Tsinghua University

Deep Metric Learning for Image and Video Understanding

Tutor: Jiwen Lu

Department of Automation, Tsinghua University, China

http://ivg.au.tsinghua.edu.cn/Jiwen_Lu/

Outline

- Part 1: Introduction
- Part 2: Mahalanobis Deep Metric Learning
- Part 3: Hamming Deep Metric Learning
- Part 4: Sampling for Deep Metric Learning
- Part 5: Conclusion and Future Directions

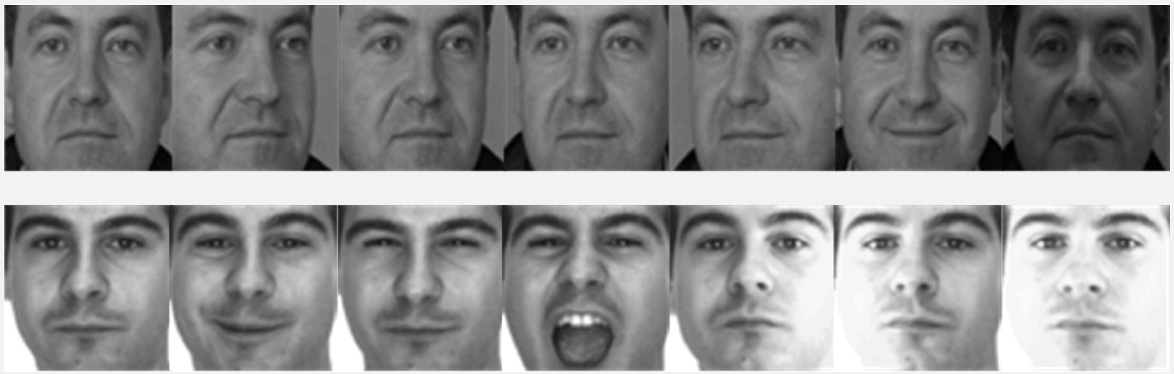
Part 1: Introduction

Why Measuring Similarity Between Objects

- Similarity: computing distances between data points.
- Performance: depending on the definitions of similarity.

Image and Video Understanding

- Face identification



- Face verification

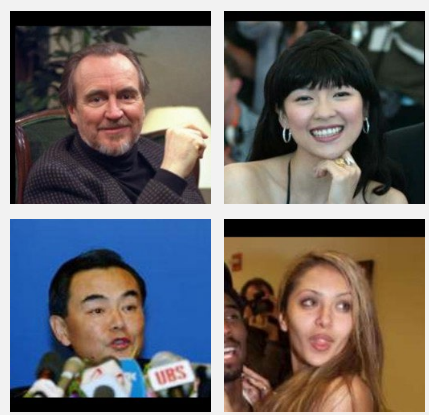


Image and Video Understanding

- Kinship verification (social media analysis)

Image and Video Understanding

- RGB-D Object Recognition (robotics)

Image and Video Understanding

- Image Classification (visual object recognition)

Image and Video Understanding

- Person Re-identification (visual surveillance)

Image and Video Understanding

- Visual Searching (multimedia technology)

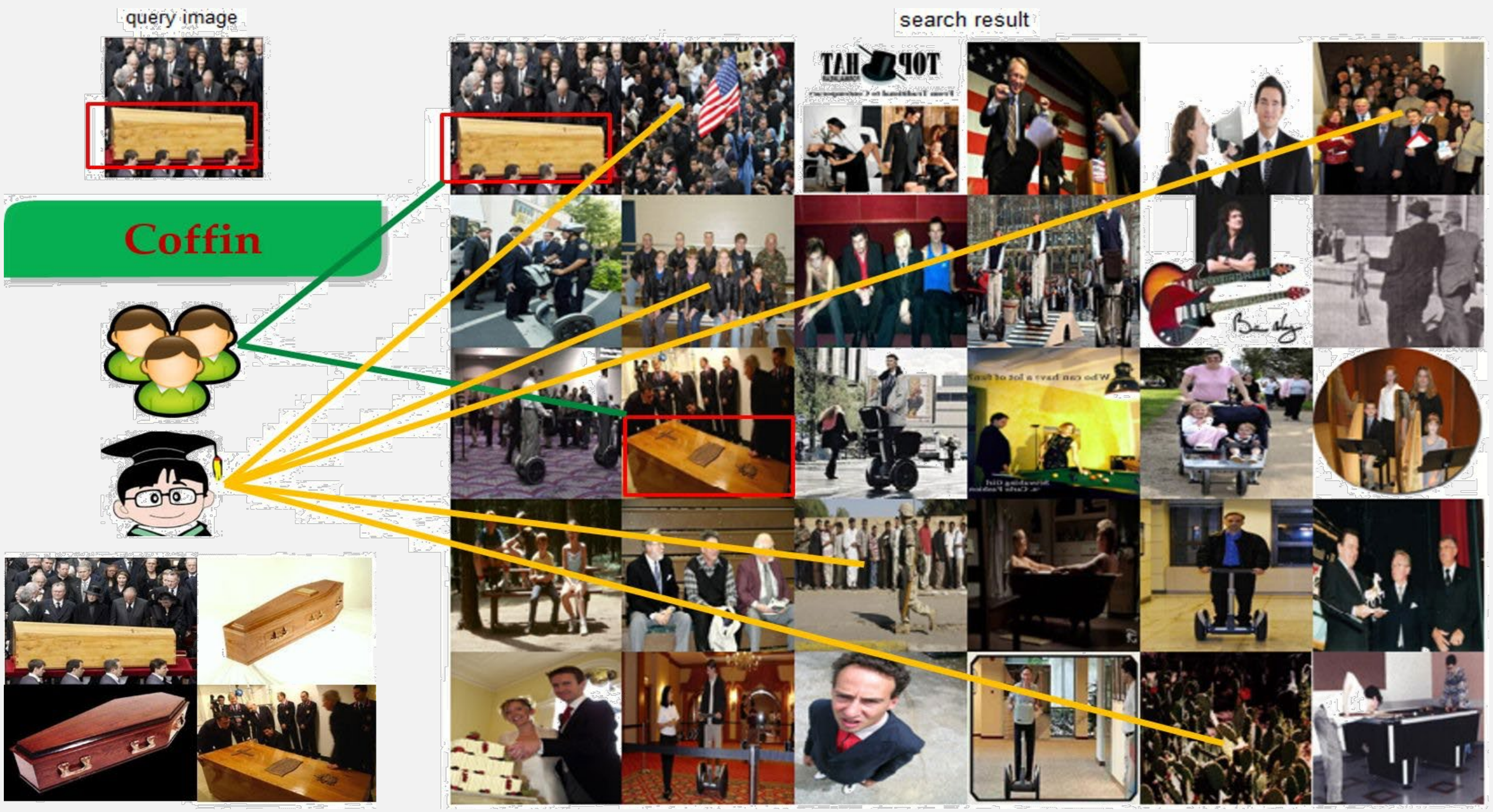


Image and Video Understanding

- Visual Tracking (visual surveillance)

Image and Video Understanding

- Activity Recognition(visual surveillance)

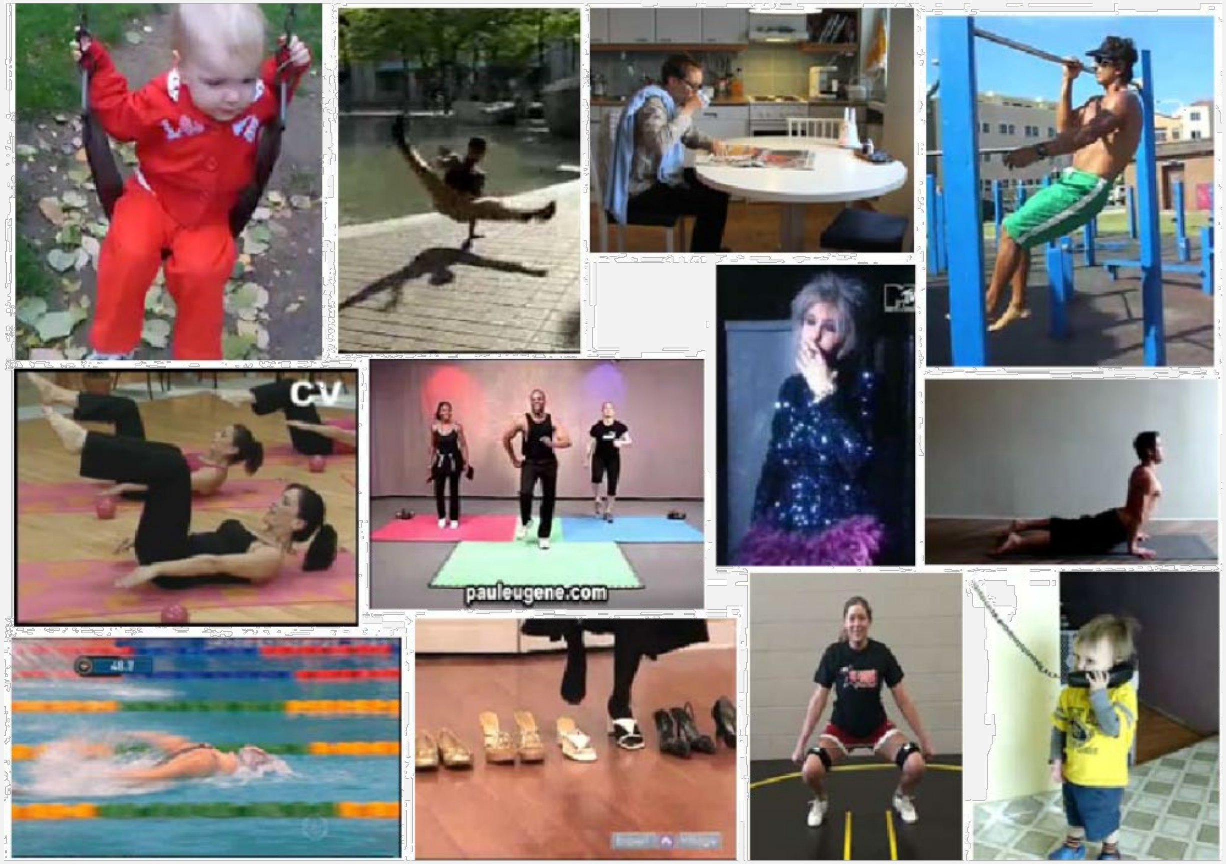
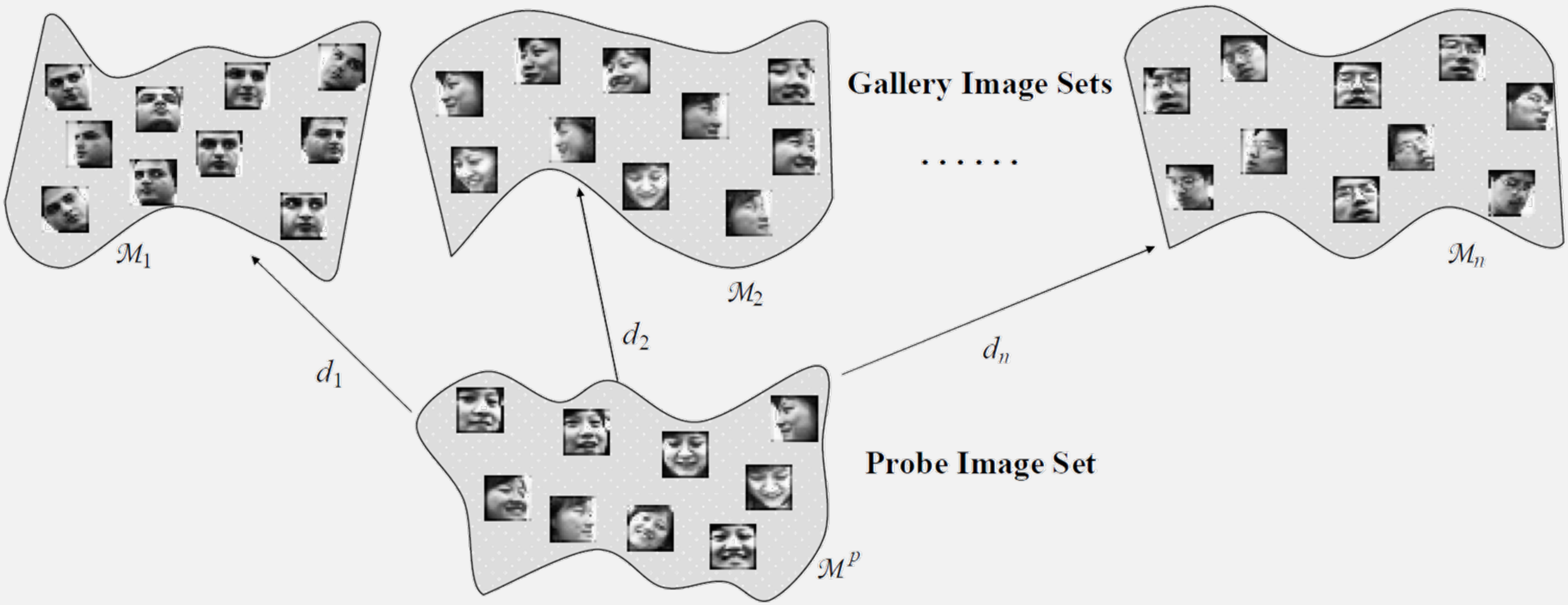


Image and Video Understanding

Image Set Classification



How to Measure Similarity: Metric

- A metric is a function that defines a distance between each pair of elements of a set.
- Formally, it is a mapping $d: \chi \times \chi \rightarrow \mathbb{R}_+$, which satisfies the following properties for all $x, y, z \in \chi$

1. $d(x, y) \geq 0$	**Non-negativity**
2. $d(x, y) = d(y, x)$	**Symmetry**
3. $d(x, z) \leq d(x, y) + d(y, z)$	**Triangle inequality**
4. $d(x, y) = 0 \Leftrightarrow x = y$	**Identity of indiscernibles**

If condition 4 is not met, we are referring to a pseudo-metric. Usually we do not distinguish between metrics and pseudo-metrics.

Learning a Metric Subspace

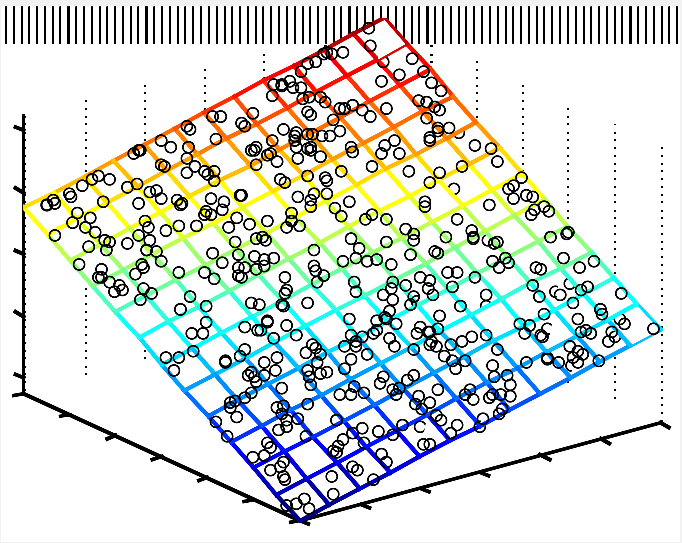
- Given a dataset $X = [x_1, x_2, \cdots x_N]$, metric learning aims to seek a low-dimensional subspace W to map each x_i to y_i, where $y_i = Wx_i$, such that some characteristics are preserved.
- This metric computes the squared distances as

$$d(x_i, x_j) = \left\| y_i - y_j \right\|_2^2 = \left\| Wx_i - Wx_j \right\|_2^2$$

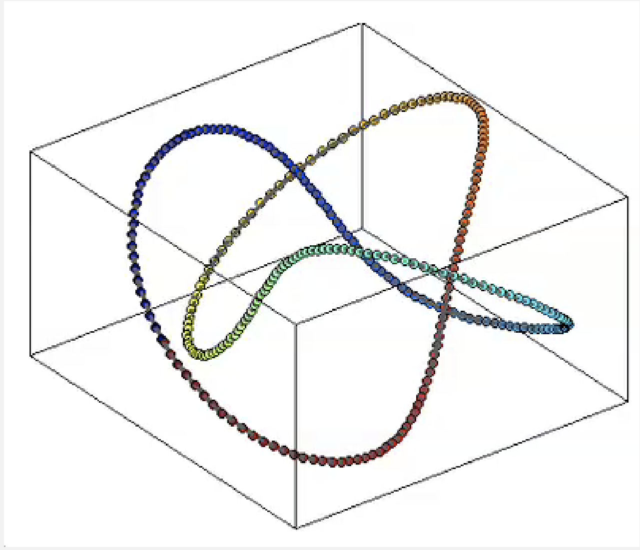
- It is easy to see that by setting W equal to the identity matrix, we fall back to common Euclidean distance.

Another View: Subspace Learning

- Eliminate redundant features
- Eliminate irrelevant features
- Extract low dimensional structure



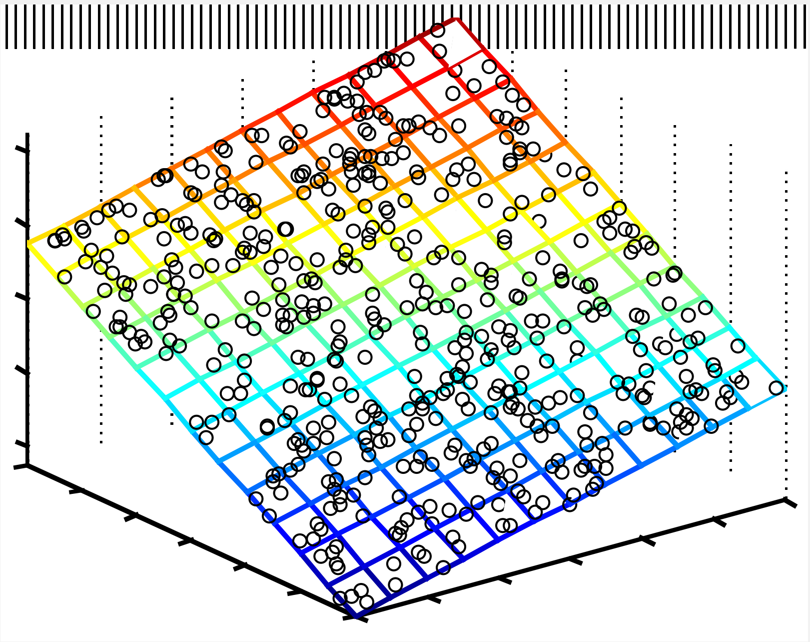
Linear

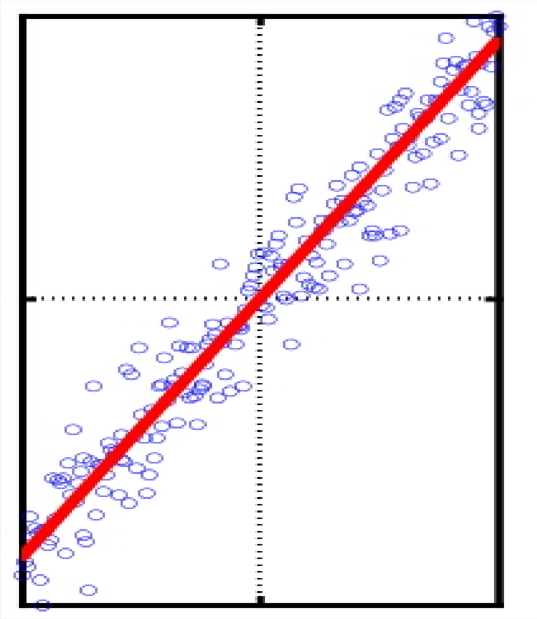


Non-Linear

Another View: Subspace Learning

- PCA





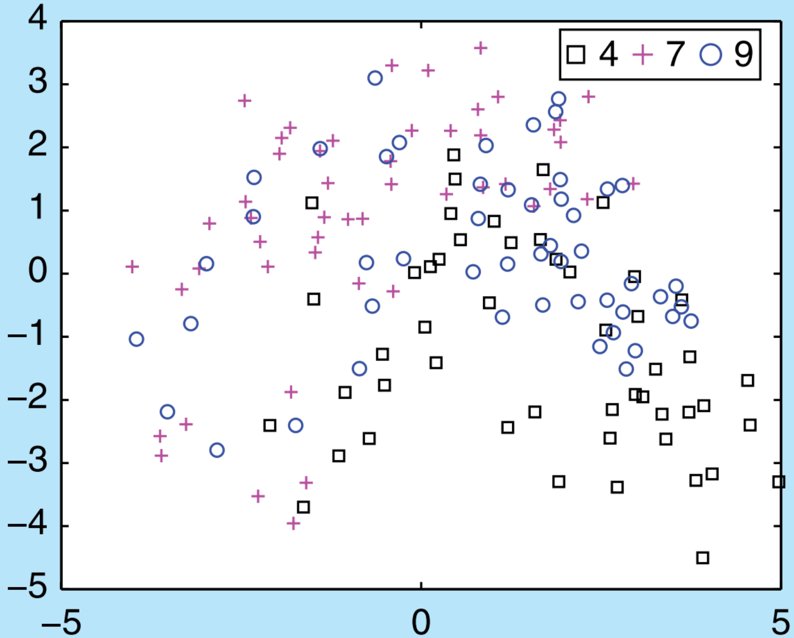
Project data into subspace of maximum variance.

Representation Subspace Learning Algorithms

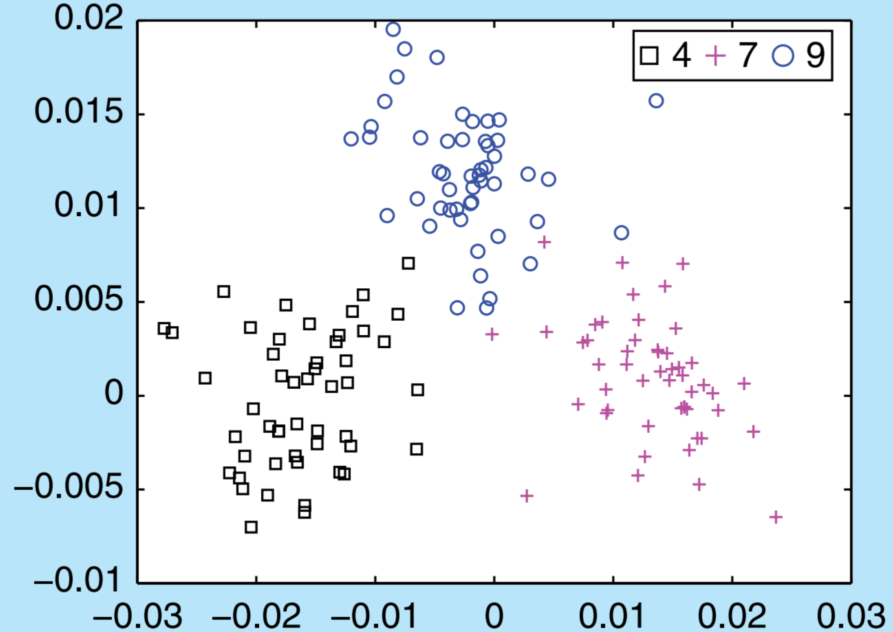
- PCA (principal component analysis) (CVPR, 1991)
- LDA (linear discriminant analysis) (PAMI, 1997)
- NMF (nonnegative matrix factorization) (Nature, 1999)
- LPP (locality preserving projections) (NIPS, 2003)
- NPE (neighborhood preserving embedding) (ICCV, 2005)
- MFA (margin fisher analysis) (CVPR, 2005)
- LDE (local discriminant embedding) (CVPR, 2005)
- DLPP (discriminant LPP) (IVC, 2006)
- SR (spectral regression) (ICCV, 2007)
- DSA (discriminant simplex analysis) (TIFS, 2008)
- CEA (conform embedding analysis) (TMM, 2008)
- SPP (sparsity preserving projections) (PR, 2010)

How Metric Learning Works

- An example on the MNIST data: PCA vs LDA



(a) PCA

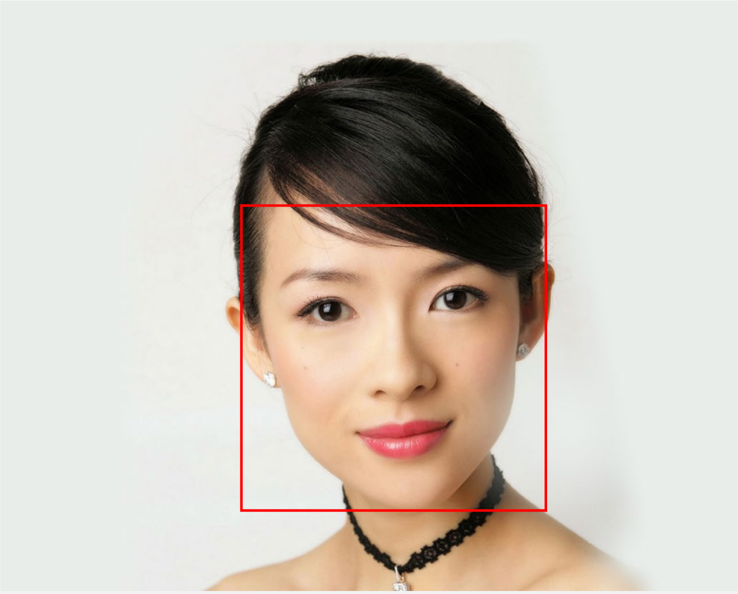


(b) LDA

[Lu et al, SPM 2017]

Challenges

High-dimensional data

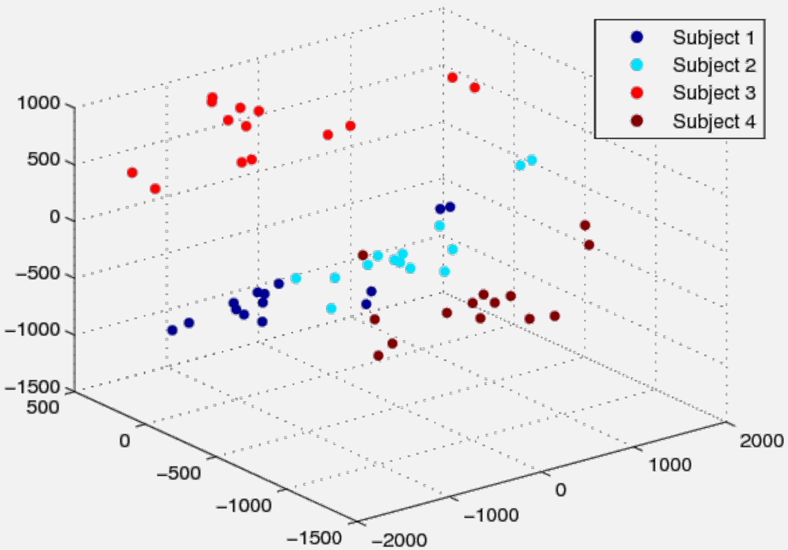


Feature vector

- Deteriorate the performances of classifiers
- High computational complexity

Challenges

Nonlinear metric space



- Large intra-class variance.
- Kernel trick encounters scalability problem.

Solutions

- Robust, compact and informative descriptors.
  - Hand-crafted
  - Learning-based
- Efficient, discriminative and scalable models.
  - Deep representation
  - Metric learning

Part 2: Mahalanobis Deep Metric Learning

Mahalanobis Distance

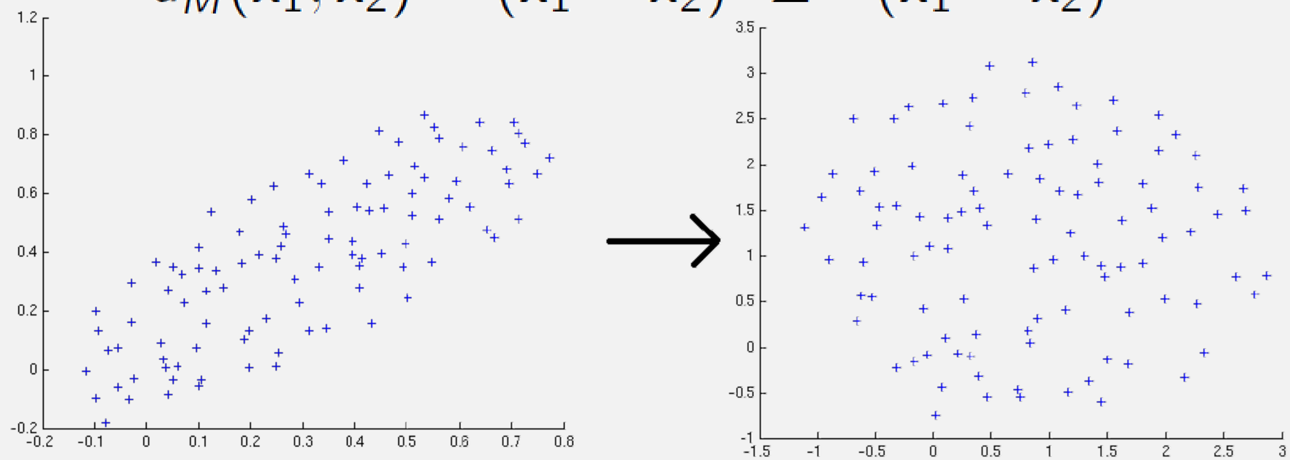
- Squared Euclidean Distance （regression problem）

$$d(\mathbf{x}_1, \mathbf{x}_2) = \|\mathbf{x}_1 - \mathbf{x}_2\|_2^2 = (\mathbf{x}_1 - \mathbf{x}_2)^T(\mathbf{x}_1 - \mathbf{x}_2)$$

Let $\Sigma = \sum_{i,j}(\mathbf{x}_i - \mu)(\mathbf{x}_j - \mu)^T$

- The Mahalanobis distance

$$d_M(\mathbf{x}_1, \mathbf{x}_2) = (\mathbf{x}_1 - \mathbf{x}_2)^T \Sigma^{-1} (\mathbf{x}_1 - \mathbf{x}_2)$$



Metric Learning

- Applying Mahalanobis distance to learn a positive semi-definite (PSD) matrix

$$d_{\mathbf{M}}(\mathbf{x}_i, \mathbf{x}_j) = \sqrt{(\mathbf{x}_i - \mathbf{x}_j)^T \mathbf{M} (\mathbf{x}_i - \mathbf{x}_j)}$$

- Relationship with subspace learning

$$\begin{aligned} d_{\mathbf{M}}(\mathbf{x}_i, \mathbf{x}_j) &= \sqrt{(\mathbf{x}_i - \mathbf{x}_j)^T \mathbf{M} (\mathbf{x}_i - \mathbf{x}_j)} \\ &= \sqrt{(\mathbf{x}_i - \mathbf{x}_j)^T \mathbf{W}^T \mathbf{W} (\mathbf{x}_i - \mathbf{x}_j)} \\ &= \|\mathbf{W}\mathbf{x}_i - \mathbf{W}\mathbf{x}_j\|_2 \end{aligned}$$

where $\mathbf{M} = \mathbf{W}^T \mathbf{W}$

Representative Metric Learning Algorithms

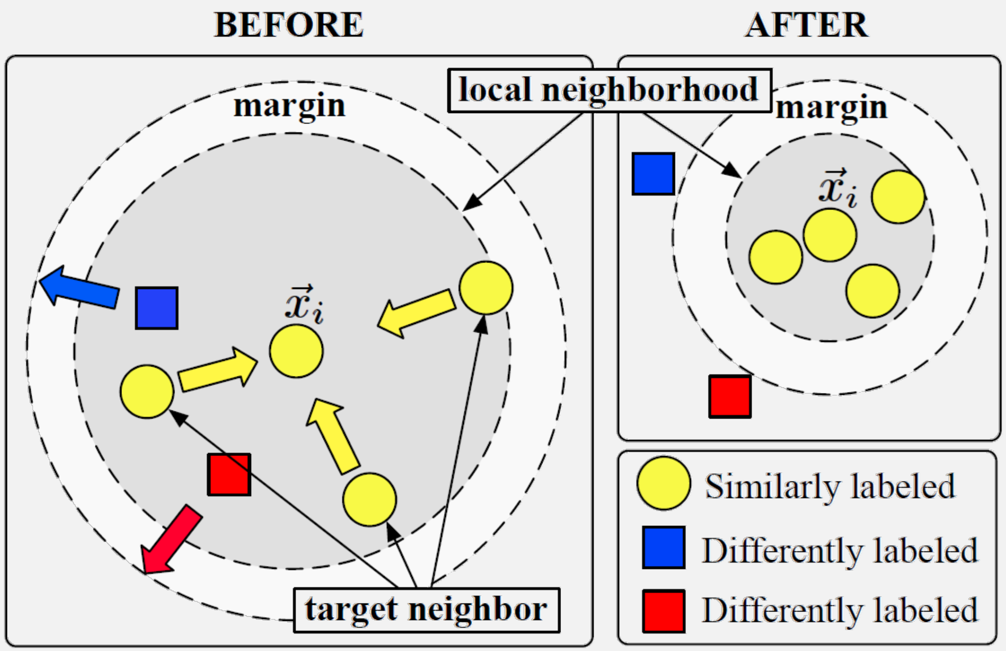
- Large Margin Nearest Neighborhood (LMNN)

Minimize $\sum_{ij} \eta_{ij} (\vec{x}_i - \vec{x}_j)^\top \mathbf{M} (\vec{x}_i - \vec{x}_j) + c \sum_{ij} \eta_{ij} (1 - y_{il}) \xi_{ijl}$ **subject to:**

(1) $(\vec{x}_i - \vec{x}_l)^\top \mathbf{M} (\vec{x}_i - \vec{x}_l) - (\vec{x}_i - \vec{x}_j)^\top \mathbf{M} (\vec{x}_i - \vec{x}_j) \geq 1 - \xi_{ijl}$

(2) $\xi_{ijl} \geq 0$

(3) $\mathbf{M} \succeq 0$.



[Weinberger et al, NIPS 2005]

Representative Metric Learning Algorithms

- Information-Theoretic Metric Learning (ITML)

$$\begin{aligned} \min_{A} \quad & \mathrm{KL}(p(\boldsymbol{x}; A_0) \| p(\boldsymbol{x}; A)) \\ \text{subject to} \quad & d_A(\boldsymbol{x}_i, \boldsymbol{x}_j) \le u \qquad (i,j) \in S, \\ & d_A(\boldsymbol{x}_i, \boldsymbol{x}_j) \ge \ell \qquad (i,j) \in D. \end{aligned}$$

where $\mathrm{KL}(p(\boldsymbol{x}; A_o) \| p(\boldsymbol{x}; A)) = \int p(\boldsymbol{x}; A_0) \log \frac{p(\boldsymbol{x}; A_0)}{p(\boldsymbol{x}; A)} \, d\boldsymbol{x}$

- The optimization function can be re-formulated as

$$\begin{aligned} \min_{A \succeq 0} \quad & D_{\ell\mathrm{d}}(A, A_0) \\ \text{s.t.} \quad & \mathrm{tr}(A(\boldsymbol{x}_i - \boldsymbol{x}_j)(\boldsymbol{x}_i - \boldsymbol{x}_j)^T) \le u \qquad (i,j) \in S, \\ & \mathrm{tr}(A(\boldsymbol{x}_i - \boldsymbol{x}_j)(\boldsymbol{x}_i - \boldsymbol{x}_j)^T) \ge \ell \qquad (i,j) \in D, \end{aligned}$$

[Davis et al, ICML 2007]

Categorization

- The structure of the input
  - Linear
  - Kernel
  - Tensor
- The label type of training samples
  - Supervised
  - Unsupervised
  - Semi-supervised
- The architecture of models
  - Shallow models
  - Deep learning

Categorization

- The supervision type of training samples
  - Weakly-supervised
  - Strongly-supervised
- The number of metrics
  - Single-metric Learning
  - Multi-metric Learning
- The type of distances
  - Mahalanobis-distance metric learning
  - Hamming-distance metric learning

2.1 Discriminative Deep Metric Learning

[1] **Jiwen Lu**, Junlin Hu, and Jie Zhou, Deep metric learning for visual understanding: an overview of recent advances, **IEEE Signal Processing Magazine**, 2017.
[2] Junlin Hu, **Jiwen Lu***, and Yap-Peng Tan, Discriminative deep metric learning for face verification in the wild, **CVPR**, 2014.
[3] **Jiwen Lu**, Junlin Hu, and Yap-Peng Tan, Discriminative deep metric learning for face and kinship verification, **TIP**, 2017.
[4] Junlin Hu, **Jiwen Lu***, and Yap-Peng Tan, Deep metric learning for visual tracking, **TCSVT**, 2016.
[5] **Jiwen Lu**, Gang Wang, Weihong Deng, Pierre Moulin, and Jie Zhou, Multi-manifold deep metric learning for image set classification, **CVPR**, 2015.

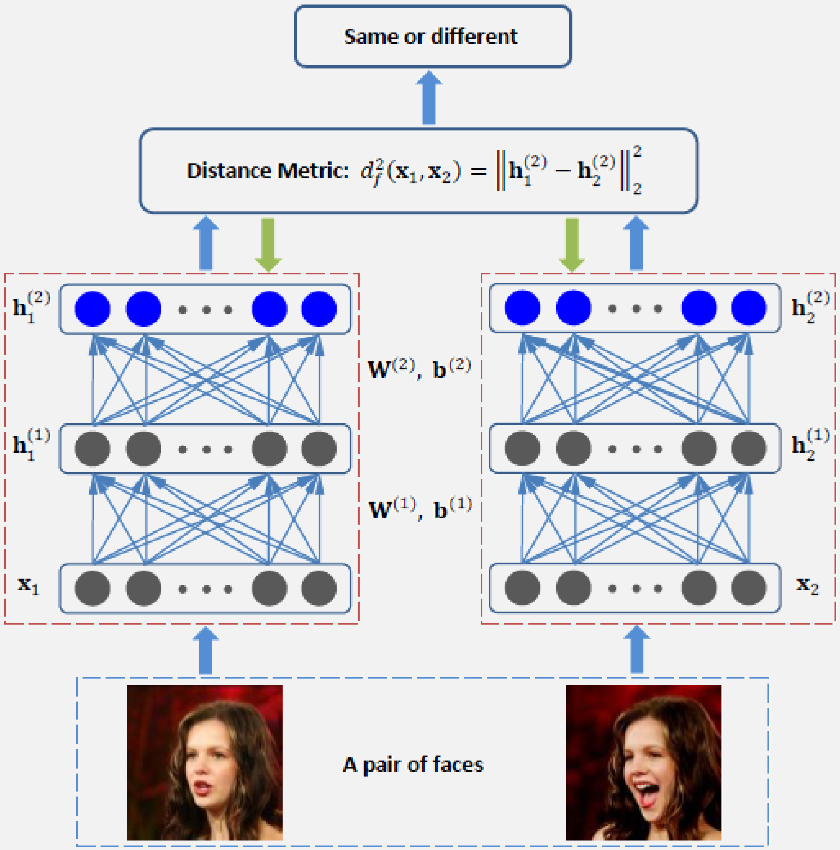
Discriminative Deep Metric Learning

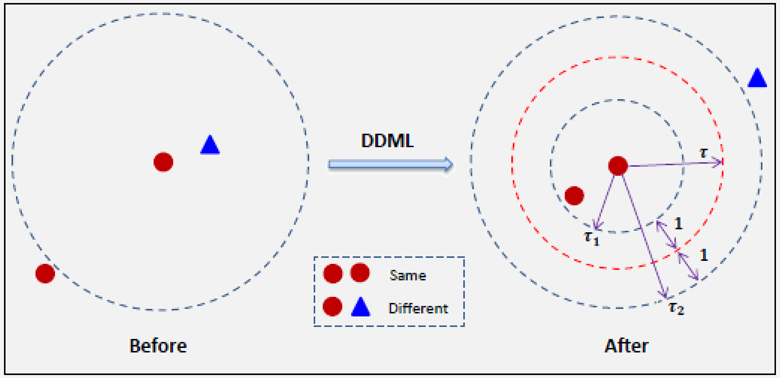
$$\begin{aligned} d_{\mathbf{M}}(\mathbf{x}_i, \mathbf{x}_j) &= \sqrt{(\mathbf{x}_i - \mathbf{x}_j)^T \mathbf{M} (\mathbf{x}_i - \mathbf{x}_j)} \\ &= \sqrt{(\mathbf{x}_i - \mathbf{x}_j)^T \mathbf{W}^T \mathbf{W} (\mathbf{x}_i - \mathbf{x}_j)} \\ &= \|\mathbf{W}\mathbf{x}_i - \mathbf{W}\mathbf{x}_j\|_2 \end{aligned}$$

Motivation

- Conventional metric learning methods only seek a linear mapping, which cannot capture the nonlinear manifold where face images usually lie on.
- The kernel trick can be employed to implicitly map face samples into a high-dimensional feature space and then learn a distance metric in the high-dimensional space. However, these methods cannot explicitly obtain the nonlinear mapping functions, which usually suffer from the scalability problem.

Discriminative Deep Metric Learning





$$\ell_{ij}\left(\tau - d_f^2(\mathbf{x}_i, \mathbf{x}_j)\right) > 1.$$

$$\begin{aligned} \arg\min_f \; J &= J_1 + J_2 \\ &= \frac{1}{2} \sum_{i,j} g\Big(1 - \ell_{ij}\big(\tau - d_f^2(\mathbf{x}_i, \mathbf{x}_j)\big)\Big) \\ &+ \frac{\lambda}{2} \sum_{m=1}^{M} \left(\|\mathbf{W}^{(m)}\|_F^2 + \|\mathbf{b}^{(m)}\|_2^2 \right) \end{aligned}$$

$$f(\mathbf{x}) = \mathbf{h}^{(M)} = s\left(\mathbf{W}^{(M)}\mathbf{h}^{(M-1)} + \mathbf{b}^{(M)}\right) \in \mathbb{R}^{p^{(M)}}$$

Discriminative Deep Metric Learning

$$
\begin{aligned}
\frac{\partial J}{\partial \mathbf{W}^{(m)}} &= \sum_{i,j} \left(\Delta_{ij}^{(m)} \mathbf{h}_i^{(m-1)^T} + \Delta_{ji}^{(m)} \mathbf{h}_j^{(m-1)^T} \right) \\
&+ \lambda\, \mathbf{W}^{(m)} \\
\frac{\partial J}{\partial \mathbf{b}^{(m)}} &= \sum_{i,j} \left(\Delta_{ij}^{(m)} + \Delta_{ji}^{(m)} \right) + \lambda\, \mathbf{b}^{(m)}
\end{aligned}
$$

where

$$
\begin{aligned}
\Delta_{ij}^{(M)} &= g'(c)\ell_{ij} \left(\mathbf{h}_i^{(M)} - \mathbf{h}_j^{(M)} \right) \odot s'\left(\mathbf{z}_i^{(M)} \right) \\
\Delta_{ji}^{(M)} &= g'(c)\ell_{ij} \left(\mathbf{h}_j^{(M)} - \mathbf{h}_i^{(M)} \right) \odot s'\left(\mathbf{z}_j^{(M)} \right) \\
\Delta_{ij}^{(m)} &= \left(\mathbf{W}^{(m+1)^T} \Delta_{ij}^{(m+1)} \right) \odot s'\left(\mathbf{z}_i^{(m)} \right) \\
\Delta_{ji}^{(m)} &= \left(\mathbf{W}^{(m+1)^T} \Delta_{ji}^{(m+1)} \right) \odot s'\left(\mathbf{z}_j^{(m)} \right)
\end{aligned}
$$

$$
\begin{aligned}
c &\triangleq 1 - \ell_{ij}\left(\tau - d_f^2(\mathbf{x}_i, \mathbf{x}_j)\right) \\
\mathbf{z}_i^{(m)} &\triangleq \mathbf{W}^{(m)} \mathbf{h}_i^{(m-1)} + \mathbf{b}^{(m)}
\end{aligned}
$$

Discriminative Deep Metric Learning

$$\mathbf{W}^{(m)} = \mathbf{W}^{(m)} - \mu \frac{\partial J}{\partial \mathbf{W}^{(m)}}$$

$$\mathbf{b}^{(m)} = \mathbf{b}^{(m)} - \mu \frac{\partial J}{\partial \mathbf{b}^{(m)}}$$

Activation function:

$$s(z) = \tanh(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$$

$$s'(z) = \tanh'(z) = 1 - \tanh^2(z)$$

Initialization:

$$\mathbf{W}^{(m)} \sim U\left[-\frac{\sqrt{6}}{\sqrt{p^{(m)} + p^{(m-1)}}}, \frac{\sqrt{6}}{\sqrt{p^{(m)} + p^{(m-1)}}}\right]$$

Algorithm 1: DDML

Input: Training set: $\mathbf{X} = \{(\mathbf{x}_i, \mathbf{x}_j, \ell_{ij})\}$, number of network layers $M+1$, threshold τ, learning rate μ, iterative number I_t, parameter λ, and convergence error ε.

Output: Weights and biases: $\{\mathbf{W}^{(m)}, \mathbf{b}^{(m)}\}_{m=1}^{M}$.

// Initialization:

Initialize $\{\mathbf{W}^{(m)}, \mathbf{b}^{(m)}\}_{m=1}^{M}$ according to Eq. (20).

// Optimization by back prorogation:

for $t = 1, 2, \cdots, I_t$ **do**

Randomly select a sample pair $(\mathbf{x}_i, \mathbf{x}_j, \ell_{ij})$ in $\mathbf{X}$.

Set $\mathbf{h}_i^{(0)} = \mathbf{x}_i$ and $\mathbf{h}_j^{(0)} = \mathbf{x}_j$, respectively.

// Forward propagation

for $m = 1, 2, \cdots, M$ **do**

Do forward propagation to get $\mathbf{h}_i^{(m)}$ and $\mathbf{h}_j^{(m)}$.

end

// Computing gradient

for $m = M, M-1, \cdots, 1$ **do**

Obtain gradient by back propagation according to Eqs. (8) and (9).

end

// Back propagation

for $m = 1, 2, \cdots, M$ **do**

Update $\mathbf{W}^{(m)}$ and $\mathbf{b}^{(m)}$ according to Eqs. (16) and (17).

end

Calculate J_t using Eq (7).

If $t > 1$ and $|J_t - J_{t-1}| < \varepsilon$, go to **Return**.

end

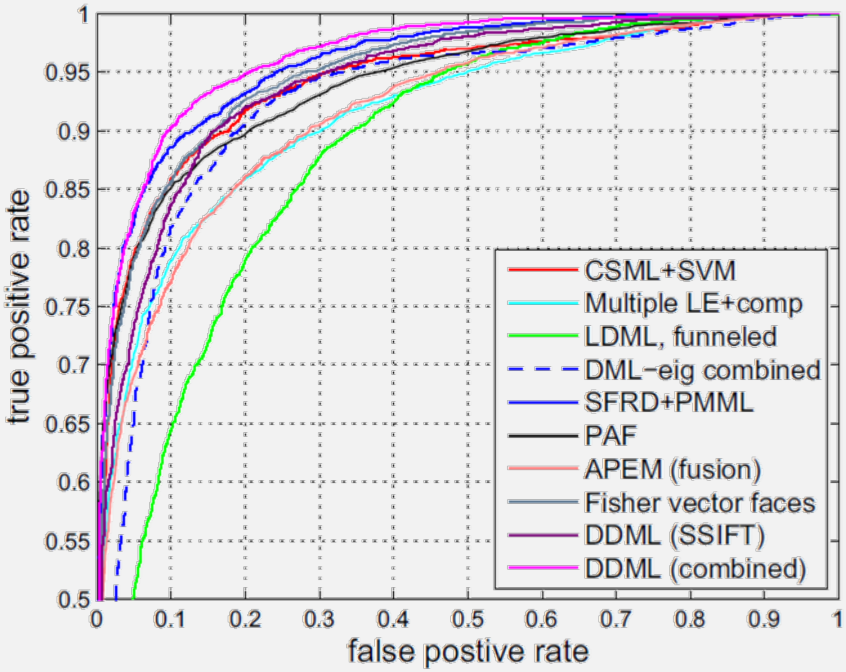
Return: $\{\mathbf{W}^{(m)}, \mathbf{b}^{(m)}\}_{m=1}^{M}$.

Experiments on Face Recognition

- Learned deep metric with combined features achieves the highest performance.

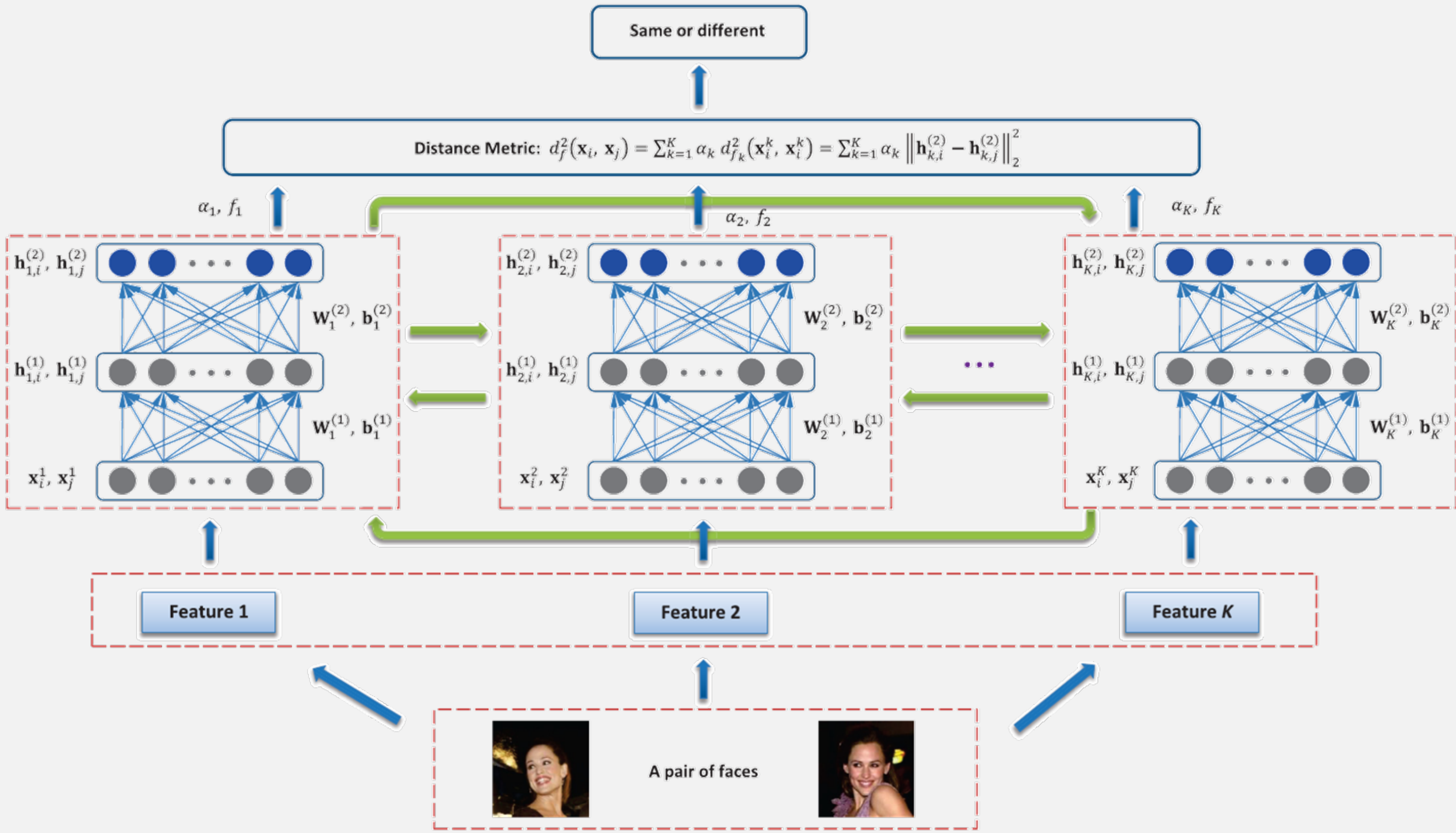
Table 1. Comparison of the mean verification rate and standard error (%) with the shadow metric learning method on the LFW dataset under the image restricted setting.

Feature	DDML	DSML
DSIFT (original)	86.78 ± 2.09	83.68 ± 2.06
DSIFT (square root)	87.25 ± 1.62	84.42 ± 1.80
LBP (original)	85.47 ± 1.85	81.88 ± 1.90
LBP (square root)	87.02 ± 1.62	84.08 ± 1.21
SSIFT (original)	86.98 ± 1.37	84.02 ± 1.47
SSIFT (square root)	$\mathbf{87.83 \pm 0.93}$	84.52 ± 1.38
All features	$\mathbf{90.68 \pm 1.41}$	87.45 ± 1.45



- Junlin Hu, **Jiwen Lu***, and Yap-Peng Tan, Discriminative deep metric learning for face verification in the wild, **CVPR**, 2014.

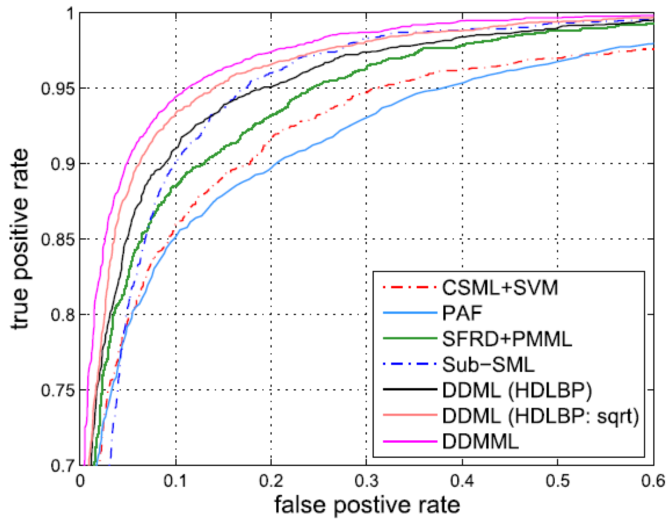
Discriminative Deep Multi-Metric Learning



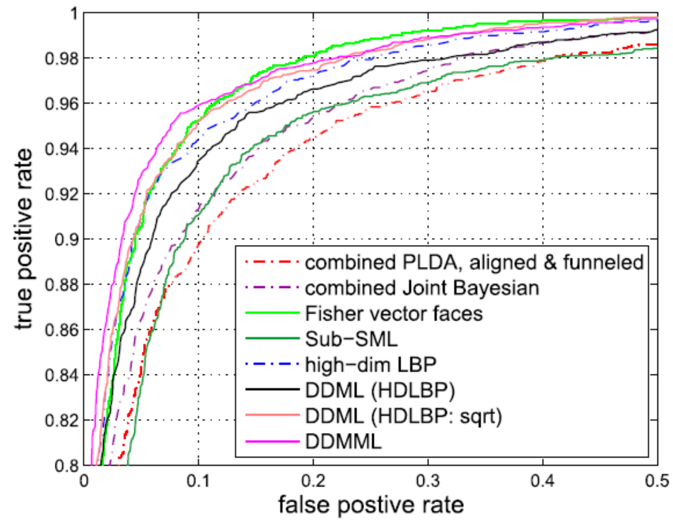
- Jiwen Lu, Junlin Hu, and Yap-Peng Tan, Discriminative deep metric learning for face and kinship verification, TIP, 2017.

Experiments

LFW



(a) restricted

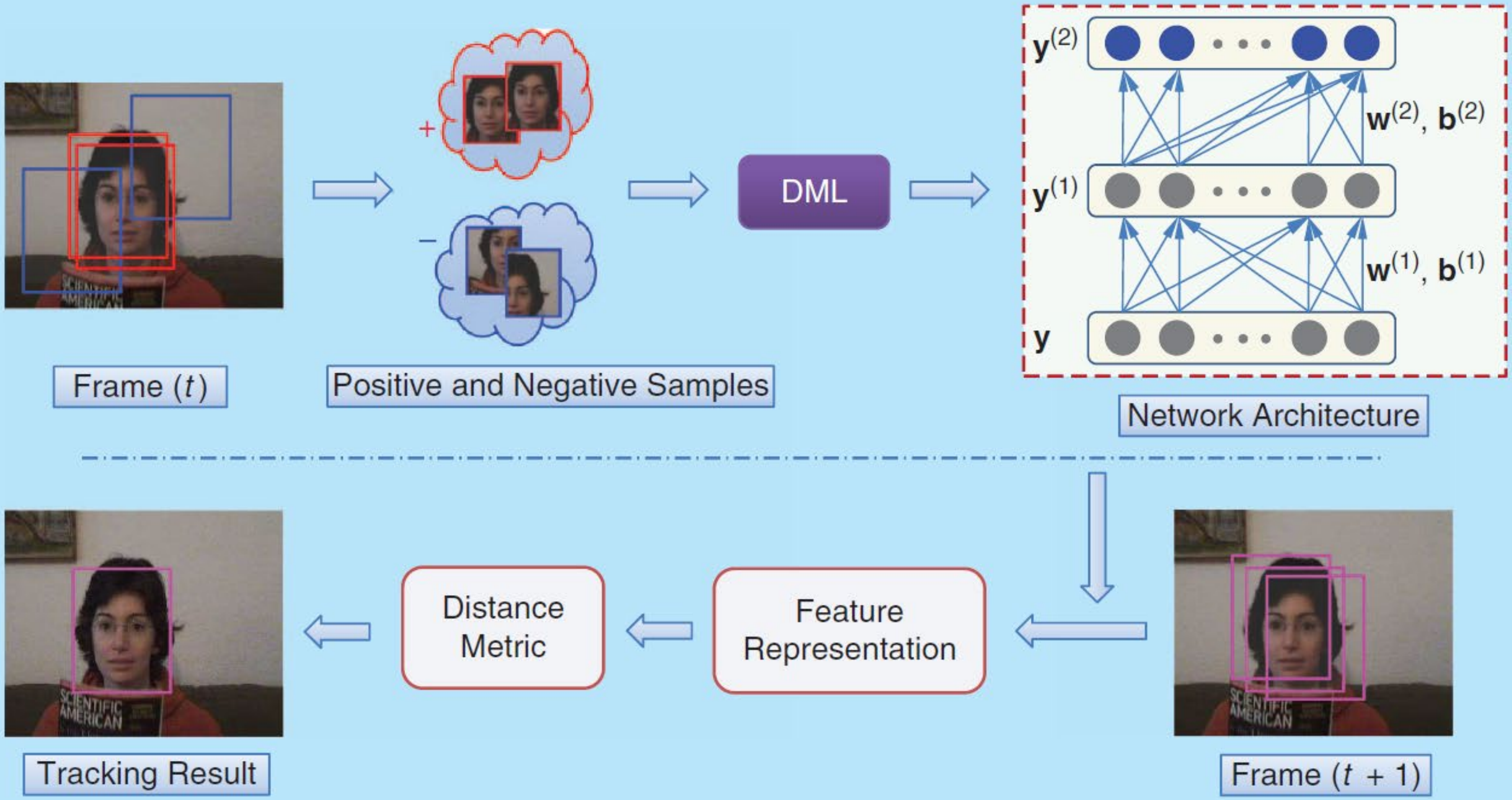


(b) unrestricted

KinFaceW

Method	Feature	KinFaceW-I					KinFaceW-II				
		F-S	F-D	M-S	M-D	Mean	F-S	F-D	M-S	M-D	Mean
DSML	LBP	70.8	67.2	72.5	74.0	71.1	72.4	64.3	67.6	71.2	68.9
DSML	DSIFT	70.0	70.9	73.9	78.1	73.2	75.6	63.8	70.0	74.7	71.0
DSML	HOG	73.9	69.1	70.8	76.9	72.7	74.9	66.5	73.1	73.4	72.0
DSML	LPQ	78.3	72.6	75.1	80.5	76.6	80.0	75.2	76.4	78.3	77.5
DSMML	All	80.4	75.5	77.6	82.1	78.9	83.2	76.0	79.0	81.0	79.8
DDML	LBP	78.4	71.9	75.8	75.8	75.5	81.4	73.8	78.1	77.2	77.6
DDML	DSIFT	78.0	75.9	76.5	83.3	78.4	82.5	75.7	79.1	79.2	79.1
DDML	HOG	80.5	72.8	75.4	81.2	77.5	80.9	75.7	78.8	77.0	78.1
DDML	LPQ	83.8	77.0	78.1	86.6	81.4	84.8	82.6	79.4	81.8	82.2
DDMML	All	**86.4**	**79.1**	**81.4**	**87.0**	**83.5**	**87.4**	**83.8**	**83.2**	**83.0**	**84.3**

Deep Metric Learning for Visual Tracking



- Junlin Hu, **Jiwen Lu***, and Yap-Peng Tan, Deep metric learning for visual tracking, **TCSVT**, 2016.

Deep Metric Learning for Visual Tracking

Visual Tracking

- Dynamical Model: the state transition distribution is modelled by a zero-mean Gaussian distribution, and six affine transformation parameters are assumed to be independent.
- Observation Model: the similarity (or confidence) between template and particle is:

$$p(\mathbf{y}_t|\mathbf{s}_t) = \frac{1}{\Gamma}\exp\Big(-\gamma\ d_f^2(\mathbf{y}_t, \mathbf{m}_t)\Big)$$

Deep Metric Learning for Visual Tracking

Visual Tracking

- Positive samples: Sample image patches around the target within a radius of a few pixels, and resize them into size 32 * 32.
- Negative samples: Sample far away from the target regions, containing both the background and parts of the target object.

Template Update:

Incremental principal component analysis

Deep Metric Learning for Visual Tracking

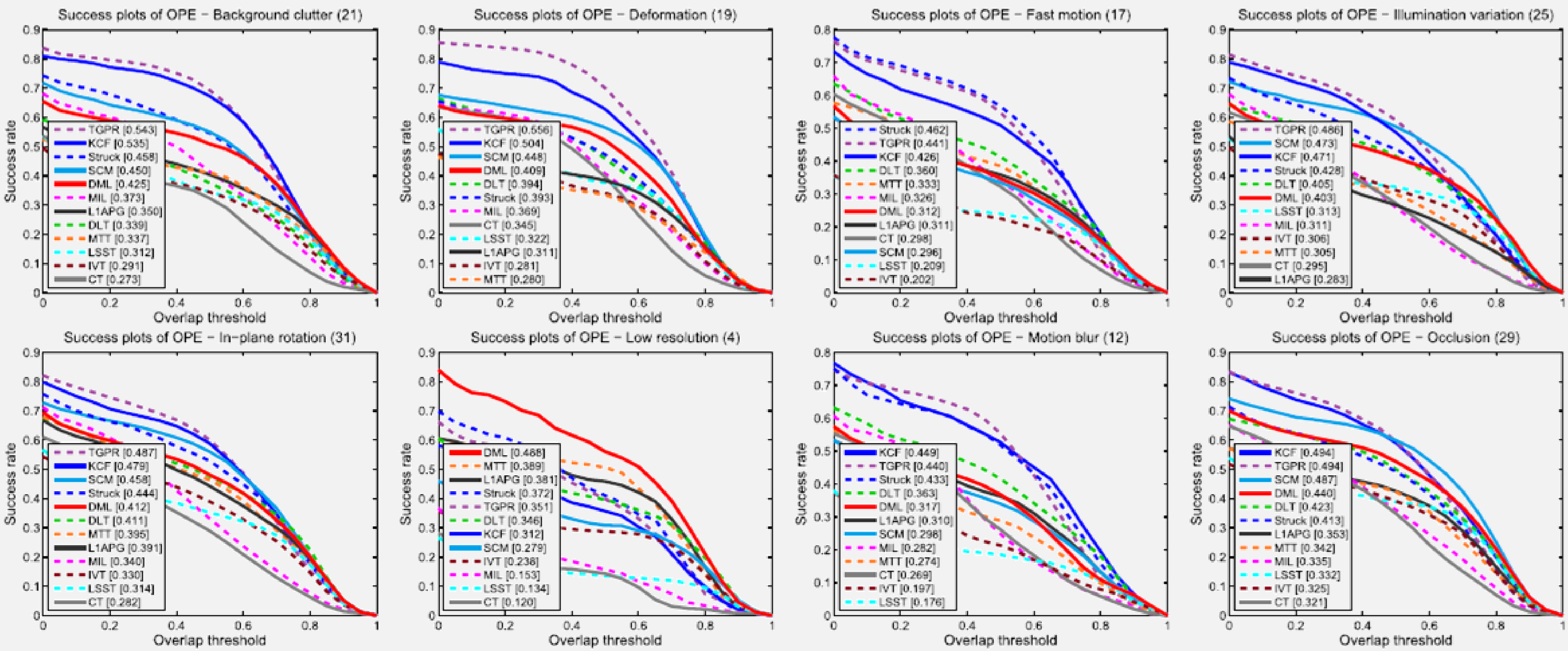
Formulation:

$$\min_{f} \mathcal{O} = \frac{1}{\mathcal{P}} \sum_{\ell_{ij}=1} d_f^2(\mathbf{y}_i, \mathbf{y}_j) - \frac{\alpha}{\mathcal{N}} \sum_{\ell_{ij}=-1} d_f^2(\mathbf{y}_i, \mathbf{y}_j) + \beta \sum_{k=1}^{\mathcal{K}} \left(\left\| \mathbf{W}^{(k)} \right\|_F^2 + \left\| \mathbf{b}^{(k)} \right\|_2^2 \right),$$

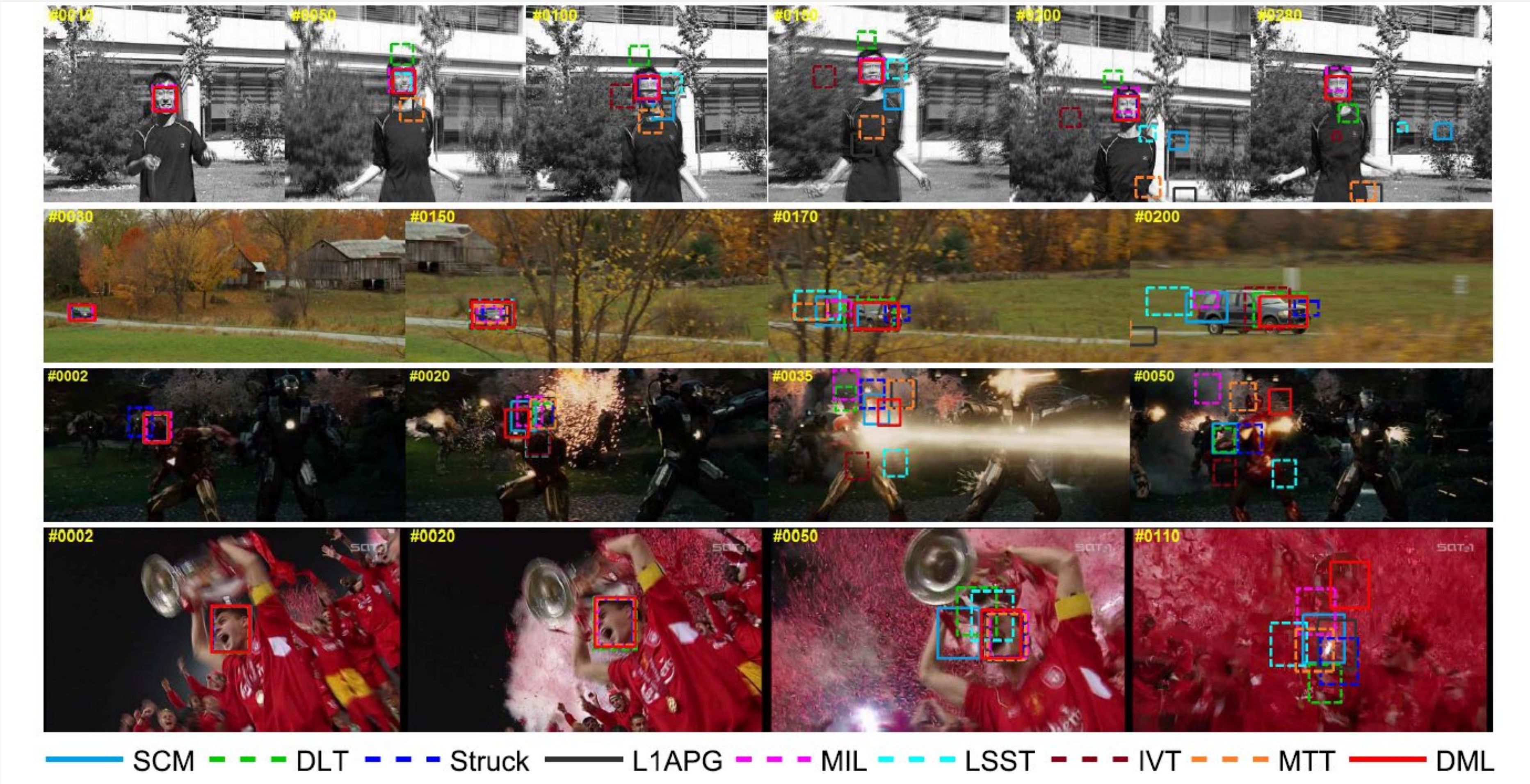
DML aims to seek an optimal nonlinear mapping ***f*** by minimizing the intra-class variations of positive pairs and maximizing the interclass variations of negative pairs in the transformed subspace for utilizing more discriminative information.

Quantitative Analysis

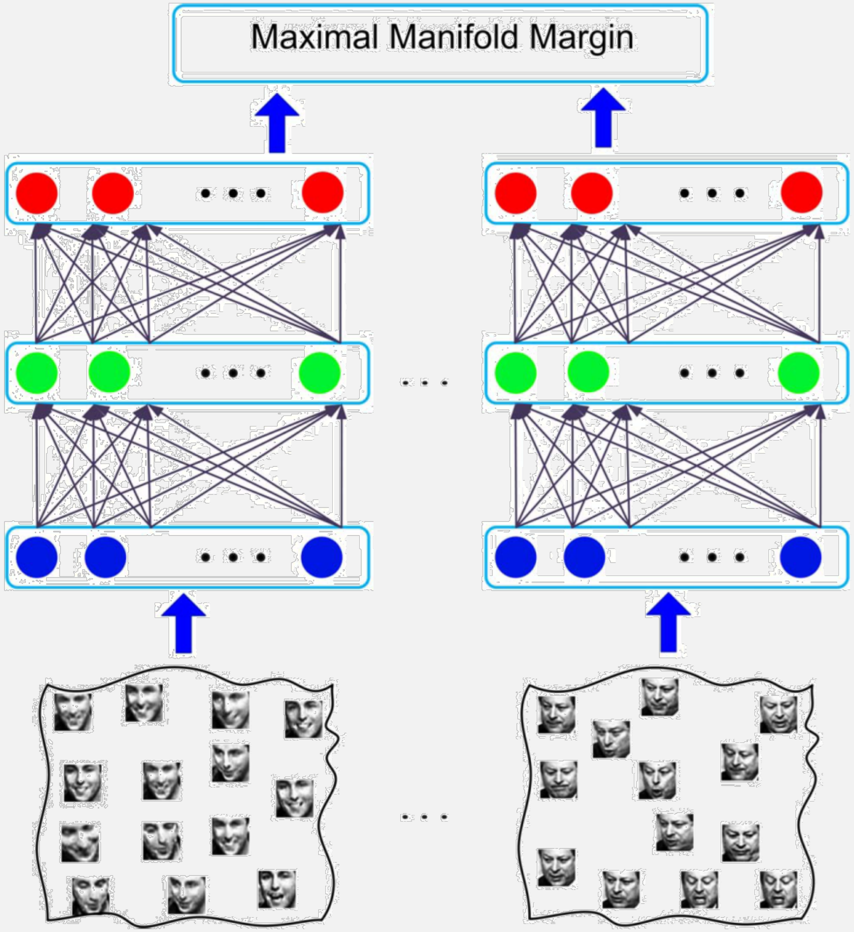
- The proposed DML tracker (in red curve) is ranked fifth among these trackers in both the success and precision plots.



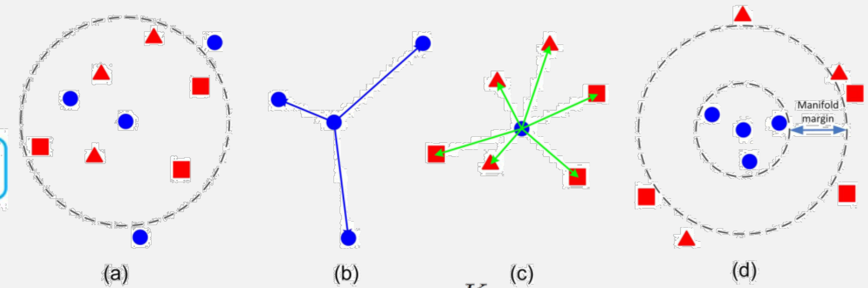
Qualitative Analysis



Multi-Manifold Deep Metric Learning



$$h_{ci}^{L} = s(W_c^L h_{ci}^{L-1} + b_c^L)$$



$$D_1(h_{ci}^L) = \frac{1}{K_1}\sum_{p=1}^{K_1} \|h_{ci}^L - h_{cip}^L\|_2^2$$

$$D_2(h_{ci}^L) = \frac{1}{K_2}\sum_{q=1}^{K_2} \|h_{ci}^L - h_{ciq}^L\|_2^2$$

Objective function

$$\begin{aligned}\min_{f_1, f_2, \cdots, f_C} H &= H_1 + \frac{\lambda}{2} H_2 \\ &= \sum_{c=1}^{C}\sum_{i=1}^{N_c} g\left(D_1(h_{ci}^L) - D_2(h_{ci}^L)\right) \\ &+ \frac{\lambda}{2}\sum_{c=1}^{C}\sum_{l=1}^{L}\left(\|W_c^l\|_F^2 + \|b_c^l\|_2^2\right)\end{aligned}$$

Experimental Results

Method	Honda	Mobo	YTC	PubFig	ETH-80	Year
MSM [38]	92.5 ± 2.3	96.5 ± 2.0	61.7 ± 4.3	57.4 ± 1.7	75.5 ± 4.9	1998
DCC [16]	92.6 ± 2.5	88.9 ± 2.5	65.8 ± 4.5	45.5 ± 1.5	91.8 ± 3.7	2006
MMD [36]	92.1 ± 2.3	92.5 ± 2.9	67.7 ± 3.8	46.3 ± 1.5	86.5 ± 4.5	2008
MDA [34]	94.5 ± 3.2	94.4 ± 2.5	68.1 ± 4.3	48.6 ± 1.6	89.2 ± 3.7	2009
AHISD [2]	91.5 ± 1.8	94.1 ± 1.5	66.5 ± 4.5	62.1 ± 1.4	78.6 ± 4.7	2010
CHISD [2]	93.7 ± 1.9	95.8 ± 1.3	67.4 ± 4.7	64.5 ± 1.5	79.7 ± 4.3	2010
SANP [13]	95.3 ± 3.1	96.1 ± 1.5	68.3 ± 5.2	78.5 ± 1.4	80.5 ± 4.7	2011
CDL [35]	97.4 ± 1.3	92.5 ± 2.9	69.7 ± 4.5	65.5 ± 1.5	86.5 ± 3.7	2012
DFRV [5]	97.4 ± 1.9	94.4 ± 2.3	74.5 ± 4.5	74.5 ± 1.4	87.5 ± 2.7	2012
LMKML [27]	98.5 ± 2.5	94.5 ± 2.5	75.2 ± 3.9	72.5 ± 1.5	92.5 ± 4.5	2013
SSDML [40]	93.5 ± 2.8	95.1 ± 2.2	74.3 ± 4.5	65.5 ± 1.7	87.5 ± 4.7	2013
SFDL [26]	98.5 ± 1.5	96.5 ± 2.3	75.7 ± 3.4	78.5 ± 1.7	90.5 ± 4.7	2014
MMDML	$\mathbf{100.0 \pm 0.0}$	$\mathbf{97.8 \pm 1.0}$	$\mathbf{78.5 \pm 2.8}$	$\mathbf{82.5 \pm 1.2}$	$\mathbf{94.5 \pm 3.5}$	

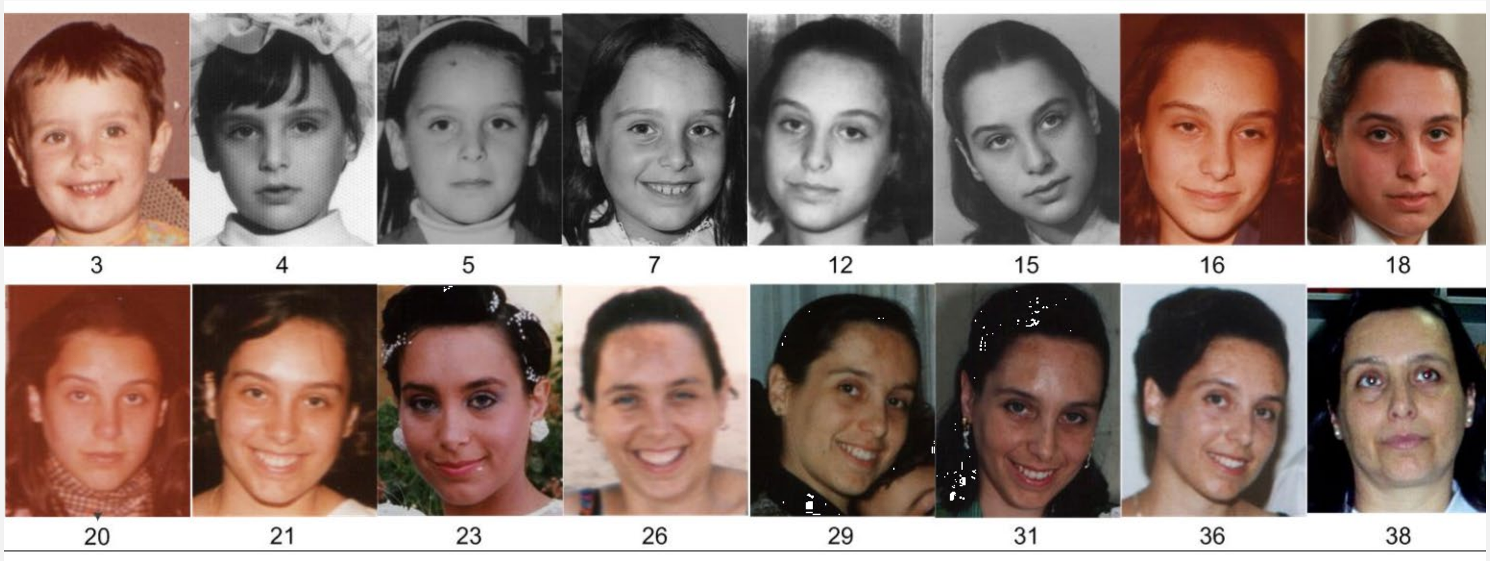
Average classification rates of different methods on different datasets

2.2 Order-Preserving Deep Metric Learning

[6] Hao Liu, **Jiwen Lu***, Jianjiang Feng, and Jie Zhou, Ordinal deep learning for facial age estimation, **T-CSVT**, 2018, accepted.
[7] Hao Liu, **Jiwen Lu***, Jianjiang Feng, and Jie Zhou, Label-sensitive deep metric learning for facial age estimation, **T-IFS**, 2018.

Problem Setting

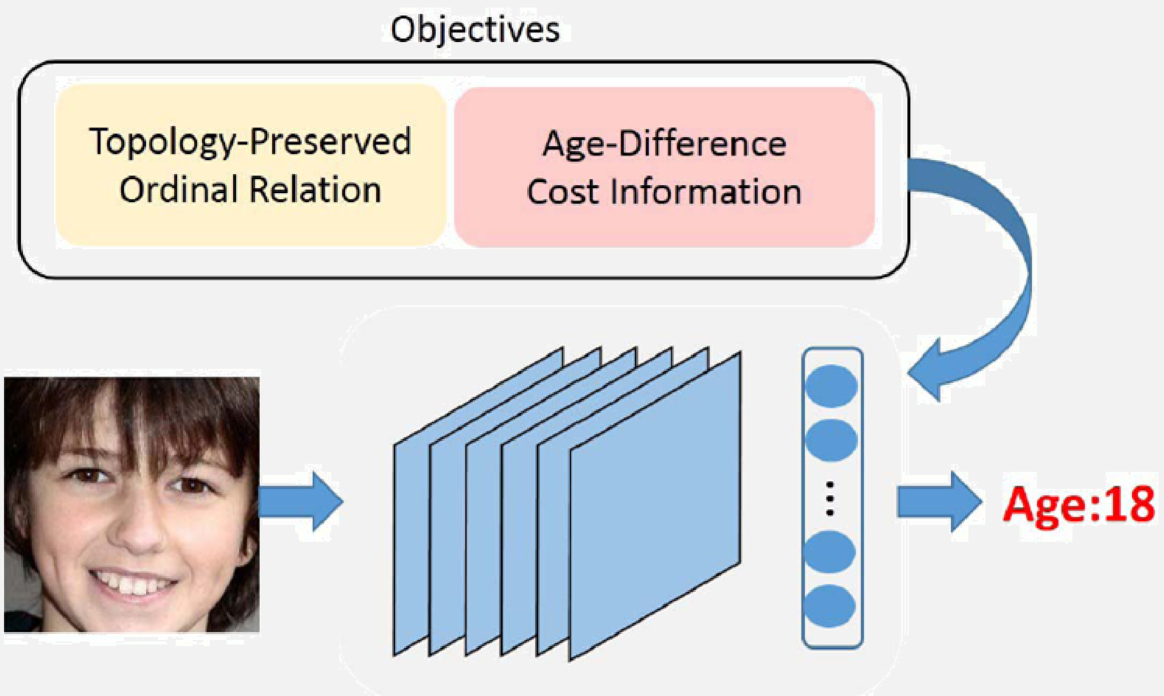
- Facial Age Estimation, *e.g.* FG-NET



- Challenges
  - Nonlinear relationship between facial images and age labels including facial variations due to expressions, cluttered background and occlusions
  - Age labels exhibits in an chronological order (ordinal problem).

Ordinal Deep Metric Learning

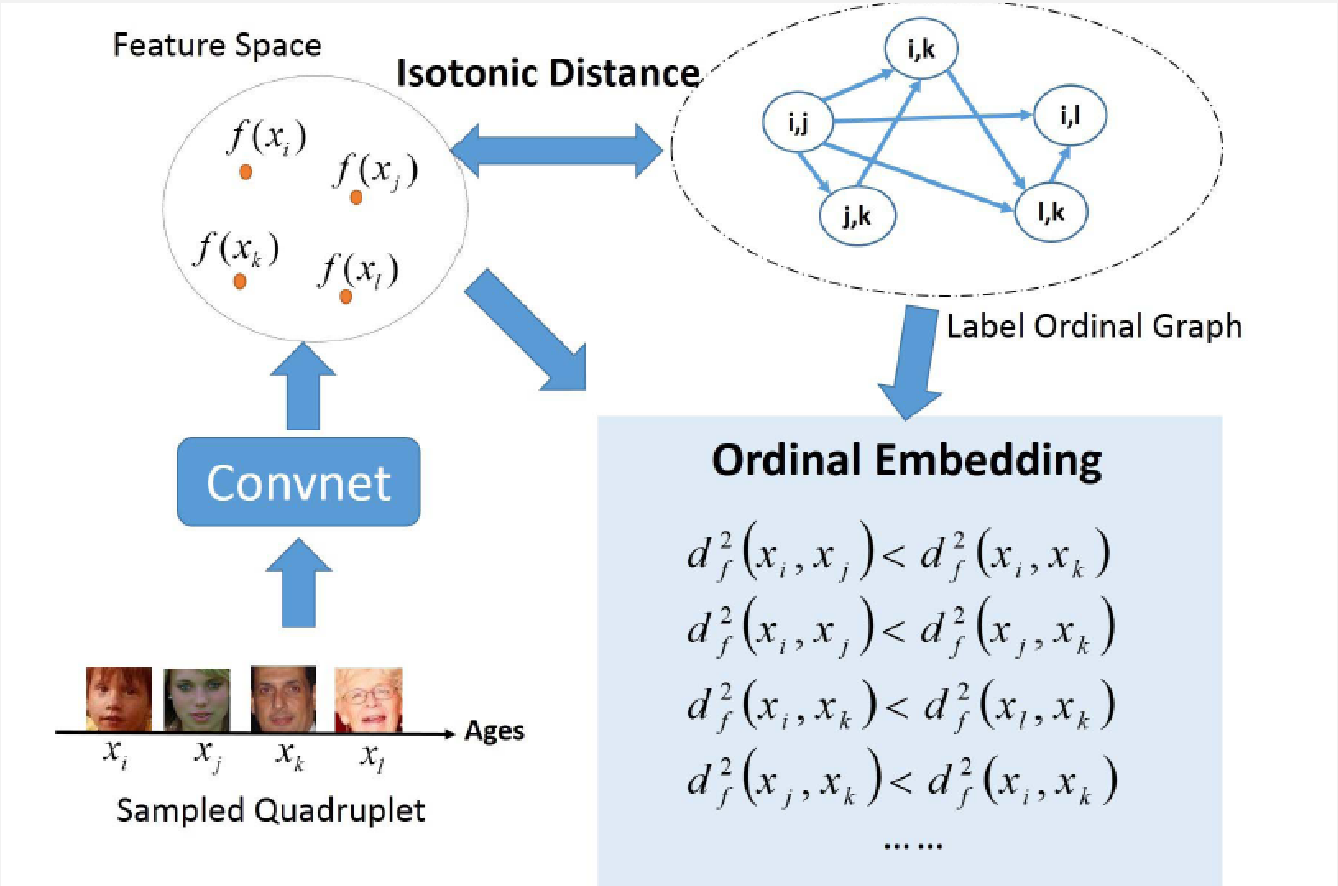
- Two criterions to exploit ordinal relation in the learned metric.



- Hao Liu, **Jiwen Lu***, Jianjiang Feng, and Jie Zhou, Ordinal deep learning for facial age estimation, **TCSVT**, 2018, accepted.

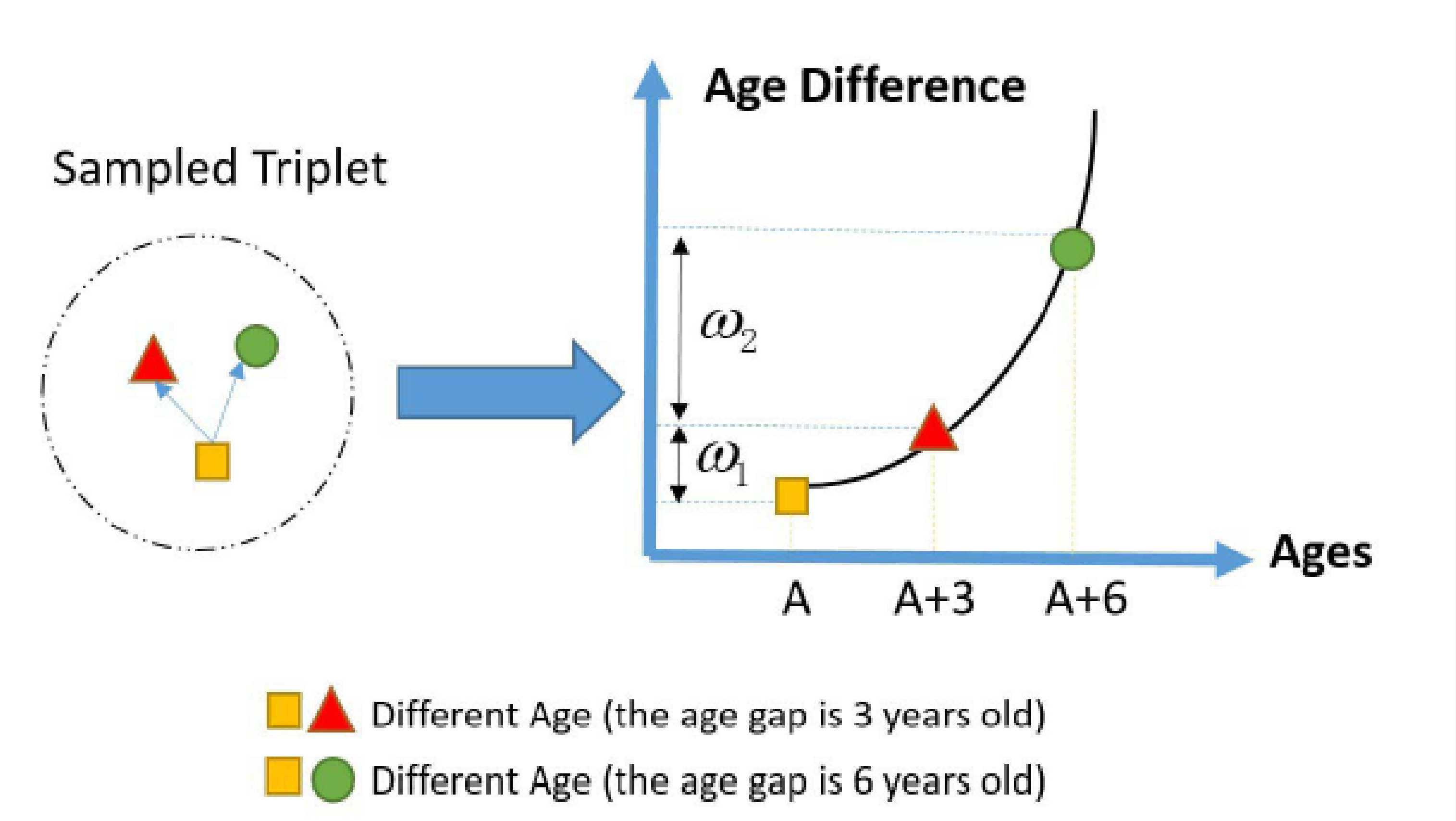
Ordinal Deep Metric Learning

- Topology-Preserving Ordinal Relation



Ordinal Deep Metric Learning

- Age-Difference Cost Information



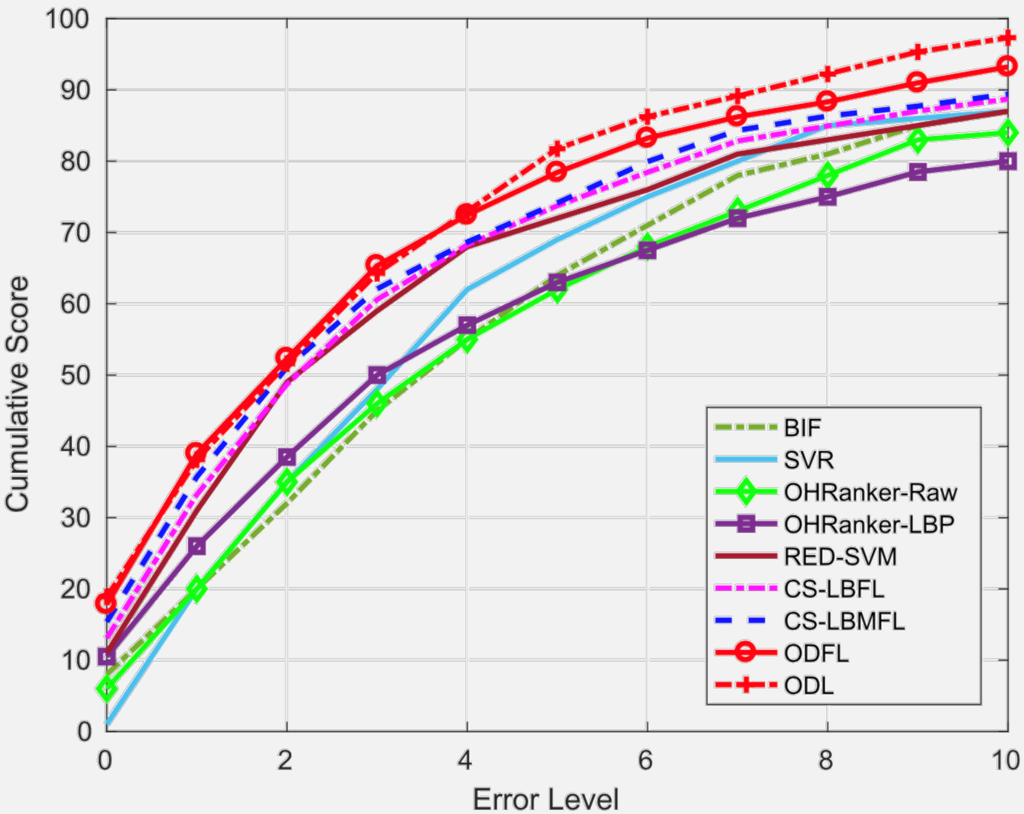
Ordinal Deep Metric Learning

- Formulation

$$\begin{aligned}\min_{\{\mathbf{W},\mathbf{b}\}} J &= J_1 + \lambda_1 J_2 + \lambda_2 J_3 \\ &= \sum_{v_{ij}, v_{kl} \in G} \zeta(v_{ij}, v_{kl}) \cdot \max[0, \alpha - d_f^2(\mathbf{x}_i, \mathbf{x}_j) + d_f^2(\mathbf{x}_k, \mathbf{x}_l)] \\ &\quad + \lambda_1 \sum_{p}^{P} \left(1 - \ell_{p1,p2}(\tau - d_f^2(\mathbf{x}_{p1}, \mathbf{x}_{p2})) \cdot \omega_{y_{p1}, y_{p2}}\right) \\ &\quad + \lambda_2 \sum_{m=1}^{M} (\|\mathbf{W}^{(m)}\|_F^2 + \|\mathbf{b}^{(m)}\|_2^2),\end{aligned}$$

- Optimization: landmark-based relaxation

Experiments on In-the-wild Dataset

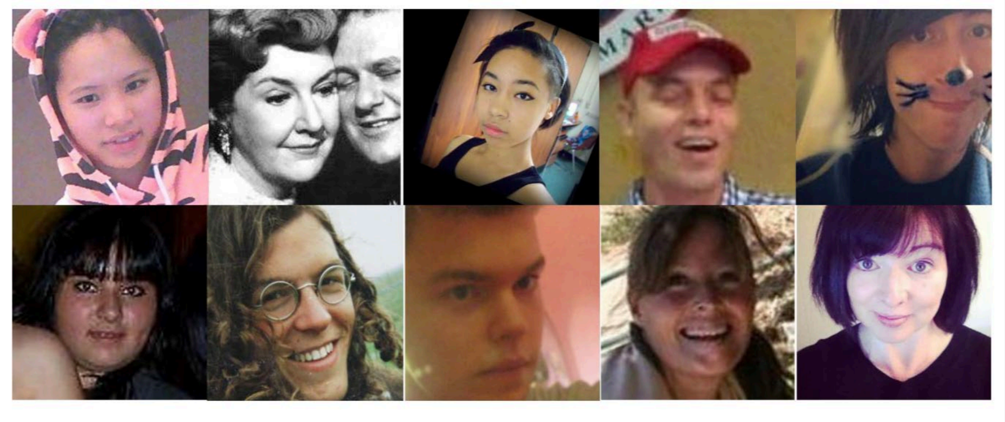


Hand-Crafted Methods	MAE
BIF+KNN	8.24
Raw+OHRanker [9]	6.25
LBP+OHRanker [9]	4.92
BIF+OHRanker [9]	4.48
MLP [22]	6.95
RUN [71]	5.78
AGES [1]	6.77
LARR [28]	5.07
PFA [72]	4.97
KAGES [73]	6.18
MSA [74]	5.36
SSE [75]	5.21
mKNN [76]	5.21
MTWGP [21]	4.83
RED-SVM [8]	5.21
PLO [77]	4.82
LDL [22]	5.77
CA-SVR [63]	4.67
CSOHR [68]	4.70
CS-LBFL [12]	4.43
CS-LBMFL [12]	4.36
CPNN [22]	4.76
Deep Learning-Based Methods	**MAE**
Deep Reg	4.88
GA-DFL [53]	3.93
ODFL + OHRanker	3.89
ODL (Cross-Entropy)	**3.71**

Age Estimation Results

- Selected examples where errors are below one year old.

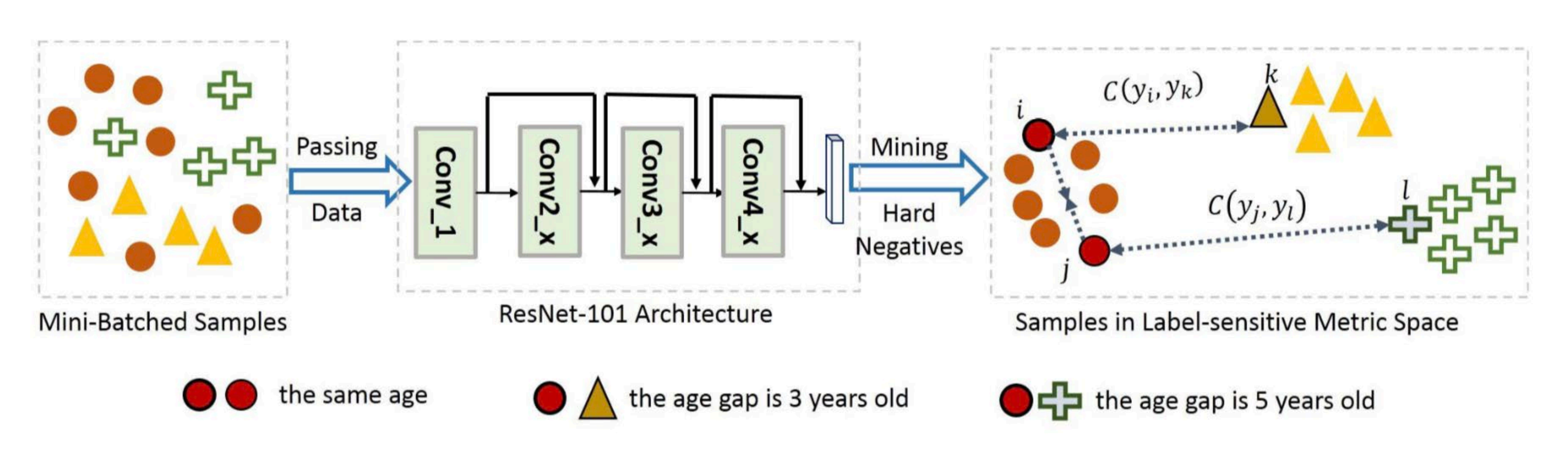
- Selected examples where errors are larger than 5 years old.



Label-Sensitive Deep Metric Learning

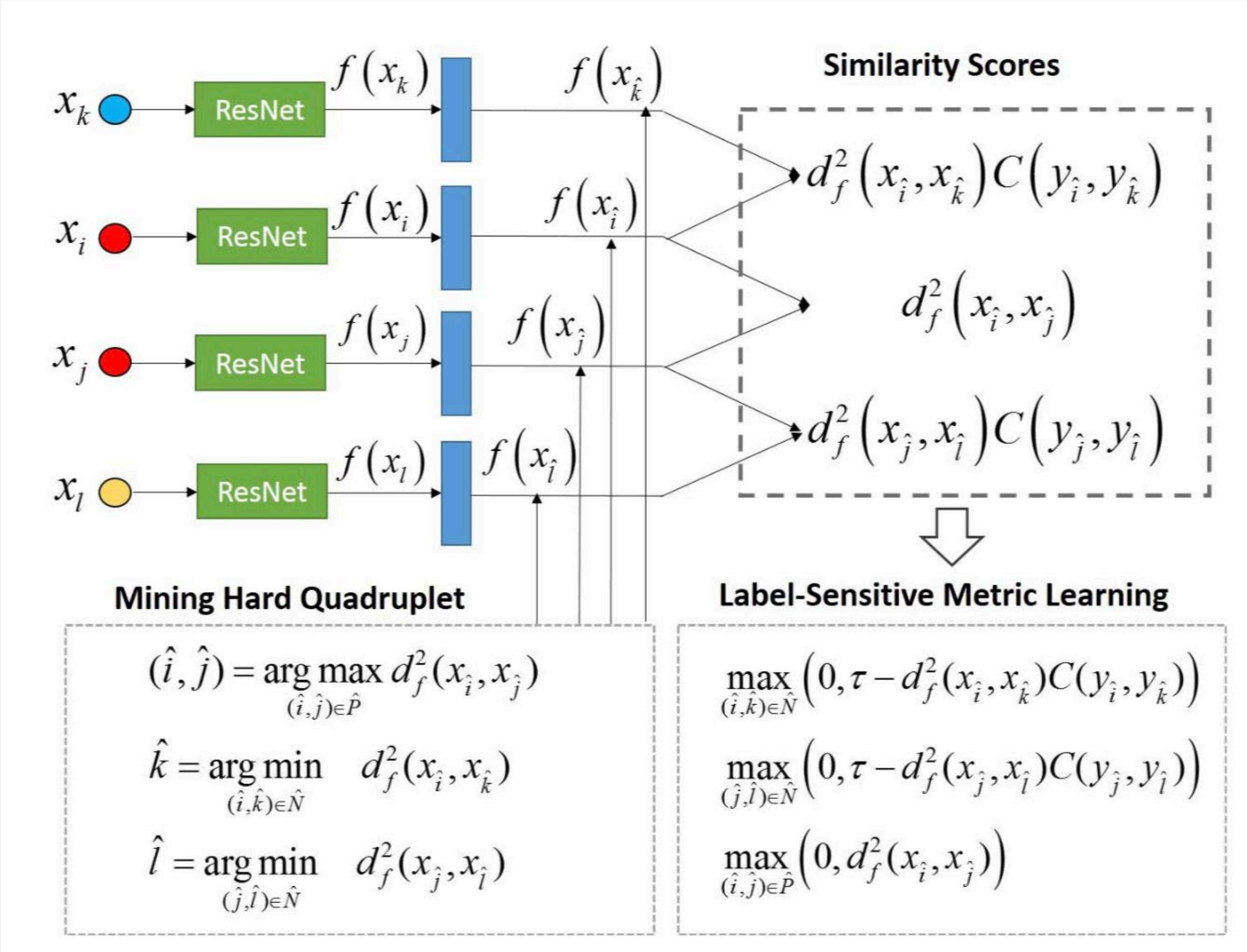
Motivation

- Total negative samples catastrophically costs
- Mining hard meaning samples in the learned metric



- Hao Liu, **Jiwen Lu*,** Jianjiang Feng, and Jie Zhou, Label-sensitive deep metric learning for facial age estimation, **TIFS**, 2018.

Label-Sensitive Deep Metric Learning



Label-Sensitive Deep Metric Learning

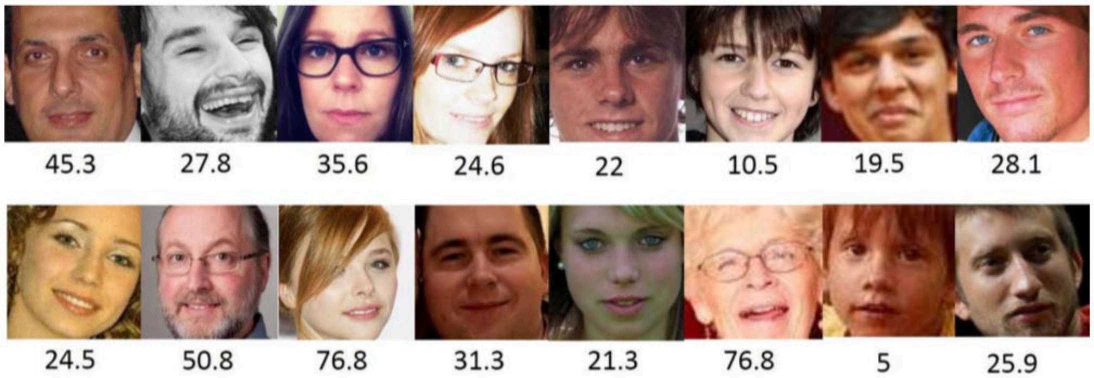
Formulation

$$\min_f \; J = J_1 + \lambda J_2 + \mu J_3$$
$$= \sum_{(\hat{i},\hat{j},\hat{k},\hat{l})} \left(\varepsilon_{\hat{i},\hat{k}} + \epsilon_{\hat{j},\hat{l}}\right) + \lambda \sum_{(\hat{i},\hat{j})} \rho_{\hat{i},\hat{j}} + \mu \|\mathbf{W}\|_F^2,$$
$$\text{subject to} \quad \max_{(\hat{i},\hat{k})\in\hat{\mathcal{N}}} \left(0, \tau - d_f(\mathbf{x}_{\hat{i}}, \mathbf{x}_{\hat{k}}) C(y_{\hat{i}}, y_{\hat{k}})\right)^2 \le \varepsilon_{\hat{i},\hat{k}},$$
$$\max_{(\hat{j},\hat{l})\in\hat{\mathcal{N}}} \left(0, \tau - d_f(\mathbf{x}_{\hat{j}}, \mathbf{x}_{\hat{l}}) C(y_{\hat{j}}, y_{\hat{l}})\right)^2 \le \epsilon_{\hat{j},\hat{l}},$$
$$\max_{(\hat{i},\hat{j})\in\hat{\mathcal{P}}} \left(0, d_f(\mathbf{x}_{\hat{i}}, \mathbf{x}_{\hat{j}})\right)^2 \le \rho_{\hat{i},\hat{j}},$$
$$\varepsilon_{\hat{i},\hat{k}} \ge 0, \quad \epsilon_{\hat{j},\hat{l}} \ge 0, \quad \rho_{\hat{i},\hat{j}} \ge 0,$$

Hard-Mining

$$(\hat{i}, \hat{j}) = \arg\max_{(\hat{i},\hat{j})\in\hat{\mathcal{P}}} \; d_f^2(\mathbf{x}_{\hat{i}}, \mathbf{x}_{\hat{j}}),$$
$$\hat{k} = \arg\min_{(\hat{i},\hat{k})\in\hat{\mathcal{N}}} \; d_f^2(\mathbf{x}_{\hat{i}}, \mathbf{x}_{\hat{k}}),$$
$$\hat{l} = \arg\min_{(\hat{j},\hat{l})\in\hat{\mathcal{N}}} \; d_f^2(\mathbf{x}_{\hat{j}}, \mathbf{x}_{\hat{l}}),$$

Evaluation on the Challenge Dataset



Method	Model Description	Gaussian Error	External Datasets
BIF [11]	BIF [11] + KNN	0.89	-
BIF [11]	BIF [11] + OHRANK [9]	0.55	-
VGG (softmax, Exp) [74]	Deep Expectation	0.51	-
VGG (softmax, Exp) [74]	Deep Expectation	**0.28**	D_6
VGG (softmax, Exp) [75]	with pretrained VGG-16 Face Net [64]	**0.28**	D_6
CS-LBFL [15]	Cost-Sensitive Local Binary Feature Learning	0.45	-
Best from DCNN [31]	deep convolutional neural networks	0.359	D_1, D_2, D_3
Cascaded-CNN [32]	with error correction	0.355	D_3, D_4,D_5
Cascaded-CNN [32]	with end-to-end finetuning	0.312	D_3, D_4,D_5
Cascaded-CNN [32]	with end-to-end finetuning and error correction	0.297	D_3, D_4,D_5
LSDML	with OHRANK [9]	0.37	-
M-LSDML	with OHRANK [9]	0.34	D_2,D_5
LSDML	with end-to-end finetuning [19]	0.328	-
M-LSDML	with end-to-end finetuning [19]	0.315	D_2,D_5

D_1-CASIA-WebFace [76], D_2-MORPH [46], D_3-AdienceFaces [46]
D_4-Images of Groups [77], D_5-FG-NET [22], D_6-IMDB-WIKL (https://data.vision.ee.ethz.ch/cvl/rrothe/imdb-wiki/)

2.3 Deep Structural Metric Learning

[8] Hao Liu, **Jiwen Lu***, Jianjiang Feng, and Jie Zhou, Learning deep sharable and structural detectors for face alignment, **T-IP**, 2017.
[9] Hao Liu, **Jiwen Lu***, Jianjiang Feng, and Jie Zhou, Two-stream transformer networks for video-based face alignment, **T-PAMI**, 2018, accepted.

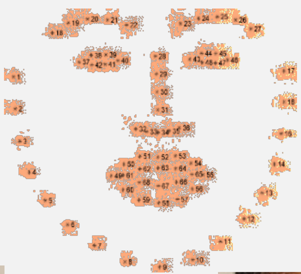
Face Alignment From a Metric Learning View

- Input: Image pixels
- Output: Facial landmarks

  Point distribution model

$$\mathbf{S} = [p_1, p_2, \cdots, p_l, \cdots, p_L] \in \mathbb{R}^{2L}$$

Shape Prior



- Objective

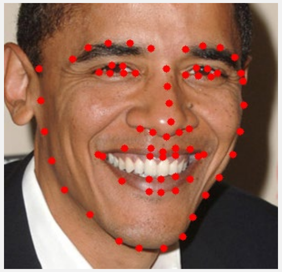
$$J = \|\hat{\mathbf{S}} - \mathbf{S}^*\|_2^2$$

Subspace Learning

GT shape coordinates

Euclidean Metric

image



coordinates

Existing Solutions

❑ Model-based Optimization

- PCA shape model
- holistic and local appearance
- active shape and appearance fitting

❑ Cascaded Shape Regression

- shape refinement
- shape-index features
- cascaded/coarse-to-fine

- ASM [Coots et al., CVIU 1995]
- AAM [Coots et al., PAMI 2004]
- CLM [Coots et al., BMVC 2006]

- ESR, [Cao et al., CVPR 2012]
- SDM, [Xiong et al., CVPR 2013]
- CFSS, [Zhu et al., CVPR 2015]

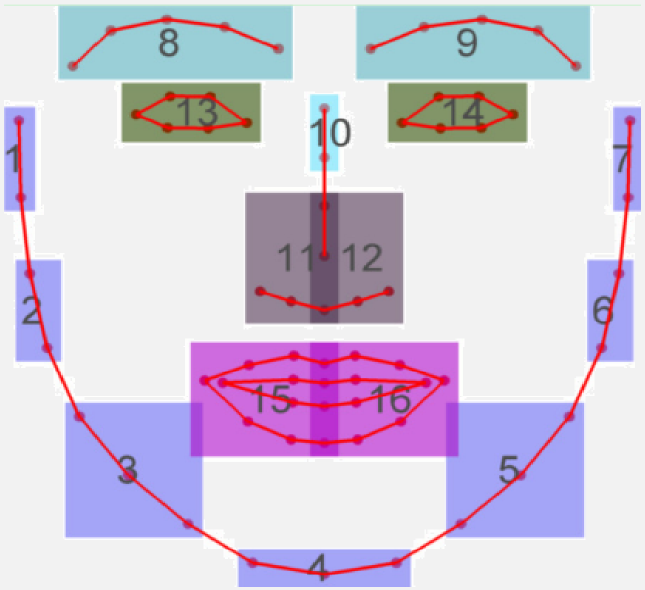
Key Points for Alignment Metric

- Hand-crafted Representation
  - HOG, SIFT, geometric-based (2D-3D projection)
- Shape-informative Representation
  - Local and global → Structural Learning
  - Robustness → Hierarchical Learning
- Knowledge-sharable Representation
  - Correlated Attributes → Multi-task Learning
  - Video-based → Spatial-temporal modeling

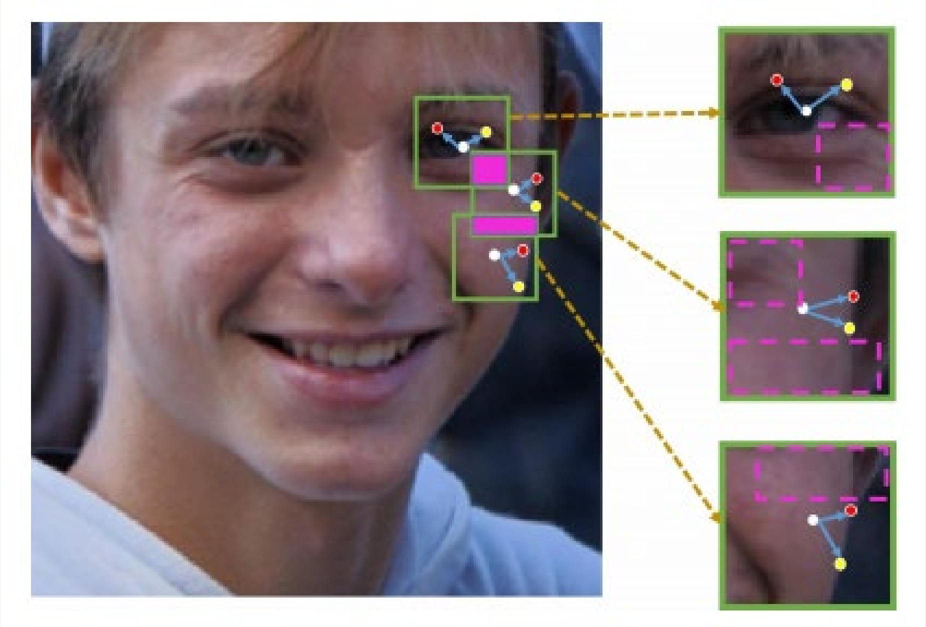
- Hao Liu, **Yueqi Duan, Jiwen Lu**, Representation Learning for Face Alignment and Face Recognition, **FG Tutorial**, 2018.

Deep Structural Metric Learning

Motivation



Semantic Facial Parts

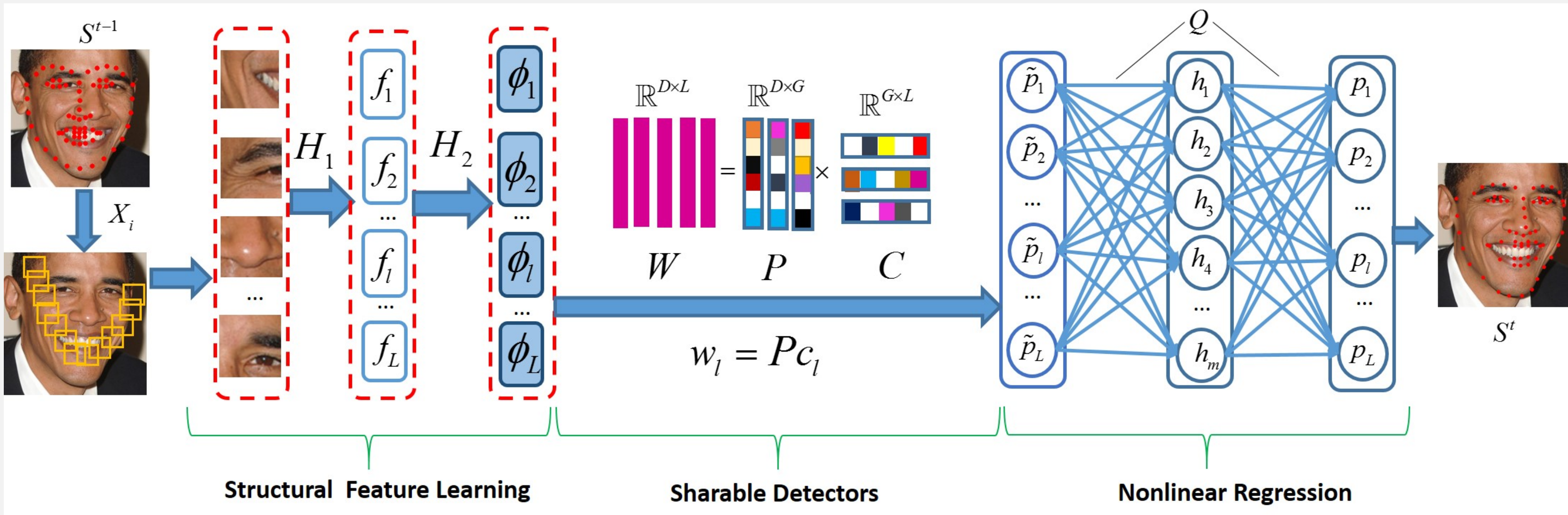


Structural learning from neighbouring landmarks

- Hao Liu, **Jiwen Lu*,** Jianjiang Feng, and Jie Zhou, Learning deep sharable and structural detectors for face alignment, **TIP**, 2017.

Deep Structural Metric Learning

☐ Architecture



☐ Objective

$$\min_{\{\mathbf{P},\mathbf{C},\mathbf{H},\mathbf{Q}\}} J = J_1(\mathbf{P},\mathbf{C},\mathbf{H},\mathbf{Q}) + J_2(\mathbf{P},\mathbf{C})$$

$$= \sum_{j}^{G} \sum_{i}^{N} \frac{1}{2} \left\| \mathbf{S}_i^* - \mathbf{S}_i^0 - \mathbf{Q}\left[(\mathbf{P}c_j)^T \mathbf{\Phi}_i \right] \right\|_2^2$$

$$+ (\gamma \|\mathbf{C}\|_1 + \beta \|\mathbf{P}\|_1)$$

Experimental Results

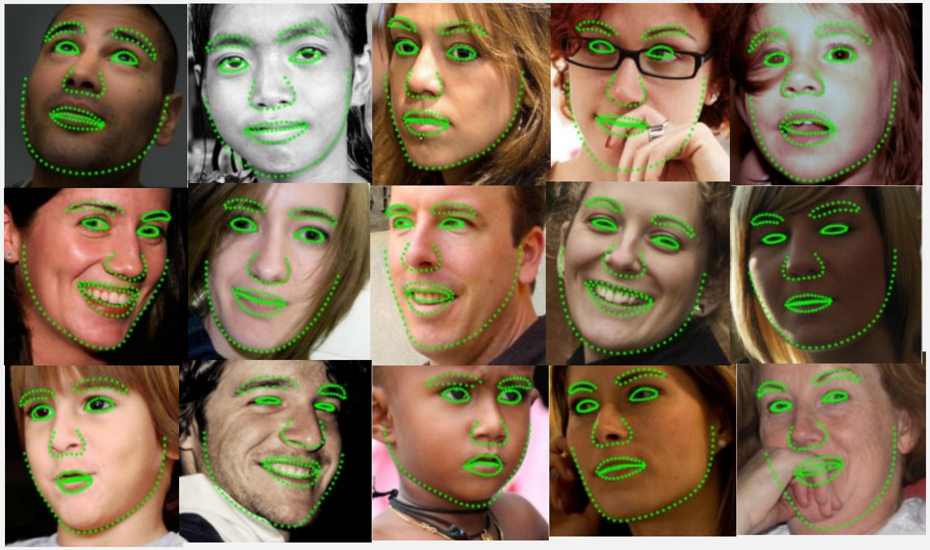
- Robustness to various poses

Method	LFPW 68-pts	HELEN 68-pts	HELEN 192-pts	Common Set 68-pts	Challenging Set 68-pts	Full Set 68-pts
FPLL	8.29	8.16	-	8.22	18.33	10.20
DRMF	6.57	6.70	-	6.65	19.79	9.22
RCPR	6.56	5.93	6.50	6.18	17.26	8.35
GN-DPM	5.92	5.69	-	5.78	-	-
SDM	5.67	5.50	5.85	5.57	15.40	7.50
CFAN	5.44	5.53	-	5.50	-	-
ERT	-	-	4.90	-	-	6.40
BPCPR	-	-	-	5.24	16.56	7.46
ESR	-	-	5.70	5.28	17.00	7.58
LBF	-	-	5.41	4.95	11.98	6.32
LBF fast	-	-	5.80	5.38	15.50	7.37
Deep Reg	-	-	-	4.51	13.80	6.31
CFSS	4.87	*4.63*	*4.74*	*4.73*	*9.98*	*5.76*
CFSS Practical	4.90	4.72	4.84	4.73	10.92	5.99
TCDCN	*4.57*	4.60	*4.63*	4.80	8.60	*5.54*
DCRFA	*4.57*	*4.25*	-	*4.19*	**8.42**	**5.02**
R-DSSD*	4.77	4.31	4.95	4.57	10.86	5.91
R-DSSD	**4.52**	**4.08**	**4.62**	**4.16**	*9.20*	5.59

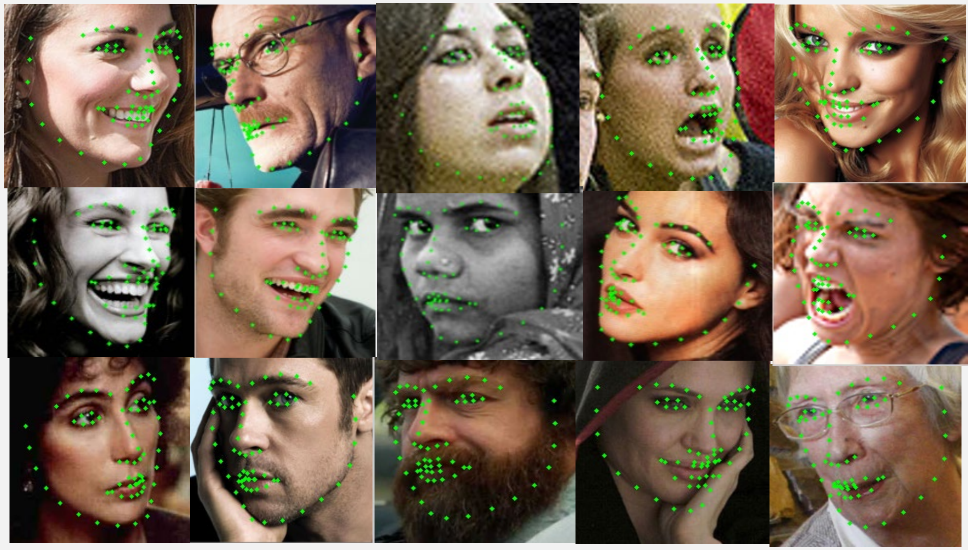
- Hao Liu, **Jiwen Lu***, Jianjiang Feng, and Jie Zhou, Learning deep sharable and structural detectors for face alignment, **TIP**, 2017.

Evaluation on Landmark Density

- Robustness to density, expression and poses



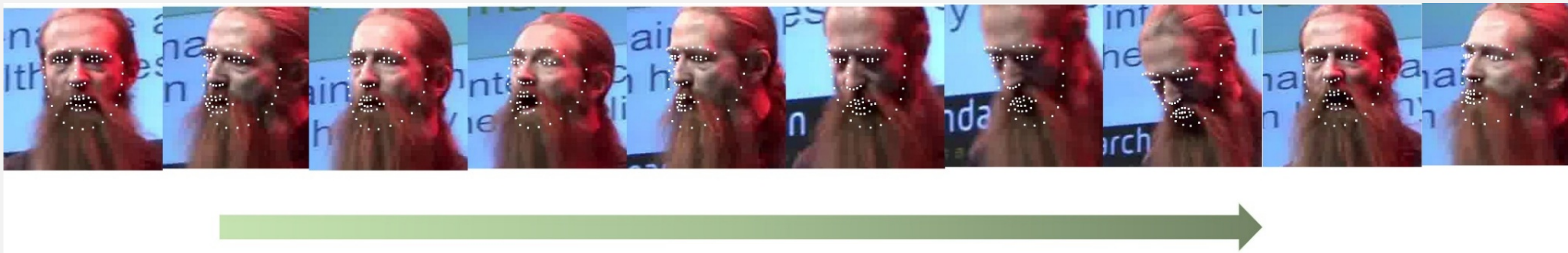
HELEN 192-pts



IBUG 68-pts

- Hao Liu, **Jiwen Lu*,** Jianjiang Feng, and Jie Zhou, Learning deep sharable and structural detectors for face alignment, **TIP**, 2017.

Deep Spatial-Temporal Metric Learning

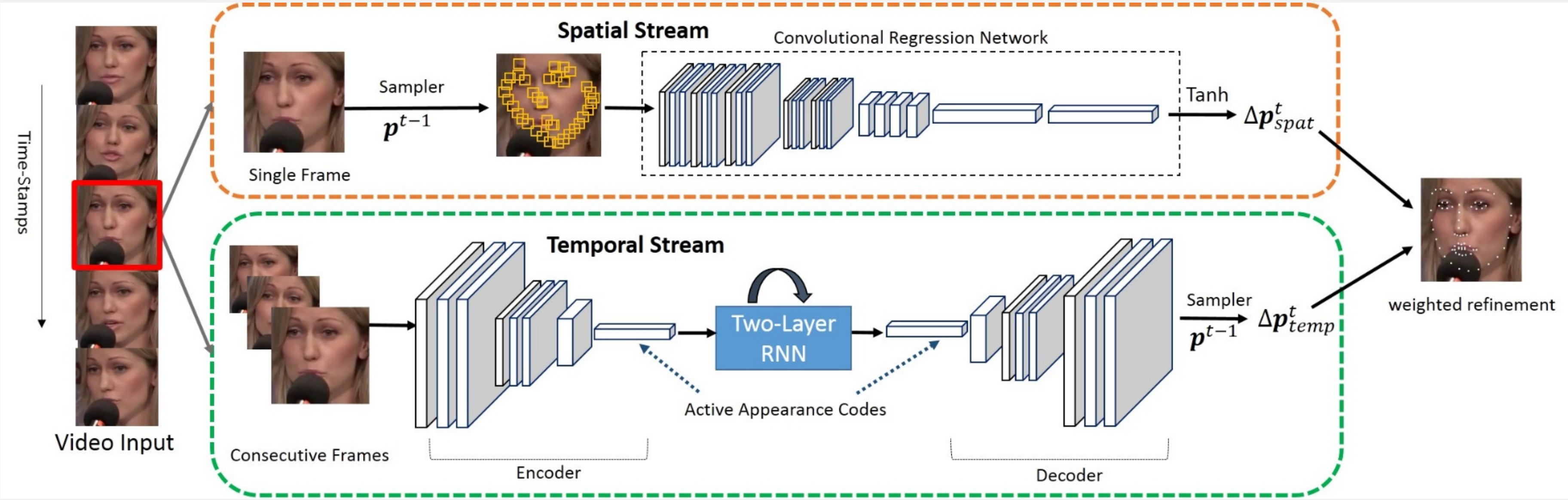


☐ Problem Formulation

- Input：face sequence $\mathbf{x}_i^{1:T} = \{\mathbf{x}_i^1, \mathbf{x}_i^2, ..., \mathbf{x}_i^t, ..., \mathbf{x}_i^T\}$
- Output：landmarks for t-th frame $\mathbf{p}_i^t = [p_1, p_2, \cdots, p_l, \cdots, p_L]_i^{t'}$
- Goal: sequential face alignment $\{\mathbf{x}^t\}^{t=1:T} \longrightarrow \{\mathbf{p}^t\}^{t=1:T}$

☐ Hao Liu, **Jiwen Lu*,** Jianjiang Feng, and Jie Zhou, Two-stream transformer networks for video-based face alignment, **TPAMI**, 2017, accepted.

Two-Stream Deep Metric Learning



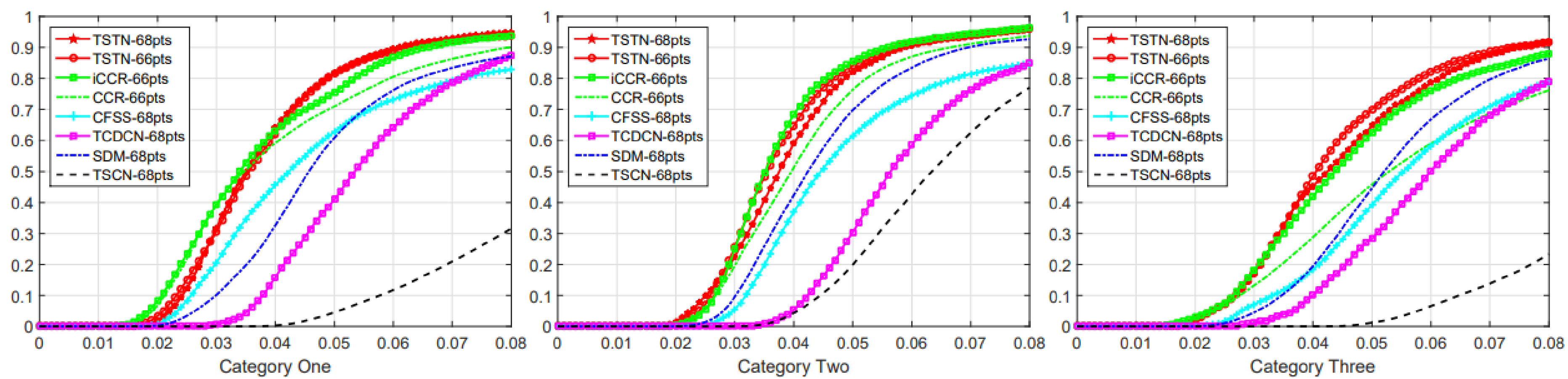
$$\min_f J = \sum_i^N \sum_t^T \frac{1}{2} \left\| \Delta \mathbf{p}_i^t - \beta_1 f_{\text{spat}}\left(\mathbf{x}_i^t\right) - \beta_2 f_{\text{temp}}\left(\mathbf{x}_i^t\right) \right\|_2^2,$$

$$\text{subject to} \quad \beta_1 + \beta_2 = 1.$$

- Hao Liu, **Jiwen Lu*,** Jianjiang Feng, and Jie Zhou, Two-stream transformer networks for video-based face alignment, **TPAMI**, 2017, accepted.

Quantitative Evaluation

Methods	Model Description	Category 1	Category 2	Category 3	Challset [25]	-pts	Year
SDM [46]	Cascaded Linear Regression	7.41	6.18	13.04	7.44		2013
TSCN [35][1]	Two-Stream Action Network	11.61	11.59	17.67	-		2014
TSCN [35][1,2]	Two-Stream Action Network	12.54	7.25	13.13	-		2014
CFSS [50]	Coarse-to-Fine Shape Searching	7.68	6.42	13.67	5.92	68	2015
PIEFA [26]	Personalized Ensemble Learning	-	-	-	6.37		2015
REDN [25]	Recurrent Auto-Encoder Net	-	-	-	6.25		2016
TCDCN [49]	Multi-Task Deep CNN	7.66	6.77	14.98	7.27		2016
TSTN	Two-Stream Transformer Net	**5.36**	**4.51**	**12.84**	**5.59**		-
CCR [32]*	Cascaded Continuous Regression	7.26	5.89	15.74	-		2016
iCCR [32]*	Cascaded Continuous Regression	*6.71*	**4.00**	*12.75*	-	66	2016
TSTN	Two-Stream Transformer Net	**5.21**	*4.23*	**10.11**	-		-



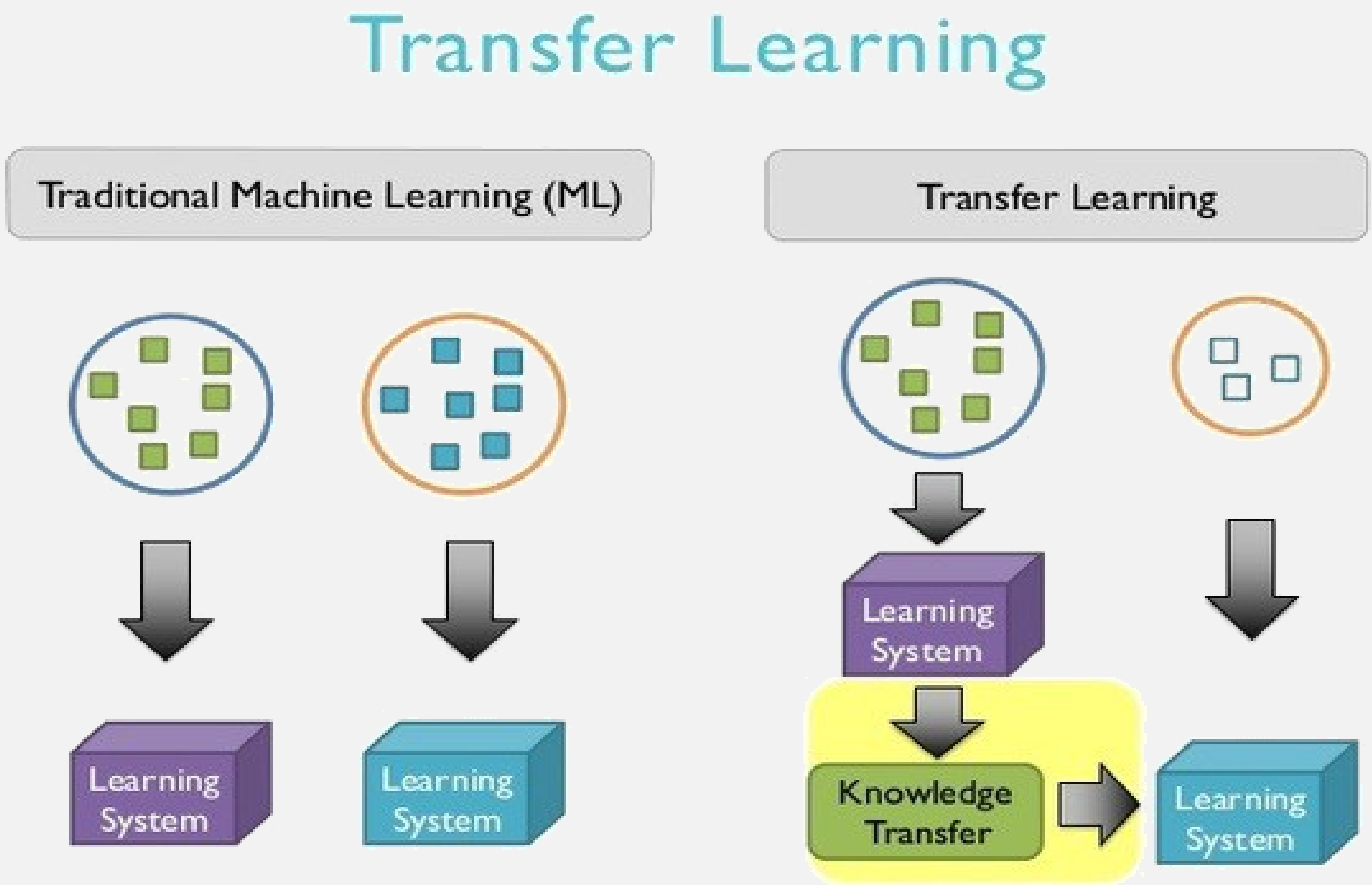
Qualitative Evaluation

Drifting(SDM & iCCR)

2.4 Deep Transfer Metric Learning

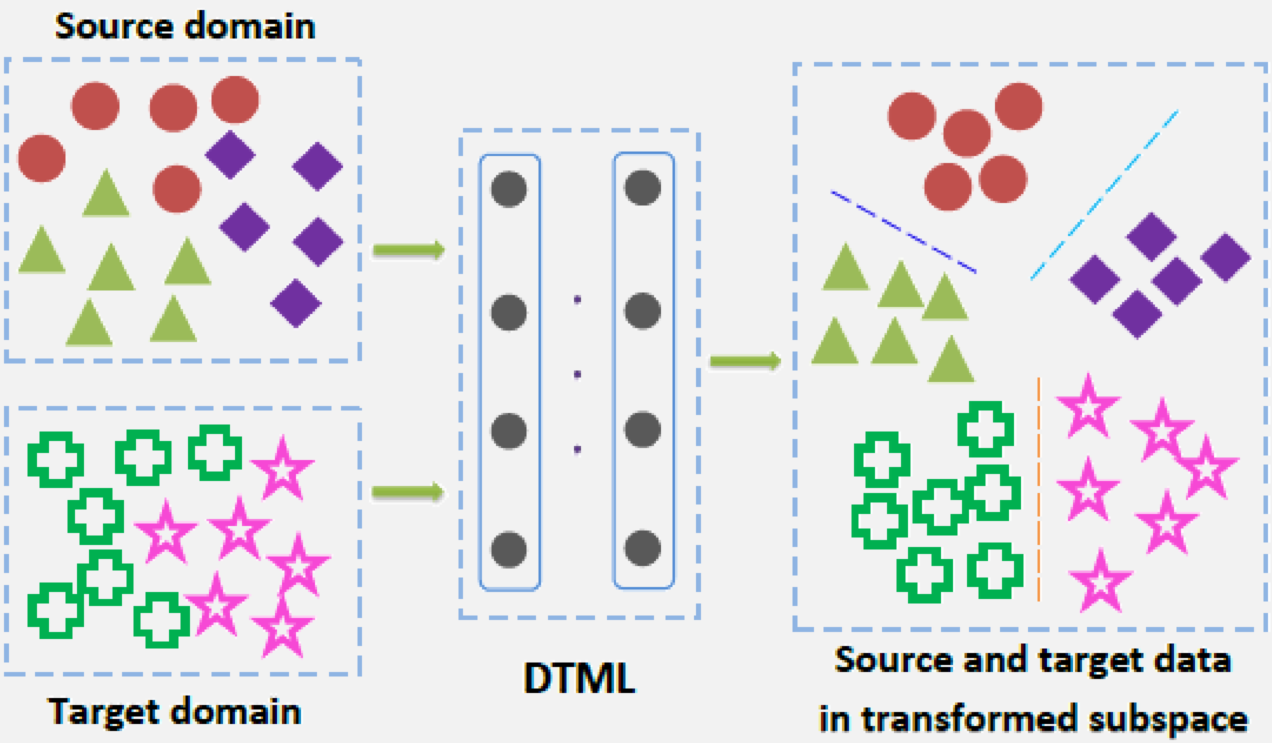
[10] Junlin Hu, **Jiwen Lu***, and Yap-Peng Tan, Deep transfer metric learning, **CVPR**, 2015.
[11] Junlin Hu, **Jiwen Lu***, Yap-Peng Tan, and Jie Zhou, Deep transfer metric learning, **T-IP**, 2016.

Deep Transfer Metric Learning



Deep Transfer Metric Learning

- Basic Idea of the proposed method



- Junlin Hu, **Jiwen Lu***, and Yap-Peng Tan, Deep transfer metric learning, **CVPR**, 2015.

Deep Transfer Metric Learning

- Formulation

$$\min_{f^{(M)}} J = S_c^{(M)} - \alpha\, S_b^{(M)} + \gamma \sum_{m=1}^{M} \left(\|\mathbf{W}^{(m)}\|_F^2 + \|\mathbf{b}^{(m)}\|_2^2 \right),$$

- Intra-class

$$S_c^{(m)} = \frac{1}{Nk_1} \sum_{i=1}^{N} \sum_{j=1}^{N} P_{ij}\, d_{f^{(m)}}^2(\mathbf{x}_i, \mathbf{x}_j),$$

- Inner-class

$$S_b^{(m)} = \frac{1}{Nk_2} \sum_{i=1}^{N} \sum_{j=1}^{N} Q_{ij}\, d_{f^{(m)}}^2(\mathbf{x}_i, \mathbf{x}_j),$$

- Maximum Mean Discrepancy

$$D_{ts}^{(m)}(\mathcal{X}_t, \mathcal{X}_s) = \left\| \frac{1}{N_t} \sum_{i=1}^{N_t} f^{(m)}(\mathbf{x}_{ti}) - \frac{1}{N_s} \sum_{i=1}^{N_s} f^{(m)}(\mathbf{x}_{si}) \right\|_2^2 .$$

Deep Transfer Metric Learning

Optimization

$$\frac{\partial J}{\partial \mathbf{W}^{(m)}} = \frac{2}{Nk_1}\sum_{i=1}^{N}\sum_{j=1}^{N} P_{ij}\left(\mathbf{L}_{ij}^{(m)}\mathbf{h}_i^{(m-1)^T} + \mathbf{L}_{ji}^{(m)}\mathbf{h}_j^{(m-1)^T}\right) - \frac{2\alpha}{Nk_2}\sum_{i=1}^{N}\sum_{j=1}^{N} Q_{ij}\left(\mathbf{L}_{ij}^{(m)}\mathbf{h}_i^{(m-1)^T} + \mathbf{L}_{ji}^{(m)}\mathbf{h}_j^{(m-1)^T}\right) + 2\beta\left(\frac{1}{N_t}\sum_{i=1}^{N_t}\mathbf{L}_{ti}^{(m)}\mathbf{h}_{ti}^{(m-1)^T} + \frac{1}{N_s}\sum_{i=1}^{N_s}\mathbf{L}_{si}^{(m)}\mathbf{h}_{si}^{(m-1)^T}\right) + 2\gamma\,\mathbf{W}^{(m)},$$

$$\frac{\partial J}{\partial \mathbf{b}^{(m)}} = \frac{2}{Nk_1}\sum_{i=1}^{N}\sum_{j=1}^{N} P_{ij}\left(\mathbf{L}_{ij}^{(m)} + \mathbf{L}_{ji}^{(m)}\right) - \frac{2\alpha}{Nk_2}\sum_{i=1}^{N}\sum_{j=1}^{N} Q_{ij}\left(\mathbf{L}_{ij}^{(m)} + \mathbf{L}_{ji}^{(m)}\right) + 2\beta\left(\frac{1}{N_t}\sum_{i=1}^{N_t}\mathbf{L}_{ti}^{(m)} + \frac{1}{N_s}\sum_{i=1}^{N_s}\mathbf{L}_{si}^{(m)}\right) + 2\gamma\,\mathbf{b}^{(m)},$$

$$\mathbf{L}_{ij}^{(M)} = \left(\mathbf{h}_i^{(M)} - \mathbf{h}_j^{(M)}\right)\odot\varphi'\left(\mathbf{z}_i^{(M)}\right),$$

$$\mathbf{L}_{ji}^{(M)} = \left(\mathbf{h}_j^{(M)} - \mathbf{h}_i^{(M)}\right)\odot\varphi'\left(\mathbf{z}_j^{(M)}\right),$$

$$\mathbf{L}_{ij}^{(m)} = \left(\mathbf{W}^{(m+1)^T}\mathbf{L}_{ij}^{(m+1)}\right)\odot\varphi'\left(\mathbf{z}_i^{(m)}\right),$$

$$\mathbf{L}_{ji}^{(m)} = \left(\mathbf{W}^{(m+1)^T}\mathbf{L}_{ji}^{(m+1)}\right)\odot\varphi'\left(\mathbf{z}_j^{(m)}\right),$$

$$\mathbf{L}_{ti}^{(M)} = \left(\frac{1}{N_t}\sum_{j=1}^{N_t}\mathbf{h}_{tj}^{(M)} - \frac{1}{N_s}\sum_{j=1}^{N_s}\mathbf{h}_{sj}^{(M)}\right)\odot\varphi'\left(\mathbf{z}_{ti}^{(M)}\right),$$

$$\mathbf{L}_{si}^{(M)} = \left(\frac{1}{N_s}\sum_{j=1}^{N_s}\mathbf{h}_{sj}^{(M)} - \frac{1}{N_t}\sum_{j=1}^{N_t}\mathbf{h}_{tj}^{(M)}\right)\odot\varphi'\left(\mathbf{z}_{si}^{(M)}\right),$$

$$\mathbf{L}_{ti}^{(m)} = \left(\mathbf{W}^{(m+1)^T}\mathbf{L}_{ti}^{(m+1)}\right)\odot\varphi'\left(\mathbf{z}_{ti}^{(m)}\right),$$

$$\mathbf{L}_{si}^{(m)} = \left(\mathbf{W}^{(m+1)^T}\mathbf{L}_{si}^{(m+1)}\right)\odot\varphi'\left(\mathbf{z}_{si}^{(m)}\right),$$

Deep Transfer Metric Learning

Iteration

$$\mathbf{W}^{(m)} = \mathbf{W}^{(m)} - \lambda \frac{\partial J}{\partial \mathbf{W}^{(m)}},$$
$$\mathbf{b}^{(m)} = \mathbf{b}^{(m)} - \lambda \frac{\partial J}{\partial \mathbf{b}^{(m)}},$$

Algorithm 1: DTML

Input: Training set: labeled source domain data $\mathcal{X}_s$ and unlabeled target domain data $\mathcal{X}_t$;
Parameters: α, β, γ, M, k_1, k_2, learning rate λ, convergence error ε, and total iterative number T.

for $k = 1, 2, \cdots, T$ **do**
- Do forward propagation to all data points;
- Compute compactness $S_c^{(M)}$ by $\boxed{4}$;
- Compute separability $S_b^{(M)}$ by $\boxed{5}$;
- Obtain MMD term $D_{ts}^{(M)}(\mathcal{X}_t, \mathcal{X}_s)$ by $\boxed{6}$;
- **for** $m = M, M-1, \cdots, 1$ **do**
  - Compute $\partial J / \partial \mathbf{W}^{(m)}$ and $\partial J / \partial \mathbf{b}^{(m)}$ by back-propagation using $\boxed{8}$ and $\boxed{9}$;
- **end**
- *// Updating weights and biases*
- **for** $m = 1, 2, \cdots, M$ **do**
  - $\mathbf{W}^{(m)} \longleftarrow \mathbf{W}^{(m)} - \lambda\ \partial J / \partial \mathbf{W}^{(m)}$;
  - $\mathbf{b}^{(m)} \longleftarrow \mathbf{b}^{(m)} - \lambda\ \partial J / \partial \mathbf{b}^{(m)}$;
- **end**
- $\lambda \longleftarrow 0.95 \times \lambda$; *// Reducing the learning rate*
- Obtain J_k by $\boxed{7}$;
- If $|J_k - J_{k-1}| < \varepsilon$, go to **Output**.

end

Output: Weights and biases $\{\mathbf{W}^{(m)}, \mathbf{b}^{(m)}\}_{m=1}^{M}$.

Deep Supervised Transfer Metric Learning

Objective function

$$\min_{f^{(M)}} J = J^{(M)} + \sum_{m=1}^{M-1} \omega^{(m)} \, h\big(J^{(m)} - \tau^{(m)}\big)$$

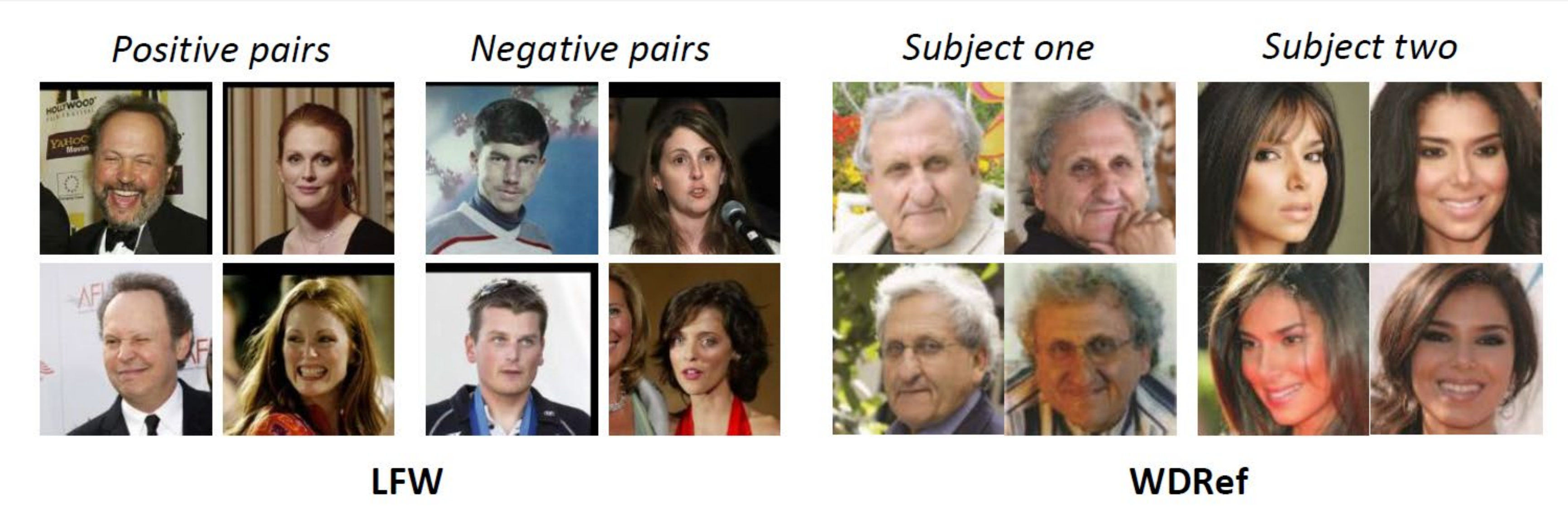
$$J^{(m)} = S_c^{(m)} - \alpha \, S_b^{(m)} + \beta \, D_{ts}^{(m)}(\mathcal{X}_t, \mathcal{X}_s) + \gamma \left(\|\mathbf{W}^{(m)}\|_F^2 + \|\mathbf{b}^{(m)}\|_2^2 \right),$$

Motivation: DTML considers supervised information at the top layer of the network, and ignores discriminative information of the outputs at the hidden layers. To better exploit such information, DSTML considers outputs of all layers to learn the deep metric network.

- Junlin Hu, **Jiwen Lu***, and Yap-Peng Tan, Deep transfer metric learning, **TIP**, 2016.

Experimental Results

☐ Cross-Dataset Face Verification

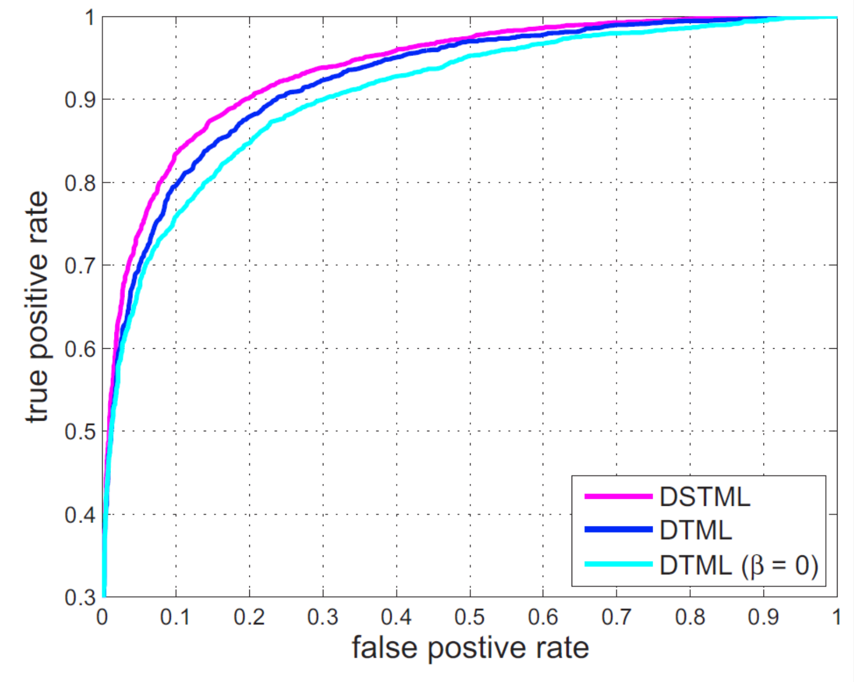


- ✓ Feature representation: LBP
- ✓ Target domain: Labeled Faces in the Wild (LFW)
- ✓ Source Domain: Wide and Deep Reference (WDRef)

Experimental Results

Method	Transfer	Accuracy (%)
DDML [16]	*no*	83.16 ± 0.80
STML	*yes*	83.60 ± 0.75
STML ($\beta = 0$)	*no*	82.57 ± 0.81
DTML	*yes*	85.58 ± 0.61
DTML ($\beta = 0$)	*no*	83.80 ± 0.55
DSTML	*yes*	$\mathbf{87.32 \pm 0.67}$

Verification rate (%) of different methods.



ROC curves of different methods.

Experimental Results

Cross-Dataset Person Re-identification

- ✓ Feature representation: LBP and color histogram
- ✓ Datasets: VIPER, i-LIDS, CAVIAR, 3DPeS

Experimental Results

Method	Source	$r = 1$	$r = 5$	$r = 10$	$r = 30$
L_1	-	3.99	8.73	12.59	25.32
L_2	-	4.24	8.92	12.66	25.35
DDML [16]	i-LIDS	5.63	12.91	21.71	41.80
	CAVIAR	5.91	13.53	19.86	37.92
	3DPeS	6.67	17.16	23.87	41.65
DTML ($\beta = 0$)	i-LIDS	5.88	13.72	21.03	41.49
	CAVIAR	6.02	13.81	20.33	38.46
	3DPeS	7.20	18.04	25.96	43.80
DTML	i-LIDS	6.68	15.73	23.20	46.42
	CAVIAR	6.17	13.10	19.65	37.78
	3DPeS	8.51	**19.40**	**27.59**	**47.91**
DSTML	i-LIDS	6.11	16.01	23.51	45.35
	CAVIAR	6.61	16.93	24.40	41.55
	3DPeS	**8.58**	19.02	26.49	46.77

Top r matched results of different methods on the VIPeR dataset

Experimental Results

Method	Source	$r=1$	$r=5$	$r=10$	$r=30$
L_1	-	20.65	36.44	48.52	88.34
L_2	-	20.19	36.43	48.55	87.69
DDML [16]	VIPeR	23.80	42.15	55.61	90.73
	i-LIDS	22.72	41.36	56.92	90.06
	3DPeS	23.85	44.30	57.81	90.27
DTML ($\beta=0$)	VIPeR	23.71	42.57	56.15	90.55
	i-LIDS	23.09	42.81	58.43	90.41
	3DPeS	25.11	46.71	59.69	91.99
DTML	VIPeR	23.88	42.36	55.60	92.12
	i-LIDS	26.06	47.37	61.70	**94.23**
	3DPeS	26.10	47.80	61.31	93.02
DSTML	VIPeR	26.05	44.33	57.02	92.80
	i-LIDS	25.91	44.47	58.88	93.33
	3DPeS	**28.18**	**49.96**	**63.67**	94.13

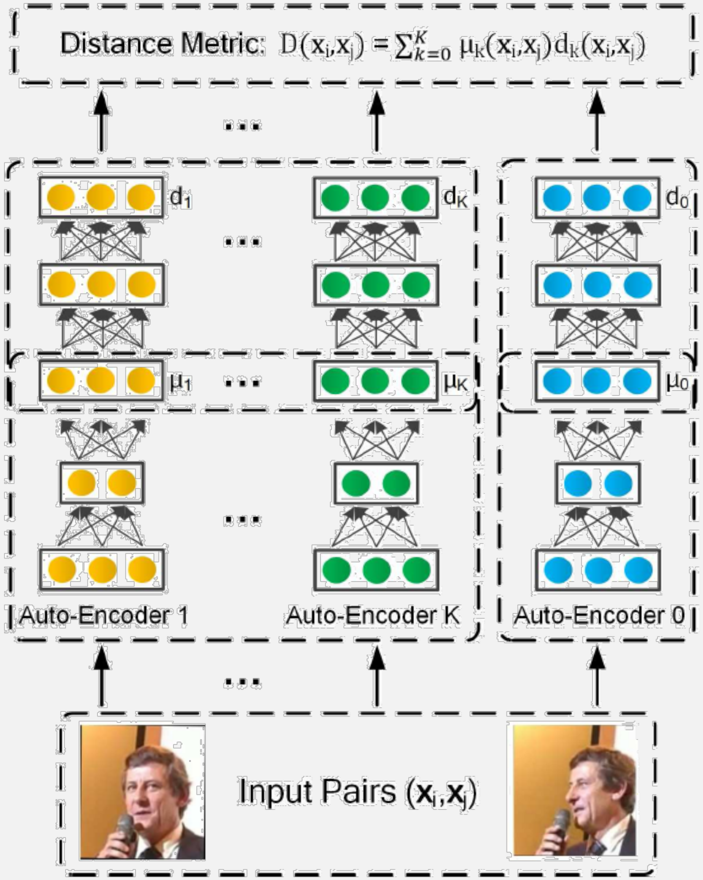
Top r matched results of different methods on the CAVIAR dataset

2.5 Deep Multi-Metric Learning

[12] Yueqi Duan, **Jiwen Lu***, Jianjiang Feng, and Jie Zhou, Deep localized metric learning, **T-CSVT**, 2018, accepted.
[13] Junlin Hu, **Jiwen Lu***, and Yap-Peng Tan, Sharable and individual multi-view metric learning, **T-PAMI**, 2018.

Deep Localized Metric Learning

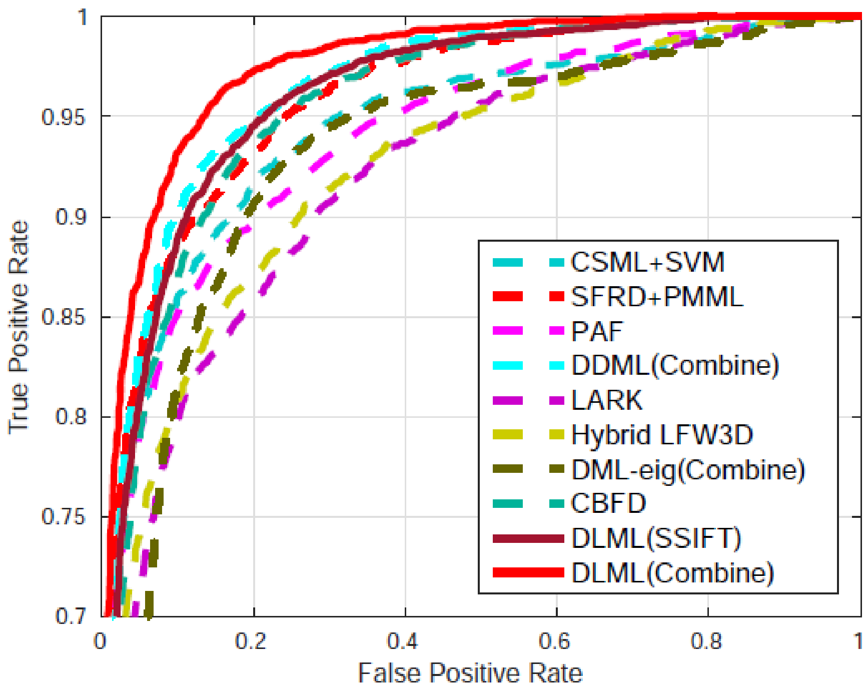
- Motivation



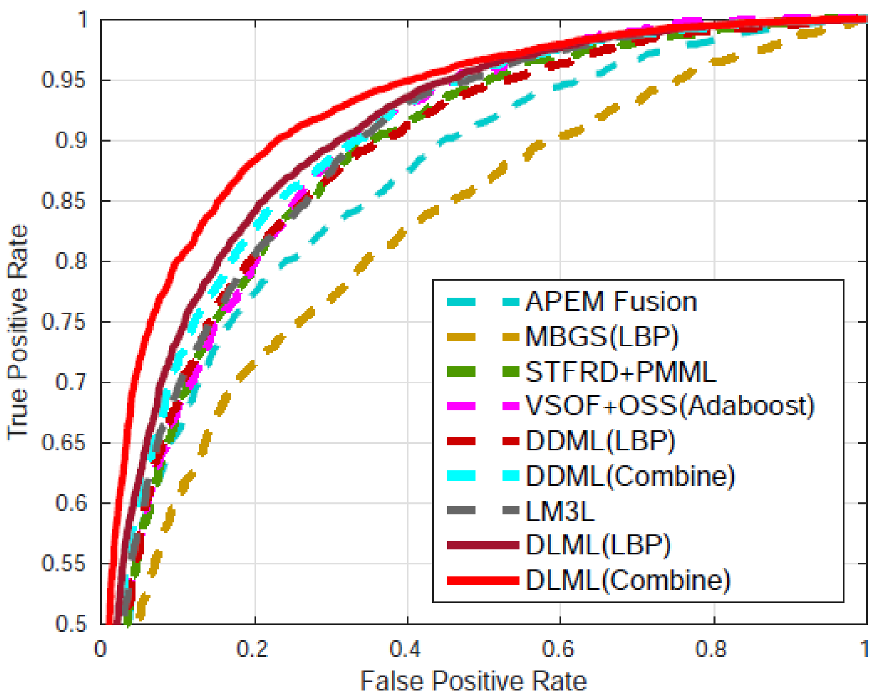
- Yueqi Duan, **Jiwen Lu*,** Jianjiang Feng, and Jie Zhou, Deep localized metric learning, **TCSVT**, 2018, accepted.

Quantitative Curves

- LFW



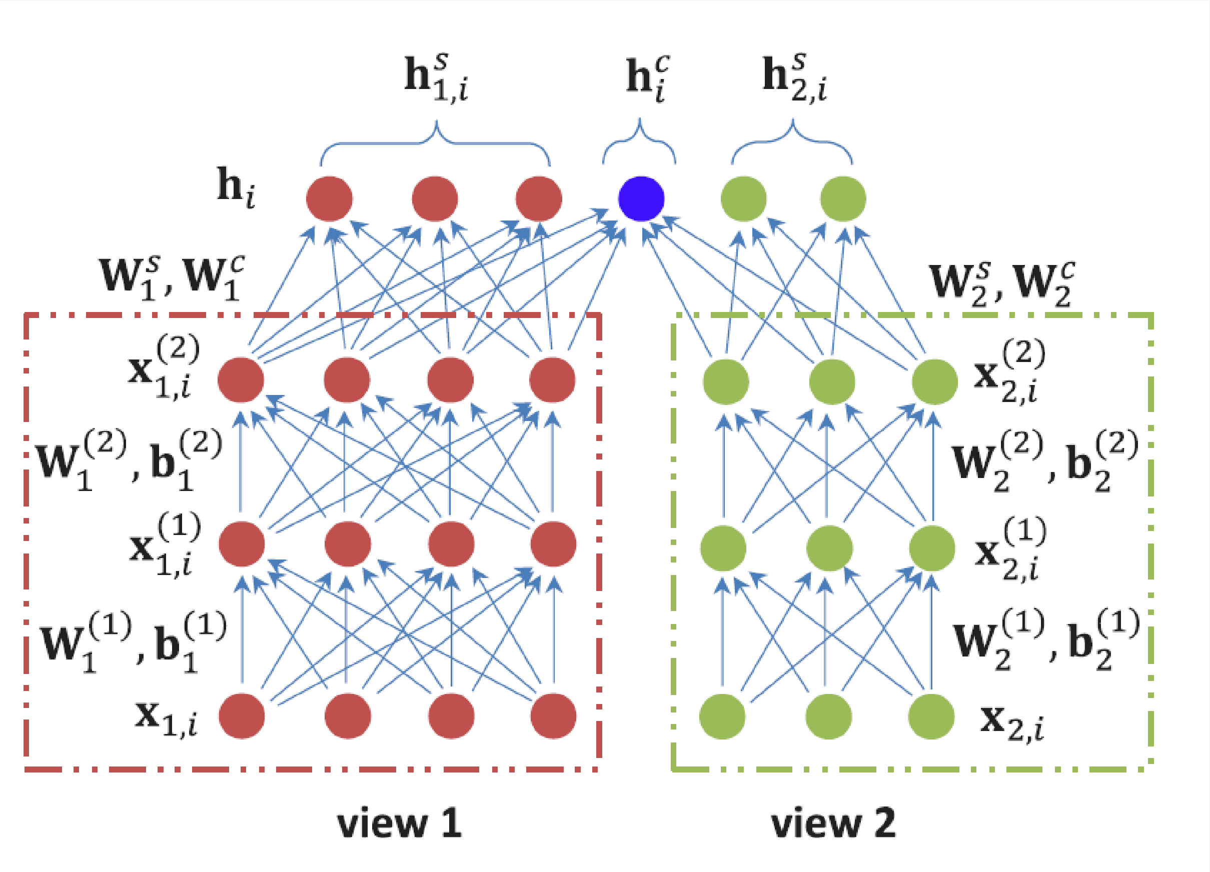
- YTF



- Yueqi Duan, **Jiwen Lu*,** Jianjiang Feng, and Jie Zhou, Deep localized metric learning, **TCSVT**, 2018, accepted.

Sharable and Individual Deep Metric Learning

Enhanced Idea



Sharable and Individual Deep Metric Learning

Formulation

$$
\begin{aligned}
d^2_{\boldsymbol{\Theta}}(\mathbf{x}_i, \mathbf{x}_j) &= \left\|\mathbf{h}_i - \mathbf{h}_j\right\|_2^2 \\
&= \sum_{\kappa=1}^{K} \left\|\mathbf{h}^s_{\kappa,i} - \mathbf{h}^s_{\kappa,j}\right\|_2^2 + \left\|\mathbf{h}^c_i - \mathbf{h}^c_j\right\|_2^2 \\
&= \sum_{\kappa=1}^{K} \left\|\mathbf{W}^s_\kappa\Big(f_\kappa(\mathbf{x}_{\kappa,i}) - f_\kappa(\mathbf{x}_{\kappa,j})\Big)\right\|_2^2 \\
&\quad + \left\|\frac{1}{K}\sum_{\kappa=1}^{K} \mathbf{W}^c_\kappa\Big(f_\kappa(\mathbf{x}_{\kappa,i}) - f_\kappa(\mathbf{x}_{\kappa,j})\Big)\right\|_2^2,
\end{aligned}
$$

$$
\begin{aligned}
\min_{\boldsymbol{\Theta}} \; J = & \frac{1}{|\mathcal{S}|}\sum_{(i,j)\in\mathcal{S}} [d^2_{\boldsymbol{\Theta}}(\mathbf{x}_i, \mathbf{x}_j) - \tau_s]_+ \\
& + \frac{1}{|\mathcal{D}|}\sum_{(i,j)\in\mathcal{D}} [\tau_d - d^2_{\boldsymbol{\Theta}}(\mathbf{x}_i, \mathbf{x}_j)]_+ \\
& + \lambda \sum_{\kappa=1}^{K}\Big(\|\mathbf{W}^s_\kappa\|_F^2 + \|\mathbf{W}^c_\kappa\|_F^2\Big) \\
& + \lambda \sum_{\kappa=1}^{K}\sum_{m=1}^{M_\kappa}\Big(\|\mathbf{W}^{(m)}_\kappa\|_F^2 + \|\mathbf{b}^{(m)}_\kappa\|_2^2\Big).
\end{aligned}
$$

- Junlin Hu, **Jiwen Lu*,** and Yap-Peng Tan, Sharable and individual multi-view metric learning, **TPAMI**, 2018.

Experimental Results

- Face verification

Feature	SvML	SvDML
HOG	86.77 ± 0.54	87.27 ± 0.72
LBP	84.90 ± 0.48	85.70 ± 0.41
SIFT	85.00 ± 0.28	86.57 ± 0.39
Con.	87.45 ± 0.46	88.03 ± 0.39
Feature	MvML-s	MvDML-s
HOG, LBP, SIFT	87.52 ± 0.42	88.15 ± 0.35
Feature	MvML-c	MvDML-c
HOG, LBP, SIFT	80.75 ± 0.56	81.61 ± 0.50
Feature	MvML	MvDML
HOG, LBP, SIFT	$\mathbf{88.58 \pm 0.36}$	$\mathbf{90.23 \pm 0.53}$

Experimental Results

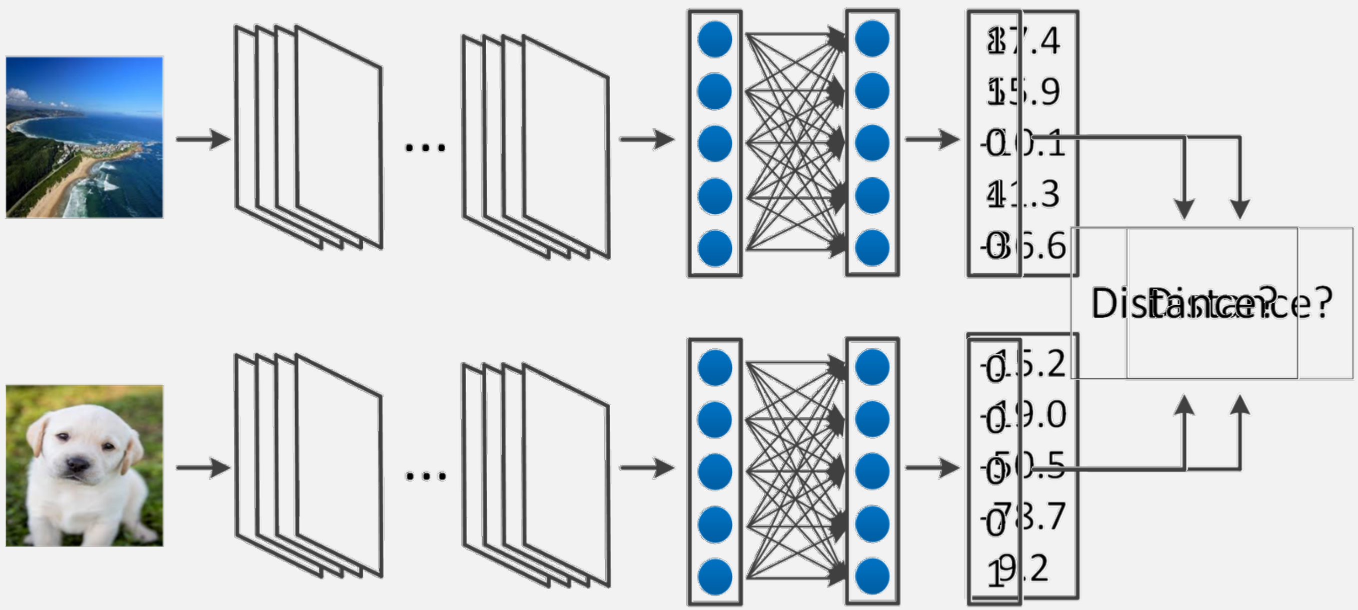
- Kinship verification

Method	Accuracy (%)
CSML+SVM, aligned [7]	88.00 ± 0.37
SFRD+PMML [18]	89.35 ± 0.50
Sub-SML [29]	89.73 ± 0.38
VMRS [30]	91.10 ± 0.59
DDML [21]	90.68 ± 1.41
LM^3L [19]	89.57 ± 1.53
Sub-SML + Hybrid on LFW3D [27]	91.65 ± 1.04
HPEN + HD-LBP + DDML [28]	92.57 ± 0.36
HPEN + HD-Gabor + DDML [28]	92.80 ± 0.47
MvML (+HDLBP)	91.37 ± 0.29
MvDML (+HDLBP)	$\mathbf{93.27 \pm 0.28}$

Part 3: Hamming Deep Metric Learning

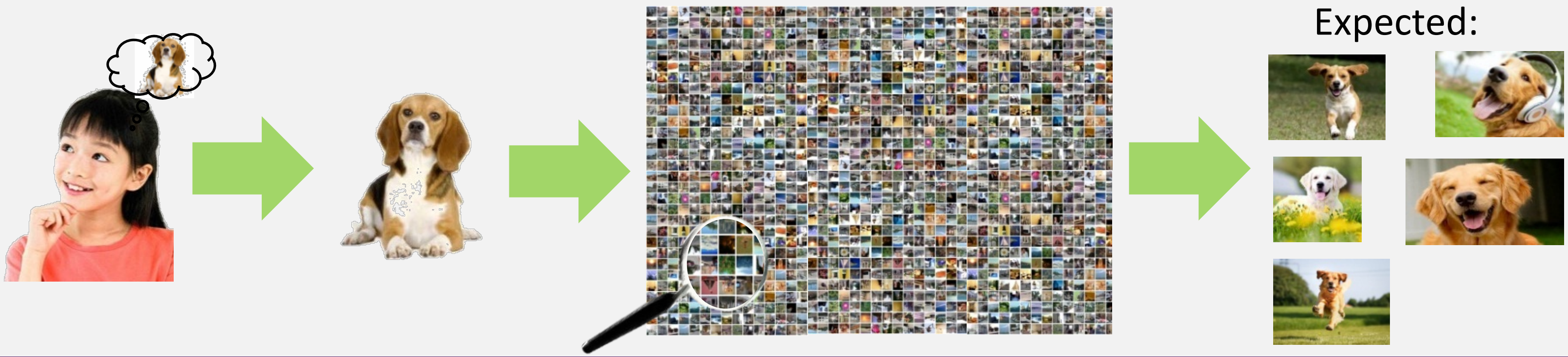
What is Hamming DML?

- Mahalanobis deep metric learning
  - Input → Deep neural network → Real-valued embedding
- Hamming deep metric learning
  - Input → Deep neural network → Binary embedding

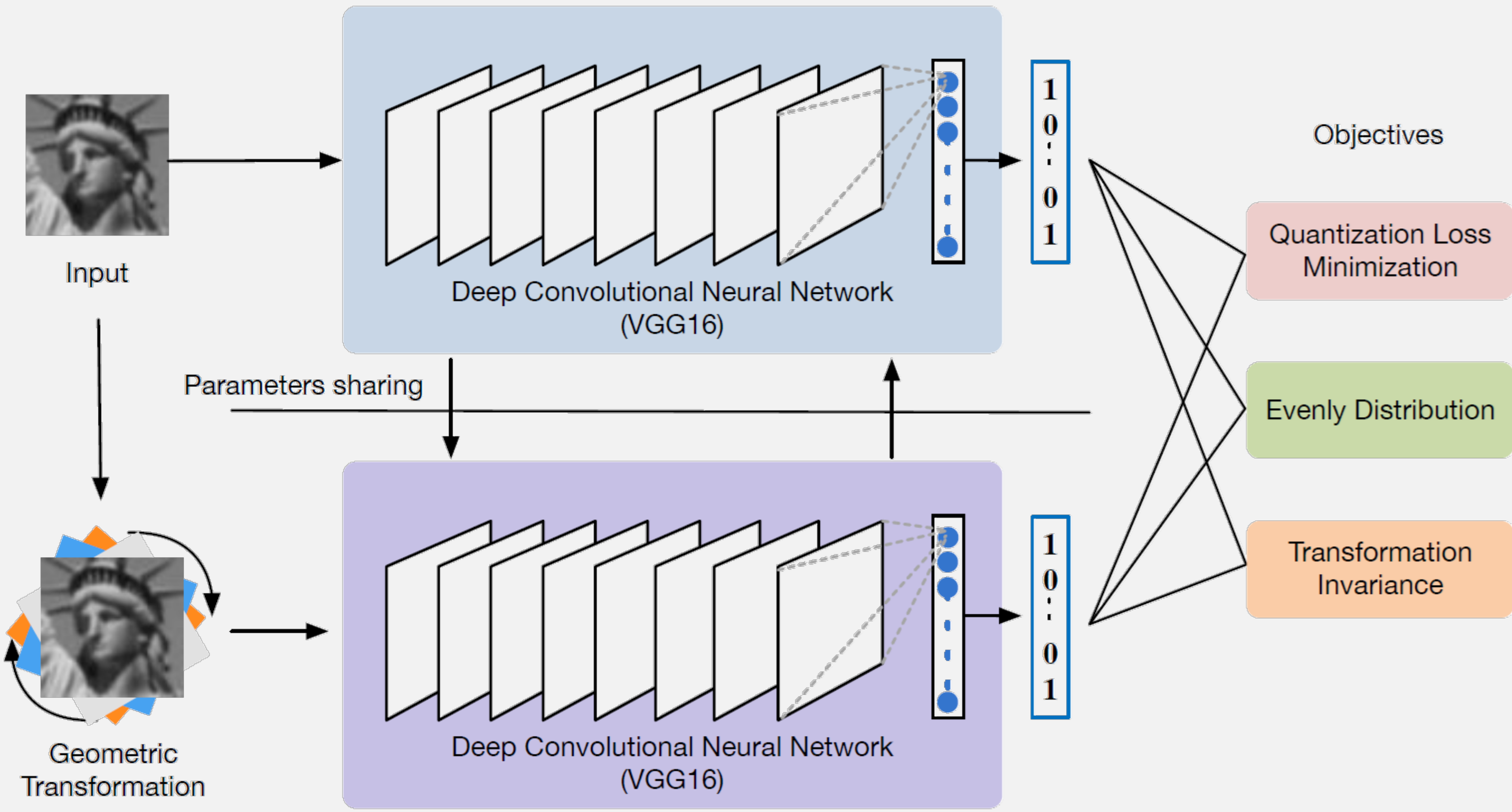


Why Binary?

- Considering an online image searching system:
  - Offline: training model, gallery features extraction, storage
  - Online: probe feature extraction, matching
  - Hamming DML presents high storage efficiency and matching speed
- Lightweight models: efficient for training and feature extraction
- Heavyweight models: strong discriminative power



Unsupervised Hamming DML



- Kevin Lin, **Jiwen Lu***, Chu-Song Chen, Jie Zhou, Learning Compact Binary Descriptors with Unsupervised Deep Neural Networks, **CVPR**, 2016.

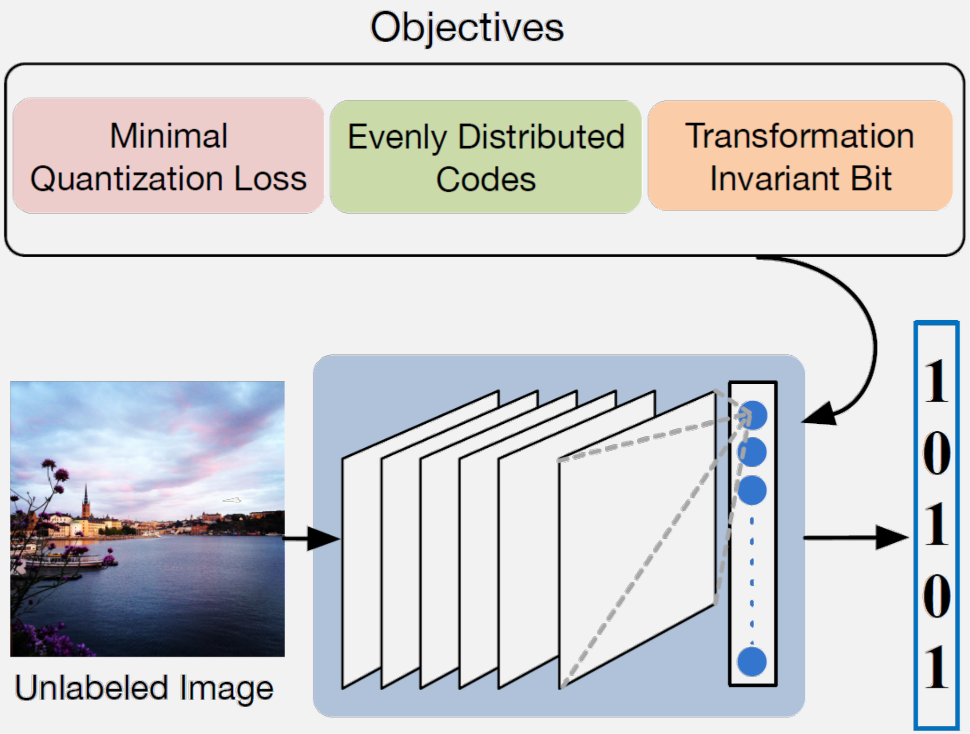
Unsupervised Hamming DML

- Objective function

$$
\begin{aligned}
\min_{\mathcal{W}} L(\mathcal{W}) = \alpha L_1(\mathcal{W}) + \beta L_2(\mathcal{W}) + \gamma L_3(\mathcal{W}) \\
= \alpha \sum_{n=1}^{N} ||(b_n - 0.5) - \mathcal{F}(x_n; \mathcal{W})||^2 \\
+ \beta \sum_{m=1}^{M} ||(\mu_m - 0.5)||^2 \\
+ \gamma \sum_{n=1}^{N} \sum_{\theta=-R}^{R} \mathcal{C}(\theta) ||b_{n,\theta} - b_n||^2,
\end{aligned}
$$

Unsupervised Hamming DML

- Learning transformation-invariant bits



- Kevin Lin, **Jiwen Lu***, Chu-Song Chen, Jie Zhou, Ming-Ting Sun, Unsupervised Deep Learning of Compact Binary Descriptors, **TPAMI**, 2018.

Unsupervised Hamming DML

- Objective function

$$
\begin{aligned}
\min_W E(W) &= \alpha E_1(W) + \beta E_2(W) + \gamma E_3(W) \\
&= \alpha \sum_{k=1}^{K} \sum_{n=1}^{N} ||b_{nk} - \mathcal{F}_k(x_n; W_k)||^2 \\
&+ \beta \sum_{k=1}^{K} ||\mu_k - 0.5||^2 \\
&+ \gamma \sum_{k=1}^{K} \sum_{n=1}^{N} \{(y)d + (1-y)(K-d)\}
\end{aligned}
$$

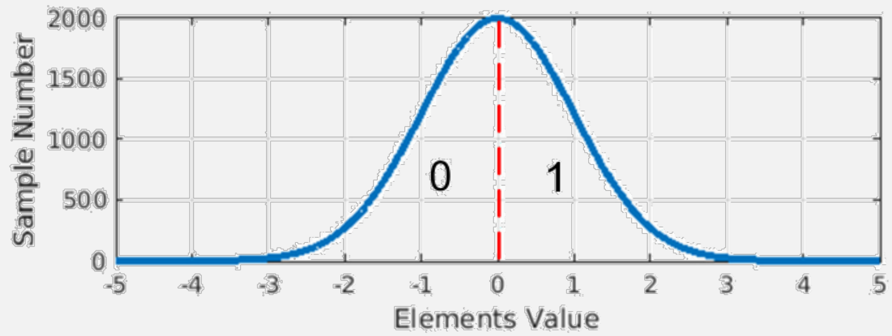
Experimental Results

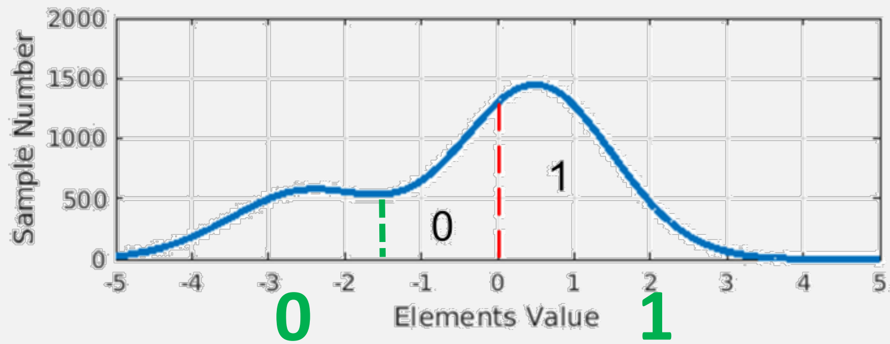
- The CIFAR-10 dataset

Method	16 bit	32 bit	64 bit
GIST + SpeH [30]	12.55	12.42	12.56
GIST + SH [29]	12.95	14.09	13.89
GIST + PCAH [60]	12.91	12.60	12.10
GIST + LSH [27]	12.55	13.76	15.07
GIST + PCA-ITQ [28]	15.67	16.20	16.64
VGG16 + LSH	10.67	10.57	10.03
VGG16 + PCA-ITQ	20.97	21.74	22.32
DH [45]	16.17	16.62	16.96
Huang *et al.* [44]	16.82	17.01	17.21
UH-BDNN [43]	17.83	18.52	-
Ours	**21.70**	**20.64**	**23.07**

Quantization Loss Minimization

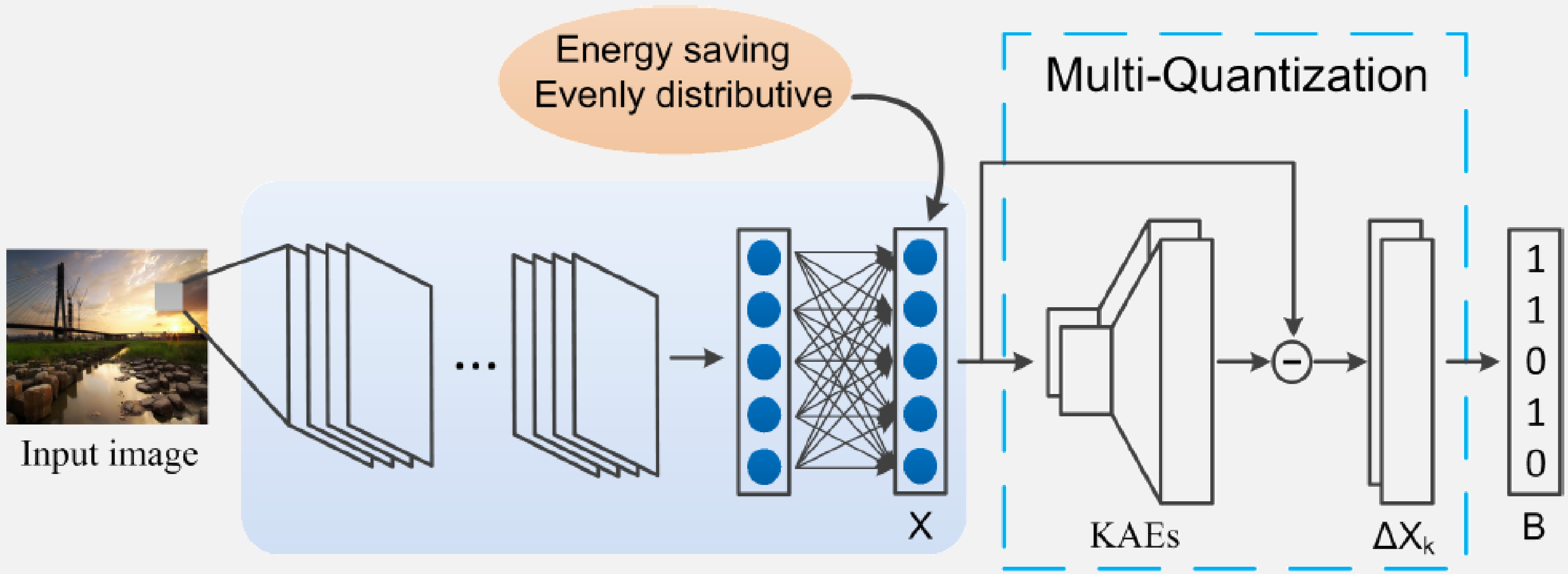
- Learning data-dependent binarization





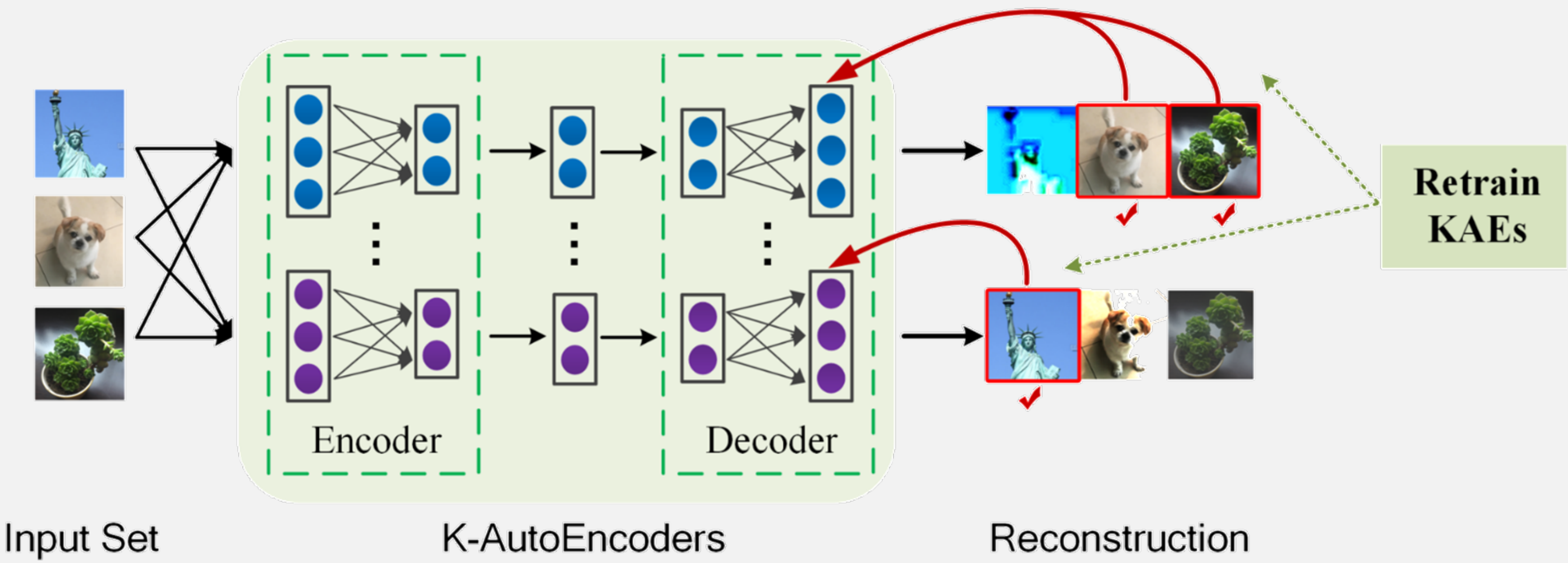
- Yueqi Duan, **Jiwen Lu***, Ziwei Wang, Jianjiang Feng, Jie Zhou, Learning Deep Binary Descriptor with Multi-Quantization, **CVPR**, 2017.

Quantization Loss Minimization



Quantization Loss Minimization

- Iteratively perform two steps:
  - Associate each image with an Autoencoder
  - Retrain KAEs with the corresponding images



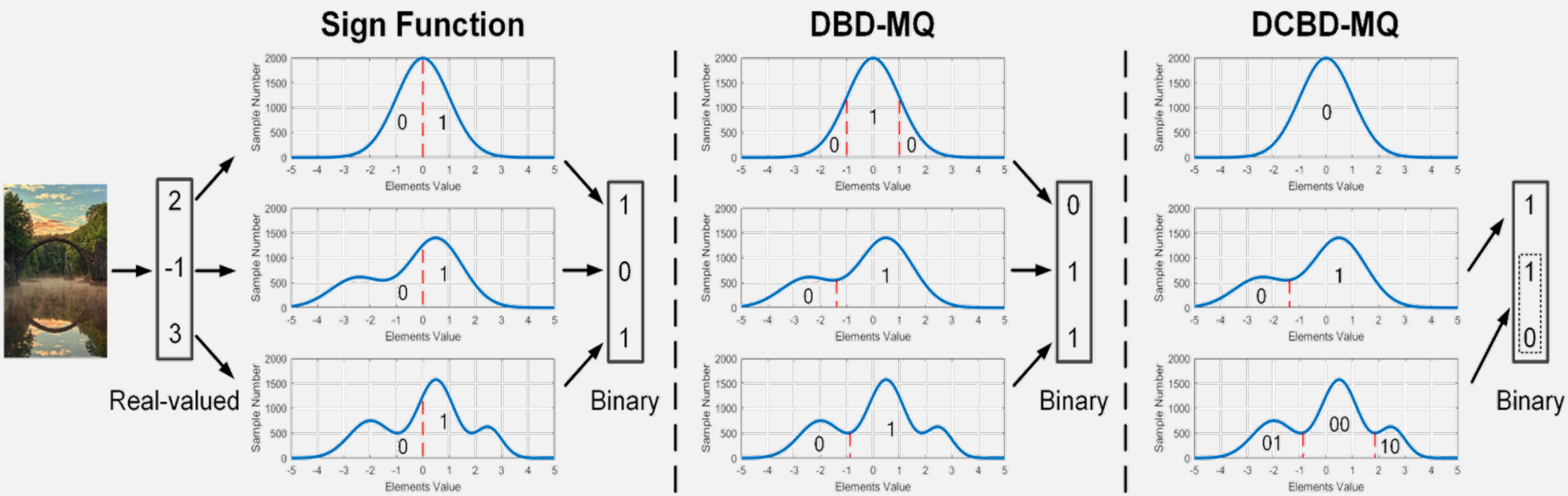
Quantization Loss Minimization

- Objective function
  - Reconstruction error minimization, regularization, large variations

$$
\begin{aligned}
\min_{\mathbf{X},\mathbf{W}_k} J &= J_1 + \lambda_1 J_2 + \lambda_2 J_3 \\
&= \sum_{n=1}^{N} \varepsilon_{nk_n}^2 + \lambda_1 \sum_{k=1}^{K} \sum_{l} ||\mathbf{W}_k^{(l)}||_F^2 \\
&- \lambda_2 \mathrm{tr}((\mathbf{X}-\mathbf{U})^T(\mathbf{X}-\mathbf{U}))
\end{aligned}
$$

Quantization Loss Minimization

- Competition in feature dimensions



- Yueqi Duan, **Jiwen Lu***, Ziwei Wang, Jianjiang Feng, Jie Zhou, Learning Deep Binary Descriptor with Multi-Quantization, **TPAMI**, 2018.

Experimental Results

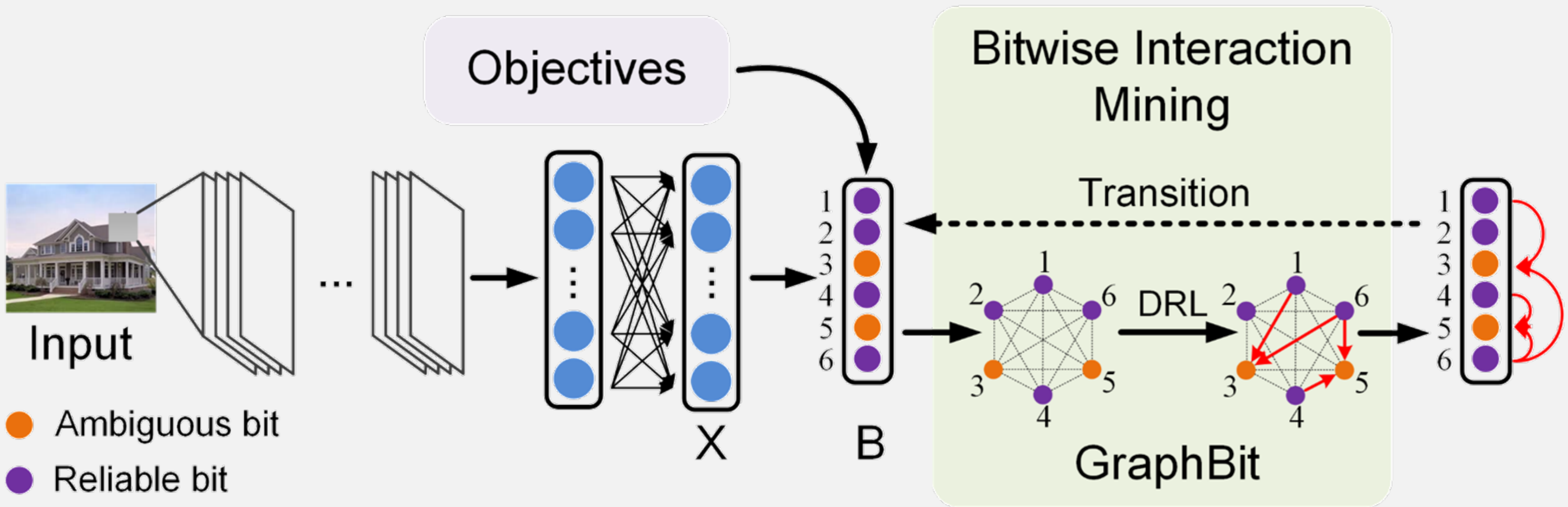
- The CIFAR-10 dataset

Method	16 bits	32 bits	64 bits
KMH [24]	13.59	13.93	14.46
SphH [26]	13.98	14.58	15.38
SpeH [71]	12.55	12.42	12.56
SH [57]	12.95	14.09	13.89
PCAH [69]	12.91	12.60	12.10
LSH [3]	12.55	13.76	15.07
PCA-ITQ [22]	15.67	16.20	16.64
DH [16]	16.17	16.62	16.96
DeepBit [39]	19.43	24.86	27.73
DBD-MQ [15]	21.53	26.50	31.85
DCBD-MQ	**30.58**	**33.01**	**36.59**

Bitwise Interaction Mining

GraphBit

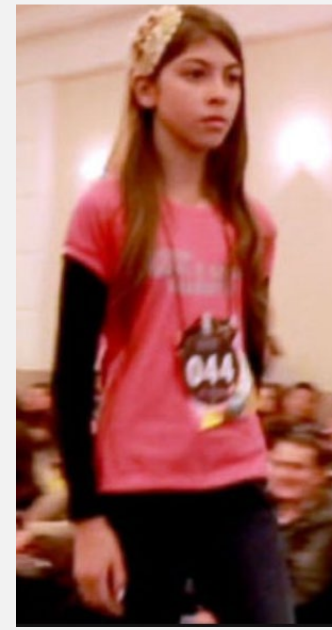
- Node: element ---- Edge: bitwise interaction

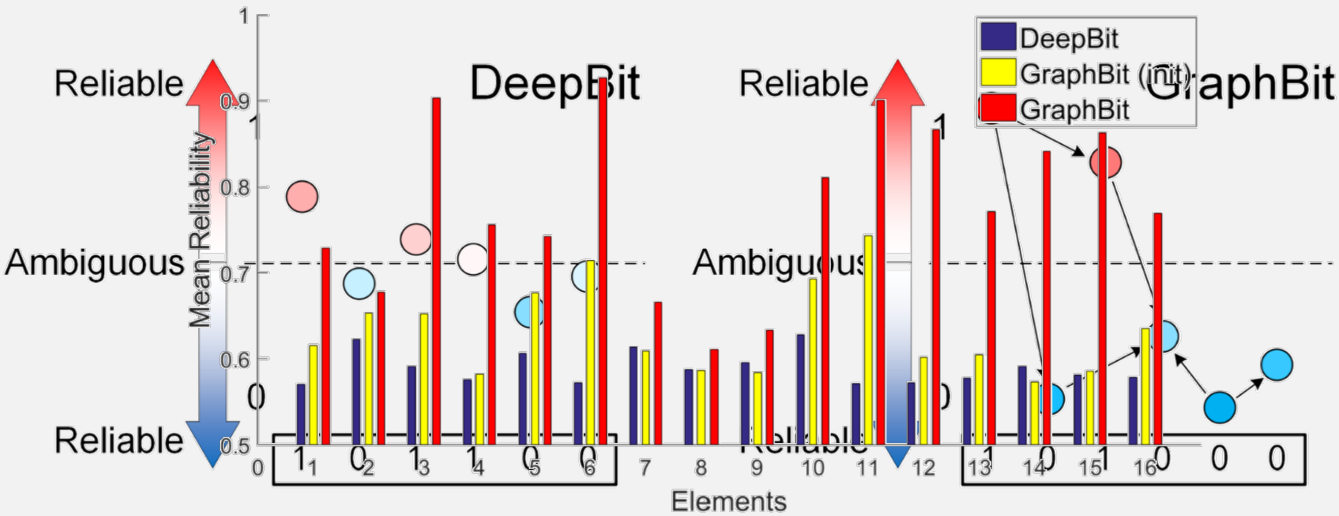


Yueqi Duan, Ziwei Wang, **Jiwen Lu***, Xudong Lin, Jie Zhou, GraphBit: Bitwise Interaction Mining via Deep Reinforcement Learning, **CVPR**, 2018.

Bitwise Interaction Mining

- A person in 5 feet 9 inches
  - Tall (1) / Short (0)? ⟵ **Ambiguous Bit**
  - Tall (1) / Short (0) ⟵ **Reliable Bit**
- Additional information from other bits
  - Male (1) / Female (0), Adult (1) / Child (0)





Bitwise Interaction Mining

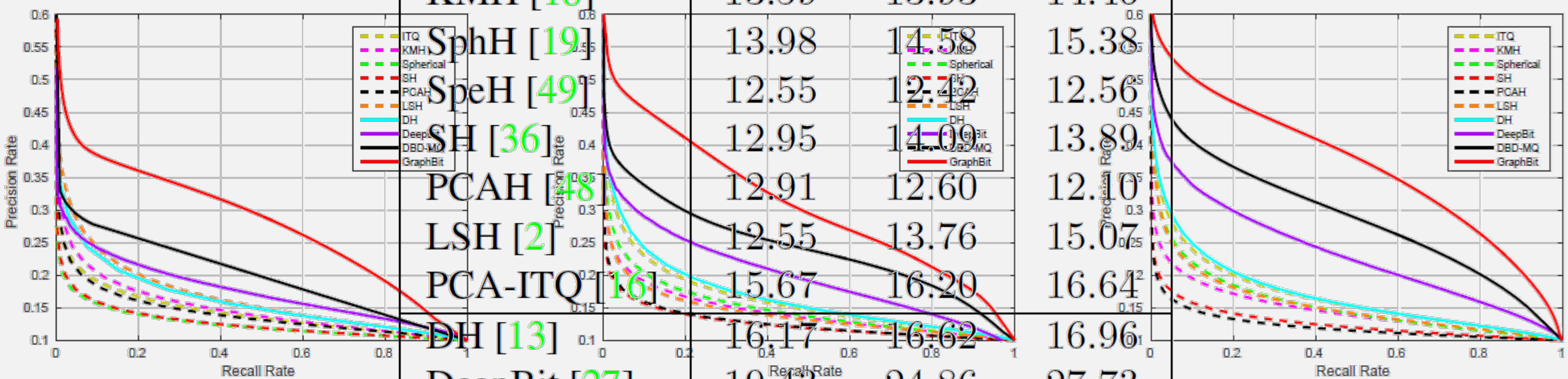
- Objective function
  - Even distribution, uncertainty minimization, independence

$$
\begin{aligned}
\min J &= J_1 + \alpha J_2 + \beta J_3 \\
&= \sum_{k=1}^{K} || \sum_{n=1}^{N} (b_{kn} - 0.5) ||^2 \\
&- \alpha \sum_{n=1}^{N} \Big(\sum_{b_{rn} \notin \mathbf{b}_s^T} I(b_{rn}; \mathbf{x}_n) + \sum_{\Phi} I(b_{sn}; \mathbf{x}_n, b_{tn}) \Big) \\
&+ \beta \sum_{n=1}^{N} \sum_{\Phi} || p(b_{sn} | \mathbf{x}_n) - p(b_{sn} | \mathbf{x}_n, b_{tn}) ||^2
\end{aligned}
$$

Experimental Results

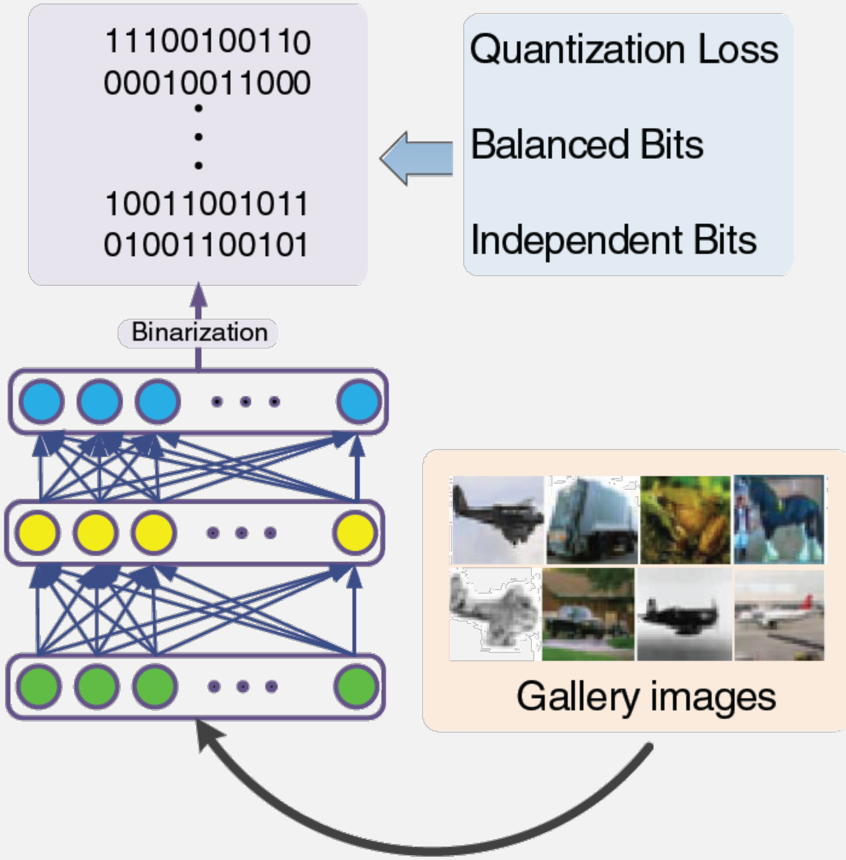
- The CIFAR-10 dataset

Method	16 bits	32 bits	64 bits
KMH [18]	13.59	13.93	14.46
SphH [19]	13.98	14.58	15.38
SpeH [49]	12.55	12.42	12.56
SH [36]	12.95	14.09	13.89
PCAH [48]	12.91	12.60	12.10
LSH [2]	12.55	13.76	15.07
PCA-ITQ [16]	15.67	16.20	16.64
DH [13]	16.17	16.62	16.96
DeepBit [27]	19.43	24.86	27.73
DBD-MQ [12]	21.53	26.50	31.85
GraphBit (init)	27.95	32.77	36.16
GraphBit	**32.15**	**36.74**	**39.90**



(a) 16 bits (b) 32 bits (c) 64 bits

Hamming DML for Image Search



$$\begin{aligned}
\arg\min_{\mathbf{W},\mathbf{c}} J &= \frac{1}{2}\|\mathbf{B}-\mathbf{H}^M\|_F^2 \\
&- \frac{\lambda_1}{2}(\mathrm{tr}(\frac{1}{N}\mathbf{H}^M(\mathbf{H}^M)^T) + \alpha\mathrm{tr}(\Sigma_B - \Sigma_W)) \\
&+ \frac{\lambda_2}{2}\sum_{m=1}^{M}\|\mathbf{W}^m(\mathbf{W}^m)^T - \mathbf{I}\|_F^2 \\
&+ \frac{\lambda_3}{2}\sum_{m=1}^{M}(\|\mathbf{W}^m\|_F^2 + \|\mathbf{c}^m\|_2^2)
\end{aligned}$$

$$\begin{aligned}
\Sigma_W &= \frac{1}{N_S}\sum_{(\mathbf{x}_i,\mathbf{x}_j)\in\mathcal{S}}(\mathbf{h}_i^M-\mathbf{h}_j^M)(\mathbf{h}_i^M-\mathbf{h}_j^M)^T \\
&= \frac{1}{N_S}\mathrm{tr}((\mathbf{H}_{s1}^M-\mathbf{H}_{s2}^M)(\mathbf{H}_{s1}^M-\mathbf{H}_{s2}^M)^T) \\
\Sigma_B &= \frac{1}{N_D}\sum_{(\mathbf{x}_i,\mathbf{x}_j)\in\mathcal{D}}(\mathbf{h}_i^M-\mathbf{h}_j^M)(\mathbf{h}_i^M-\mathbf{h}_j^M)^T \\
&= \frac{1}{N_D}\mathrm{tr}((\mathbf{H}_{d1}^M-\mathbf{H}_{d2}^M)(\mathbf{H}_{d1}^M-\mathbf{H}_{d2}^M)^T)
\end{aligned}$$

- Venice Erin Liong, **Jiwen Lu***, Gang Wang, Pierre Moulin, Jie Zhou, Deep Hashing for Compact Binary Codes Learning, **CVPR**, 2015.

Hamming DML for Image Search

- Multi-label supervision

$$\Sigma_w^{(l)} = \sum_{i=1}^{N} \delta_{il}(\mathbf{h}_i^M - \mu_l)(\mathbf{h}_i^M - \mu_l)^\top$$

$$\Sigma_b^{(l)} = \sum_{i=1}^{N} \delta_{il}(\mu_l - \mu)(\mu_l - \mu)^\top$$

$$\Sigma_w = \sum_{l=1}^{L} \Sigma_w^{(l)}$$

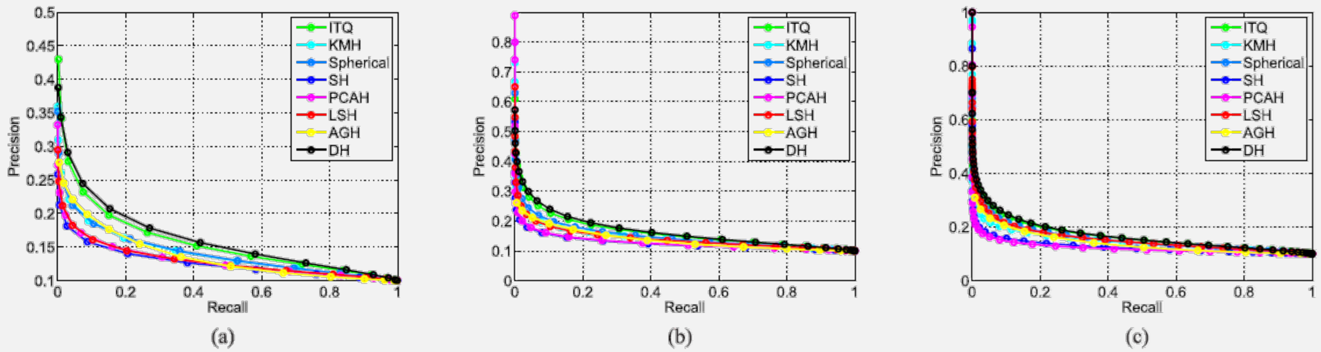
$$\Sigma_b = \sum_{l=1}^{L} \Sigma_b^{(l)}$$

- **Jiwen Lu**, Venice Erin Liong, Jie Zhou, Deep Hashing for Scalable Image Search, **TIP**, 2017.

Experimental Results

☐ The CIFAR-10 dataset

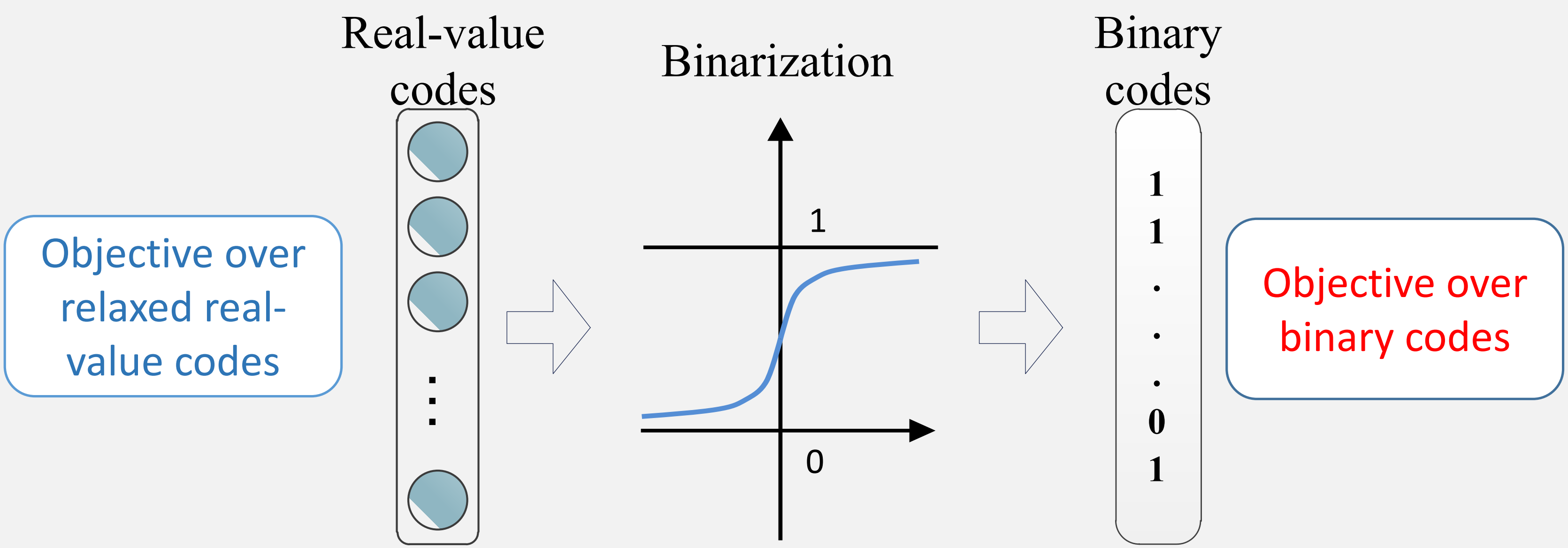
Method	Hamming ranking (mAP, %)			precision (%) @ sample = 500			precision (%) @ r=2	
	16	32	64	16	32	64	16	32
PCA-ITQ [13]	15.67	16.20	16.64	22.46	25.30	27.09	22.60	14.99
KMH [15]	13.59	13.93	14.46	20.28	21.97	22.80	22.08	5.72
Spherical [16]	13.98	14.58	15.38	20.13	22.33	25.19	20.96	12.50
SH [81]	12.55	12.42	12.56	18.83	19.72	20.16	18.52	20.60
PCAH [74]	12.91	12.60	12.10	18.89	19.35	18.73	21.29	2.68
LSH [1]	12.55	13.76	15.07	16.21	19.10	22.25	16.73	7.07
AGH [41]	13.64	13.61	13.54	22.61	23.28	25.48	21.25	**24.53**
DH	**16.17**	**16.62**	**16.96**	**23.79**	**26.00**	**27.70**	**23.33**	15.77
SPLH [74]	17.61	20.20	20.98	25.32	29.43	32.22	23.05	30.47
MLH [48]	18.37	20.49	21.89	24.43	29.60	33.01	23.52	28.72
BRE [32]	14.42	15.14	15.88	20.68	22.86	25.14	20.89	20.29
KSH [40]	14.83	15.25	15.11	20.79	22.16	23.59	20.73	7.62
FastHash [37]	29.73	34.54	38.15	37.60	42.04	48.78	40.77	26.88
CCA-ITQ [13]	14.64	16.27	16.42	23.06	27.23	27.67	19.26	28.08
SDisH [59]	29.35	35.81	37.43	**39.48**	43.87	47.43	31.79	**42.77**
SDH	**31.01**	**35.88**	**38.50**	30.94	**47.32**	**50.95**	**69.18**	14.41



清華大学 Tsinghua University
i-VisionGroup

Discrete Hamming DML

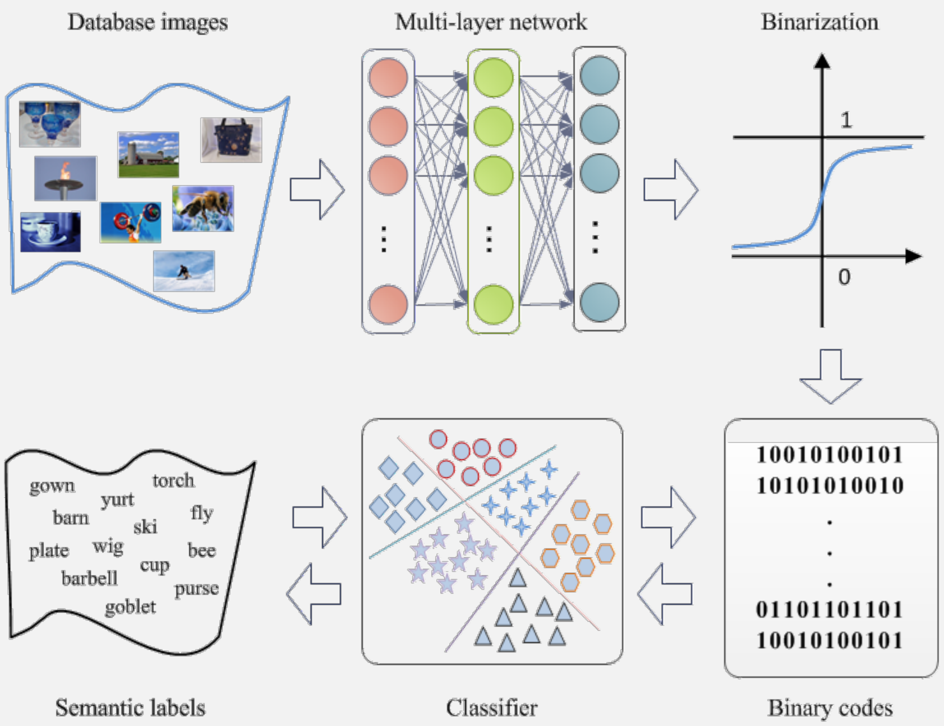
- Optimization over binary codes rather than relaxed real-value codes



- Zhixiang Chen, **Jiwen Lu***, Jianjiang Feng, Jie Zhou, Nonlinear Discrete Hashing, **TMM**, 2017.

Discrete Hamming DML

- Solve the discrete optimization problem to eliminate the quantization error accumulation
- Exploit the nonlinear relationship of samples with nonlinear hashing functions



Discrete Hamming DML

- Objective Function
  - Maximizing classification accuracy
  - Maximizing information with bit independency
  - Minimizing the quantization loss

$$\arg\min_{\boldsymbol{B},\boldsymbol{P},\{\boldsymbol{F}^{(m)}\}_{m=1}^{M},\boldsymbol{Y}} \quad \mathcal{Q} = \mathcal{Q}_{\boldsymbol{P}} + \lambda_1 \mathcal{Q}_I + \lambda_2 \mathcal{Q}_{\boldsymbol{F}} + \lambda_3 \mathcal{Q}_R$$

$$s.t. \quad \boldsymbol{B} \in \{-1, 1\}^{n\times r}$$

Discrete Hamming DML

- Optimization
- Bit independency

$$\mathcal{Q}_I(\boldsymbol{B}) = \|\boldsymbol{B} - \boldsymbol{Y}\|_F^2,$$

$$\Omega = \{\boldsymbol{Y} \in \mathbb{R}^{n \times r} | \boldsymbol{Y}^T \boldsymbol{Y} = n\boldsymbol{I}_r\}$$

- Discrete optimization through coordinate descent

$$\arg\min_{\boldsymbol{B}} \mathcal{Q} = tr(\boldsymbol{P}\boldsymbol{B}^T\boldsymbol{B}\boldsymbol{P}^T) - 2tr(\boldsymbol{B}^T\boldsymbol{U})$$

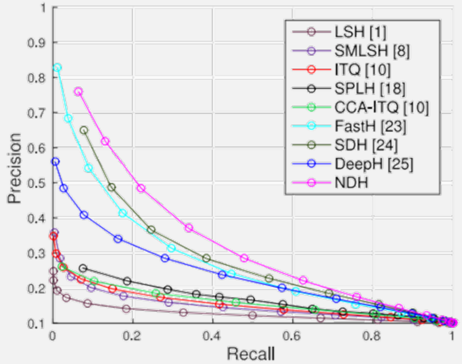
$$\arg\min_{\boldsymbol{b}_i} (\boldsymbol{p}_i^T \hat{\boldsymbol{P}} \hat{\boldsymbol{B}}^T - \boldsymbol{u}_i^T)\boldsymbol{b}_i$$

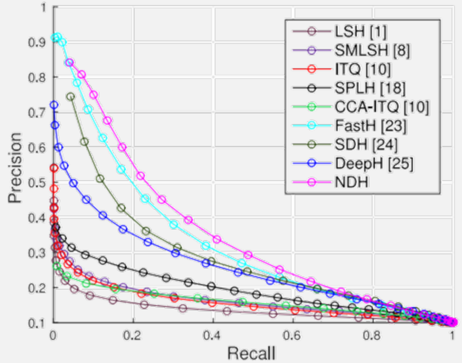
$$\boldsymbol{b}_i = sgn(\boldsymbol{u}_i - \hat{\boldsymbol{B}}\hat{\boldsymbol{P}}^T\boldsymbol{p}_i)$$

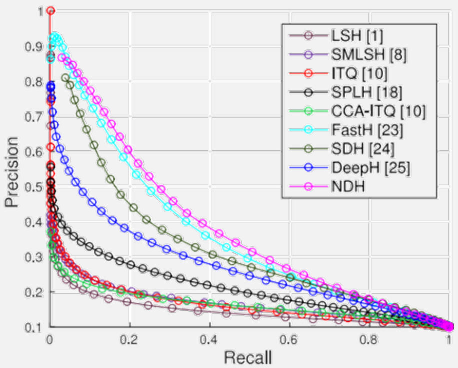
Experimental Results

- The CIFAR-10 dataset

Methods	Mean average precision(%)			Precision@500(%)			Precision@(radius==2)(%)		
	16	32	64	16	32	64	16	32	64
LSH [1]	12.63	13.70	14.62	15.32	17.23	19.36	16.67	6.35	0.1
SMLSH [8]	14.96	16.41	16.98	17.82	19.75	20.36	18.28	14.65	4.03
ITQ [10]	15.57	15.80	16.57	19.91	21.04	22.53	22.89	15.66	1.44
SPLH [18]	17.08	19.38	21.21	21.22	26.39	29.34	16.70	27.17	30.02
CCA-ITQ [10]	16.21	16.02	16.49	24.63	24.44	26.77	21.45	28.22	26.47
FastH [23]	27.94	33.09	36.55	37.74	43.13	46.84	37.76	34.42	11.64
SDH [24]	29.21	29.22	32.67	39.08	39.62	42.15	30.19	36.90	38.98
DeepH [25]	24.04	25.96	27.53	32.45	34.99	36.85	33.25	37.42	25.43
NDH	33.75	35.93	37.90	43.58	46.67	48.24	36.10	43.62	32.32





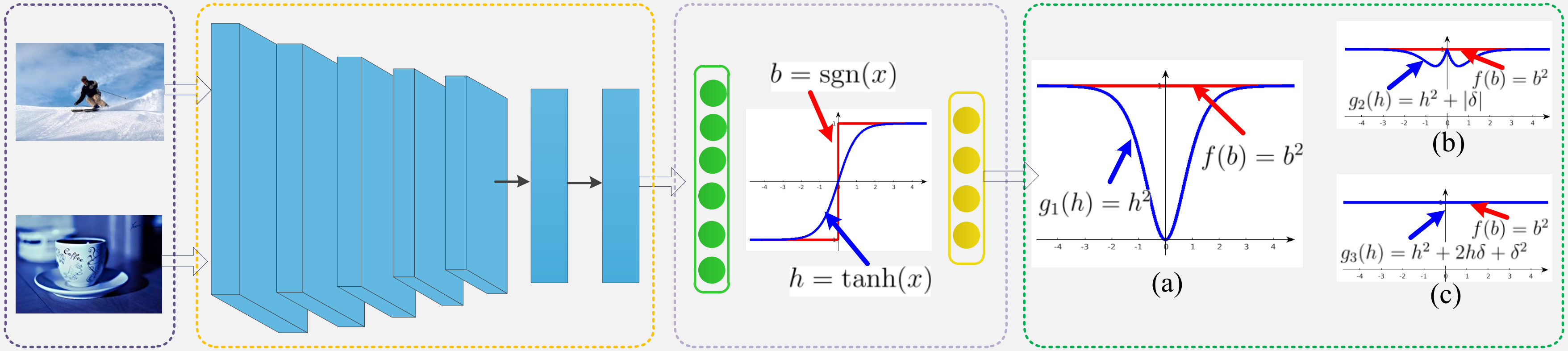


Discrepancy Minimization

- Intractable optimization of the objective over the binary codes

$$B \in \{-1, 1\}^{n \times l}$$

- Gradient based optimization of the deep neural network

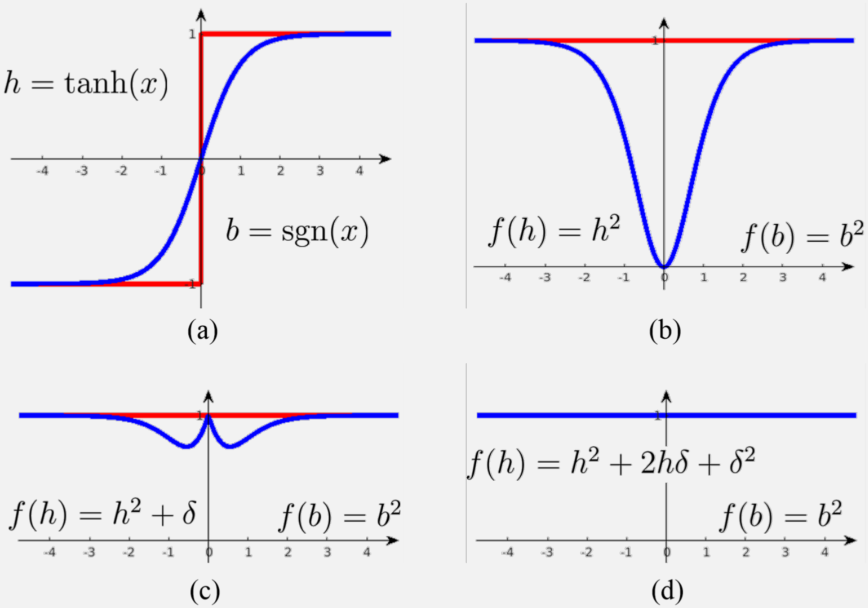


- Zhixiang Chen, Xin Yuan, **Jiwen Lu***, Qi Tian, Jie Zhou, Deep Hashing by Discrepancy Minimization, **CVPR**, 2018.

Discrepancy Minimization

- Discrepancy minimization
- Series expansion

$$
\begin{aligned}
\mathcal{L}(\boldsymbol{B}) &= \mathcal{L}(\boldsymbol{H}+\Delta) \\
&= \mathcal{L}(\boldsymbol{H}) + \sum_{i=1}^{n\times l} \frac{\partial \mathcal{L}(\boldsymbol{H})}{\partial \vec{h}_i} \vec{\Delta}_i \\
&+ \frac{1}{2} \sum_{i=1}^{n\times l} \sum_{j=1}^{n\times l} \frac{\partial^2 \mathcal{L}(\boldsymbol{H})}{\partial \vec{h}_i \partial \vec{h}_j} \vec{\Delta}_i \vec{\Delta}_j + \dots
\end{aligned}
$$



Discrepancy Minimization

- Objective Function
  - Pairwise similarity preservation
  - Expansion with series
  - Quantization loss minimization with large effect of high order terms

$$\arg\min_{\boldsymbol{B}} \mathcal{L}(\boldsymbol{B}) = \mathrm{tr}\left(\boldsymbol{B}^T \hat{\boldsymbol{D}} \boldsymbol{B}\right), \quad s.t. \quad \boldsymbol{B} \in \{-1,1\}^{n\times l}$$

$$\begin{aligned} \arg\min_{\boldsymbol{H},\Delta} \mathcal{L}(\boldsymbol{H},\Delta) = & \ \mathrm{tr}\left(\boldsymbol{H}^T \hat{\boldsymbol{D}} \boldsymbol{H}\right) \\ & + \lambda_1 \mathrm{tr}\left(\Delta^T \left(\hat{\boldsymbol{D}}^T + \hat{\boldsymbol{D}}\right) \boldsymbol{H}\right) \\ & + \lambda_2 \mathrm{tr}\left(\Delta^T \hat{\boldsymbol{D}} \Delta\right), \\ s.t. \quad & (\boldsymbol{H} + \Delta) \in \{-1,1\}^{n\times l} \end{aligned}$$

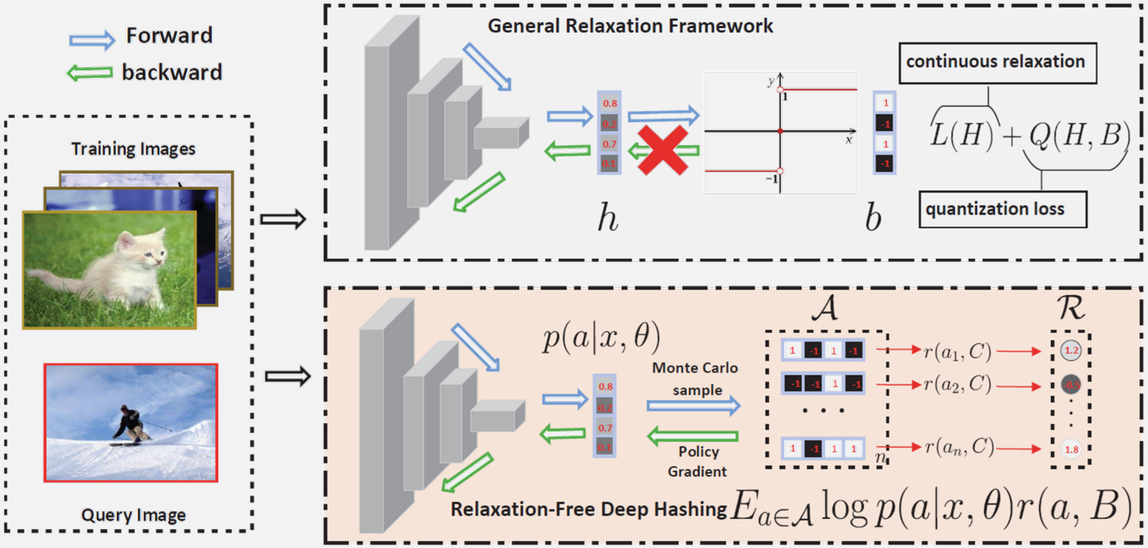
Experimental Results

- The CIFAR-10 dataset

Methods	CIFAR-10			
	16	32	48	64
LSH [9]	0.1314	0.1582	0.1723	0.1785
SH [46]	0.1126	0.1325	0.1113	0.1466
ITQ [10]	0.2312	0.2432	0.2482	0.2531
KSH [31]	0.3216	0.3285	0.3371	0.4412
ITQ-CCA [10]	0.3142	0.3612	0.3662	0.3921
FastH [22]	0.4532	0.4577	0.4672	0.4854
SDH [36]	0.4122	0.4301	0.4392	0.4465
CNNH [47]	0.5373	0.5421	0.5765	0.5780
DNNH [20]	0.5978	0.6031	0.6087	0.6166
DPSH [21]	0.6367	0.6412	0.6573	0.6676
DSH [27]	0.6792	0.6465	0.6624	0.6713
HashNet [2]	0.6857	0.6923	0.7183	0.7187
DMDH	**0.7037**	**0.7191**	**0.7319**	**0.7373**

Relaxation-Free Hamming DML

- Most deep hashing can't be trained in a truly end-to-end manner with non-smooth sign activations
- A relaxation-free framework with reformulating the hashing layer as sampling via policy gradient



- Xin Yuan, Liangliang Ren, **Jiwen Lu***, Jie Zhou, Relaxation-Free Deep Hashing via Policy Gradient, **ECCV**, 2018.

Relaxation-Free Hamming DML

- Weighted Reward Function

$$r(\boldsymbol{a}_i) = -\frac{1}{2}\sum_{j=1}^{n} \hat{s}_{ij}(K - \mathbf{b}_i^T \hat{\mathbf{b}}_j)$$

$$s.t. \quad \mathbf{b}_i, \hat{\mathbf{b}}_j \in \{-1, +1\}^K$$

where

$$\hat{s}_{ij} = \begin{cases} \beta, & \text{if } s_{ij} = 1 \\ \beta - 1, & \text{otherwise} \end{cases}$$

- Policy Gradient with REINFORCE

$$\nabla_\theta \mathcal{L}(\theta) = -\sum_i \mathbb{E}_{\boldsymbol{a}_i \in \mathcal{A}_i}[r(\boldsymbol{a}_i)\nabla_\theta \log(P_\theta(\boldsymbol{a}_i|\boldsymbol{x}_i))]$$

- REINFORCE with a Baseline

$$\nabla_\theta \mathcal{L}(\theta) \approx -\frac{1}{T}\sum_i \sum_t [r(\boldsymbol{a}_i^t)\nabla_\theta \log(P_\theta(\boldsymbol{a}_i^t|\boldsymbol{x}_i))]$$

Relaxation-Free Hamming DML

Out-of-Sample Extensions

- Deterministic Generation

$$b_q^k = \begin{cases} +1, & \text{if } \pi^{(k)}_{\boldsymbol{x}_q,\theta} > 0.5 \\ -1, & \text{otherwise} \end{cases}$$

- Stochastic Generation

$$b_q^k = \begin{cases} +1, & \text{with probability} \quad \pi^{(k)}_{\boldsymbol{x}_q,\theta} \\ -1, & \text{with probability} \quad 1 - \pi^{(k)}_{\boldsymbol{x}_q,\theta} \end{cases}$$

Experimental Results

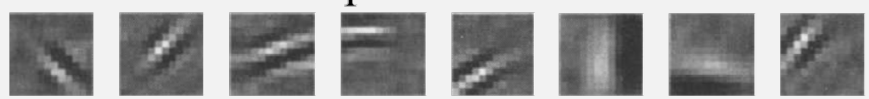
- The CIFAR-10, NUS-WIDE and ImageNet datasets

Methods	CIFAR-10 (%)				NUS-WIDE (%)				ImageNet (%)			
	16	32	48	64	16	32	48	64	16	32	48	64
LSH [22]	12.9	15.2	16.9	17.8	40.3	49.2	49.3	55.1	10.1	23.5	30.1	34.9
SH [25]	12.2	13.5	12.1	12.6	47.9	49.1	49.8	51.5	20.8	32.7	39.5	42.0
ITQ [6]	21.3	23.4	23.8	25.3	56.7	60.3	62.2	62.6	32.5	46.2	51.3	55.6
CCA-ITQ [6]	31.4	36.1	36.6	37.9	50.9	54.4	56.8	67.6	26.6	43.6	54.8	58.0
KSH [3]	35.6	40.8	53.1	44.1	40.6	40.8	38.7	39.8	16.0	28.8	34.2	39.4
FastH [30]	45.3	46.1	48.7	50.3	51.9	61.0	64.7	65.2	22.8	44.7	51.7	55.6
SDH [31]	40.2	42.0	44.9	45.6	53.4	61.8	63.1	64.5	29.9	45.1	54.9	59.3
CNNH [23]	48.8	51.2	53.4	53.6	61.2	62.3	62.1	63.7	28.8	44.7	52.8	55.6
DNNH [24]	55.5	55.8	58.1	62.3	68.1	71.3	71.8	72.0	29.7	46.3	54.0	56.6
DPSH [37]	64.6	66.1	67.7	68.6	71.5	72.6	73.8	75.3	32.6	54.6	61.7	65.4
DSH [35]	68.9	69.1	70.3	71.6	71.8	72.3	74.2	75.6	34.8	55.0	62.9	66.5
HashNet [36]	70.3	71.1	71.6	73.9	73.3	75.2	76.2	77.6	50.6	62.9	66.3	68.4
PGDH	**73.6**	**74.1**	**74.7**	**76.2**	**76.1**	**78.0**	**78.6**	**79.2**	**51.8**	**65.3**	**70.7**	**71.6**

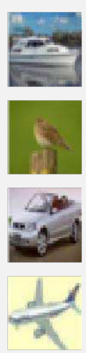
Sparse Hamming DML

- Assumption
  - images are generally descripted in terms of a small group of structural primitives

primitives



samples

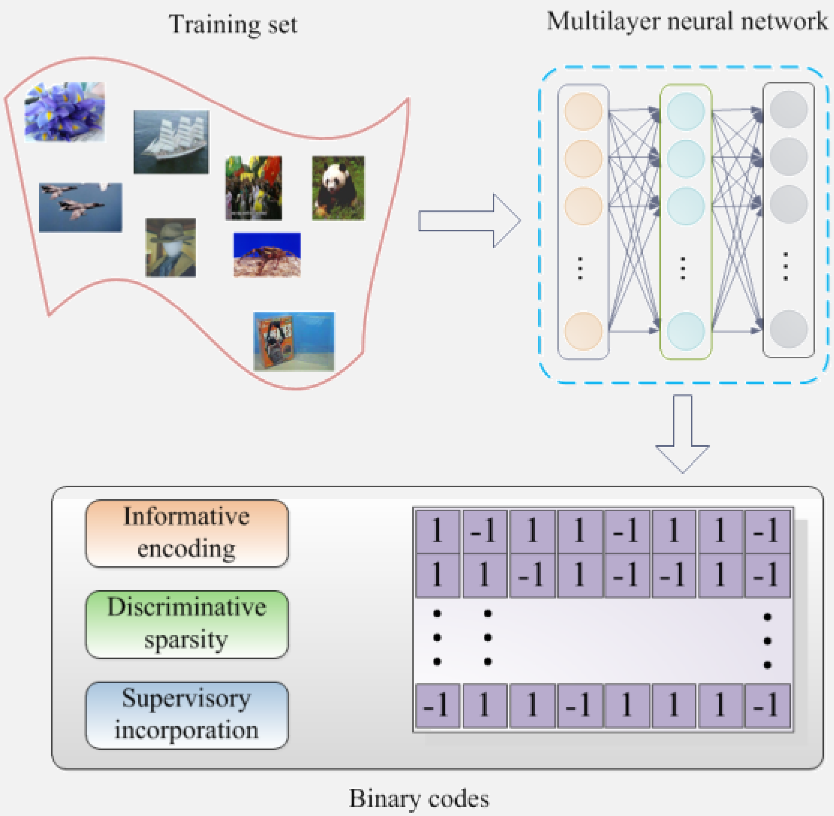


$$\begin{bmatrix} 0.3 & 0 & 0.7 & 0 & 0 & 0 & 0 & 0.1 \\ 0 & 0.4 & 0 & 0.3 & 0 & 0 & 0 & 0 \\ 0.5 & 0 & 0 & 0 & 0 & 0.1 & 0 & 0 \\ 0 & 0 & 0.9 & 0 & 0 & 0 & 0.2 & 0 \end{bmatrix}$$

- Zhixiang Chen, **Jiwen Lu***, Jianjiang Feng, Jie Zhou, Nonlinear Sparse Hashing, **TMM**, 2017.

Sparse Hamming DML

- Capture salient structure of image samples with sparsity constraint
- Exploit the nonlinear relationship of samples with nonlinear hashing functions



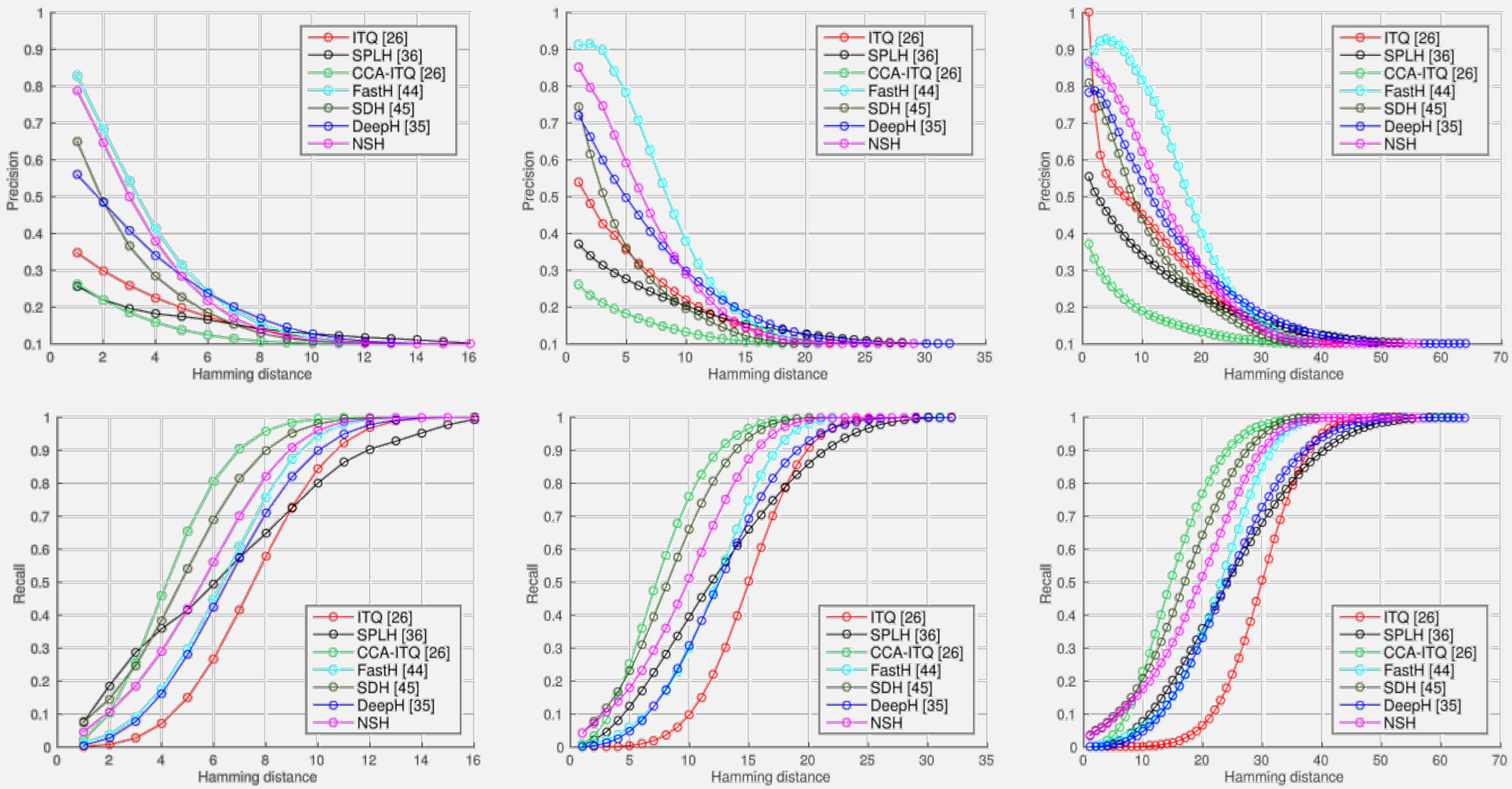
Sparse Hamming DML

- Objective Function
  - Informative binary encoding
  - Discriminative sparse constraint
  - Supervision incorporated learning

$$
\begin{aligned}
\arg\min_{\boldsymbol{B},\boldsymbol{W},\boldsymbol{c},\boldsymbol{P}} \; \mathcal{Q} &= \mathcal{Q}_{be} + \lambda_0 \mathcal{Q}_{se} + \lambda_1 \mathcal{Q}_{sl}, \\
&= ||\boldsymbol{B} - \boldsymbol{H}||_F^2 - \gamma tr(\boldsymbol{H}\boldsymbol{H}^T) \\
&+ \lambda_0 \|\boldsymbol{H}\|_{2,1} + \lambda_1 ||\boldsymbol{Y} - \boldsymbol{P}\boldsymbol{B}^T||_F^2
\end{aligned}
$$

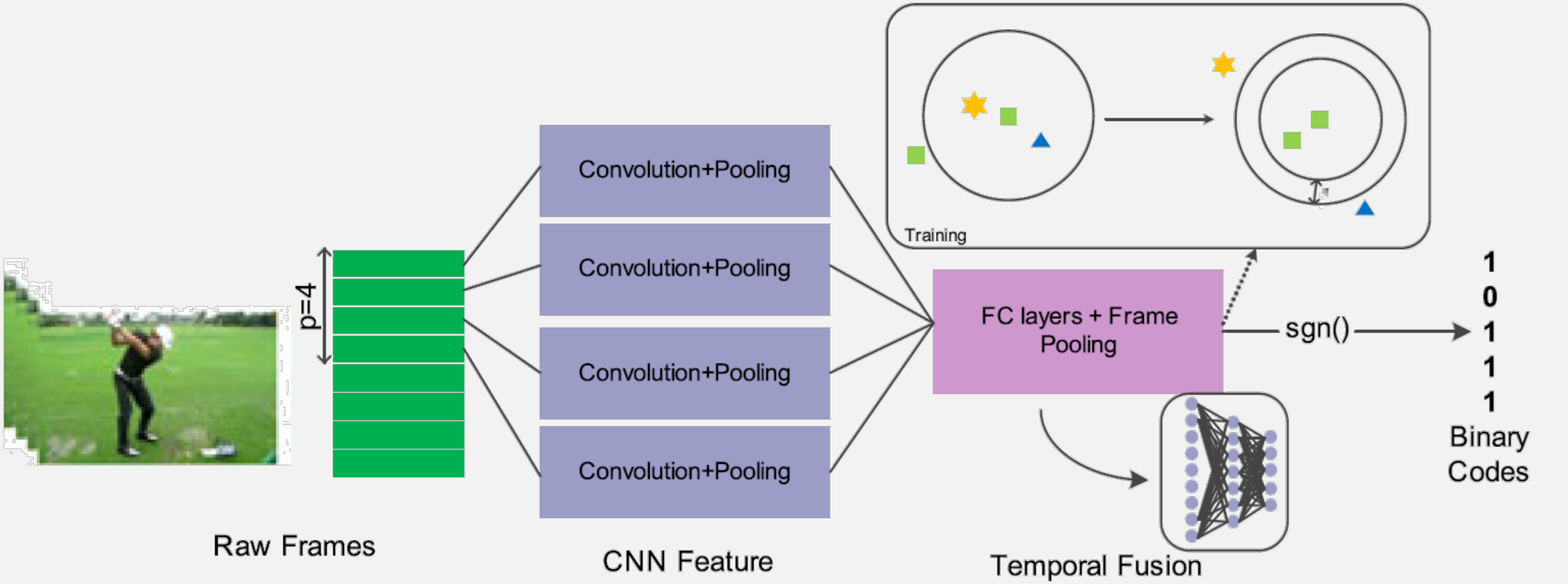
Experimental Results

- The CIFAR-10 dataset



Hamming DML for Video Search

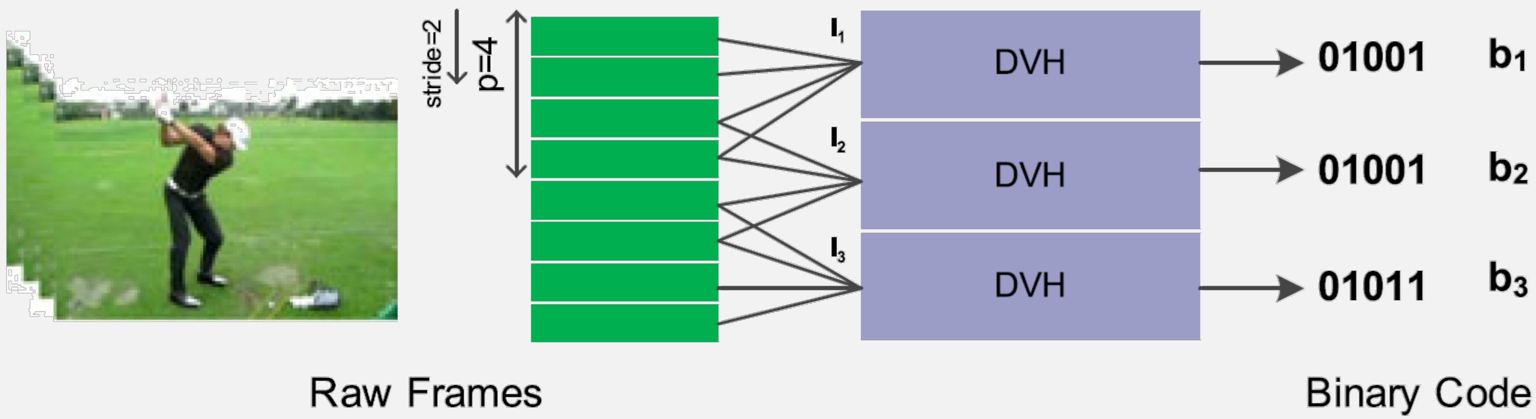
- Exploit spatio-temporal information



- Venice Erin Liong, **Jiwen Lu***, Yap-Peng, Jie Zhou, Deep Video Hashing, **TMM**, 2017.

Hamming DML for Video Search

- Binary code extraction



$$
\begin{aligned}
\min_{\mathbf{b}_v, \mathbf{b}_v} J = & \; J_1 + \lambda J_2 \\
= & \; f(1 - \delta_{u,v}(\theta - d_{u,v}(\mathbf{b}_u, \mathbf{b}_v))) \\
& + \lambda(\|s(\mathbf{I}_u) - \mathbf{b}_u\|_F^2 + \|s(\mathbf{I}_v) - \mathbf{b}_v\|_F^2)
\end{aligned}
$$

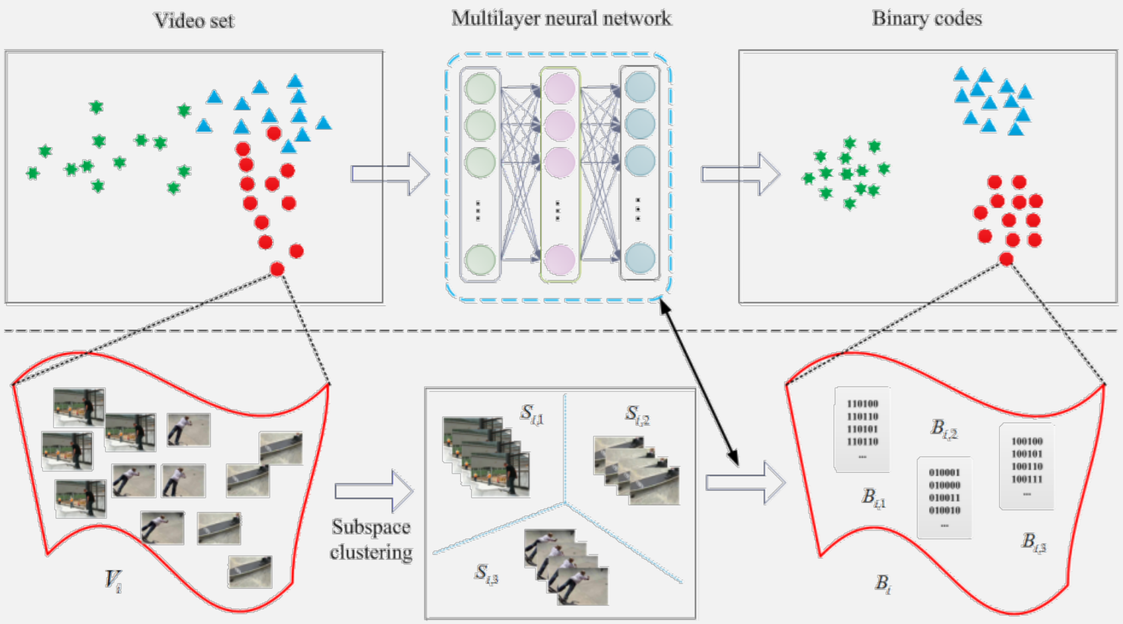
Experimental Results

- The Columbia Consumer Video dataset

Method	Hamming ranking (mAP, %)			precision (%) @ N = 100			precision (%) @ r = 2	
	16	32	64	16	32	64	16	32
PCAH [28]	20.83	21.45	19.37	25.80	26.50	25.51	3.03	0
PCA-ITQ [6]	22.49	24.13	24.42	27.71	28.99	29.61	13.43	0
AGH [38]	14.91	15.22	11.24	20.52	23.37	20.16	13.43	1.58
KSH [31]	32.43	34.34	35.40	36.27	38.33	38.75	18.27	7.64
CCA-ITQ [6]	36.58	38.18	38.32	39.13	40.41	40.51	16.15	7.17
FastHash [37]	34.72	38.37	38.47	38.83	40.85	41.37	12.73	5.36
DVH	**38.54**	**41.08**	**41.51**	**40.29**	**42.08**	**42.23**	**37.32**	**23.10**

Structural Hamming DML

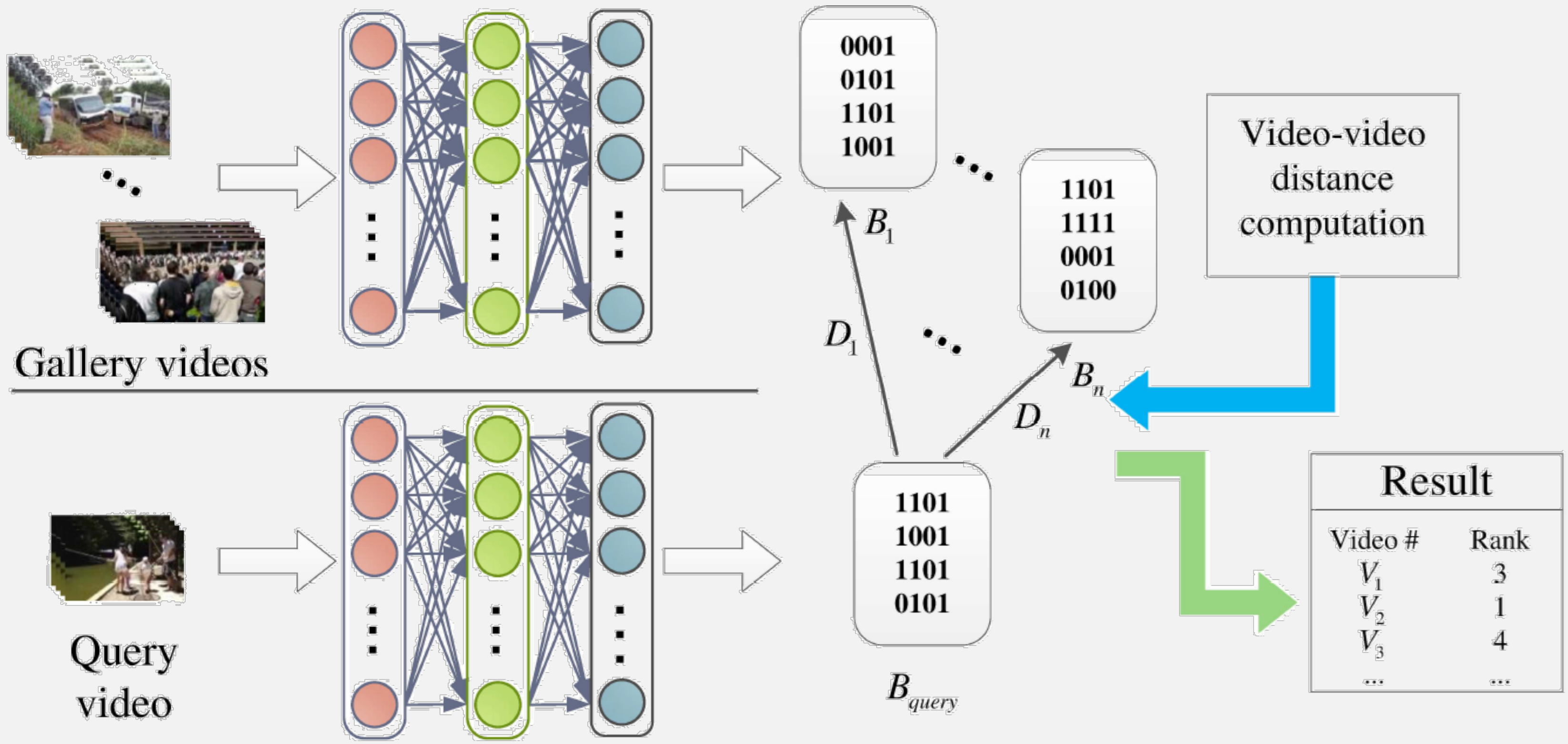
- Exploiting both the structural information between frames and nonlinear relationship between videos samples



- Zhixiang Chen, **Jiwen Lu***, Jianjiang Feng, Jie Zhou, Nonlinear Structural Hashing for Scalable Video Search, **TCSVT**, 2018.

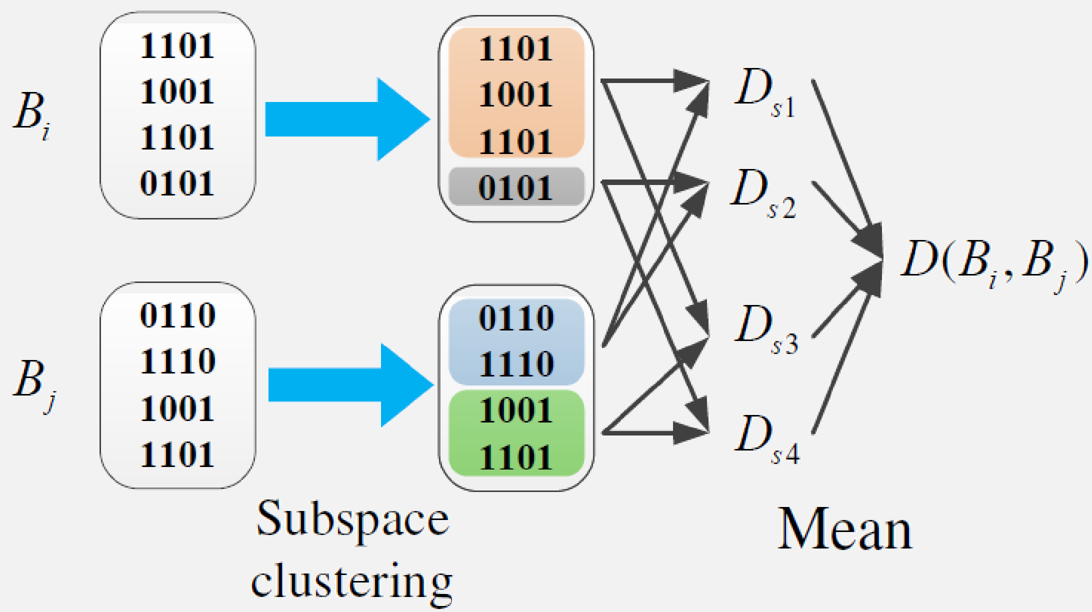
Structural Hamming DML

- Workflow to generate ranking list for similarity search



Structural Hamming DML

- Computation of distance between binary code matrices of videos



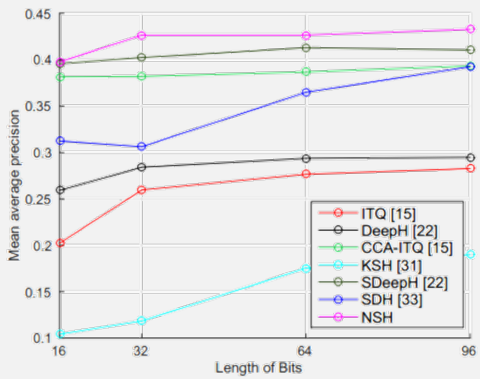
Structural Hamming DML

- Objective Function
  - inter-video similarity loss based on discriminative distance metric
  - intra-video similarity loss to embed scene consistent constraint

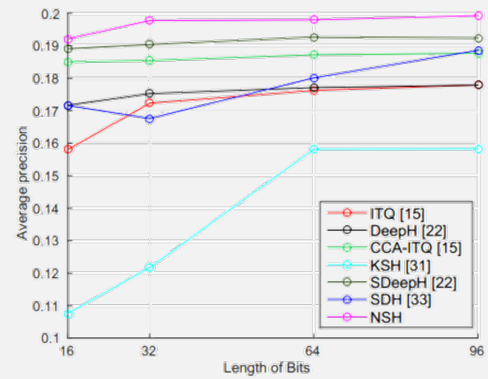
$$
\begin{aligned}
\underset{\{\boldsymbol{W}^{(k)},\boldsymbol{c}^{(k)}\}_{k=1}^{K}}{\arg\min} \quad \mathcal{L} &= \mathcal{L}_v + \lambda_1 \mathcal{L}_f + \lambda_2 \mathcal{L}_r \\
&= \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \ell_{i,j}\left(D(\boldsymbol{B}_i, \boldsymbol{B}_j) - \tau\right) \\
&+ \frac{\lambda_1}{2}\sum_{i=1}^{N} \left\|\boldsymbol{R}_i \boldsymbol{B}_i^T\right\|_2^2 \\
&+ \frac{\lambda_2}{2}\sum_{k=1}^{K} \left(\left\|\boldsymbol{W}^{(k)}\right\|_2^2 + \left\|\boldsymbol{c}^{(k)}\right\|_2^2\right)
\end{aligned}
$$

Experimental Results

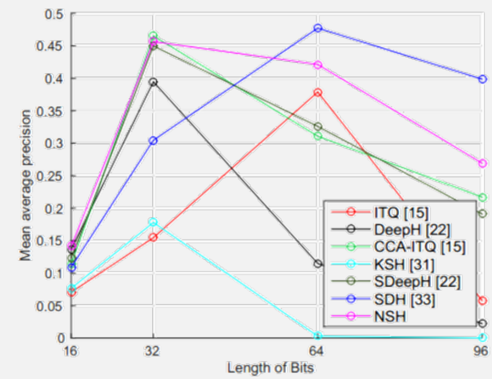
◻ The Columbia Consumer Video dataset



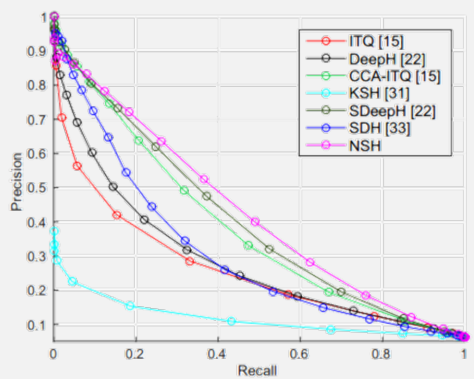
(a) mean Average Precision



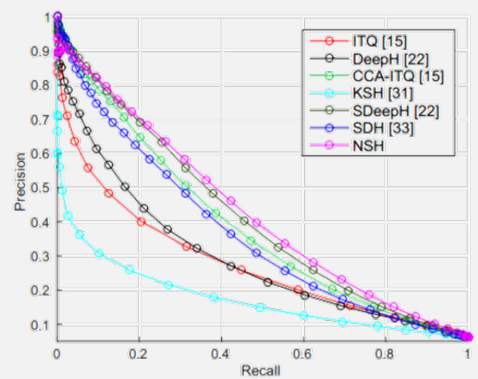
(b) Precision @500



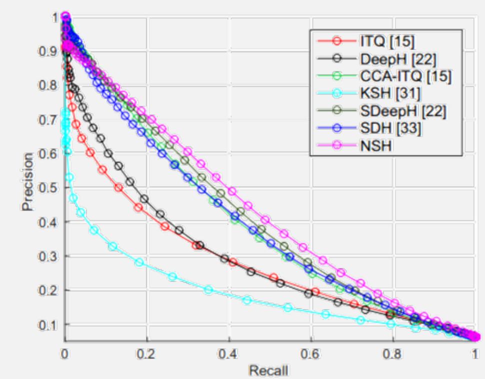
(c) Precision within Hamming radius 2



(a) 32 bits

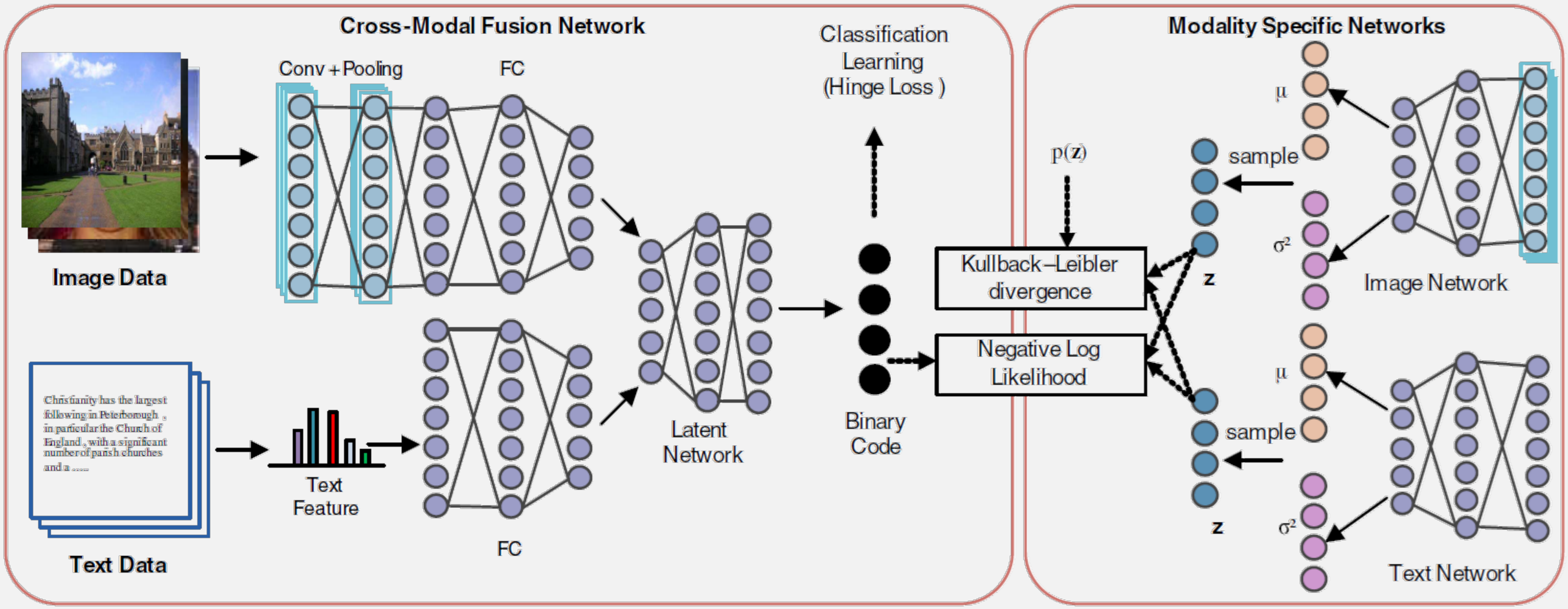


(b) 64 bits



(c) 96 bits

Cross-Modal Hamming DML



- Venice Erin Liong, **Jiwen Lu***, Yap-Peng Tan, Jie Zhou, Cross-Modal Deep Variational Hashing, **ICCV**, 2017.

Cross-Modal Hamming DML

- Cross-modal fusion network

$$
\begin{aligned}
\min_{\mathbf{B},\mathbf{M},\theta_u,\theta_v,\theta_w} J &= J_1 + \lambda J_2 \\
&= \|\mathbf{M}\|_F^2 + \sum_{n}^{N} \xi_n + \lambda(\|\mathbf{B}-\mathbf{H}\|_F^2) \\
&\forall n,j \;\; \mathbf{y}_{n,j}(\mathbf{m}_j^\top \mathbf{b}_n) \geq 1 - \xi_n \\
&\forall n \;\; \mathbf{b}_n = \{-1, 1\}
\end{aligned}
$$

Cross-Modal Hamming DML

- Modality-specific networks

$$
\begin{aligned}
\min_{\theta} \mathcal{L} &= \sum_{i=1}^{N}\sum_{k=1}^{K}\mathcal{L}_{NLL} + \sum_{i=1}^{N}\alpha\mathcal{L}_{KLD} \\
&= \sum_{i=1}^{N}\sum_{k=1}^{K} -\log(1 + e^{b_i^{(k)} z_{*i}^{(k)}}) \\
&- \frac{\alpha}{2}\sum_{i=1}^{N}\sum_{j=1}^{J}(1 + \log((\sigma_{*i}^{(j)})^2 - (\mu_{*i}^{(j)})^2 - (\sigma_{*i}^{(j)})^2)
\end{aligned}
$$

Experimental Results

- Query images or texts/tags

	Wiki				IAPRTC12				NUS-WIDE			
Method	16 bits	32 bits	64 bits	128 bits	16 bits	32 bits	64 bits	128 bits	16 bits	32 bits	64 bits	128 bits
CVH [14]	0.2383	0.2038	0.1791	0.1580	0.5370	0.5409	0.5242	0.4962	0.5045	0.5484	0.5588	0.5583
CCA-ITQ [8]	0.3328	0.3216	0.3064	0.328	0.5587	0.5853	0.5895	0.5855	0.5400	0.5960	0.6194	0.6229
PDH [23]	0.3251	0.3258	0.3436	0.3438	0.5927	0.6085	0.6302	0.6450	0.5687	0.6148	0.6475	0.6793
LSSH [37]	0.3645	0.3713	0.3777	0.3580	0.5440	0.5769	0.5964	0.5985	0.5547	0.5734	0.5980	0.5968
CMFH [5]	0.2665	0.2755	0.2876	0.2950	0.5601	0.5829	0.6079	0.6179	0.4772	0.5301	0.5763	0.6258
SCM [36]	0.1387	0.1367	0.1413	0.1359	0.5665	0.5051	0.4548	0.4178	0.5190	0.4837	0.4495	0.4189
SePH - *km* [17]	0.4144	0.4354	0.4374	0.4472	0.6177	0.6447	0.6500	0.6781	0.6524	0.6526	0.6637	0.6696
DisCMH [35]	0.3754	0.3936	0.3901	0.3915	0.6174	0.6596	0.6503	0.6594	0.6826	0.7583	0.7752	0.7605
CMDVH	**0.4242**	**0.4430**	**0.4519**	**0.4442**	**0.7196**	**0.7727**	**0.8004**	**0.7902**	**0.8503**	**0.8755**	**0.8801**	**0.8910**

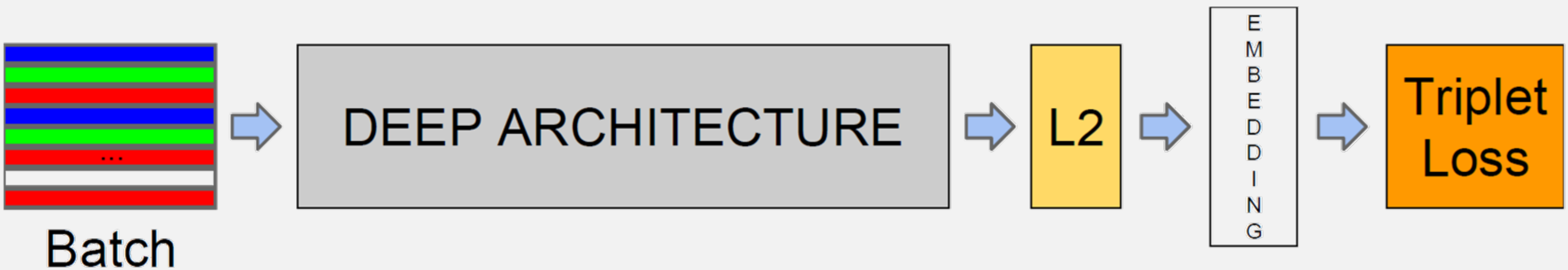
	Wiki				IAPRTC12				NUS-WIDE			
Method	16 bits	32 bits	64 bits	128 bits	16 bits	32 bits	64 bits	128 bits	16 bits	32 bits	64 bits	128 bits
CVH [14]	0.3882	0.3362	0.2567	0.2297	0.5677	0.5784	0.5610	0.5362	0.5280	0.5732	0.5864	0.5807
CCA-ITQ [8]	0.5463	0.5505	0.5593	0.5633	0.5863	0.6123	0.6143	0.6053	0.5753	0.6151	0.6405	0.6360
PDH [23]	0.5432	0.5592	0.57554	0.58474	0.5960	0.6133	0.6345	0.6488	0.5844	0.6402	0.6817	0.7087
LSSH [37]	0.6061	0.6256	0.6384	0.6376	0.4868	0.5264	0.5547	0.5724	0.5857	0.6242	0.6293	0.6464
CMFH [5]	0.3955	0.4105	0.4473	0.4807	0.5592	0.5834	0.6084	0.6187	0.4965	0.5432	0.5995	0.6405
SCM [36]	0.1322	0.1429	0.1556	0.1494	0.6521	0.5697	0.4776	0.4213	0.5485	0.5033	0.4481	0.3920
SePH - *km* [17]	0.7007	0.6999	0.7099	0.7153	0.6105	0.6340	0.6404	0.6730	0.6604	0.6766	0.7043	0.7024
DisCMH [35]	0.6772	0.6602	0.6632	0.6537	0.6532	0.6910	0.6921	0.6949	0.6519	0.7378	0.7535	0.7511
CMDVH	**0.7270**	**0.7326**	**0.7383**	**0.7371**	**0.7348**	**0.7744**	**0.8038**	**0.8111**	**0.8270**	**0.8328**	**0.8403**	**0.8782**

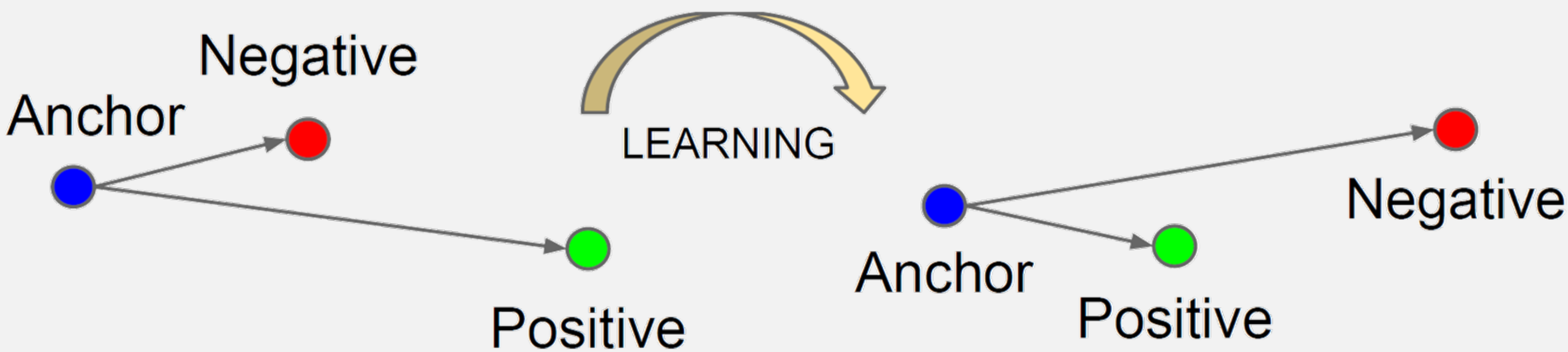
Part 4: Sampling for Deep Metric Learning

Semi-Hard Negative Mining

- FaceNet

$$\sum_{i}^{N}\left[\|f(x_i^a)-f(x_i^p)\|_2^2-\|f(x_i^a)-f(x_i^n)\|_2^2+\alpha\right]_+$$





[Schroff et al., CVPR'15]

Semi-Hard Negative Mining

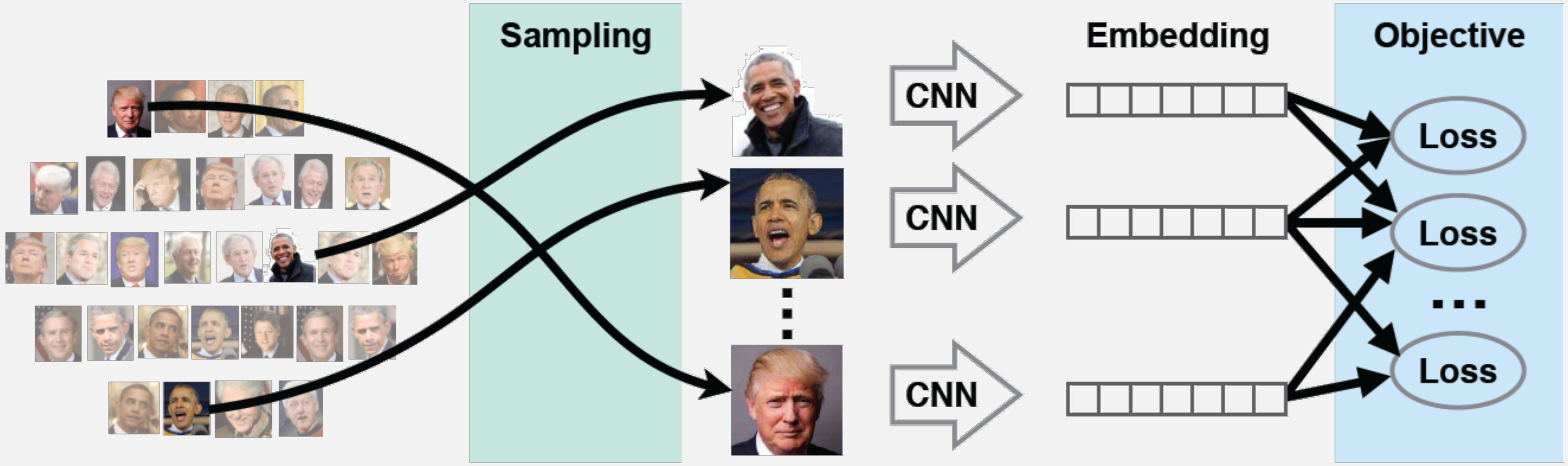
- Using all the positive samples
- Selecting semi-hard negative samples

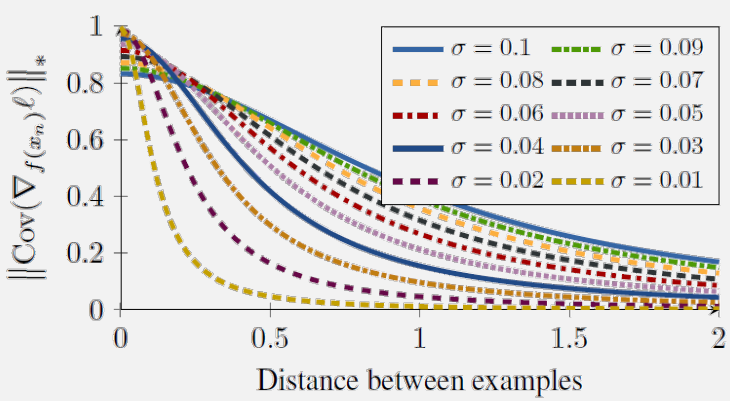
Selecting the hardest negatives can in practice lead to bad local minima early on in training, specifically it can result in a collapsed model (*i.e.* $f(x) = 0$). In order to mitigate this, it helps to select x_i^n such that

$$\|f(x_i^a) - f(x_i^p)\|_2^2 < \|f(x_i^a) - f(x_i^n)\|_2^2 \ . \qquad (3)$$

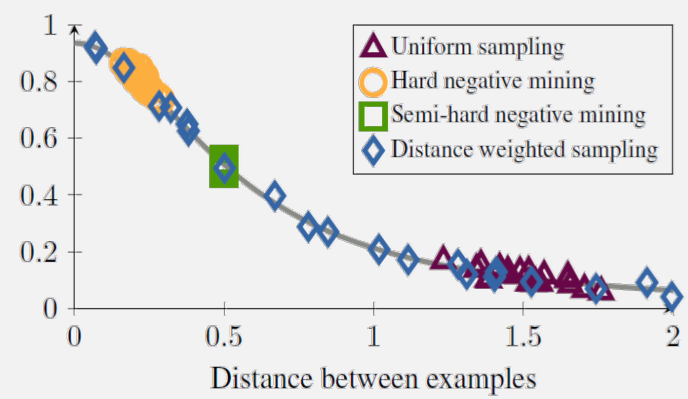
We call these negative exemplars *semi-hard*, as they are further away from the anchor than the positive exemplar, but still hard because the squared distance is close to the anchor-positive distance. Those negatives lie inside the margin α.

Sampling Matters for DML





(a) Variance of gradient at different noise levels.



(b) Sample distribution for different strategies.

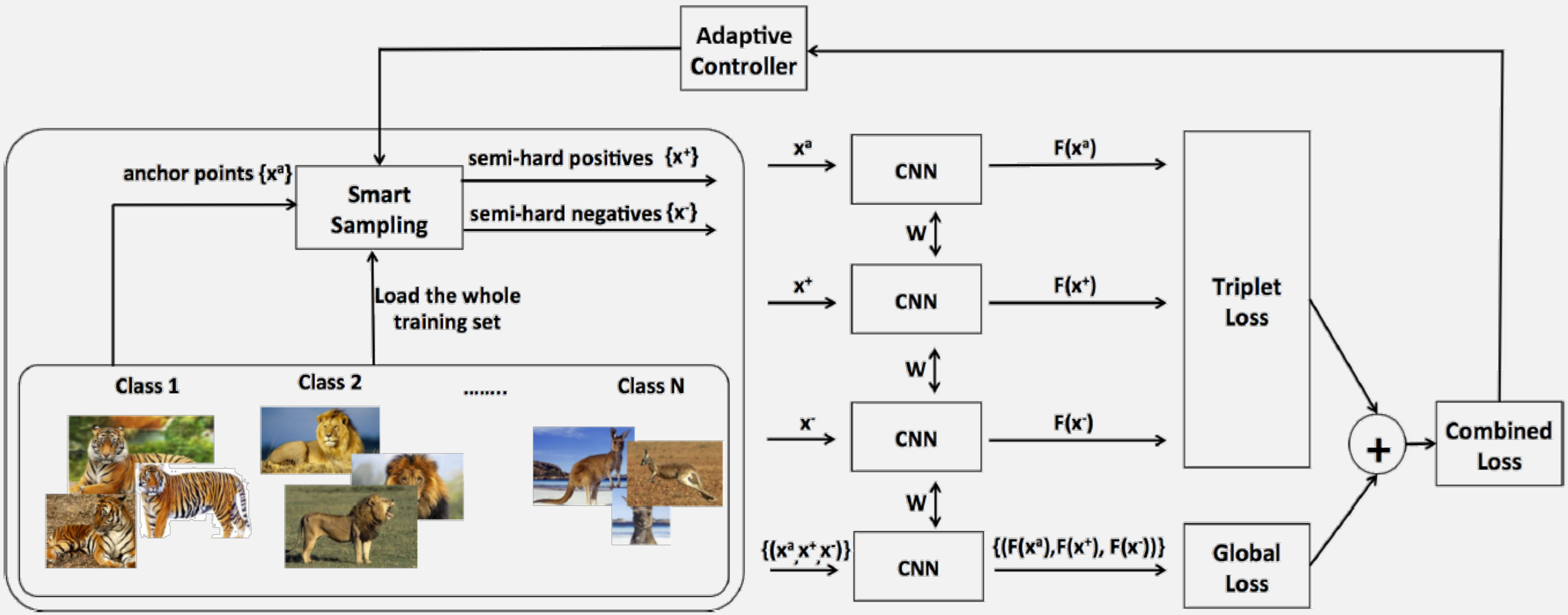
[Wu et al., ICCV'17]

Experiments Results

- The Stanford Online Products dataset

k	1	10	100	1000
Random				
Contrastive loss [11]	30.1	51.6	72.3	88.4
Margin	37.5	56.3	73.8	88.3
Semi-hard				
Contrastive loss [11]	49.4	67.4	81.8	92.1
Triplet ℓ_2^2 [25]	49.7	68.1	82.5	92.9
Triplet ℓ_2	47.4	67.5	83.1	93.6
Margin	<u>61.0</u>	<u>74.6</u>	85.3	93.6
Distance weighted				
Contrastive loss [11]	39.2	60.8	79.1	92.2
Triplet ℓ_2^2 [25]	53.4	70.8	83.8	93.4
Triplet ℓ_2	54.5	72.0	<u>85.4</u>	**94.4**
Margin	**61.7**	**75.5**	**86.0**	<u>94.0</u>
Margin (pre-trained)	72.7	86.2	93.8	98.0

Smart Mining for DML

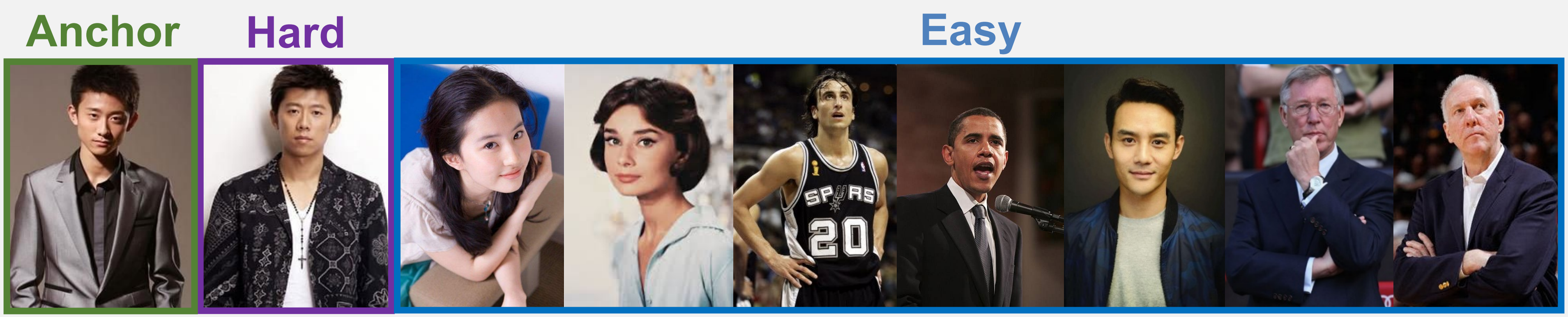


[Harwood et al., ICCV'17]

Deep Adversarial Metric Learning

- Easy negatives usually account for the vast majority

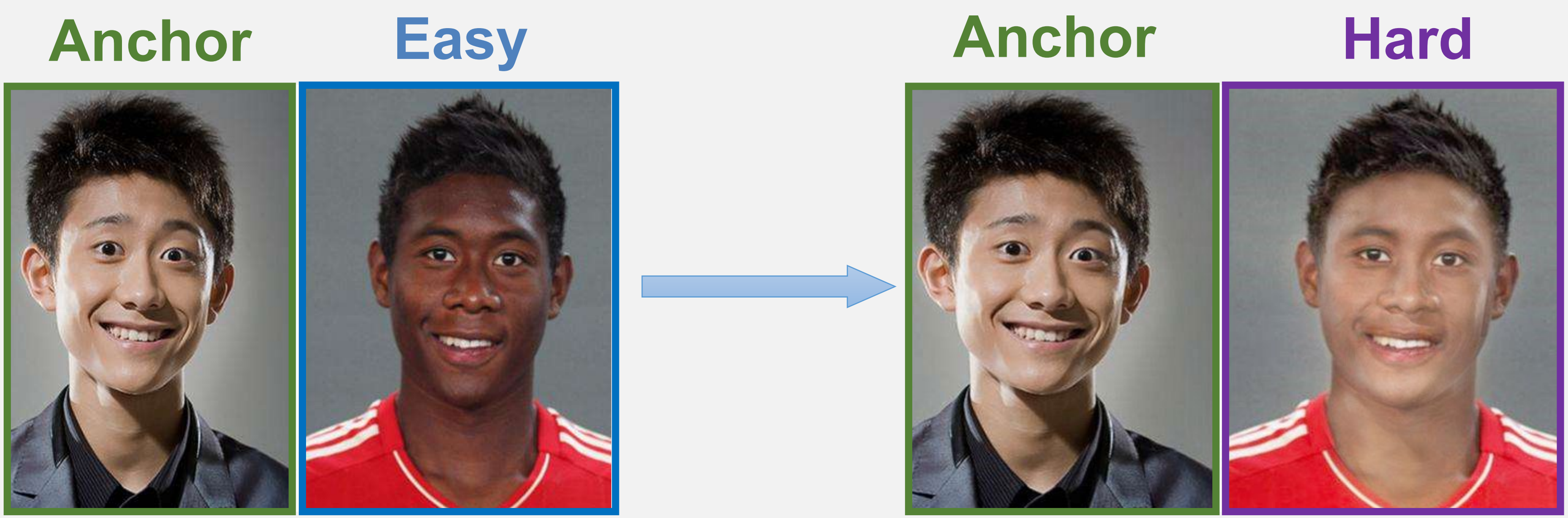
- Are easy negatives really useless?



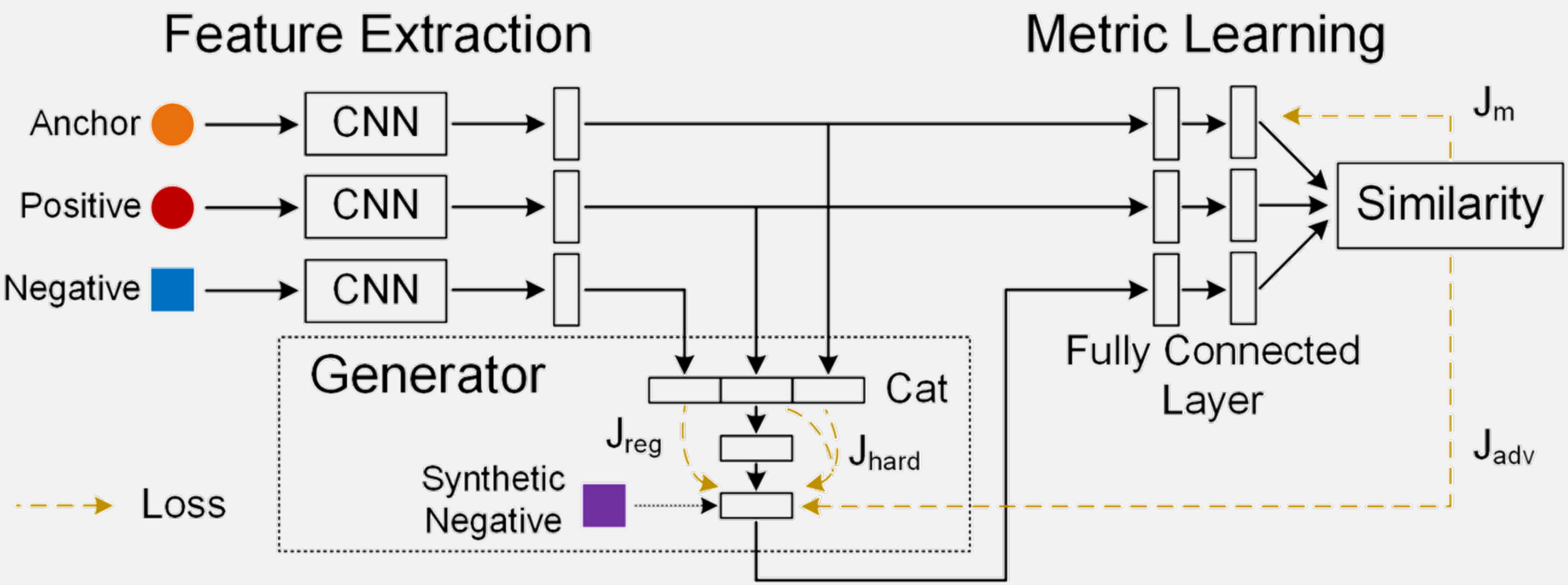
- Yueqi Duan, Wenzhao Zheng, Xudong Lin, **Jiwen Lu***, Jie Zhou, Deep Adversarial Metric Learning, **CVPR**, 2018.

Deep Adversarial Metric Learning

- DAML: Exploit the potentials of easy negatives through adversarial hard negative generation



Deep Adversarial Metric Learning



Deep Adversarial Metric Learning

- Objective function

$$\min_{\theta_g,\theta_f} J = J_{\text{gen}} + \lambda J_{\text{m}}$$

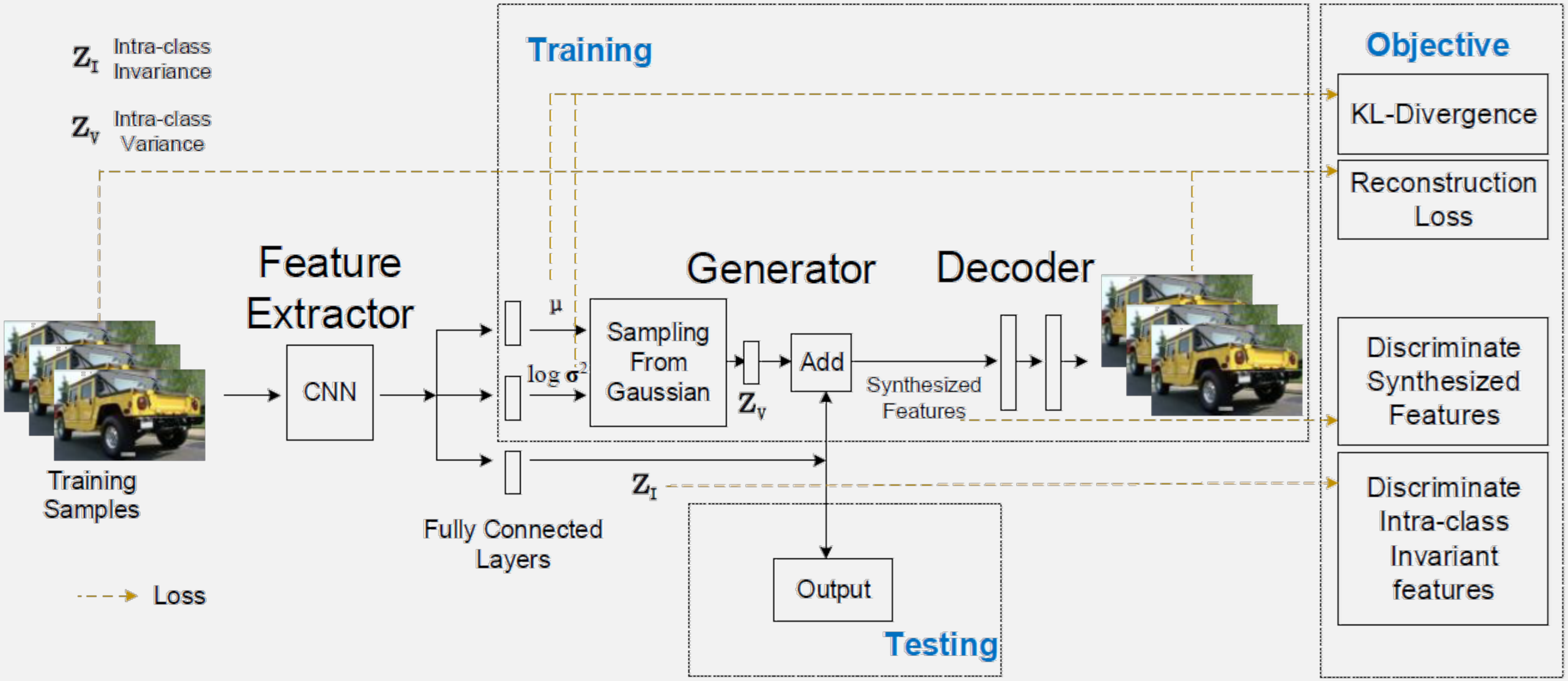
$$\begin{aligned}
\min_{\theta_g} J_{\text{gen}} &= J_{\text{hard}} + \lambda_1 J_{\text{reg}} + \lambda_2 J_{\text{adv}} \\
&= \sum_{i=1}^{N} (||\widetilde{\mathbf{x}}_i^- - \mathbf{x}_i||_2^2 + \lambda_1 ||\widetilde{\mathbf{x}}_i^- - \mathbf{x}_i^-||_2^2 \\
&+ \lambda_2 [D(\widetilde{\mathbf{x}}_i^-, \mathbf{x}_i)^2 - D(\mathbf{x}_i^+, \mathbf{x}_i)^2 - \alpha]_+)
\end{aligned}$$

Experimental Results

- The Stanford Online Products dataset

Method	NMI	F_1	R@1	R@10	R@100
DDML	83.4	10.7	42.1	57.8	73.7
Triplet+N-pair	86.4	21.0	58.1	76.0	89.1
Angular	87.8	26.5	67.9	83.2	92.2
Contrastive	82.4	10.1	37.5	53.9	71.0
DAML (cont)	**83.5**	**10.9**	**41.7**	**57.5**	**73.5**
Triplet	86.3	20.2	53.9	72.1	85.7
DAML (tri)	**87.1**	**22.3**	**58.1**	**75.0**	**88.0**
Lifted	87.2	25.3	62.6	80.9	91.2
DAML (lifted)	**89.1**	**31.7**	**66.3**	**82.8**	**92.5**
N-pair	87.9	27.1	66.4	82.9	92.1
DAML (N-pair)	**89.4**	**32.4**	**68.4**	**83.5**	**92.3**

Deep Variational Metric Learning



- Xudong Lin, Yueqi Duan, Qiyuan Dong, **Jiwen Lu***, Jie Zhou, Deep Variational Metric Learning, **ECCV**, 2018.

Deep Variational Metric Learning

- Assumption
  - Intra-class variance obeys the same distribution independent on classes.
- Method
  - Separate intra-class variance and class centers
  - Keep the intra-class variance and learn its distribution
  - Generate discriminative samples with the distribution
- Contribution
  - Explicitly learn class centers
  - Make full use of the dataset

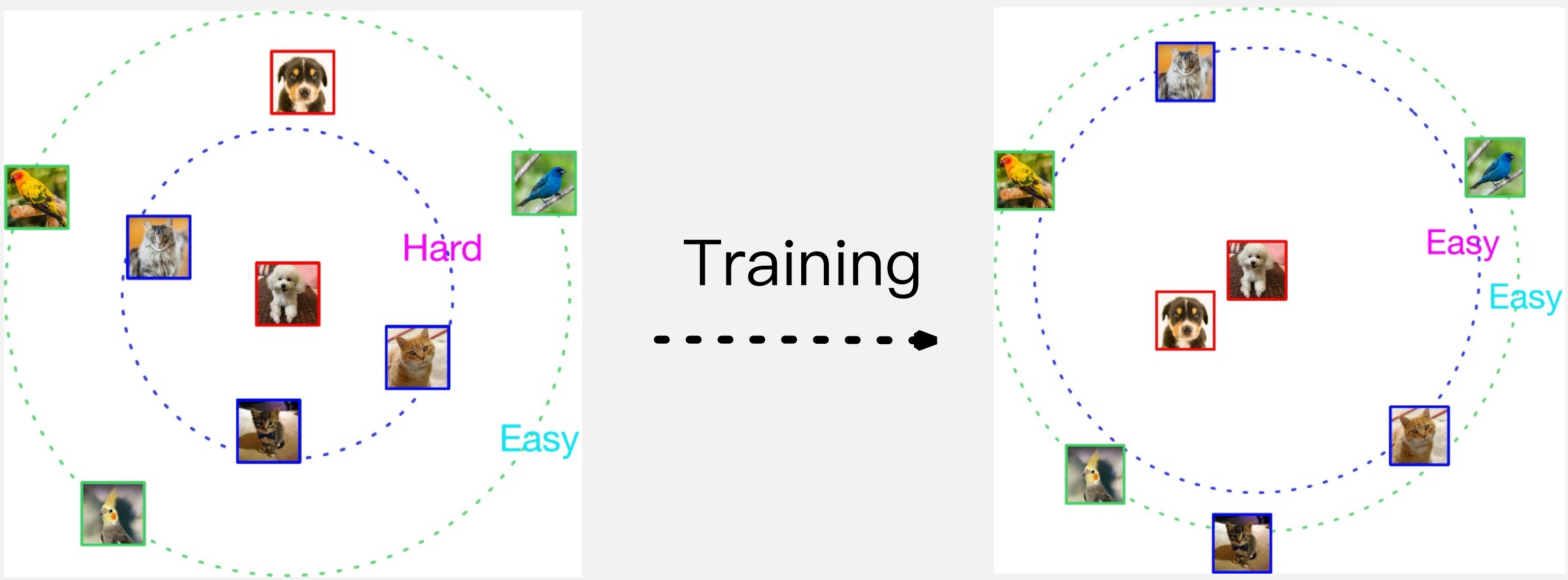
Experiments Results

- The Stanford Online Products dataset

Method	NMI	F_1	R@1	R@10	R@100
Triplet [36,18]	86.5	20.2	54.9	71.5	85.2
DVML+Triplet	**89.0**	**31.1**	**66.5**	**82.3**	**91.8**
N-pair [23]	87.9	27.1	66.4	82.9	92.1
DVML+N-pair	**90.2**	**37.1**	**70.0**	**85.1**	**93.7**
Contrastive [7]	83.5	10.4	37.4	52.7	69.4
Lifted [25]	88.4	30.6	65.2	81.3	91.7
Angular [33]	87.7	26.4	66.8	82.8	92.0
Triplet_2+DWS [37]	89.0	31.1	66.8	82.0	91.0
DVML+Triplet_2+DWS	**90.8**	**37.2**	**70.2**	**85.2**	**93.8**
HDC [40]	-	-	69.5	84.4	92.8
Proxy-NCA [16]	-	-	73.7	-	-

Hardness-Aware Deep Metric Learning

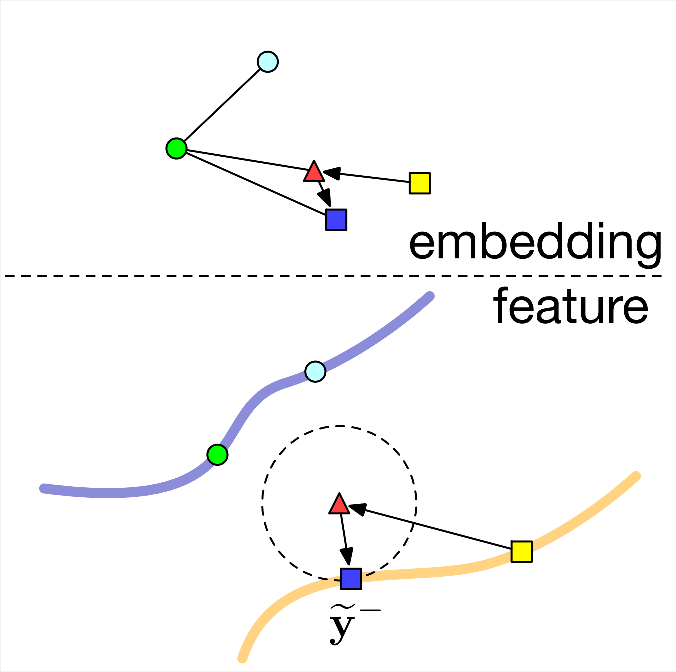
- Hard negative mining is widely applied in DML
- Fewer and fewer hard negatives as training goes!

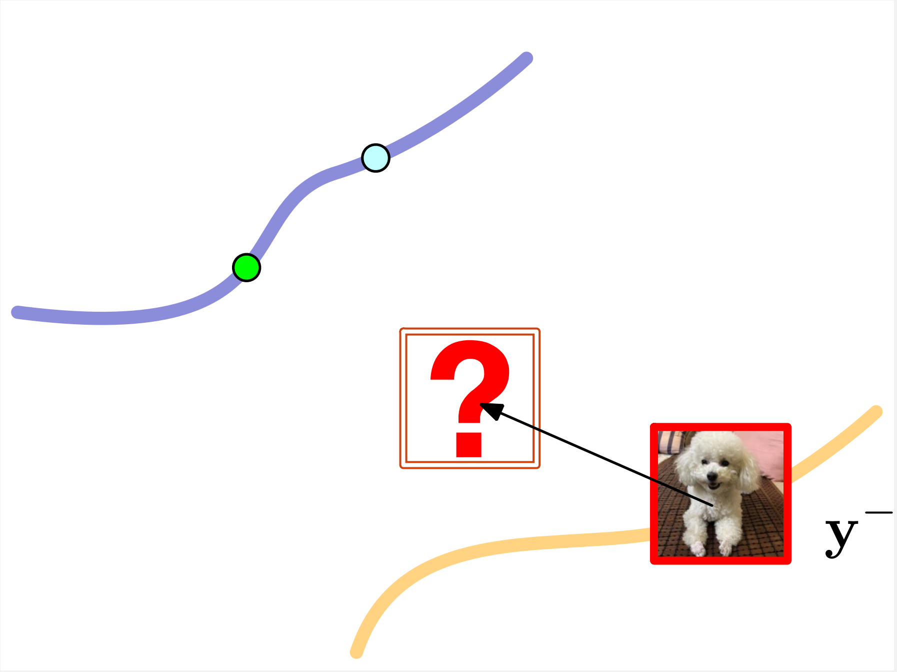


- Wenzhao Zheng, Zhaodong Chen, **Jiwen Lu***, Jie Zhou, Hardness-Aware Deep Metric Learning, **CVPR**, 2019.

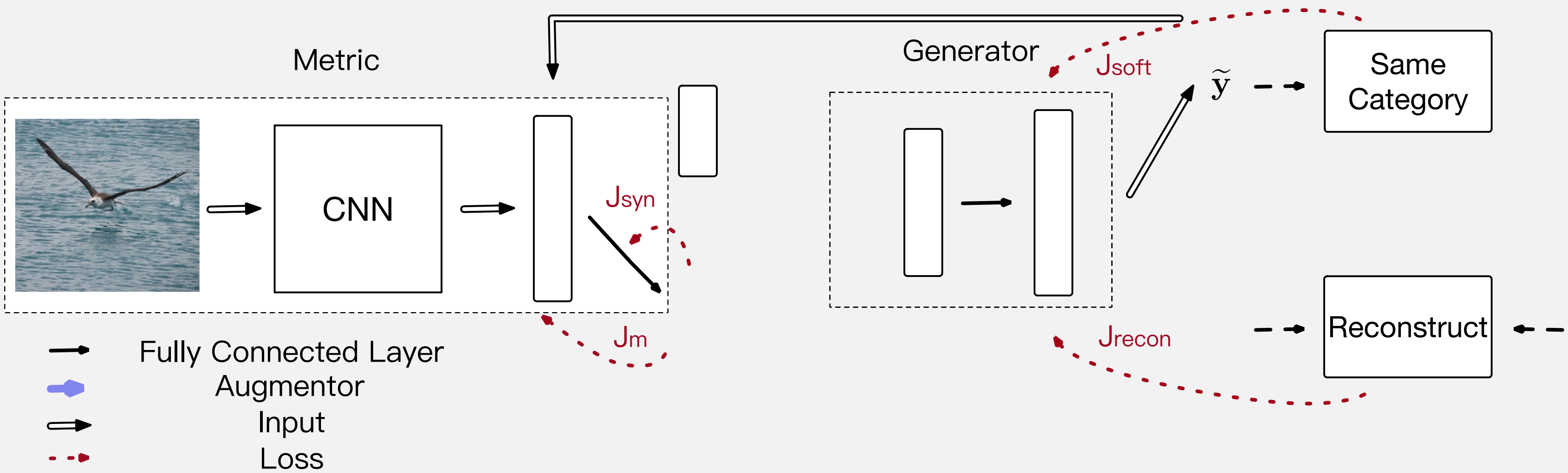
Hardness-Aware Deep Metric Learning

- HDML: Generate hardness-aware synthetics that are adaptive to the current status of the metric





Hardness-Aware Deep Metric Learning



Hardness-Aware Deep Metric Learning

- Generator objective:

$$\begin{aligned} J_{gen} &= J_{recon} + \lambda J_{soft} \\ &= c(\mathbf{Y}, \mathbf{Y}') + \lambda J_{soft}(\widetilde{\mathbf{Y}}, \mathbf{L}) \\ &= \sum_{\substack{\mathbf{y} \in \mathbf{Y} \\ \mathbf{y}' \in \mathbf{Y}'}} ||\mathbf{y} - \mathbf{y}'||^2 + \lambda \sum_{\substack{\widetilde{\mathbf{y}} \in \widetilde{\mathbf{Y}} \\ l \in \mathbf{L}}} j_{soft}(\widetilde{\mathbf{y}}, l) \end{aligned}$$

- Metric objective:

$$\begin{aligned} J_{metric} &= e^{-\frac{\beta}{J_{gen}}} J_m + (1 - e^{-\frac{\beta}{J_{gen}}}) J_{syn} \\ &= e^{-\frac{\beta}{J_{gen}}} J(\mathbf{T}) + (1 - e^{-\frac{\beta}{J_{gen}}}) J(\widetilde{\mathbf{T}}) \end{aligned}$$

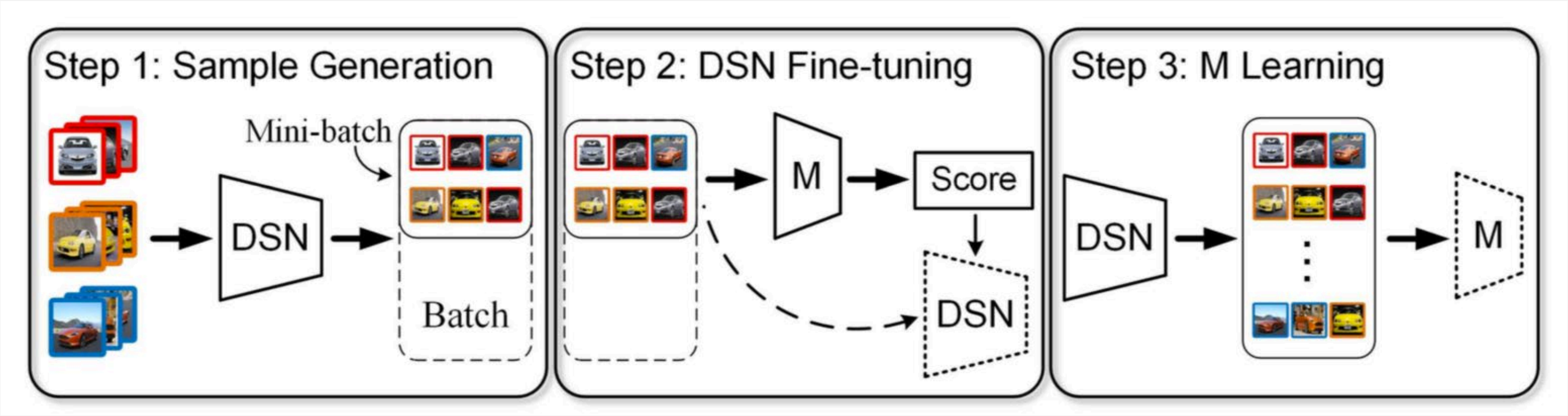
Experimental Results

- The Cars196 dataset

Method	NMI	F_1	R@1	R@2	R@4	R@8
Contrastive	42.3	10.5	27.6	38.3	51.0	63.9
DDML	41.7	10.9	32.7	43.9	56.5	68.8
Lifted	57.8	25.1	59.9	70.4	79.6	87.0
Angular	62.4	31.8	71.3	80.7	87.0	91.8
Triplet	52.9	17.9	45.1	57.4	69.7	79.2
Triplet hard	55.7	22.4	53.2	65.4	74.3	83.6
DAML (Triplet)	56.5	22.9	60.6	72.5	82.5	89.9
HDML (Triplet)	**59.4**	**27.2**	**61.0**	**72.6**	**80.7**	**88.5**
N-pair	62.7	31.8	68.9	78.9	85.8	90.9
DAML (N-pair)	66.0	36.4	75.1	83.8	89.7	93.5
HDML (N-pair)	**69.7**	**41.6**	**79.1**	**87.1**	**92.1**	**95.5**

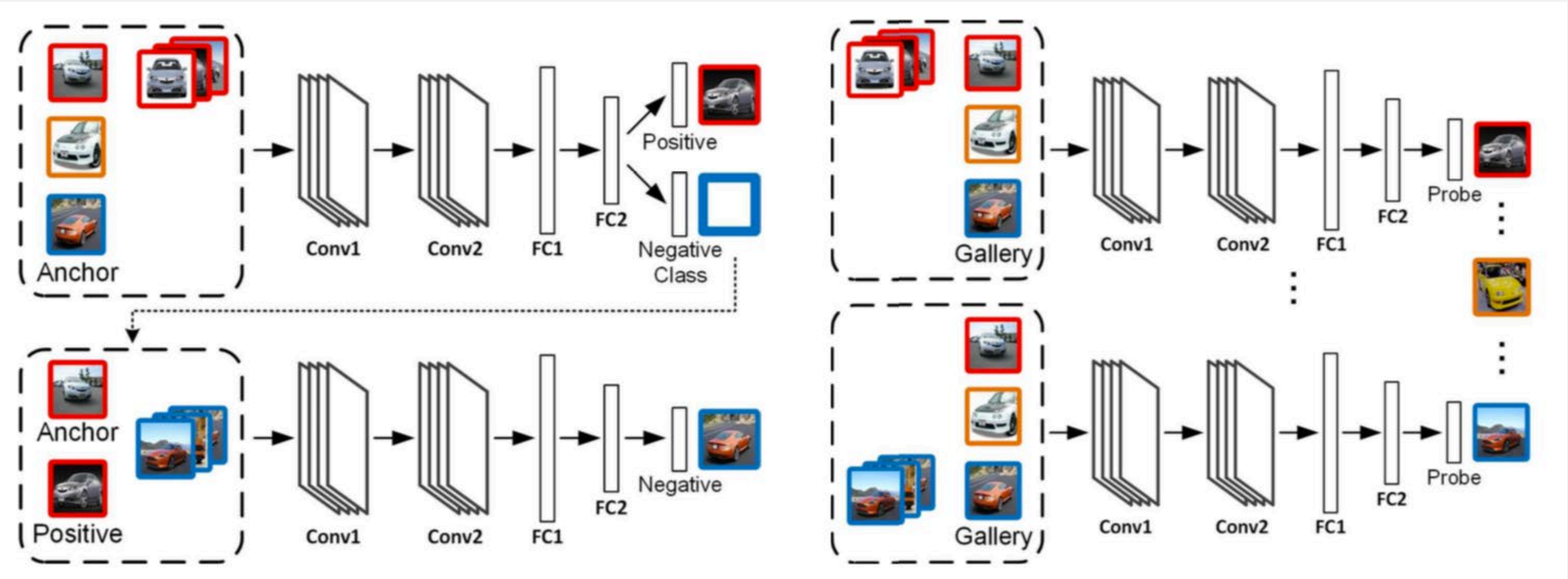
Deep embedding learning with discriminative sampling policy

- Existing methods sample training data by searching which is inefficient and easy to fall into local optimum
- DE-DSP: Learn a deep sampler to adaptively select samples beneficial for training



- Yueqi Duan, Lei Chen, **Jiwen Lu***, Jie Zhou, Deep embedding learning with discriminative sampling policy, **CVPR**, 2019.

Deep embedding learning with discriminative sampling policy



Deep sampler network for triplet embedding

Deep sampler network for N-pair embedding

Experiments Results

- The CUB-200-2011 dataset
- The Cars196 dataset

Method	CUB-200-2011					Cars196				
	NMI	F_1	R@1	R@2	R@4	NMI	F_1	R@1	R@2	R@4
DDML	47.3	13.1	31.2	41.6	54.7	41.7	10.9	32.7	43.9	56.5
Lifted	56.4	22.6	46.9	59.8	71.2	57.8	25.1	59.9	70.4	79.6
Clustering	59.2	-	48.2	61.4	71.8	59.0	-	58.1	70.6	80.3
Angular	61.0	30.2	53.6	65.0	75.3	62.4	31.8	71.3	80.7	87.0
DAML	61.3	29.5	52.7	65.4	75.5	66.0	36.4	75.1	83.8	89.7
Triplet	49.8	15.0	35.9	47.7	59.1	52.9	17.9	45.1	57.4	69.7
Semi-hard (Triplet)	50.3	16.4	37.9	50.4	63.0	53.3	18.5	52.4	65.2	75.1
DE-DSP (Triplet)	**53.7**	**19.8**	**41.0**	**53.2**	**64.8**	**55.0**	**22.3**	**59.3**	**71.3**	**81.3**
N-pair	60.2	28.2	51.9	64.3	74.9	62.7	31.8	68.9	78.9	85.8
DE-DSP (N-pair)	**61.7**	**30.5**	**53.6**	**65.5**	**76.9**	**64.4**	**33.3**	**72.9**	**81.6**	**88.8**

Part 5: Conclusion and Future Directions

Summary

- Learning effective distance metrics can better measure the similarity of samples. Hence, better visual analysis performance can be obtained.

- Different deep learning strategies are developed for different recognition tasks with different settings. Improved performance can be obtained when suitable metric learning methods are designed and employed.

- Sampling plays an equal important role with the loss function in deep metric learning

Future Directions

- **Scalability:** large-scale metric learning
  - Online learning
  - Batch based learning
- **New settings:**
  - Deep metric learning for ranking
  - Multi-task deep metric learning
  - Deep metric learning for structured data
  - Multi-modal metric learning
- **Robustness:** metric learning with noisy/missing labels
- **Unsupervised deep metric learning:** Mahalanobis deep metric learning for clustering

References

- Junlin Hu**, Jiwen Lu*,** and Yap-Peng Tan, Sharable and individual multi-view metric learning, *TPAMI*, vol. 40, no. 9, pp. 2281-2288, 2018.
- Hao Liu, **Jiwen Lu*,** Jianjiang Feng, and Jie Zhou, Two-stream transformer networks for video-based face alignment, *TPAMI*, vol. 40, no. 9, pp. 2281-2288, 2018.
- Hao Liu, **Jiwen Lu*,** Jianjiang Feng, and Jie Zhou, Ordinal deep learning for facial age estimation, *TCSVT,* 2018, accepted.
- Yueqi Duan, **Jiwen Lu*,** Jianjiang Feng, and Jie Zhou, Deep localized metric learning, T*CSVT,* 2018, accepted.
- Hao Liu, **Jiwen Lu*,** Jianjiang Feng, and Jie Zhou, Label-sensitive deep metric learning for facial age estimation, *TIFS*, vol. 13, no. 2, pp. 292-305, 2018.
- **Jiwen Lu,** Junlin Hu, and Jie Zhou, Deep metric learning for visual understanding: an overview of recent advances, *IEEE Signal Processing Magazine*, vol. 34, no. 6, pp. 76-84, 2017.
- **Jiwen Lu,** Junlin Hu, and Yap-Peng Tan, Discriminative deep metric learning for face and kinship verification, *TIP*, vol. 26, no. 9, pp. 4042-4054, 2017.
- Hao Liu, **Jiwen Lu*,** Jianjiang Feng, and Jie Zhou, Learning deep sharable and structural detectors for face alignment, *TIP*, vol. 26, no. 4, pp. 1666-1678, 2017.
- Junlin Hu, **Jiwen Lu***, Yap-Peng Tan, and Jie Zhou, Deep transfer metric learning, *TIP*, vol. 25, no. 12, pp. 5576-5588, 2016.

References

- Junlin Hu, **Jiwen Lu***, and Yap-Peng Tan, Deep metric learning for visual tracking, *TCSVT*, vol. 26, no. 11, 2056-2068, 2016.
- **Jiwen Lu**, Gang Wang, and Jie Zhou, Simultaneous feature and dictionary learning for image set based face recognition, *TIP*, vol. 26, no. 8, pp. 4042-4054, 2017.
- Anran Wang, **Jiwen Lu**, Jianfei Cai, Tat-Jen Cham, and Gang Wang, Large-margin multi-modal deep learning for RGB-D object recognition, *TMM*, vol. 17, no. 11, pp. 1887-1898, 2015.
- **Jiwen Lu**, Gang Wang, Weihong Deng, Pierre Moulin, and Jie Zhou, Multi-manifold deep metric learning for image set classification, *CVPR*, 2015.
- Junlin Hu, **Jiwen Lu***, and Yap-Peng Tan, Deep transfer metric learning, *CVPR*, 2015.
- Junlin Hu, **Jiwen Lu***, and Yap-Peng Tan, Discriminative deep metric learning for face verification in the wild, *CVPR*, 2014.
- Kilian Q. Weinberger, John Blitzer, Lawrence K. Saul: Distance Metric Learning for Large Margin Nearest Neighbor Classification, *NIPS*, 2005.
- Jason V. Davis, Brian Kulis, Prateek Jain, Suvrit Sra, Inderjit S. Dhillon: Information-theoretic metric learning, *ICML*, 2007.
- Yueqi Duan, Lei Chen, **Jiwen Lu**, and Jie Zhou, Deep embedding learning with discriminative sampling policy, *CVPR*, 2019.

References

- Yueqi Duan, **Jiwen Lu***, Ziwei Wang, Jianjiang Feng, and Jie Zhou, Learning deep binary descriptor with multi-quantization, *CVPR*, 2017.
- Yueqi Duan, **Jiwen Lu*,** Ziwei Wang, Jianjiang Feng, and Jie Zhou, Learning deep binary descriptor with multi-quantization, *TPAMI*, 2018, accepted.
- Yueqi Duan, Ziwei Wang, **Jiwen Lu***, Xudong Lin, and Jie Zhou, GraphBit: bitwise interaction mining via deep reinforcement learning, *CVPR*, 2018.
- Venice Erin Liong, **Jiwen Lu***, Gang Wang, Pierre Moulin, and Jie Zhou, Deep hashing for compact binary codes learning, *CVPR*, 2015.
- **Jiwen Lu**, Venice Erin Liong, and Jie Zhou, Deep hashing for scalable image search, *TIP*, vol. 26, no. 5, pp. 2352-2367, 2017.
- Zhixiang Chen, **Jiwen Lu***, Jianjiang Feng, and Jie Zhou, Nonlinear discrete hashing, *TMM,* vol. 19, no. 1, pp. 123-135, 2017.
- Zhixiang Chen, **Jiwen Lu***, Jianjiang Feng, and Jie Zhou, Nonlinear sparse hashing, *TMM,* vol. 19, no. 9, pp. 1996-2009, 2017.
- Zhixiang Chen, Xin Yuan, **Jiwen Lu***, Qi Tian, and Jie Zhou, Deep hashing by discrepancy minimization, *CVPR*, 2018.
- Kevin Lin, **Jiwen Lu***, Chu-Song Chen, and Jie Zhou, Learning compact binary descriptors with unsupervised deep neural networks, *CVPR*, 2016.
- Kevin Lin, **Jiwen Lu***, Chu-Song Chen, Jie Zhou, and Ming-Ting Sun, Unsupervised deep learning of compact binary descriptors, *TPAMI*, 2018, accepted.

References

- Xin Yuan, Liangliang Ren, **Jiwen Lu***, and Jie Zhou, Relaxation-free deep hashing via policy gradient, *ECCV*, 2018.
- Venice Erin Liong, **Jiwen Lu***, Yap-Peng Tan, and Jie Zhou, Deep video hashing, *TMM*, vol. 19, no. 6, pp. 1209-1219, 2017.
- Zhixiang Chen, **Jiwen Lu***, Jianjiang Feng, and Jie Zhou, Nonlinear structural hashing for scalable video search, *TCSVT*, vol. 28, no. 6, pp. 1421-1433, 2018.
- Venice Erin Liong, **Jiwen Lu*,** Yap-Peng Tan, and Jie Zhou, Cross-modal deep variational hashing, *ICCV*, 2017.
- Florian Schroff, Dmitry Kalenichenko, and James Philbin, Facenet: a unified embedding for face recognition and clustering, *CVPR*, 2015.
- Chao-Yuan Wu, R. Manmotha, Alexander J. Smola, and Philipp Krahenbuhl, Sampling matters in deep embedding learning, *ICCV*, 2017.
- Ben Harwood, Vijay Kumar B G, Gustavo Carneiro, Ian Reid, and Tom Drummond, Smart mining for deep metric learning, *ICCV*, 2017.
- Yueqi Duan, Wenzhao Zheng, Xudong Lin, **Jiwen Lu***, and Jie Zhou, Deep adversarial metric learning, *CVPR*, 2018.
- Xudong Lin, **Yueqi Duan**, Qiyuan Dong, **Jiwen Lu***, and Jie Zhou, Deep variational metric learning, *ECCV*, 2018.
- Wenzhao Zheng , Zhaodong Chen, **Jiwen Lu***, and Jie Zhou, Hardness-aware deep metric learning, CVPR, 2019.