supervised similarity and representation learning

Giorgos Tolias

Czech Technical University in Prague

- pairwise similarity – supervision
- transfer learning
- metric learning
- pairwise loss
- hard negative examples
- cnn architectures

pairwise similarity

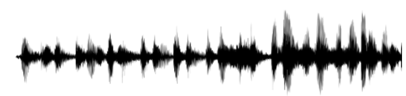
- human cognitive process involves ability to detect similarities between objects
- objects can be images, text documents, sound, etc...
- use deep learning to estimate pairwise similarity

 ↔ 

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.

↔

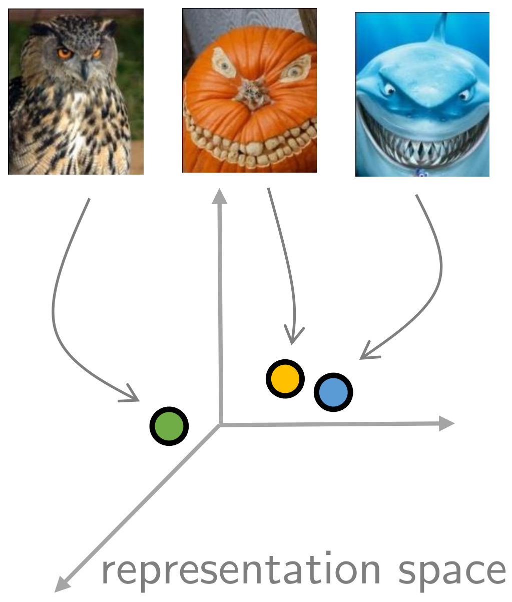
Skateboarding is an action sport that involves riding and performing tricks using a skateboard, as well as a recreational activity, an art form, an entertainment industry job, and a method of transportation.[1] Skateboarding has been shaped and influenced by many skateboarders throughout the years. A 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

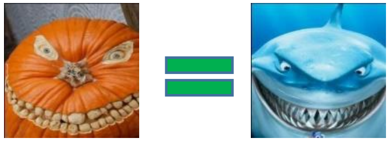
representation and similarity

- space of input examples $\mathcal{X}$
- embedding function
  - maps input examples to the representation space
  - $f : \mathcal{X} \to \mathbb{R}^D$
  - $\mathbf{x} = f(x), x \in \mathcal{X}$
- similarity measure
  - $s : \mathbb{R}^D \times \mathbb{R}^D \to \mathbb{R}$
  - symmetric: $s(\mathbf{x}, \mathbf{z}) = s(\mathbf{z}, \mathbf{x})$
- equivalently for distance
  - $d : \mathbb{R}^D \times \mathbb{R}^D \to \mathbb{R}$

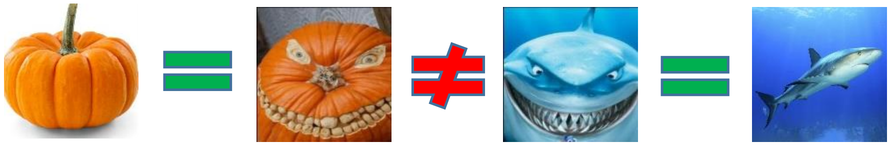


representation and similarity learning

- definition of good similarity 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



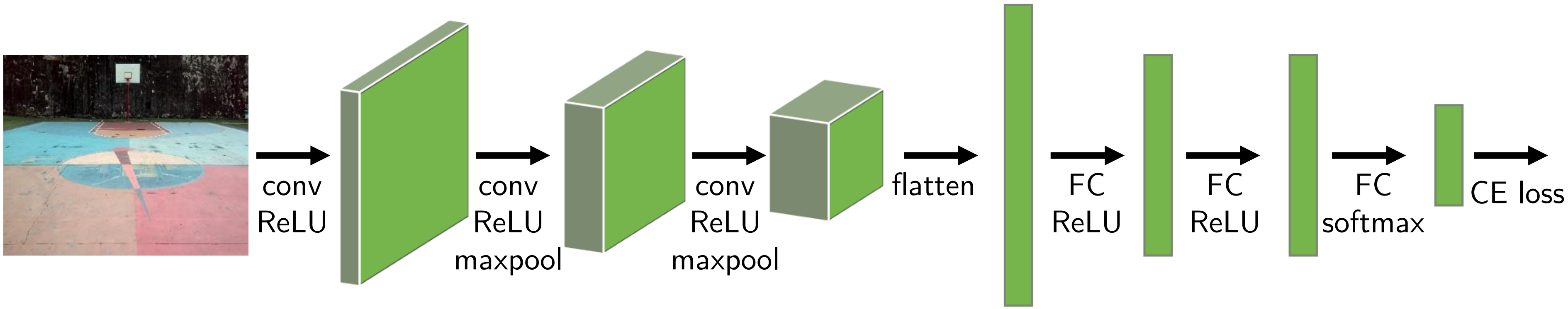
task a



task b

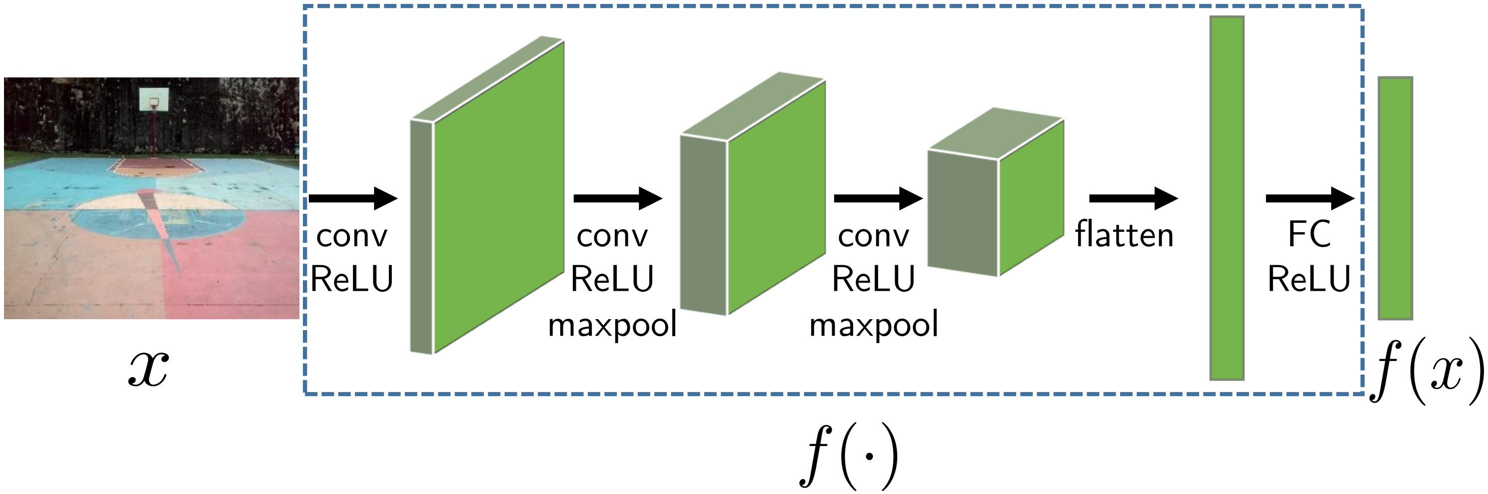
- supervised learning for
  - representation function $f(\cdot)$
  - similarity measure $s(\cdot,\cdot)$
  - both

transfer learning



- pre-trained network is given, e.g. trained for classification with cross-entropy loss

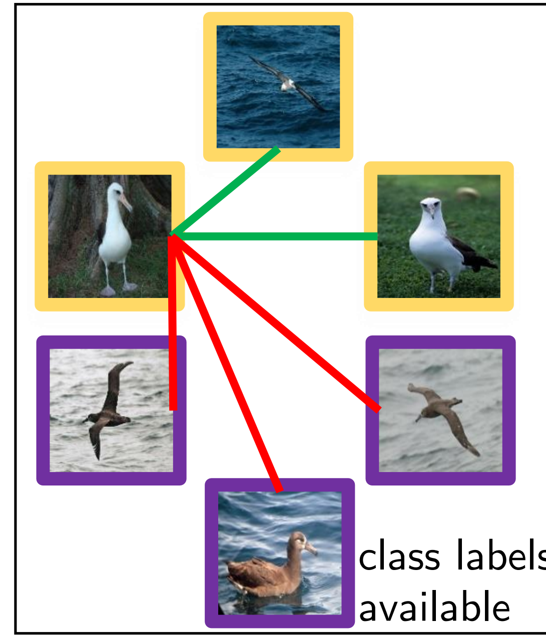
transfer learning

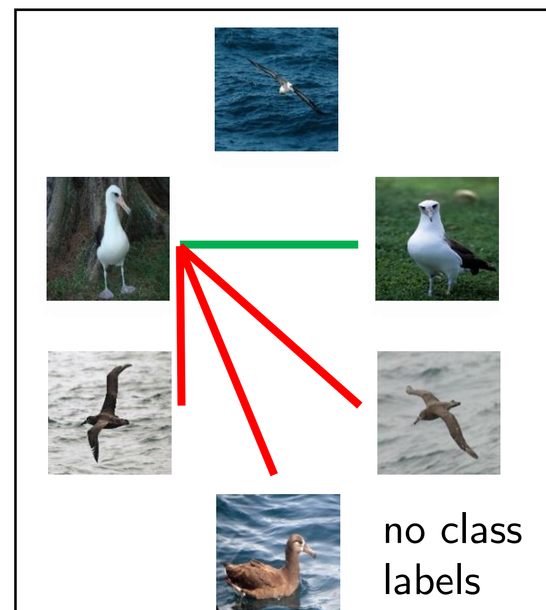


- pre-trained network is given, e.g. trained for classification with cross-entropy loss
- user internal activation vectors as representation
- use existing metrics to estimate pairwise similarity
  - Euclidean distance, Manhattan distance, cosine similarity, ...

supervision

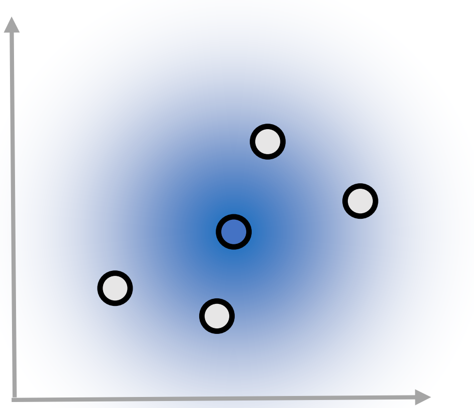
- supervision for representation/similarity learning
  - labels at the level of pairs
- similar pairs $(x_i, x_j) \in \mathcal{S}$
- dissimilar pairs $(x_i, x_j) \in \mathcal{D}$
- if discrete class labels are available
  - within class pairs are similar
  - across class pairs are dissimilar
- minimize loss
  - similarity: $\min_{\phi} \ell(s_\phi, \mathcal{S}, \mathcal{D}; \phi)$
  - representation: $\min_{\theta} \ell(f_\theta, \mathcal{S}, \mathcal{D}; \theta)$
  - both: $\min_{\theta,\phi} \ell(f_\theta, s_\phi, \mathcal{S}, \mathcal{D}; \theta, \phi)$



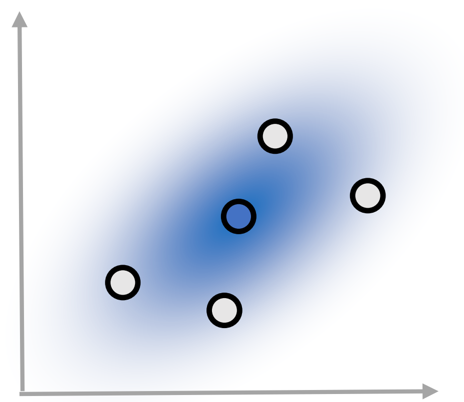


metric learning

- learn a parametric distance function from the data
- 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

- learn a parametric distance function from the data

- 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

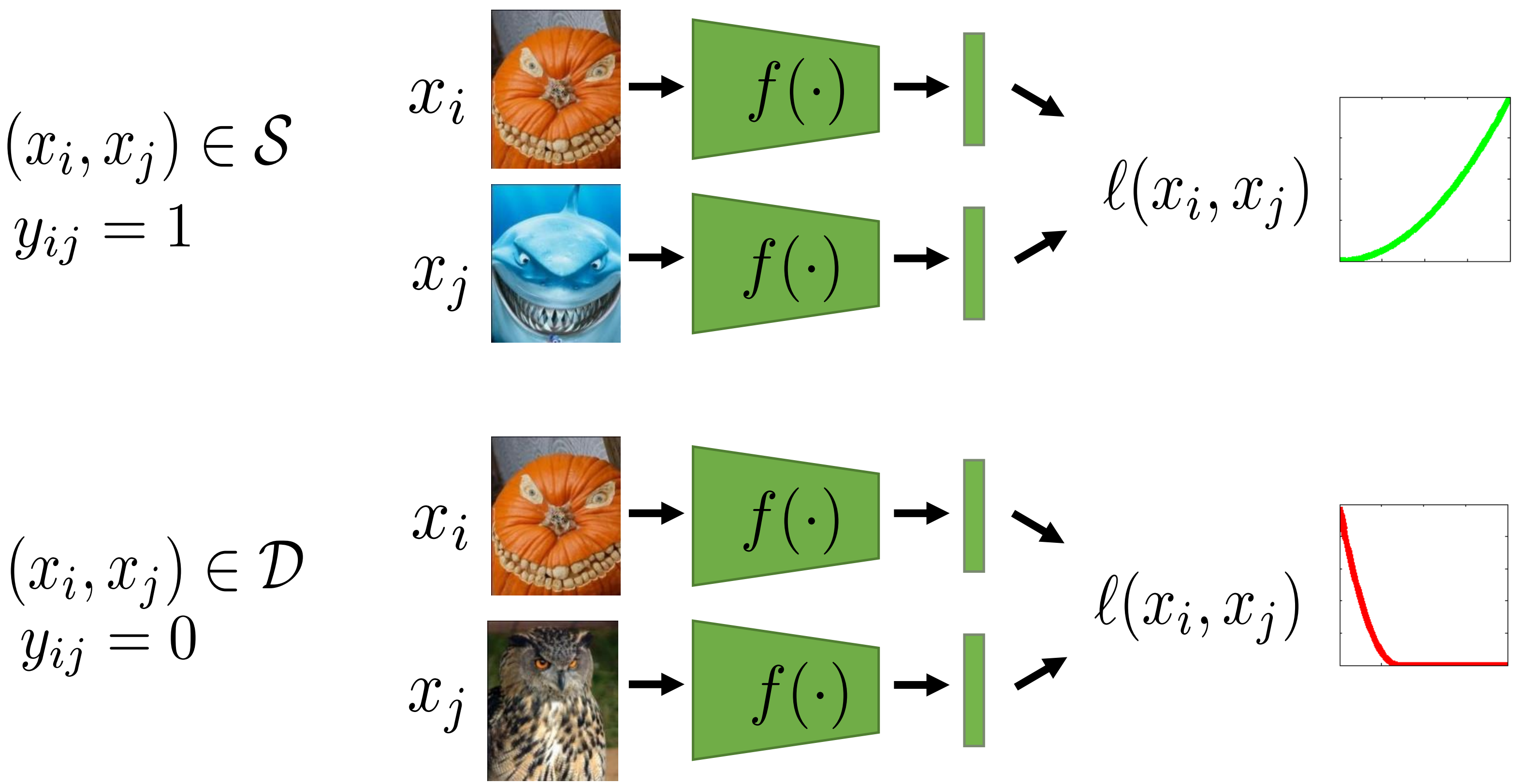
- general case: $f : \mathbb{R}^D \to \mathbb{R}^{D'}$ is a feed-forward network

- input examples are images: $f : \mathcal{X} \to \mathbb{R}^{D'}$ is a CNN

contrastive loss

[Hadsell et al. 20006]

- two branch network; 2 networks that share weights



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

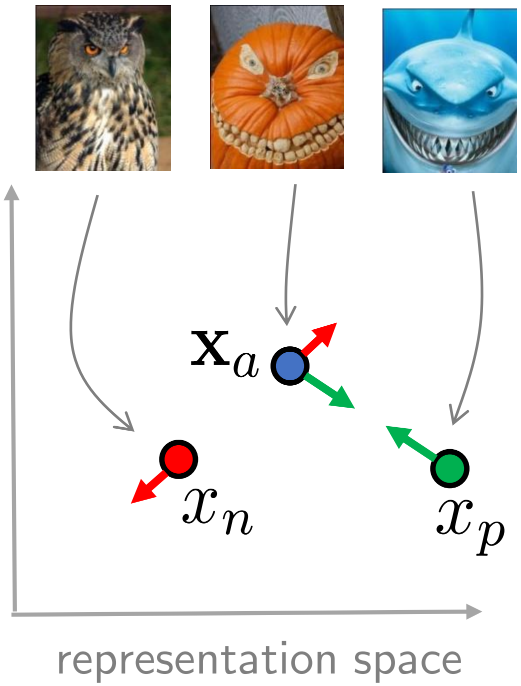
contrastive loss

- 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

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



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

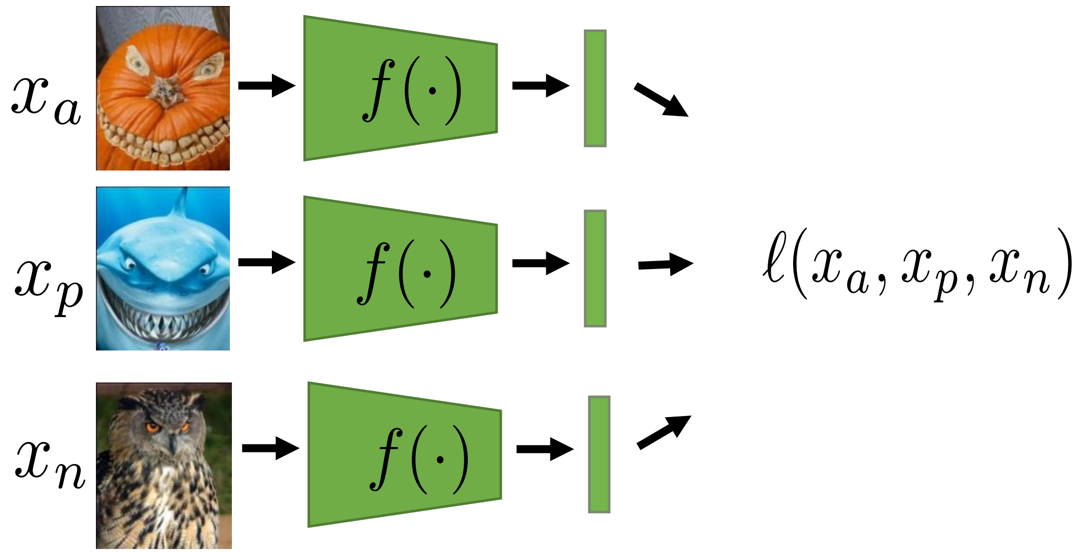
triplet loss

[Schroff et al. 2015]

- three branch network; 3 networks that share weights

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

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



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

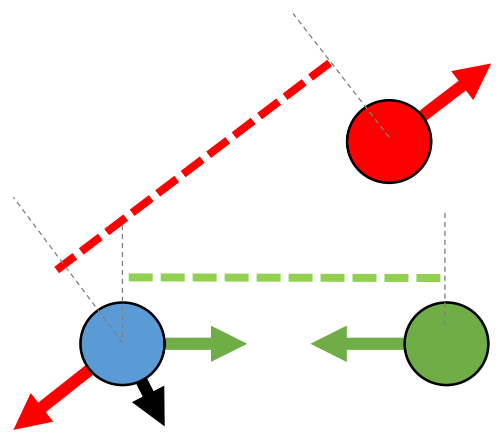
triplet loss

- gradients

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

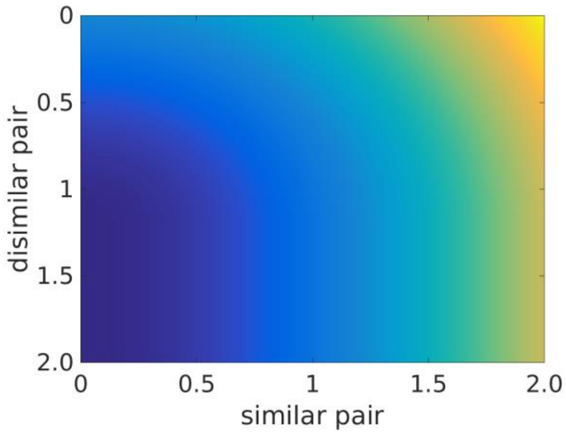
$$\frac{\partial \ell}{\partial \mathbf{x}_n} = 2(\mathbf{x}_a - \mathbf{x}_n)$$

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

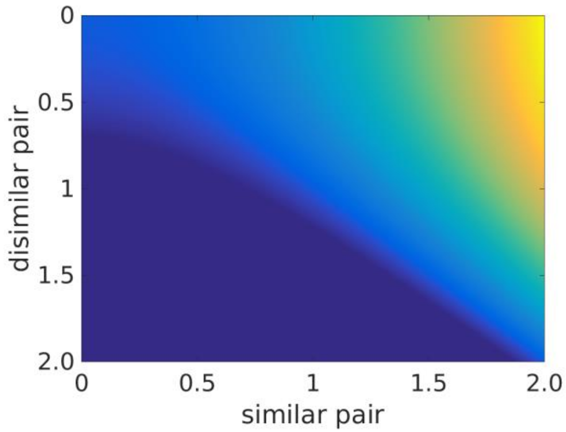


$$\ell(x_a, x_p, x_n) = [||\mathbf{x}_a - \mathbf{x}_p||_2^2 - ||\mathbf{x}_a - \mathbf{x}_n||_2^2 + \alpha]_+$$

pairwise losses

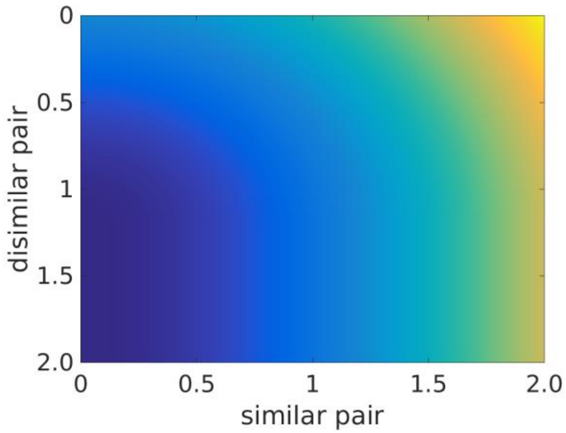


contrastive

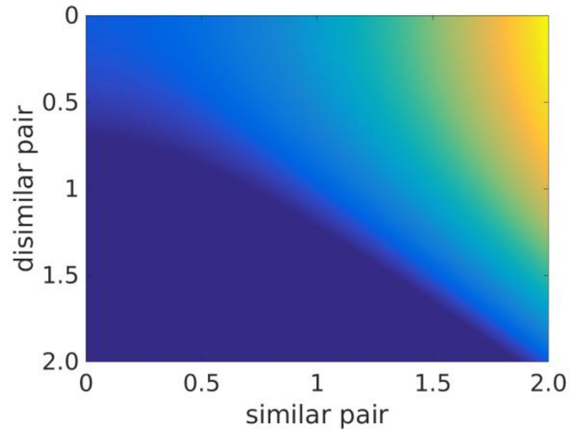


triplet

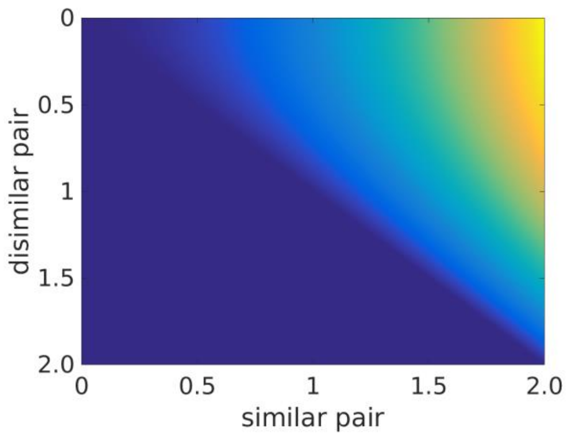
pairwise losses

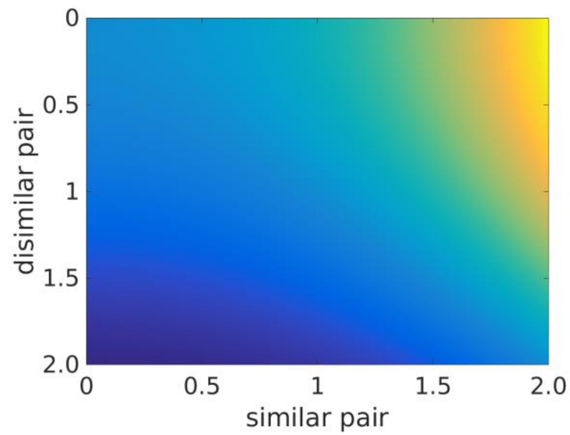


contrastive



triplet





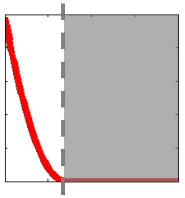
$[\mathbf{x}_a^\top \mathbf{x}_n - \mathbf{x}_a^\top \mathbf{x}_p]_+$

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

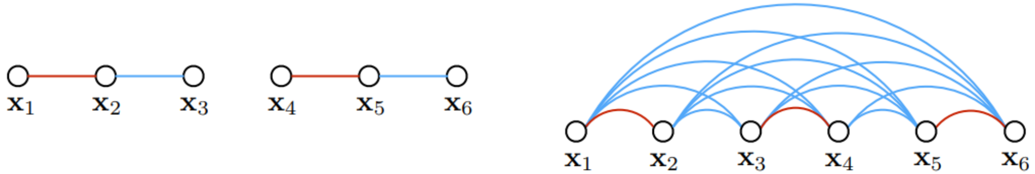
multiple negatives: $\log(1 + \sum_i e^{\mathbf{x}_a^\top \mathbf{x}_{n_i} - \mathbf{x}_a^\top \mathbf{x}_p})$

hard negatives

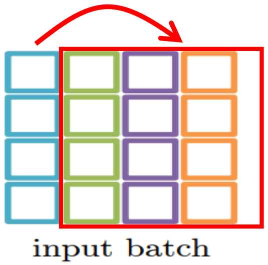
- include all pairs: computationally costly
- random sampling: zero loss for many pairs/triplets



- hard negatives: dissimilar pairs, nearby in the representation space
- repeat hard negative mining during training
- include all negatives in a batch [Song et al. 2016]

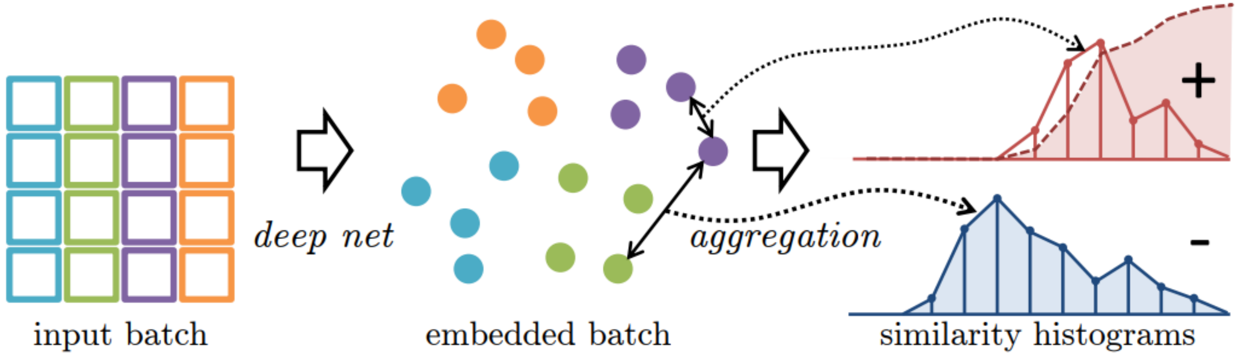


- within batch hardest

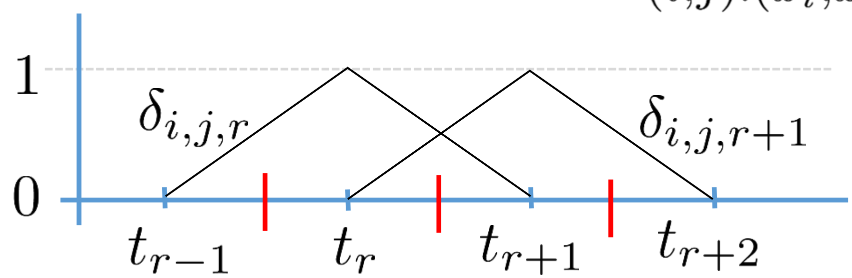


- semi-hard mining [Schroff et al. 2015]
- distance-weighted sampling [Wu et al. 2017]

histogram loss [Ustinova & Lempitsky, 2016]



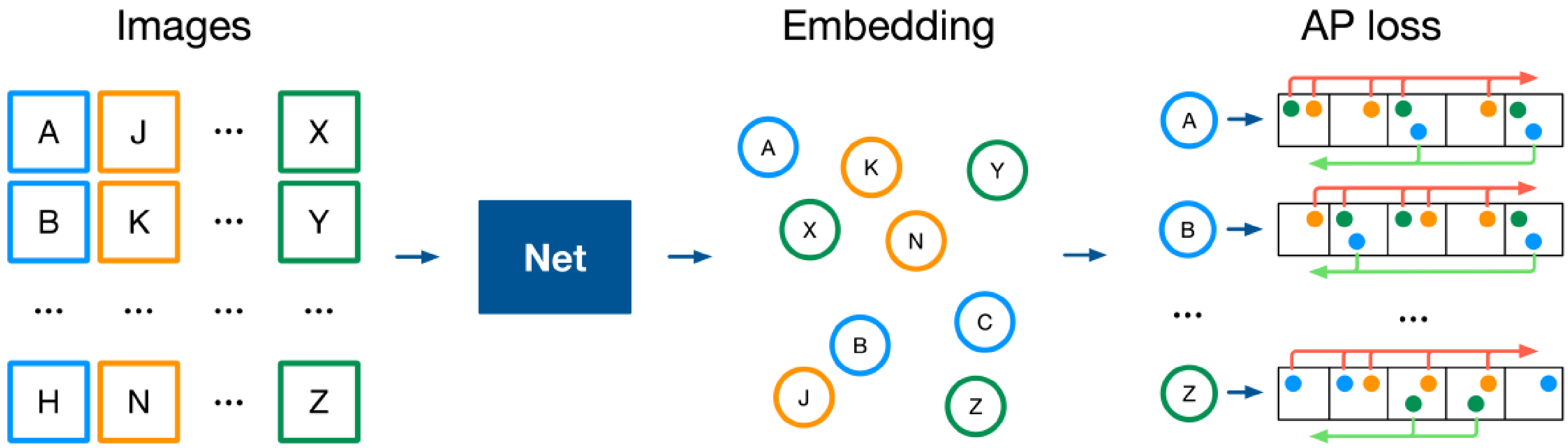
histogram for similar pairs: $h_r^{\mathcal{S}} = \frac{1}{|\mathcal{S}|} \sum_{(i,j):(x_i,x_j)\in\mathcal{S}} \delta_{i,j,r}$



histogram for disimilar pairs: equivalently

$$\ell(\mathcal{S},\mathcal{D}) = \sum_{r=1}^{R}\left(h_r^{\mathcal{D}}\sum_{q=1}^{r} h_q^{\mathcal{S}}\right) = \sum_{r=1}^{R}\left(h_r^{\mathcal{D}}\phi_q^{\mathcal{S}}\right)$$

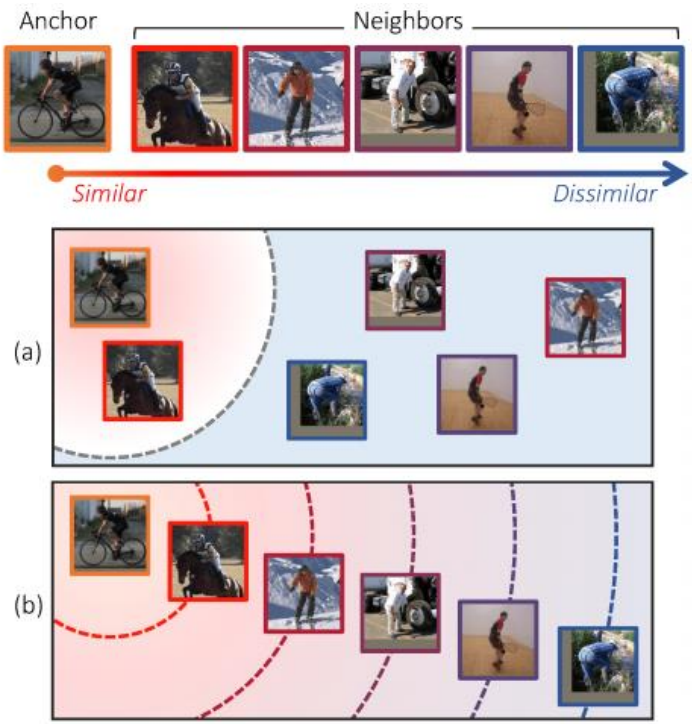
average precision loss



[Revaud et al, 2019]

- optimize objectives closer to the target task
- average precision: optimize a differentiable approximation

beyond binary supervision



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

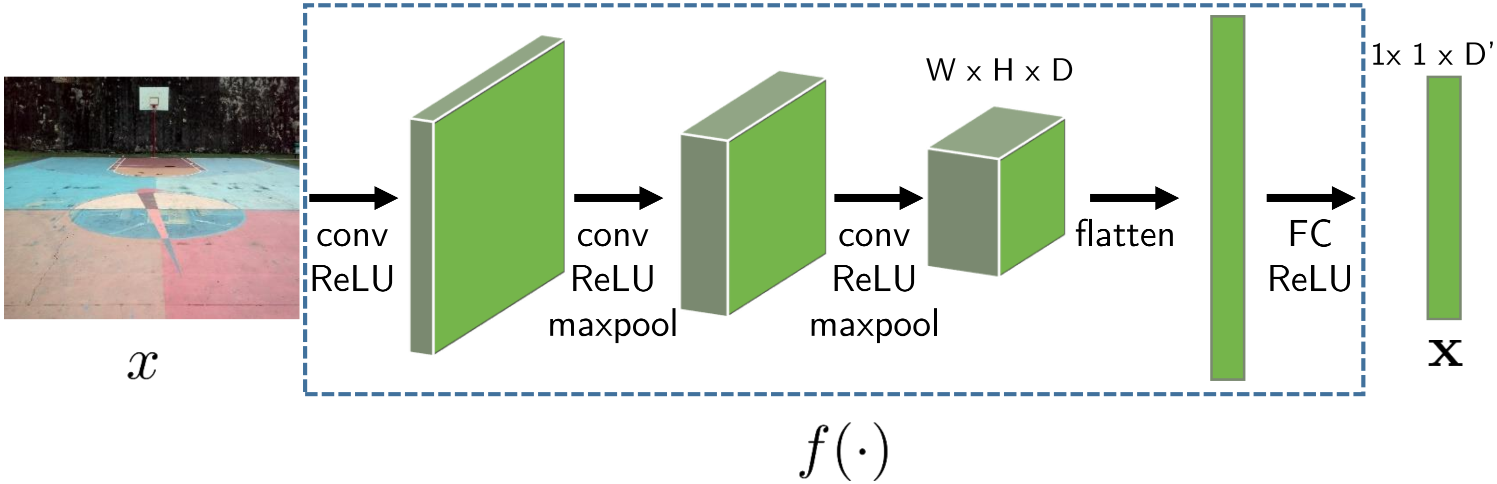
distance ratio representation space

distance ratio label space

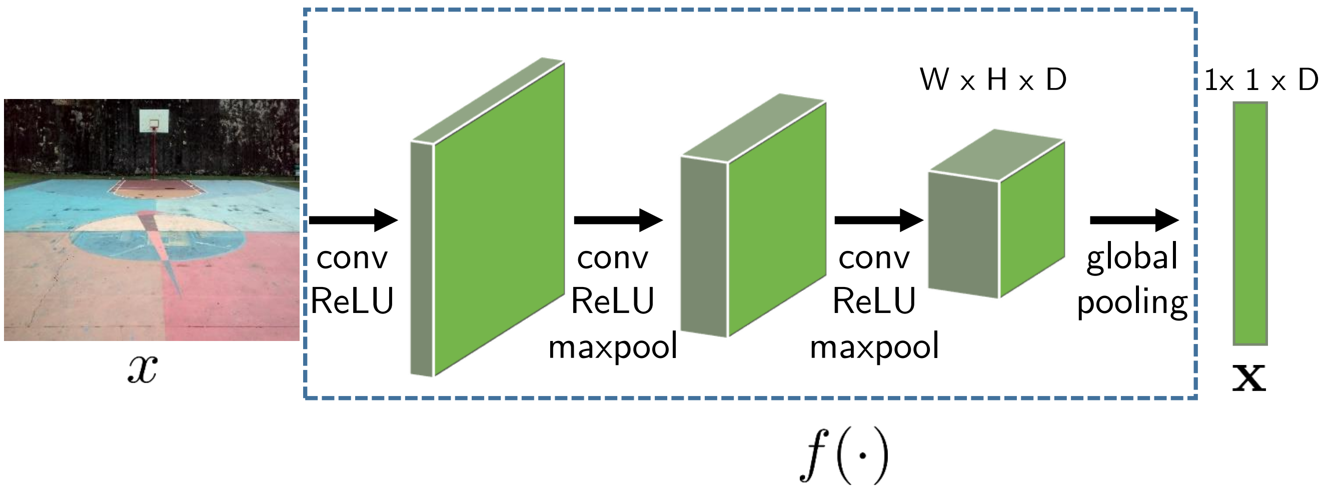
[Kim et al.'19]

cnn representation and similarity

- CNN with FC layer

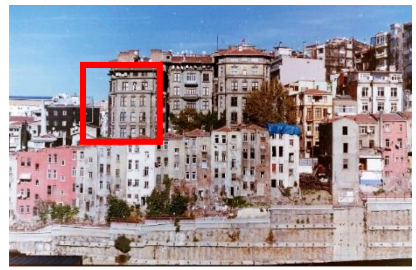


- Fully Convolutional Network (FCN)



global max pooling - representation

input image x



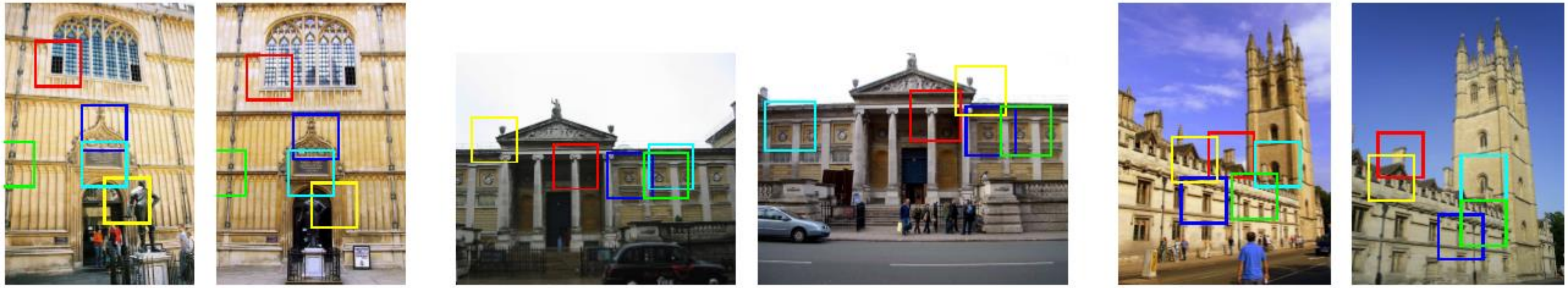
conv$_5$ filter 1 conv$_5$ filter 2 .... conv$_5$ filter i .... conv$_5$ filter D

max

- representation: $\mathbf{x} = [\mathbf{x}_{(1)}, \ldots, \mathbf{x}_{(i)}, \ldots, \mathbf{x}_{(D)}] \in \mathbb{R}^D$

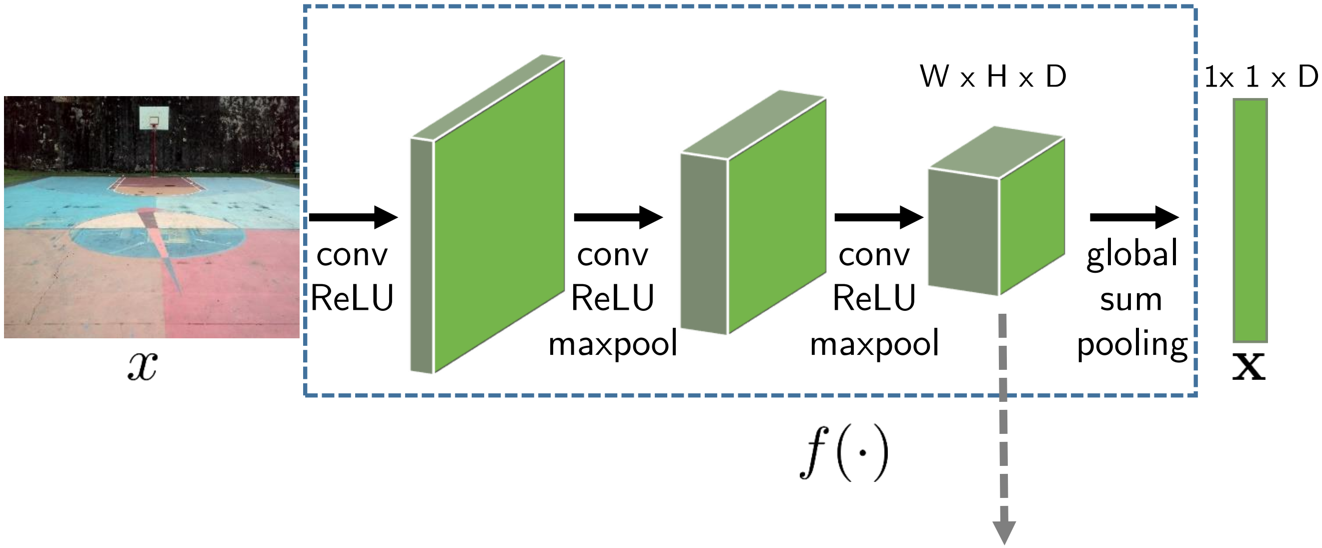
global max pooling - similarity

- cosine similarity
  - $s(x, z) = \frac{\mathbf{x}^\top \mathbf{z}}{||\mathbf{x}||_2||\mathbf{z}||_2} \propto \mathbf{x}^\top \mathbf{z}$
  - $\mathbf{v} = [\mathbf{v}_{(1)}, \ldots, \mathbf{v}_{(i)}, \ldots, \mathbf{v}_{(D)}] = [\mathbf{x}_{(1)}\mathbf{z}_{(1)}, \ldots, \mathbf{x}_{(i)}\mathbf{z}_{(i)}, \ldots, \mathbf{x}_{(D)}\mathbf{z}_{(D)}]$

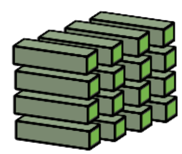


receptive field for activations corresponding to largest $\mathbf{v}_{(i)}$

global sum pooling - representation



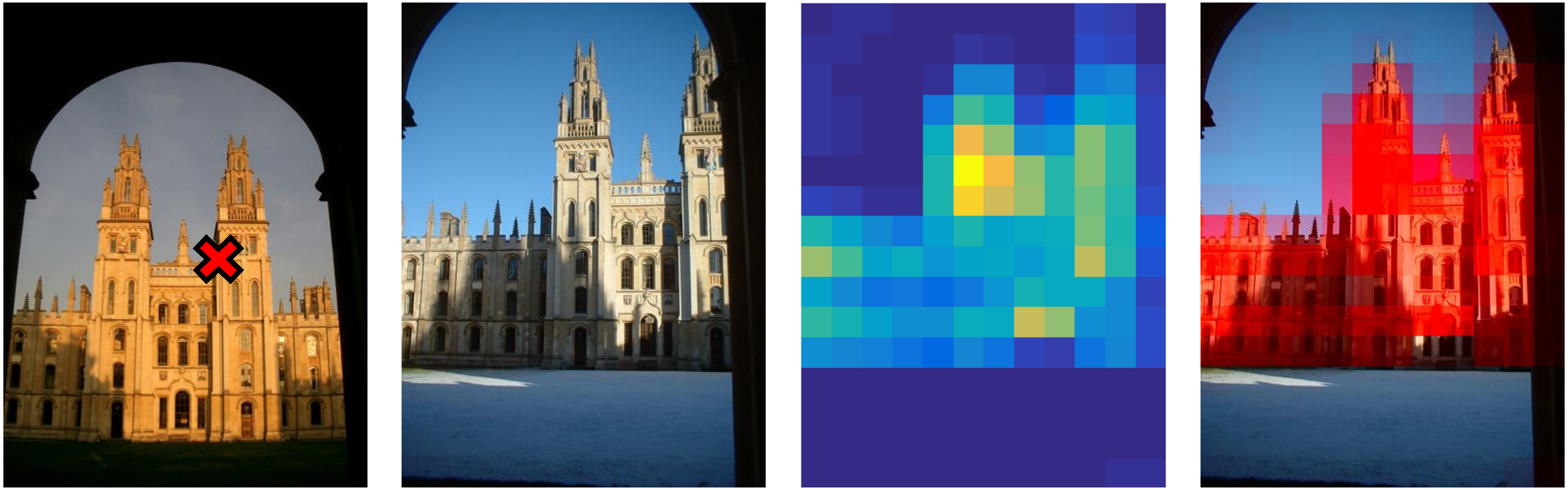
$$G(x) = \{\mathbf{u} : \mathbf{u} \in \mathbb{R}^D\}$$
with $|G(x)| = W \times H$



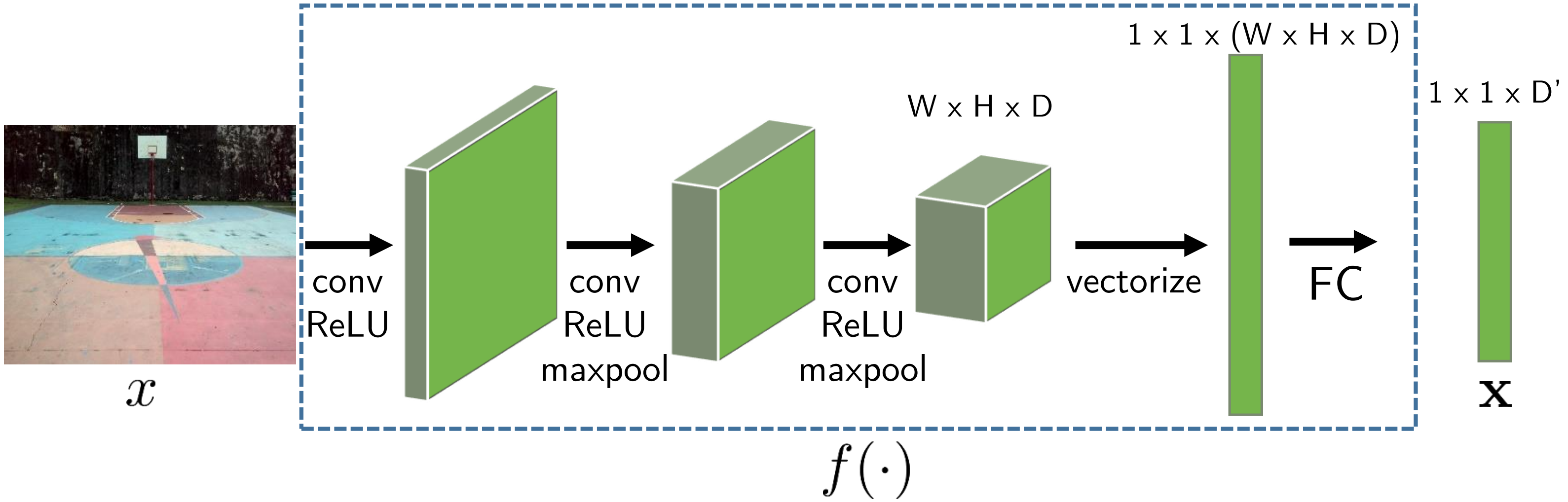
- representation: $\mathbf{x} = \sum_{\mathbf{u} \in G(x)} \mathbf{u}$

global sum pooling - similarity

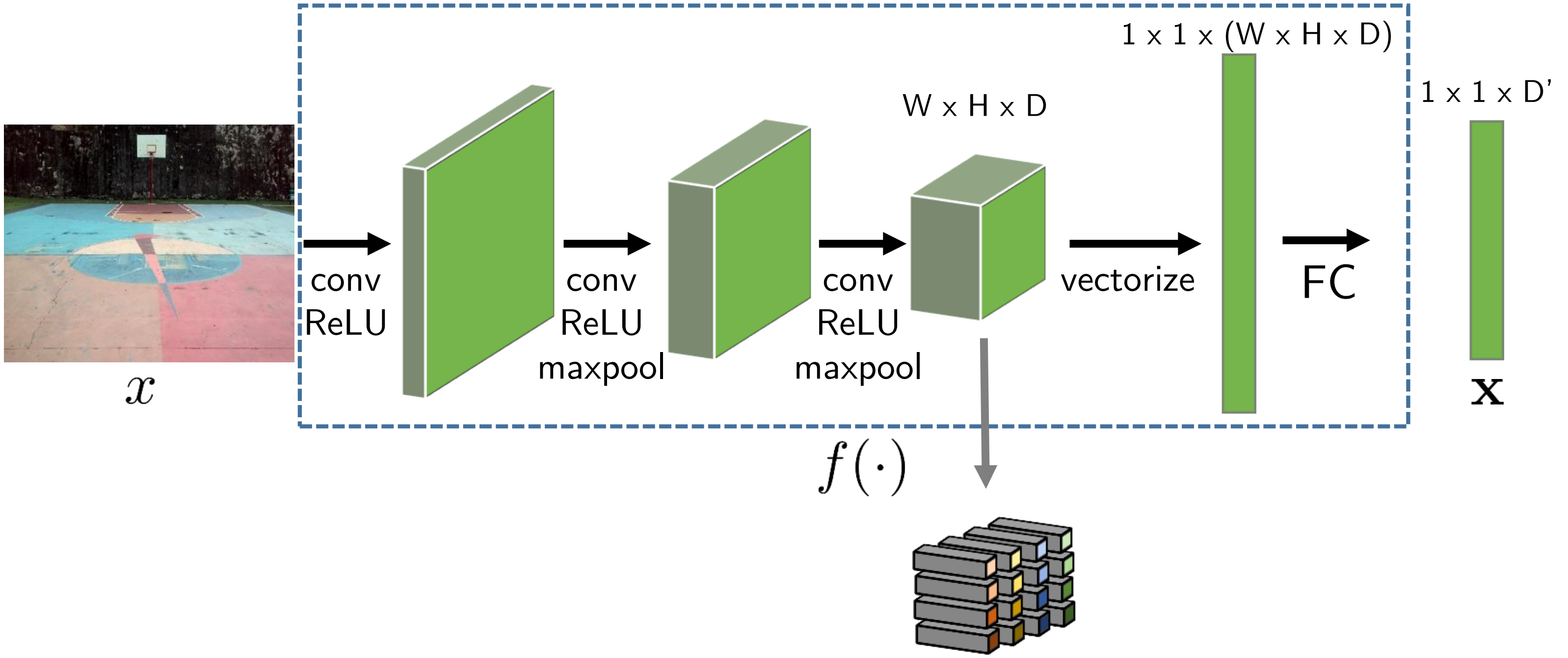
- cosine similarity
  - $s(x, z) = \frac{\mathbf{x}^\top \mathbf{z}}{||\mathbf{x}||_2 ||\mathbf{z}||_2} \propto \mathbf{x}^\top \mathbf{z}$
  - $\mathbf{x}^\top \mathbf{z} = \left(\sum_{\mathbf{u} \in G(x)} \mathbf{u} \right)^\top \sum_{\mathbf{v} \in G(z)} \mathbf{v} = \sum_{\mathbf{u} \in G(x)} \sum_{\mathbf{v} \in G(z)} \mathbf{u}^\top \mathbf{v}$
- similarity is translation invariant



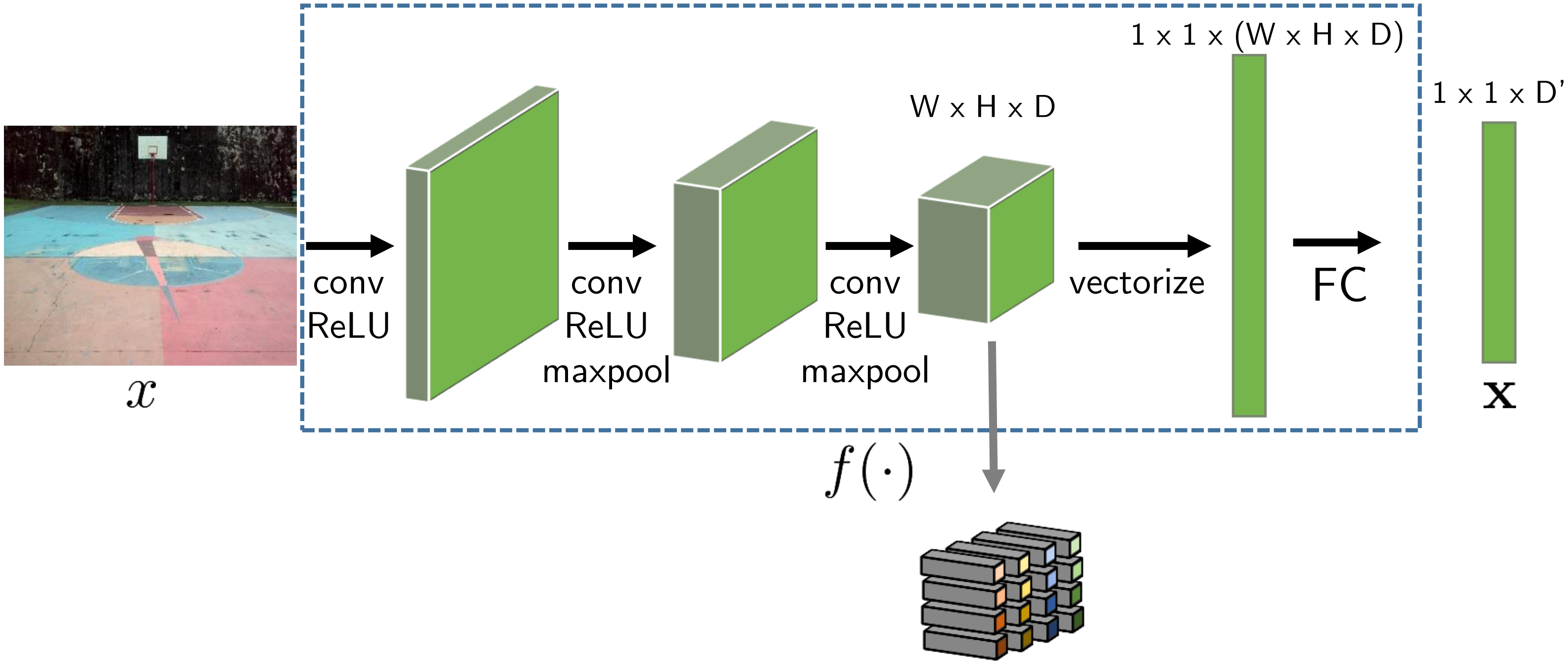
vectorize and fc layer - representation



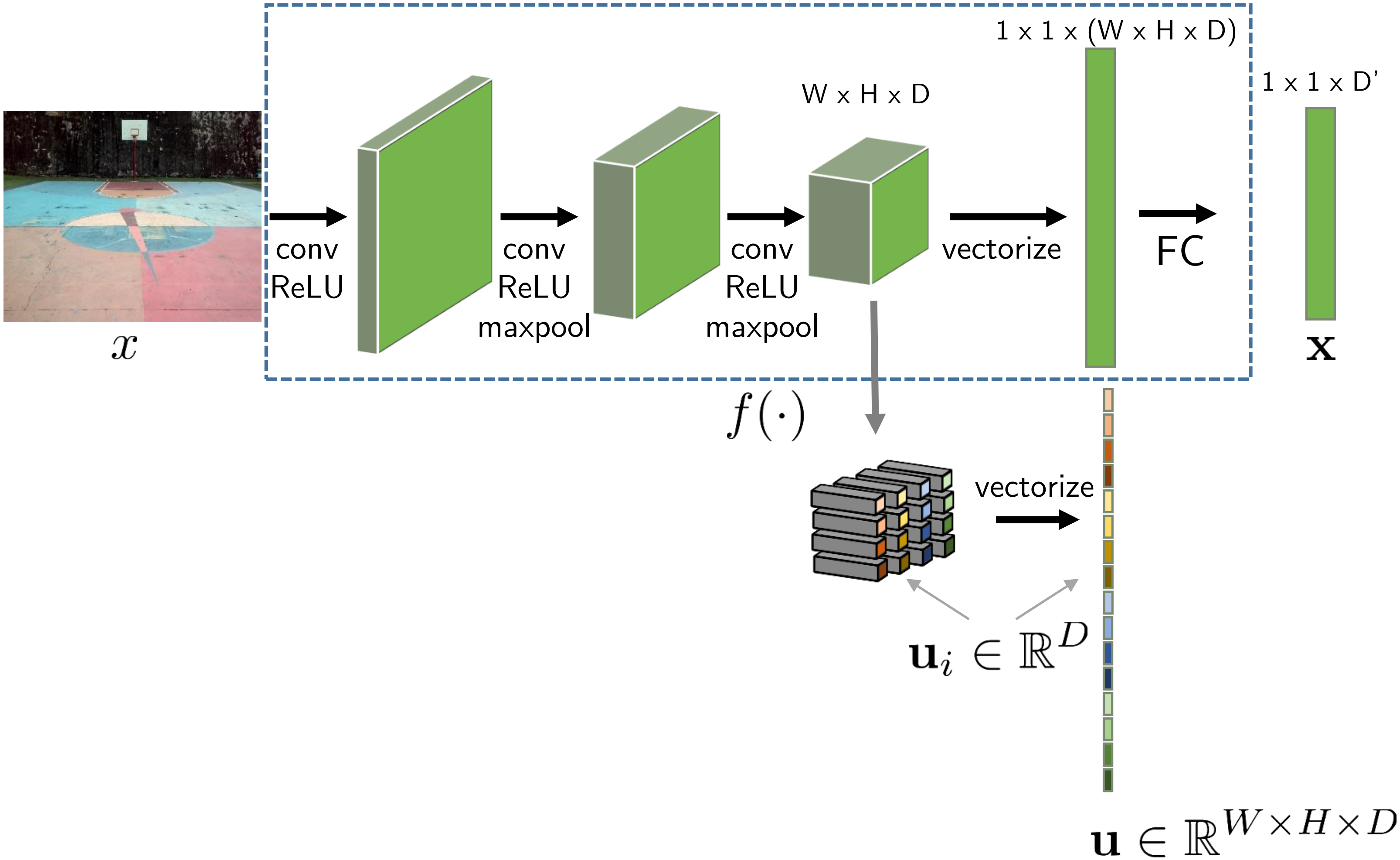
vectorize and fc layer - representation



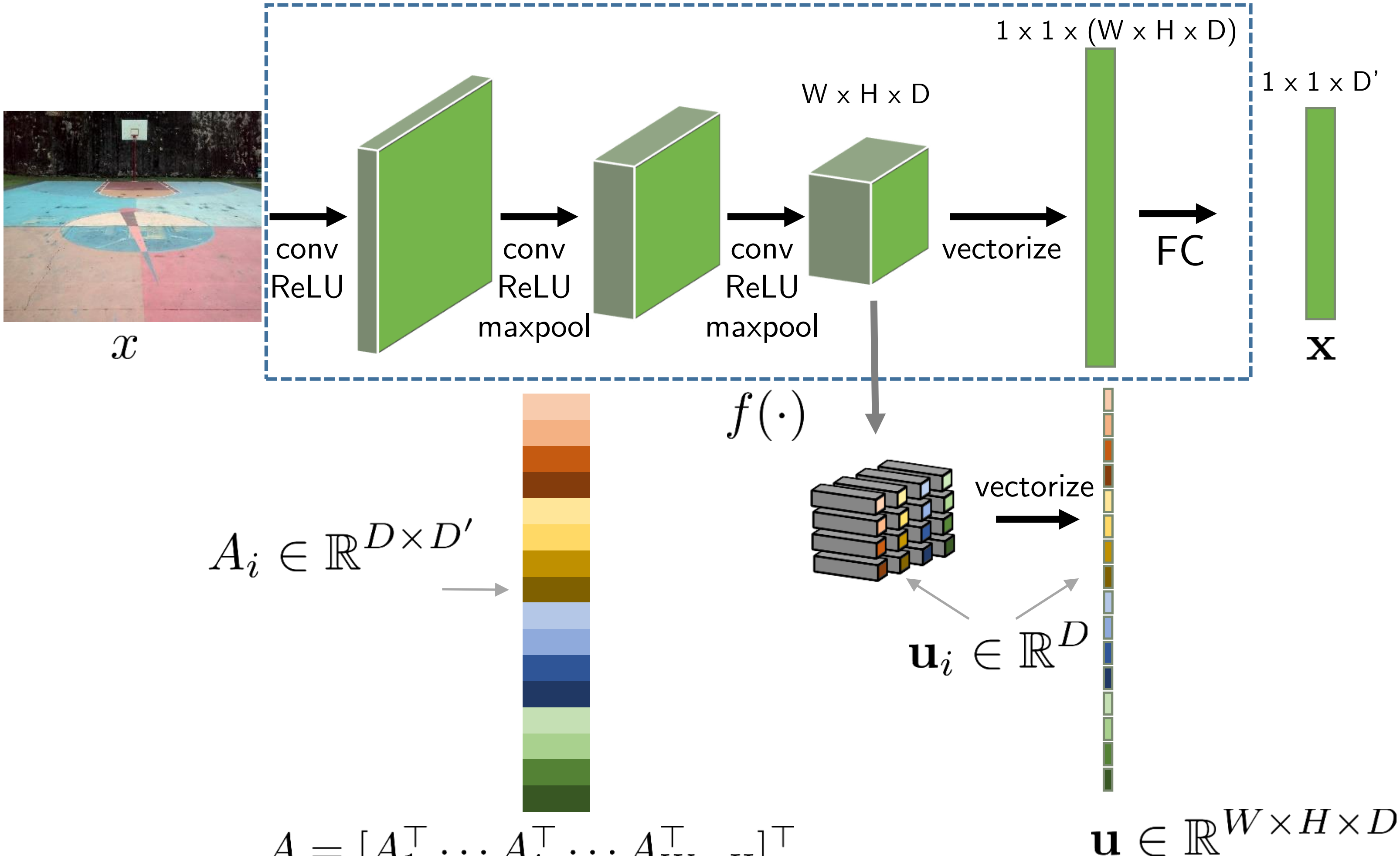
vectorize and fc layer - representation



vectorize and fc layer - representation



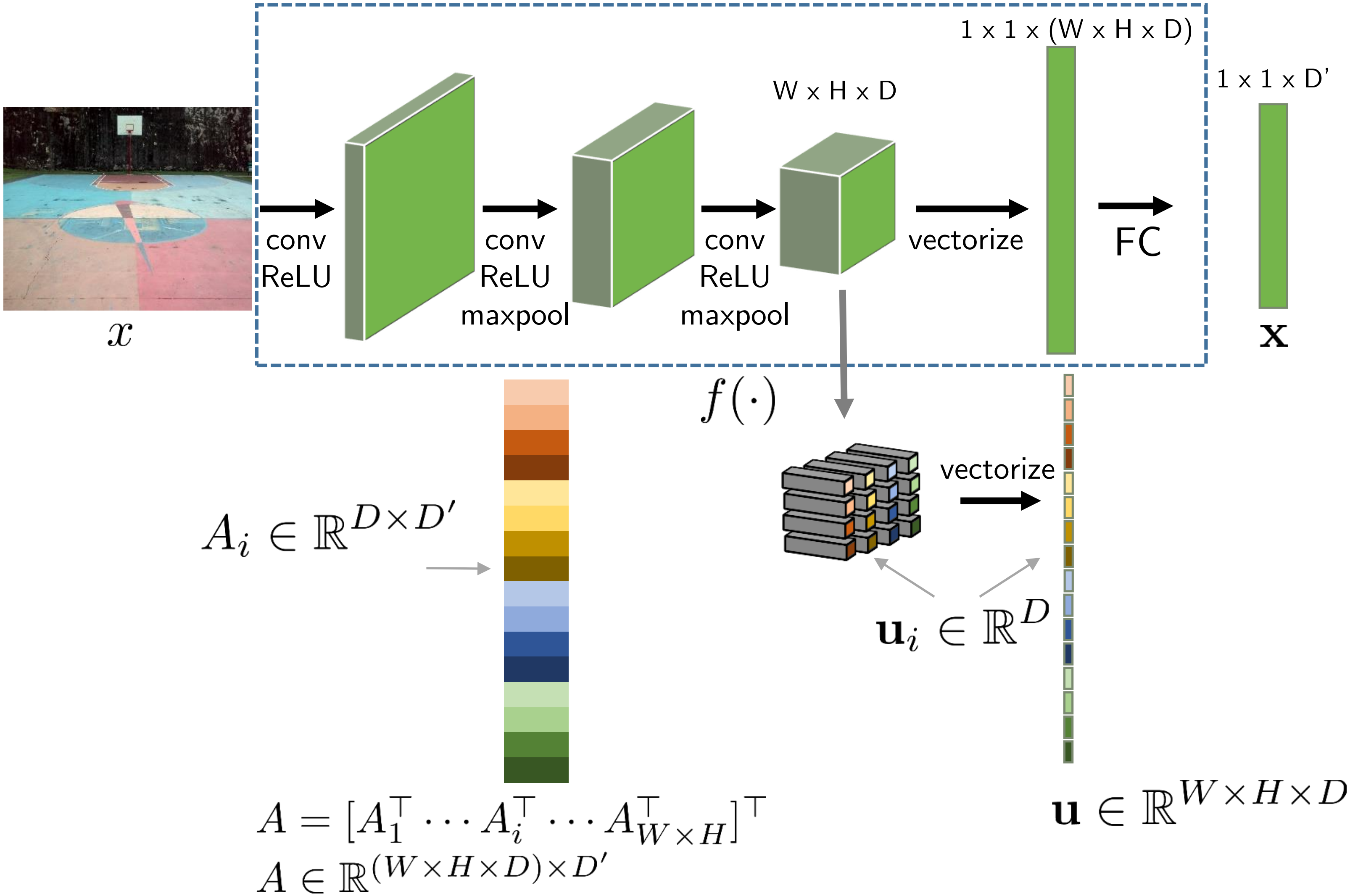
vectorize and fc layer - representation



$$A = [A_1^\top \cdots A_i^\top \cdots A_{W \times H}^\top]^\top$$

$$A \in \mathbb{R}^{(W \times H \times D) \times D'}$$

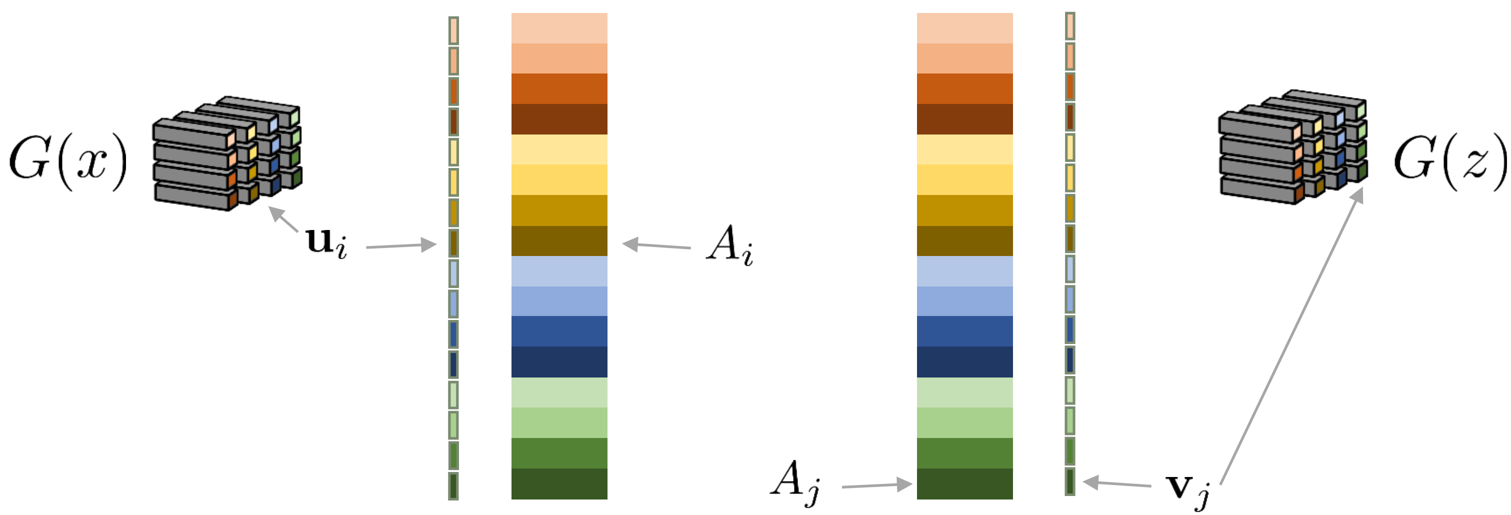
vectorize and fc layer - representation



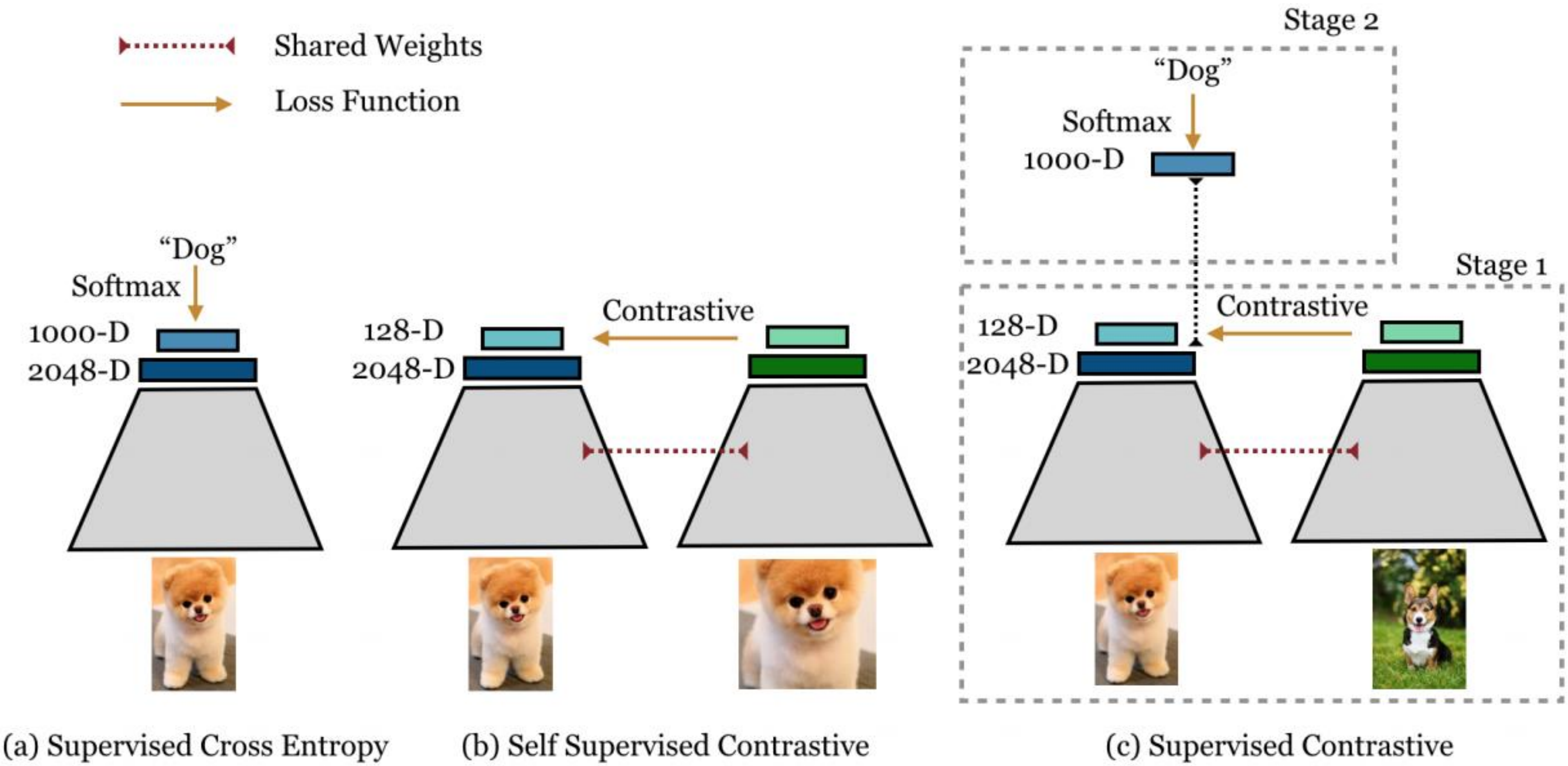
- representation: $\mathbf{x} = A^\top \mathbf{u} = \sum_{\mathbf{u}_i \in G(x)} A_i^\top \mathbf{u}_i \in \mathbb{R}^{D'}$

vectorize and fc layer - similarity

- cosine similarity
  - $s(x, z) = \frac{\mathbf{x}^\top \mathbf{z}}{||\mathbf{x}||_2 ||\mathbf{z}||_2} \propto \mathbf{x}^\top \mathbf{z}$
  - $$\mathbf{x}^\top \mathbf{z} = \left(\sum_{\mathbf{u}_i \in G(x)} A_i^\top \mathbf{u}_i \right)^\top \sum_{\mathbf{v}_j \in G(z)} A_j^\top \mathbf{v}_j$$
    $$= \sum_{\mathbf{u}_i \in G(x)} \sum_{\mathbf{v}_j \in G(z)} \mathbf{u}_i^\top A_i A_j^\top \mathbf{v}_j$$
- similarity is translation variant



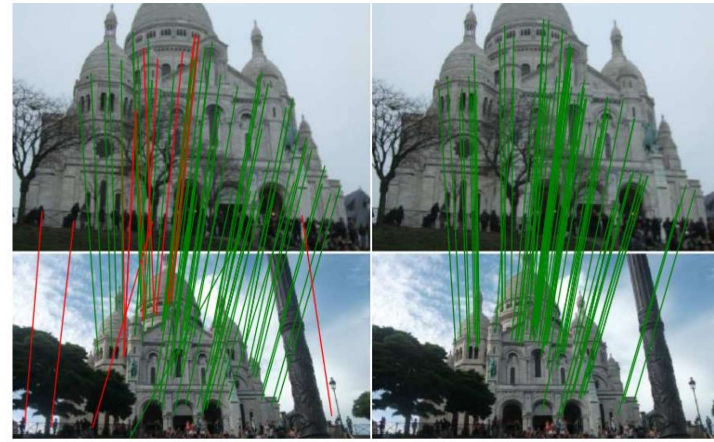
beyond supervised representation learning



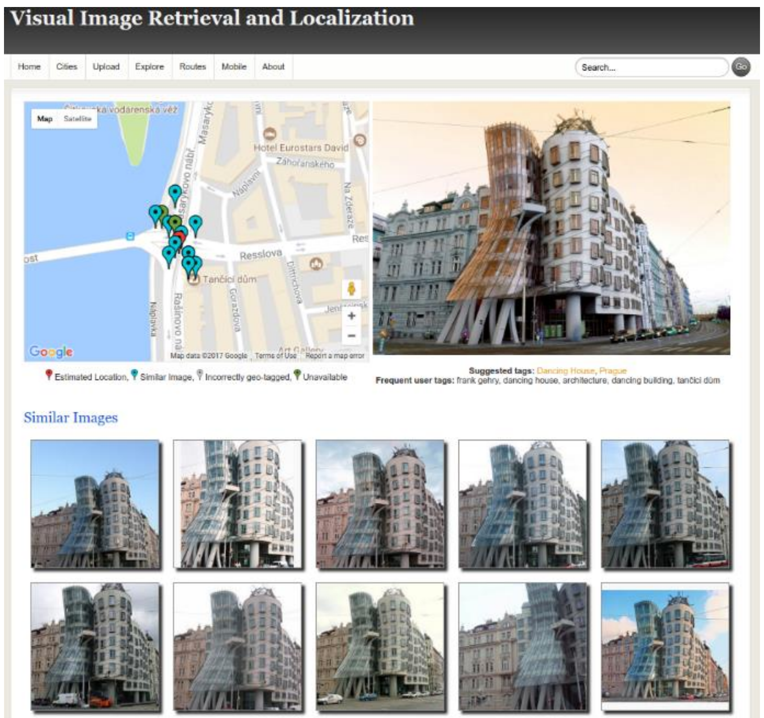
[Khosla et al. 2020]

applications

visual search



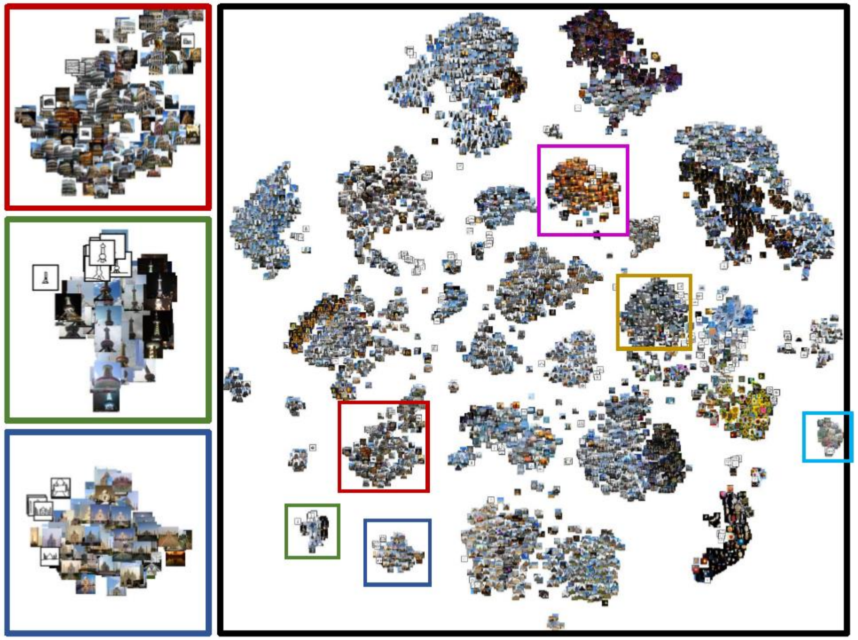
local descriptors [Mishkin'16]



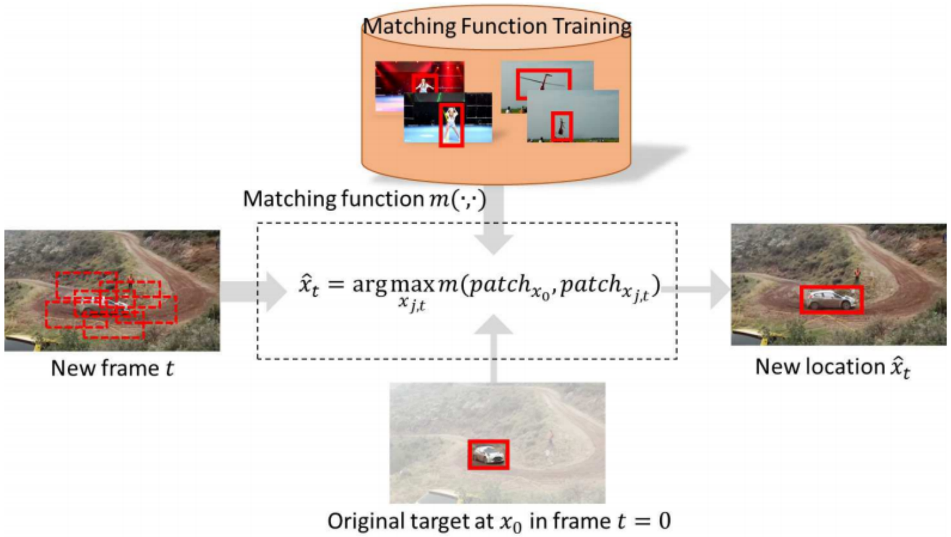
visual localization

image classification
[Song'16]

applications



data visualization



video tracking [Tao'16]

data exploration [Johnson et al.'17]