Representation learning

Retrieval using learned representations

[Krizhevsky et al. NIPS12]

Verification problems in vision

Key question: do two photos show the same object/subject? (*verification)*

Face recognition datasets (e.g. *MSRA-CF*):

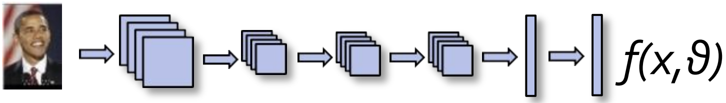
Re-identification datasets (e.g. *ViPER*):

Verification vs Classification

Key question: do two photos show the same object/subject? (*verification)*

- System must be able to handle unseen "classes"
- During training classes can be numerous,
    - small-sized, imbalanced, etc.

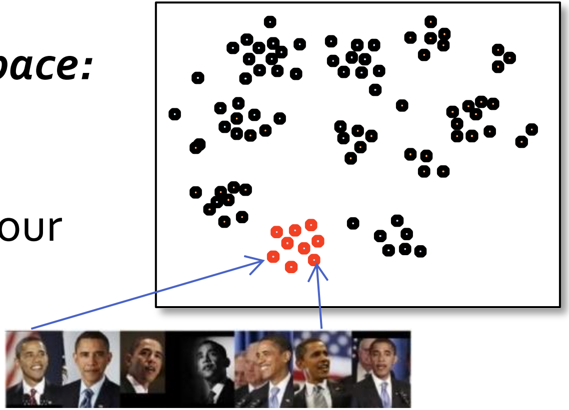
Verification as embedding learning



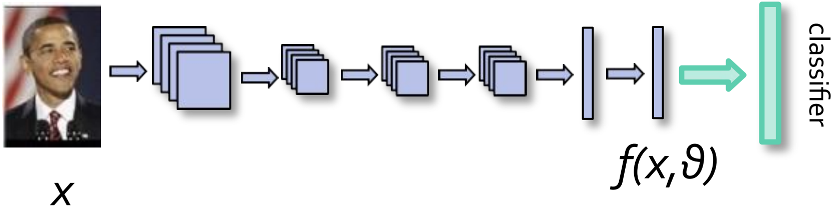
x

Semantic space:

NB: always normalize your descriptors!



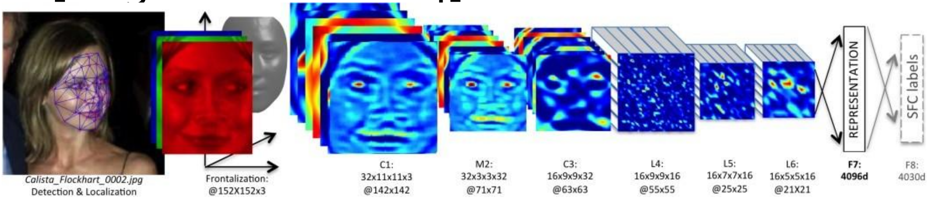
Approach 1: classification-based



- Same idea as "Train on ImageNet, use for retrieval"
- The bigger the classification dataset, the better is the performance
- Training-time classes can be seen as prototypes for test-time classes

Face verification: "Deep face"

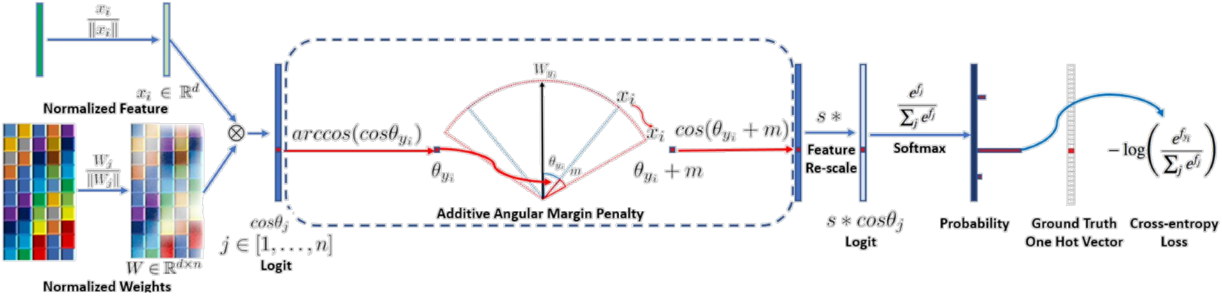
[Taigman et al. 2014]



- *Classification* network trained on 4030 people x ~1000 images.
- Target problem: *verification* (same vs different)

Adding normalization and margin (ArcFace)

Angular soft-max with margin loss:



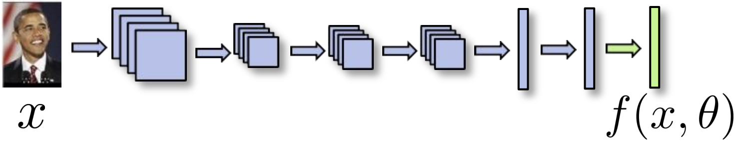
$$L_3 = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{e^{s(\cos(\theta_{y_i}+m))}}{e^{s(\cos(\theta_{y_i}+m))}+\sum_{j=1,j\neq y_i}^{n}e^{s\cos\theta_j}}$$

[Deng et al. CVPR 2019]

From classification to metric learning

- Classification requires class labels
- Metric learning requires only same-different labels
- Classification dataset can be trivially converted to metric learning dataset
- The opposite is not quite true (e.g. singletons)

Pair-based learning (*aka Siamese*)



$$L^+((x_1, x_2); \theta) = \rho\big(f(x_1, \theta), f(x_2, \theta)\big)$$
$$L^-((x_1, x_2); \theta) = \max\left(0, M - \rho\big(f(x_1, \theta), f(x_2, \theta)\big)\right)$$

Example distances:

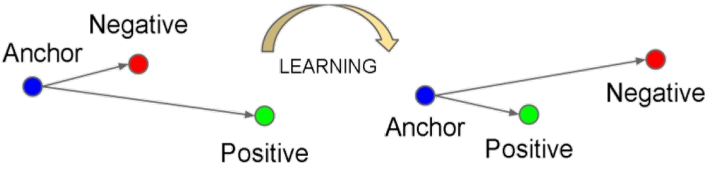
- 1 – cos
- L2 (equivalent if normalization is added)
- Separate network (verification network)

NB: all embedding-based systems work better with normalized descriptors

[Chopra et al. CVPR05]

Google "FaceNet"

[Schroff et al. CVPR15]



Simple **triplet loss**: $$\sum_i^N \left[\|f(x_i^a) - f(x_i^p)\|_2^2 - \|f(x_i^a) - f(x_i^n)\|_2^2 + \alpha \right]_+$$

- Use large mini-batches (1800, 40 images for several classes + lots of random)
- Take all positives from the batch
- Mine "*semi-hard*" negatives

$$\|f(x_i^a) - f(x_i^p)\|_2^2 < \|f(x_i^a) - f(x_i^n)\|_2^2$$

Applying ConvNets in practice: transfer learning

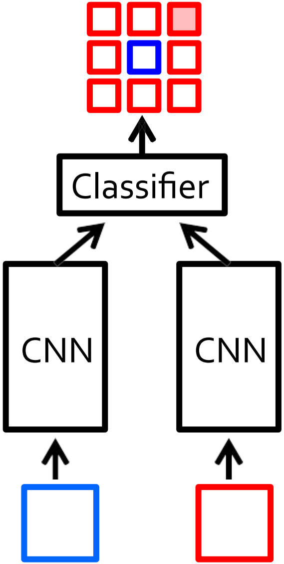
New problem

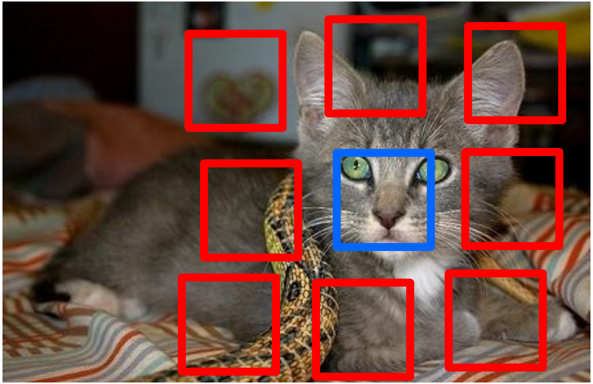
- PyTorch torchvision
- Other repositories

Self-supervised learning

- (Pre) trains representations without labels
- Especially useful for new domains with few labels (e.g. medical, robotics)
- **Meta-idea 1:** bottleneck learning (reduction to compression)
- **Meta-idea 2:** predictive learning (reduction to classification)
- **Meta-idea 3:** reduction to metric learning

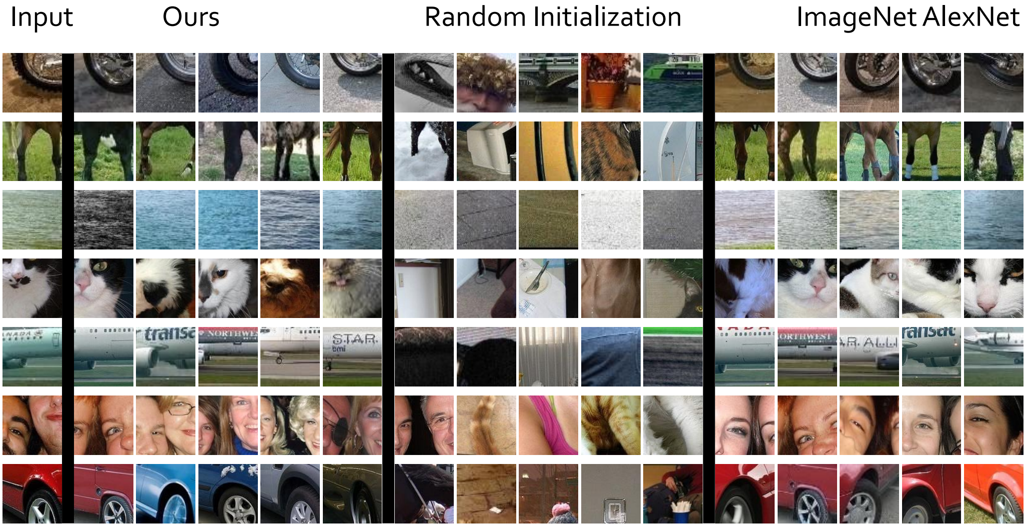
Predictive learning for still images





[Doersch et al. ICCV15]

Predictive learning for still images



[Doersch et al. ICCV15]

Self-supervised learning

- (Pre) trains representations without labels
- Especially useful for new domains with few labels (e.g. medical, robotics)
- Meta-idea 1: bottleneck learning (reduction to compression)
- Meta-idea 2: predictive learning (reduction to classification)
- **Meta-idea 3: reduction to metric learning**

SimCLR self-supervised learning

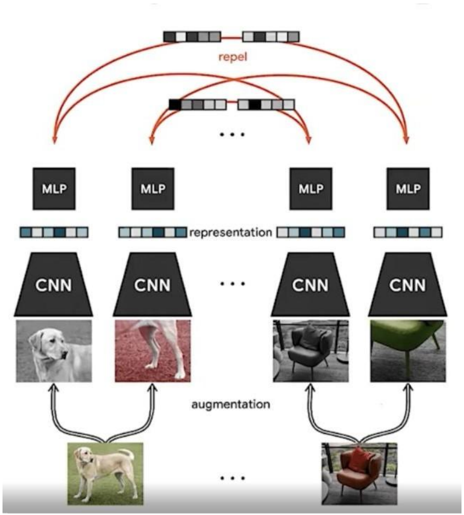
(a) Original (b) Crop and resize (c) Crop, resize (and flip) (d) Color distort. (drop) (e) Color distort. (jitter)

- Each batch consists of multiple pairs of matching images
- Each pair is the two copies of the same image with varying crop, color distortion, Gaussian blur
- We want the network to map each pair to two nearby points that are separated from remaining images
- Loss: $$\ell_{i,j}^{\text{NT-Xent}} = -\log \frac{\exp(\mathrm{sim}(\boldsymbol{z}_i, \boldsymbol{z}_j)/\tau)}{\sum_{k=1}^{2N} \mathbb{1}_{[k \neq i]} \exp(\mathrm{sim}(\boldsymbol{z}_i, \boldsymbol{z}_k)/\tau)}$$

- Requires large enough batch to work well

[Chen et al. ICML20] [Chen et al. NeurIPS20]

SimCLR self-supervised learning



https://ai.googleblog.com/2020/04/advancing-self-supervised-and-semi.html

[Chen et al. ICML20]
[Chen et al. NeurIPS20]

Types of supervised machine learning

- **"Standard" supervised learning:** multiple examples per class, starting from scratch
- **Transfer learning:** fine-tuning pretrained network
- ***Few-shot learning:*** creating network that can make class concepts from very few examples
- ***Zero-shot learning:*** creating network that can learn visual concepts from non-visual information (e.g. text or attributes)

Few-shot prompt

A simple prompt for extracting airport codes from text.

Prompt

Extract the airport codes from this text:

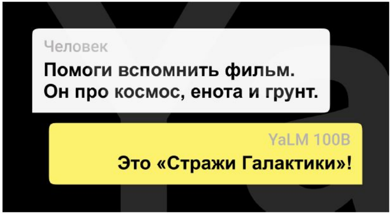
Text: "I want to fly from Los Angeles to Miami."
Airport codes: LAX, MIA

Text: "I want to fly from Orlando to Boston"
Airport codes:

Sample response

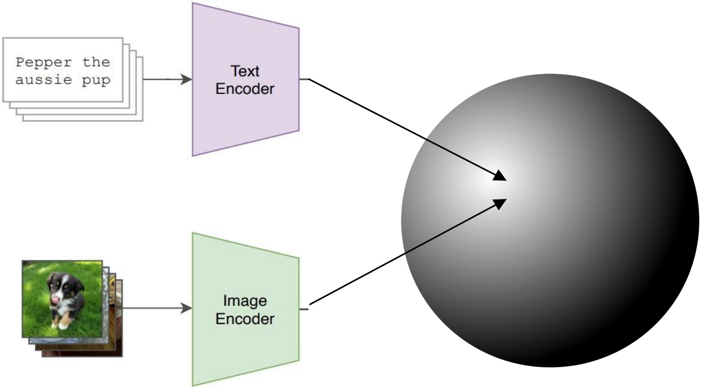
MCO, BOS

Zero-shot prompt



CLIP: connecting images and language

Can we do it without an adapter?



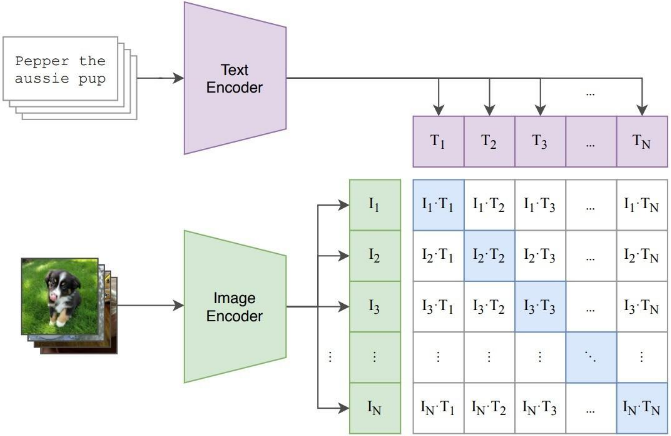
Joint space for images and text

....

....

[Radford et al. ICML21]

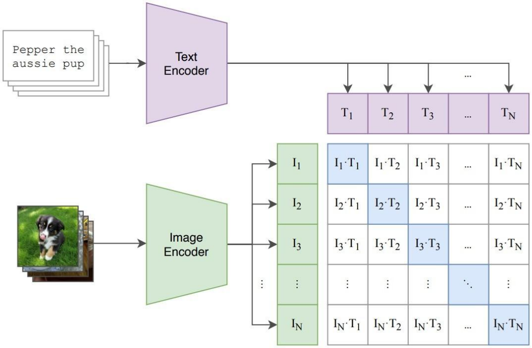
CLIP: connecting images and language




```
# symmetric loss function
labels = np.arange(n)
loss_i = cross_entropy_loss(logits, labels, axis=0)
loss_t = cross_entropy_loss(logits, labels, axis=1)
loss   = (loss_i + loss_t)/2
```

[Radford et al. ICML21]

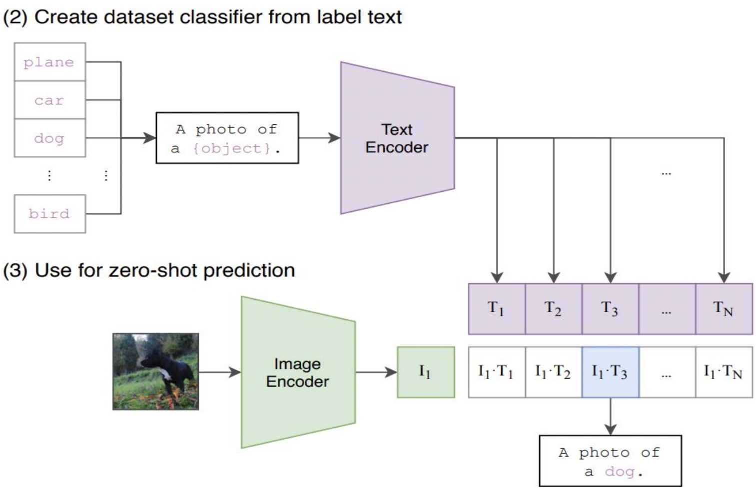
CLIP: connecting images and language



- Text encoder = Text Transformer
- Multiple architectures tried for Image encoder
- No pretraining
- Trained on 400 mln pairs from the Internet for 500,000 text queries
- MsCOCO - figures with labelled part

[Radford et al.ICML21]

CLIP: zero-shot evaluation



[Radford et al. ICML21]