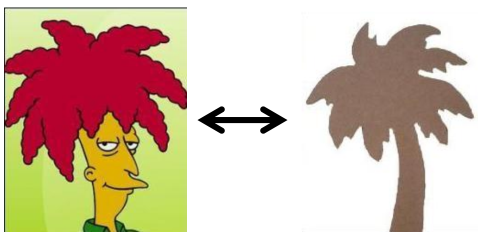
deep metric learning

Giorgos Tolias

Czech Technical University in Prague

pairwise similarity

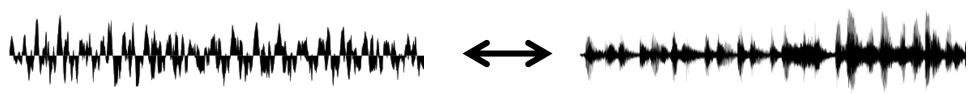
- human cognitive process involves ability to detect similarities between objects
- objects can be images, text documents, sound, etc...
- use deep learning to estimate pairwise similarity / distance



Snowboarding is a recreational activity and Winter Olympic and Paralympic sport that involves descending a snow-covered slope while standing on a snowboard attached to a rider's feet.

↔

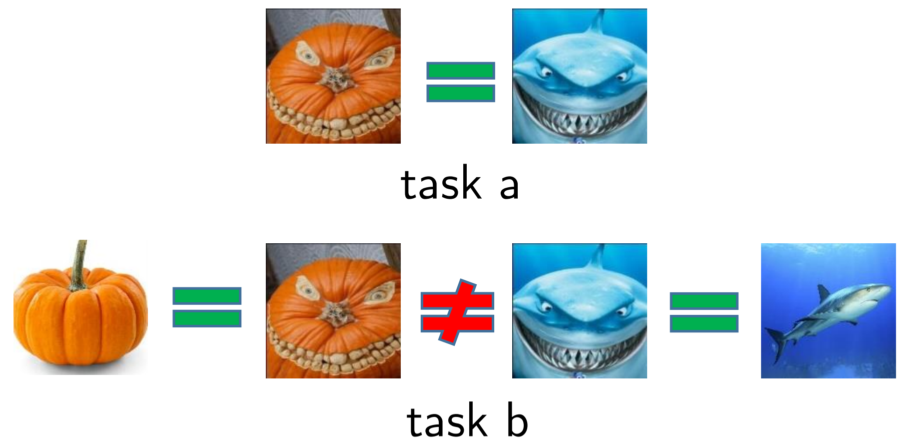
Skateboarding is an action sport that involves riding and performing tricks using a skateboard, as well as a recreational activity, an art form, an entertainment industry job, and a method of transportation.[1] Skateboarding has been shaped and influenced by many skateboarders throughout the years. A 2009 report found that the skateboarding market is worth an estimated $4.8 billion in annual revenue, with 11.08 million active skateboarders in the world.[2] In 2016, it was announced that skateboarding will be represented at the 2020 Summer Olympics in Tokyo.[3]



- applications
  - information retrieval
  - k-nearest-neighbor classification
  - clustering
  - data visualization

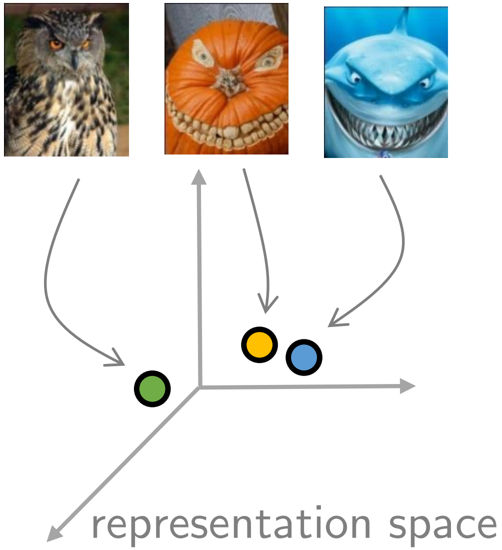
similarity / metric learning

- definition of good similarity measure (metric) is task dependent
- different semantic notion of similarity per task
  - not well captured by hand-crafted representations and standard metrics
- solution: learn it from the data

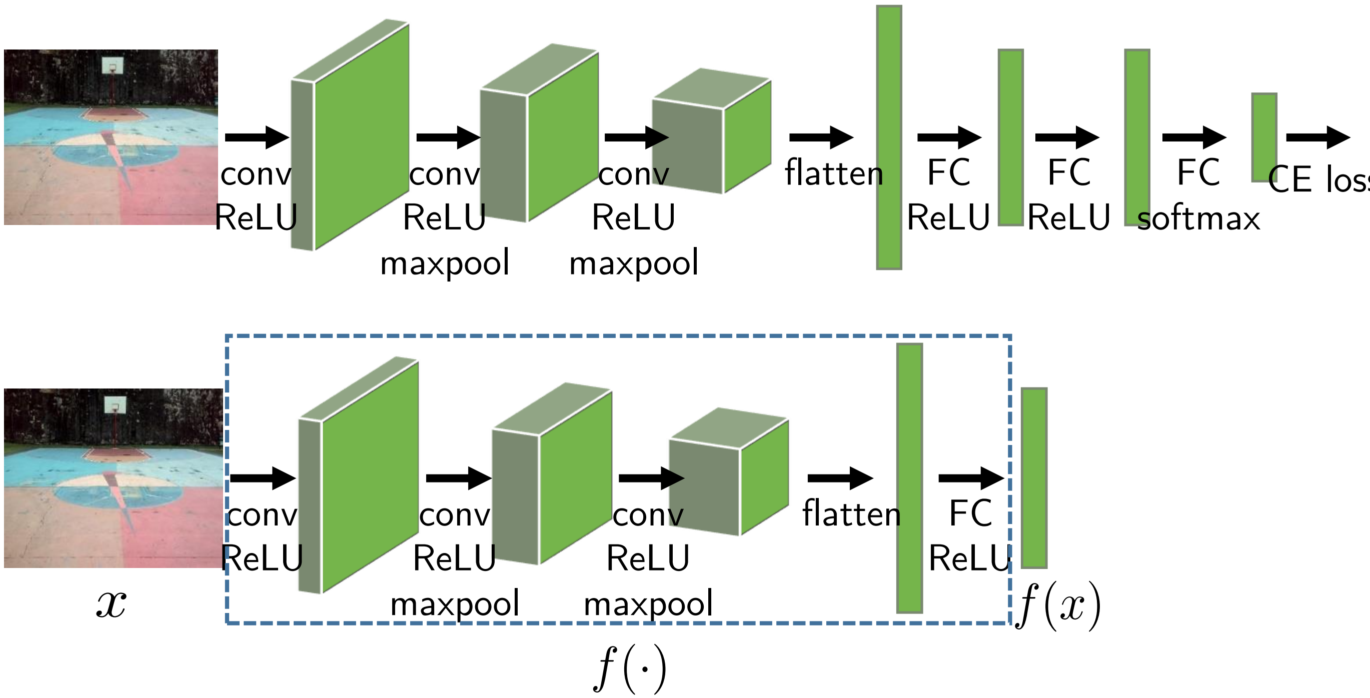


representation and similarity learning

- space of input examples $\mathcal{X}$
- learn the representation, use standard similarity measures / metrics
  - embed input examples to a representation (vector) space
  - embedding function $f : \mathcal{X} \to \mathbb{R}^D$
  - $\mathbf{x} = f(x), x \in \mathcal{X}$
  - learn the representation conditioned on a standard metric
- directly learn the similarity function / metric
  - similarity function $s : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$
  - involves learning the representation too



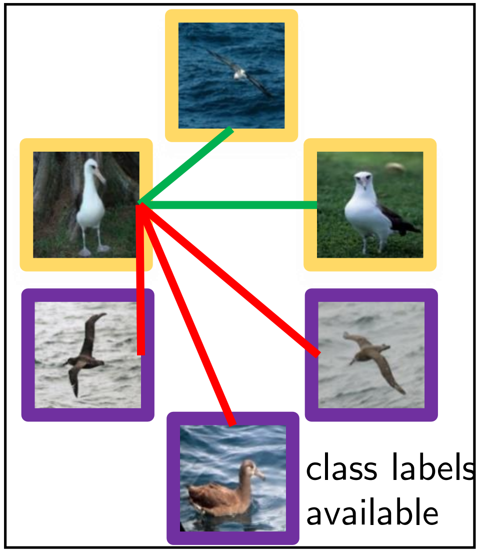
transfer learning

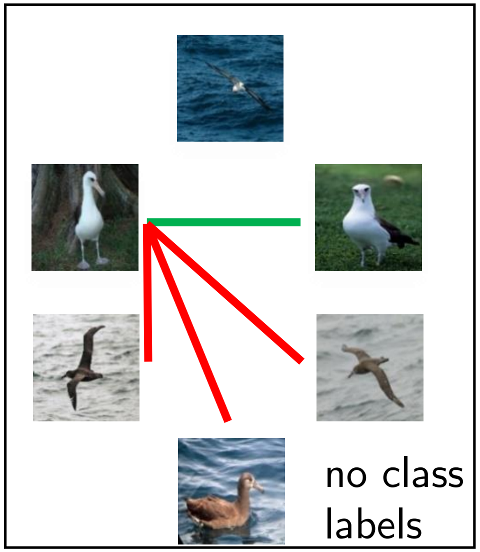


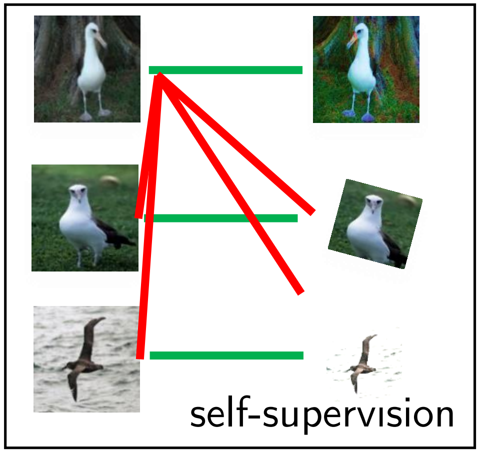
- pre-trained network is given, e.g. trained for classification with cross-entropy loss
- use internal activation vectors as representation
- use existing metrics to estimate pairwise similarity
    - Euclidean distance, cosine similarity, ...

training data - labels

- pairwise labels of training examples
  - relevant (positive, matching) pair
  - non-relevant (negative, non-matching) pair
- available image-level class labels
  - within (across) class pairs are positive (negative)
- manual annotation of pairs
  - typically very costly
- instance-discrimination
  - each image its own class
  - positives obtained by augmentations

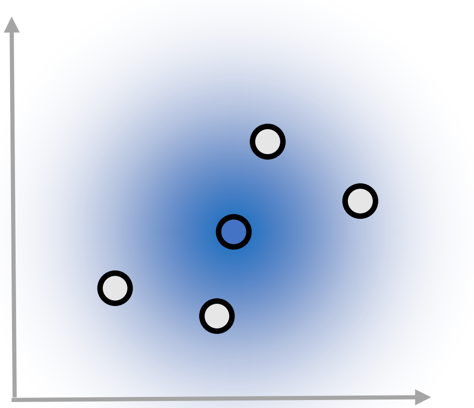




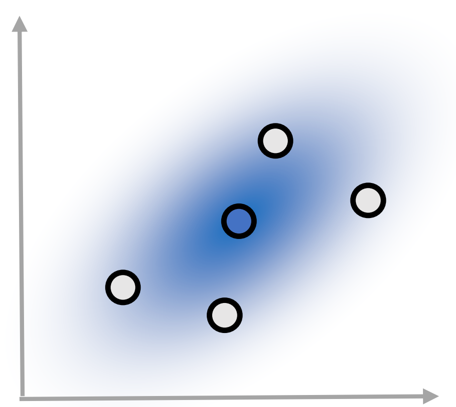


metric learning: Mahalanobis distance

- learn a parametric distance function from the data
  - input examples are vectors
- example: Mahalanobis distance
  - M is a $D \times D$ positive semi-definite matrix
  - $d_M(\mathbf{x}, \mathbf{z}) = \sqrt{(\mathbf{x} - \mathbf{z})^\top M (\mathbf{x} - \mathbf{z})}, \ \mathbf{x}, \mathbf{z} \in \mathbb{R}^D$
  - $d_M(\mathbf{x}, \mathbf{z}) = \sqrt{(\mathbf{x} - \mathbf{z})^\top L^\top L (\mathbf{x} - \mathbf{z})} = \sqrt{(L(\mathbf{x} - \mathbf{z}))^\top L (\mathbf{x} - \mathbf{z})}$
    $= \sqrt{(L\mathbf{x} - L\mathbf{z})^\top (L\mathbf{x} - L\mathbf{z})} = ||L\mathbf{x} - L\mathbf{z}||_2 = ||f(\mathbf{x}) - f(\mathbf{z})||_2$
  - mapping function $f(\mathbf{x}) = L\mathbf{x}$
  - can be modeled by a single fully-connected layer



Euclidean distance



Mahalanobis distance

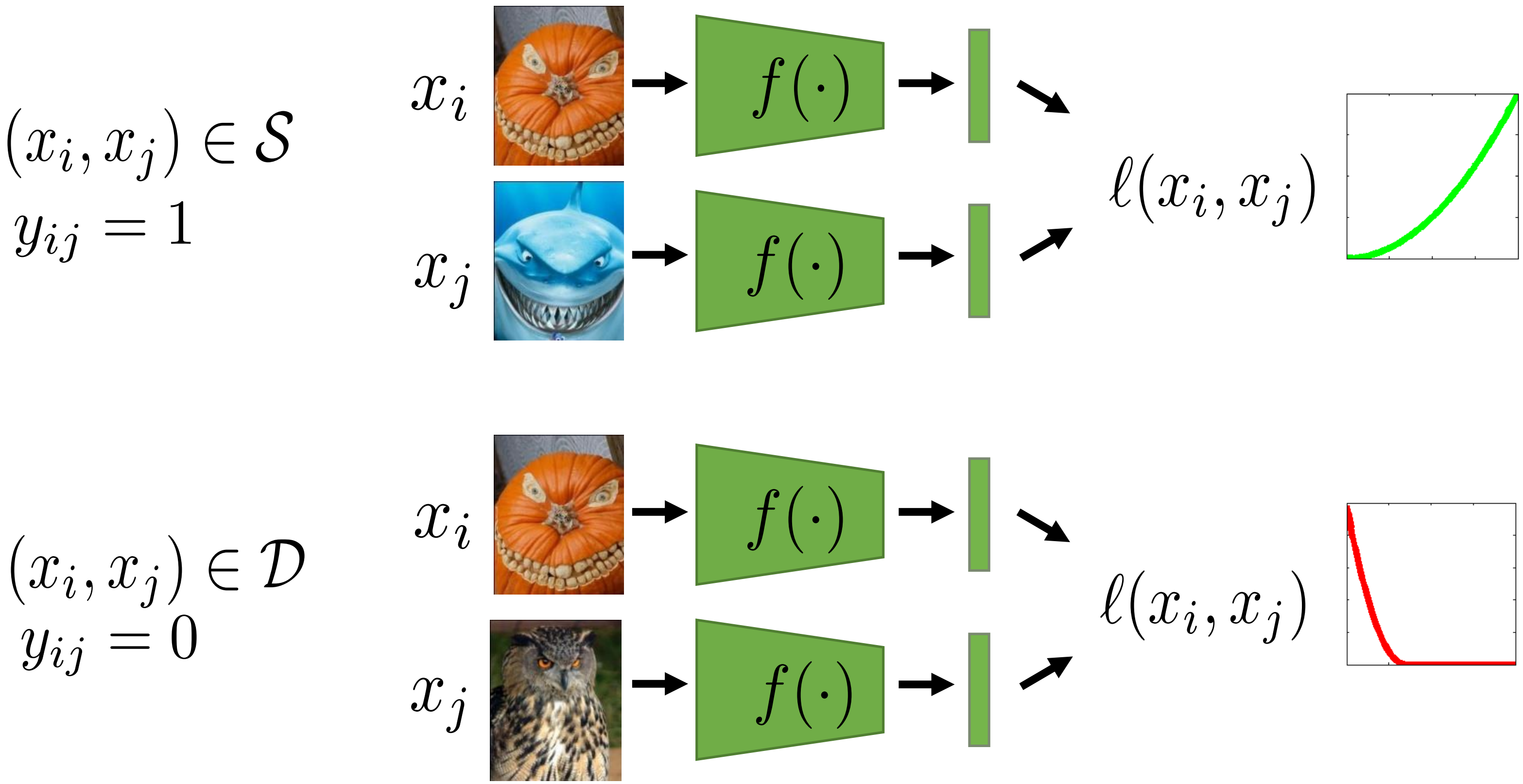
metric learning: Mahalanobis distance

- learn a parametric distance function from the data
  - input examples are vectors
- example: Mahalanobis distance
  - M is a $D \times D$ positive semi-definite matrix
  - $d_M(\mathbf{x}, \mathbf{z}) = \sqrt{(\mathbf{x} - \mathbf{z})^\top M (\mathbf{x} - \mathbf{z})}, \; \mathbf{x}, \mathbf{z} \in \mathbb{R}^D$
  - $d_M(\mathbf{x}, \mathbf{z}) = \sqrt{(\mathbf{x} - \mathbf{z})^\top L^\top L (\mathbf{x} - \mathbf{z})} = \sqrt{(L(\mathbf{x} - \mathbf{z}))^\top L (\mathbf{x} - \mathbf{z})}$
    $= \sqrt{(L\mathbf{x} - L\mathbf{z})^\top (L\mathbf{x} - L\mathbf{z})} = ||L\mathbf{x} - L\mathbf{z}||_2 = ||f(\mathbf{x}) - f(\mathbf{z})||_2$
  - mapping function $f(\mathbf{x}) = L\mathbf{x}$
  - can be modeled by a single fully-connected layer
- general cases:
  - $f : \mathbb{R}^D \to \mathbb{R}^{D'}$ is a feed-forward network
  - input examples are not vectors, $f : \mathcal{X} \to \mathbb{R}^{D'}$

contrastive loss

[Hadsell et al. 2006]

- two branch network; 2 networks that share weights



$$\ell(x_i, x_j) = \frac{1}{2} y_{ij} ||\mathbf{x}_i - \mathbf{x}_j||_2^2 + \frac{1}{2}(1 - y_{ij})[\tau - ||\mathbf{x}_i - \mathbf{x}_j||_2]_+^2$$

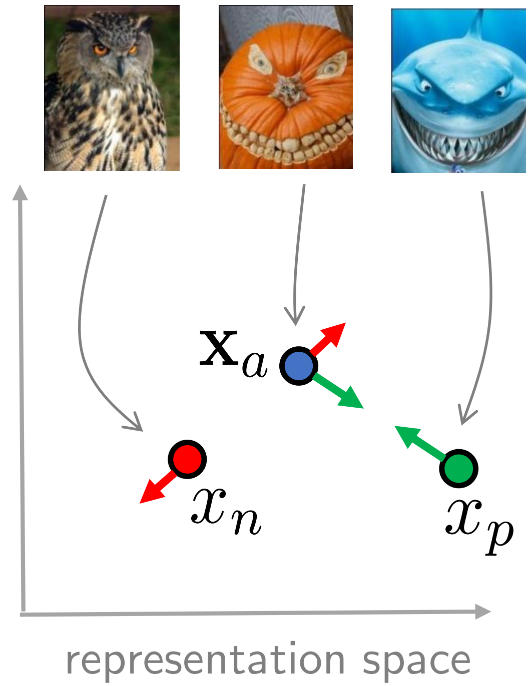
contrastive loss

- similar pair gradients

$$\frac{\partial \ell}{\partial \mathbf{x}_a} = \mathbf{x}_a - \mathbf{x}_p$$
$$\frac{\partial \ell}{\partial \mathbf{x}_p} = -\frac{\partial \ell}{\partial \mathbf{x}_a}$$

- dissimilar pair gradients

$$\frac{\partial \ell}{\partial \mathbf{x}_a} = \frac{\tau - ||\mathbf{x}_a - \mathbf{x}_n||}{||\mathbf{x}_a - \mathbf{x}_n||}(\mathbf{x}_n - \mathbf{x}_a)$$
$$\frac{\partial \ell}{\partial \mathbf{x}_n} = -\frac{\partial \ell}{\partial \mathbf{x}_a}$$



$$\ell(x_i, x_j) = \frac{1}{2} y_{ij} ||\mathbf{x}_i - \mathbf{x}_j||_2^2 + \frac{1}{2}(1 - y_{ij})[\tau - ||\mathbf{x}_i - \mathbf{x}_j||_2]_+^2$$

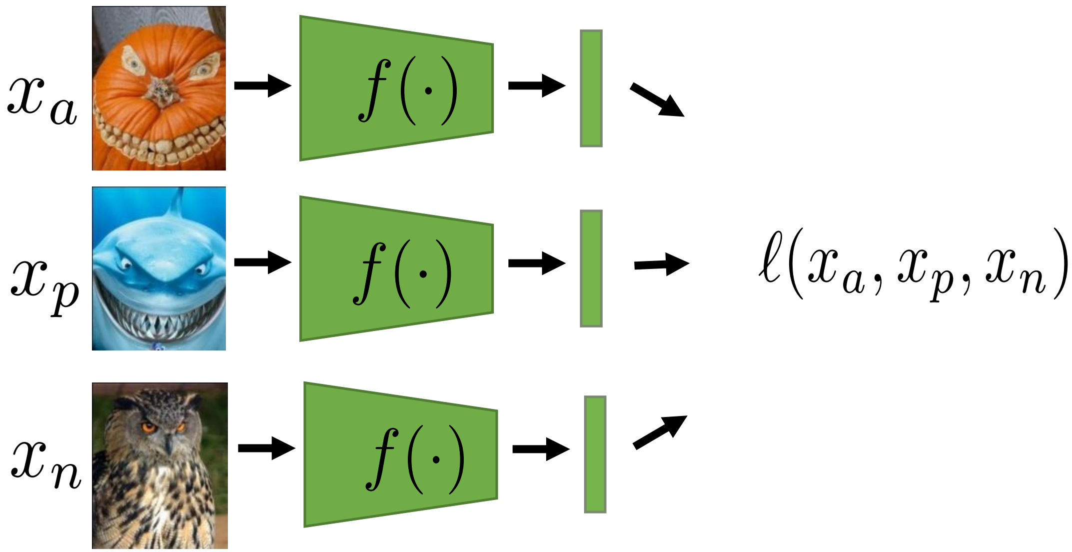
triplet loss

[Schroff et al. 2015]

- three branch network; 3 networks that share weights

$(x_a, x_p) \in \mathcal{S}$

$(x_a, x_n) \in \mathcal{D}$



$$\ell(x_a, x_p, x_n) = [||\mathbf{x}_a - \mathbf{x}_p||_2^2 - ||\mathbf{x}_a - \mathbf{x}_n||_2^2 + \alpha]_+$$

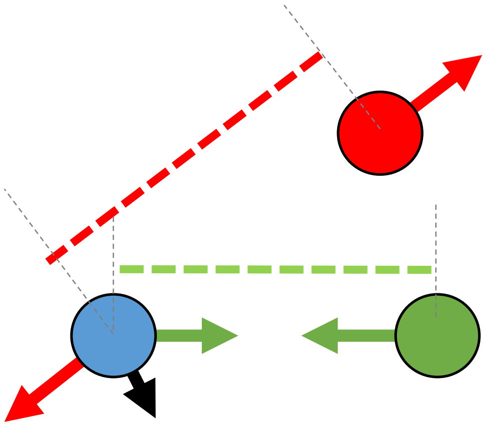
triplet loss

- gradients

$$\frac{\partial \ell}{\partial \mathbf{x}_p} = 2(\mathbf{x}_p - \mathbf{x}_a)$$

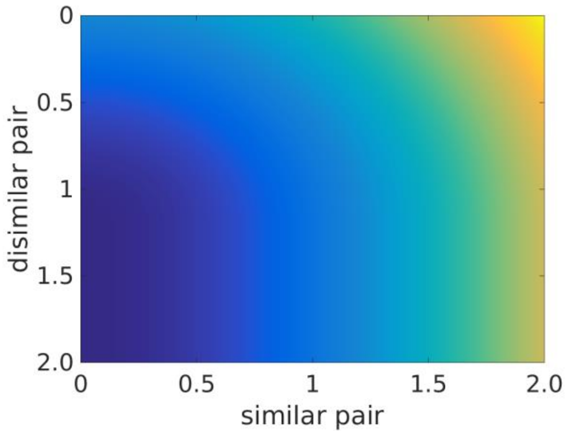
$$\frac{\partial \ell}{\partial \mathbf{x}_n} = 2(\mathbf{x}_a - \mathbf{x}_n)$$

$$\frac{\partial \ell}{\partial \mathbf{x}_a} = 2(\mathbf{x}_a - \mathbf{x}_p) - 2(\mathbf{x}_a - \mathbf{x}_n) = 2(\mathbf{x}_n - \mathbf{x}_p)$$

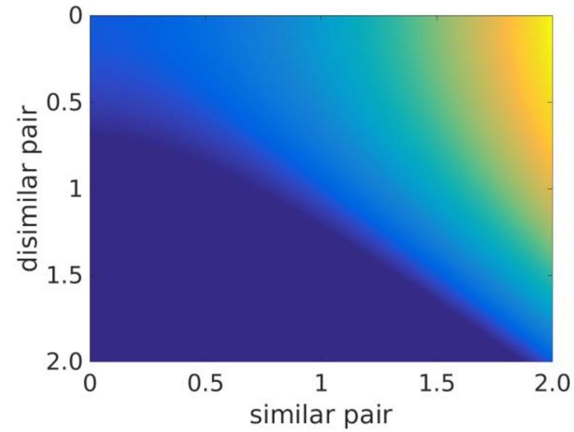


$$\ell(x_a, x_p, x_n) = [||\mathbf{x}_a - \mathbf{x}_p||_2^2 - ||\mathbf{x}_a - \mathbf{x}_n||_2^2 + \alpha]_+$$

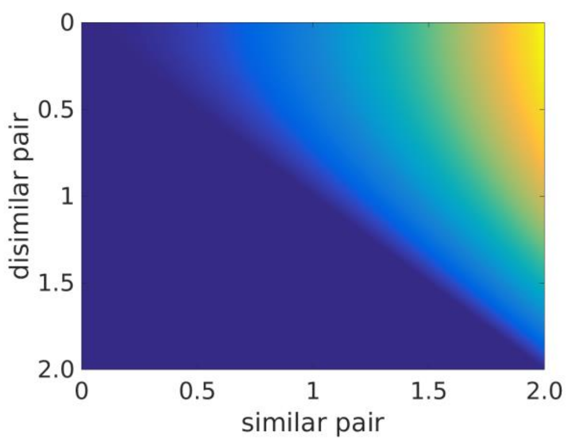
pairwise losses



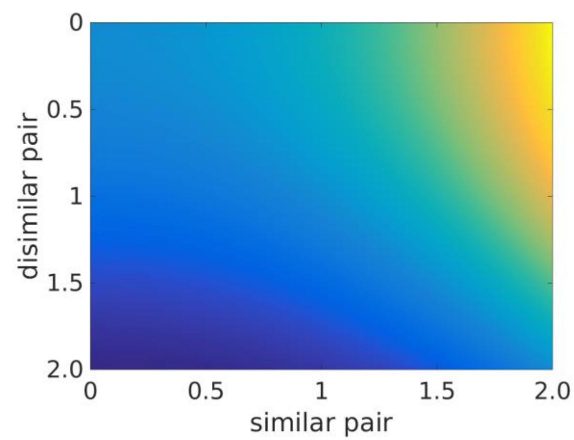
contrastive



triplet



$$[\mathbf{x}_a^\top \mathbf{x}_n - \mathbf{x}_a^\top \mathbf{x}_p]_+$$



$$\log(1 + e^{\mathbf{x}_a^\top \mathbf{x}_n - \mathbf{x}_a^\top \mathbf{x}_p})$$ [Sohn 2016]

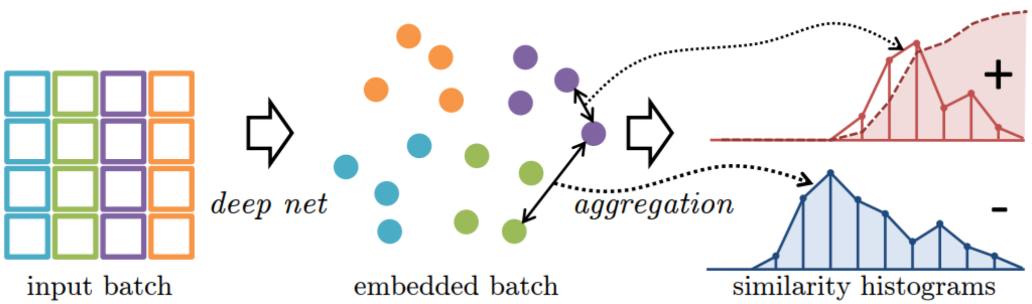
mini-batches & hard negatives

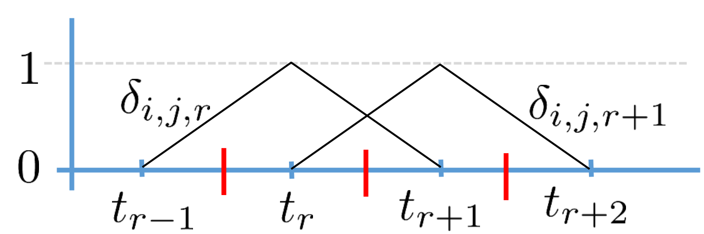
- mini-batch construction [Roth et al., 2020]
  - randomly sample n classes and b/n examples per class
  - greedy approach to maximize covered space
    - next example maximizes the distances to already included examples
  - match training dataset statistics (distribution of pairwise distances)
    - set of random mini-batches: pick to minimize distribution distance

- sampling of negatives matters
  - random sampling: zero loss for most pairs/triplets
  - hard negatives: negative pair, but nearby in the representation space
- online sampling
  - within batch single hardest, semi-hard mining [Schroff et al. 2015], distance-weighted sampling [Wu et al. 2017]
- offline sampling
  - nearest-neighbor search: guaranteed hard negatives in the batch
  - hardness changes: repeat the process during training

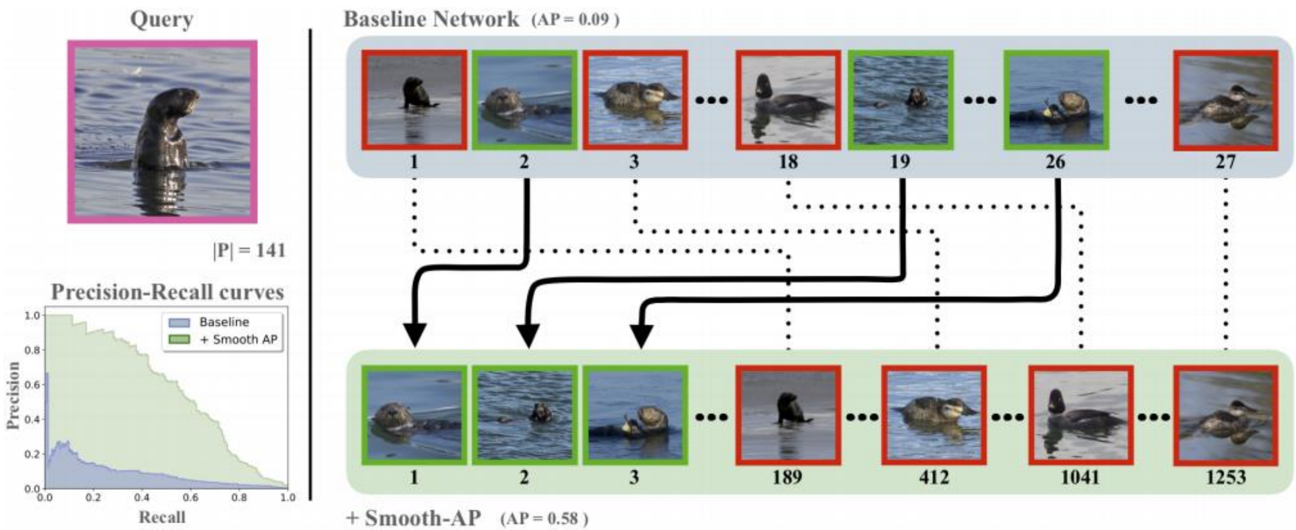
histogram loss [Ustinova & Lempitsky, 2016]

- minimize probability that similarity of a random negative pair is higher than the similarity of a random positive pair
- $\mathbb{E}_{u \sim p^-}[\Phi^+(u)] = \int_{-1}^{1} p^-(u)\Phi^+(u)\,\mathrm{d}u = \int_{-1}^{1} p^-(u)\left[\int_{-1}^{u} p^+(v)\,\mathrm{d}v\right]\mathrm{d}u$
- approximated by $\sum_{r=1}^{R}\left(h_r^- \sum_{q=1}^{r} h_q^+\right) = \sum_{r=1}^{R}(h_r^- \phi_q^+)$
- histogram for positive pairs: $h_r^+ = \frac{1}{|\mathcal{P}^+|} \sum_{(i,j):(x_i,x_j)\in\mathcal{P}^+} \delta_{i,j,r}$
- equivalently for the negative pairs





smooth AP loss



- Average-Precision (AP) is a common retrieval metric

$$\mathrm{AP}_q = \frac{1}{|\mathcal{S}_P|} \sum_{i \in \mathcal{S}_P} \text{precision@ranking}_i$$

$$= \frac{1}{|\mathcal{S}_P|} \sum_{i \in \mathcal{S}_P} \frac{\#\text{positives-up-to-ranking}_i}{\text{ranking}_i}$$

$$= \frac{1}{|\mathcal{S}_P|} \sum_{i \in \mathcal{S}_P} \frac{\mathcal{R}(i, \mathcal{S}_P)}{\mathcal{R}(i, \mathcal{S}_\Omega)}$$

- AP is not differentiable
- optimize a smooth approximation instead [Brown et al. 2020]

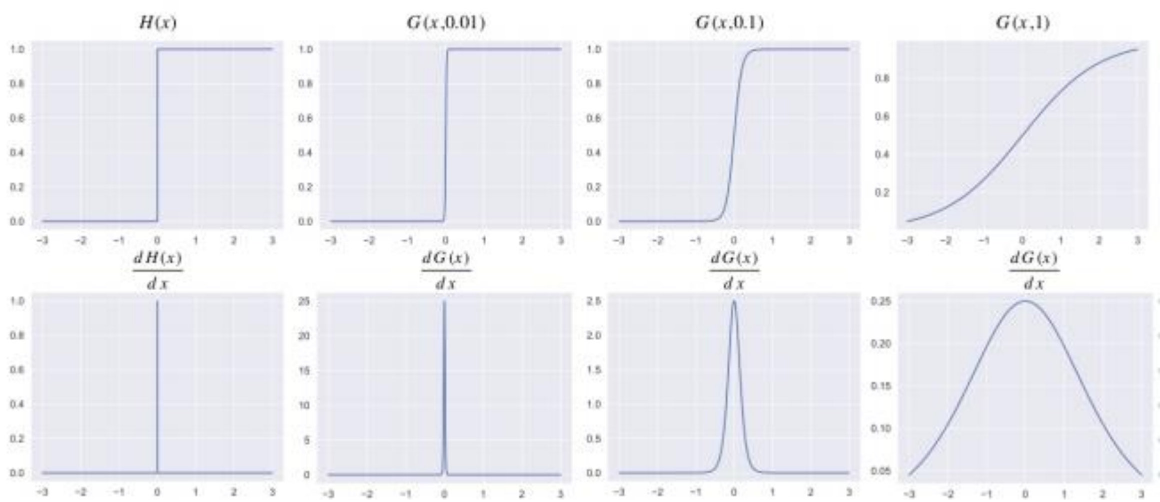
smooth AP loss

- rewrite AP as

$$AP_q = \frac{1}{|\mathcal{S}_P|} \sum_{i \in \mathcal{S}_P} \frac{1 + \sum_{j \in \mathcal{S}_P, j \neq i} \mathbb{1}\{D_{ij} > 0\}}{1 + \sum_{j \in \mathcal{S}_P, j \neq i} \mathbb{1}\{D_{ij} > 0\} + \sum_{j \in \mathcal{S}_N, j \neq i} \mathbb{1}\{D_{ij} > 0\}}$$

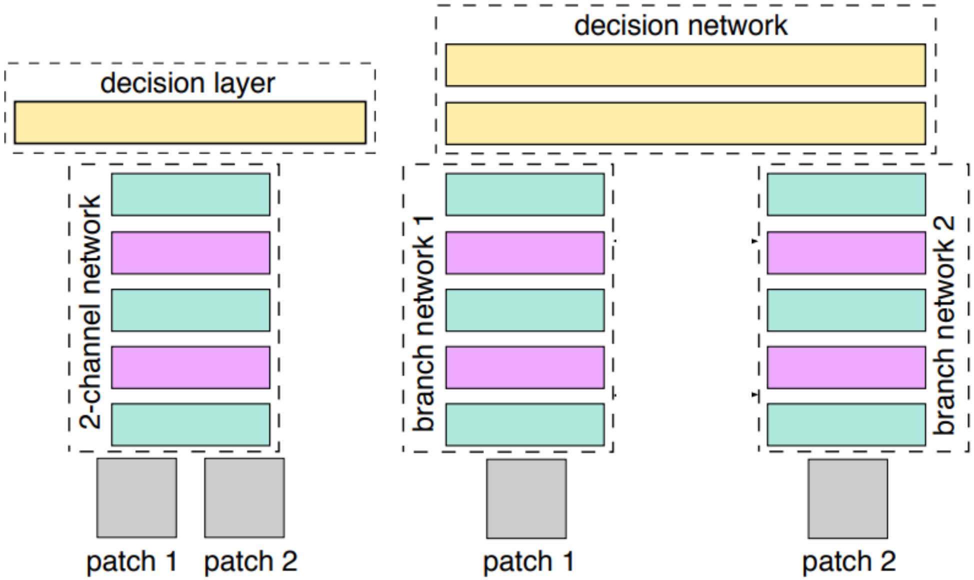
- replace the indicator function with sigmoid

$$AP_q = \frac{1}{|\mathcal{S}_P|} \sum_{i \in \mathcal{S}_P} \frac{1 + \sum_{j \in \mathcal{S}_P, j \neq i} G(D_{ij})}{1 + \sum_{j \in \mathcal{S}_P, j \neq i} G(D_{ij}) + \sum_{j \in \mathcal{S}_N, j \neq i} G(D_{ij})}$$



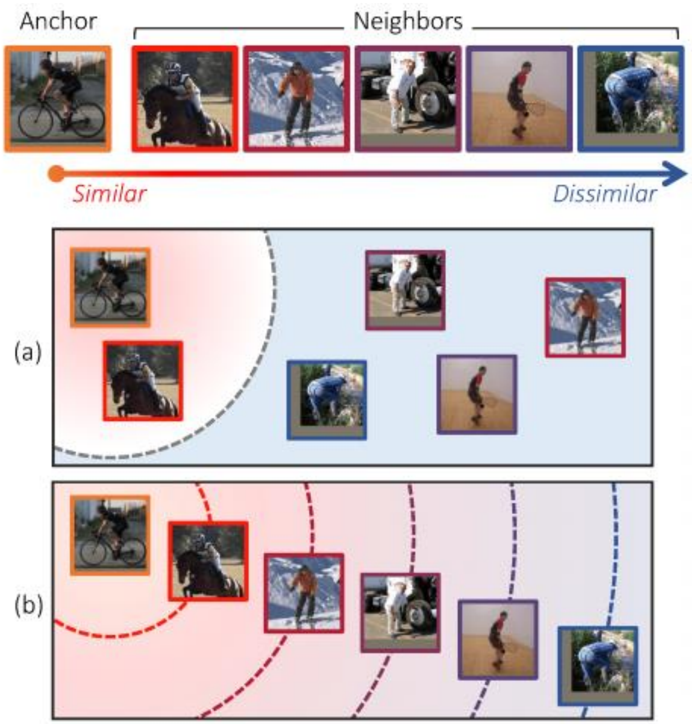
similarity function learning

- learn similarity function $s : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$
- input is an example pair
- higher cost for inference



[Zagoruyko & Komodakis, 2015]

beyond binary supervision



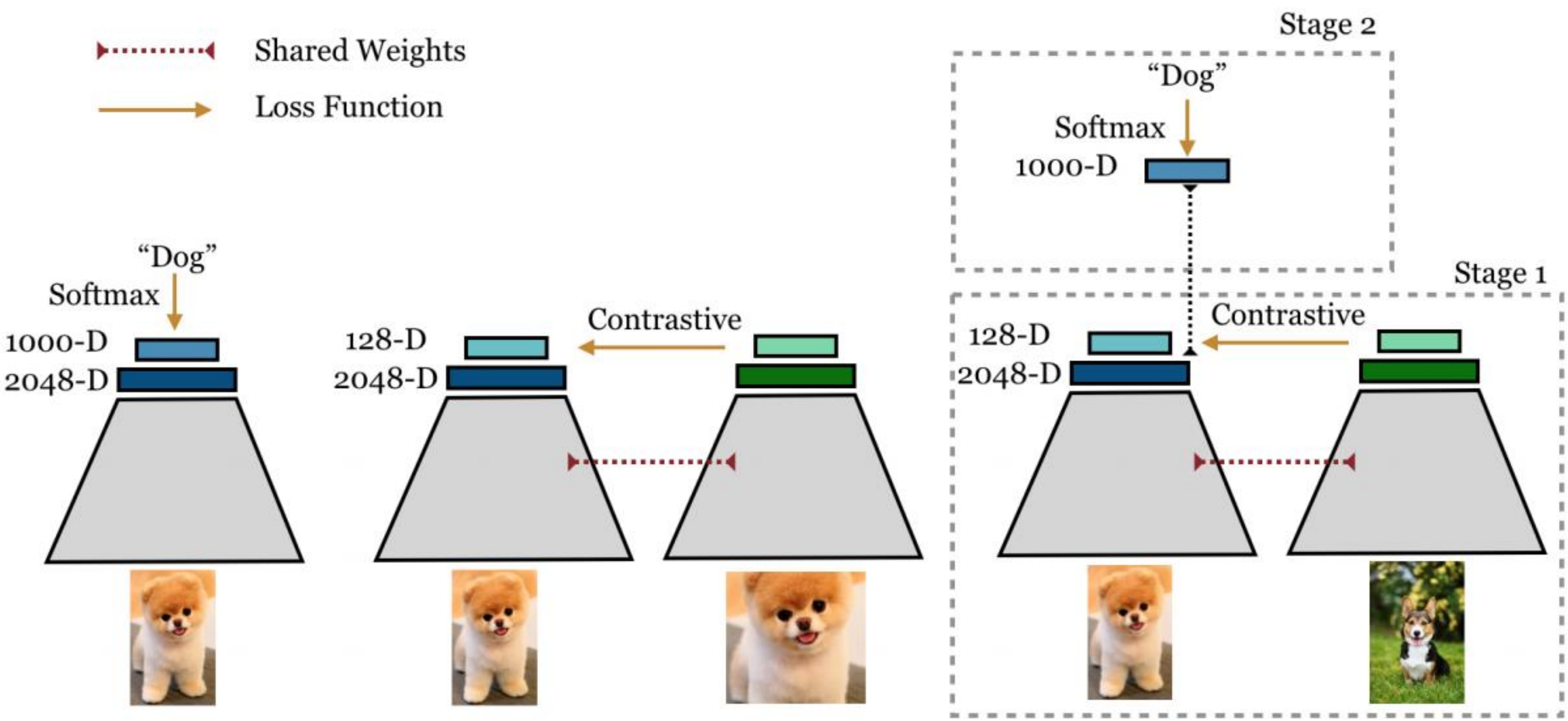
$$\ell(x_a, x_i, x_j, y_a, y_i, y_j) = \left(\log \frac{||\mathbf{x}_a - \mathbf{x}_i||_2}{||\mathbf{x}_a - \mathbf{x}_j||_2} - \log \frac{D(y_a, y_i)}{D(y_a, y_j)}\right)^2$$

distance ratio
representation space

distance ratio
label space

[Kim et al.'19]

self-supervised representation learning

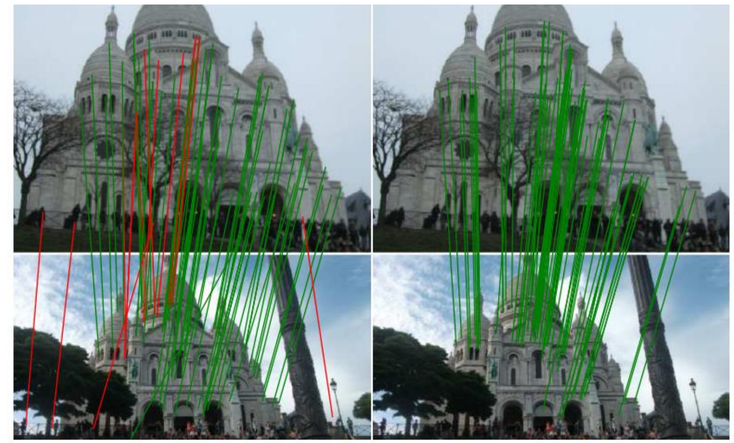


(a) Supervised Cross Entropy (b) Self Supervised Contrastive (c) Supervised Contrastive

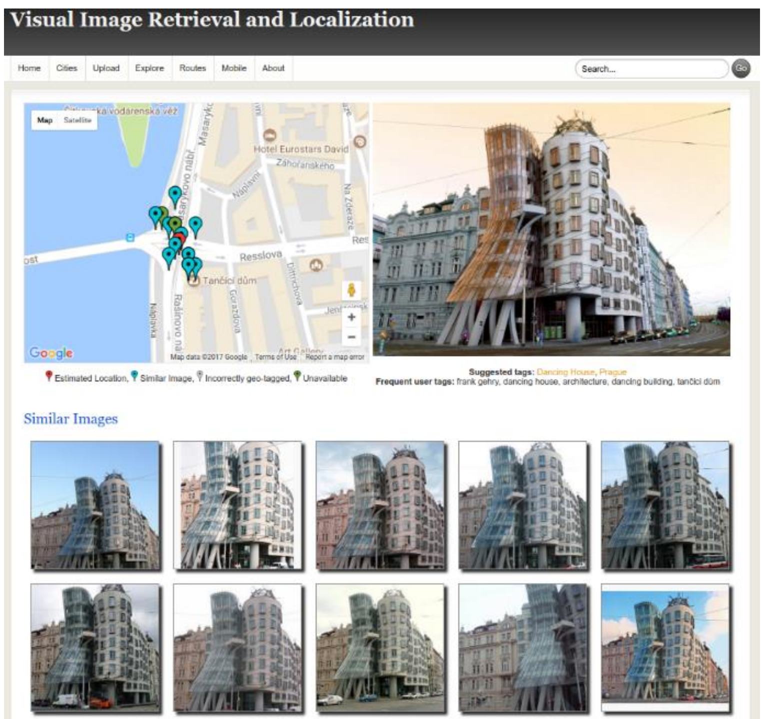
[Khosla et al. 2020]

applications

visual search



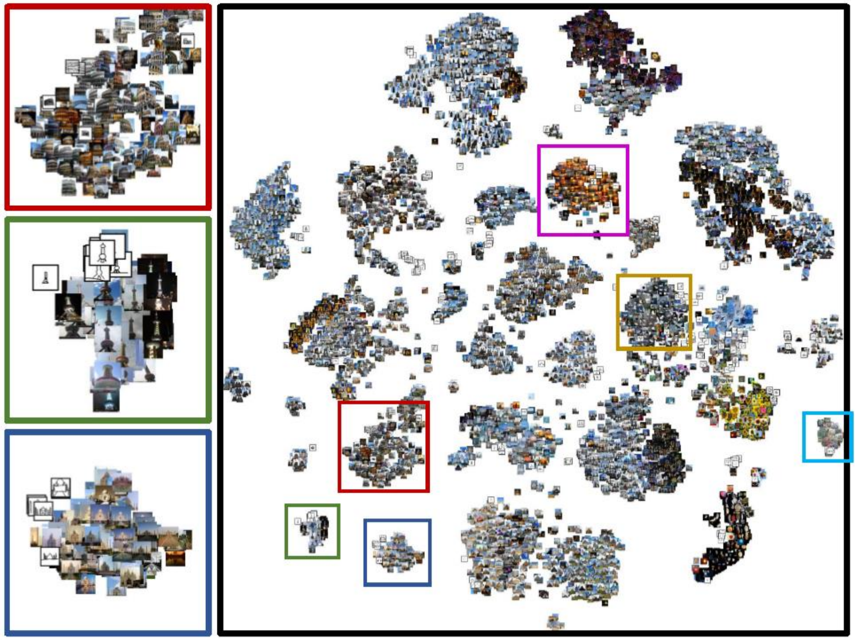
local descriptors [Mishkin'16]



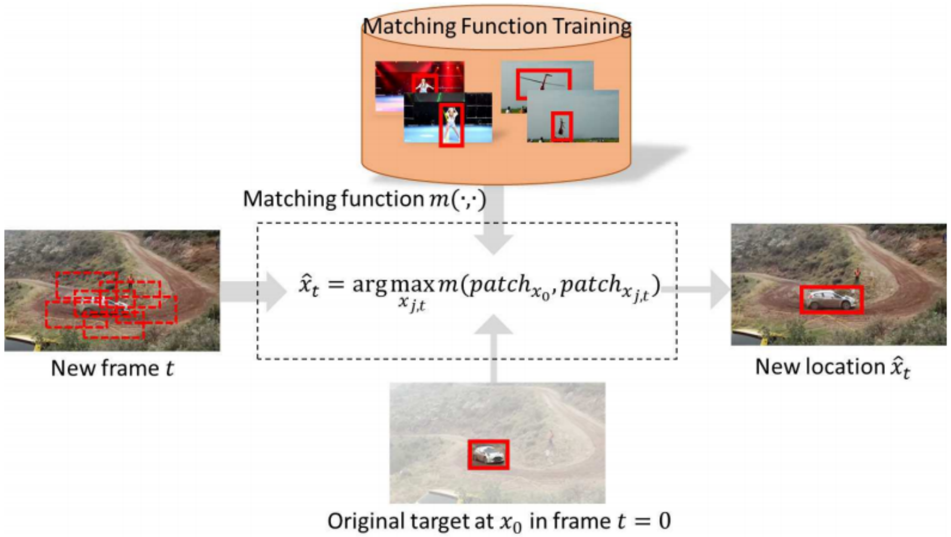
visual localization

image classification
[Song'16]

applications



data visualization



video tracking [Tao'16]

data exploration [Johnson et al.'17]