Announcements

- Pset4 out, due Thursday, April 10th
- Challenge will be released soon.
- No laptops during class. Feel free to leave now

Last time

- Benefits of CNNs
- Examples of very popular CNNs
  - LeNet
  - CIFAR-10
  - AlexNet
  - VGG-Net
  - ResNet

  Image as input
  - R3D
  - R(2+1)D

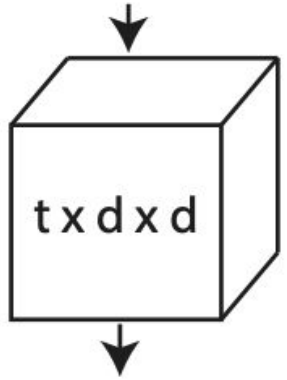
  Video as input

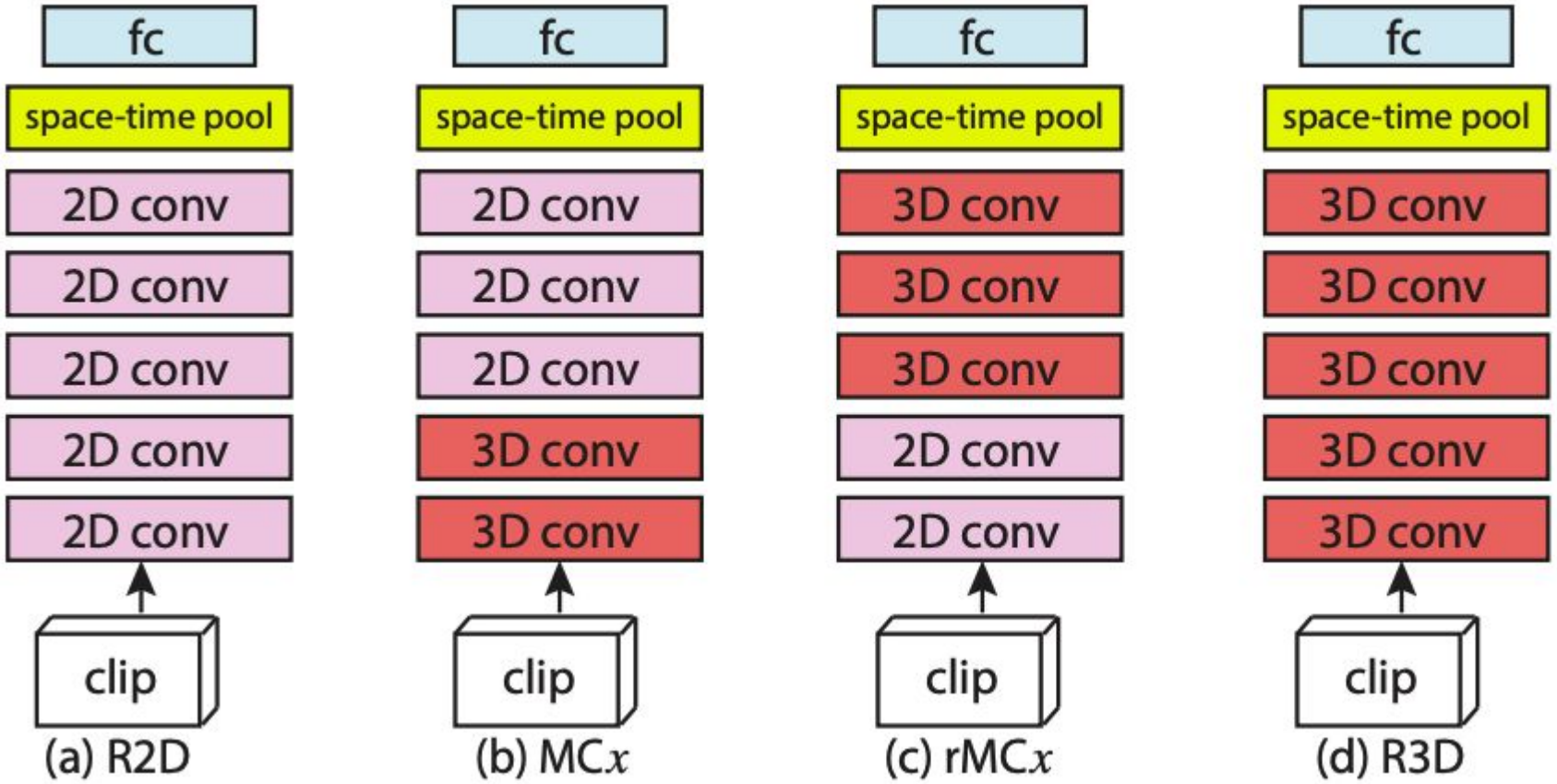
Terminology so far

Convolution Filter (or Kernel)	**M X N**
Channels	Stack of convolutional filters **C**
Feature maps	Outputs after convolution operation applied at a layer
Layers (number of layers is a hyperparam)	Stack of feature maps
Hidden layers	Layers which do not directly interact with input and output
Pooling (nothing to learn!)	Max, mean - across channels or within a feature map

Remember: We are learning all of these during forward and backward pass!

Recall: From 2D -> 3D: CNN for videos





Today

- Some practical tips while training your model.
- Forms of supervision
- Recurrent Neural Network (RNN).
- Transformers

Today

- **Some practical tips while training your model.**
- Forms of supervision
- Recurrent Neural Network (RNN).
- Transformers

Other good things to know

- Check gradients numerically by finite differences

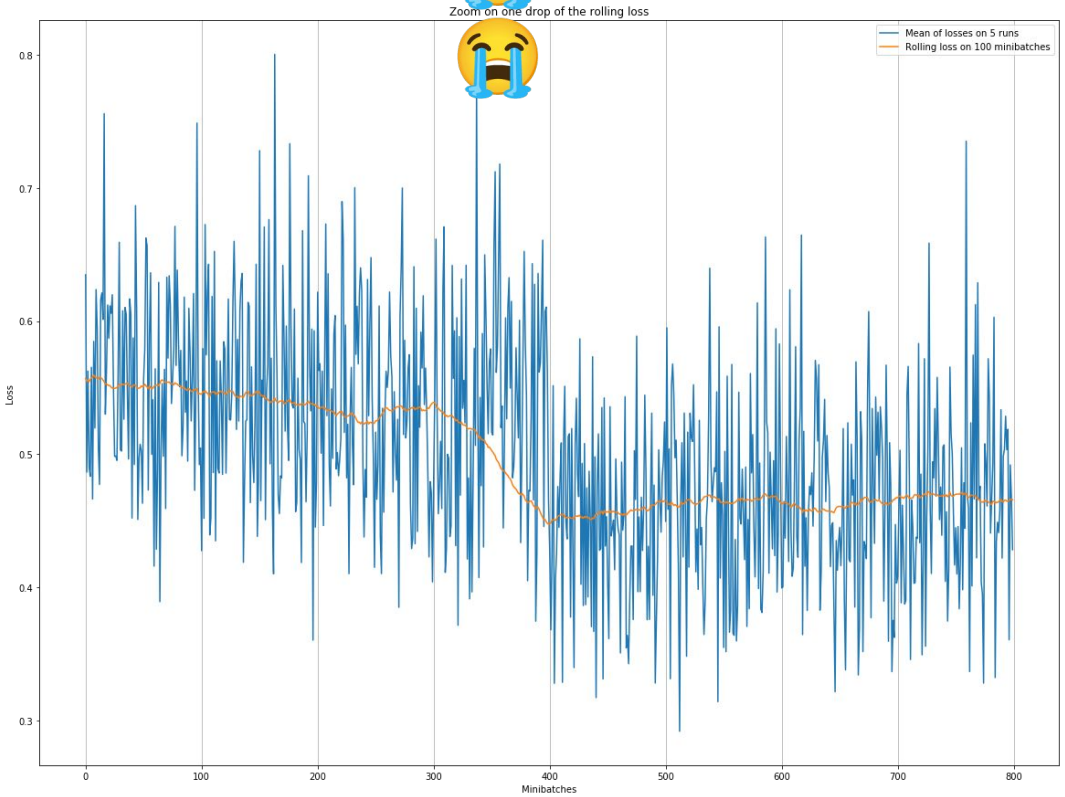
[Derived from slide by Marc'Aurelio Ranzato]

Other good things to know

- Check gradients numerically by finite differences

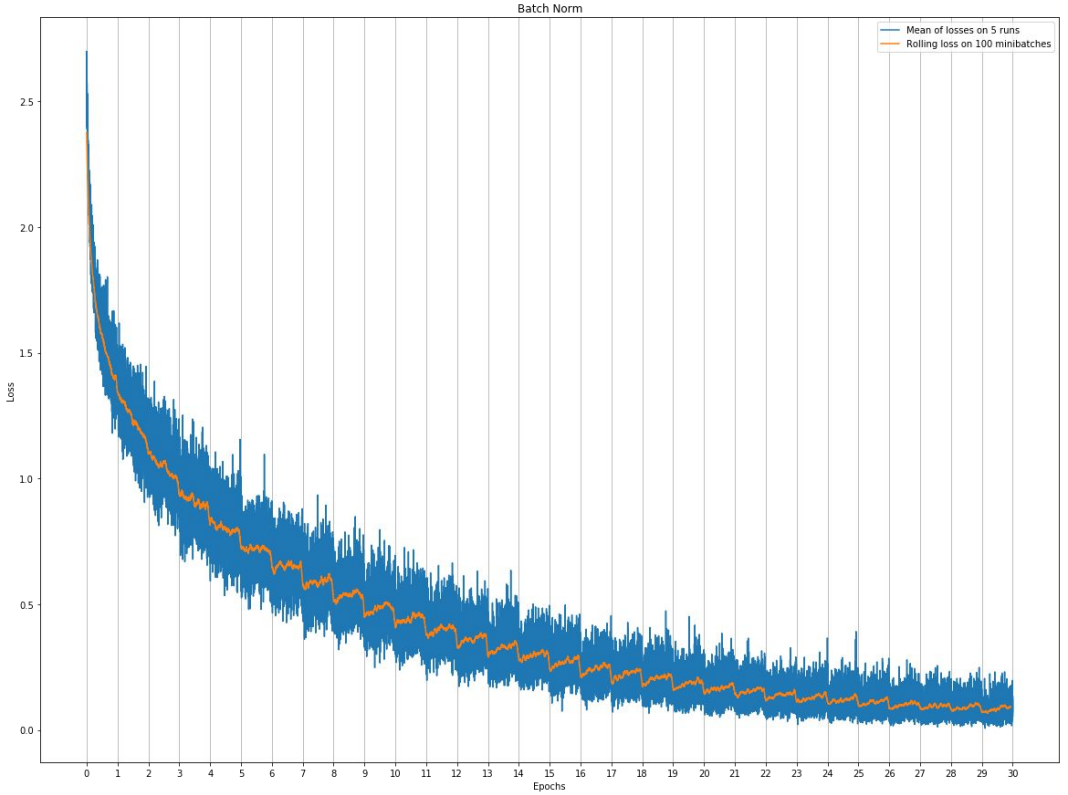
😭

Zoom on one drop of the rolling loss



🤩🤩

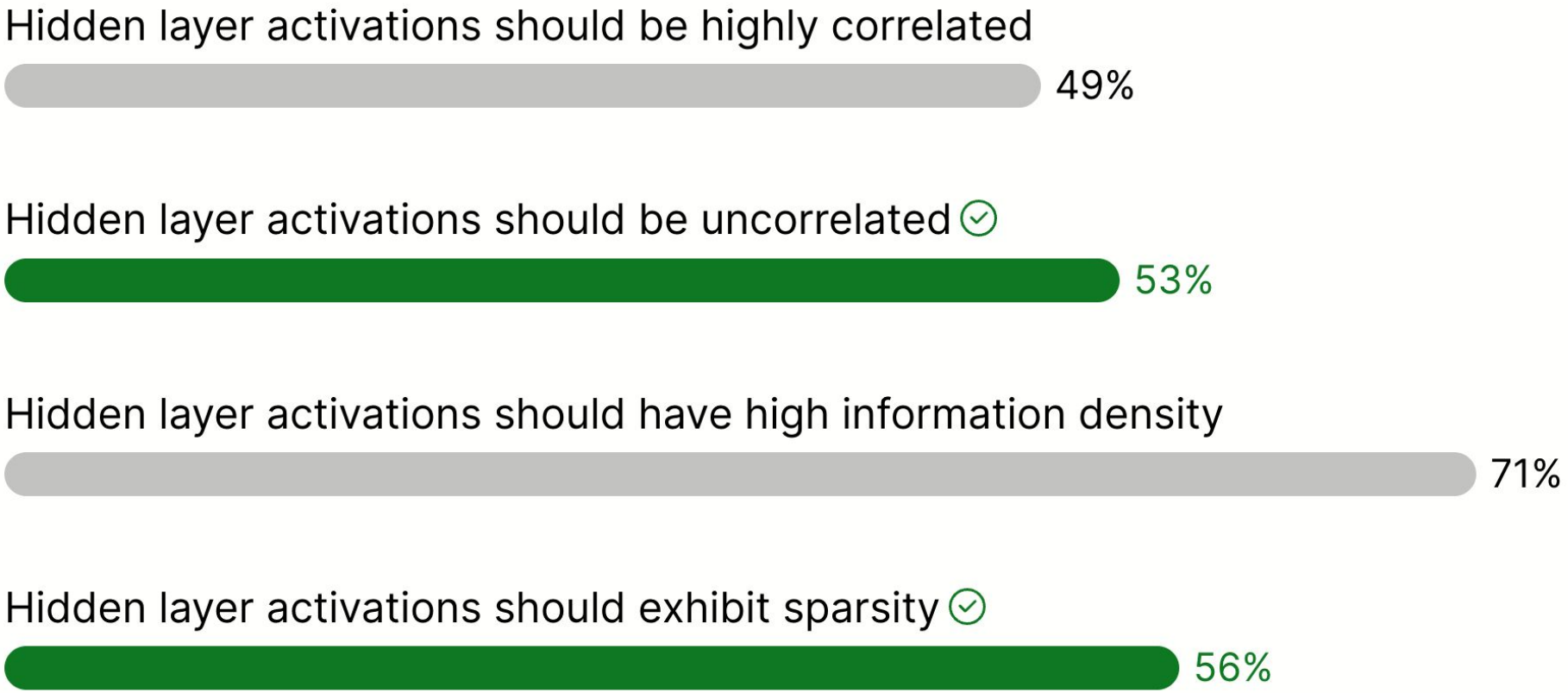
Batch Norm



[Derived from slide by Marc'Aurelio Ranzato]

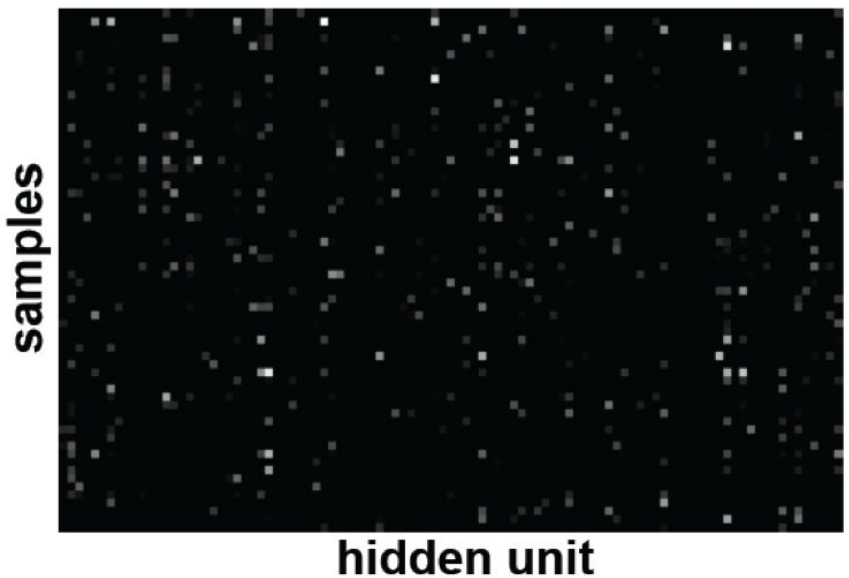
What properties should hidden layer activations exhibit? Select all that apply

What properties should hidden layer activations exhibit? Select all that apply



Other good things to know

- Visualize hidden activations — should be uncorrelated and high variance

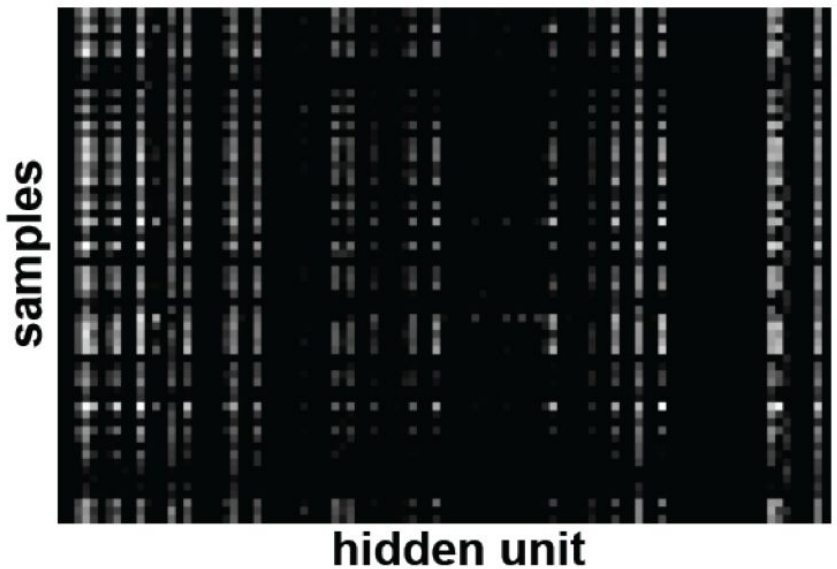


Good training: hidden units are sparse across samples and across features.

[Derived from slide by Marc'Aurelio Ranzato]

Other good things to know

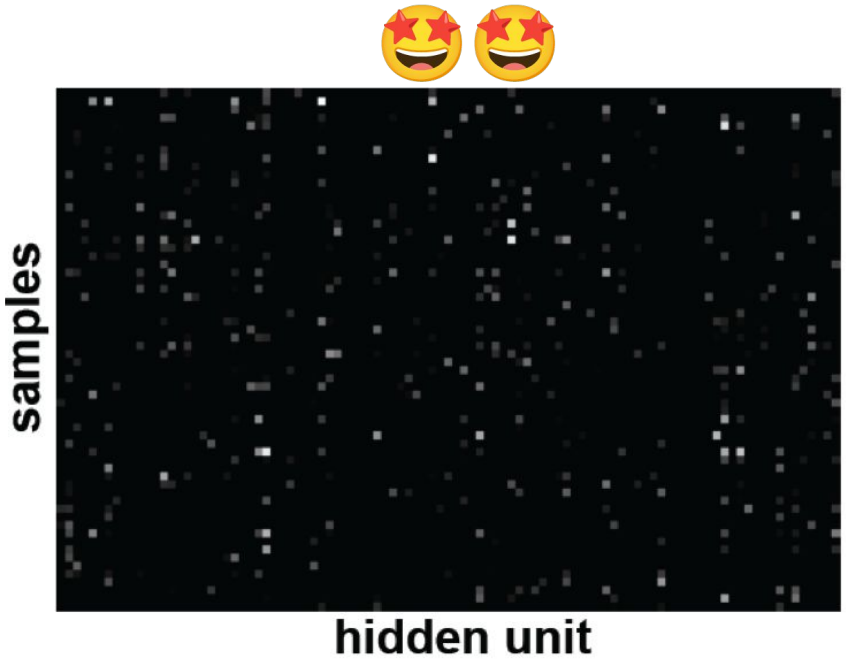
- Check gradients numerically by finite differences
- Visualize hidden activations — should be uncorrelated and high variance

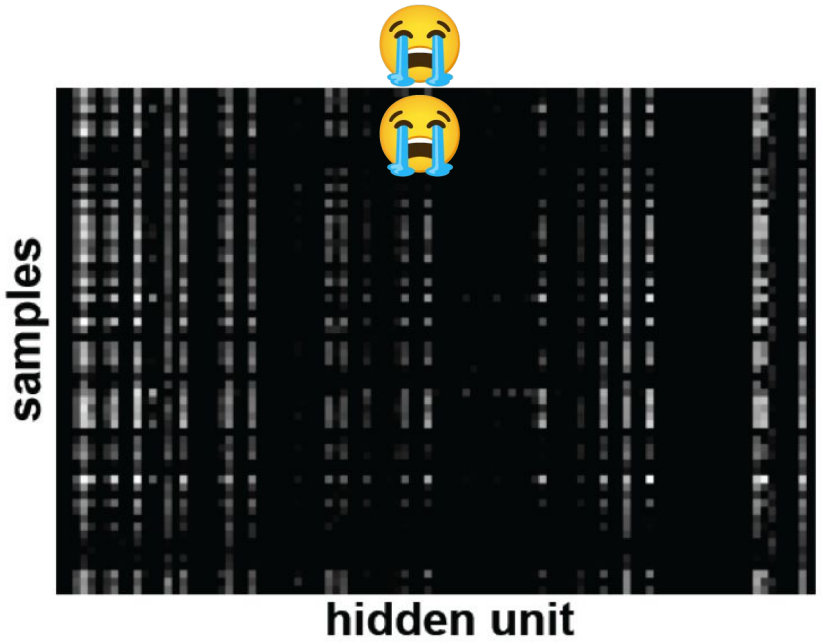


Bad training: many hidden units ignore the input and/or exhibit strong correlations.

[Derived from slide by Marc'Aurelio Ranzato]

Make sure hidden layer activations as **sparse**

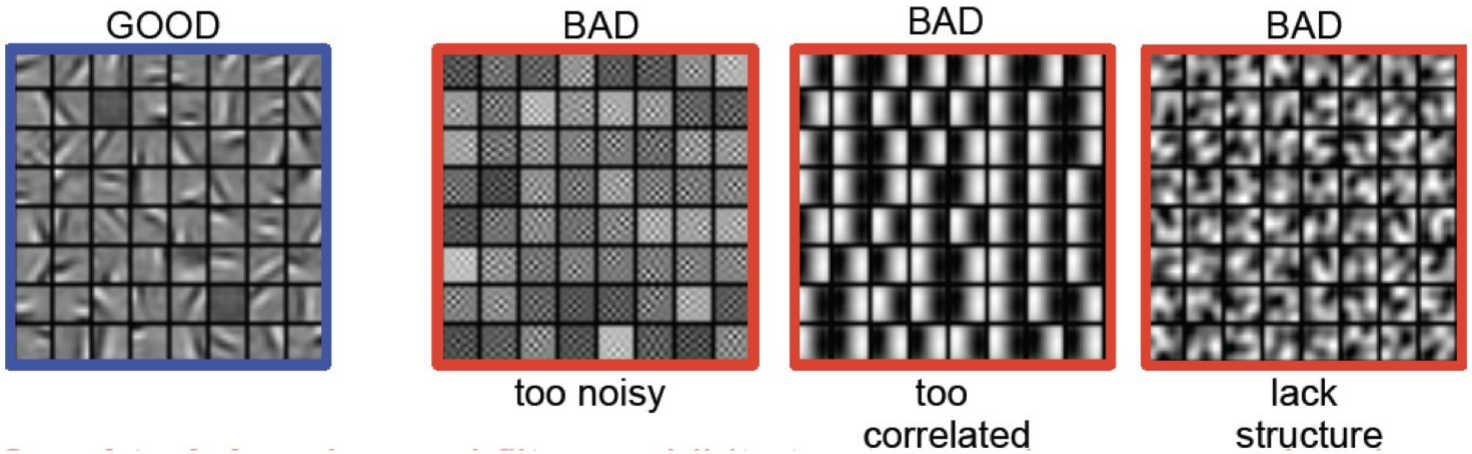




- Uncorrelated hidden activations.
  - Less information redundancy (we love sparsity)
  - Less chance of overfitting
  - Improved training

Visualize convolutional filters in each layer

- Visualize filters



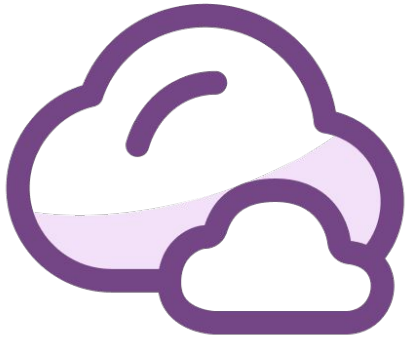
Good training: learned filters exhibit structure and are uncorrelated.

[Derived from slide by Marc'Aurelio Ranzato]

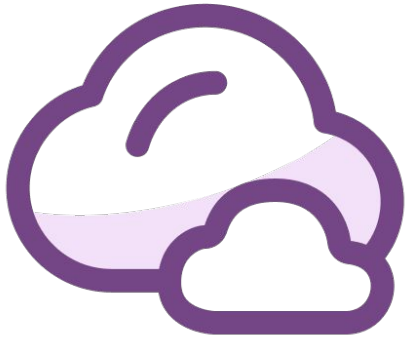
Summary: Debugging tips

- Check gradients numerically by finite differences
- Visualize hidden activations — should be uncorrelated and high variance
- Visualize filters
- **Measure error on both training and validation set**
  - **check the error / loss → 0.**

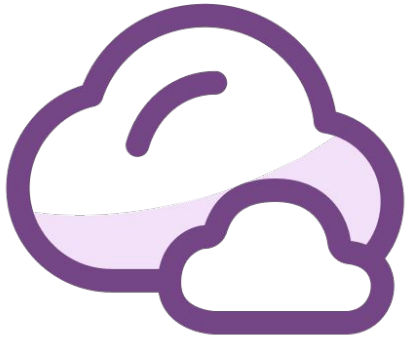
Answers to the next set of 4 questions follow



What action would you take if the training diverges?



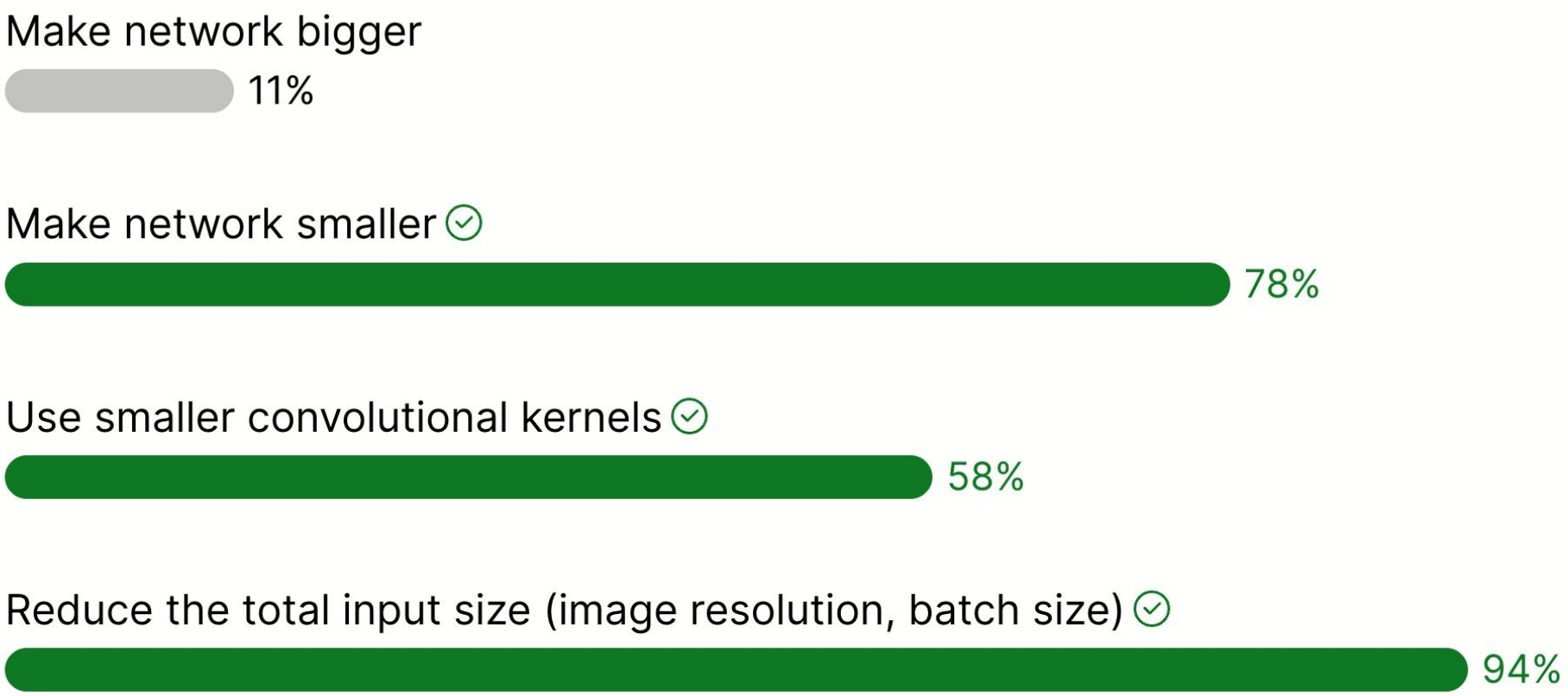
What action would you take if the loss is minimized but the training accuracy is very low



What action would you take if the network is underperforming?

What action would you take if the network training is too slow?

What action would you take if the network training is too slow?



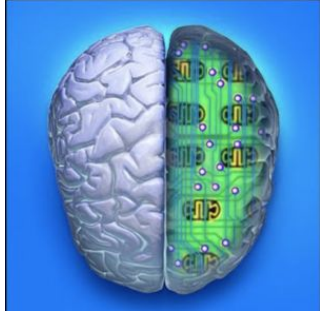
Summary: Debugging tips

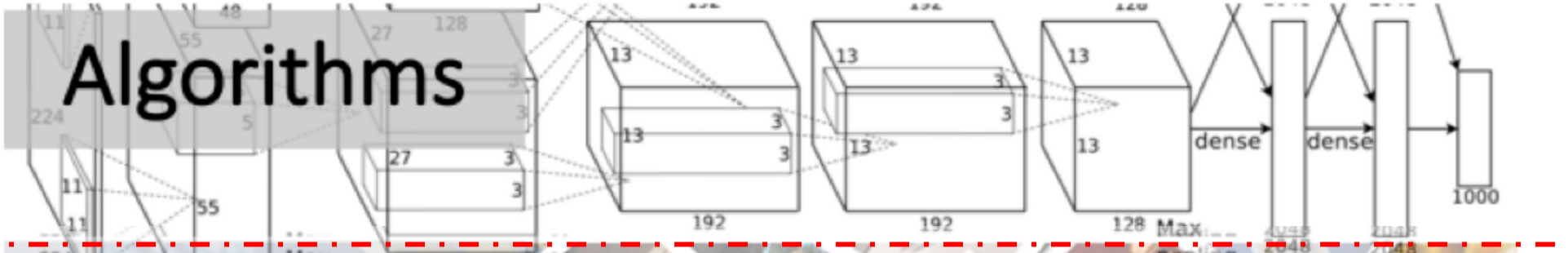
- Training diverges:
  - Learning rate may be too large → decrease learning rate
  - BPROP is buggy → numerical gradient checking
- Parameters collapse / loss is minimized but accuracy is low
  - Check loss function: Is it appropriate for the task you want to solve?
  - Does it have degenerate solutions?
- Network is underperforming
  - Compute flops and nr. params. → if too small, make net larger
  - Visualize hidden units/params → fix optimization
- Network is too slow
  - GPU,distrib. framework, make net smaller

Today

- Some practical tips while training your model.
- **Forms of supervision**
- Recurrent Neural Network (RNN).
- Transformers

What helped us get here?



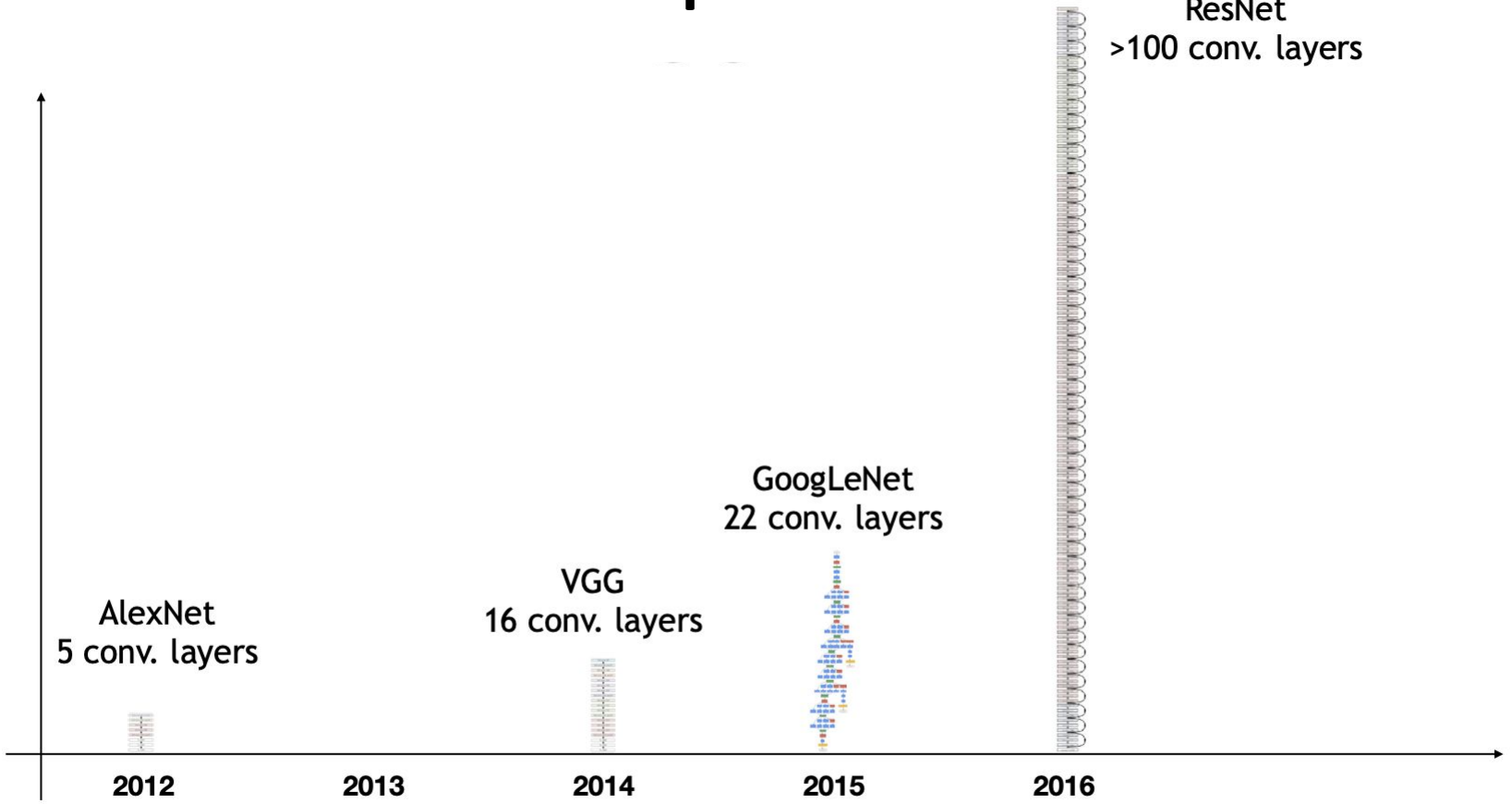


- **Convolution networks**
- RNNs
- Transformers
- Diffusion models

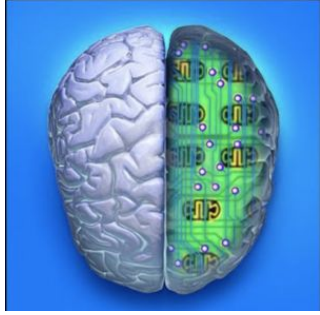
- ImageNet
- OpenImages
- LAION-5B
- Ego4D

Slide credit: Justin Johnson

Evolution of deeper models



What helped us get here?



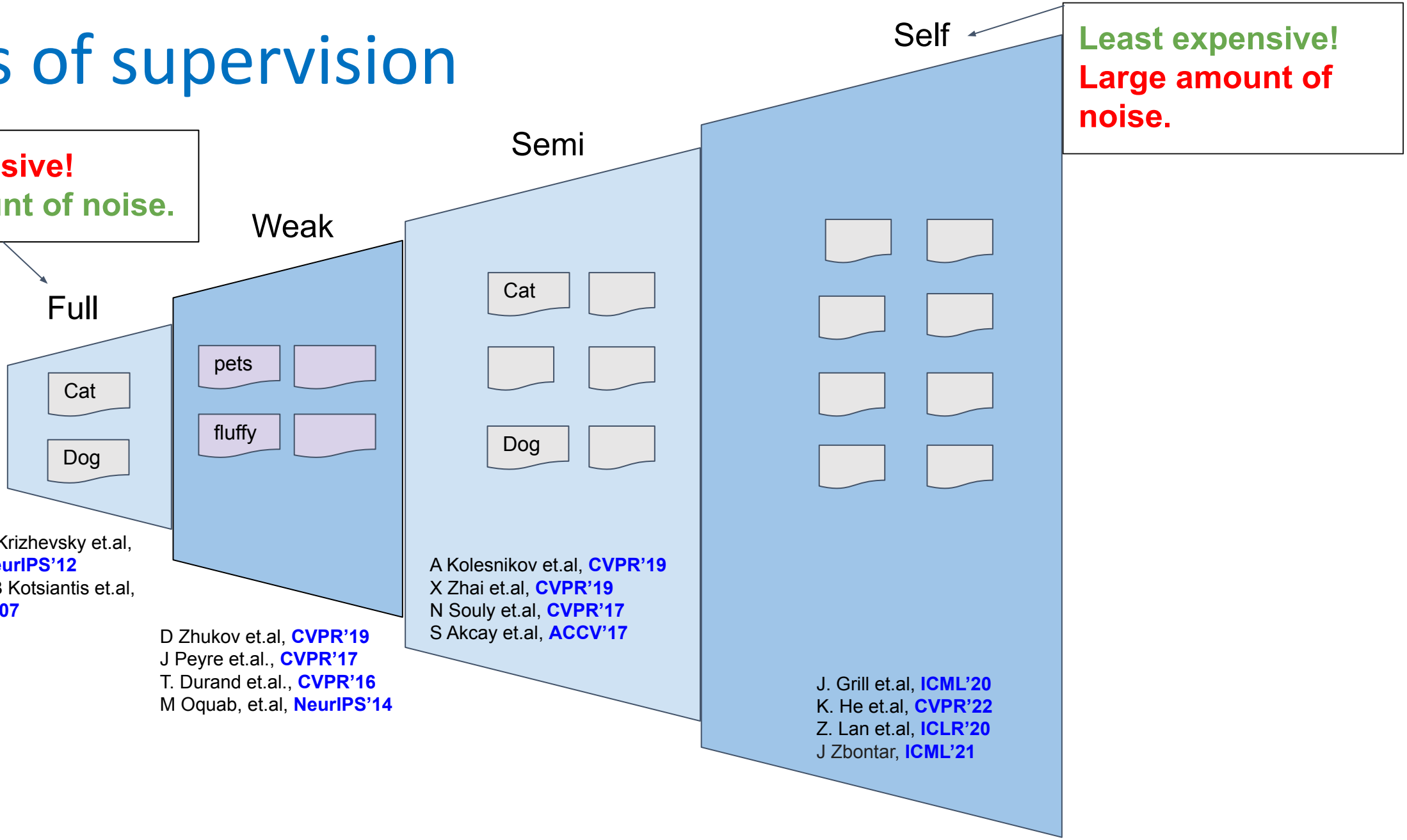
- ImageNet
- OpenImages
- LAION-5B
- Ego4D

Slide credit: Justin Johnson

Types of supervision

Most expensive!
Least amount of noise.

Least expensive!
Large amount of noise.



A.Krizhevsky et.al, **NeurIPS'12**
SB Kotsiantis et.al, **2007**

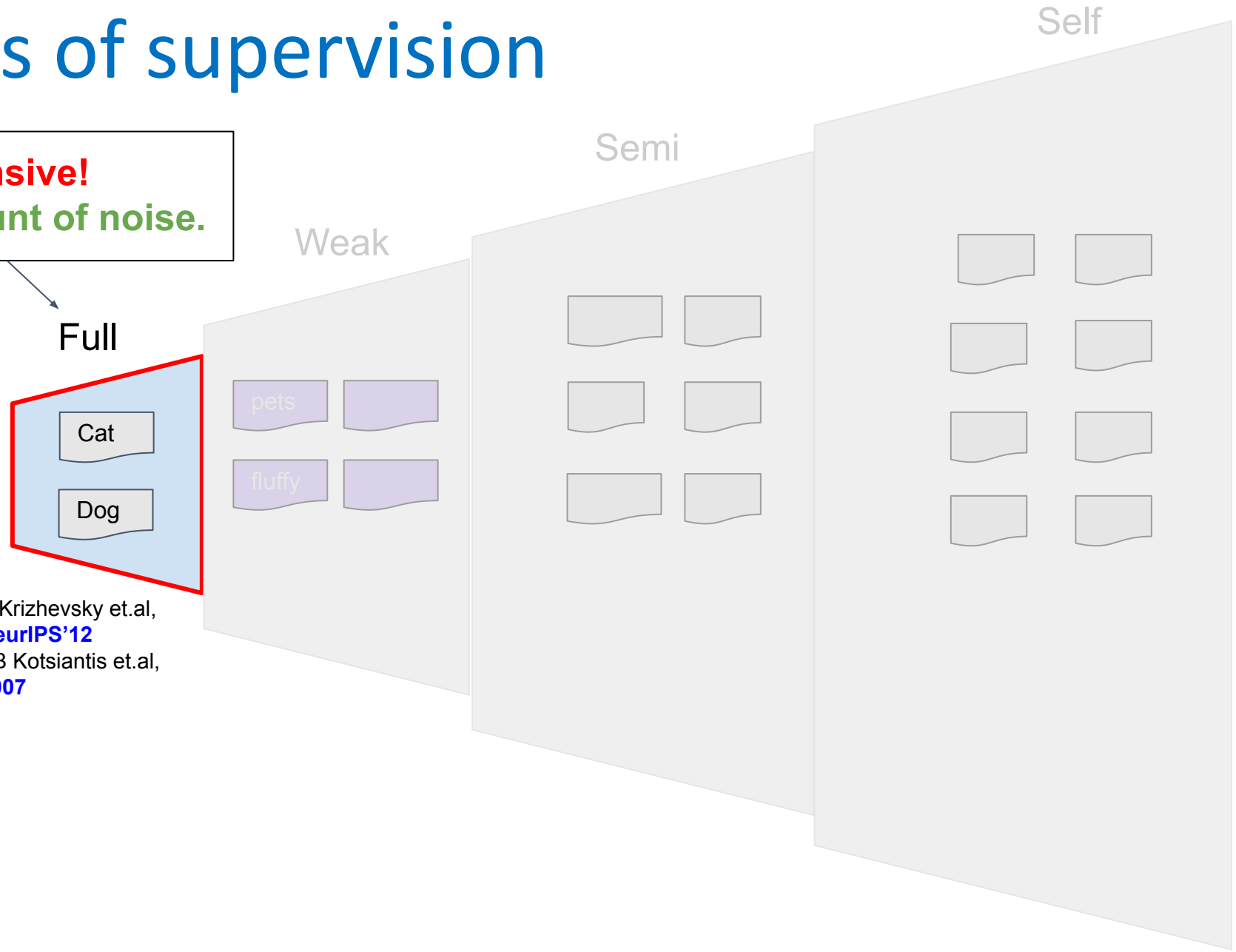
D Zhukov et.al, **CVPR'19**
J Peyre et.al., **CVPR'17**
T. Durand et.al., **CVPR'16**
M Oquab, et.al, **NeurIPS'14**

A Kolesnikov et.al, **CVPR'19**
X Zhai et.al, **CVPR'19**
N Souly et.al, **CVPR'17**
S Akcay et.al, **ACCV'17**

J. Grill et.al, **ICML'20**
K. He et.al, **CVPR'22**
Z. Lan et.al, **ICLR'20**
J Zbontar, **ICML'21**

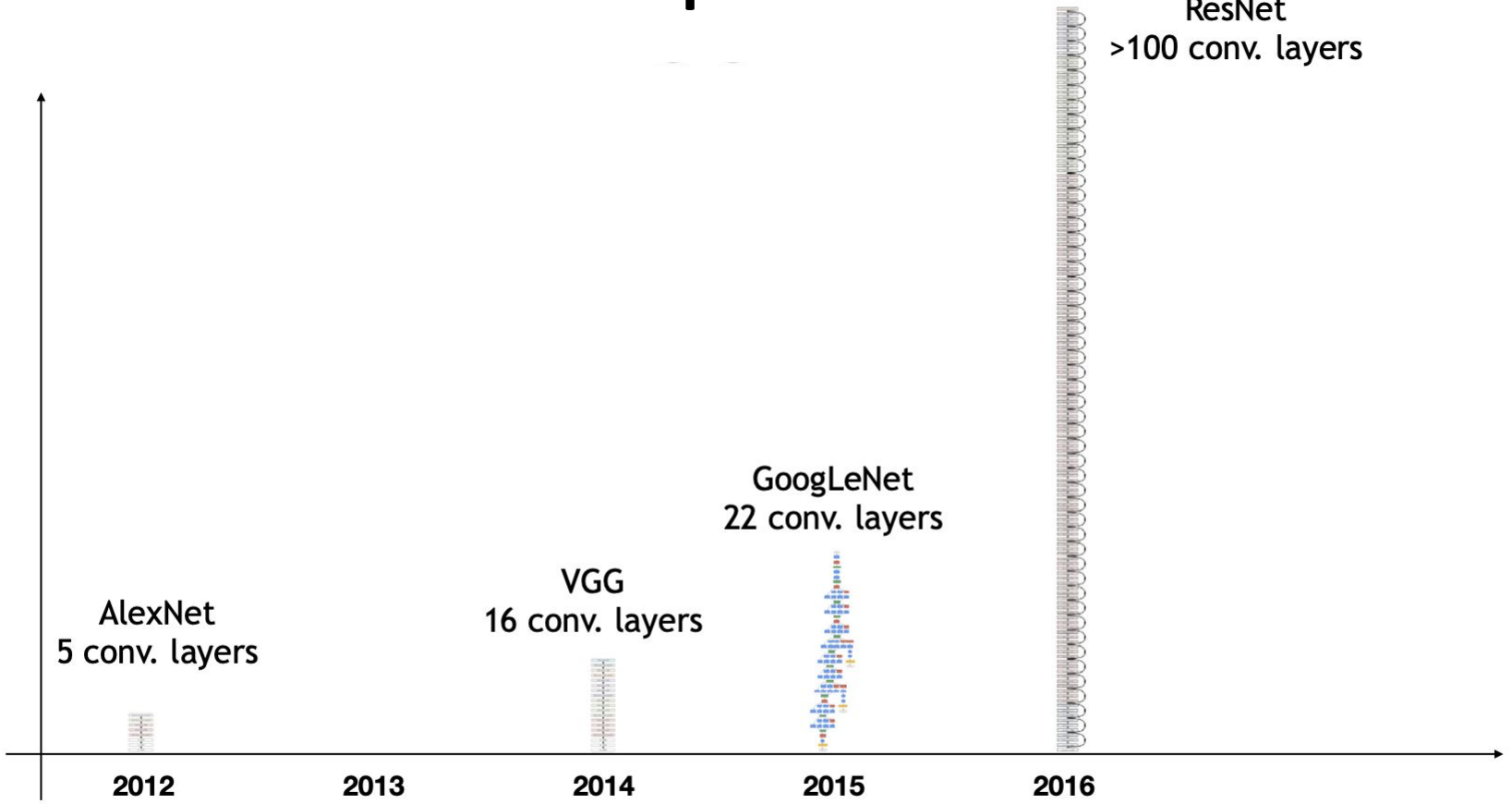
Levels of supervision

Most expensive!
Least amount of noise.



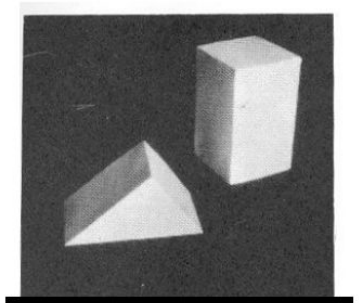
A.Krizhevsky et.al,
NeurIPS'12
SB Kotsiantis et.al,
2007

Evolution of deeper models



All involved fully supervised fine tuning

Evolution of vision datasets

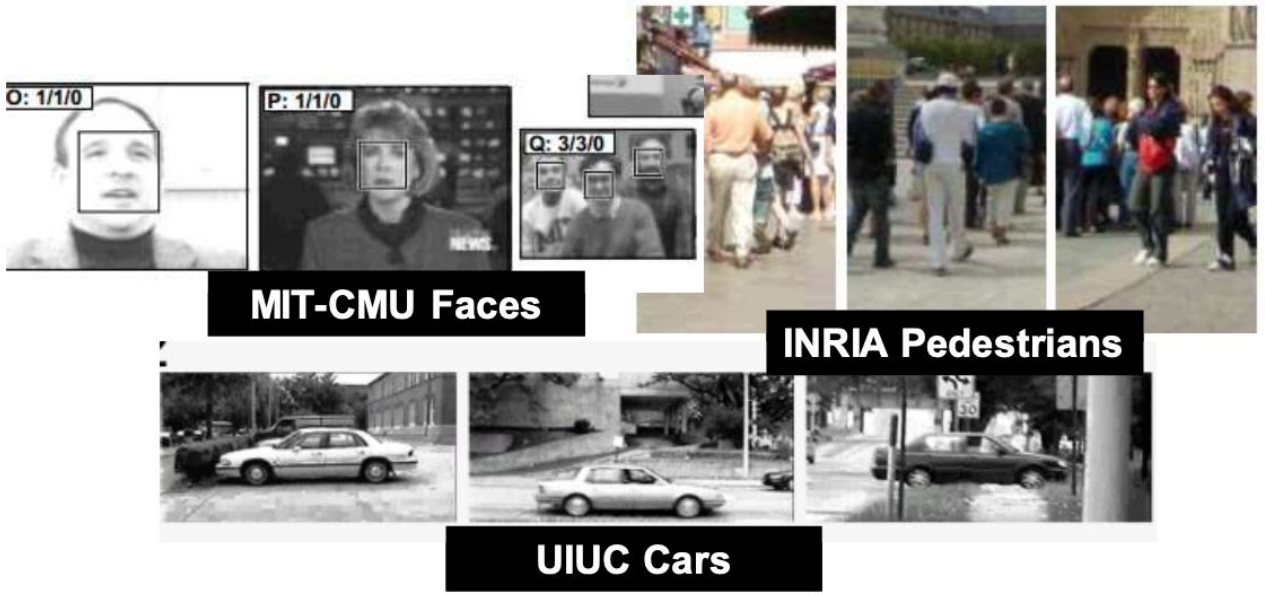


Roberts 1963

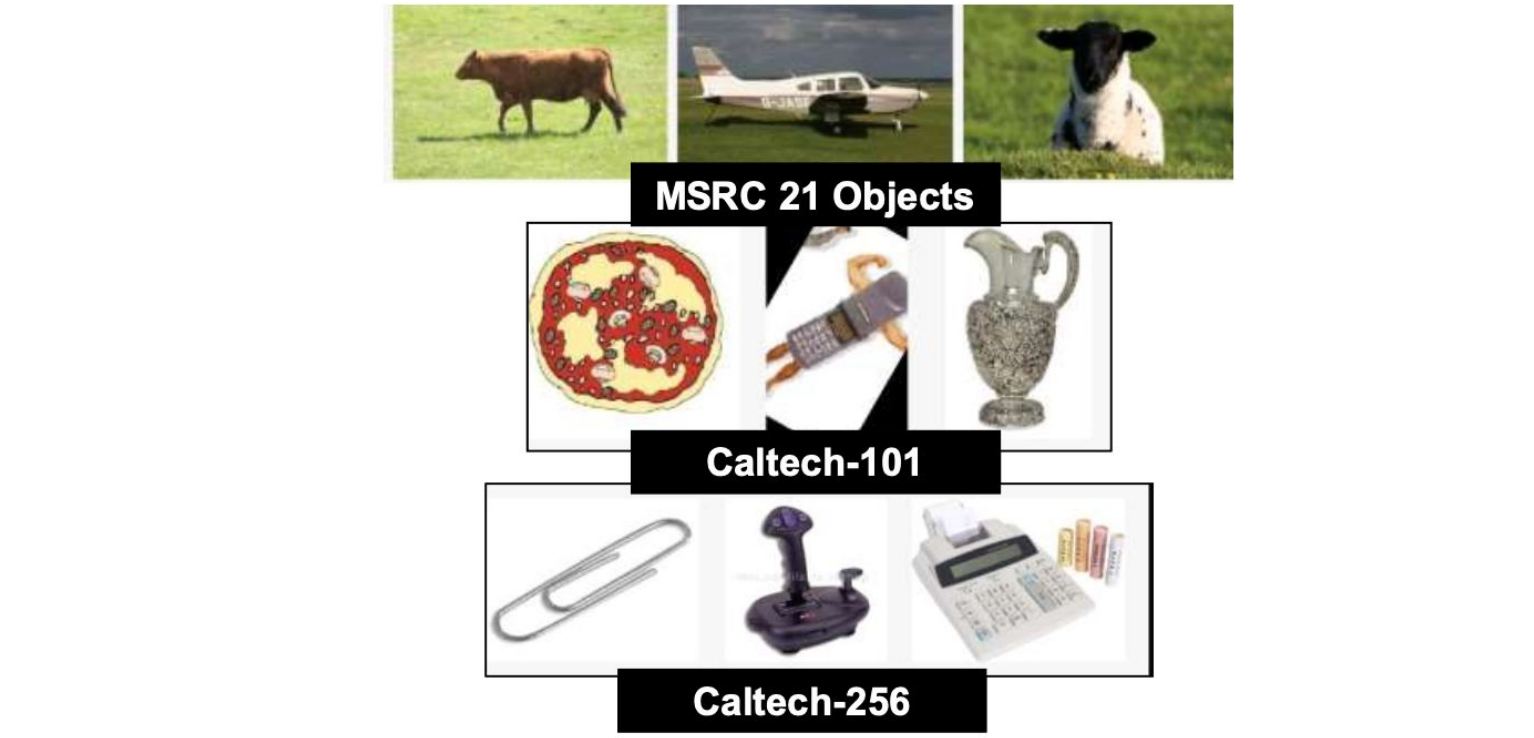
COIL

1963 ··· 1996

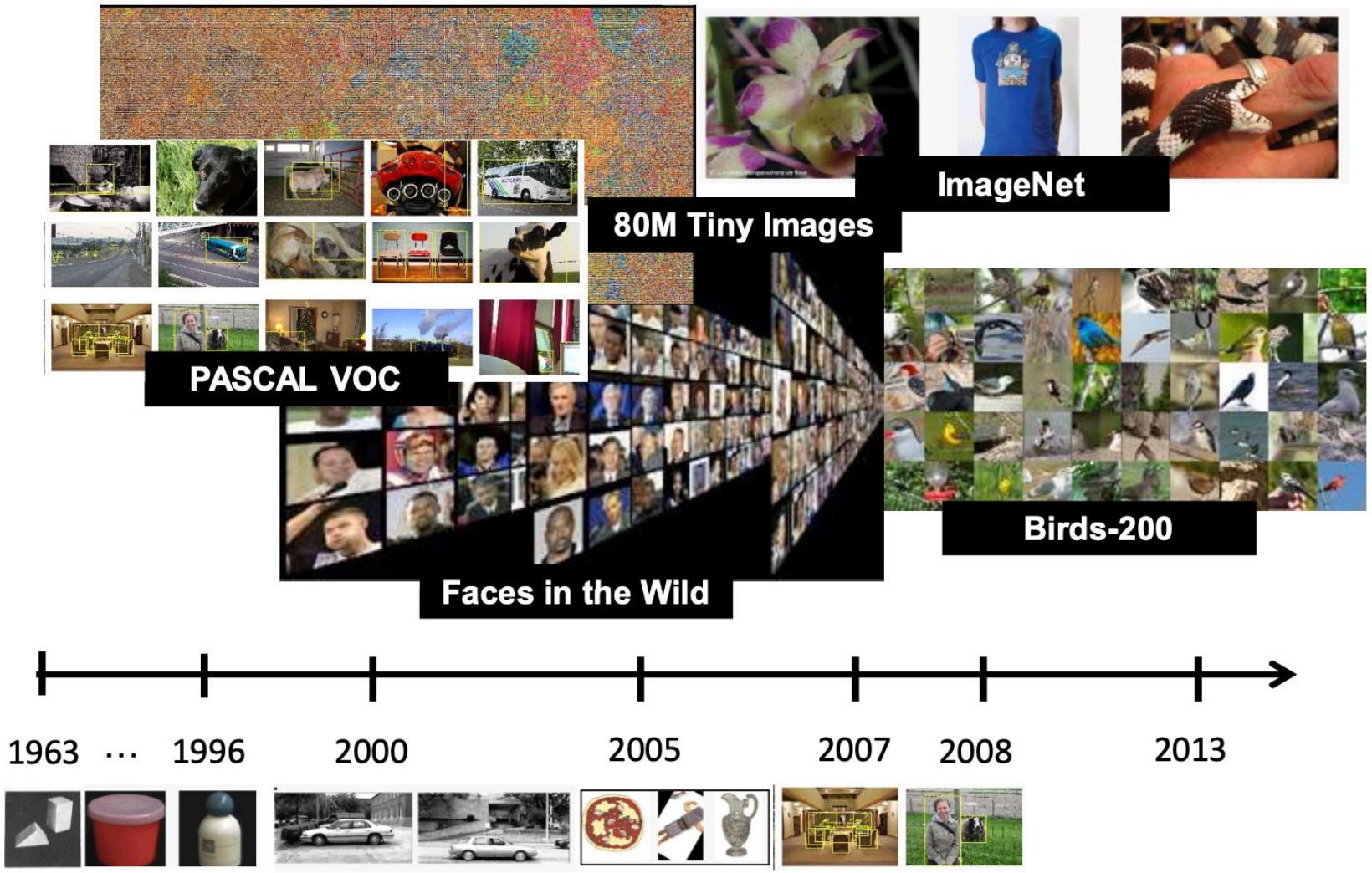
Evolution of vision datasets



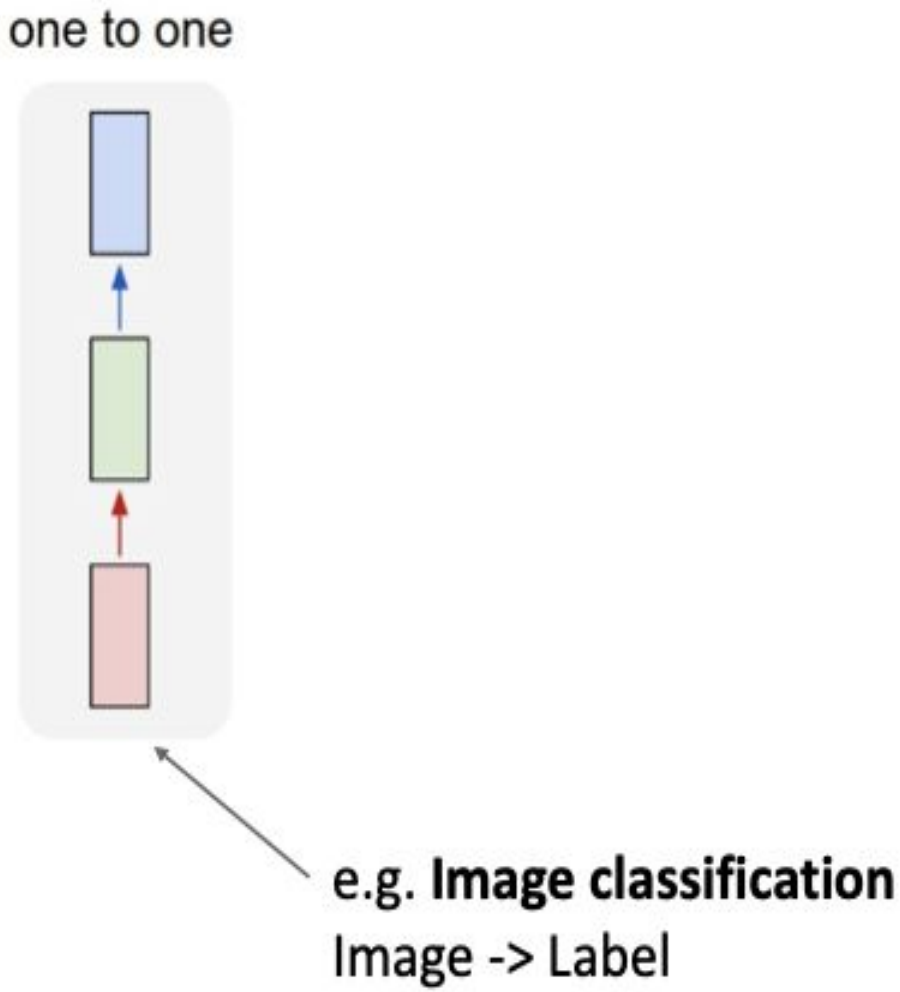
Evolution of vision datasets



Evolution of vision datasets

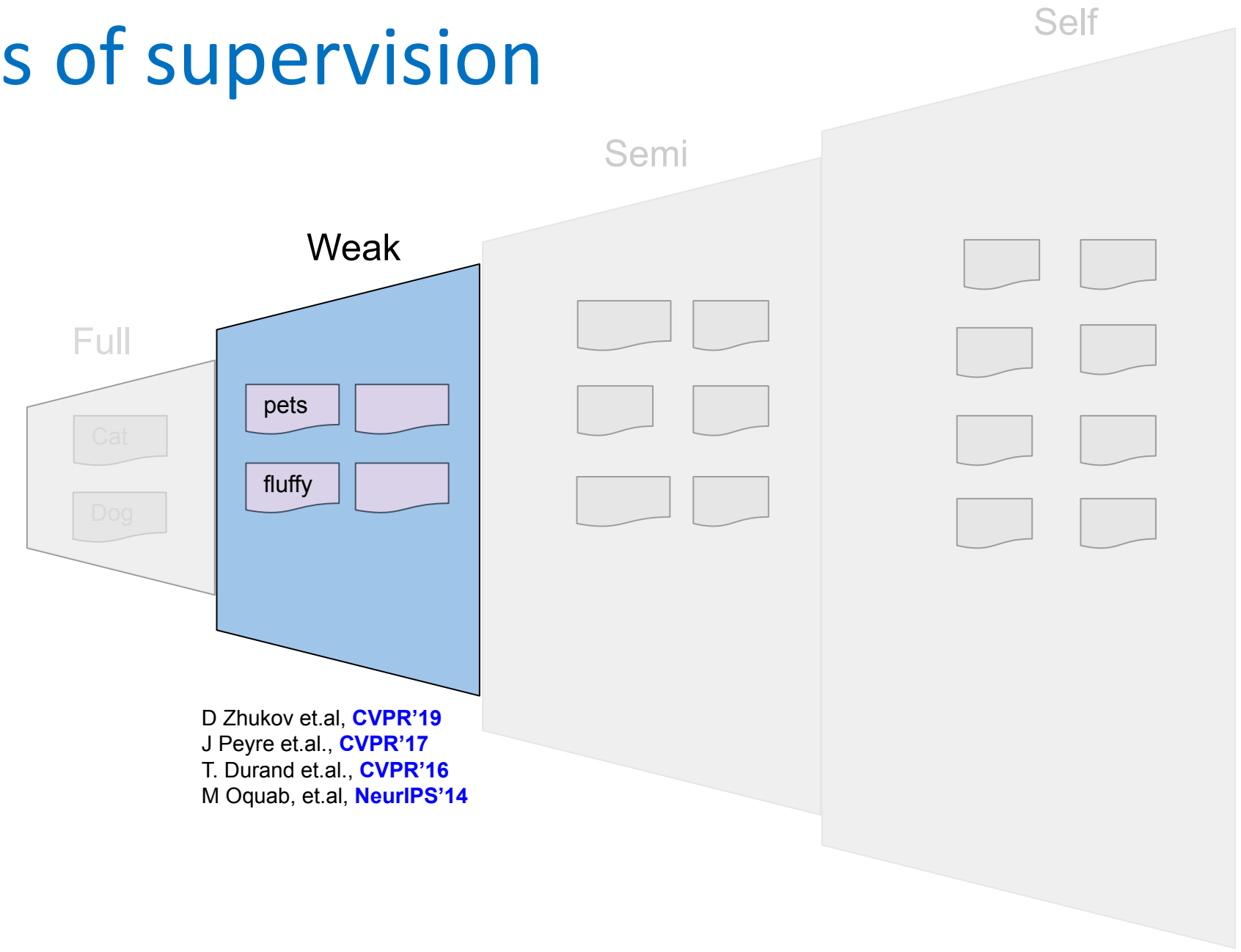


So far: Feed-forward neural networks



Slide credit: Justin Johnson

Types of supervision



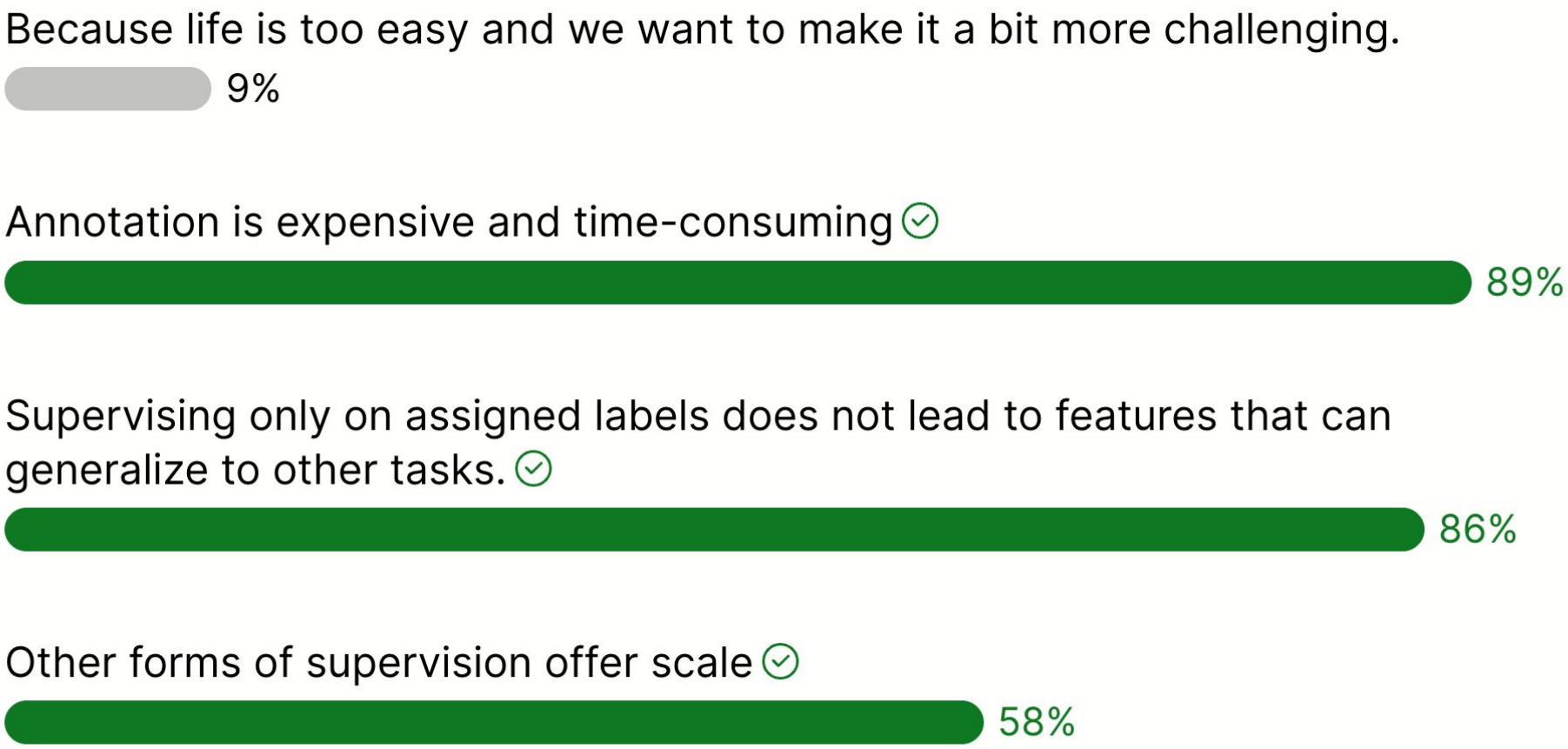
D Zhukov et.al, **CVPR'19**
J Peyre et.al., **CVPR'17**
T. Durand et.al., **CVPR'16**
M Oquab, et.al, **NeurIPS'14**

Full v.s weak supervision

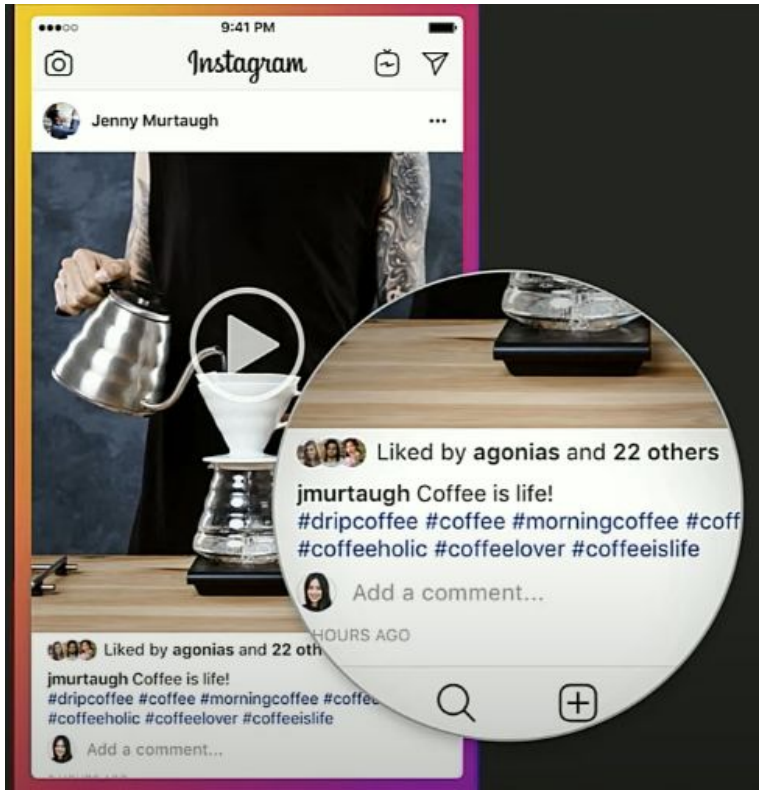
Task	Full	Weak
Classification	(image, labels)	(image, missing / noisy labels)
Detection	(image, detection boxes)	(image, object names)
Segmentation	(image, segmentation masks)	(image, object names) (image, object detection boxes)

Why should we explore weak, semi, and self-supervision?

Why should we explore weak, semi, and self-supervision?

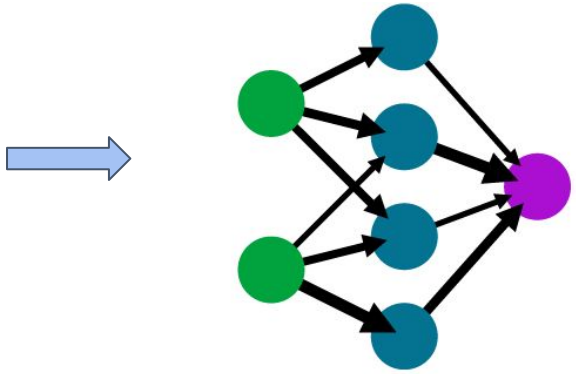


Weak supervision: An application



- Noisy labels, eg: use hashtags instead of clean, annotated labels.

Task: Video Action Classification



Cartwheeling

Applications of video action classification

Video content moderation

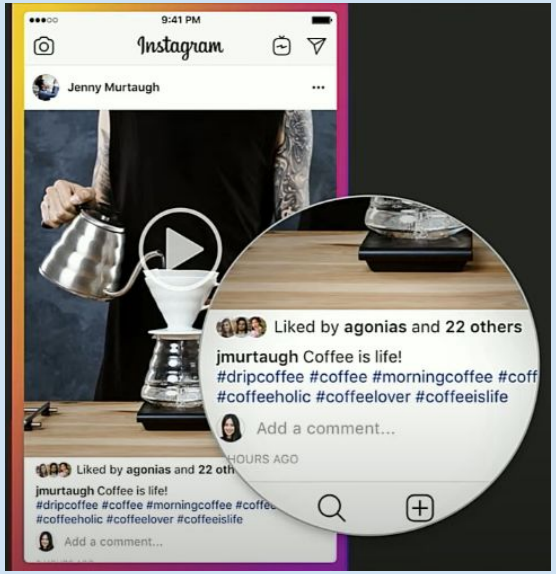
Sports video analysis

Video surveillance

Goal: Design a robust video classifier at scale

Key idea: Leverage public Instagram videos and the associated metadata for pre-training.

Step 1
Pre-training on large noisy web data



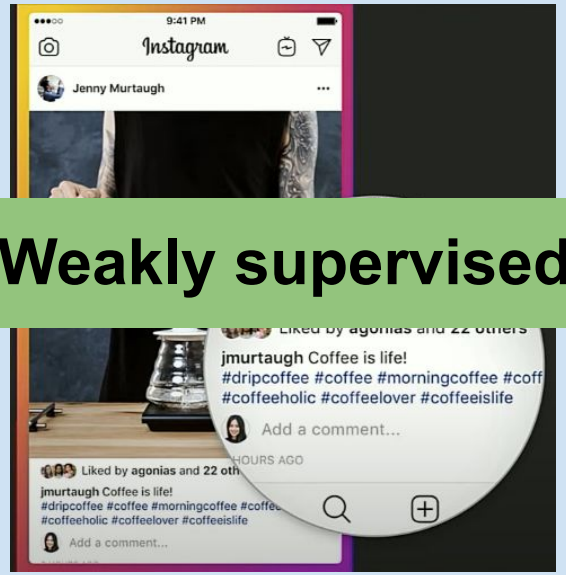
Step 2
Transfer learning on target dataset

Ghadiyaram et.al, *Large-scale weakly-supervised pre-training for video action recognition,* **CVPR'19**

Goal: Design a robust video classifier at scale

Key idea: Leverage public Instagram videos and the associated metadata for pre-training.

Step 1
Pre-training on large noisy web data



Weakly supervised

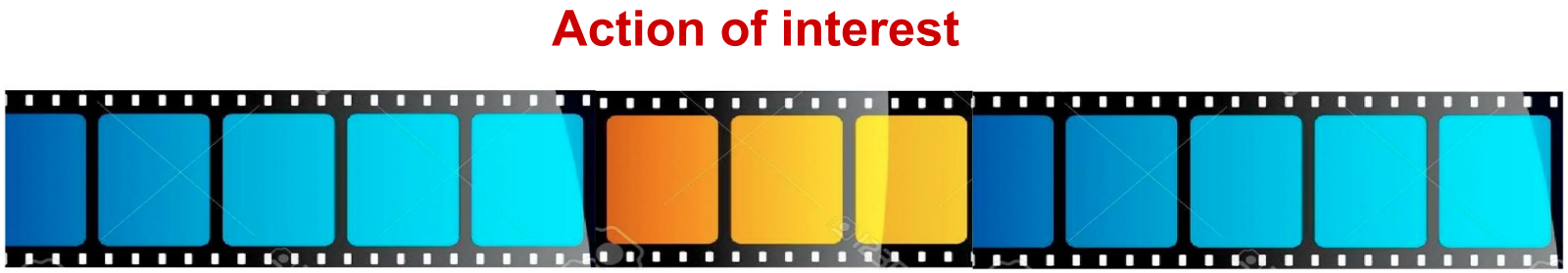
Step 2
Transfer learning on target dataset

Fully-supervised

Ghadiyaram et.al, *Large-scale weakly-supervised pre-training for video action recognition,* **CVPR'19**

Challenges with weakly supervised pre-training

- Temporal noise

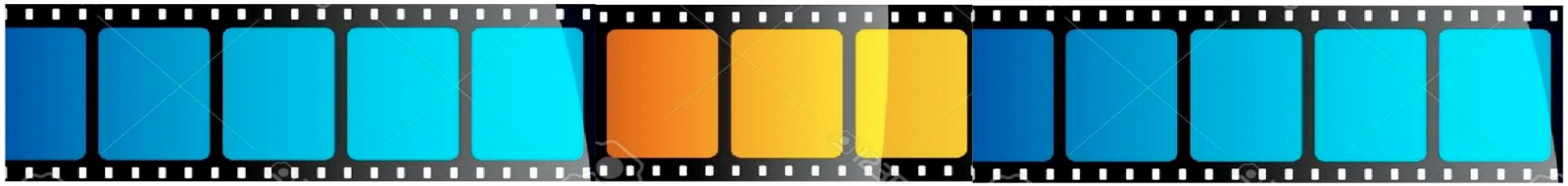


Ghadiyaram et.al, *Large-scale weakly-supervised pre-training for video action recognition,* **CVPR'19**

Challenges with weakly supervised pre-training

- Temporal noise

Action of interest



How to filter out label noise and temporal noise in the pre-training data?

#TBT

Non-visual concepts

#Tea

Incorrect tags

#Kettle

Not action oriented

Ghadiyaram et.al, *Large-scale weakly-supervised pre-training for video action recognition,* **CVPR'19**

Our solution: Design an "action-specific" pre-trained labels

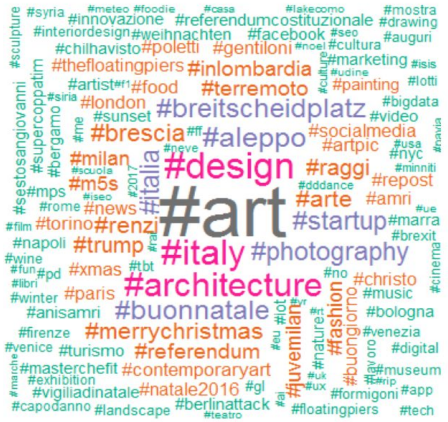
Seed labels

- Walking a dog
- .
- .

Hashtag matching

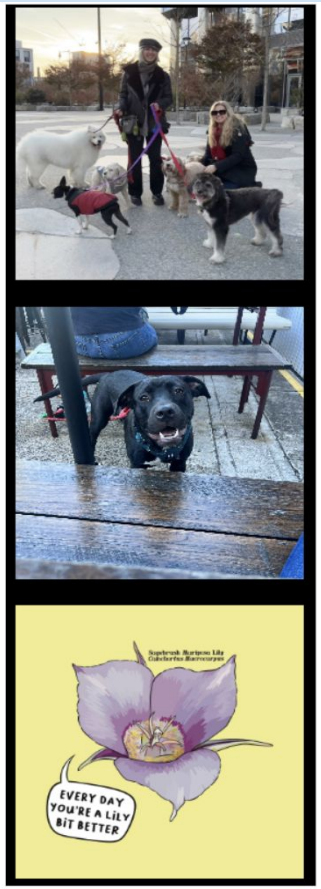
Label expansion

- walkingadog
- dogwalking
- dogwalks
- petwalks



Get associated videos

Pre-training data (65M)



Ghadiyaram et.al, *Large-scale weakly-supervised pre-training for video action recognition,* **CVPR'19**

Our solution: Design an “action-specific” pre-trained labels

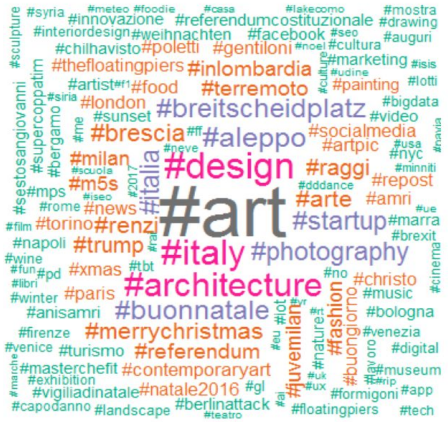
Seed labels

- Walking a dog

Label expansion

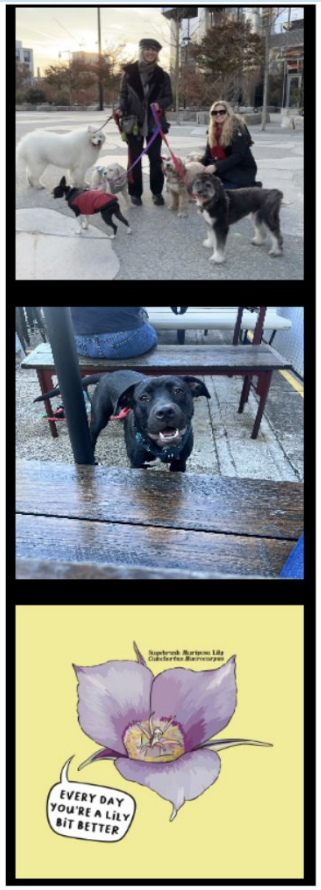
- walkingadog
- dogwalking
- dogwalks
- petwalks

Hashtag matching



Get associated videos

Pre-training data (65M)



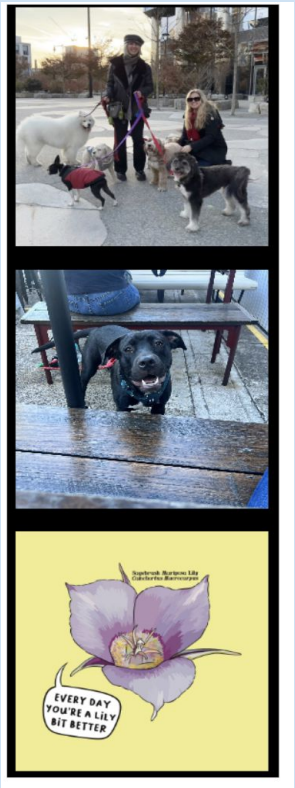
- Existing video datasets’ vocabulary (eg: Kinetics)
- Nouns (WordNet)
- Verbs (VerbNet)

Ghadiyaram et.al, *Large-scale weakly-supervised pre-training for video action recognition,* **CVPR’19**

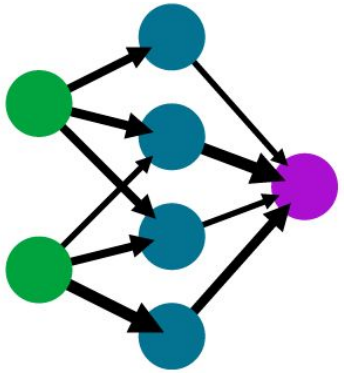
Overall Approach

Weakly supervised

Pre-training data (65M)



R(2+1)D video architecture [1]



Fully-supervised

Fine-tune on target datasets

Kinetics [3]

Epic Kitchens [2]

[1] Tran et.al, CVPR'18
[2] Damen et.al., ECCV'18
[3] Carreira et.al

Ghadiyaram et.al, *Large-scale weakly-supervised pre-training for video action recognition,* **CVPR'19**

Results

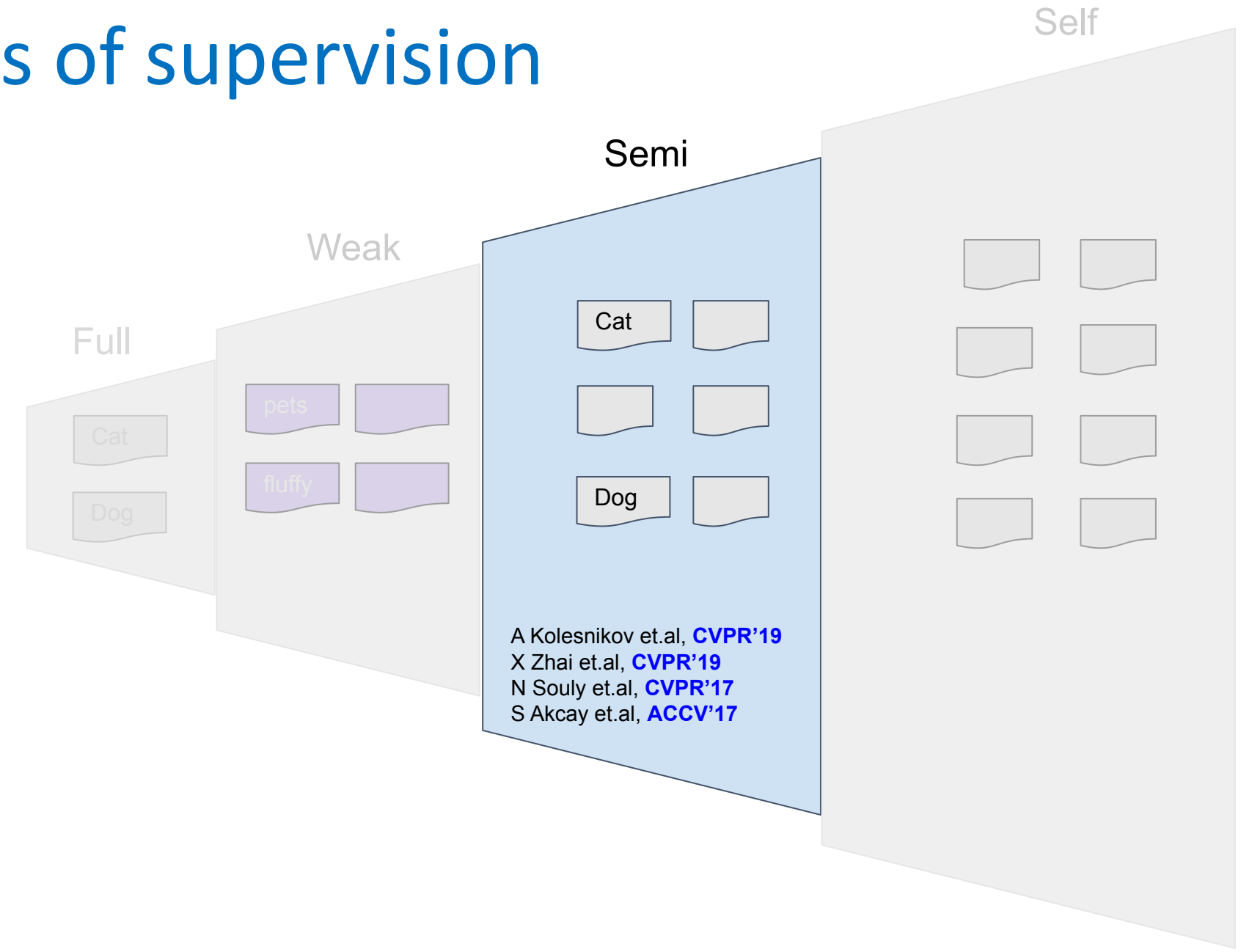
- Dataset: Kinetics

Method; pre-training	top-1	top-5	Input type
I3D-Two-Stream [11]; ImageNet	75.7	92.0	RGB + flow
R(2+1)D-Two-Stream [14]; Sports-1M	75.4	91.9	RGB + flow
3-stream SATT [69]; ImageNet	77.7	93.2	RGB + flow + audio
NL I3D [65]; ImageNet	77.7	93.3	RGB
R(2+1)D-34; Sports-1M	71.7	90.5	RGB
Ours R(2+1)D-34; IG-Kinetics	79.1	93.9	RGB
Ours R(2+1)D-34; IG-Kinetics; SE	79.6	94.2	RGB
Ours R(2+1)D-152; IG-Kinetics	80.5	94.6	RGB
Ours R(2+1)D-152; IG-Kinetics; SE	**81.3**	**95.1**	RGB

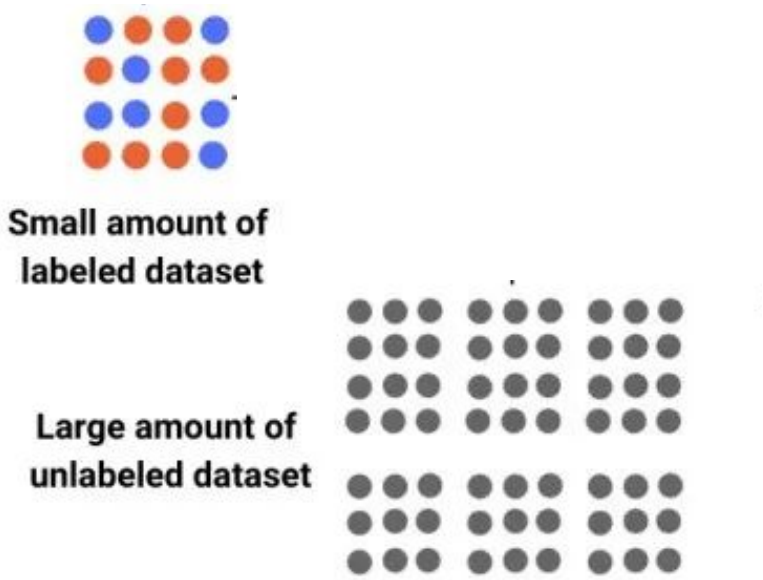
- Won a 2nd place in EPIC-Kitchens action recognition challenge, CVPR19

Ghadiyaram et.al, *Large-scale weakly-supervised pre-training for video action recognition,* **CVPR'19**

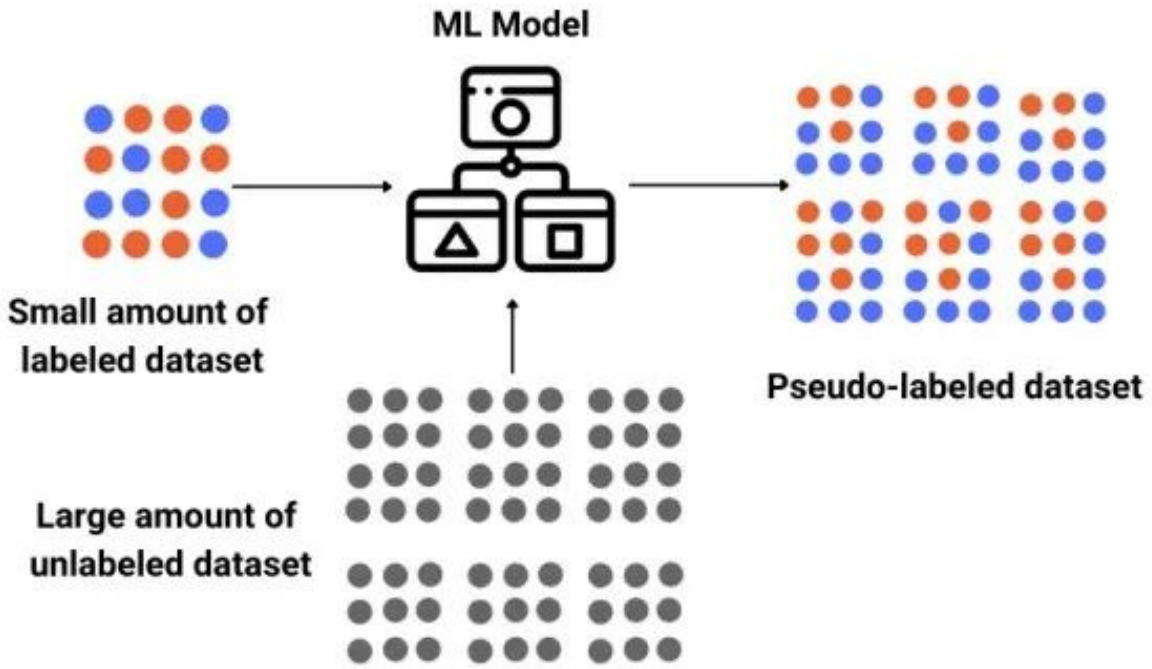
Types of supervision



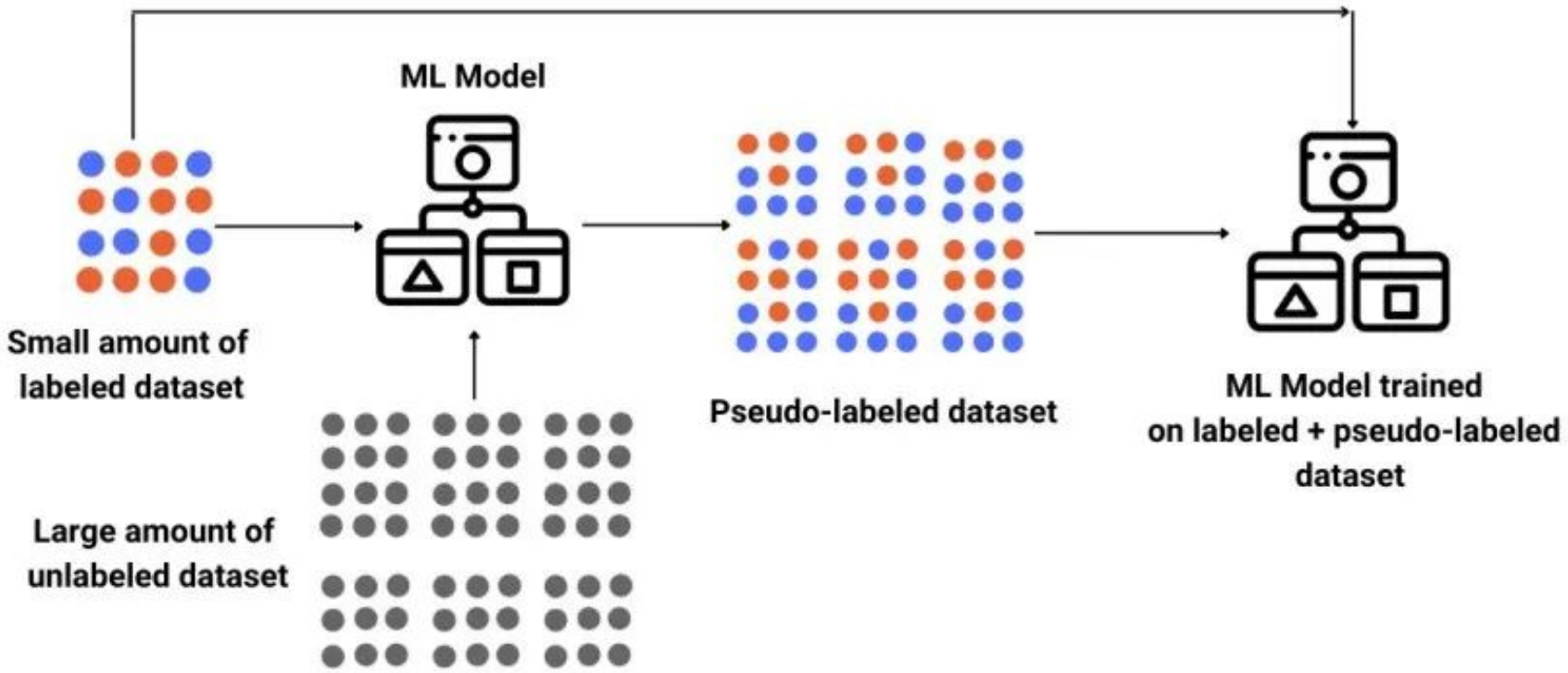
Semi supervision



Semi supervision

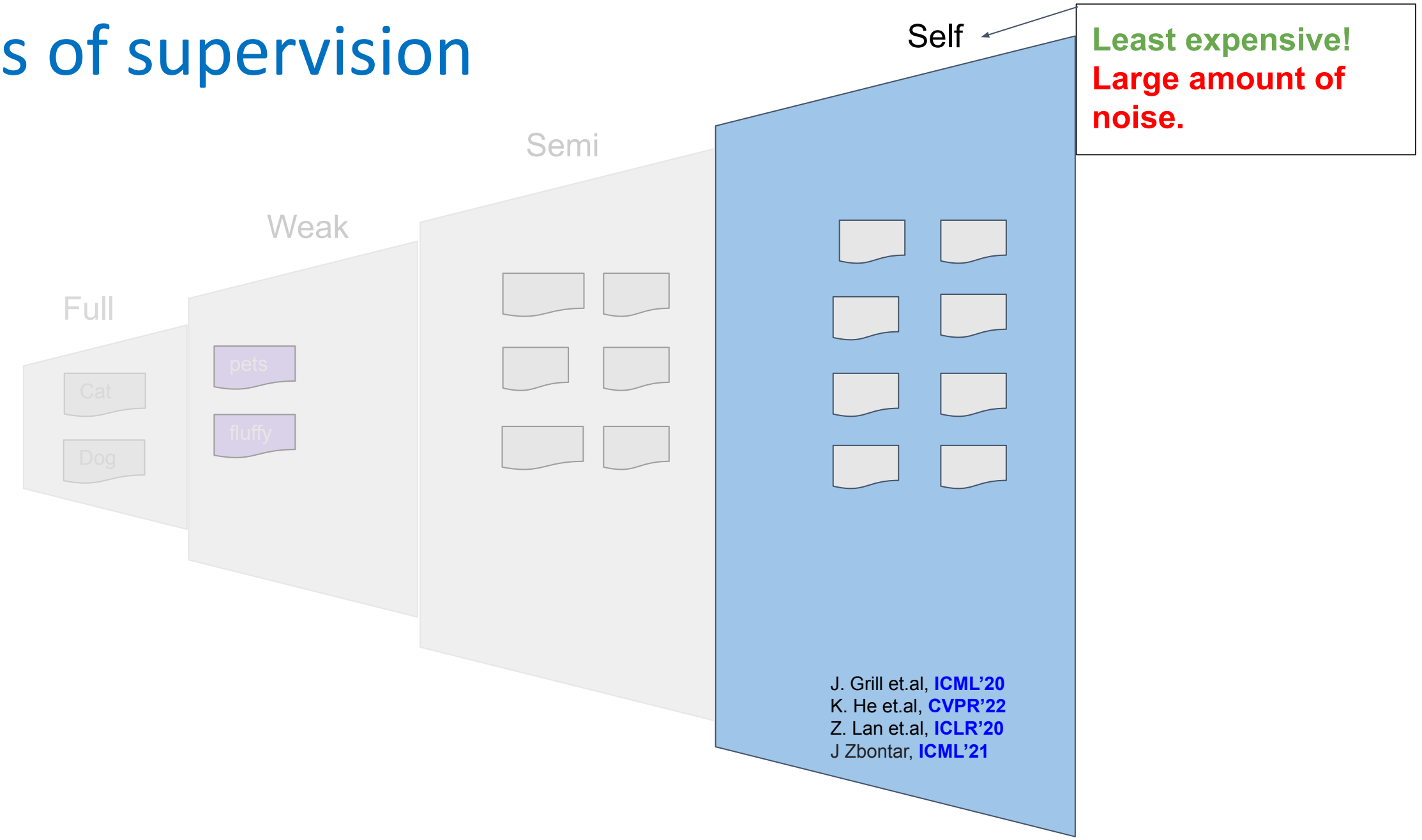


Semi supervision



- Very frequently used technique to increase the amount of training data

Types of supervision



Self-supervision

- **BYOS:** Build your own supervision.
- **Key idea:** Exploit information redundancies, context in raw data.

Self-supervision

- **BYOS:** Build your own supervision.
- **Key idea:** Exploit information redundancies, context in raw data.

Spatial context, patch layout

In-painting missing pixels

Temporal order of frames

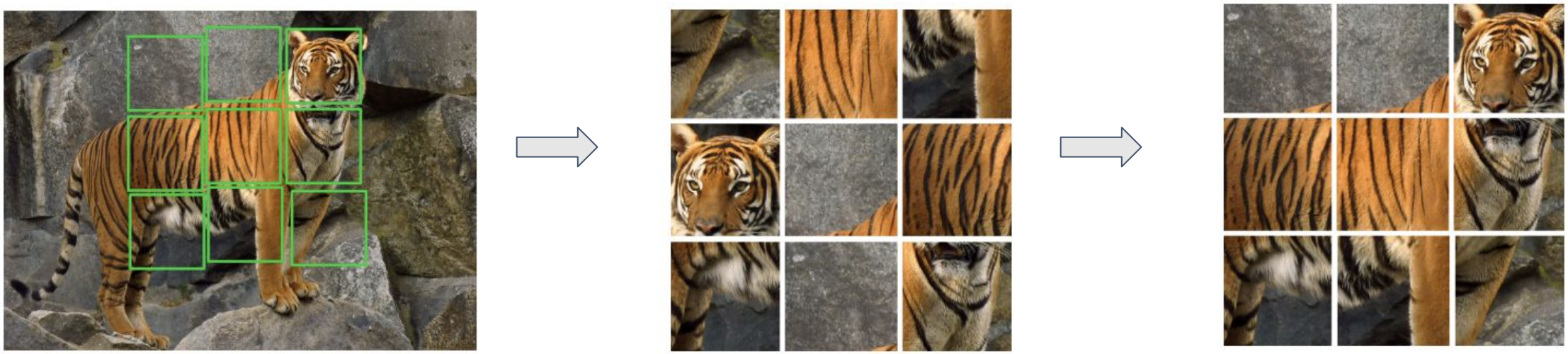
Temporal coherence in video

Pose variations

Colorization

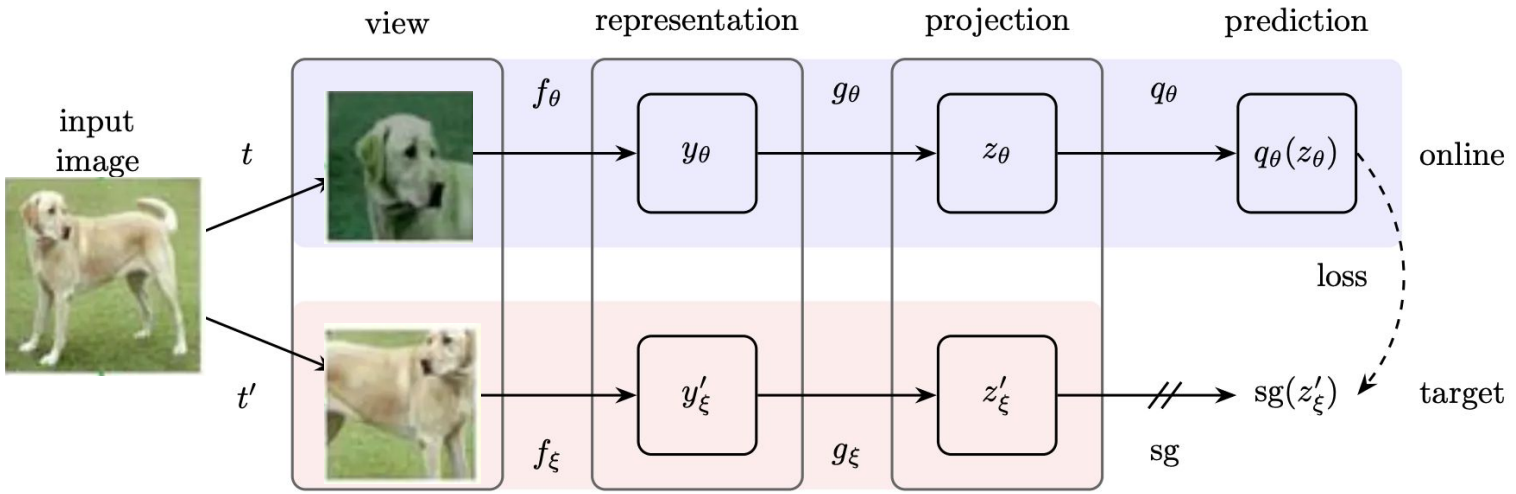
Rotation / Translations

Example pretext tasks: Jigsaw puzzle



- **Pretext task:** Predict the correct order of the patch indices

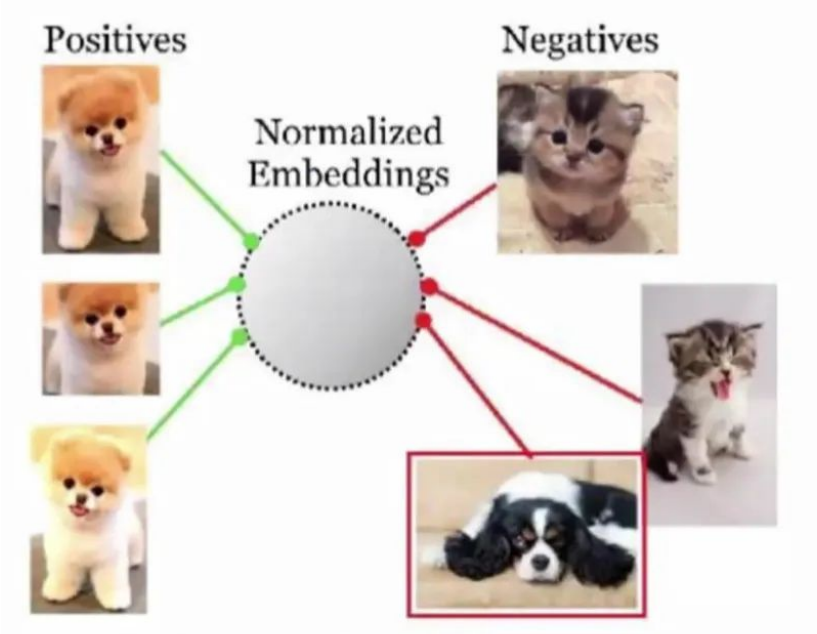
Example pretext tasks: Image transformations



Pros:

- **Learn a feature representation invariant to translations.**
- Annotation cost = 0

Self-supervision: Contrastive learning



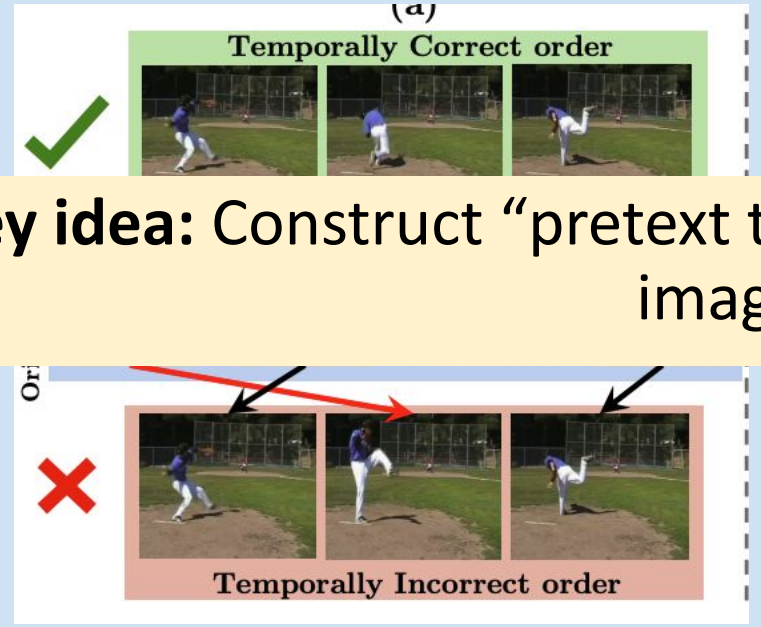
- **Explicitly pull similar data points closer together**
- **Push dissimilar data points farther apart.**
- Annotation cost = 0

Self-supervision

- What can be our “proxy” or “pretext” task?
  - Temporal coherence in video
    - Mobahi et al. 2009, Wang & Gupta 2015, Wang et al. 2016, Gao et al. 2016.
  - Audio channel – ambient sounds.
    - Owens et al. 2016, Arandjelovic & Zisserman 2017
  - Ego-motion
    - Jayaraman et al. 2015, Agrawal et al. 2015
  - Spatial context, patch layout
    - Doersch et al. 2015, Noroozi & Favaro 2016
    - Pathak et al. 2016
  - Colorization
    - Larsson et al. 2016, Zheng et al. 2016
  - Temporal order of frames
    - Misra et al. 2016

Goal: Build your own supervision

Step 1
Pretrain on "pretext" tasks



Step 2
Transfer learning on target dataset

Key idea: Construct "pretext tasks" by leveraging redundancy in the image/video data

Summary: Types of supervision