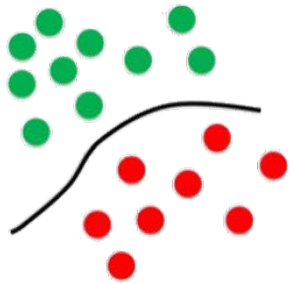
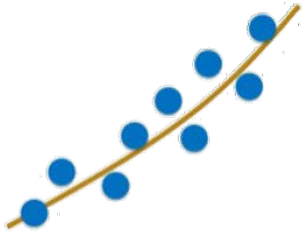
Metric Learning for Computer Vision

Some slides from Björn Ommer, Laura Leal-Taixé, Ting Chen, Jiachang Liu, Avi Kak and Charles Bouman

Grand Goal of Machine Learning in CV

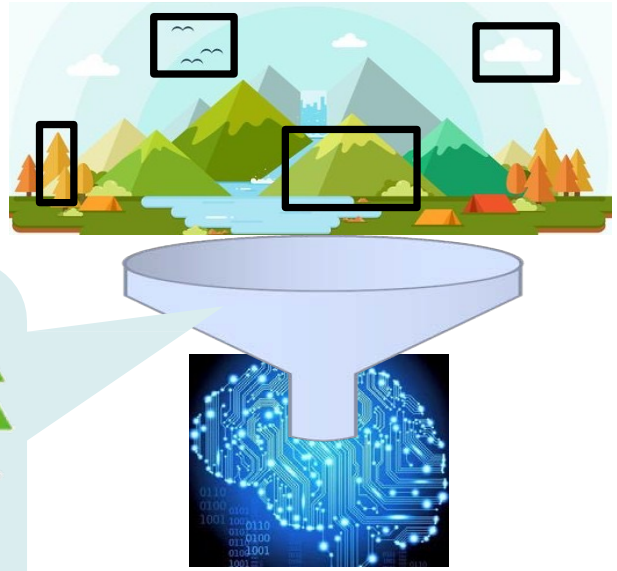
~~Learning to solve a task~~





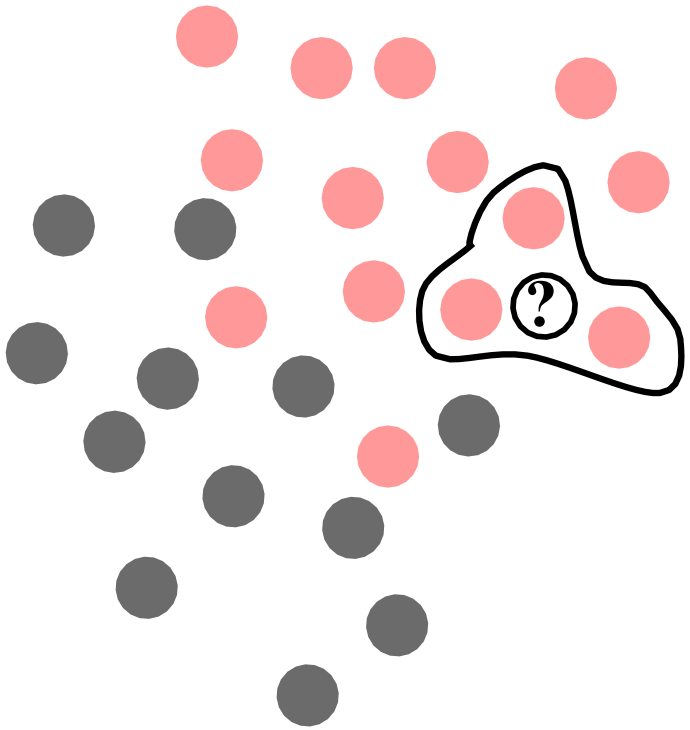
Classification Regression …

Learning a representation of the world

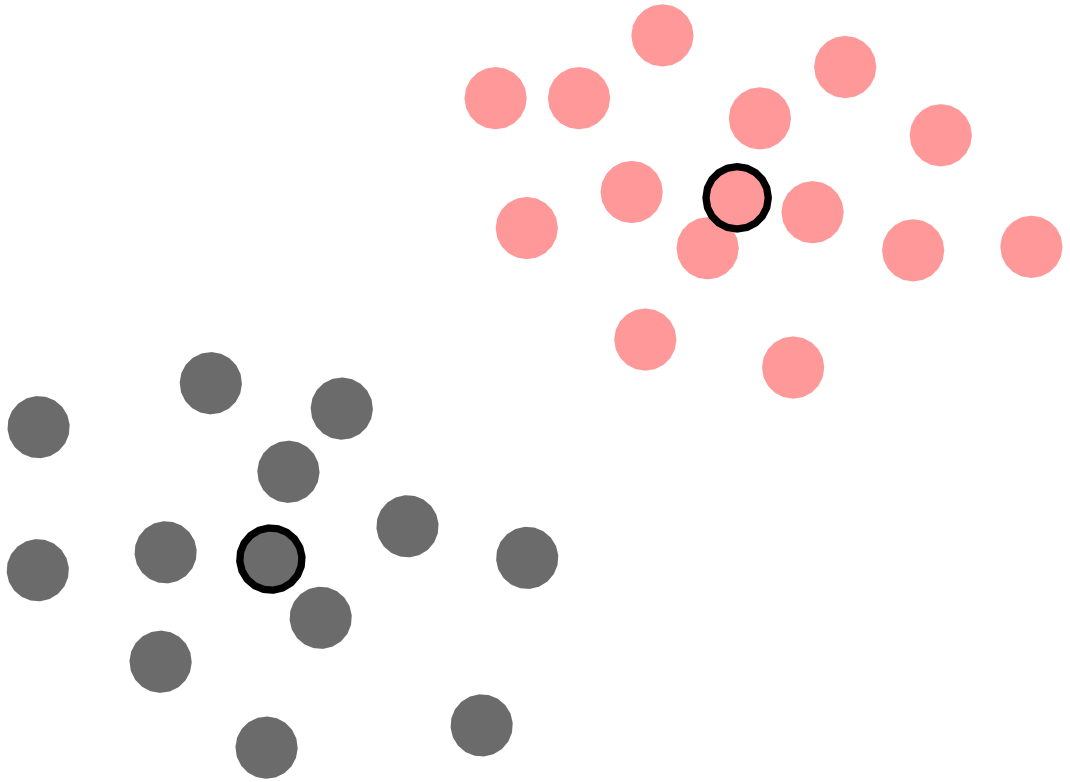


& its semantic interdependencies

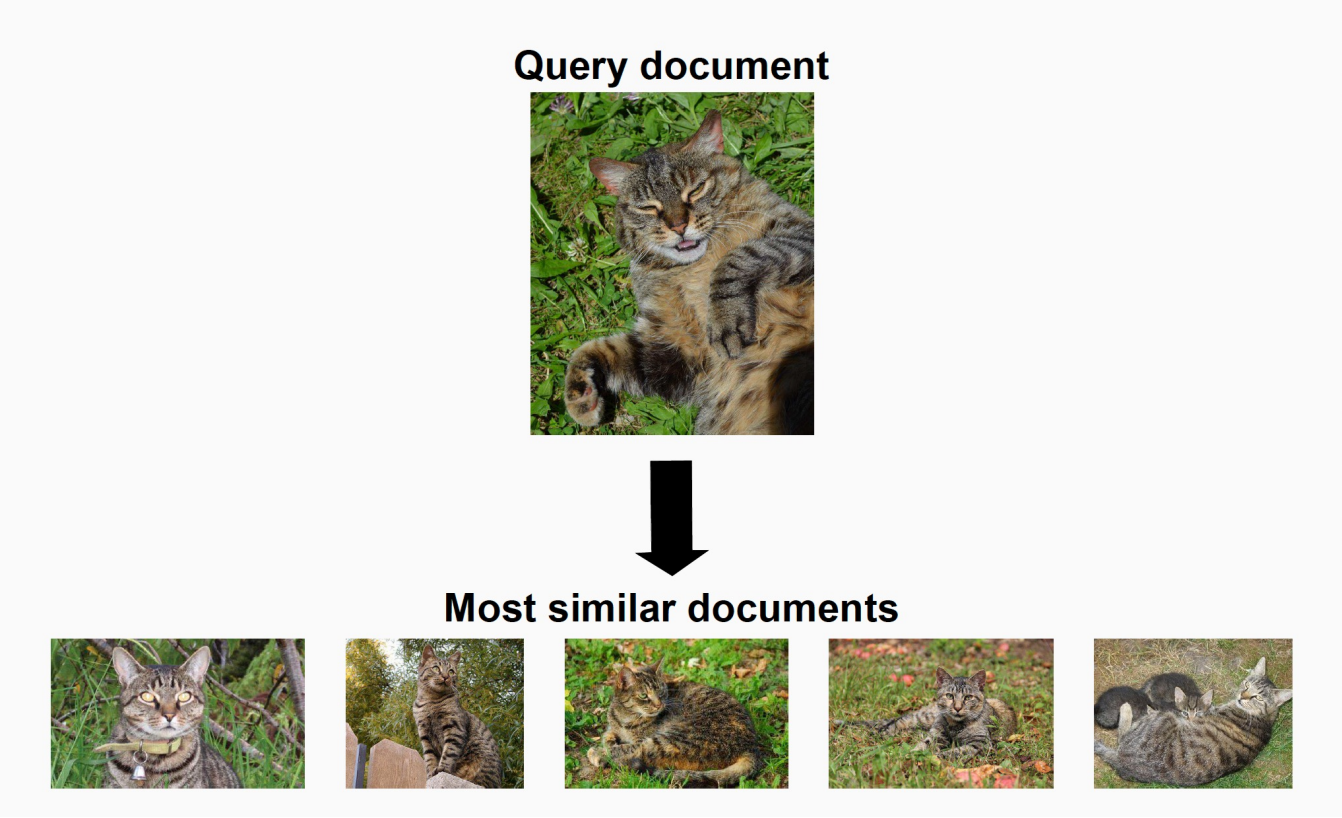
Relation matters: Nearest neighbor classification



Relation matters: grouping



Relation matters: retrieval

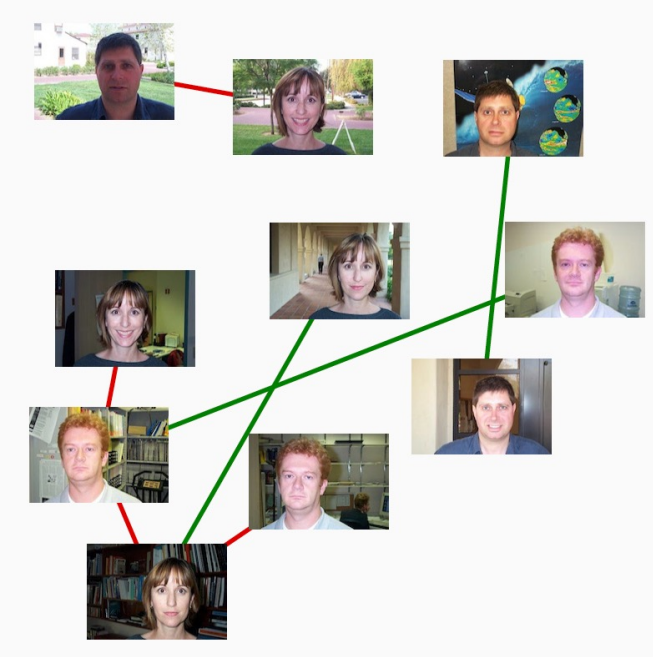


What is metric learning?

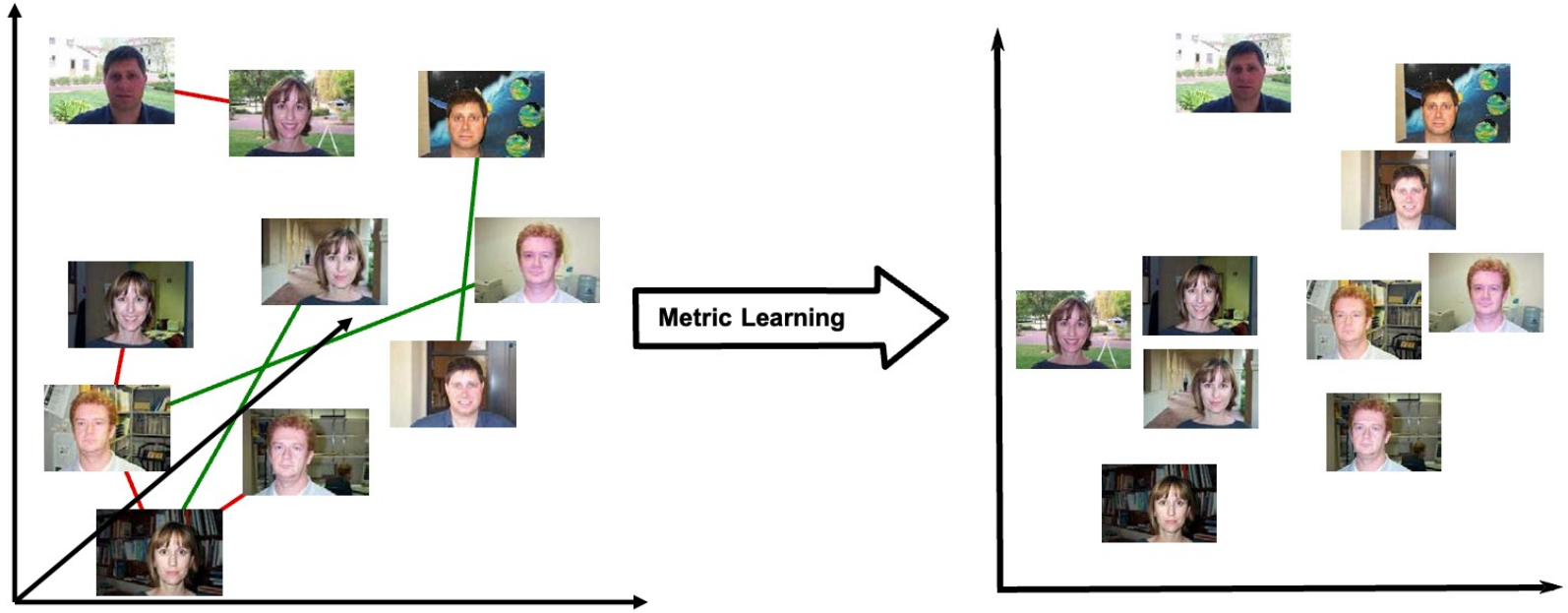
- Metric learning is about representing your data objects, such as images or text or whatever, with numerical vectors such that the data objects that are similar acquire vector representations that are close together. By the same token, we would want the dissimilar data objects to acquire representations that are far apart.

What exactly are dissimilar or similar objects?

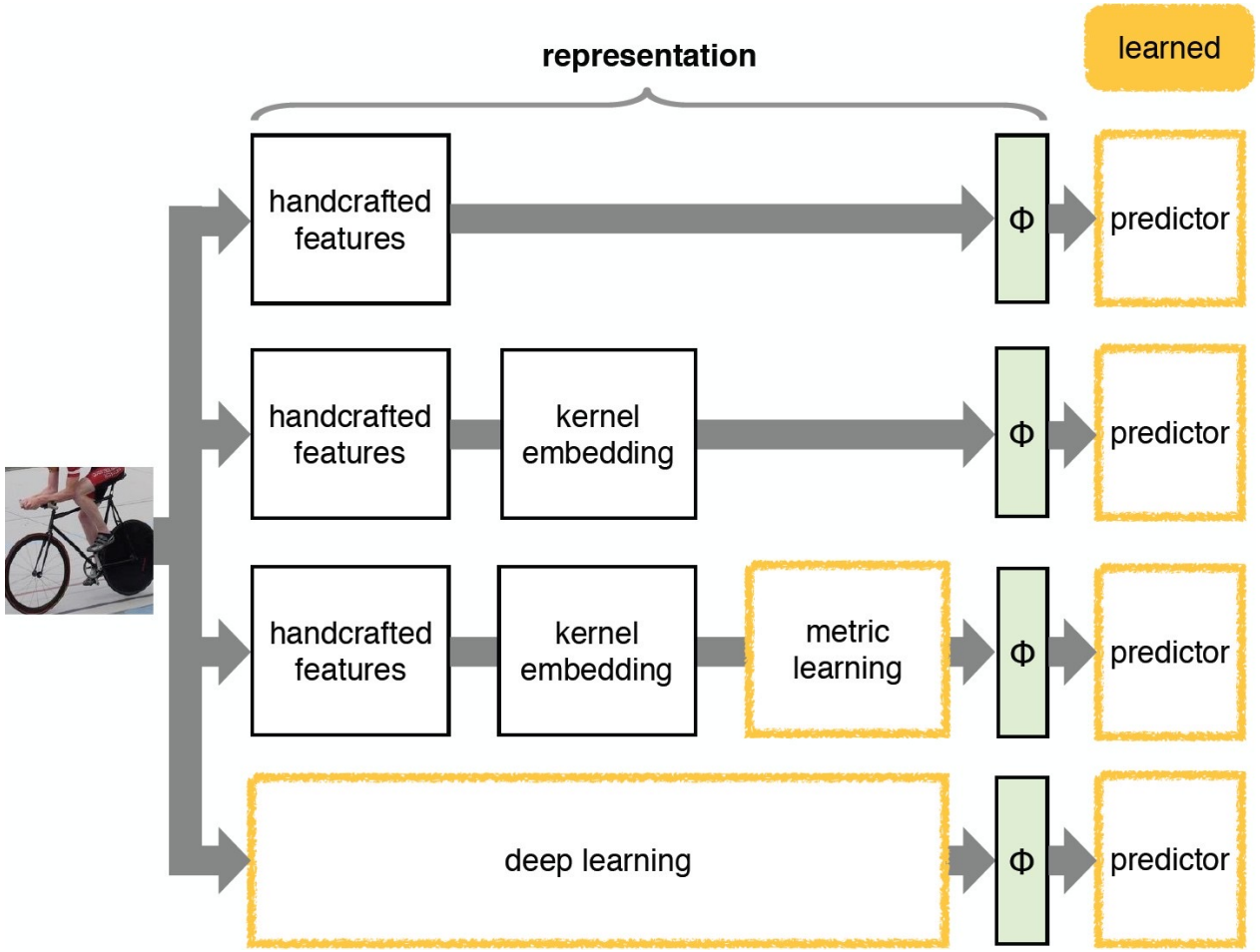
- In the domain of face images—a problem domain that has proved fertile for the development of metric learning algorithms—we would want to consider all the face images of the same human subject to be similar regardless of the pose of the head w.r.t. the camera.



Metric Learning

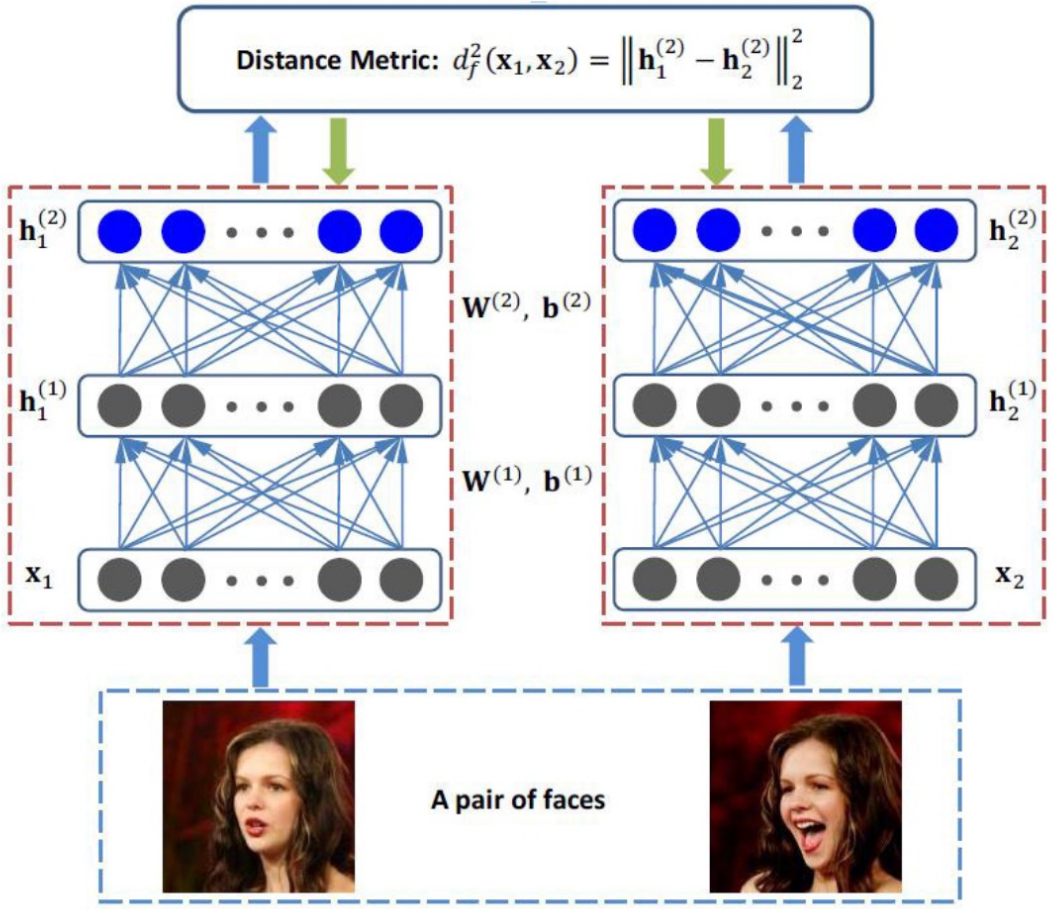


Metric Learning and Deep Learning



Credit: A. Vedaldi

Deep Metric Learning



Similarity Learning: when and why?

- Application: face recognition system so students can enter the exam room without the need for ID check

Training

Person 1

Person 2

Person 3

Losses for Deep Metric Learning

Pairwise Contrastive Loss and Triplet Loss

Pairwise Contrastive Loss

- As its name implies, Pairwise Contrastive Loss requires that a batch be reorganized in the form of pairs of training samples. There are two types of pairs to be constructed: Positive Pairs and Negative Pairs. Both items in a Positive Pair carry the same class label and the two items in a Negative Pair must carry different class labels. The piece of code that converts a batch into such pairs is called a miner.

Similarity learning: losses

- Distance function $d(A, B) = ||f(A) - f(B)||^2$
- Training: learn the parameter such that
  - If A and B depict the same person, $d(A, B)$ is small
  - If A and B depict a different person, $d(A, B)$ is large

Similarity learning: losses

- Loss function for a positive pair:
  - If A and B depict the same person, $d(A, B)$ is small

$$L(A, B) = ||f(A) - f(B)||^2$$

Similarity learning: losses

- Loss function for a negative pair:
  - If A and B depict a different person, $d(A, B)$ is large
  - Better use a Hinge loss:

margin

$$L(A, B) = \max(0, m^2 - ||f(A) - f(B)||^2)$$

If two elements are already far away, do not spend energy in pulling them even further away

Similarity learning: losses

- Contrastive loss:

$$L(A, B) = y||f(A) - f(B)||^2 + (1 - y)max(0, m^2 - ||f(A) - f(B)||^2)$$

Positive pair, reduce the distance between the elements

Negative pair, brings the elements further apart up to a margin

Chopra, Hadsell and LeCun. “Learning a similarity metric discriminatively, with application to Face verification”, CVPR 2005.

Triplet loss

- Triplet loss allows us to learn a ranking

Anchor (A)

Positive (P)

Negative (N)

We want: $||f(A) - f(P)||^2 < ||f(A) - f(N)||^2$

Schroff et al „FaceNet: a unified embedding for face recognition and clustering“. CVPR 2015

Triplet loss

- Triplet loss allows us to learn a ranking

$$||f(A) - f(P)||^2 < ||f(A) - f(N)||^2$$

$$||f(A) - f(P)||^2 - ||f(A) - f(N)||^2 < 0$$

$$||f(A) - f(P)||^2 - ||f(A) - f(N)||^2 + m < 0$$

margin

Schroff et al „FaceNet: a unified embedding for face recognition and clustering“. CVPR 2015

Triplet loss

- Triplet loss allows us to learn a ranking
  - $||f(A) - f(P)||^2 < ||f(A) - f(N)||^2$
  - $||f(A) - f(P)||^2 - ||f(A) - f(N)||^2 < 0$
  - $||f(A) - f(P)||^2 - ||f(A) - f(N)||^2 + m < 0$
- $L(A, P, N) = max(0, ||f(A) - f(P)||^2 - ||f(A) - f(N)||^2 + m)$

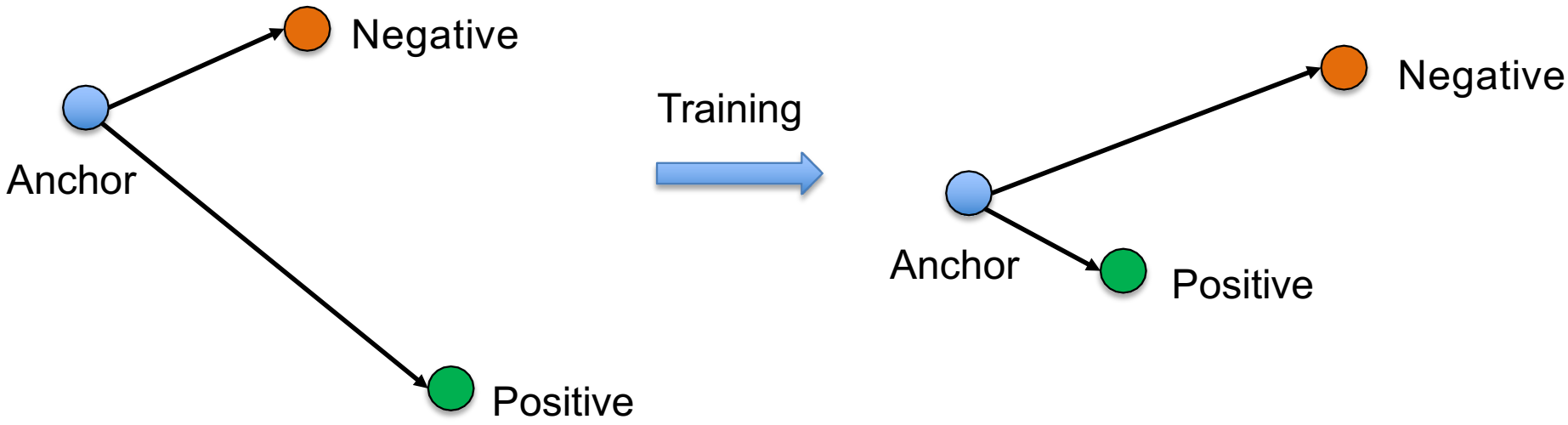
Schroff et al „FaceNet: a unified embedding for face recognition and clustering“. CVPR 2015

Triplet loss

- Triplet loss allows us to learn a ranking
  - If $d(A, P) > d(A, N)$ the loss is going to be positive
  - If $d(A, P) < d(A, N)$ then I look at the margin
- $L(A, P, N) = max(0, ||f(A) - f(P)||^2 - ||f(A) - f(N)||^2 + m)$

Schroff et al „FaceNet: a unified embedding for face recognition and clustering“. CVPR 2015

Triplet loss

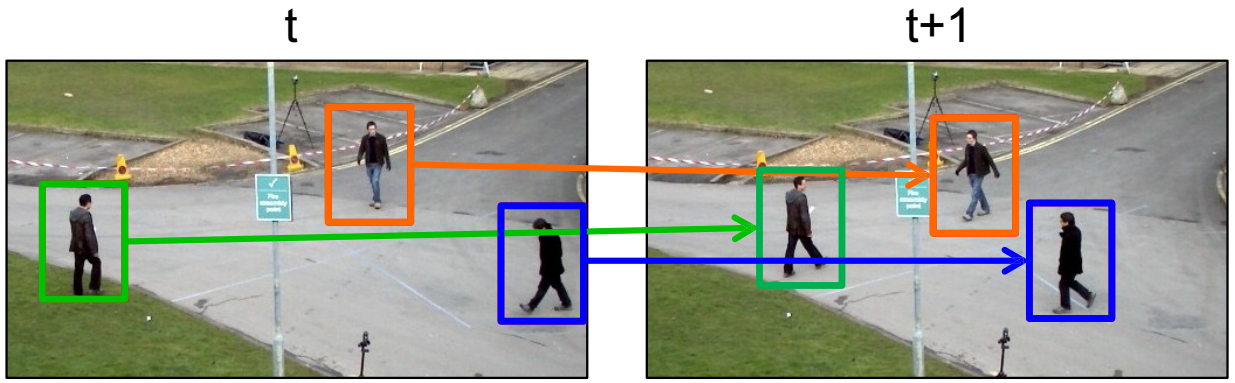


Applications of Metric Learning

1. Object Re-ID
2. Unsupervised Representation Learning
3. CLIP

Object Tracking

- Given a video, find out which parts of the image depict the same object in different frames
- Often we use detectors as starting points



Why do we need tracking?

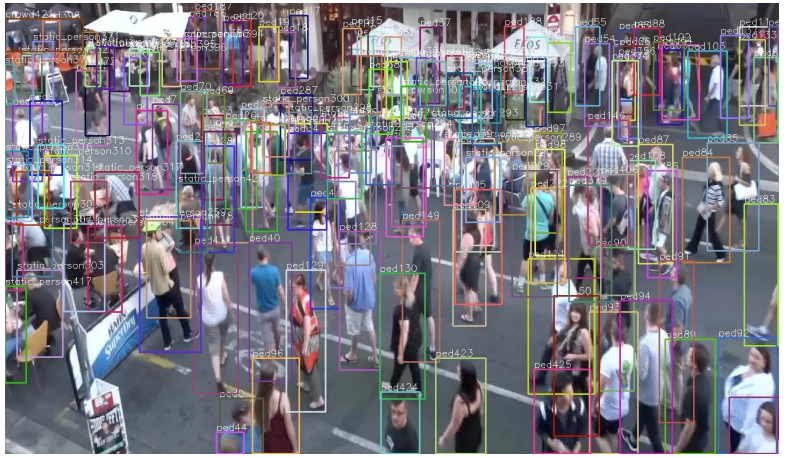
- To model objects when detection fails:
  - Occlusions
  - Viewpoint/pose/blur/illumination variations (in a few frames of a sequence)
  - Background clutter
- To reason about the dynamic world, e.g., trajectory prediction (is the person going to cross the street?)

Tracking is…

- Similarity measurement
- Correlation
- Correspondence
  - Story time: „A young graduate student asked Takeo Kanade what are the three most important problems in computer vision. Kanade replied: “Correspondence, correspondence, correspondence!”
- Matching/retrieval
- Data association

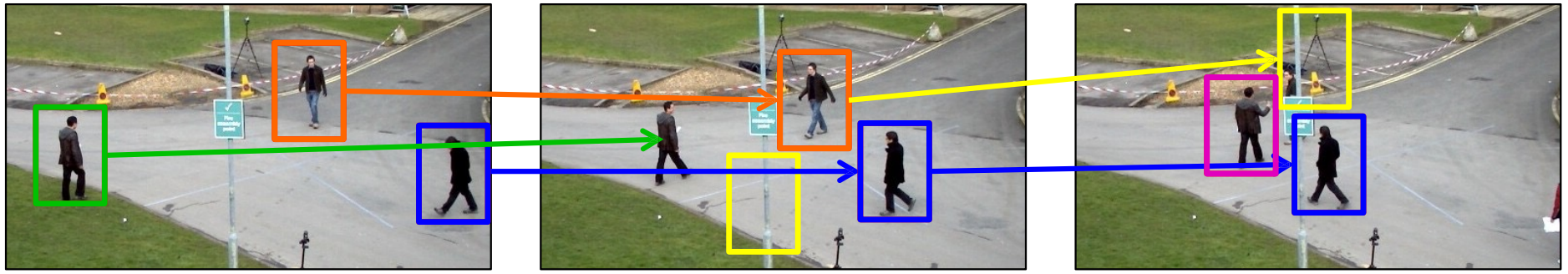
Different challenges

- Multiple objects of the same type
- Heavy occlusions
- Appearance is often very similar



Tracking-by-detection

- We will focus on algorithms where a set of detections is provided
  - Remember detections are not perfect!

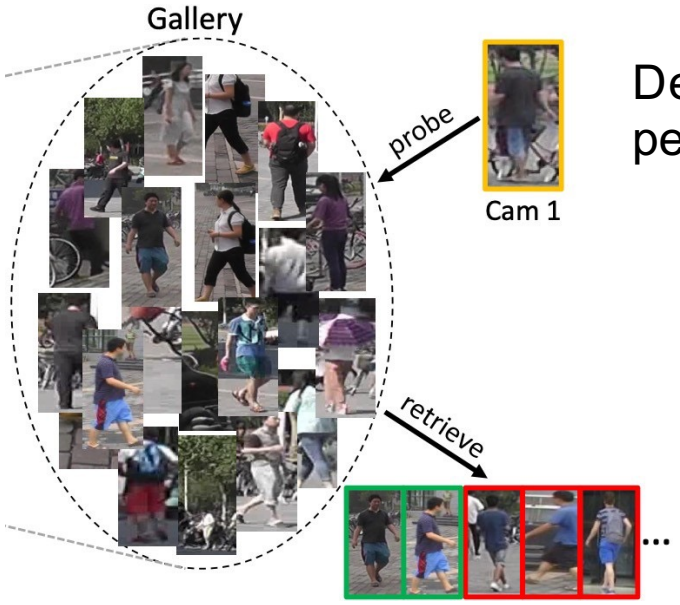


Find detections that match and form a trajectory

Object Re-ID Problem statement

- Viewing tracking as a retrieval problem

Images of different identities



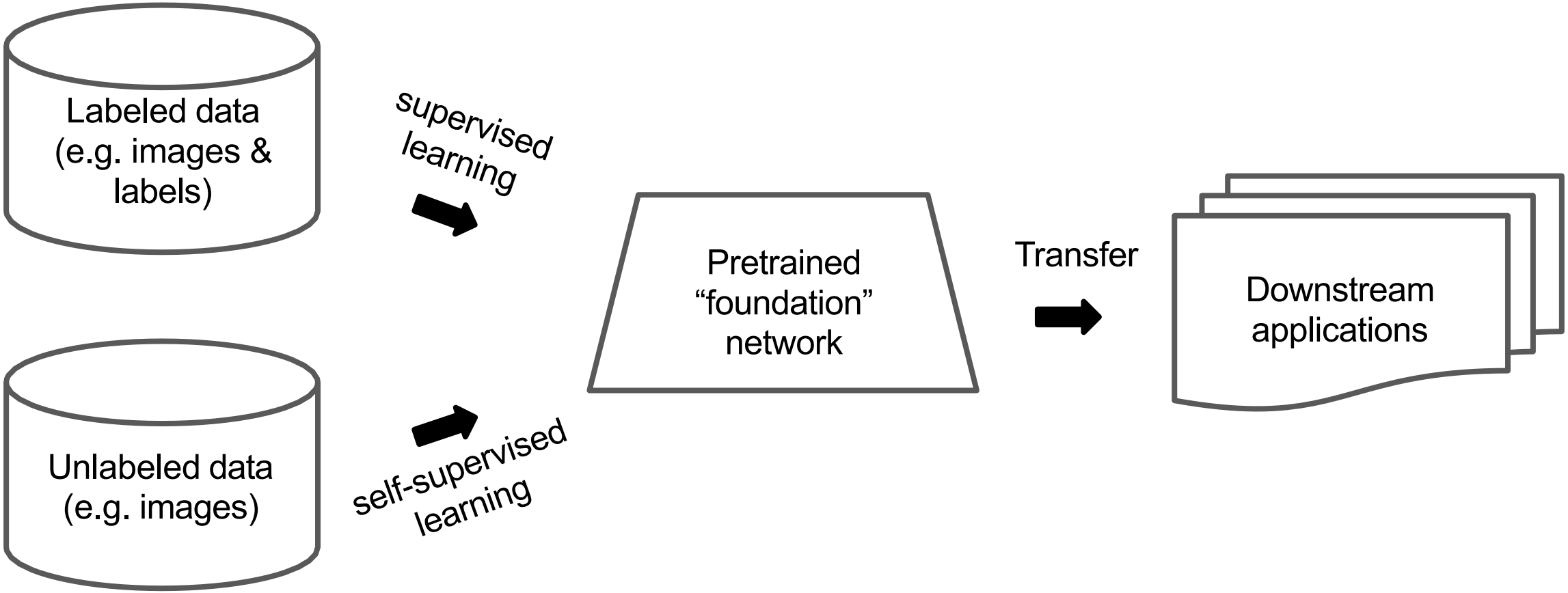
Detected person

Retrieve the top matches

How to measure (learn) the distance between two images?

Motivation for Contrastive Learning

The paradigm of learning “foundation” models



“Self-supervised learning” is “supervised learning” without specific task annotations.

SimCLR

[Chen et al, A simple framework for contrastive learning of visual representations, ICML'20]

Maximizing the agreement of representations under data transformation, using a contrastive loss in the latent/feature space.

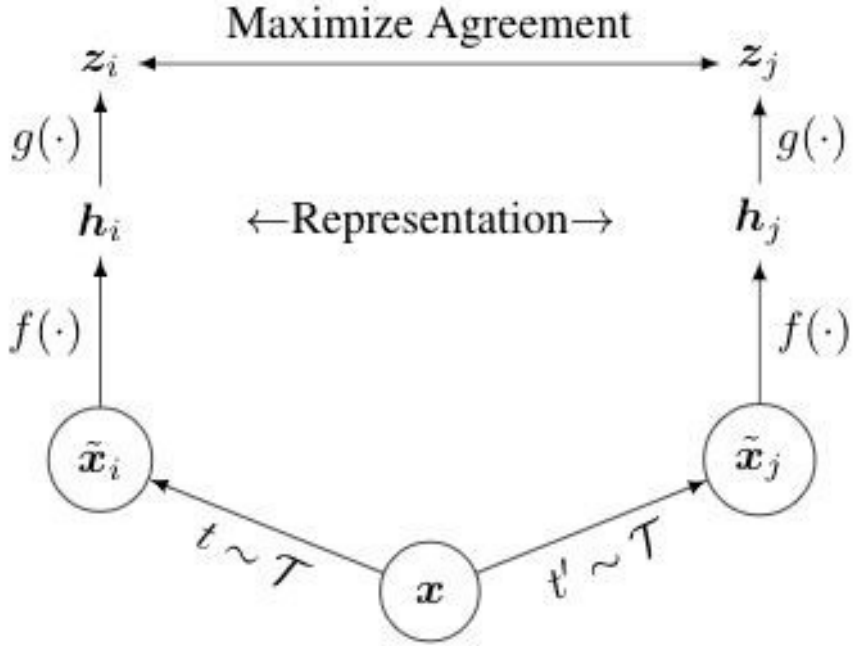
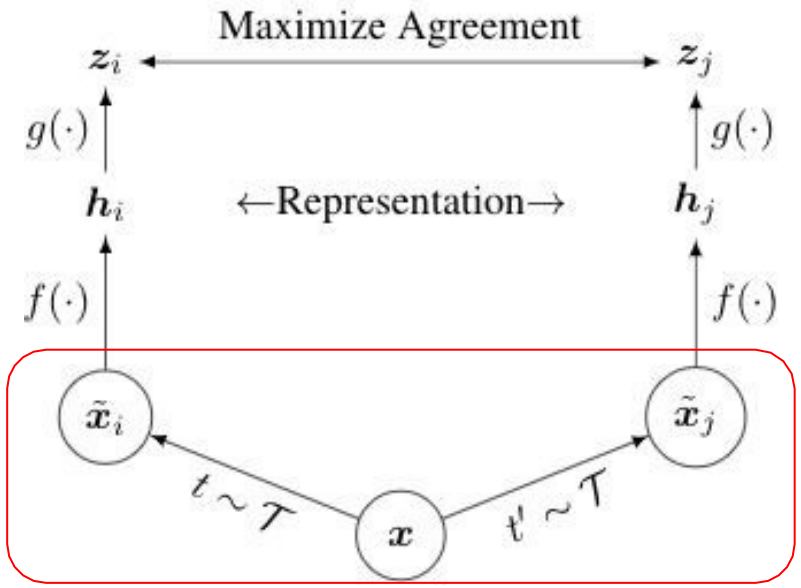


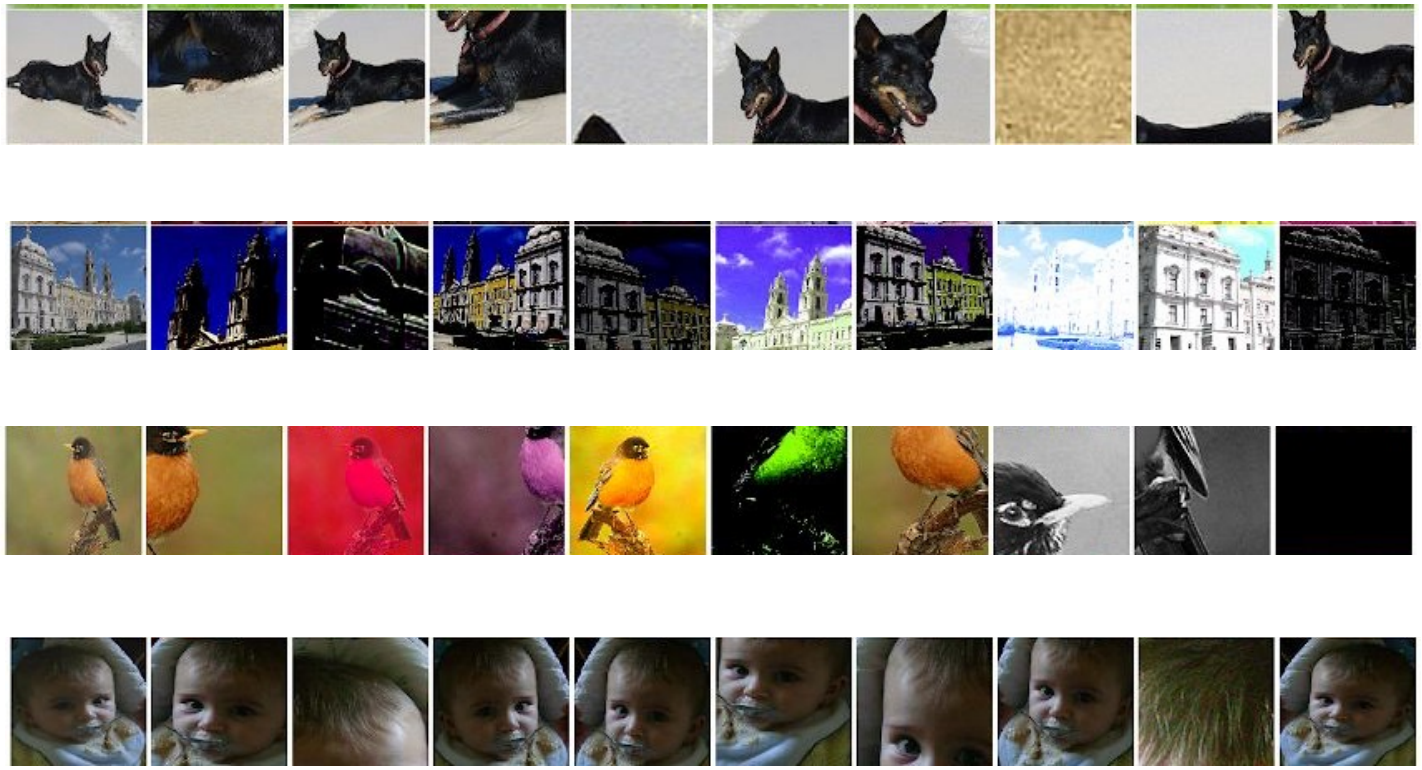
Figure 2. A framework for contrastive representation learning. Two separate stochastic data augmentations $t, t' \sim \mathcal{T}$ are applied to each example to obtain two correlated views. A base encoder network $f(\cdot)$ with a projection head $g(\cdot)$ is trained to maximize agreement in *latent representations* via a contrastive loss.

SimCLR component: data augmentation

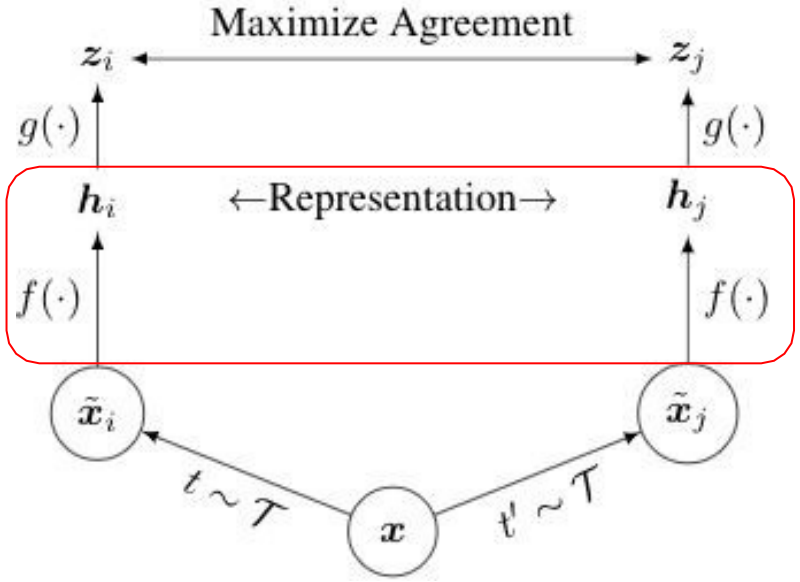
We use random crop and color distortion for augmentation.

Examples of augmentation applied to the left most images:



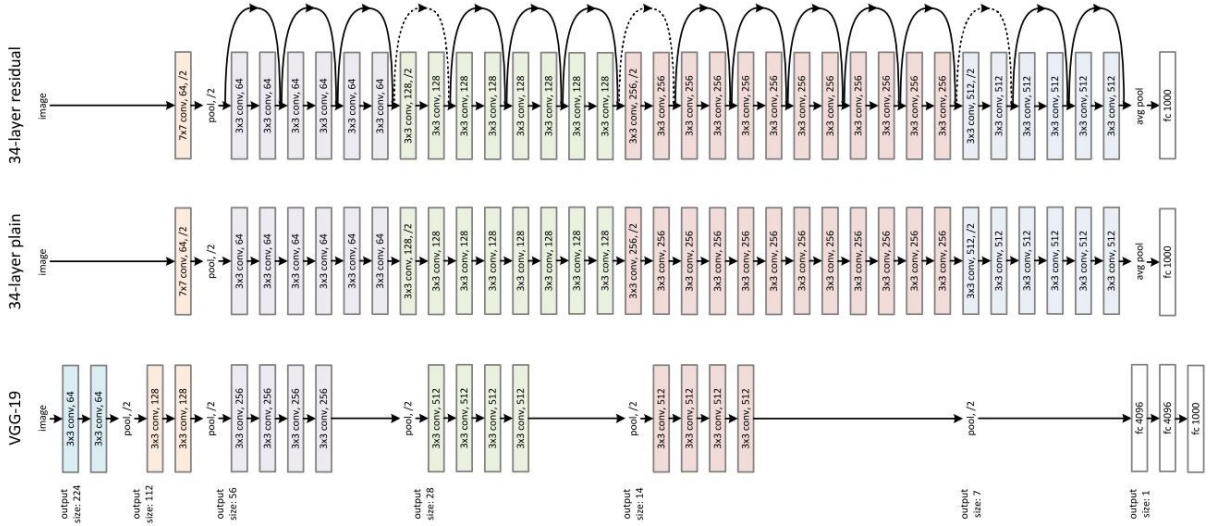


SimCLR component: encoder

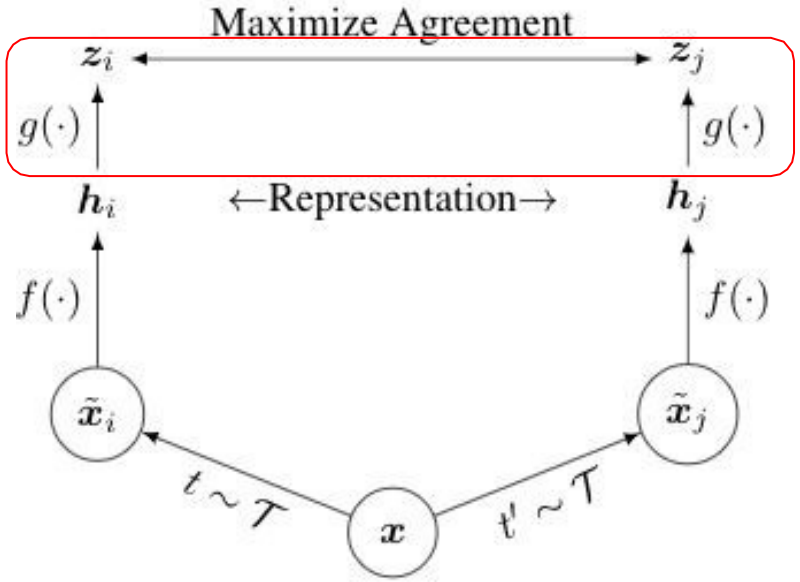


f(x) is the base network that computes internal representation.

We use (unconstrained) ResNet in this work. However, it can be other networks.

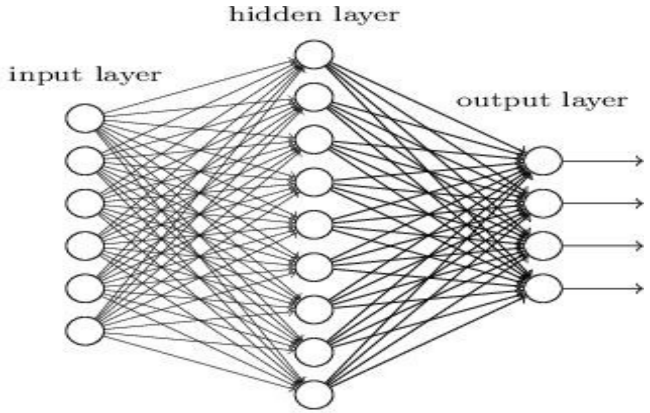


SimCLR component: projection head

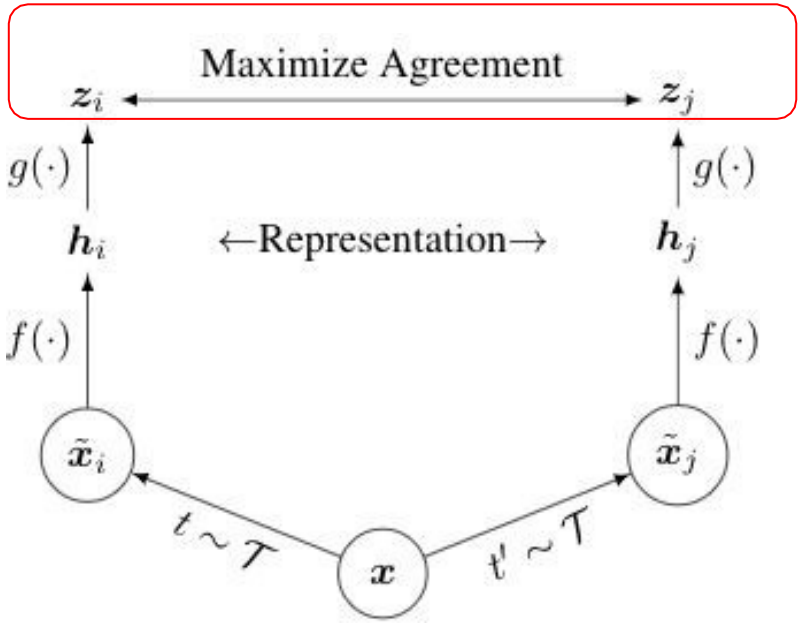


g(h) is a projection network that project representation to a latent space.

We use a MLP (with non-linearity).



SimCLR component: contrastive loss



Maximize agreement using a contrastive task:

Given {x_k} where two different examples x_i and x_j are a positive pair, identify x_j in {x_k}_{k!=i} for x_i.

Original image crop 1 crop 2 contrastive image

Loss function:

$$\text{Let } \text{sim}(\boldsymbol{u}, \boldsymbol{v}) = \boldsymbol{u}^\top \boldsymbol{v} / \|\boldsymbol{u}\| \|\boldsymbol{v}\|$$

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

SimCLR pseudo code and illustration

Algorithm 1 SimCLR's main learning algorithm.

input: batch size N, temperature τ, form of $f, g, \mathcal{T}$.
for sampled mini-batch $\{\boldsymbol{x}_k\}_{k=1}^{N}$ **do**
  for all $k \in \{1, \ldots, N\}$ **do**
    draw two augmentation functions $t \sim \mathcal{T}$, $t' \sim \mathcal{T}$
    # the first augmentation
    $\tilde{\boldsymbol{x}}_{2k-1} = t(\boldsymbol{x}_k)$
    $\boldsymbol{h}_{2k-1} = f(\tilde{\boldsymbol{x}}_{2k-1})$ # representation
    $\boldsymbol{z}_{2k-1} = g(\boldsymbol{h}_{2k-1})$ # projection
    # the second augmentation
    $\tilde{\boldsymbol{x}}_{2k} = t'(\boldsymbol{x}_k)$
    $\boldsymbol{h}_{2k} = f(\tilde{\boldsymbol{x}}_{2k})$ # representation
    $\boldsymbol{z}_{2k} = g(\boldsymbol{h}_{2k})$ # projection
  end for
  for all $i \in \{1, \ldots, 2N\}$ and $j \in \{1, \ldots, 2N\}$ **do**
    $s_{i,j} = \boldsymbol{z}_i^\top \boldsymbol{z}_j / (\tau \|\boldsymbol{z}_i\| \|\boldsymbol{z}_j\|)$ # pairwise similarity
  end for
  define $\ell(i, j)$ **as** $-s_{i,j} + \log \sum_{k=1}^{2N} \mathbb{1}_{[k \neq i]} \exp(s_{i,k})$
  $\mathcal{L} = \frac{1}{2N} \sum_{k=1}^{N} [\ell(2k-1, 2k) + \ell(2k, 2k-1)]$
  update networks f and g to minimize $\mathcal{L}$
end for
return encoder network f

GIF credit: Tom Small

What Is CLIP and What Can CLIP Do?

- CLIP stands for Contrastive Language–Image Pre-training.
- It is a network that can be directly used for **image classification**.
- It is suitable for **zero-shot learning**. This network does not require fine-tuning when predicting labels on new images.
- The classification accuracy is **more robust** across a wide range of image datasets. This is crucial because well-trained models sometimes perform poorly during the real-world deployment.

Where Have We Been in Image Classification?

- Supervised Learning
  - Training with labels on ImageNet and fine-tune on a new dataset
  - Standard ResNet architecture
  - Pick features too narrowly and specifically for the ImageNet dataset
- Unsupervised Learning
  - Training with no labels. Contrastive learning and data augmentation are used on “much much” large scale of available data
  - MoCo and SimCLR
  - Pick features too broadly. Labels are effective signals to learn features.
- “Weakly-supervised” Learning
  - Training with text descriptions instead of labels. There are also abundant data of image and text description pairs on the web.
  - VirTex, ICMLM, ConVIRT, CLIP...
  - These text descriptions can be good signals to lead neural networks to learn features.

How to Train CLIP?

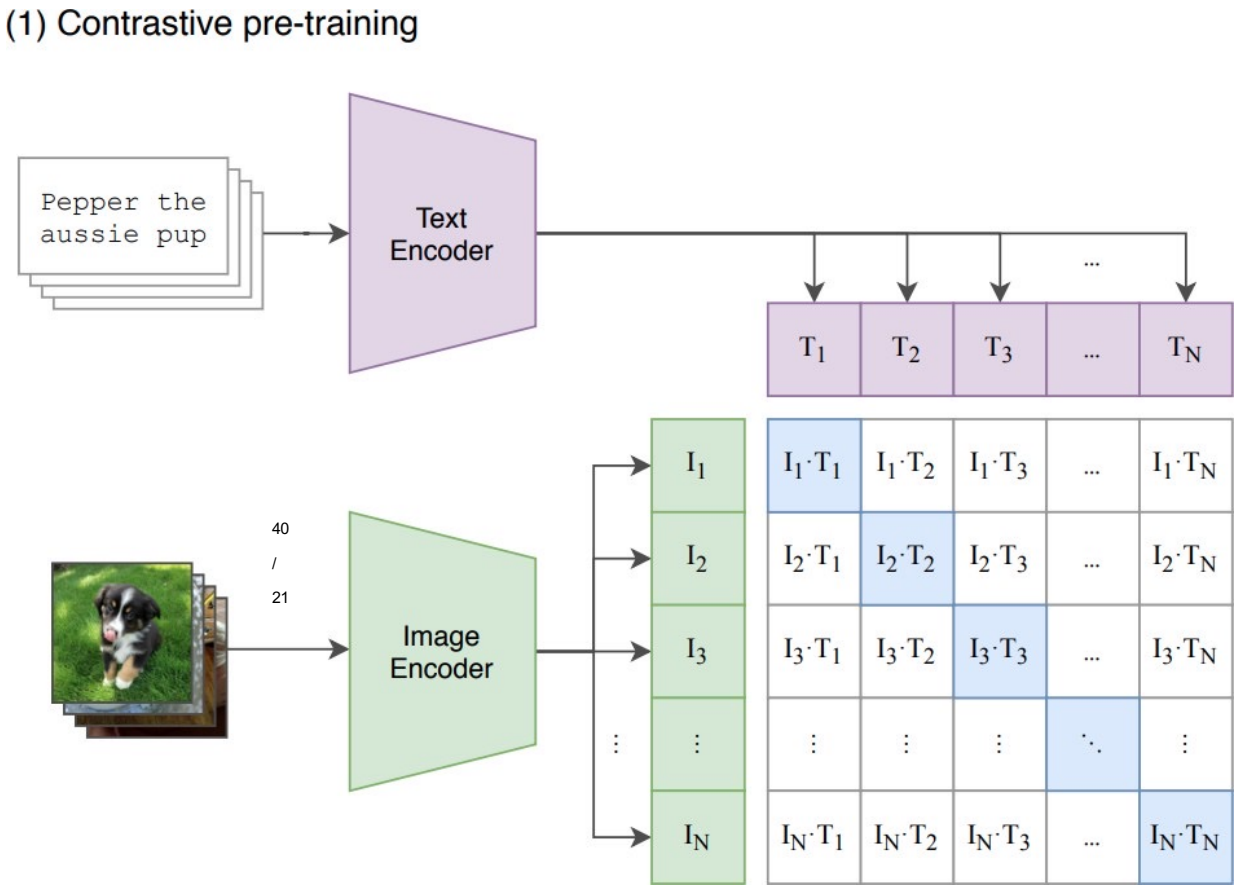


Figure: Contrastive Pre-training of language-image pairs. The text encoder is a standard transformer encoder. The extracted feature is the embedding of the CLS token. The image encoder is either a ResNet-50 or a Vision Transformer (ViT).

How to Use CLIP for Classification?

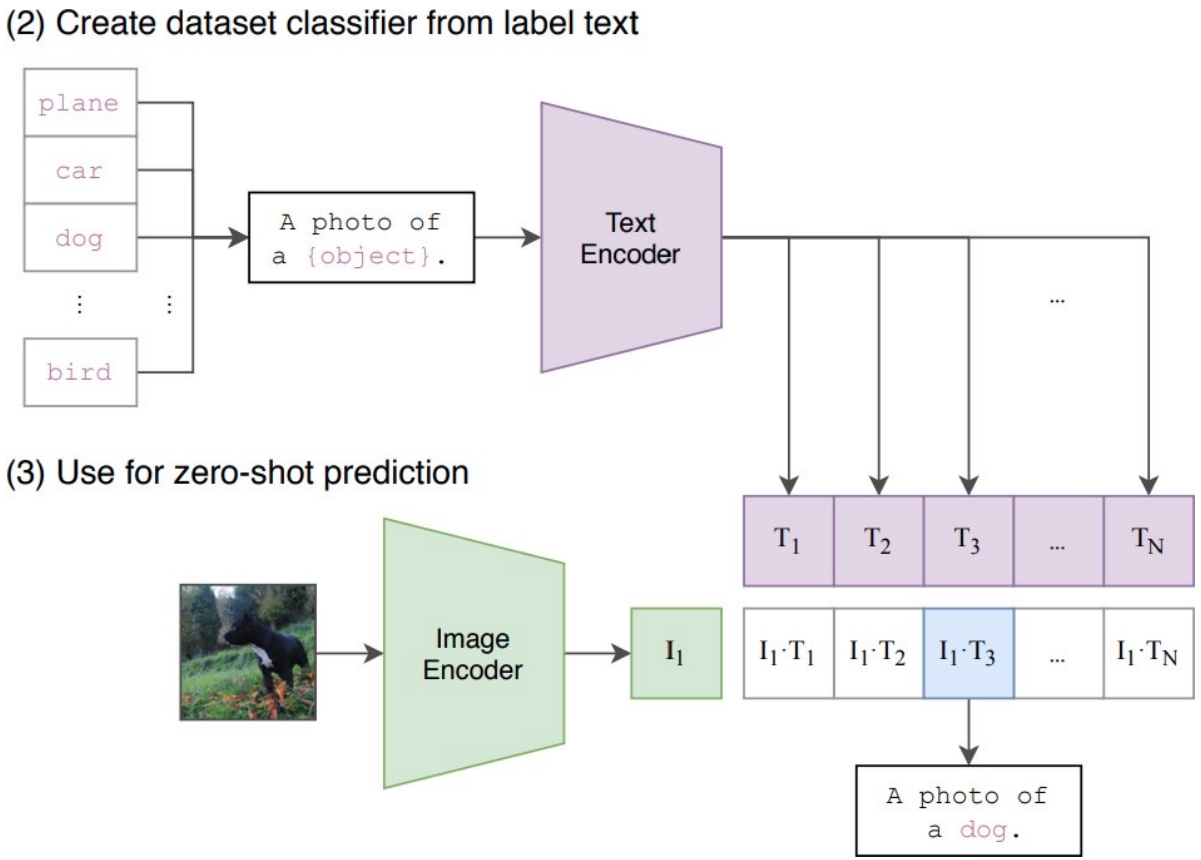
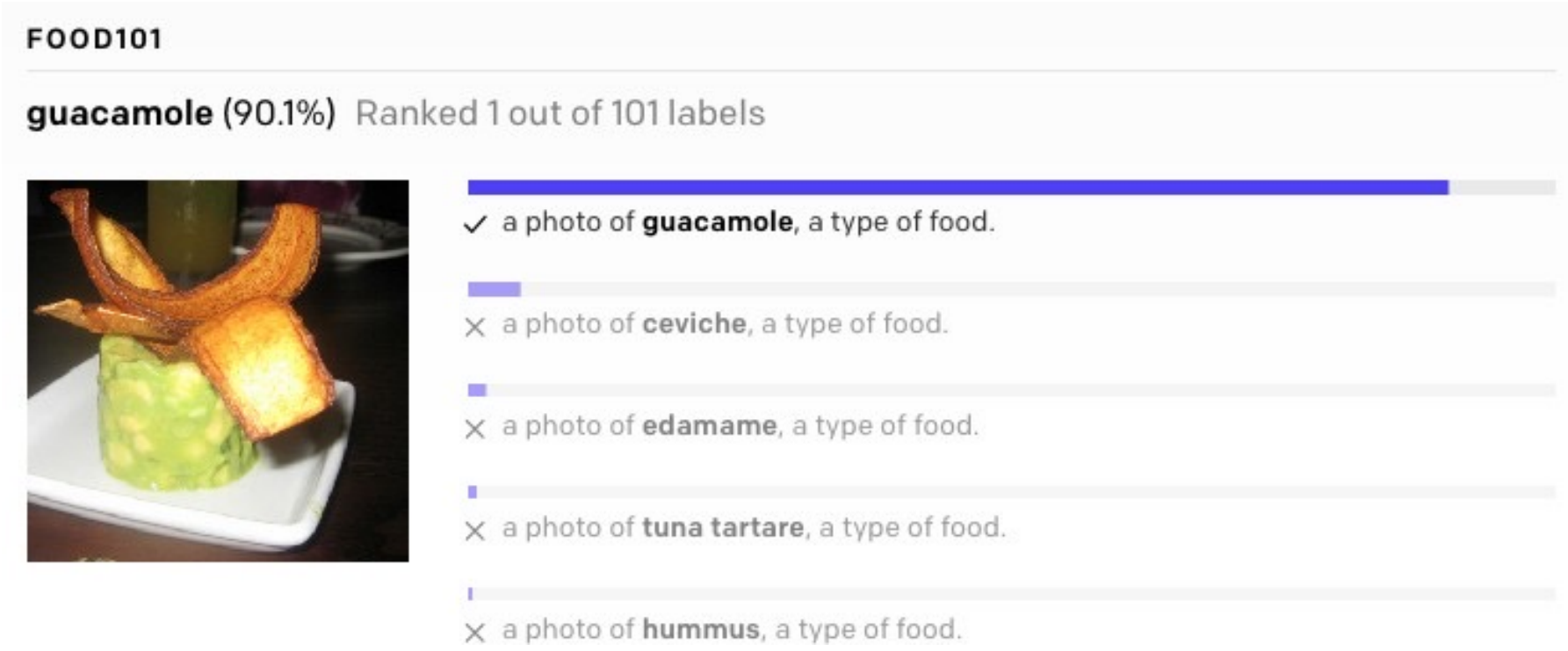


Figure: At test time the learned text encoder synthesizes a zero-shot linear classifier by embedding the names or descriptions of the target dataset's classes. This prediction setup is already very interesting because we don't need to finetune or train a top-layer classifier. As long as we include the correct label in our prediction option, this framework can perform the classification task.

How to Use CLIP for Classification?



How to Use CLIP for Classification?

YOUTUBE-BB

airplane, person (89.0%) Ranked 1 out of 23

✓ a photo of a **airplane**.

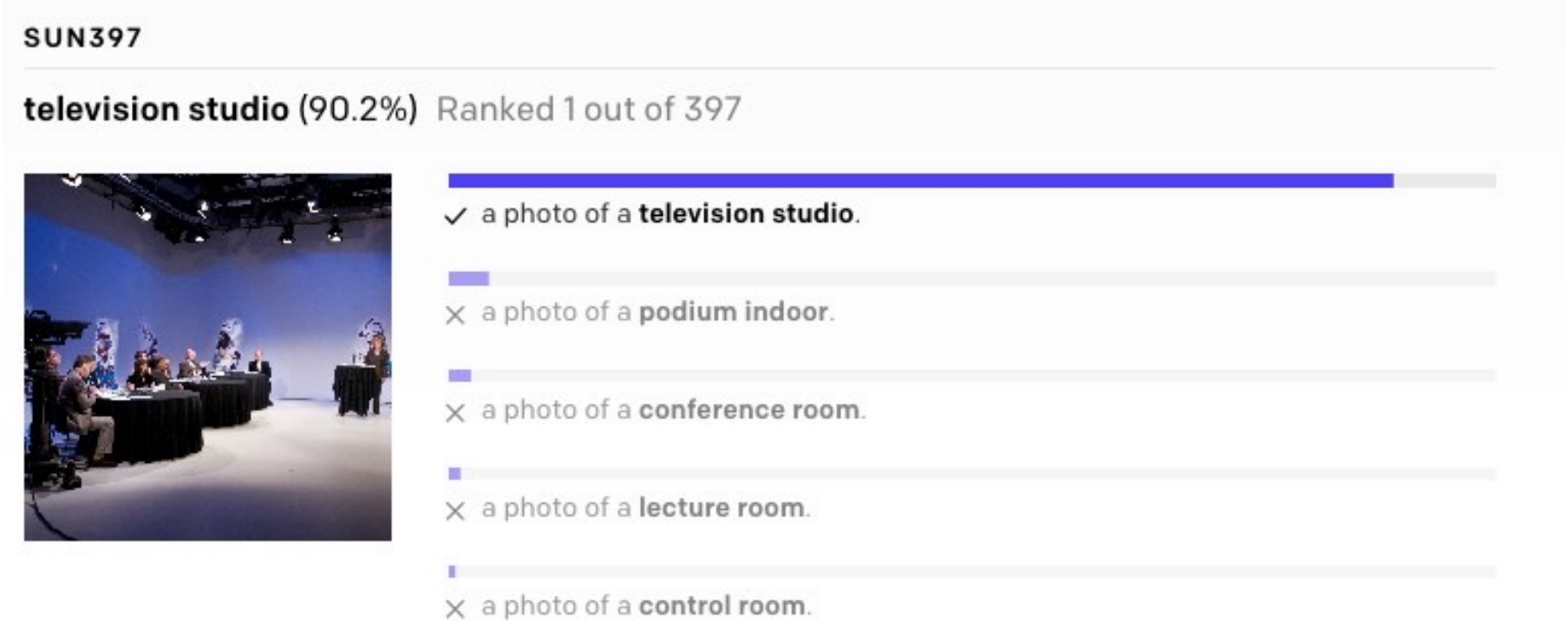
× a photo of a **bird**.

× a photo of a **bear**.

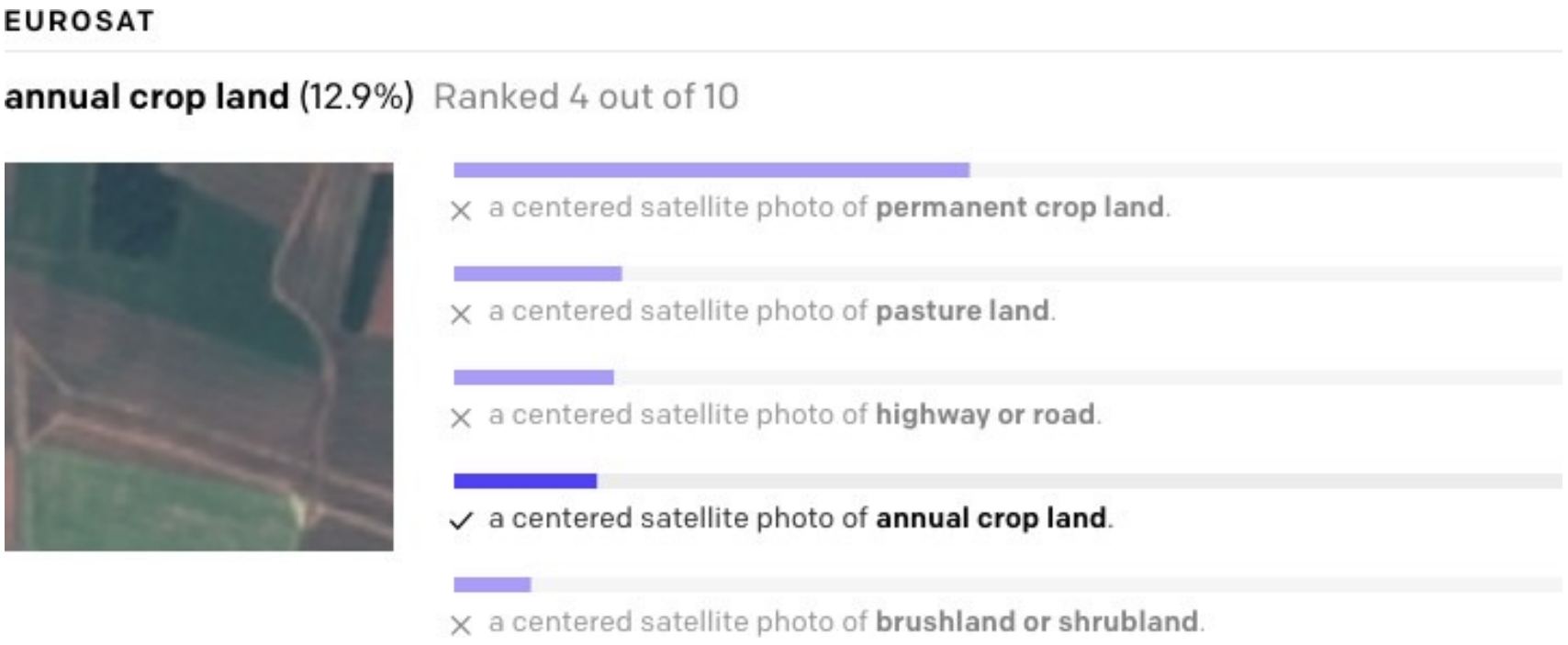
× a photo of a **giraffe**.

× a photo of a **car**.

How to Use CLIP for Classification?



How to Use CLIP for Classification?



Summary

- Metric Learning
- Losses
  - Pairwise contrastive loss
  - Triplet loss
- Applications
  - Object Re-ID for tracking
  - SimCLR
  - CLIP