# Learning Deep Embeddings with Histogram Loss

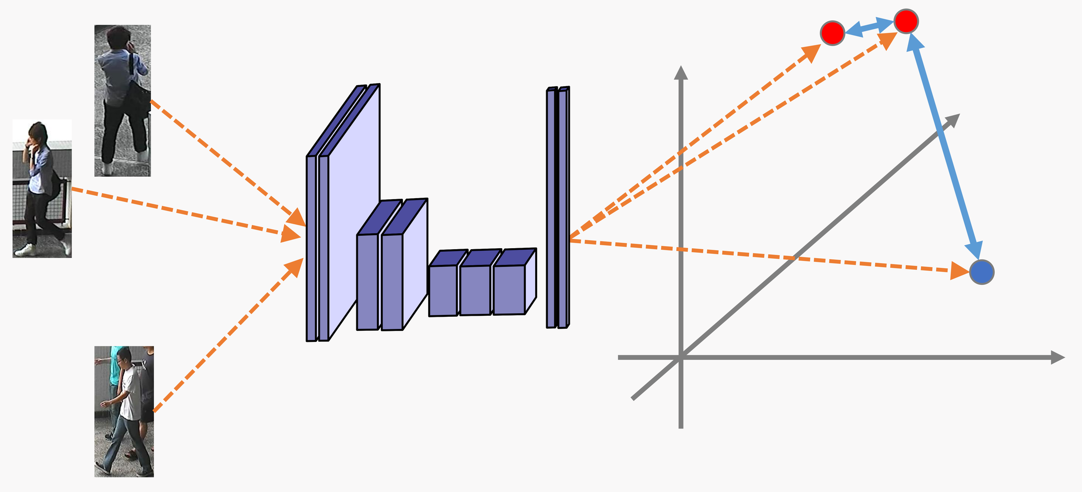
Evgeniya Ustinova, Victor Lempitsky
Skolkovo Institute of Science and Technology

## Introduction

Deep feed-forward embeddings play a crucial role across a wide range of tasks and applications

Input images are mapped into a high-dimensional *descriptor* space.

Learning *objective* = achieve the proximity of semantically-related images + avoid the proximity of semantically-unrelated images.



## Existing Loss functions

Popular loss choices implementing the learning objective:

- *Classification losses.* Train a classifier network. Use an intermediate representation as descriptors.
- *Pair-wise losses* sample pairs of training points and score them independently. E.g. *contrastive loss*:

$$L(i,j) = \begin{cases} \frac{1}{2}\left\|x_i - x_j\right\|_2^2, & y_{i,j} = 1 \\ \frac{1}{2}\max(0, M - \left\|x_i - x_j\right\|_2)^2, & y_{i,j} = -1 \end{cases}$$

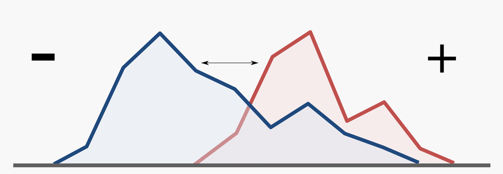
- *Triplet losses* sample triplets of examples and compare "positive" and "negative" distances within each triplet : $L(i,j,k) = \max(0, \alpha + \left\|x_i - x_j\right\|_2^2 - \|x_i - x_k\|_2^2)$, $y_{i,j} = 1, y_{i,k} = -1$
- *Quadruplet losses* compare distances for two independent pairs ("positive" and "negative").

## Problems of the existing approaches

Existing losses suffer from the following problems:

- Need to tune hyper-parameters of loss functions
- Need to pretrain the network (e.g. with classification)
- Computing the loss value for every triplet/quadruplet may be inefficient

## Our approach (overview)



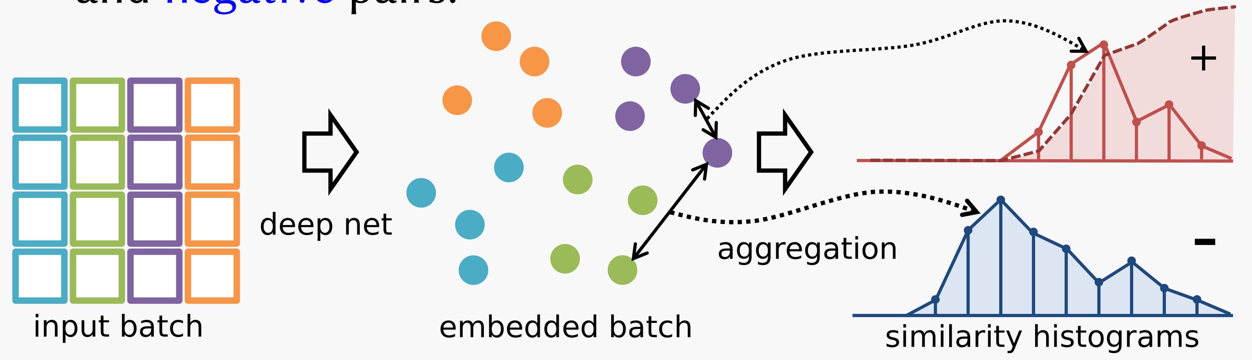

**Idea:** try to make all negative similarities less than all positive similarities without using any kinds of thresholds.

**Implementation:** for a given batch, estimate the 1D distributions of the positive and negative similarities. Compute the loss as the overlap between these two distributions.

**Our loss** = the probability of an arbitrary positive pair to have lower similarity than an arbitrary negative pair.

## Computing the loss

- Estimate distributions $p^+$ and $p^-$ for similarities in positive and negative pairs.




- Estimate the probability of the similarity in a random negative pair to be more than the similarity in a random positive pair:

$$p_{\text{reverse}} = \int_{-1}^{1} p^-(x)\left[\int_{-1}^{x} p^+(y)\,dy\right] dx = \mathbb{E}_{x\sim p^-}[\Phi^+(x)]\,, \quad (1)$$

where $\Phi^+(x)$ is the CDF (cumulative density function) of $p^+(x)$.

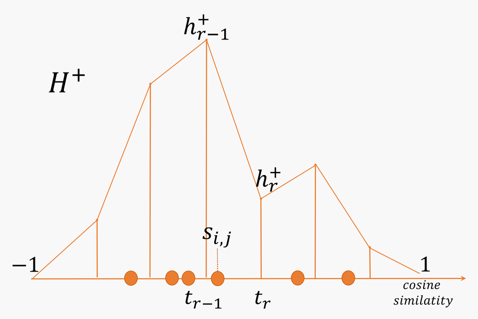
- The integral $(1)$ can then be approximated and computed as:

$$L(X,\theta) = \sum_{r=1}^{R}\left(h_r^- \sum_{q=1}^{r} h_q^+\right) = \sum_{r=1}^{R} h_r^- \phi_r^+\,, \quad (2)$$

**Relation to quadruplet loss:** Our loss takes into account all the quadruplets in the batch. Iterating across all the quadruplets is avoided. Another efficient loss based on quadruplets:

Tadmor *et al. Learning a Metric Embedding for Face Recognition using the Multibatch Method.* NIPS, 2016.

## Histogram estimation




- *Assumption:* descriptors are length-normalized, dot product is used as a similarity function.
- The similarity distributions are estimated as $R$-dimensional histograms ($H^+$ and $H^-$) with the nodes $t_1 = -1, t_2, \ldots, t_R = +1$ uniformly filling $[-1;+1]$ with the step $\Delta = \frac{2}{R-1}$.
- The value $h_r^+$ is estimated as:

$$h_r^+ = \frac{1}{\Delta|\mathcal{S}^+|}\sum_{s_{ij}\in[t_{r-1};t_r]}(s_{ij} - t_{r-1}) + \sum_{s_{ij}\in[t_r;t_{r+1}]}(t_{r+1} - s_{ij}) \quad (3)$$

## Derivatives

The Histogram loss $(2)$ is differentiable w.r.t. the pairwise similarities $s \in \mathcal{S}^+$ and $s \in \mathcal{S}^-$.

Loss derivatives w.r.t. histogram entries:

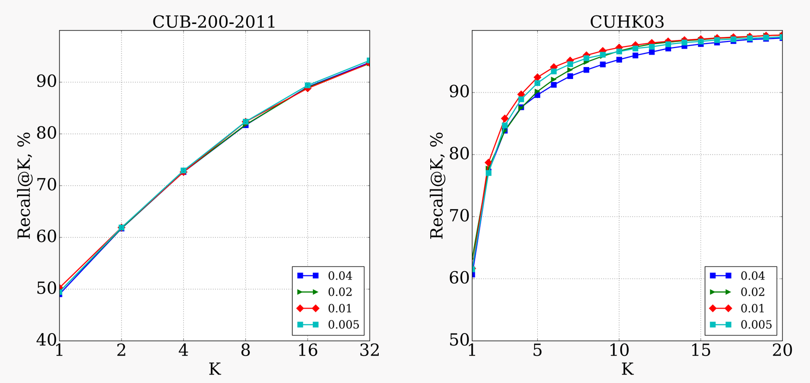
$$\frac{\partial L}{\partial h_r^-} = \sum_{q=1}^{r} h_q^+ \text{ and } \frac{\partial L}{\partial h_r^+} = \sum_{q=r}^{R} h_q^- \quad (4)$$

Histogram derivatives w.r.t. the pairwise similarities:

$$\frac{\partial h_r^+}{\partial s_{ij}} = \begin{cases} \frac{+1}{\Delta|\mathcal{S}^+|}, & \text{if } s_{ij} \in [t_{r-1}; t_r], \\ \frac{-1}{\Delta|\mathcal{S}^+|}, & \text{if } s_{ij} \in [t_r; t_{r+1}], \\ 0, & \text{otherwise}, \end{cases} \quad (5)$$

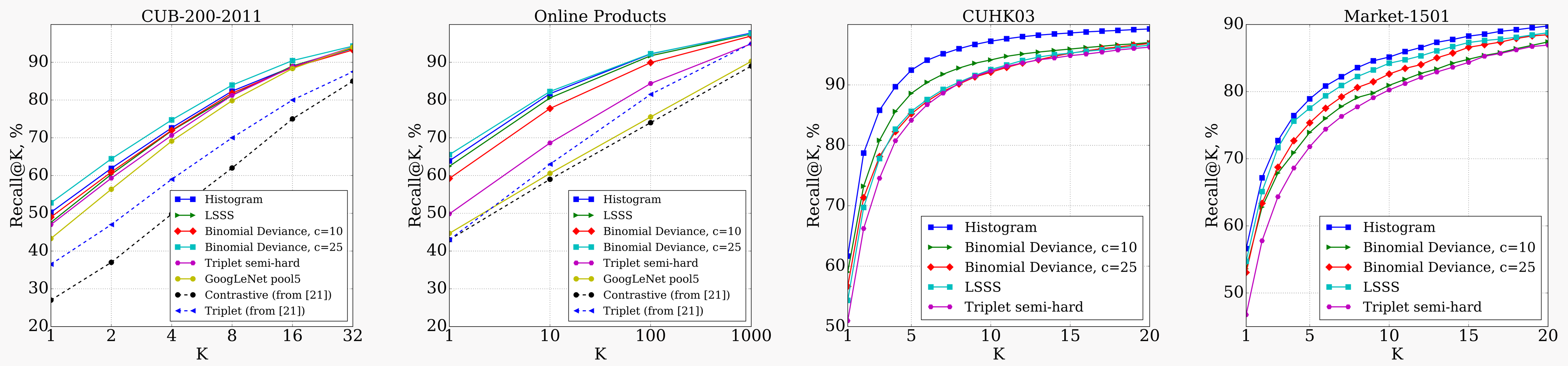
## Parameter choice

- The size of the histogram bin is the only parameter.
- For the CUB-200-2011 dataset bin size does not affect the retrieval quality.
- For CUHK03, increasing the bin size leads to some overfitting.
- 200 or so bins (bin size of 0.01) are sufficient to get good results in most cases we tried.

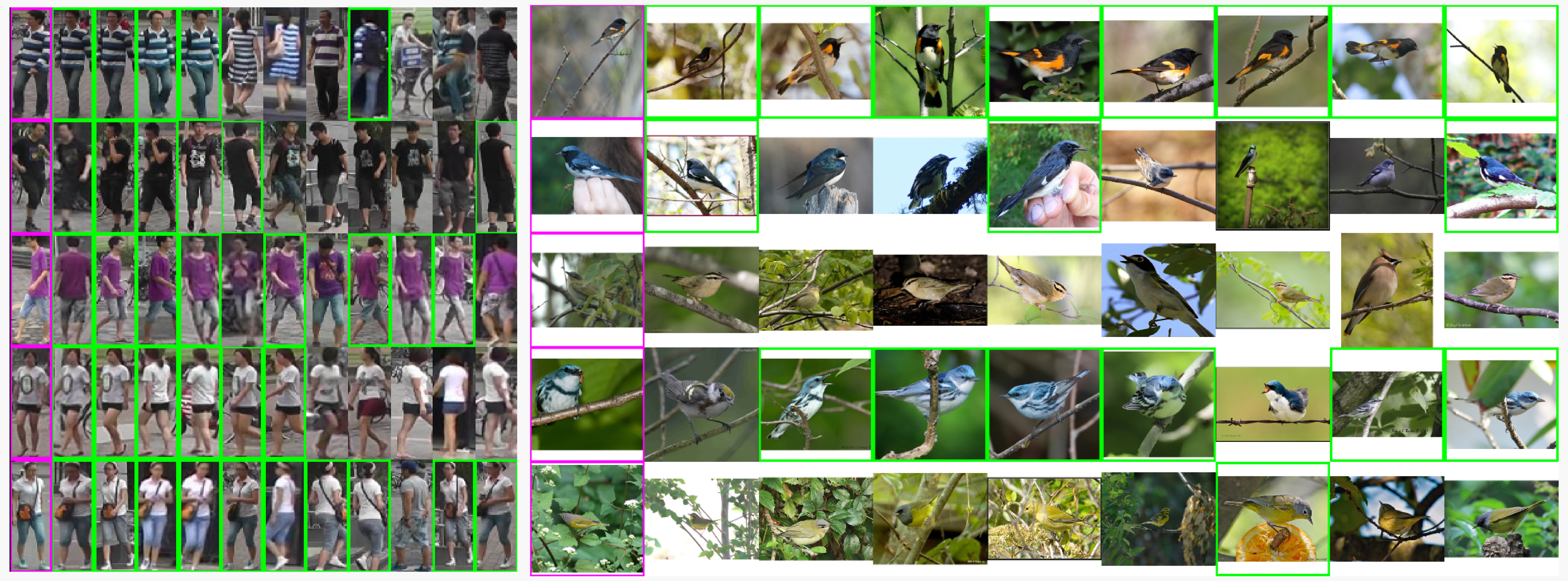



## Retrieval results

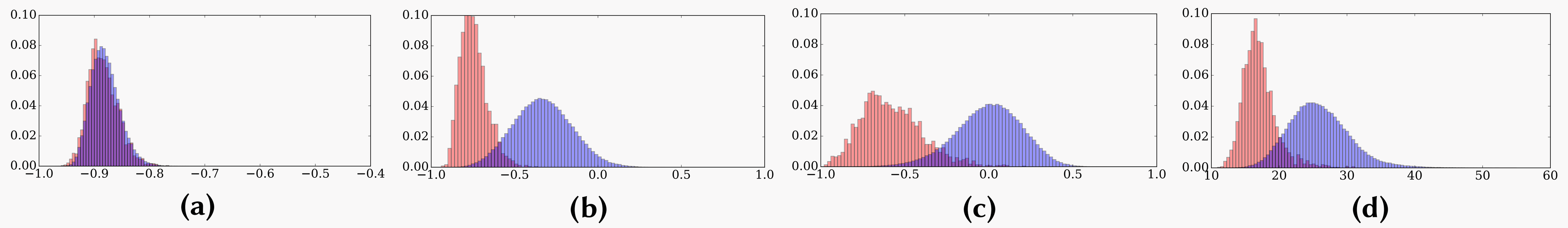
Second best for CUB-200-2011 and Online Products. Outperformed others on CUHK03 and Market-1501.




Examples of output for random queries for Market-1501 and CUB-200-2011.



Histograms for positive and negative distance distributions on the CUHK03 test set for: (a) Initial state: randomly initialized net, Network training with (b) the Histogram loss, (c) the Binomial Deviance loss, (d) the LSSS loss.




## Summary

- The new loss function for learning deep embeddings.
- Makes the distribution of positive and negative pairs less overlapping.
- Implicitly takes into account all the quadruplets in a mini-batch.
- The only tunable parameter is bin size (low sensitivity!).
- Shows competitive results on a number of datasets.

**Caffe code:** `https://github.com/madkn/HistogramLoss`