

Photogrammetry & Robotics Lab

Machine Learning for Robotics and Computer Vision Tutorial

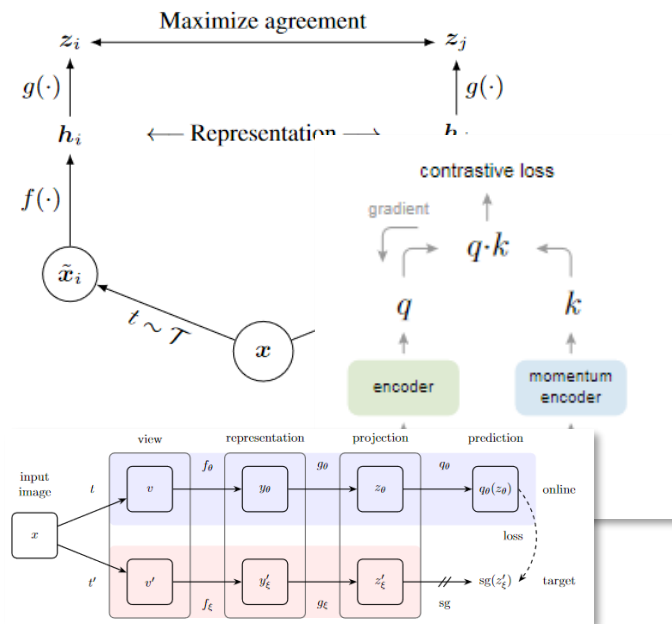
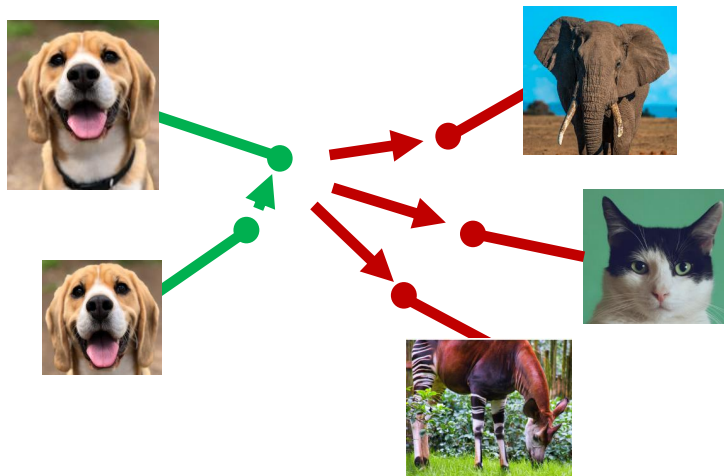
Pre-training & Self-supervision

Jens Behley

Exam Dates

- Oral Exam via Zoom in English
- Webcam must be on all the time and alone in room
- No other windows besides Zoom open.
- Date from the voting: **Wed, 25.08.2021**
- If this date still doesn't fit, contact us and we provide one alternative date

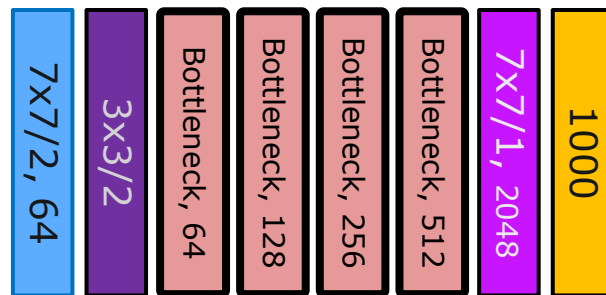
This week's lecture



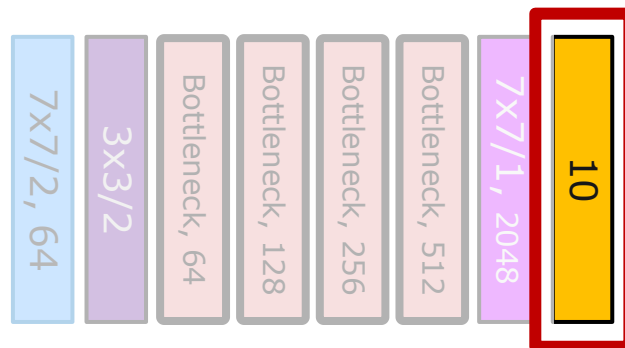
- Purely supervised training does not scale
- Using pre-trained models allows to get away with less labels!
- Self-supervised pretraining shows strong performance without any labels!

Pre-training & Fine-tuning

Stage 1:
Pre-training
(ImageNet)



Stage 2:
Fine-tuning
(Targeted dataset)



- **Idea:** Take weights from ImageNet and train only part of the network for novel task/dataset
- Training with pre-trained weights is faster and less data intensive!

Where to get pre-trained models?

- PyTorch vision contains pre-trained models for classification, segmentation, detection
- Repository of other often contain pre-trained versions of their approach

Pytorch-image-models Repo

The screenshot shows the GitHub repository page for `rwightman / pytorch-image-models`. At the top, there are buttons for `Sponsor`, `Notifications`, `Star` (11.4k), and `Fork` (1.7k). Below these are tabs for `Code`, `Issues` (27), `Pull requests` (11), `Discussions`, `Actions`, `Projects`, `Wiki`, and `Security`. The repository is currently on the `master` branch, with 28 branches and 25 tags. A `Go to file` button and a `Code` button are visible. The commit history shows a recent commit by `rwightman` titled "Remove unnecessary line from nest post ref..." with a green checkmark, commit hash `ee4d8fc`, and "2 days ago". It also shows "1,003 commits". Below the commit history is a table of files:

File	Description	Last Commit
<code>.github</code>	See if we can use tcmalloc in test runner	last month
<code>convert</code>	Move aggregation (convpool) for nest into NestLeve...	2 days ago
<code>docs</code>	Update README.md	29 days ago
<code>notebooks</code>	ImageNet-1k vs ImageNet-v2 comparison	2 years ago

On the right side, the `About` section describes the repository: "PyTorch image models, scripts, pretrained weights -- ResNet, ResNeXT, EfficientNet, EfficientNetV2, NFNet, Vision Transformer, MixNet, MobileNet-V3/V2, RegNet, DPN, CSPNet, and more". A link to `rwightman.github.io/pytorc...` is provided.

- Maintained by Ross Wightman
- Up-to-date implementation and pre-trained weights of state-of-the-art backbones
- 452(!) pretrained models/variants of common models

Feature Extraction

- The timm library provides handy methods to get just the features, see Docs:
https://rwightman.github.io/pytorch-image-models/feature_extraction/

`forward_features()`

```
import torch
import timm
m = timm.create_model('xception41', pretrained=True)
o = m(torch.randn(2, 3, 299, 299))
print(f'Original shape: {o.shape}')
o = m.forward_features(torch.randn(2, 3, 299, 299))
print(f'Unpooled shape: {o.shape}')
```

Output:

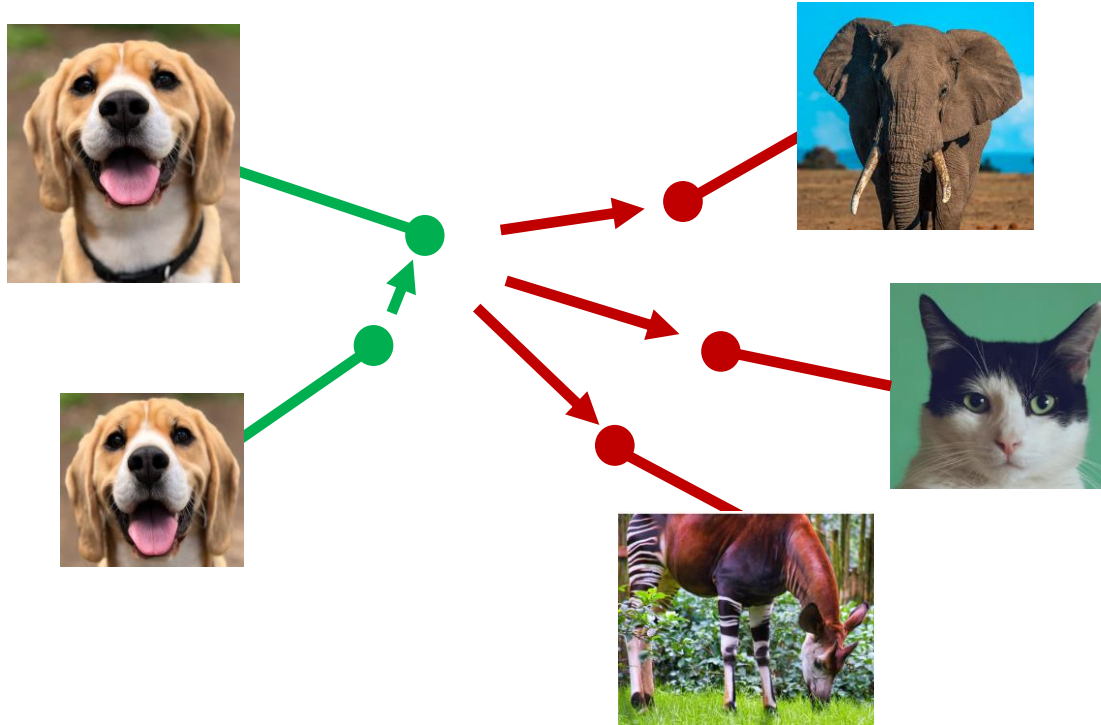
```
Original shape: torch.Size([2, 1000])
Unpooled shape: torch.Size([2, 2048, 10, 10])
```

Other Domains or Modalities



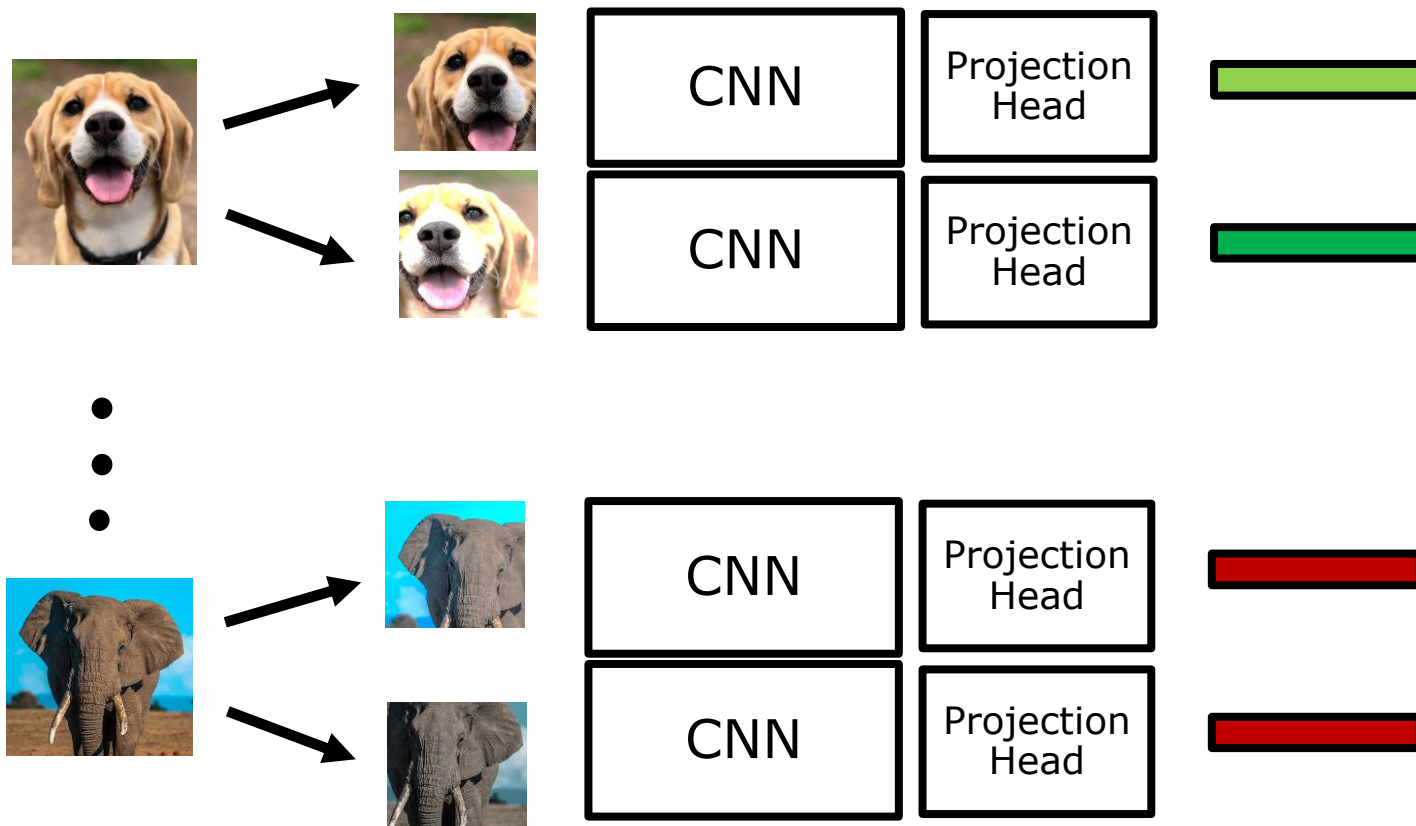
- ImageNet features or characteristics not always the best fit → **Self-supervised Learning**
- Specifically: **Contrastive Learning**

Contrastive Learning



- **Idea:** Learn representations such that similar examples (**positives**) are closer than representations of different examples (**negatives**)

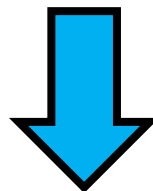
Common framework



Contrastive Loss

- Temperature scaled **contrastive loss**:

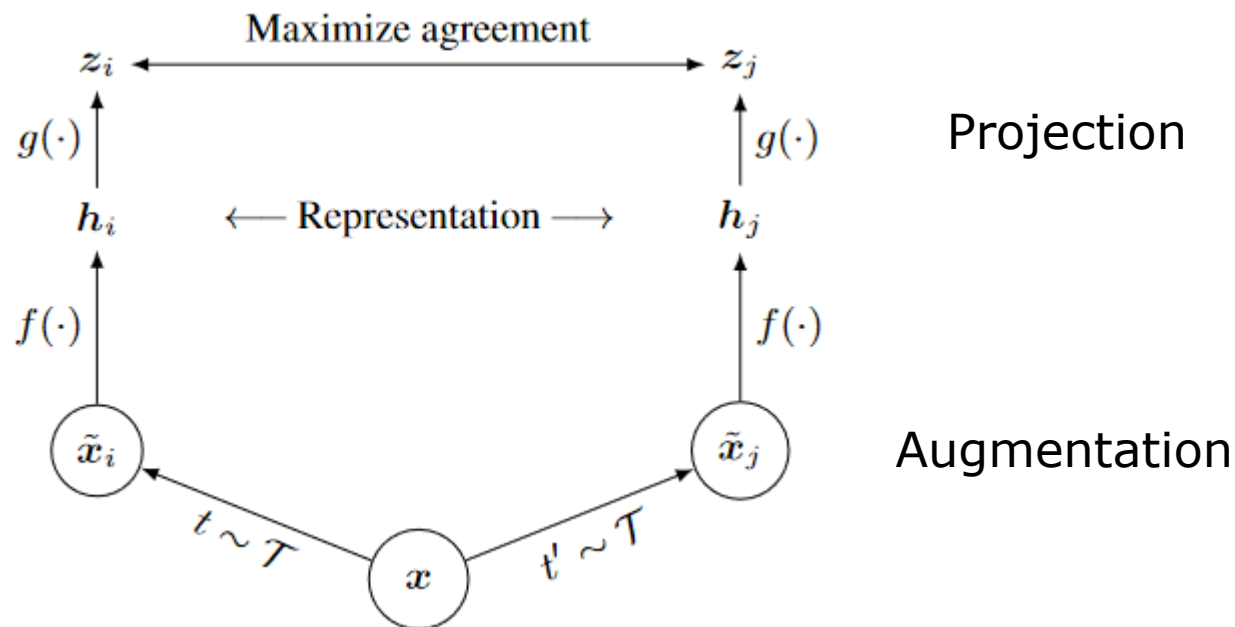
$$\ell_i = -\log \frac{\exp(\text{sim}(\mathbf{z}_i, \mathbf{z}_{i+})/\tau)}{\sum_{k \neq i} \exp(\text{sim}(\mathbf{z}_i, \mathbf{z}_k)/\tau)}$$



$$\ell_i = -\log \frac{\exp(\text{sim}(\text{green square}, \text{green square})/\tau)}{\sum_{k \neq i} \exp(\text{sim}(\text{green square}, \text{red square})/\tau)}$$

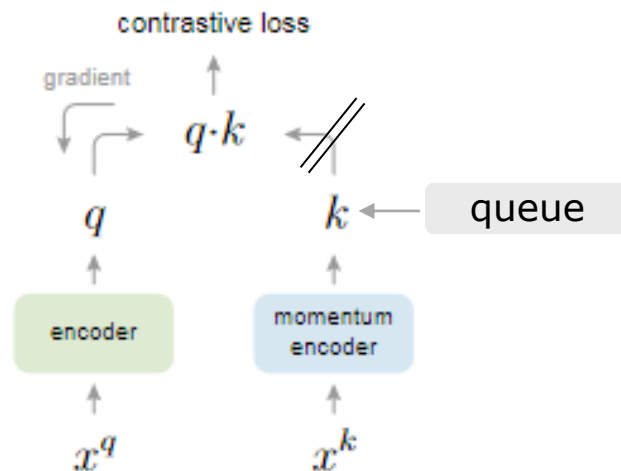
- Cosine Similarity: $\text{sim}(\mathbf{u}, \mathbf{v}) = \frac{\mathbf{u}^\top \mathbf{v}}{\|\mathbf{u}\| \|\mathbf{v}\|}$

SimCLR



- **Idea:** Learn representations by finding agreement between *projected* features
- Compute contrastive loss over projections/latents z
- Projection $g(\cdot)$ via FC \rightarrow ReLU \rightarrow FC

Momentum Encoder



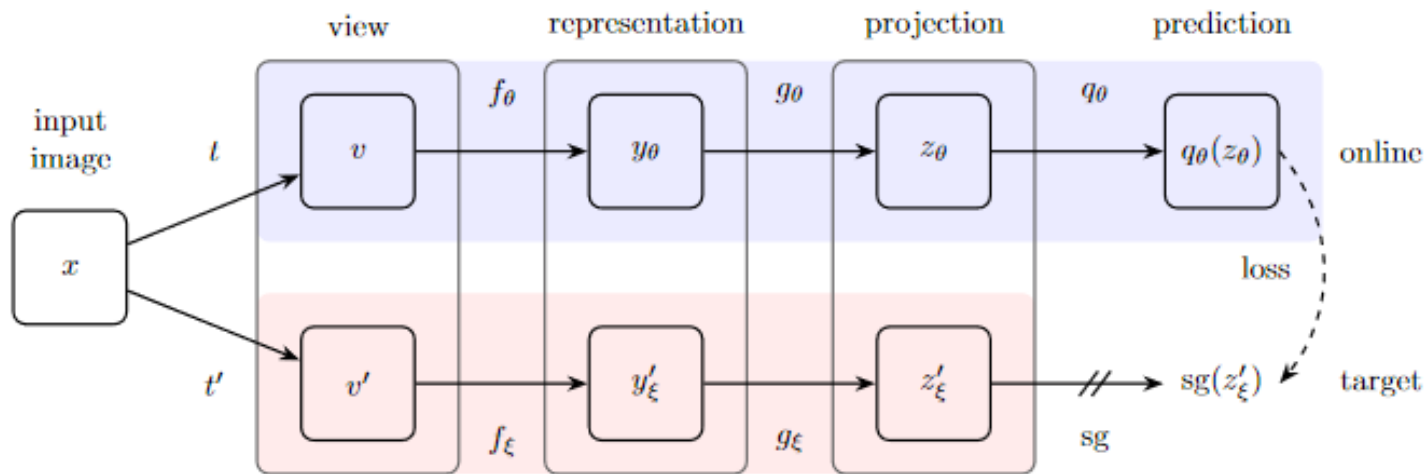
MoCo

- Only updated with weighted average between parameters of encoder θ_q and parameters of momentum encoder θ_k :

$$\theta_k \leftarrow m\theta_k + (1 - m)\theta_q$$

- Typically, large values (e.g., $m = 0.999$) better than smaller values (e.g., $m = 0.9$)

Bootstrap your own latent (BYOL)



- Augmented views are passed through online and target network
- Online network predicts output of the target network
- Important: There are no negative examples involved!

BYOL training and update

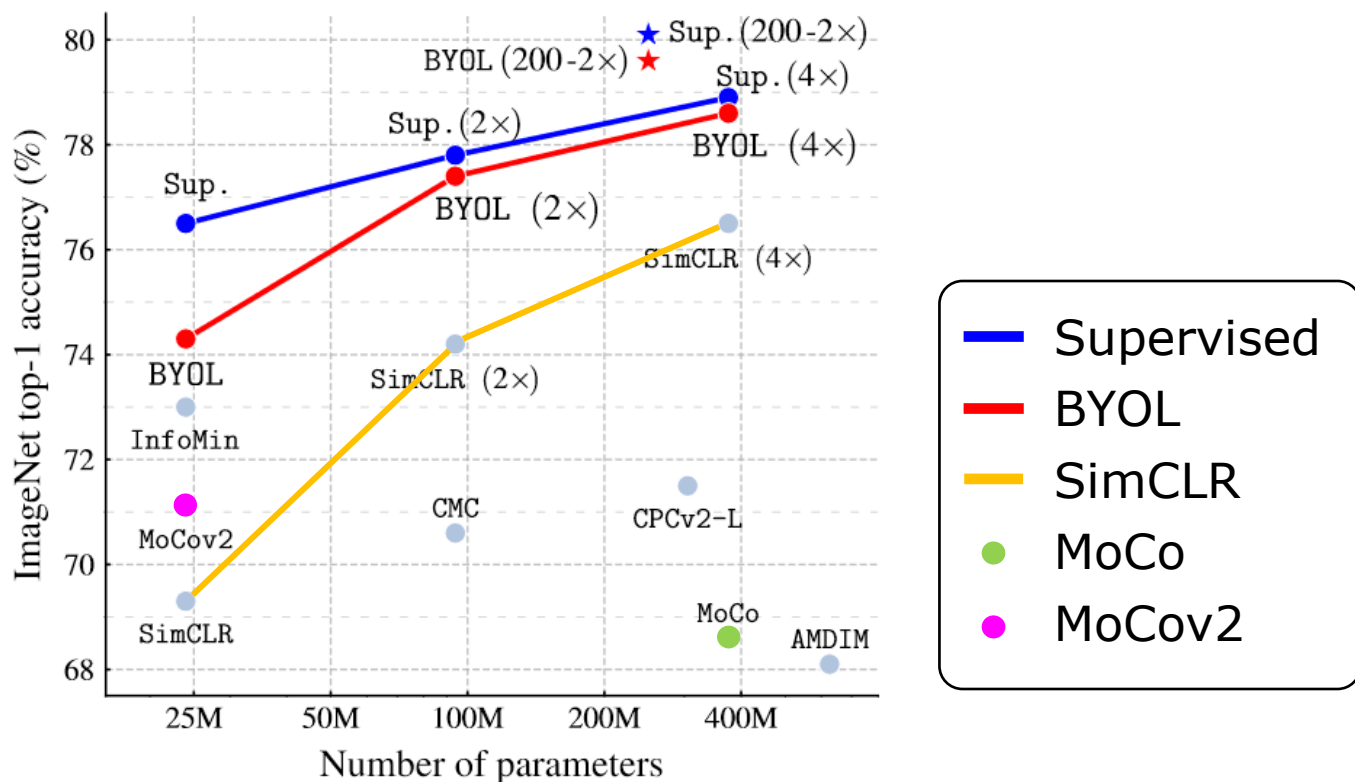
- Loss measures difference between prediction $q(z_\theta)$ and output of target network z'_ξ :

$$\ell = \left\| \frac{q(z_\theta)}{\|q(z_\theta)\|_2} - \frac{z'_\xi}{\|z'_\xi\|_2} \right\|_2^2 = 2 - 2 \cdot \frac{q(z_\theta)^\top z'_\xi}{\|q(z_\theta)\|_2 \|z'_\xi\|_2}$$

- Only online network is directly updated via backpropagation
- Target network parameters ξ are updated via momentum:

$$\xi \leftarrow m\xi + (1 - m)\theta$$

Comparison on ImageNet



- Results for ResNet50 with different widths (=number of channels), e.g., 2x, 4x
- BYOL approaches supervised training

See you next week!