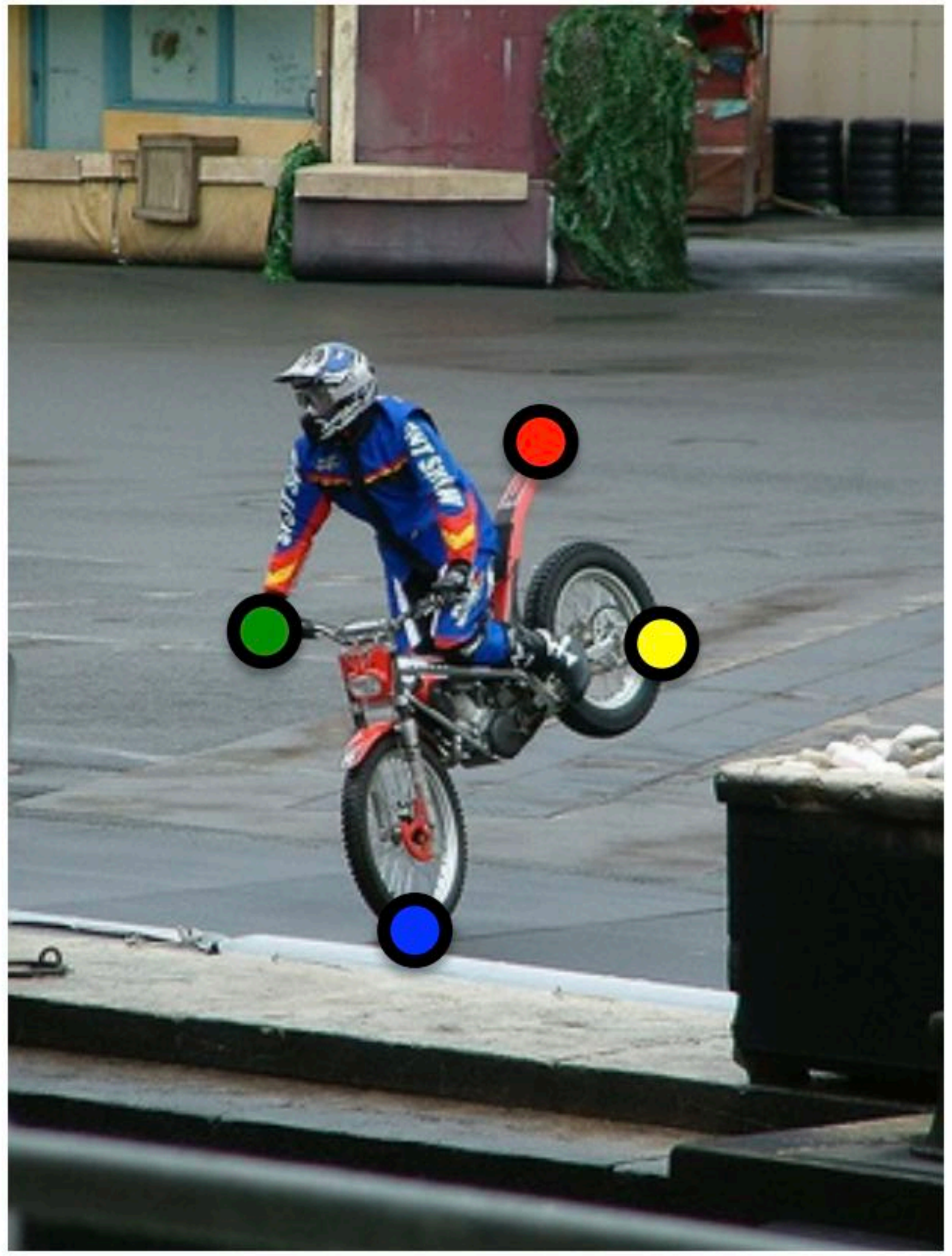
Goal: generate large amounts of supervision for use in training a model for a new task of interest



One research thrust: making human labelers more efficient

Example: "extreme clicking" is a faster way to define an object bounding box AND IT ALSO gives four points on the object's silhouette





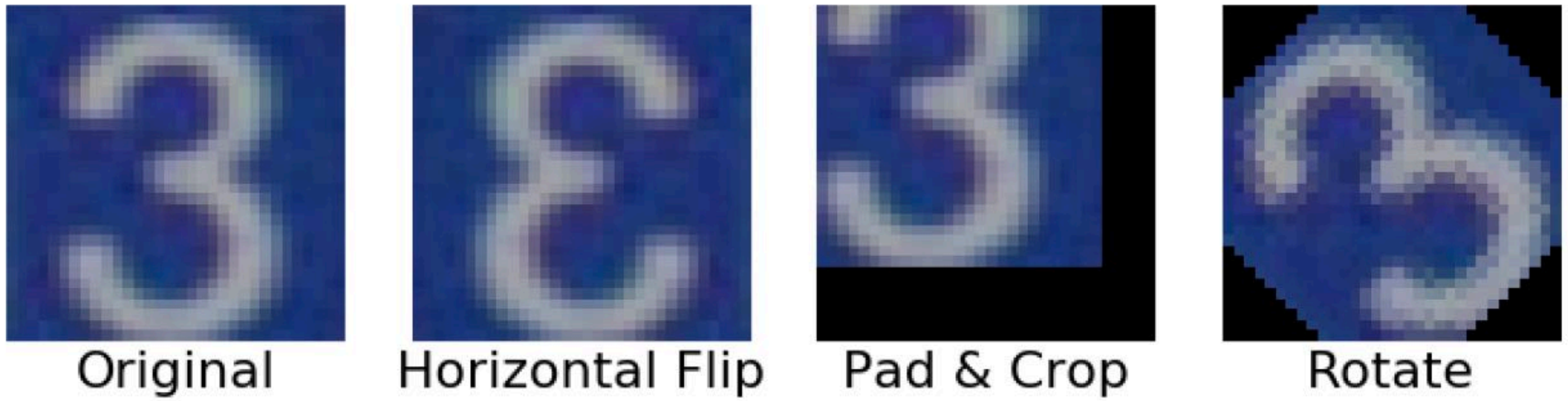
5x faster for humans to label

[Source: Papadopoulos et al. ICCV 2017]

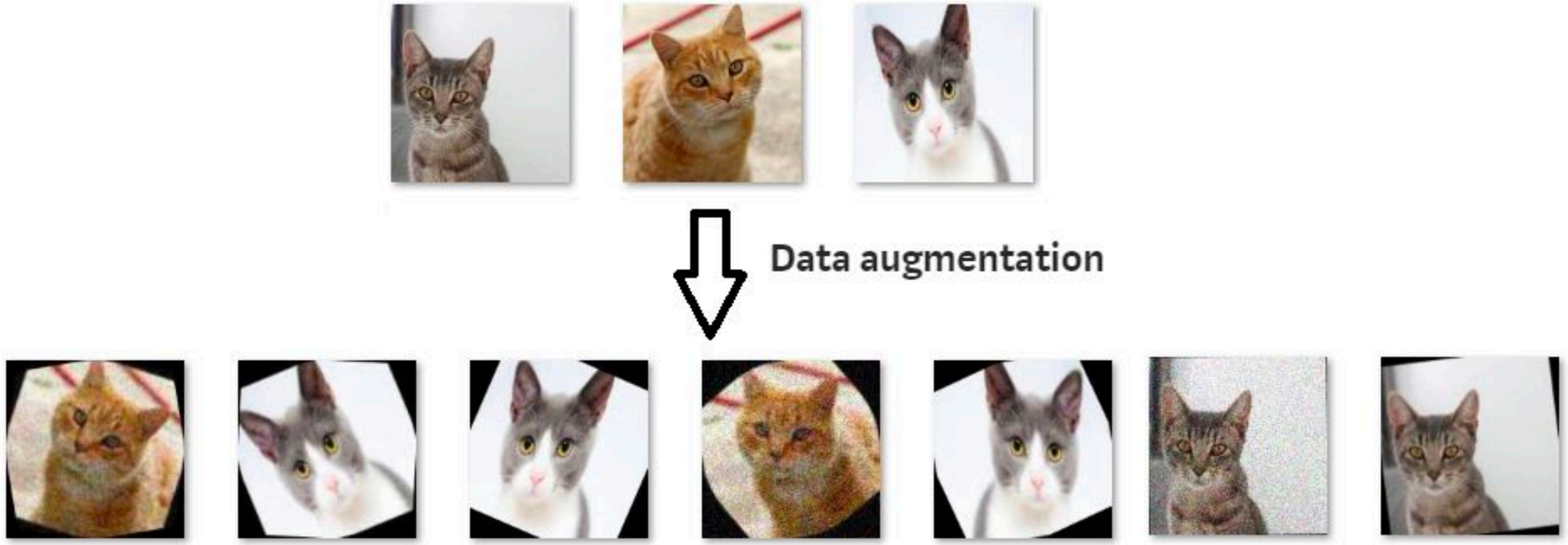
Amplify sparse human labels:
Automatically transfer labels from labeled data points to “similar” unlabeled data points

Data augmentation

Apply category-preserving transformations to images to increase size of labeled dataset.



[Image credit: Ho et al. ICML 2019]

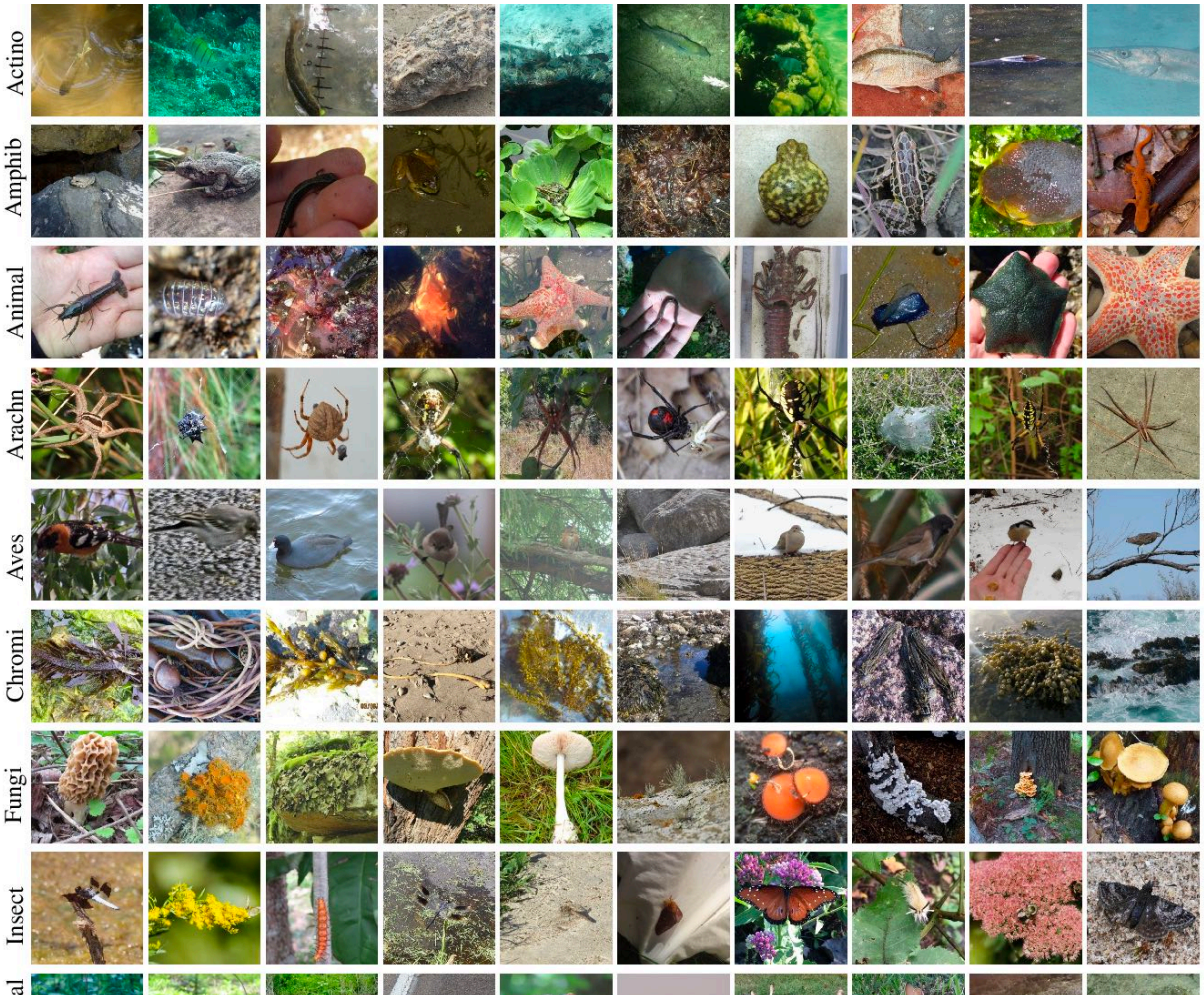


[Source: https://medium.com/@thimblot/data-augmentation-boost-your-image-dataset-with-few-lines-of-python-155c2dc1baec]

Must be mindful of which transformations are label preserving for a task

Example: iNaturalist dataset

Is color change a good data augmentation?



ring 2023

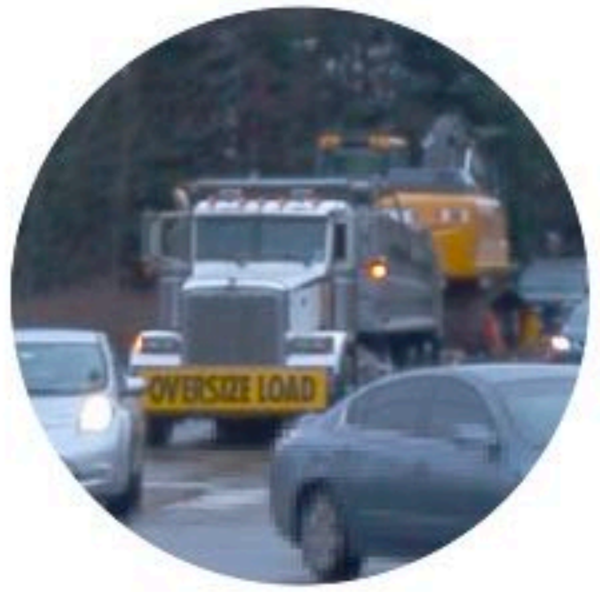
Label transfer via visual similarity

If I know this image contains a cactus, then visually similar images in my unlabeled collection likely also contain a cactus as well.

Saguaro cactus

visually similar

What are good ways to define similar?



"Oversize load"

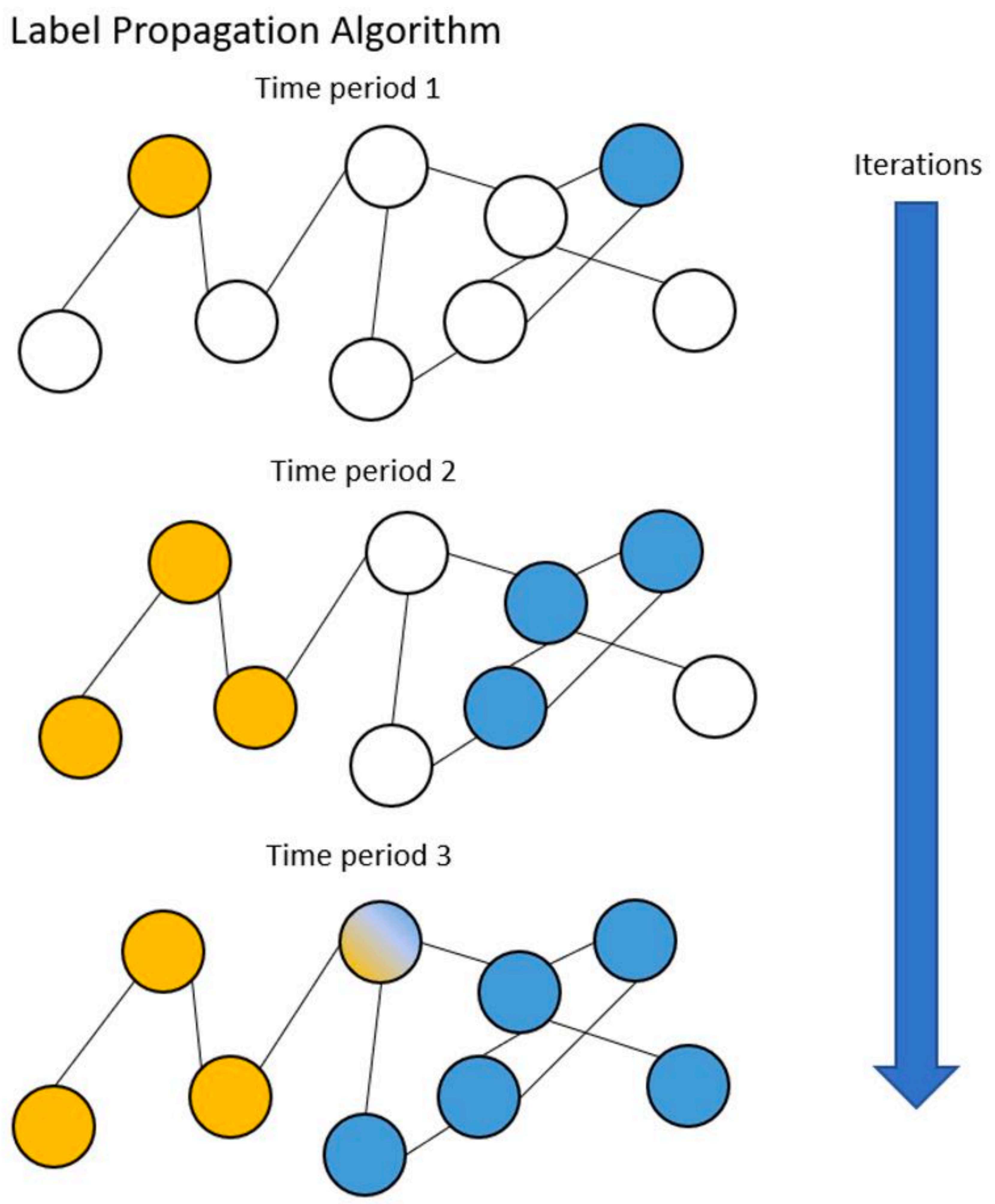
has same text

https://blog.waymo.com/2020/02/content-search.html

Label transfer via label propagation

- Given graph of unlabeled data points
  - e.g., nodes = images, edge weights given by visual similarity

- "Diffuse" sparse labels onto unlabeled nodes



[Image credit: https://www.cylynx.io/blog/efficient-large-graph-label-propagation-algorithm/]

Key idea in all these techniques: bringing in additional priors

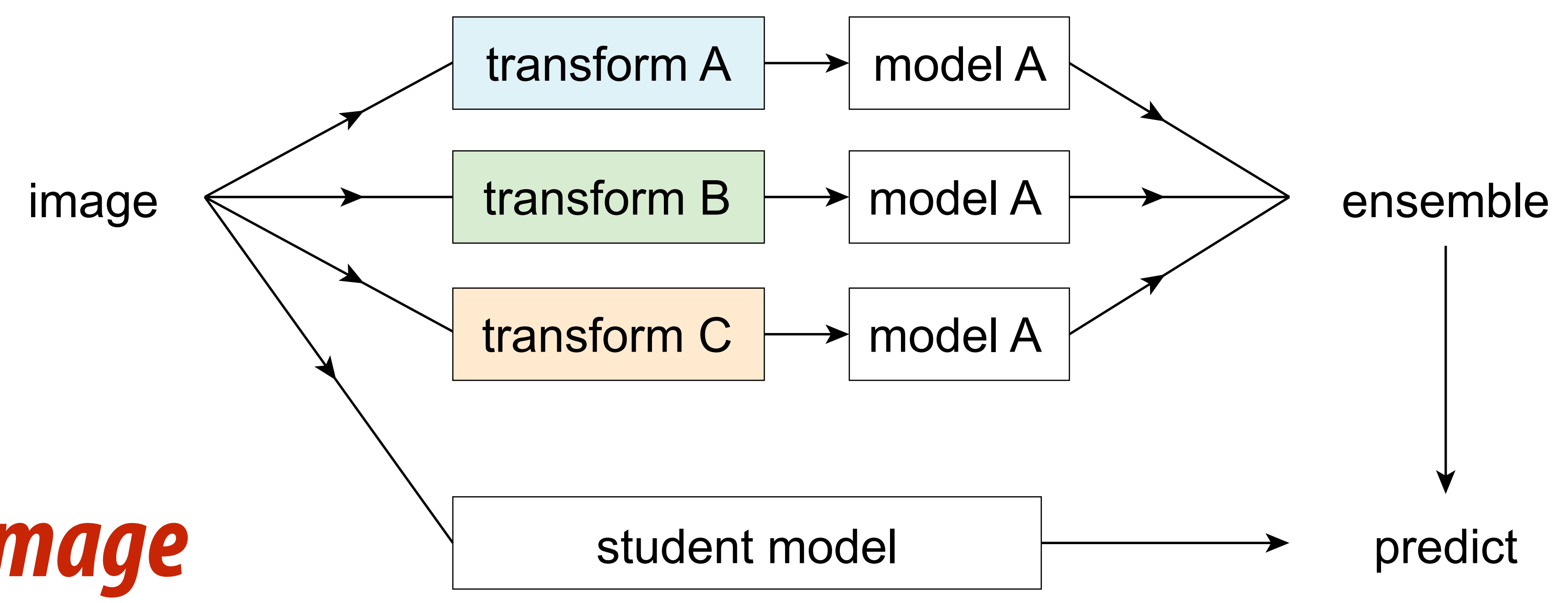
Priors from previous examples:

1. Similar images likely have the same label (knn, label prop, clustering)

2. Certain transformations on data point will not change its label

Using a trained model to supervise itself

- Example: omni-supervised learning
- Train original model using smaller labeled training set
- Evaluate model on different augmentations of unlabeled image
  - *Ensemble model's predictions to estimate "ground truth" label for image*



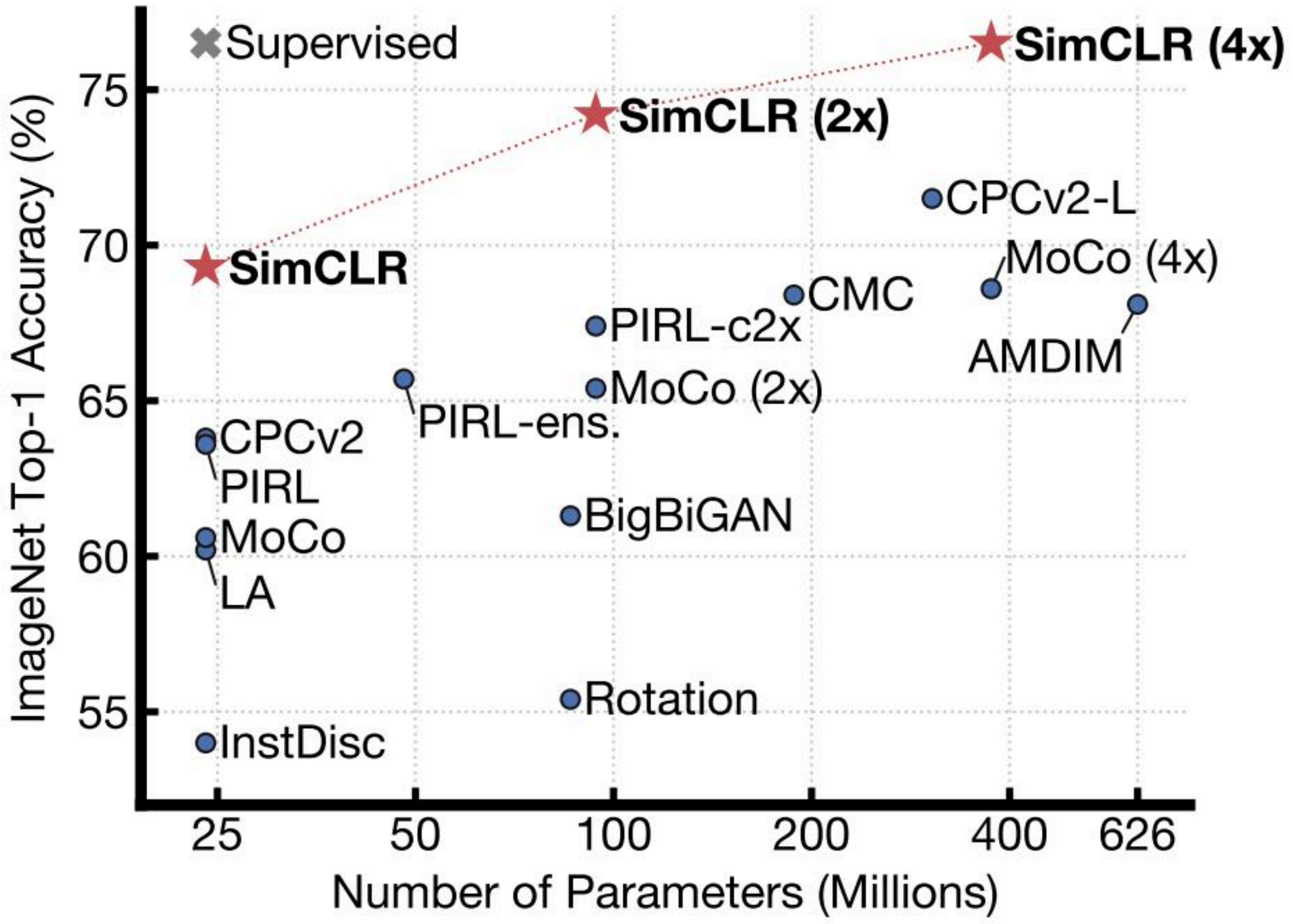
- Re-train model on both labeled images AND estimated labels from ensemble

backbone	DD	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
ResNet-50		37.1	59.1	39.6	20.0	40.0	49.4
ResNet-50	✓	**37.9**	**60.1**	**40.8**	**20.3**	**41.6**	**50.8**
ResNet-101		39.2	61.0	42.3	21.7	42.9	52.3
ResNet-101	✓	**40.1**	**62.1**	**43.5**	**21.7**	**44.3**	**53.7**
ResNeXt-101-32×4		40.1	62.4	43.2	22.6	43.7	53.7
ResNeXt-101-32×4	✓	**41.0**	**63.3**	**44.4**	**22.9**	**45.5**	**54.8**

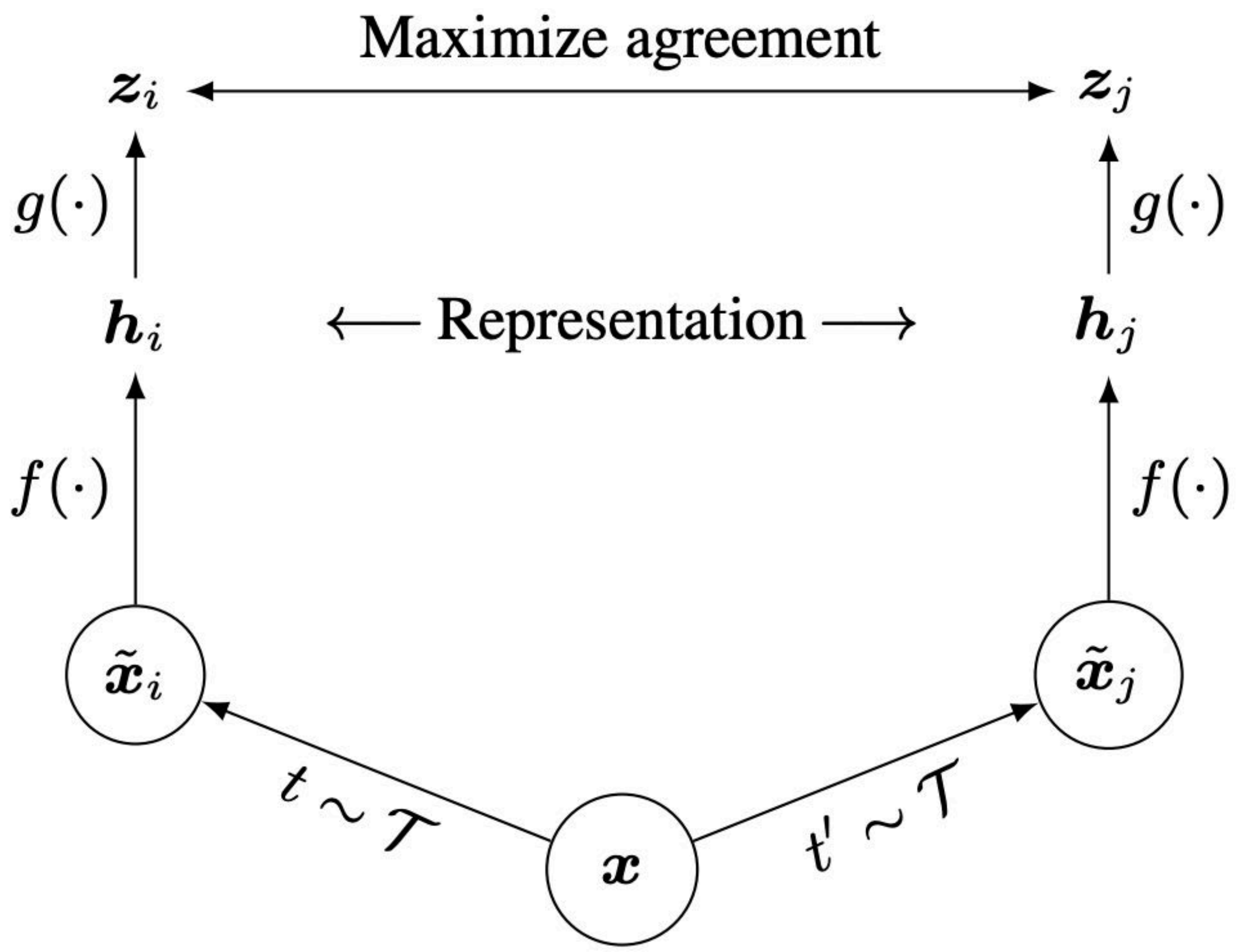
[Source: Radosavovic et al. CVPR 2018]

Modern trend: unsupervised pre-training

- Unsupervised pre-training at scale (using lots of data and using large models) learns good representations
- e.g. SimCLR, based on contrastive loss
- Give training image x, apply augmentation t(x) (crop, resize, flip)



- Train DNN with contrastive loss that encourages projection of different transformations of the same image x to be close (g(f(t(x))) close to g(f(t'(x)))), transformations of different images to be far.



[Image credit:SimCLR paper, Chen et al. NeurIPS 2020]

Providing supervision by writing programs

Encode external priors in programs

- Example: temporal consistency prior: the state of world should not change significantly from frame to frame

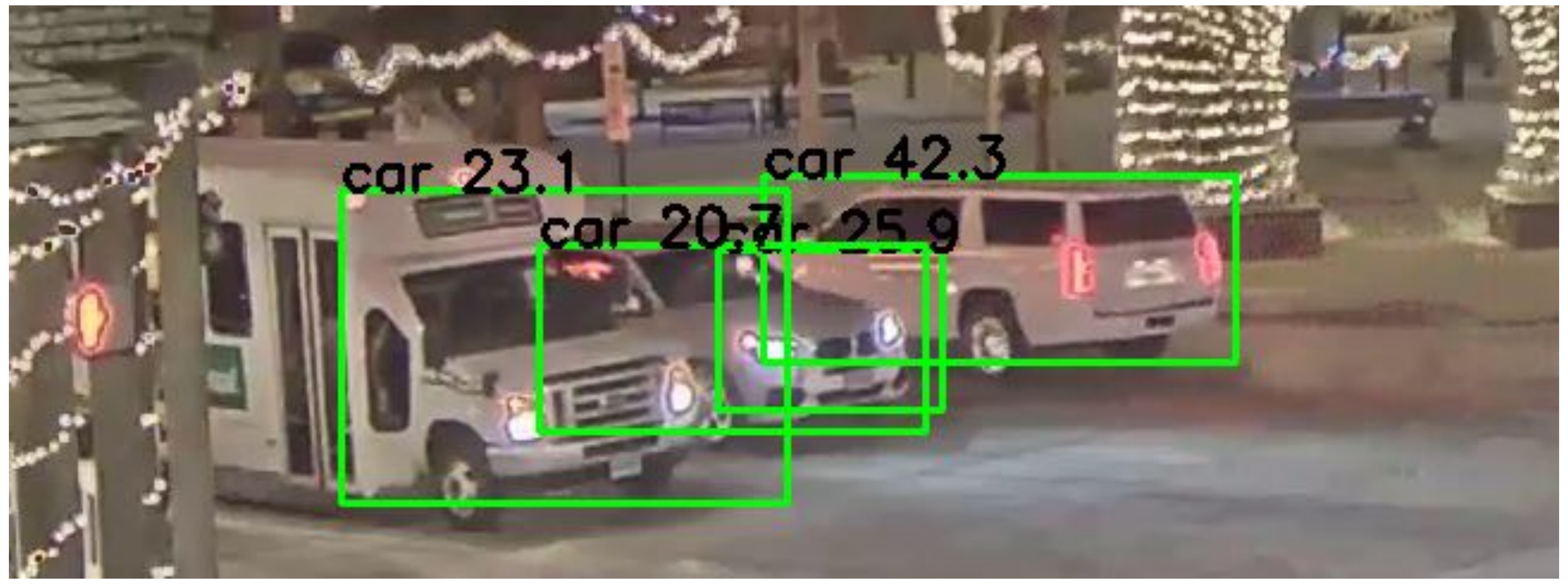
Frame 1

Frame 2

Frame 3

- Example: domain-knowledge prior: objects like cars cannot overlap in space

(a) Example error 1.

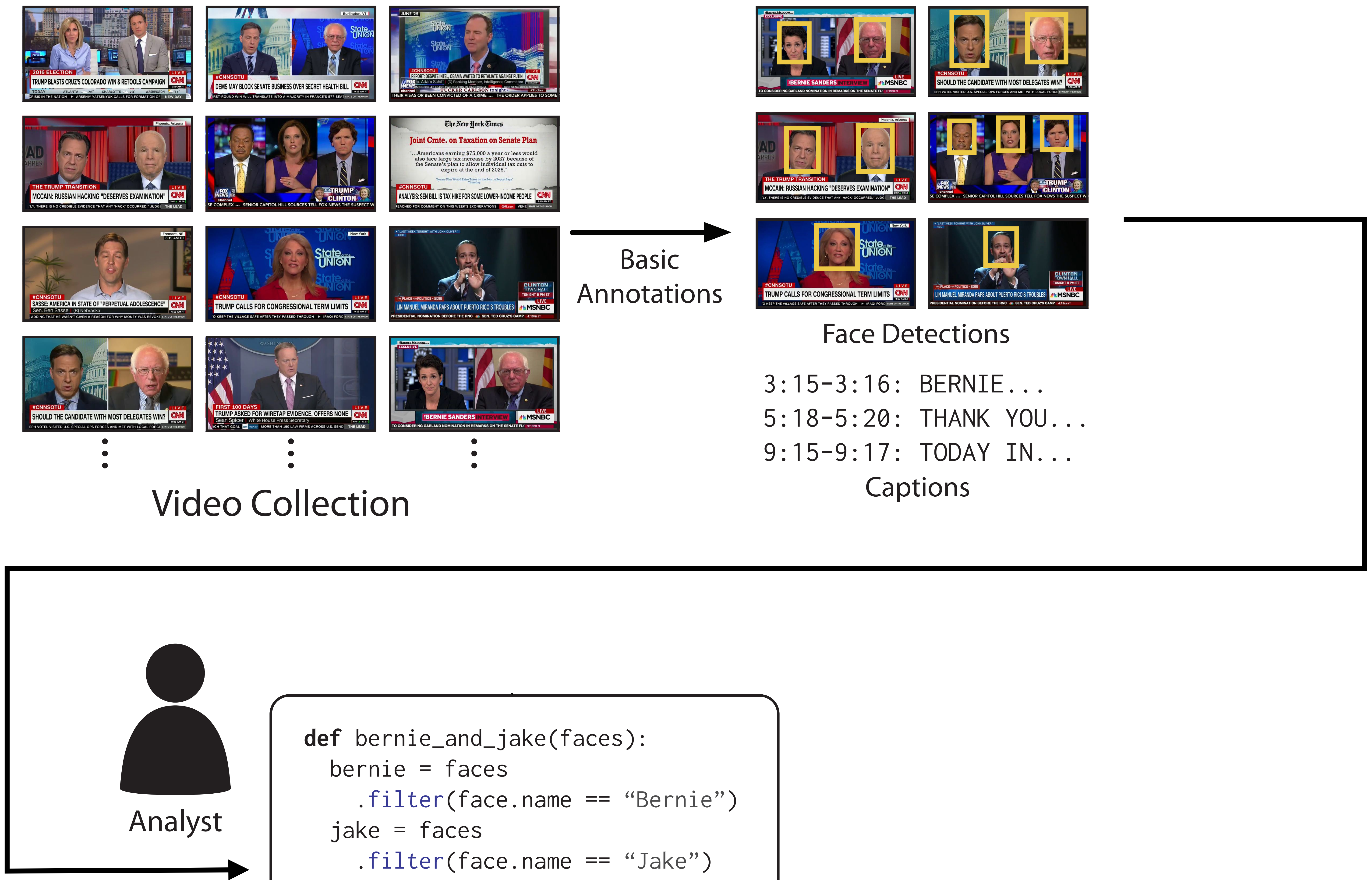


(b) Example error 2.

[Source: Kang et al. MLSys 2020]

DB queries as concept "detectors"

(find elements in database matching this predicate)



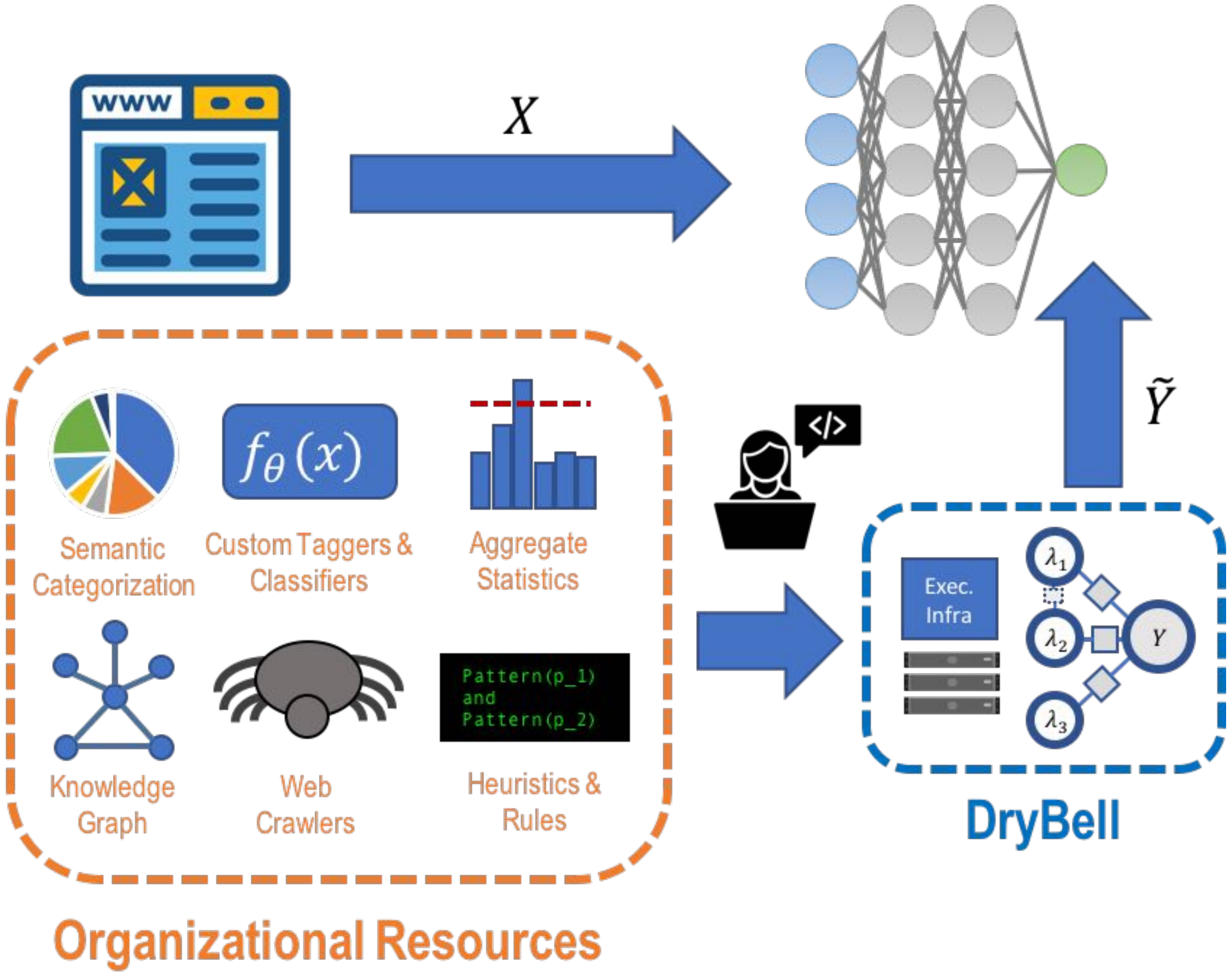

```
def bernie_and_jake(faces):
  bernie = faces
    .filter(face.name == "Bernie")
  jake = faces
    .filter(face.name == "Jake")

  bernie_and_jake = bernie
      .join(jake,
        predicate = time_overlaps,
        merge_op = span)

  return bernie_and_jake
```

[Source: Fu et al. 2019]

Example: three-person panels

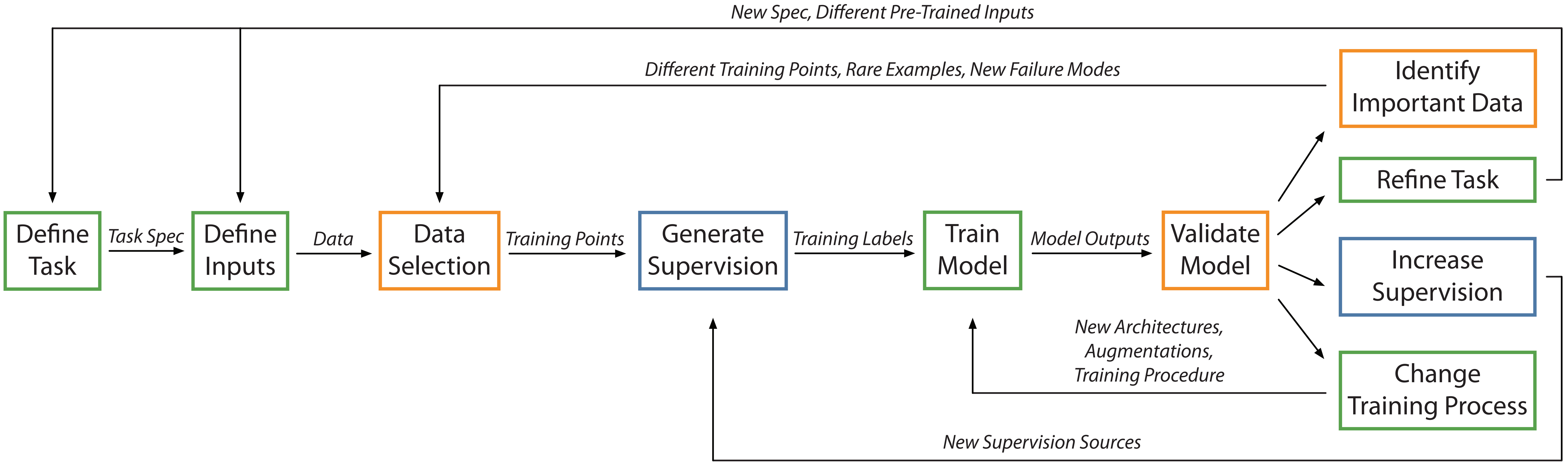
(three faces, bounding boxes greater than 30% of screen height, in horizontal alignment)

Today's discussion:
using weak supervision via "data programming"



A few more thoughts on systems for ML model development

ML model development is an iterative process

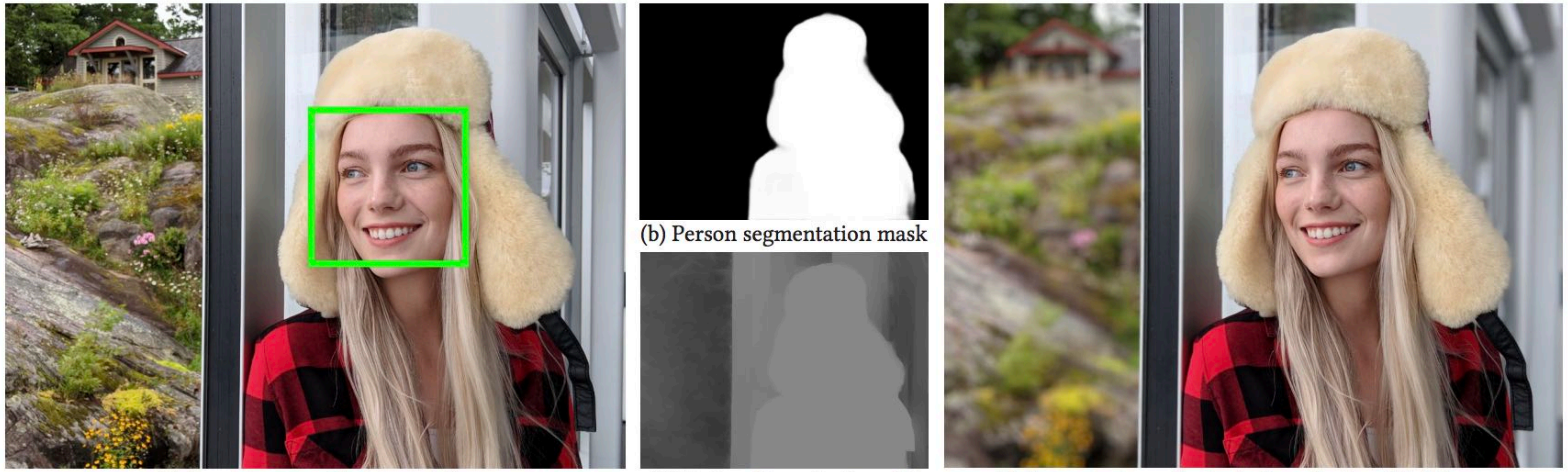


Example: does TensorFlow help with data curation?

"We cannot stress strongly enough the importance of good training data for this segmentation task: choosing a wide enough variety of poses, discarding poor training images, cleaning up inaccurate [ground truth] polygon masks, etc. With each improvement we made over a 9-month period in our training data, we observed the quality of our defocused portraits to improve commensurately."

Synthetic Depth-of-Field with a Single-Camera Mobile Phone

NEAL WADHWA, RAHUL GARG, DAVID E. JACOBS, BRYAN E. FELDMAN, NORI KANAZAWA, ROBERT CARROLL, YAIR MOVSHOVITZ-ATTIAS, JONATHAN T. BARRON, YAEL PRITCH, and MARC LEVOY,
Google Research



(a) Input image with detected face

(b) Person segmentation mask

(c) Mask + disparity from DP

(d) Our output synthetic shallow depth-of-field image

Thought experiment:
I ask you to train a car or person detector for a specific intersection

Suggested "going further" readings

- See Overton (from Apple) and Ludwig (from Uber) papers listed under suggested readings for today's lecture.