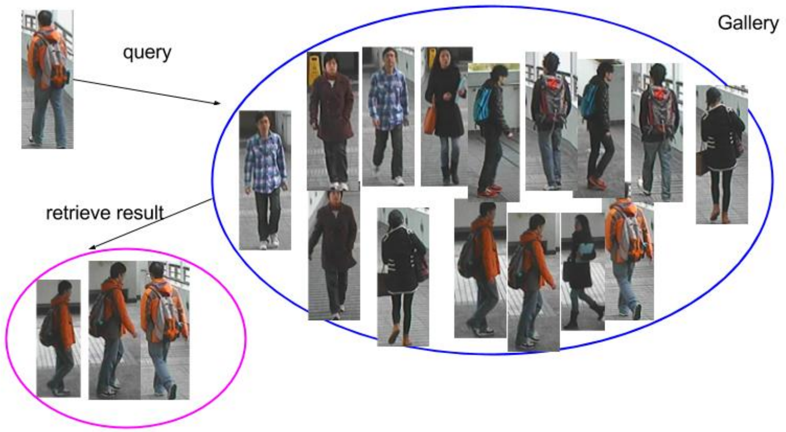
Deep Feature Embedding Learning for Person Re-Identification Using Lifted Structured Loss

Zhangping He, Zhendong Zhang, and **Cheolkon Jung**
School of Electronic Engineering
Xidian University, China

Person Re-Identification

- **Person Re-id**: Given a query, find the **matched pedestrians** across multiple cameras, viewed as an image retrieval problem

- **Challenges**
  - Low resolution video images
  - Viewpoint changes
  - Changes in human body poses
  - Illumination variations
  - Background clutters
  - Occlusions

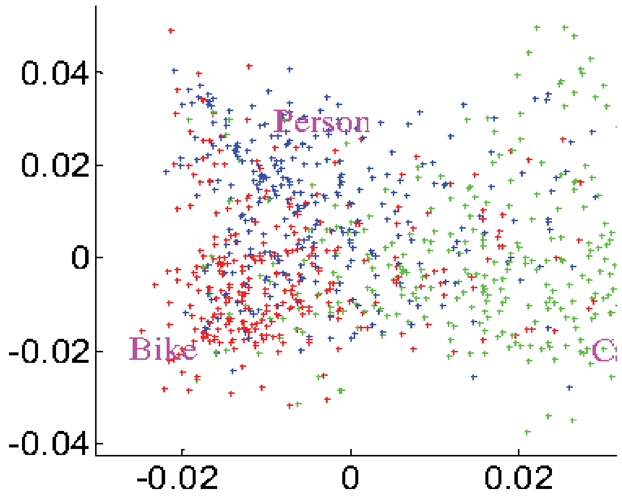


Viewpoint Illumination Occlusion

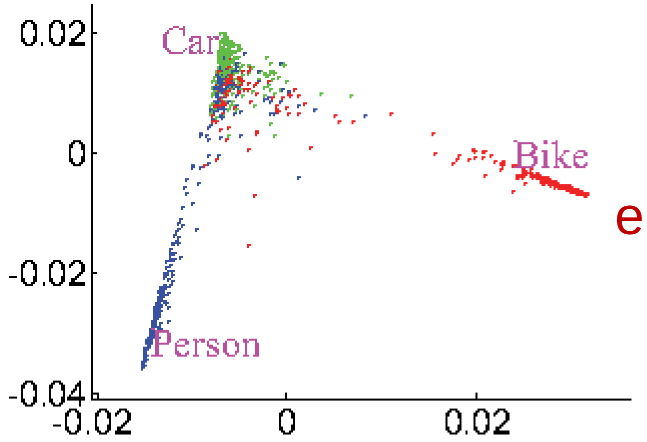
Feature Embedding Learning

- **Feature embedding**:
  - Given feature x, we get $f(x)$.
  - Map similar points to close ones and different points to distant ones, become more discriminative
  - Robust to pose changes
  - Obtained by **deep neural networks**
- **Feature embedding learning**:
  - Metric learning by optimizing loss function

$$D_{i,j}^2 = \left\| f(x_i) - f(x_j) \right\|_2^2$$



Feature



Feature embedding

Contrastive Loss (CVPR 2006)

- **Contrastive loss**:
  - Minimize positive pair distances while penalizing negative pair distances
  - Contrastive embedding is trained on paired data $\{(x_i, x_j, y_{i,j})\}$

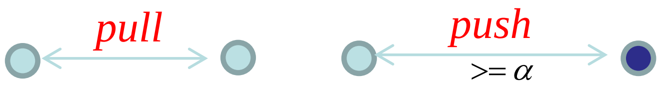
$$L_{Contrastive} = \frac{1}{2m}\sum_{i=1}^{m}\left\{ y_{i,j} D_{i,j}^2 + (1 - y_{i,j})\left[\alpha - D_{i,j}^2\right]_+ \right\}$$

m: Number of images; (x_i, x_j): Pair;

$D_{i,j}^2 = \left\| f(x_i) - f(x_j) \right\|_2^2$: Distance where *f* is feature embedding output;

$y_{i,j} \in \{0,1\}$: Same class or not;

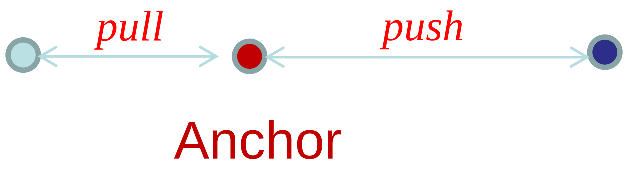
$[\cdot]_+$: Hinge loss;



Triplet Loss (CVPR 2015)

- **Triplet loss**:
  - Introduced for face recognition and clustering;
  - **Less greedy** than contrastive loss due to using Anchor
  - Triplet data $\left\{\left(x_{\mathrm{a}}^{i}, x_{p}^{i}, x_{n}^{i}\right)\right\}$: $\left\{\left(x_{\mathrm{a}}^{i}, x_{p}^{i}\right)\right\}$ -Same class, $\left\{\left(x_{\mathrm{a}}^{i}, x_{n}^{i}\right)\right\}$ - Different class;
  - Loss function:

$$L_{\text{triplet}} = \frac{1}{2m}\sum_{i=1}^{m}\max\{0, D_{ia,ip}^{2} - D_{ia,in}^{2} + \alpha\} \qquad D_{i,j}^{2} = \left\| f(x_i) - f(x_j) \right\|_2^2$$

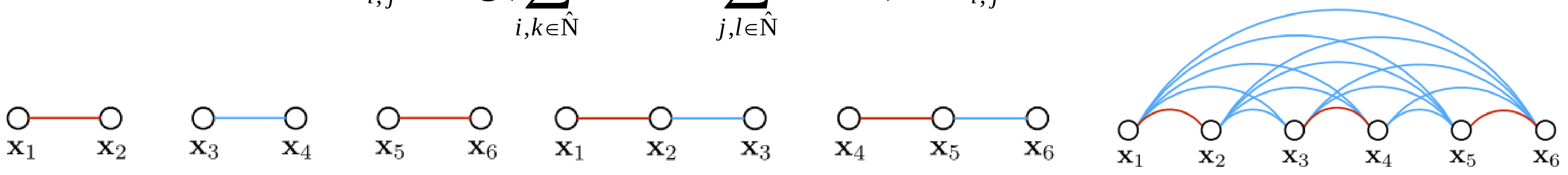


Lifted Structured Loss (CVPR 2016)

- **Lifted Structured loss**:
  - Make a **full use of batch information** based on all positive and negative pairs of samples in the training set, but non-smooth
  - Use a **smooth upper bound** in the loss function

$$\tilde{L}_{\text{lifted}} = \frac{1}{2\left|\hat{P}\right|} \sum_{(i,j)\in\hat{P}} \max(0, \tilde{L}_{i,j})^2$$

$$\tilde{L}_{i,j} = \log\Big(\sum_{i,k\in\hat{\mathrm{N}}} \mathrm{e}^{\alpha - D_{i,\mathrm{k}}} + \sum_{j,l\in\hat{\mathrm{N}}} \mathrm{e}^{\alpha - D_{j,l}}\Big) + D_{i,j}$$



Contrastive Loss | Triplet Loss | Lifted Structured Loss

Proposed Method

- **Proposed Lifted Structured Loss**:
  - The number of negative samples is varying (not equal) compared with positive pairs, and thus the number of the summation term is uncertain.
  - **Imbalance** between the log term and $D_{i,j}$
  - We use the mean of log term so that $L_{i,j}$ is robust to the difference between positive and negative pairs.
  - Also, we use $D^2_{i,j}$ for **fast convergence** instead of $D_{i,j}$ ($L_{i,j}$: [$\alpha-4,\alpha$])

Proposed Method

- **Proposed Lifted Structured Loss:**

$$L_{i,j} = \log\left(\frac{1}{|\hat{\mathrm{T}}_{i,j}|} \left(\sum_{(i,\mathrm{k}) \in N} \mathrm{e}^{\alpha - D_{i,\mathrm{k}}^2} + \sum_{(j,l) \in N} \mathrm{e}^{\alpha - D_{j,l}^2} \right) \right) + D_{i,j}^2$$

$$L_{struct} = \frac{1}{2|\hat{P}|} \sum_{(i,j) \in \hat{P}} \max(0, L_{i,j})$$

- **Combination with Identification Loss:**

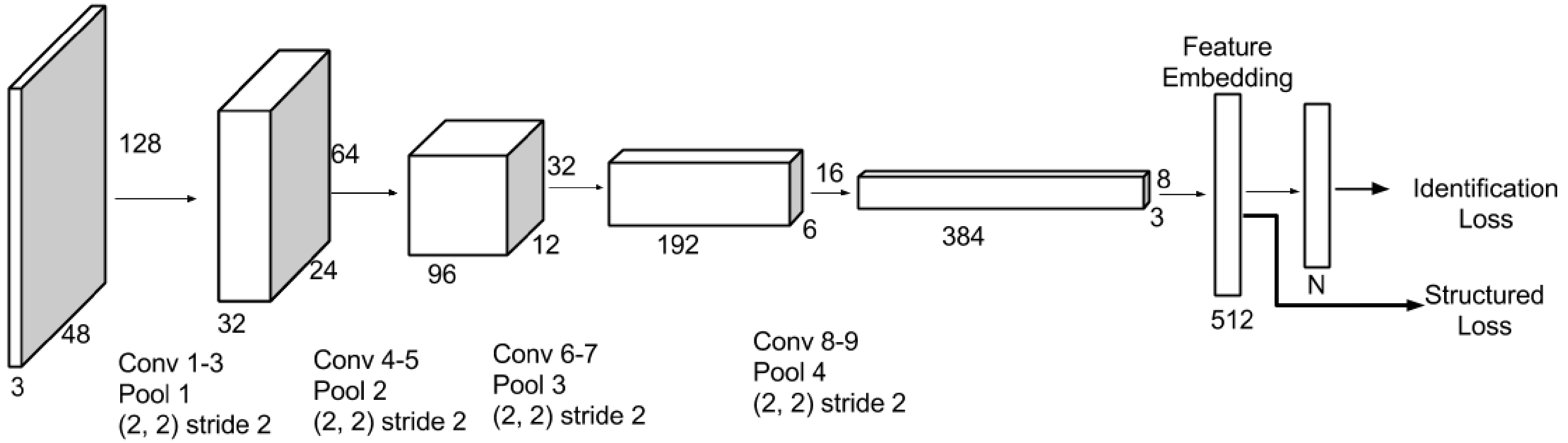
$$q_i = \mathrm{softmax}(W_i^T f(x))$$

$$L_{id} = \sum_i -p_i \log q_i$$

$$L = L_{struct} + \lambda L_{id}$$

Proposed Method

- **Network Architecture:**



- 9 Convolutional layers: 3×3 filters with stride 1 and zero paddings.
  - Dimensions from Conv1 to Conv9: 32, 32, 32, 64, 96, 128, 192, 256, 384.
- 4 Max pooling layers: 2×2 filters with stride 2.
- Batch normalization after each convolutional layer or FC layer to speed up the training.
- Leaky rectified linear unit (LReLU) is used after these layers.

Experimental Results

- **Datasets:**
  - CUHK01, CUHK03 and VIPeR
- **Data Preparation:**
  - We resize all training images to **128×48**.
  - We sample 3 images around an image center with small translation and augment the data with images reflected on a vertical mirror: Total **5 images** from one image.
- **Evaluation Protocol:**
  - **Cumulative match curve (CMC)** metric

Experimental Results

- **Parameter Setting:**
  - $\lambda = 1.0$;
  - SGD: Initial learning rate 0.001, decayed by 0.1 after 20,000 iterations.
  - α in structured loss is 3.0, while α in contrastive and triplet losses is 1.0. Batch size: 64, iteration number: 30,000.

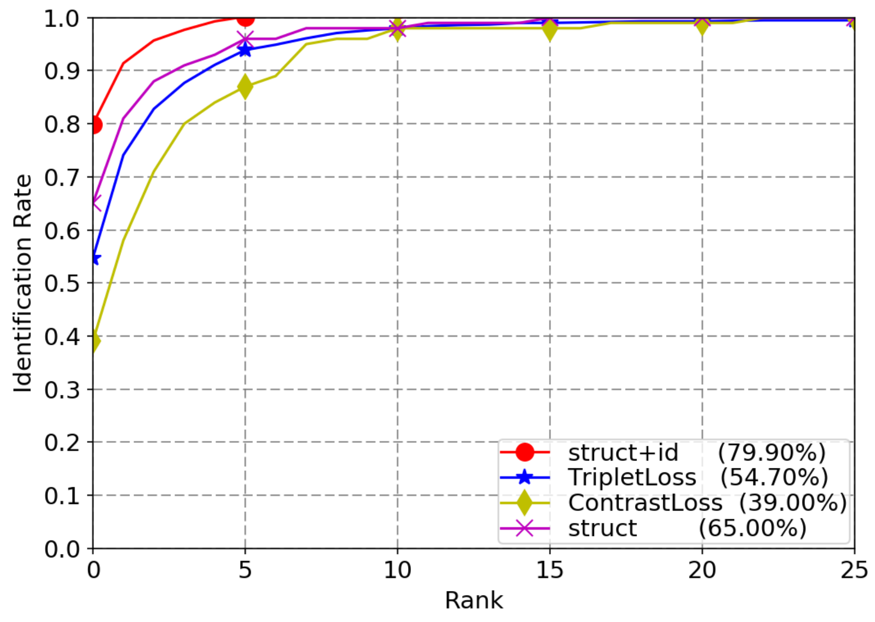
- **Two sets of experiments:**
  - Evaluation of the proposed loss with contrastive loss and triplet loss
  - Performance comparison with state-of-art person re-id methods.

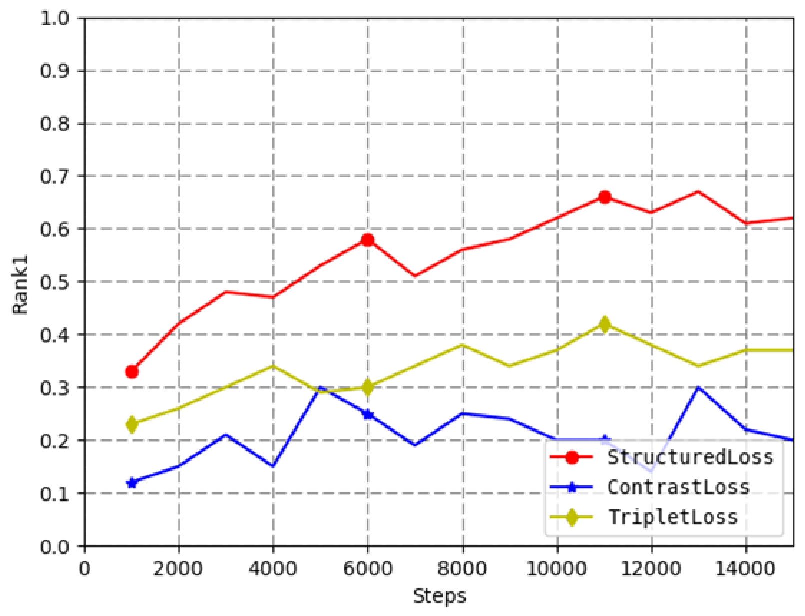
Experimental Results

Red box: Identified person

Experimental Results

- **Loss Function Comparison:**
  - Experiments under the same CNN architecture with different loss functions (Steps for training)





Experimental Results

- **Experiments on CUHK03 (Labeled, Detected)**

Table 1. Accuracy Comparison on CUHK03 (Labeled)

methods	rank1	rank5	rank10
kLFDA[18]	48.20	59.34	66.38
IDLA[5]	54.74	86.50	94.00
NullRe-id[19]	58.90	85.60	92.45
Ensembles[20]	62.10	89.10	94.30
Gated Siamese[3]	68.10	88.10	94.60
NX-Corr M[21]	72.43	95.51	98.40
Proposed	**81.9**	**96.7**	**98.7**

Table 3. Accuracy Comparison on CUHK03 (Detected)

methods	rank1	rank5	rank10
IDLA[5]	45.0	76.0	83.5
NullRe-id[19]	53.70	83.05	93.00
Siamese LSTM[24]	57.3	80.1	88.3
Joint Learning[25]	52.17	85.00	92.00
Gated Siamese[3]	61.8	80.9	88.3
NX-Corr M[21]	72.04	96.00	98.26
Improved Embedding [4]	**82.1**	96.2	98.2
Proposed	79.9	**97.1**	**98.7**

Experimental Results

- **Experiments on CUHK01 and VIPeR**

Small size dataset

Table 2. Accuracy Comparison on CUHK01

methods	rank1	rank5	rank10
IDLA[5]	47.5	71.6	80.3
NullRe-id[19]	69.1	86.9	91.8
MCP-CNN[7]	53.7	84.3	91.0
NX-Corr M[21]	65.04	89.76	94.4
Proposed	**70.2**	**90.2**	**95.5**

Table 4. Accuracy Comparison on VIPeR

methods	rank1	rank5	rank10
Joint Learning[25]	35.8	-	-
Gated Siamese[3]	37.8	66.9	77.4
Siamese LSTM[24]	42.4	68.7	79.4
Ensembles[20]	45.9	77.5	88.9
MCP-CNN[7]	47.8	74.7	84.8
SCSP[23]	**53.5**	82.6	91.5
NullRe-id[19]	51.2	82.1	90.5
Improved Embedding[4]	50.4	77.6	85.8
LSSCDL[22]	42.7	**84.3**	**91.9**
Proposed	47.3	76.6	88.1

Conclusions

- **Deep feature embedding learning** for person re-id based on **lifted structured loss**.
  - The proposed person re-id is based on **CNN**, and combines lifted structured loss and identification loss into loss function.
  1) Feature embedding on test images using CNN, i.e deep feature embedding learning
  2) Normalization of embedding into a unit vector
  3) Computing the distance between all pairs from two camera views

- Experimental results
  - Proposed method outperforms state-of-the-arts on CUHK01 and CUHK03
  - A little worse on VIPeR, i.e. **small size dataset**

THANK YOU!