Bits of Vision: Representation Learning

Recap

- Network architecture for learning from images
  - Convolutional layer
  - Convolutional neural networks
    - Basic (~2012): Conv layers + FC
    - Deep (~2016): Residual connection, Bottleneck, …
    - Tiny (~2020): Depthwise convolution, Inverted bottlenecks, …

- **Focus.** We can train a large model with minimal computation

Today

- We explore key ideas in visual representation learning
  - i.e., learning useful features, with minimal supervision

  - Data augmentation
  - Transfer learning
  - Semi-supervised Learning
  - Self-supervised Learning

- **Focus.** We can utilize large datasets with minimal human labeling efforts

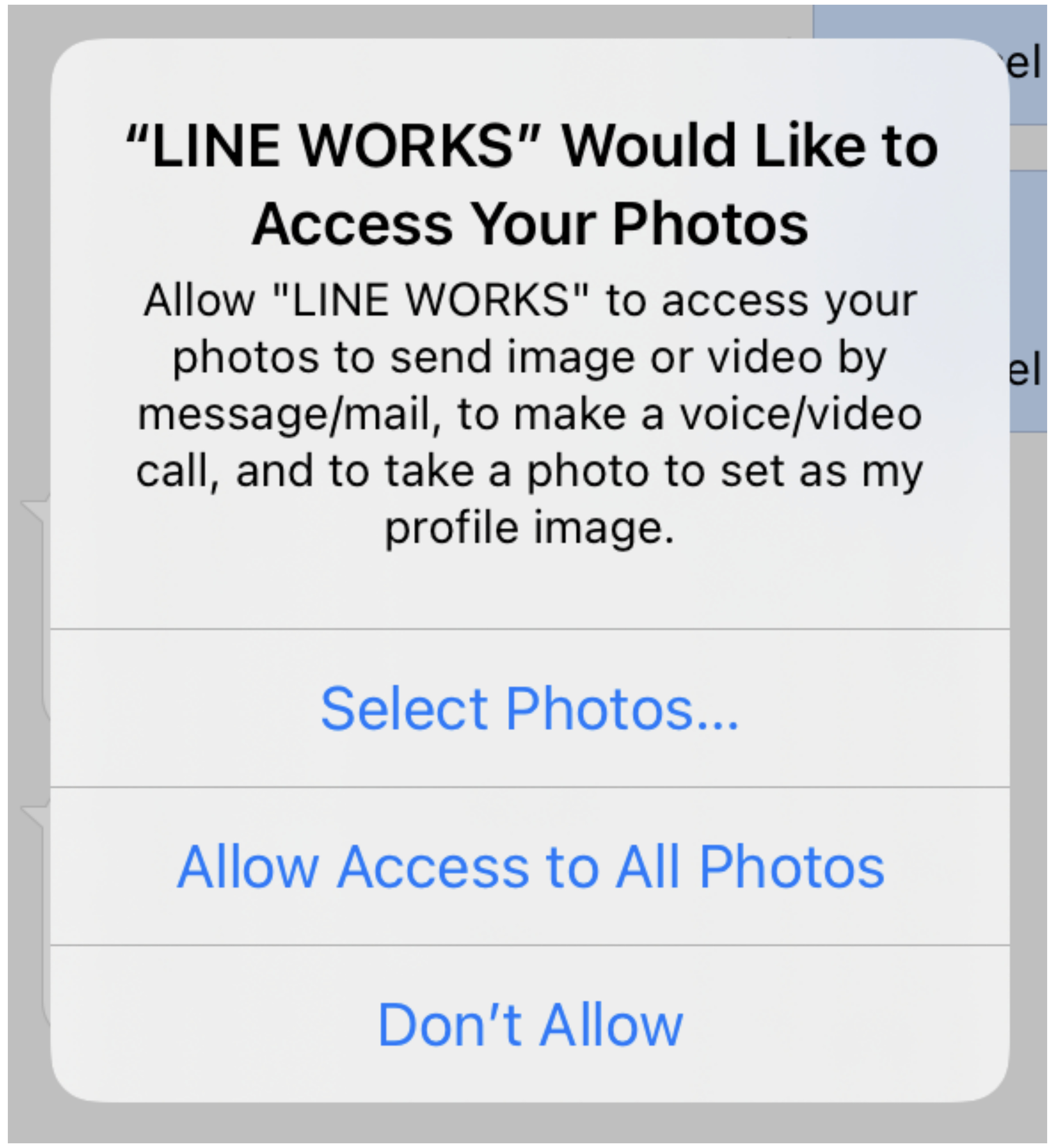
Visual Supervised Learning: Ideas for Enlarging the Dataset

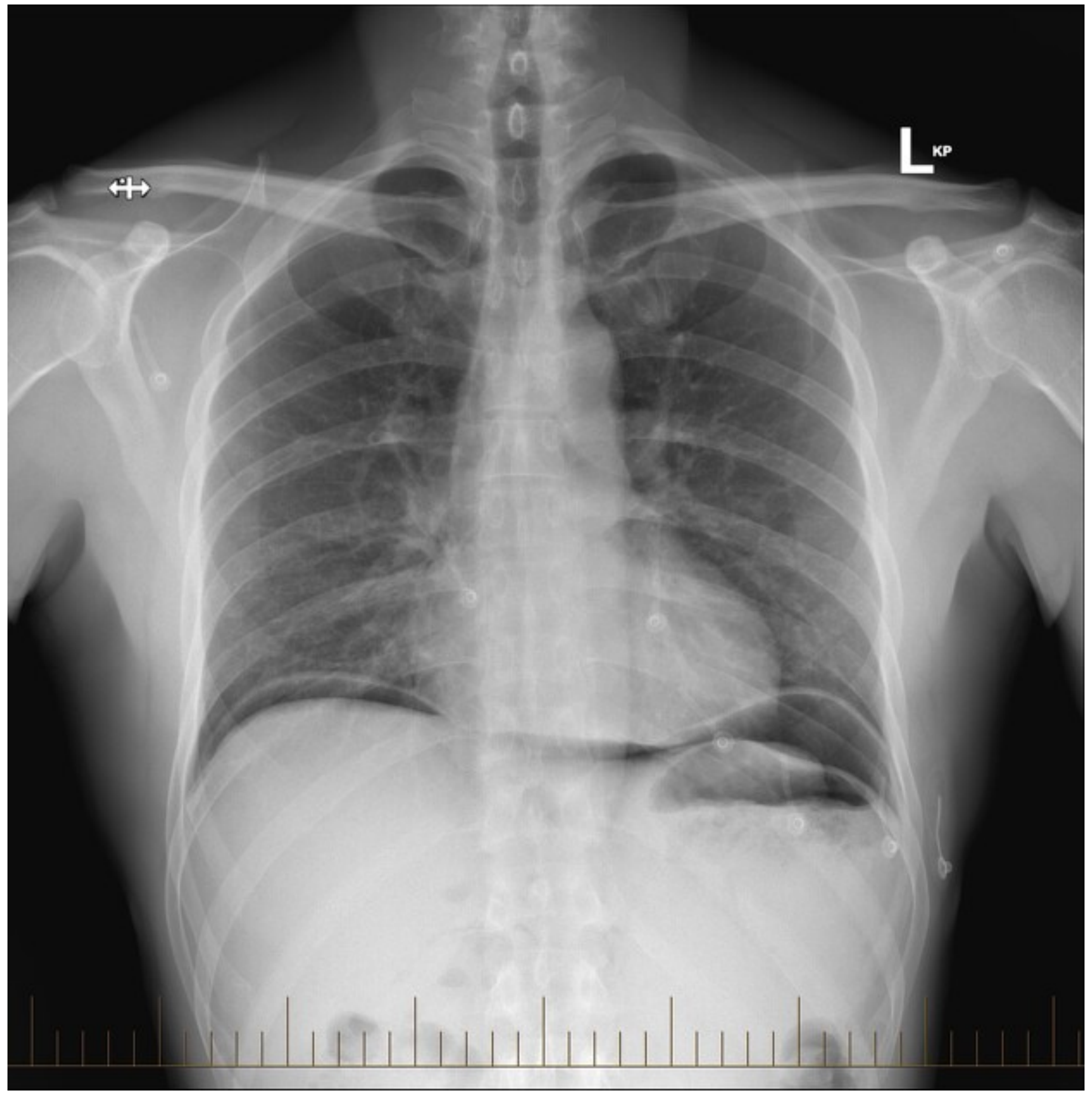
Problem

- To get a high performance, we need to scale up both model and data
- **Problem.** Collecting a large, annotated visual dataset is challenging

(1) Collecting Image
  - License & Privacy
  - Special Domains

(2) Pre-Processing
  - Filtering
  - Resizing
  - Denoising





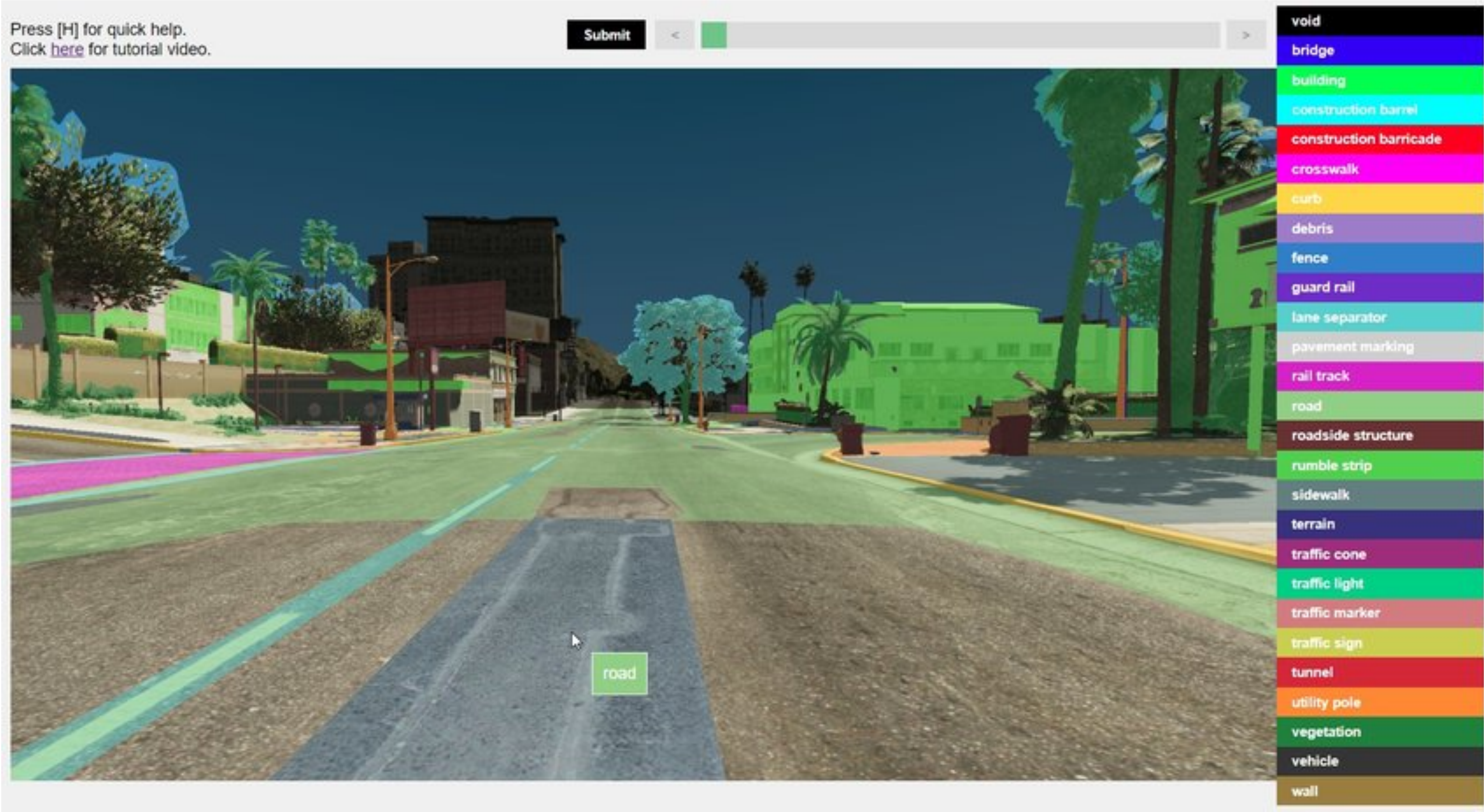
Problem

(3) Labeling

- Hiring annotators
  - Requires Expertise
  - Much Labor

(4) Post-Processing

- Quality Review
- Dataset Balancing

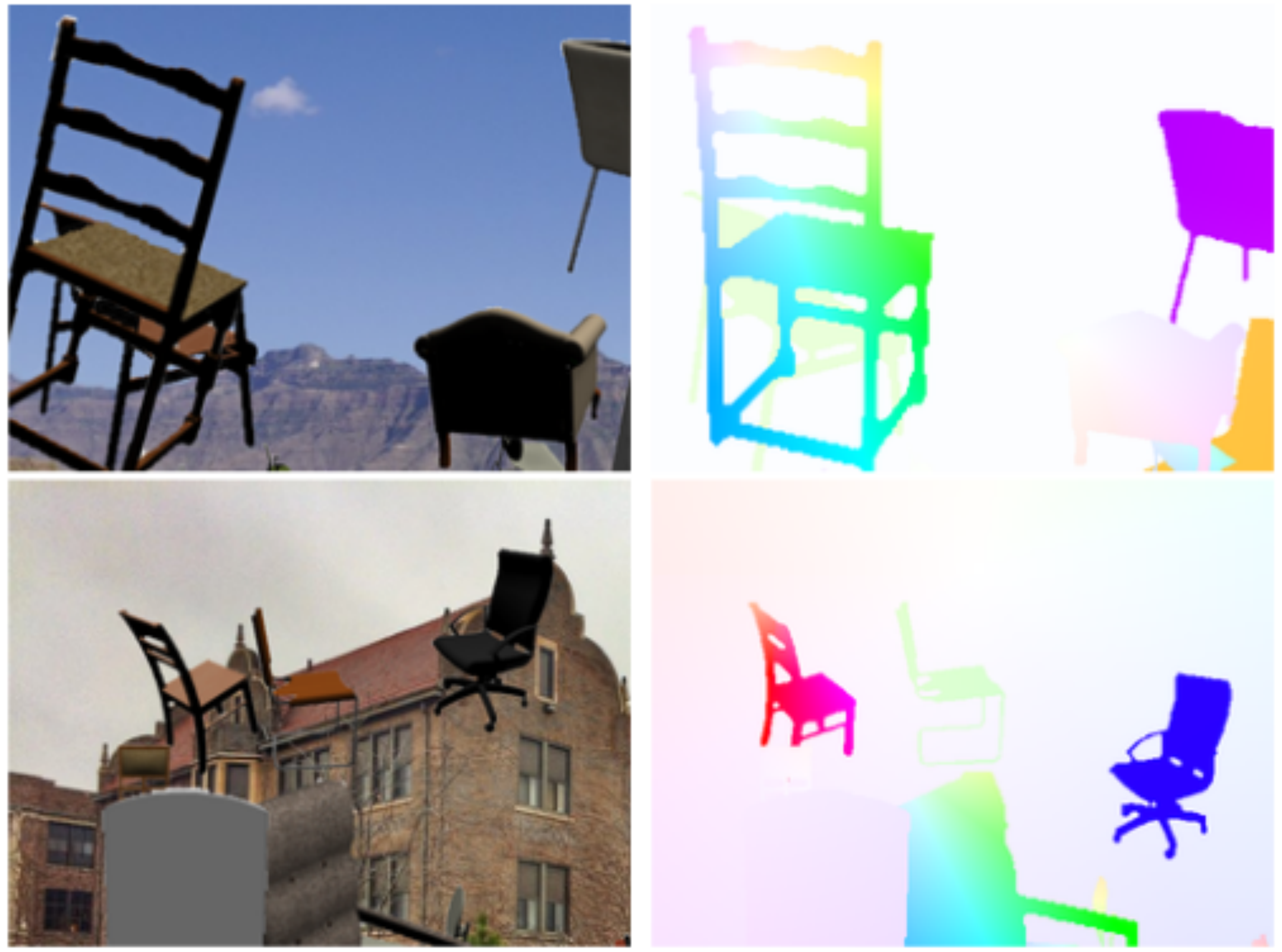


Overview

- **Idea.** Develop effective algorithms to utilize alternate data sources

(1) Fake but realistic data

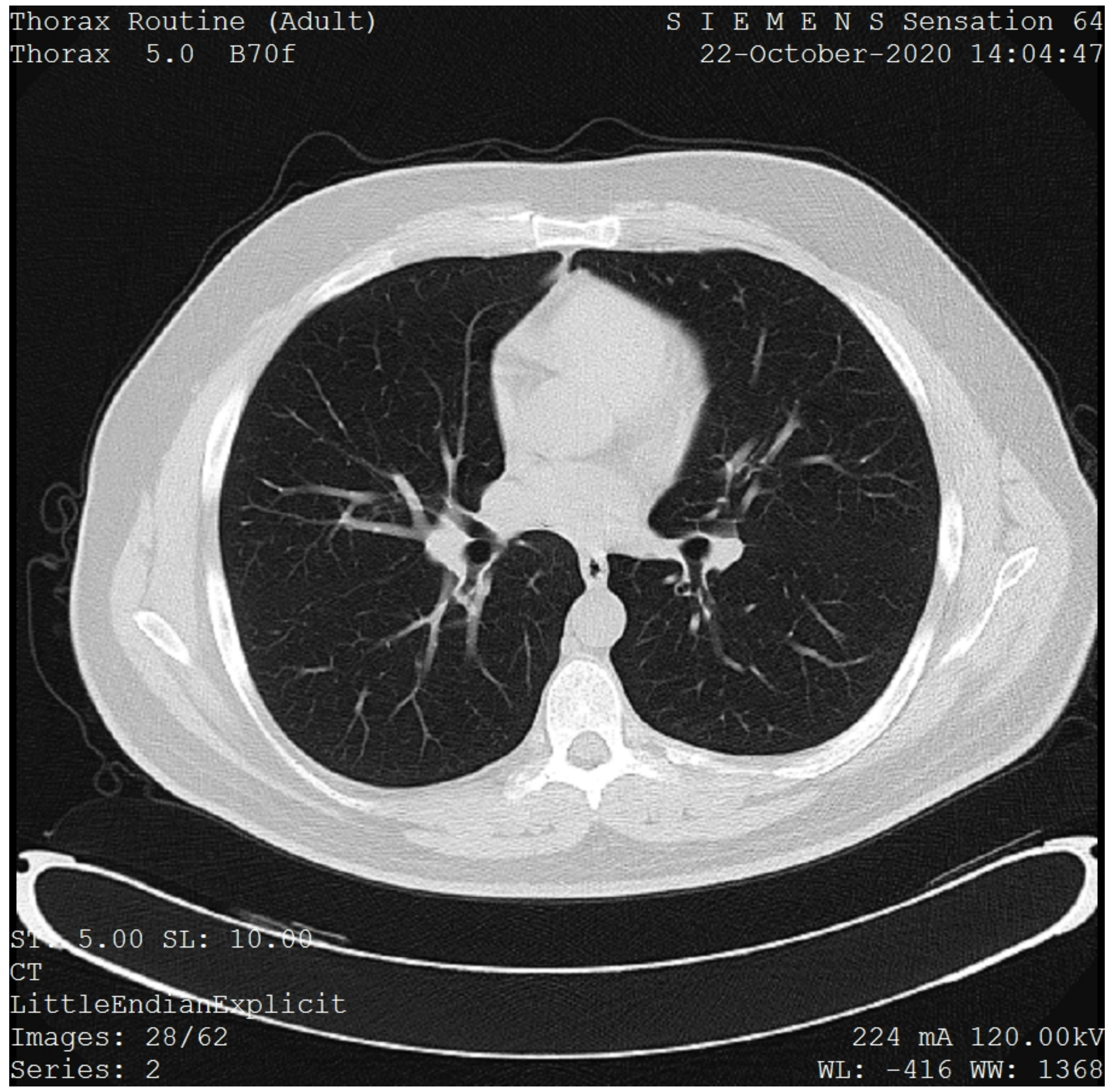
- Data augmentations ✅
- Synthetic Dataset
- Generated Images



Overview

(2) Data from related domains

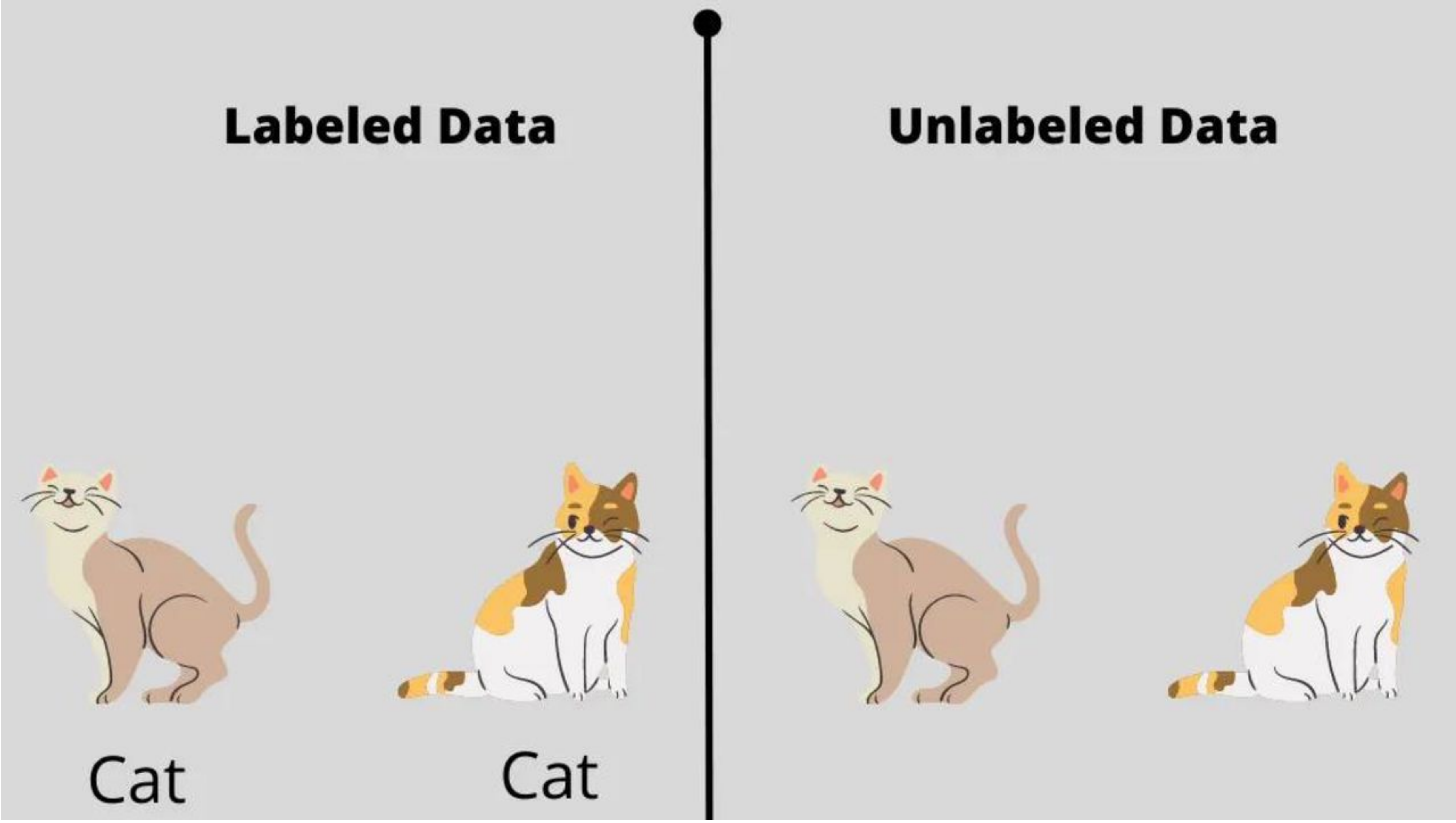
- Transfer learning ✅
- Few-shot learning
- Continual learning
- Meta-learning



Overview

(3) Utilize unlabeled data (as in K-Means, PCA)

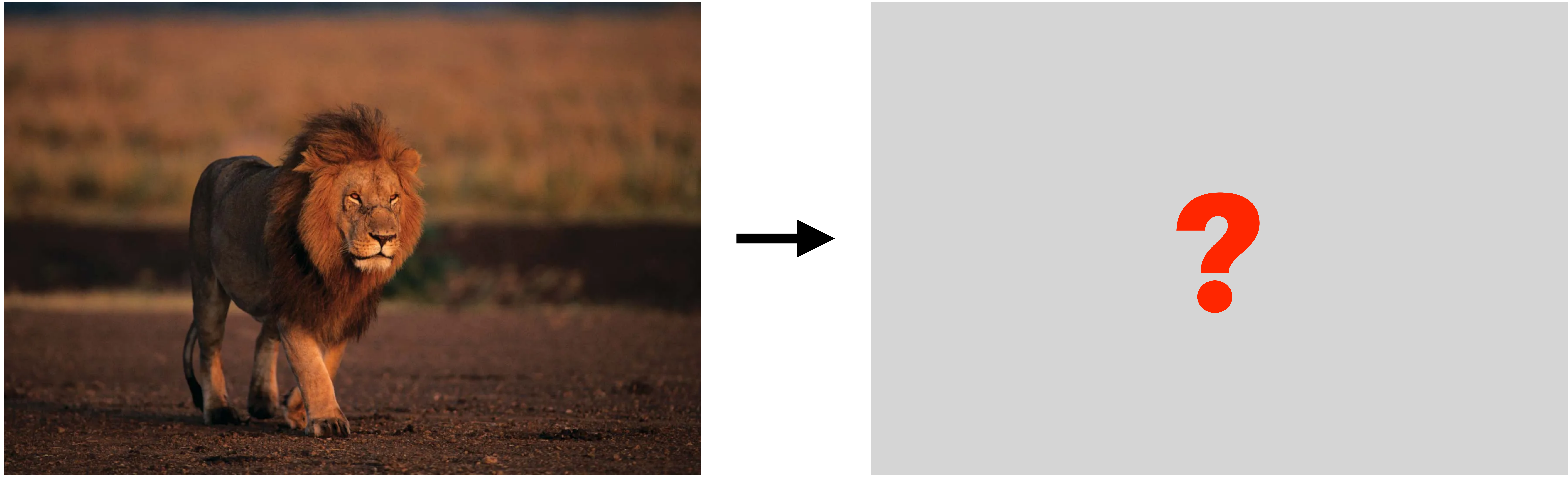
- Semi-Supervised Learning ✅
- Self-Supervised Learning ✅
- Generative Modeling
  - Later
- Weakly supervised Learning
  - Later



Data augmentation

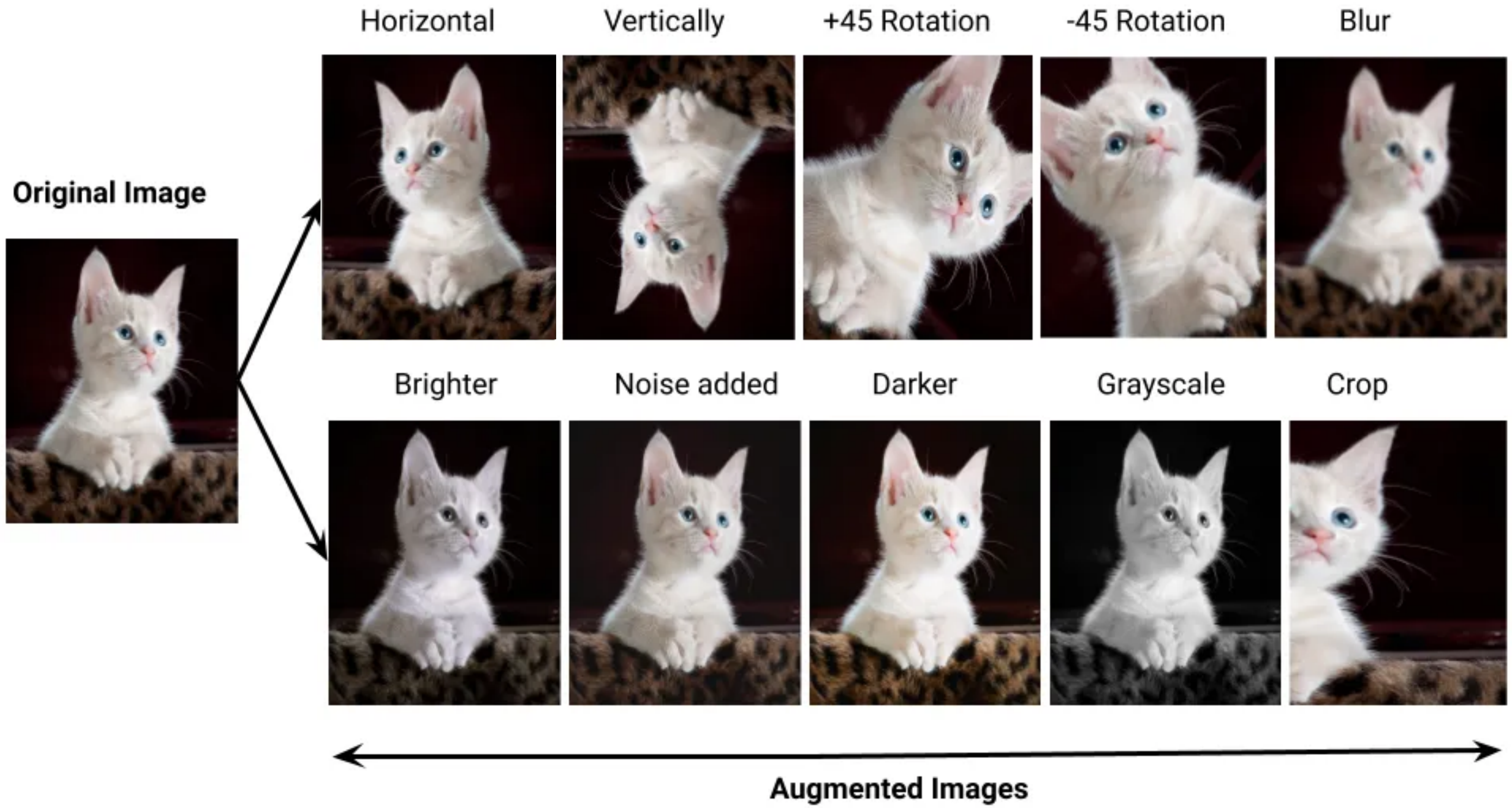
Data augmentation

- **Idea.** Apply generic operations on existing labeled data to make a realistic yet new images
  - Caution. Choose the operations that does not alter semantics



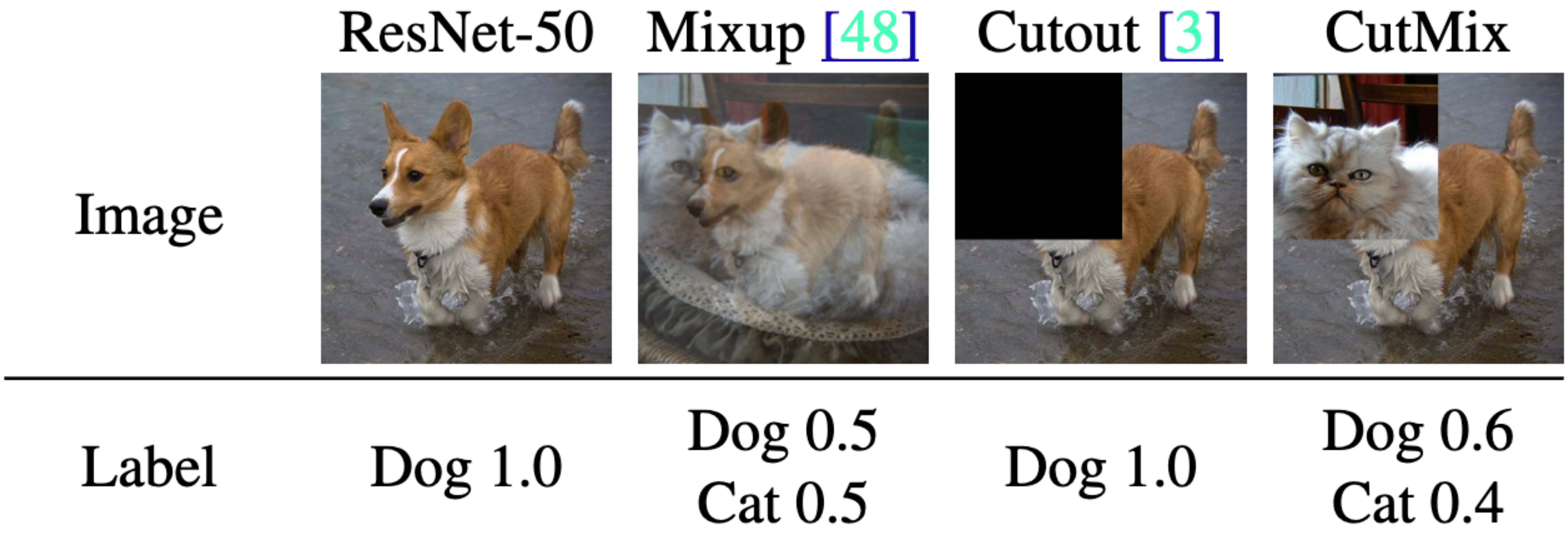
Data augmentation

- **Basic operations.** Flip, Resize, Shift, Crop, Rotate, Blur, Grayscale, …
  - Caution. In many cases, avoid vertical flipping or extreme resizing



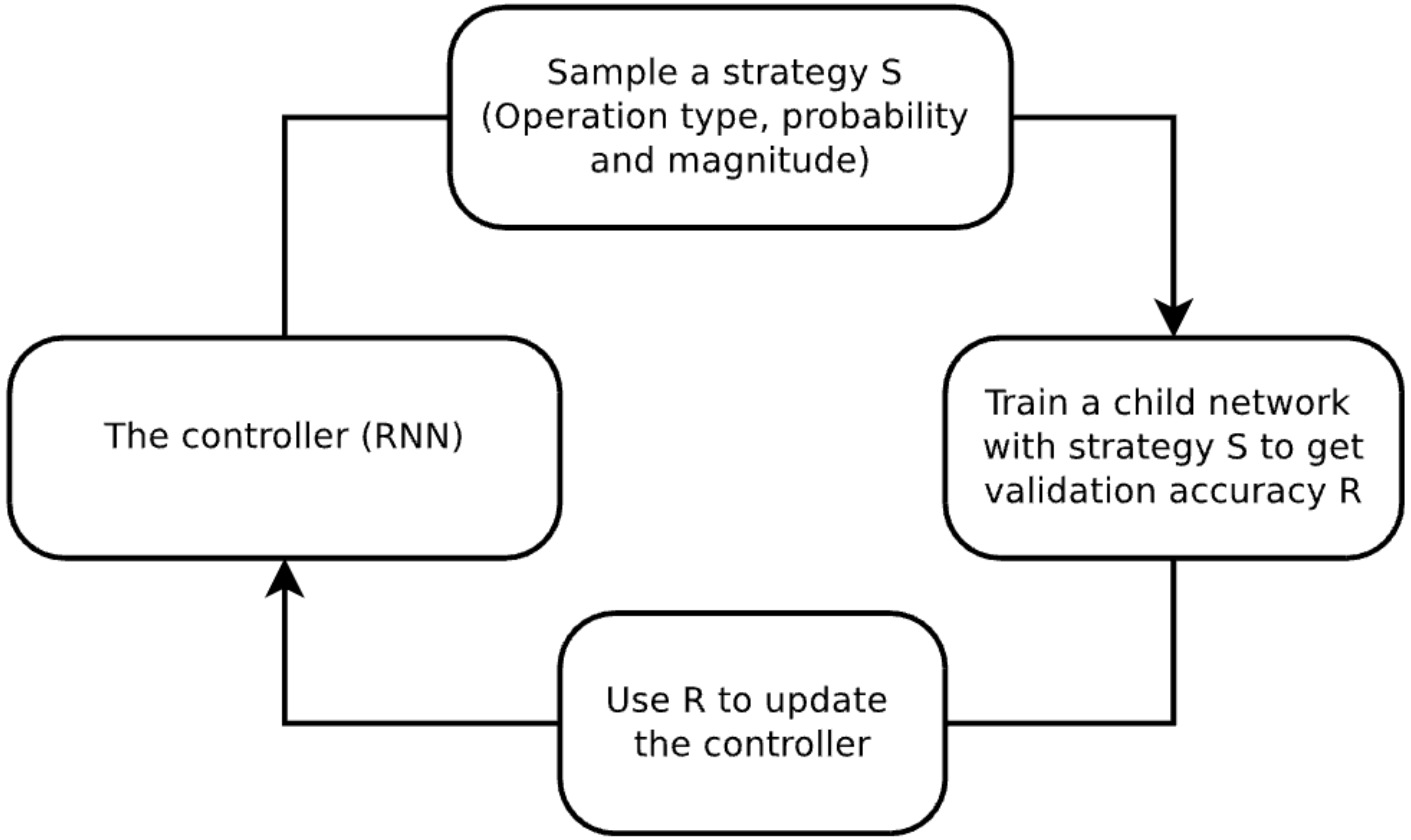
Data augmentation

- **Mixing images.** It is also typical to mix multiple images
  - SMOTE/MixUp. Overlay multiple images, and give mixed labels
  - CutMix. Combine multiple pieces of images wihtout overlapping
  - CutOut. Combine with a black canvas



Data augmentation

- **AutoAugment.** Train an image augmentation strategy that works best with the target dataset-model pair
    - Requires a lot of training
        - Followed up by RandAugment, Fast AutoAugment, …



Touvron et al., "DeiT III: Revenge of the ViT," ECCV 2022

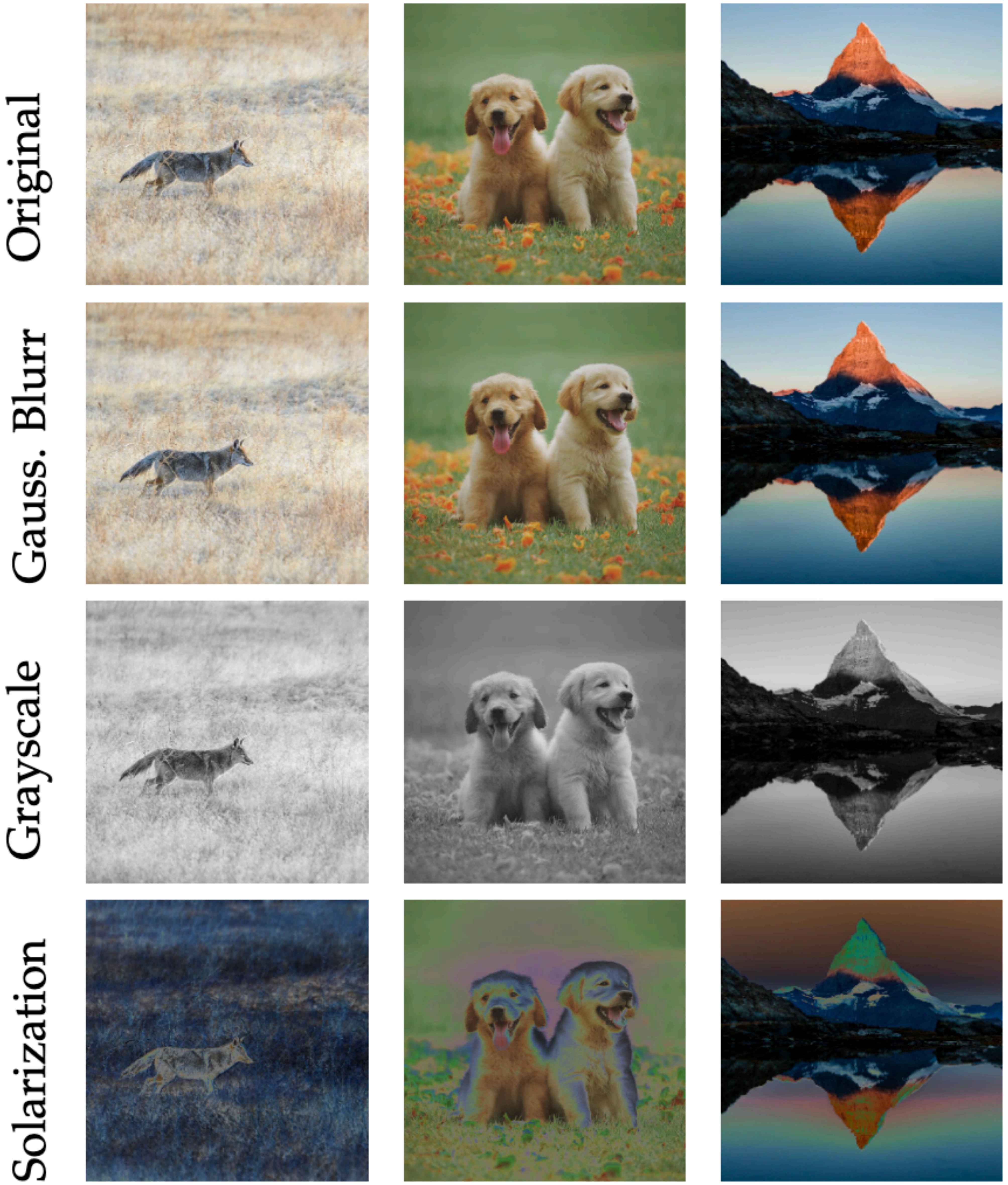
Data augmentation

- **Recently.** Avoid excessive augmentation
  - Augmentation makes us lose information anyways, which is potentially very important
    - e.g., horizontal flips vs. lefty
  - Self-supervised training works well, so we have less motivations

- Typical to use:

  Gaussian Blur + Grayscale + Solarization

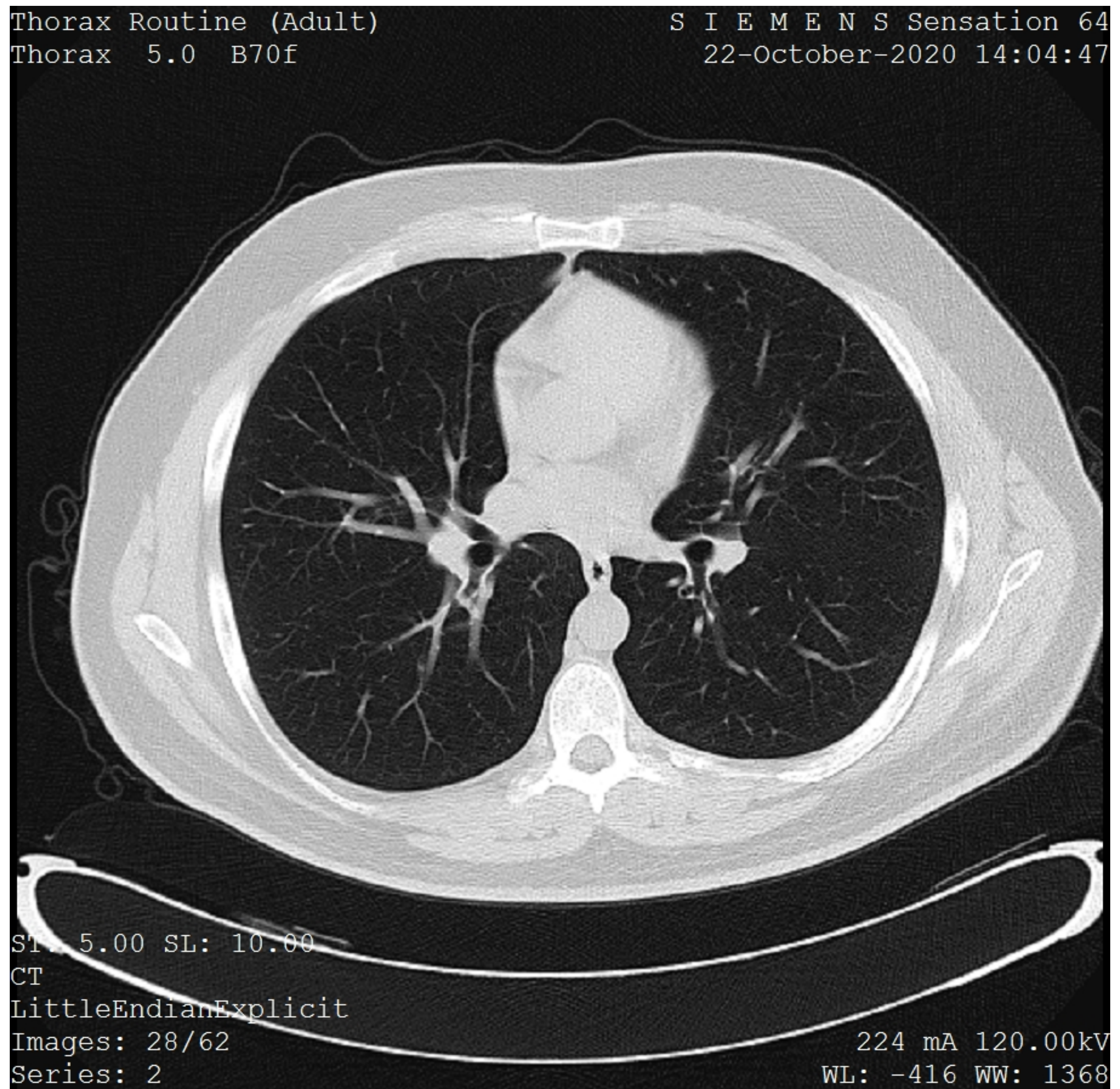
  (optional: horizontal flip, color jitter)



Transfer Learning

Transfer learning

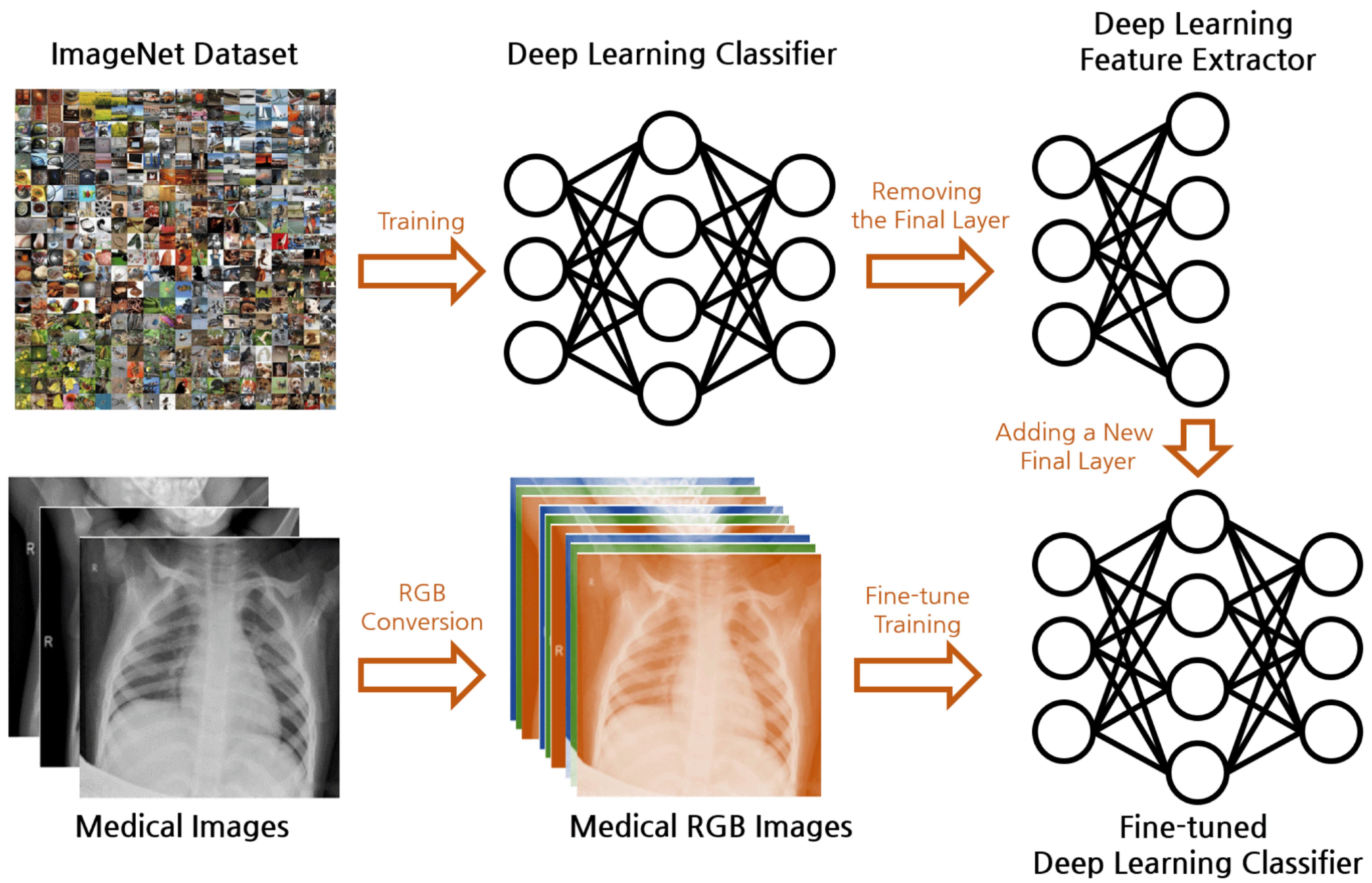
- Considers a setup where:
  - **Target.** We have a domain with very scarce data
    - MRI images, with very rare disease
  - **Source.** There is a relevant, data-plenty domain
    - Natural Images — Google has a 3B-scale dataset
    - Medical MRI Images — other body parts, other diseases



Transfer learning

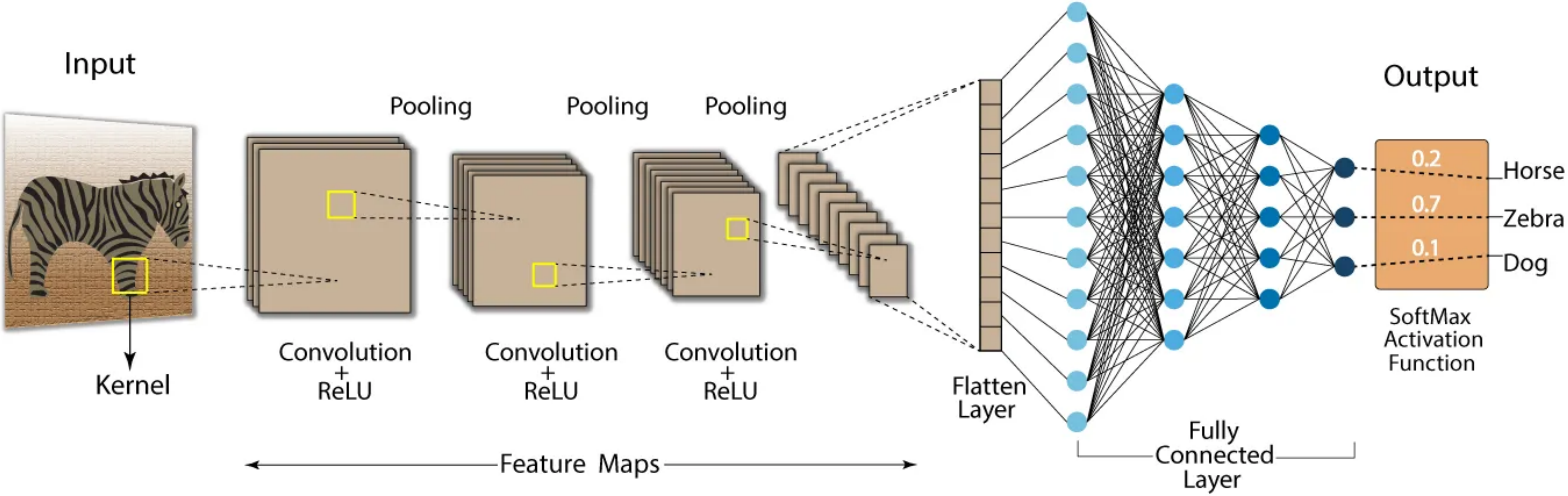
- **Idea.** Transfer the source domain knowledge to the target domain
  - Train a model on the source domain # called "pretraining"
  - Re-use the model weights on the target domain
  - Train further on the target domain data

- **Intuition.** There should be some common knowledge that is shared across different domains
  - Neurons that discern shapes, such as "circles" or "triangles"



Transfer learning

- There are many different ways to transfer the parameters
- **Common.** Cut off the final layer (classifier) from the model
  - <u>Reason</u>. The number of classes doesn't match



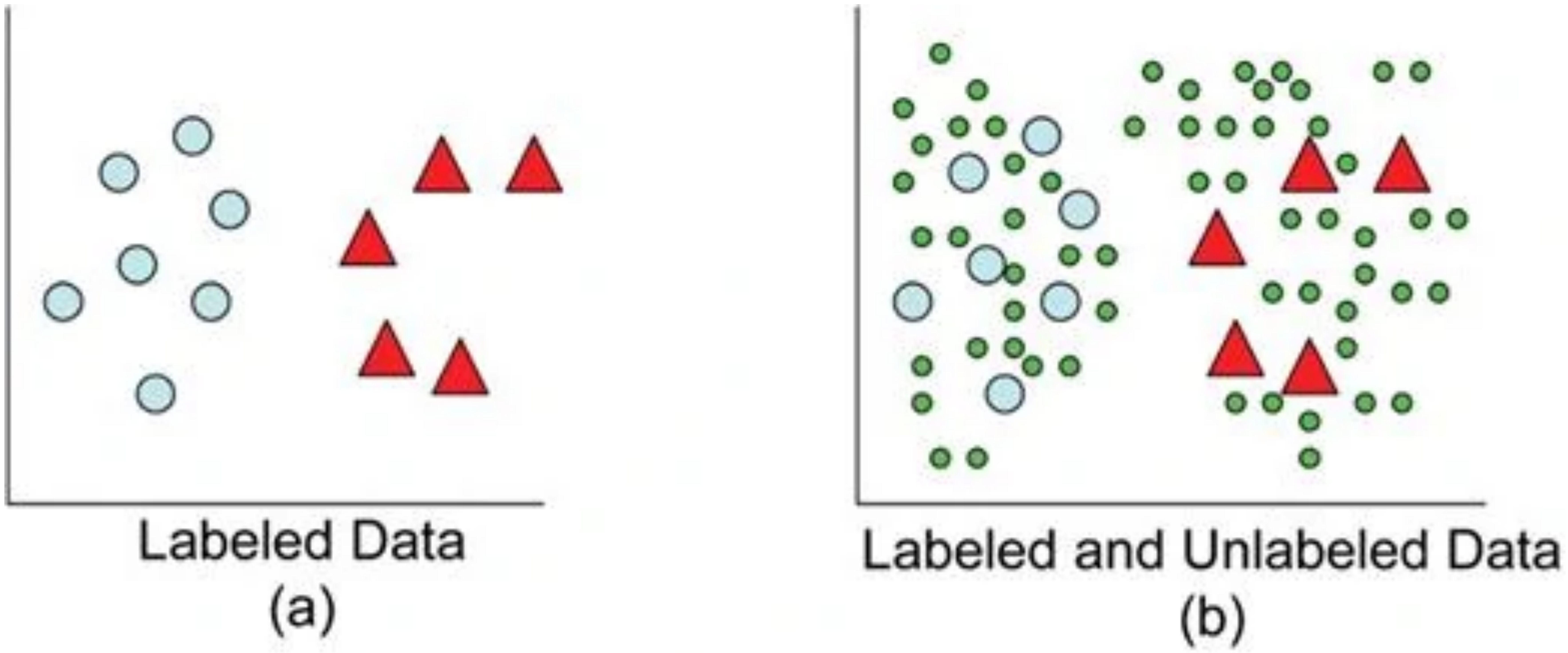
Transfer learning

- **Differences.** Choose one option:
  - Freezing. Freeze the pre-trained weights, and train only the newly added classification layer
    - Very scarce target domain data
    - Small domain discrepancy — avoiding forgetting useful info
  - Fine-tuning. Train weights further on new data
    - Relatively abundant target domain data
    - Large domain discrepancy
  - Adding Parameters. Add additional neurons / layers / weights and train them together, with or without freezing pretrained weights

Semi-Supervised Learning: Pseudo-Labeling Approach

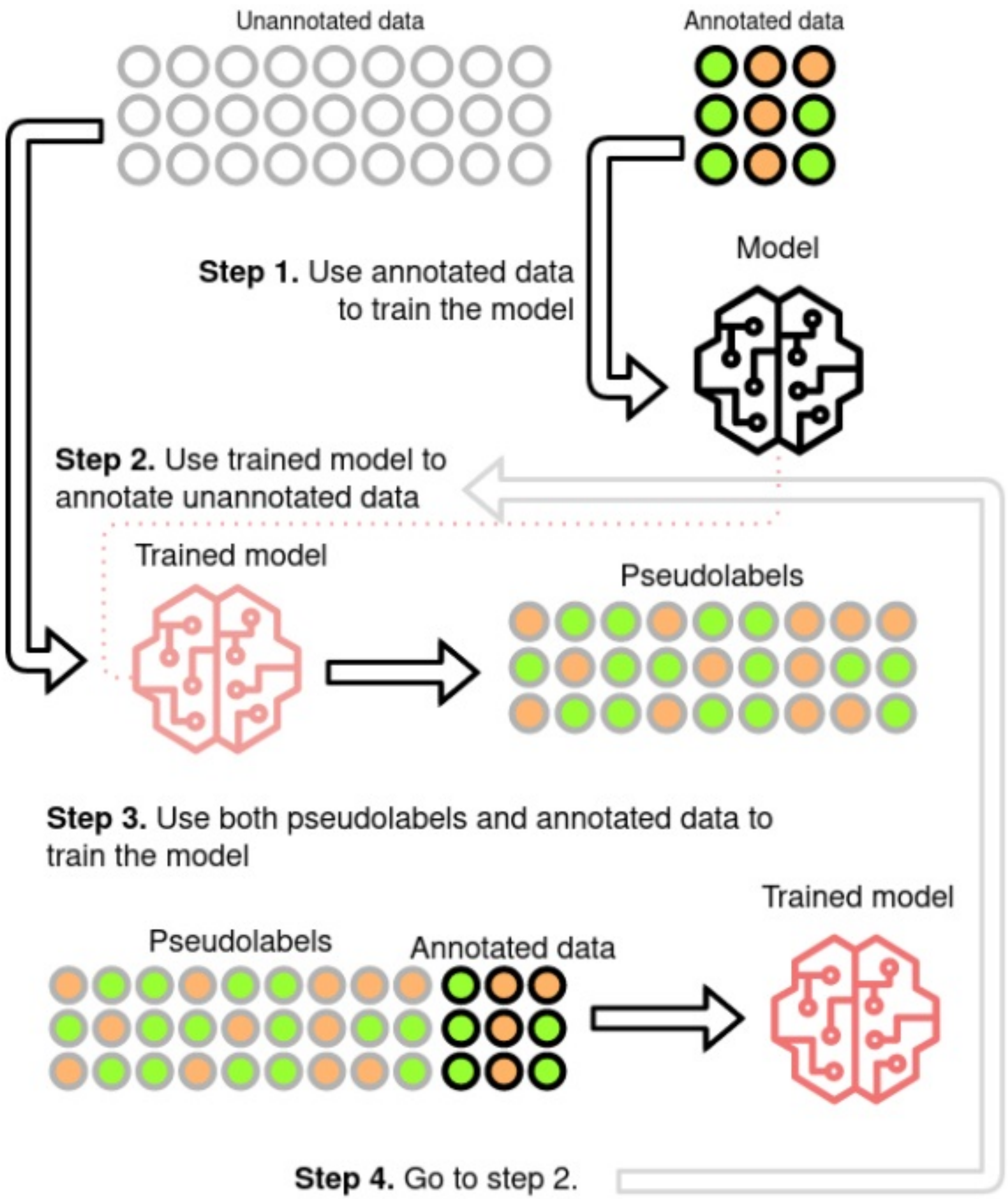
Semi-supervised learning

- Consider the case where we have both:
  - Small number of labeled data
  - Large number of unlabeled data
- **Question.** How can we utilize the unlabeled data effectively?



Semi-supervised learning

- **Idea.** Give pseudo-labels for samples
  - Train a model on the labeled data
  - Predict on the unlabeled data
    - If model is confident on a sample, put a pseudo-label on the sample
  - Continue training, with both the labeled dataset & pseudo-labeled datasets
  - Repeat
- **Note.** Not the only way, but popular



Self-Supervised Learning

Self-supervised learning

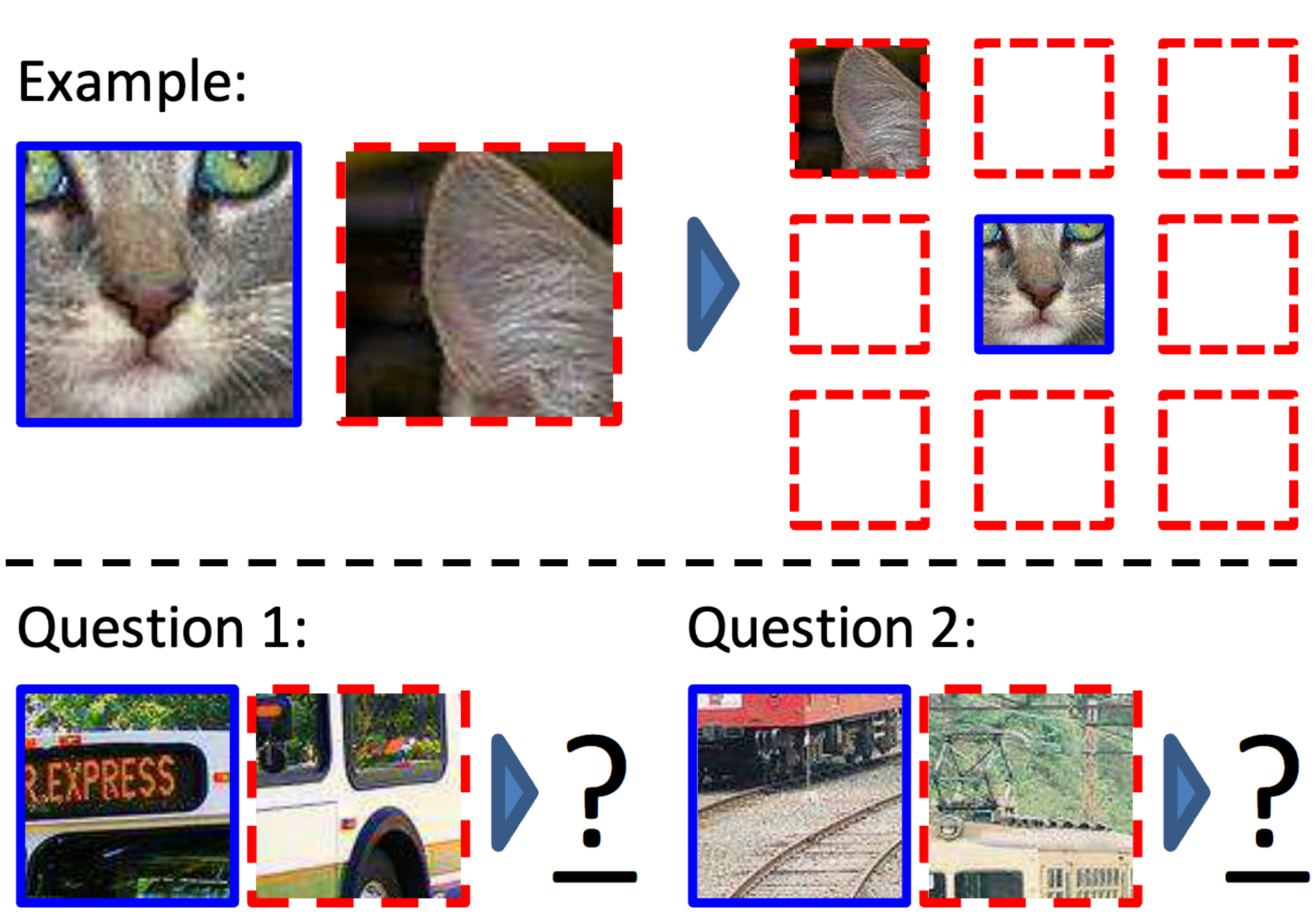
- Consider the case where we have a lot of unlabeled data
- **Goal.** Train a representation that can be transferred well

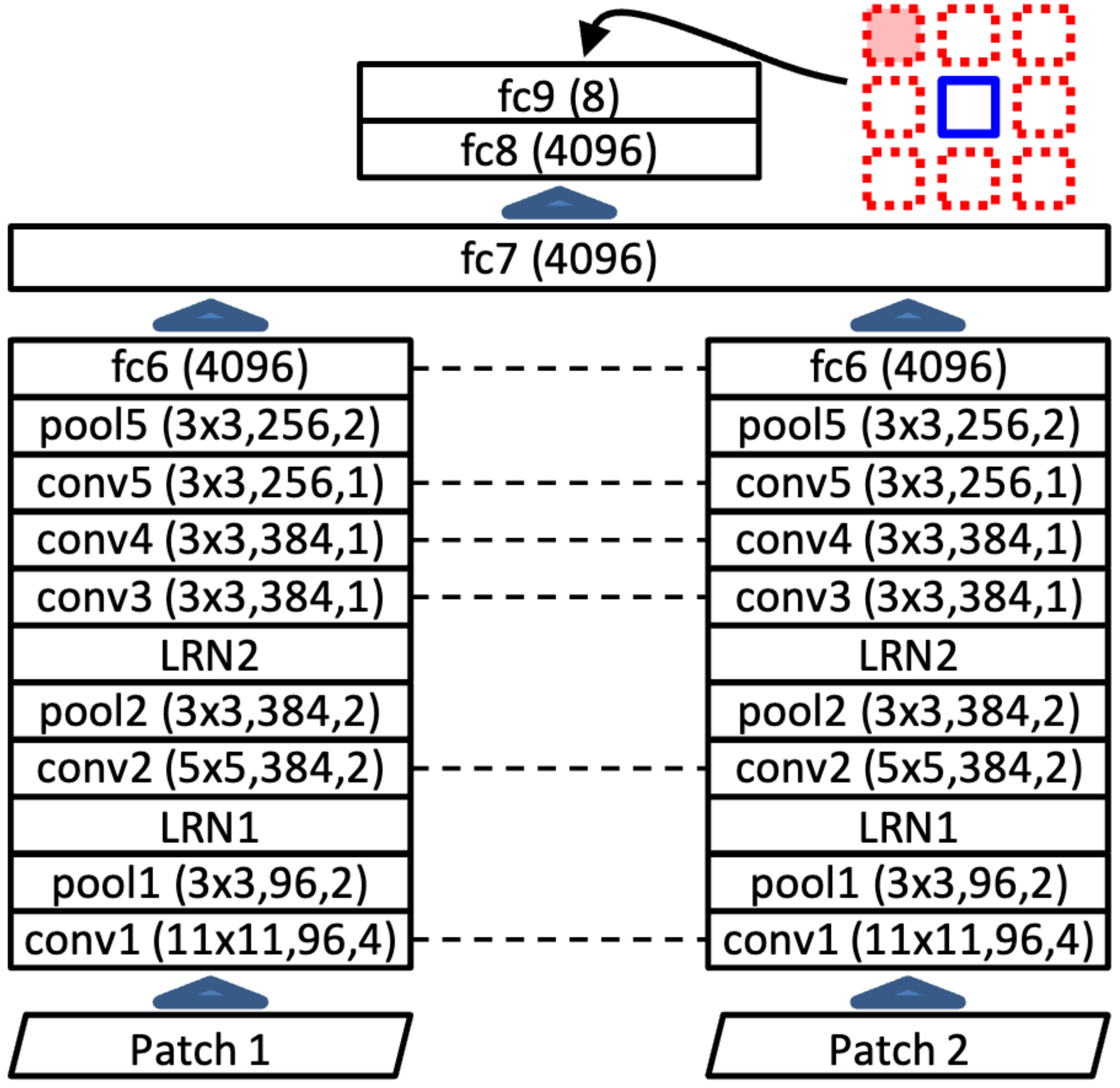
- **Idea.** Let the unlabeled data itself become a task
  - There are two major approaches:
    - Pretext task
    - Joint embedding

Doersch et al., "Unsupervised visual representation learning by context prediction" ICCV 2015

Pretext Task

- Train a model to solve a synthetic-but-useful task
- **Example (Context Prediction, 2015)**
  - Predict the relative location of a patch with respect to another

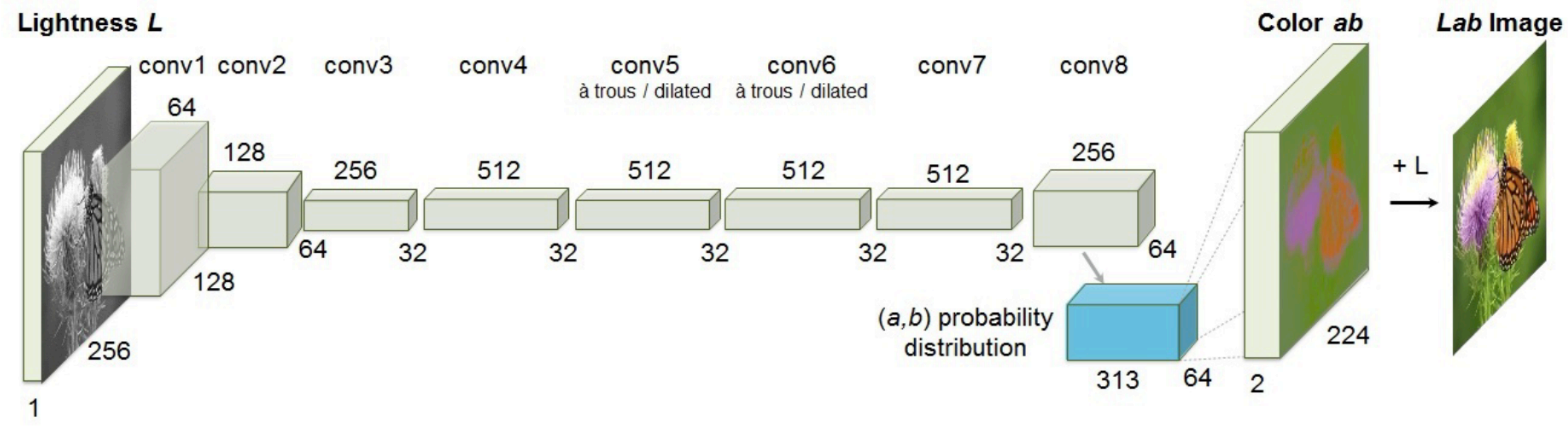




Zhang et al. "Colorful image colorization" ECCV 2016

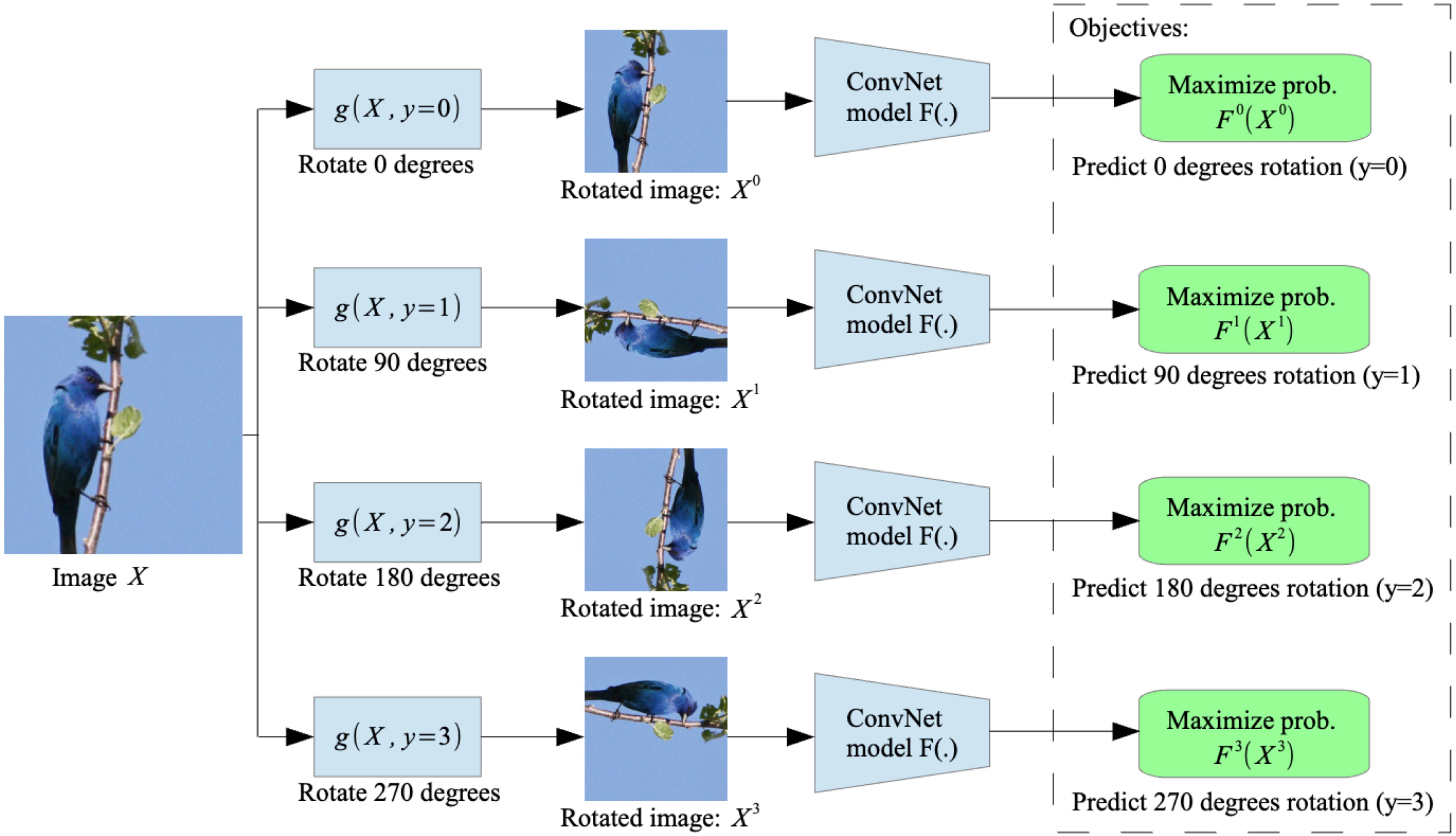
Pretext Task

- **Example (Colorization, 2016)**
  - Predict the color of each pixel, from the gray-scaled image



Pretext Task

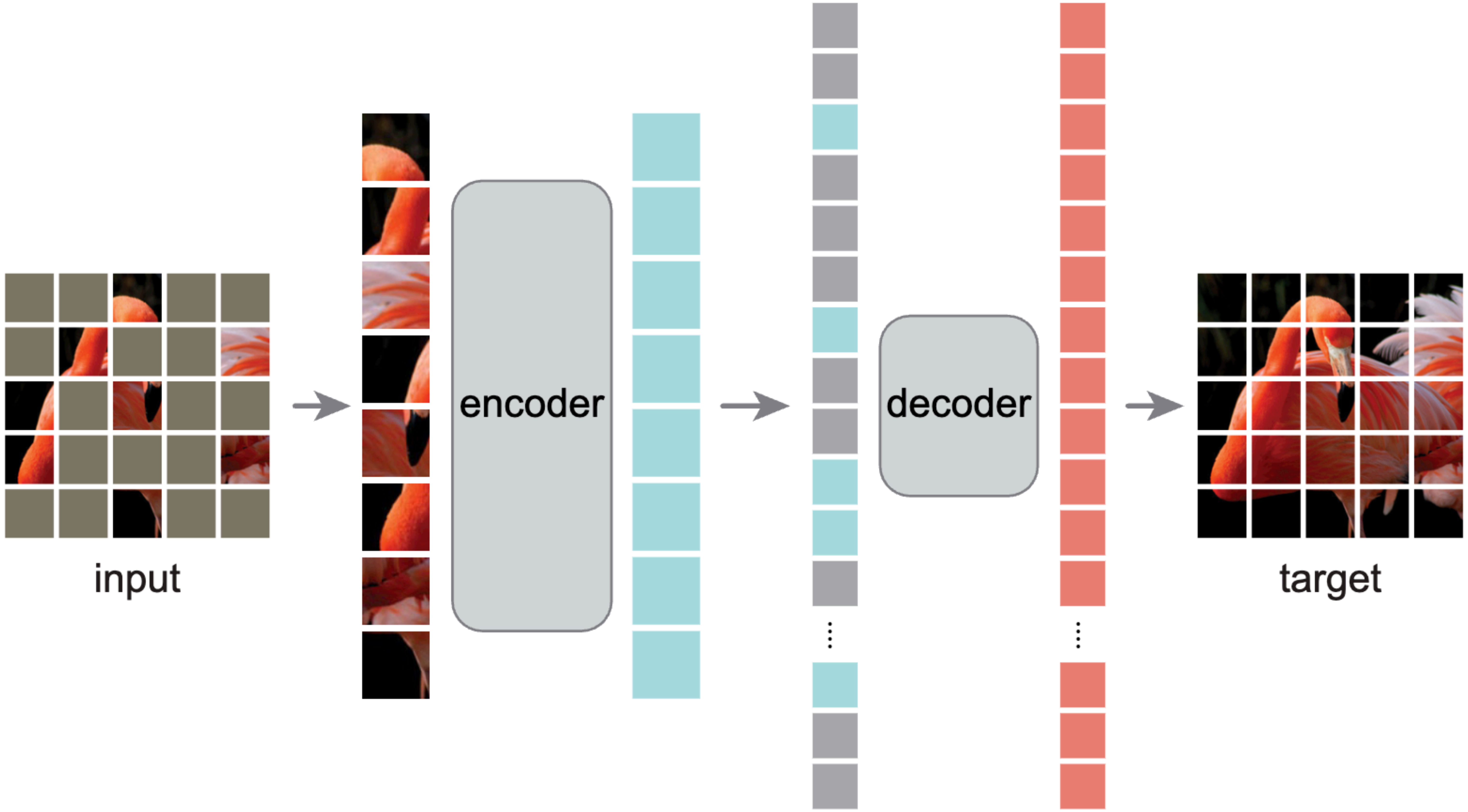
- **Example (Rotation, 2018)**
  - Predict how much the target image has been rotated



He et al. “Masked autoencoders are scalable vision learners,” CVPR 2022

Pretext Task

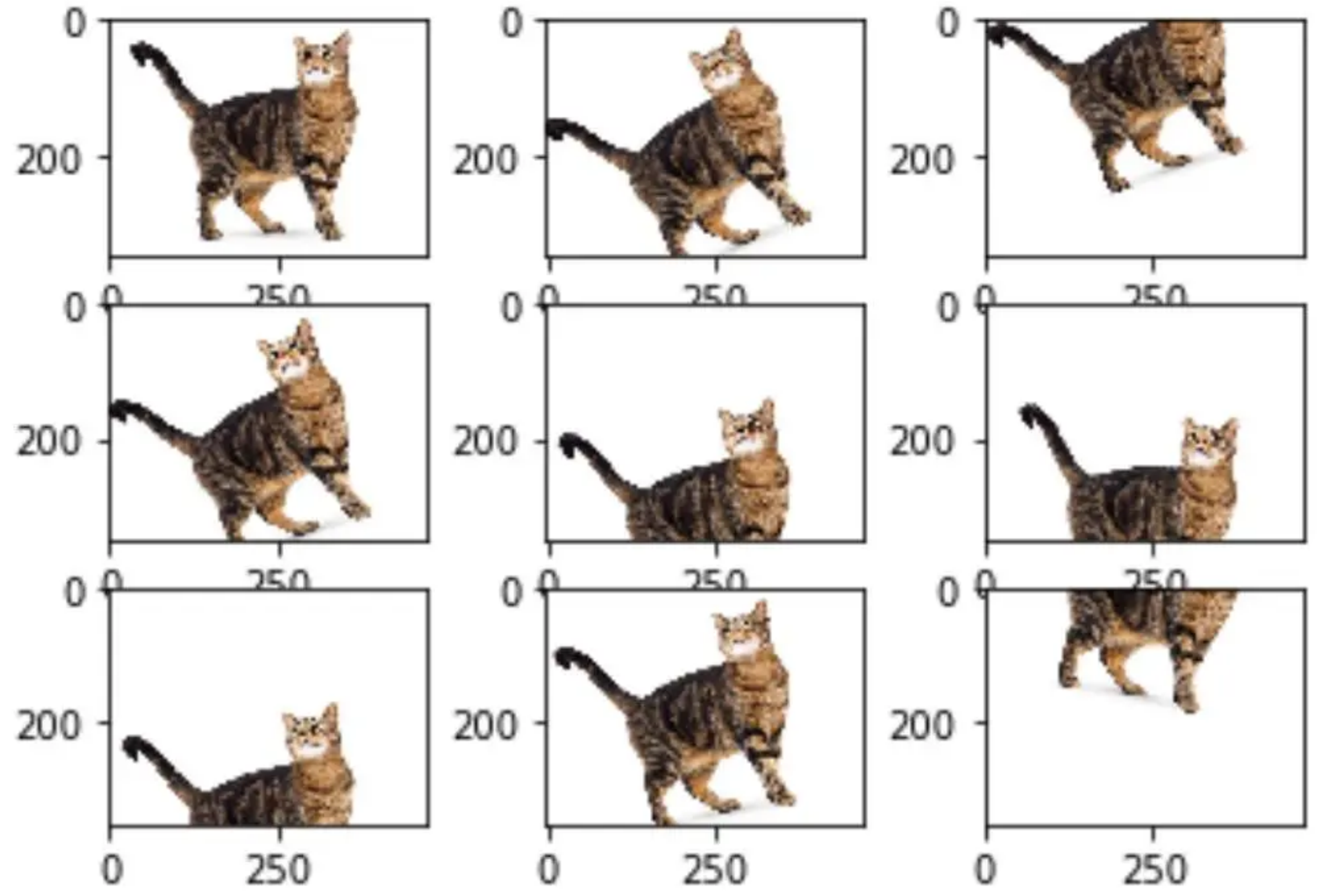
- **Example (Masked Modeling, 2022)**
  - Complete the masked-out details from the given image
    - we’ll revisit this later, requires knowing transformers



Joint embedding

- **Idea.** Train a model to be indifferent to certain operations
  - The representation of an image should be similar to that of a slighted perturbed image (positive sample)
- **Problem.** This becomes a minimization problem with a trivial solution

$$\min_{\theta} \mathbb{E}\|f_\theta(\mathbf{x}) - f_\theta(\mathbf{x}_{\text{perturbed}})\|^2$$

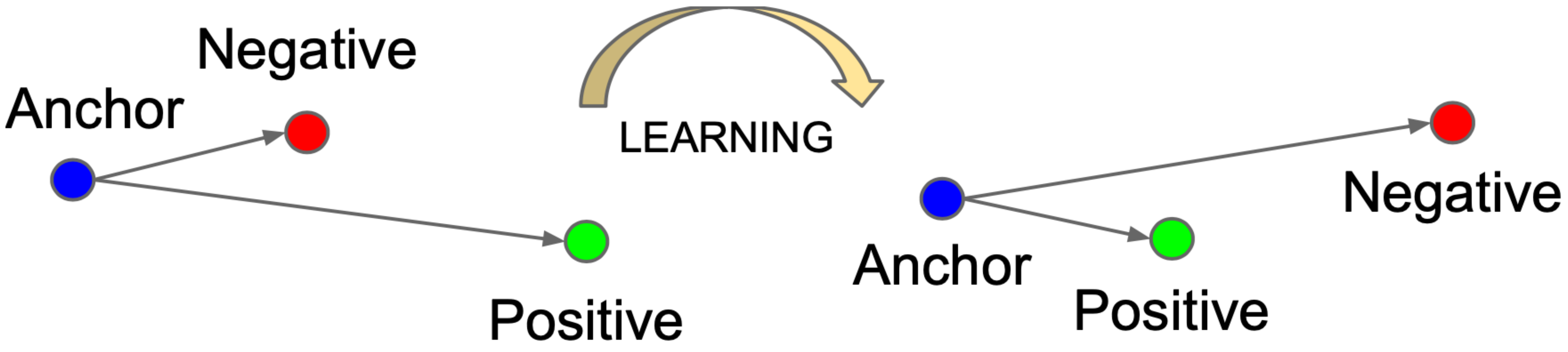


Joint embedding

- **Solution.** Add a negative sample — from which we want to maximize the distance
  - The optimization problem now becomes:

$$\min_{\theta} \left(\mathbb{E}\|f_\theta(\mathbf{x}) - f_\theta(\mathbf{x}_{\text{perturbed}})\|^2 - \mathbb{E}\|f_\theta(\mathbf{x}) - f_\theta(\mathbf{y})\|^2 \right)$$

    - The negative sample $\mathbf{y}$ is an independently drawn image (extremely less likely to be a cat)
    - called "triplet loss"



Chen et al., “A Simple Framework for Contrastive Learning of Visual Representations” ICML 2020

Joint embedding

- **Example (Simple Contrastive Learning, 2020)**
  - Draw a large batch of data: $\mathbf{x}_1, \ldots, \mathbf{x}_N$
  - Randomly augment each sample twice, and get their representations

$$\mathbf{x}_i \mapsto \mathbf{z}_{2k}, \mathbf{z}_{2k-1}$$

  - Use the cross-entropy-like loss

$$\ell_{i,j} = -\log \frac{\exp(\mathrm{sim}(\boldsymbol{z}_i, \boldsymbol{z}_j)/\tau)}{\sum_{k=1}^{2N} \mathbb{1}_{[k \neq i]} \exp(\mathrm{sim}(\boldsymbol{z}_i, \boldsymbol{z}_k)/\tau)}$$

    - Here, $\mathrm{sim}(\mathbf{u}, \mathbf{v}) = \mathbf{u}^\top \mathbf{v} / \|\mathbf{u}\| \, \|\mathbf{v}\|$ is the cosine similarity.

Joint embedding

- **Example (Simple Contrastive Learning, 2020)**
  - The total loss will be:

$$\mathcal{L} = \frac{1}{2N} \sum_{k=1}^{N} \left[\ell(2k-1, 2k) + \ell(2k, 2k-1) \right]$$

Self-supervised learning

- **As of now.** Masked modeling is slightly more preferred option over others
  - Joint embedding requires a large-batch training
    - Very memory-heavy
  - Joint embedding heavily relies on human-derived concepts

Next up

- Visual generative models

</lecture 16>