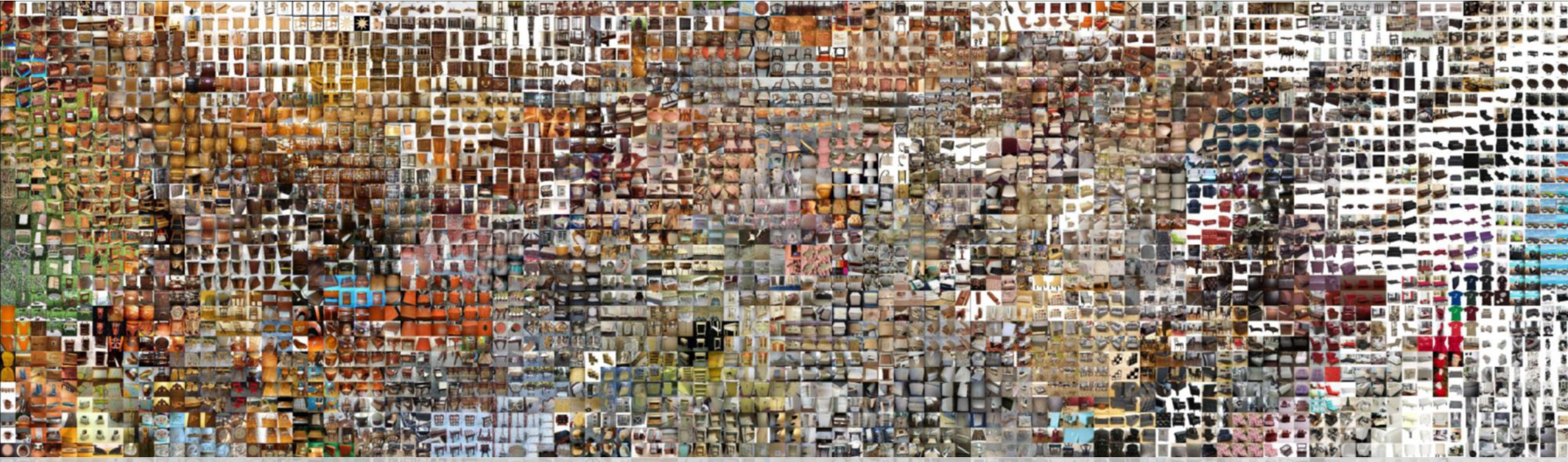


Proxy Anchor Loss for Deep Metric Learning

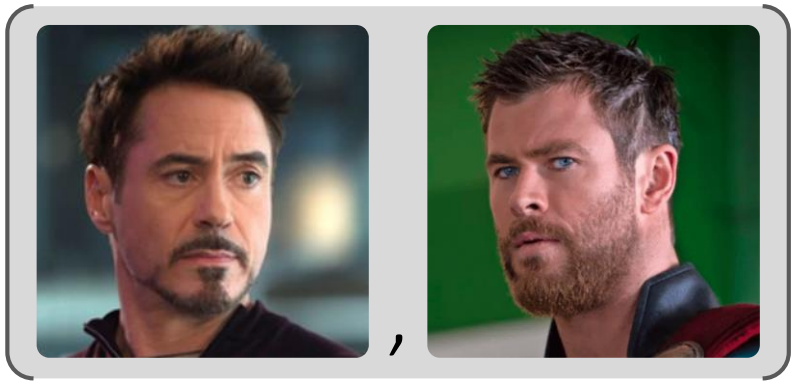
Sungyeon Kim Dongwon Kim Minsu Cho Suha Kwak

{tjddus9597, kdwon, mscho, suha.kwak}@postech.ac.kr

POSTECH

Metric Learning

How much similar/dissimilar semantically?

D  $<$ D 

Metric: Function that quantifies a distance

Metric Learning: Learning a metric from a set of data

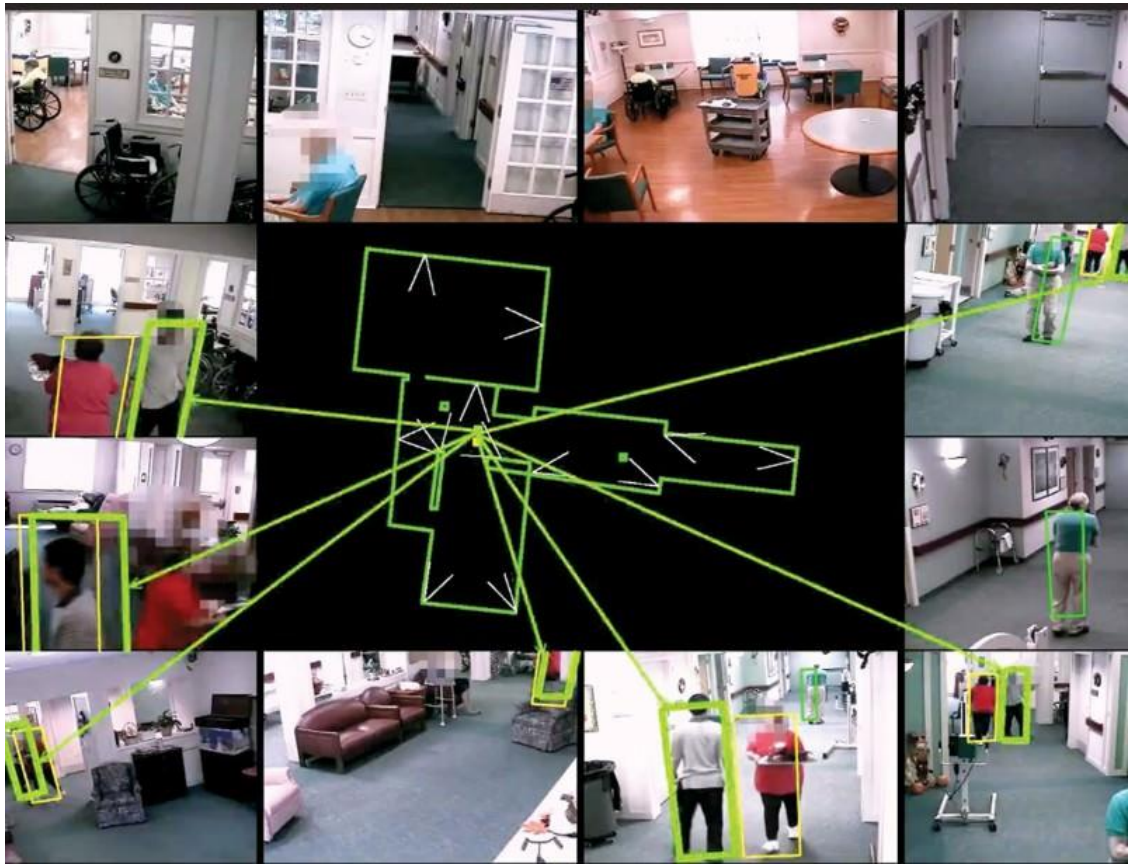
Applications

Content-based image retrieval

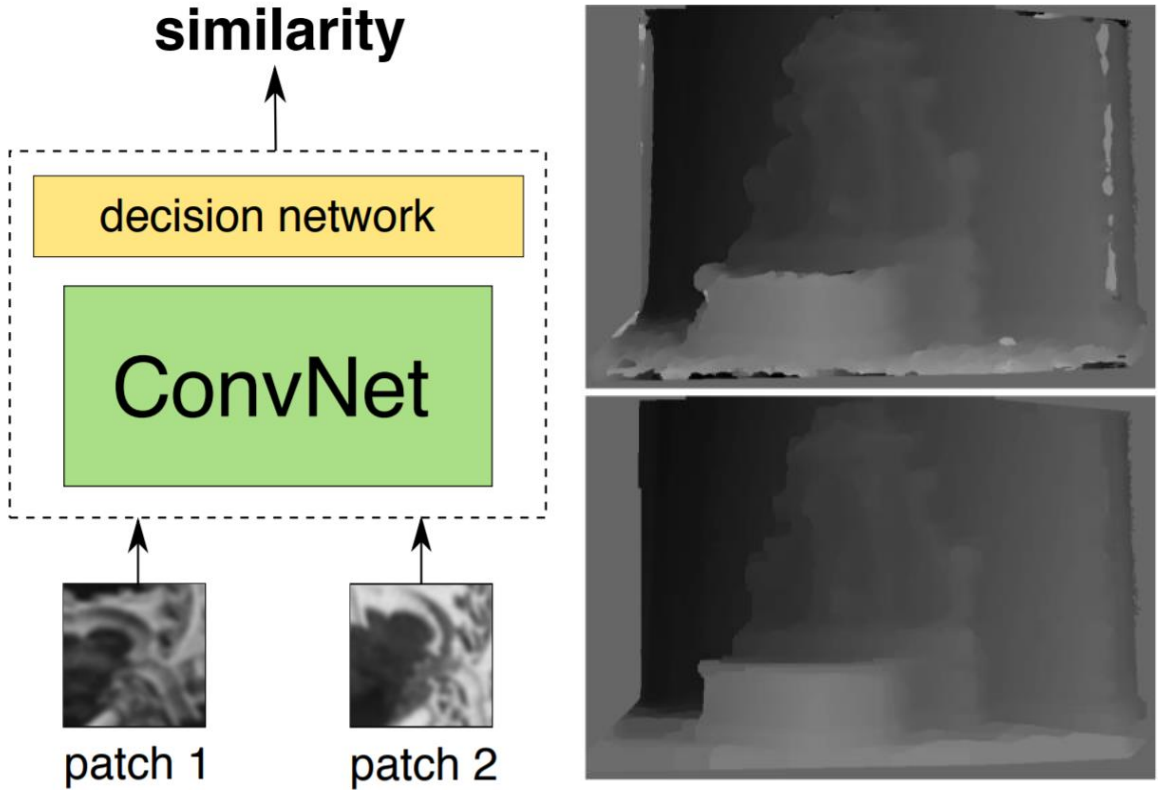
Face verification/identification[1]

[1] FaceNet: A unified embedding for face recognition and clustering, CVPR 2015

Applications



Person re-identification[2]

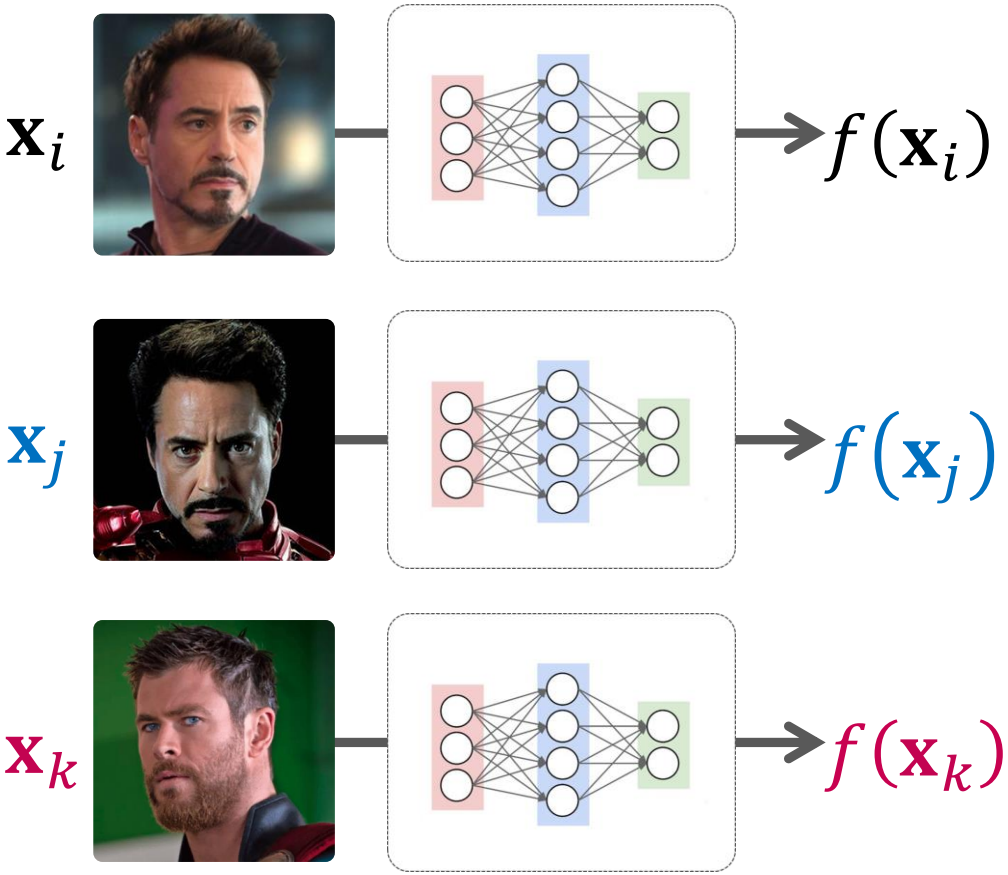


Patch matching/stereo imaging[3]

[2] Beyond triplet loss: a deep quadruplet network for person re-identification, CVPR 2017
[3] Learning to compare image patches via convolutional neural networks, CVPR 2015

Deep Metric Learning

Learning a deep embedding network f so that semantically similar images are closely grouped together



Distance = Semantic dissimilarity

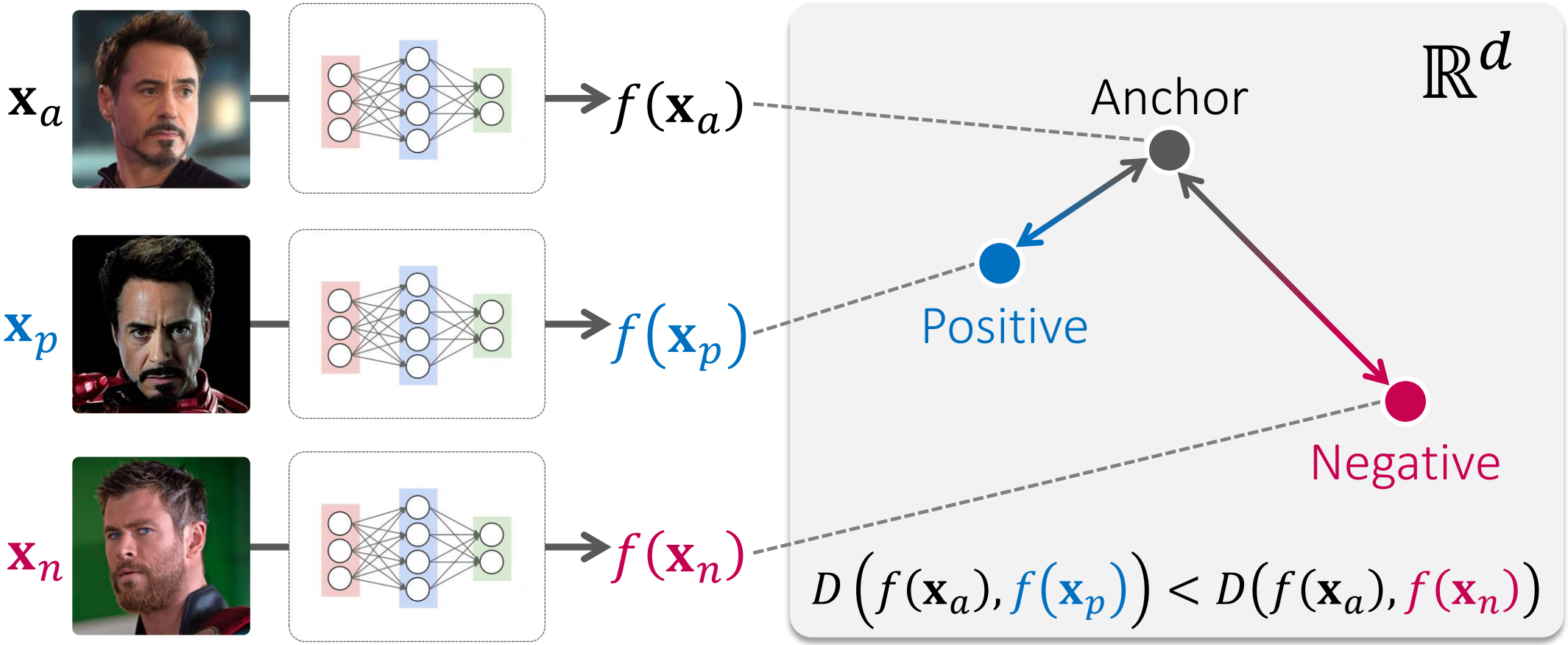
This quality of the embedding space is mainly determined by ***loss functions*** *used for training the network.*

$\mathbb{R}^d$

Examples of Metric Learning Losses

- Triplet rank loss[1]

$$\ell_{\text{tri}}(a,p,n) = \left[D\left(f_a, f_p\right) - D\left(f_a, f_n\right) + \delta\right]_+$$

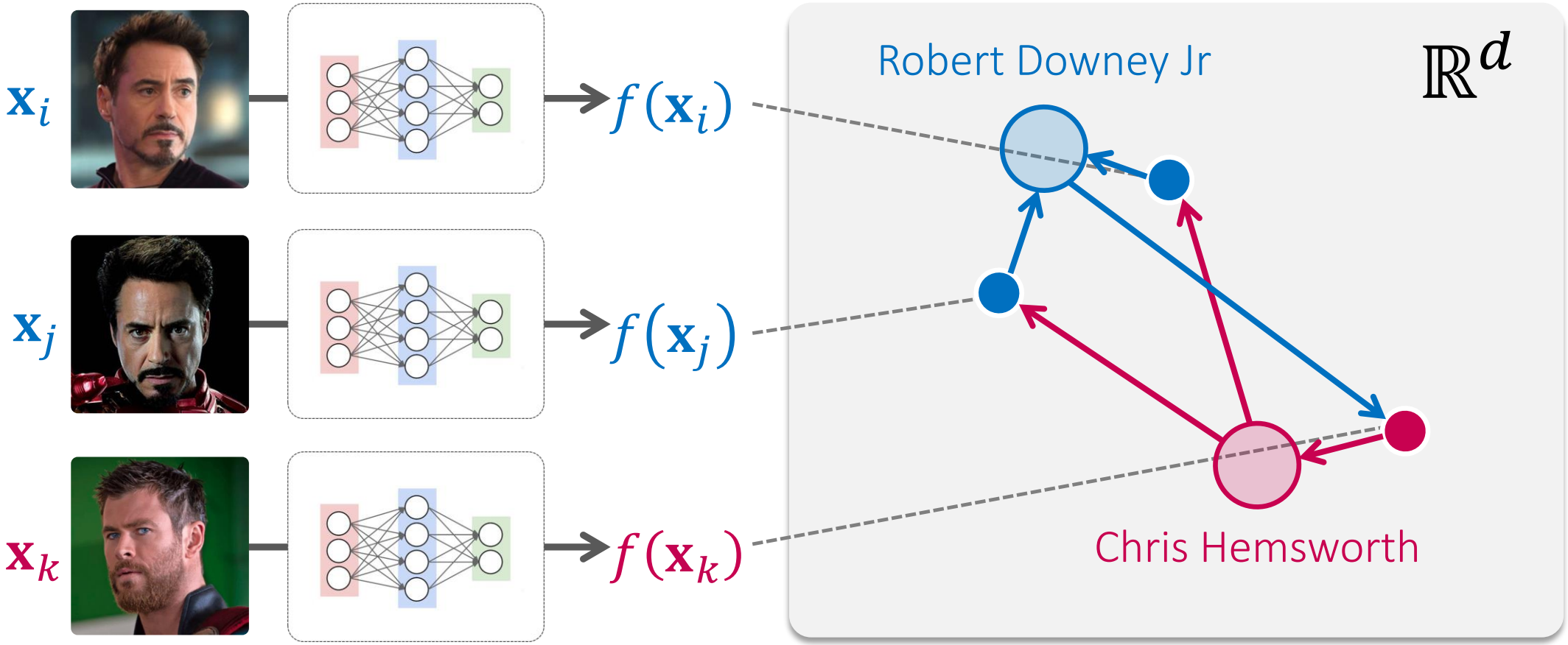


[1] FaceNet: A unified embedding for face recognition and clustering, CVPR 2015

Examples of Metric Learning Losses

- Proxy NCA loss[6]

$$\ell_{\text{proxyNCA}}(B) = \sum_{i \in B} \left\{ D(f_i, p^+) - \log \sum_{p^- \in P^-} \exp\left(-D(f_i, p^-)\right) \right\}$$



[6] No fuss distance metric learning using proxies, ICCV 2017

Two Categories of Existing Losses

- Pair-based losses
  - (+) Exploiting *data-to-data relations*, fine-grained relations between data
  - (–) Prohibitively high training complexity

  - Examples
    - Contrastive loss[4]

$$\ell_{\text{ctr}}(i,j) = y_{ij} D(f_i, f_j)^2 + (1 - y_{ij})[\delta - D(f_i, f_j)]_+^2$$

    - Triplet rank loss[1]

$$\ell_{\text{tri}}(a,p,n) = [D(f_a, f_p) - D(f_a, f_n) + \delta]_+$$

    - N-pair loss[5]

$$\ell_{\text{NP}}(a,p,n_1,\dots,n_{N-1}) = \log\left(1 + \sum_{i=1}^{N-1} \exp\left(D(f_a, f_p) - D(f_a, f_{n_i})\right)\right)$$

[1] FaceNet: A unified embedding for face recognition and clustering, CVPR 2015
[4] Learning a similarity metric discriminatively with application to face verification, CVPR 2005
[5] Improved deep metric learning with multi-class N-pair loss objective, NeurIPS 2016

Two Categories of Existing Losses

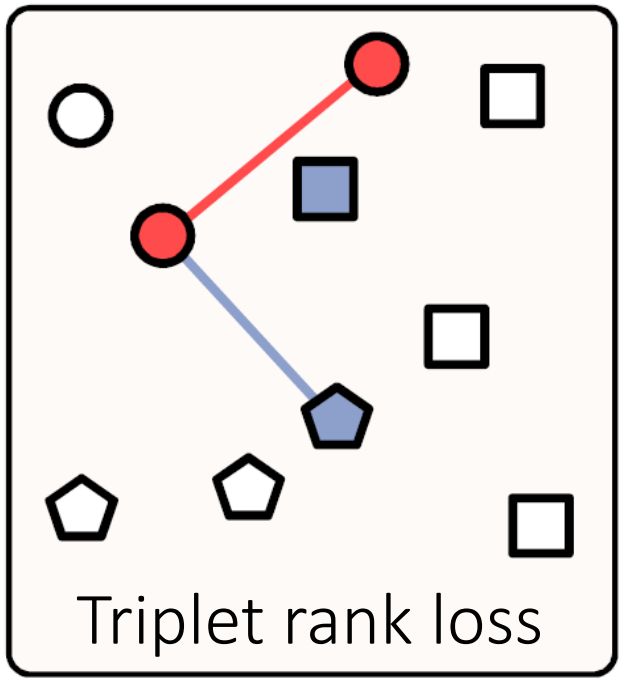
- Proxy-based losses
  - Proxy
    - Representative of a subset of training data
    - Learned as a part of the network parameters
  - Taking each data point as an anchor and associating it with proxies
  - (+) Lower training complexity, faster convergence in general
  - (+) More robust against label noises and outliers
  - (–) Leveraging impoverished data-to-proxy relations only
  - Example: Proxy-NCA loss[6]

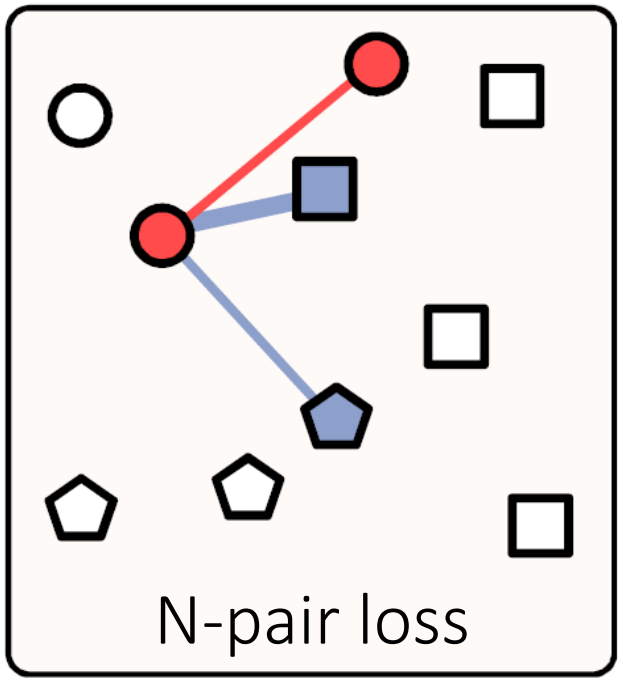
$$\ell_{\text{proxyNCA}}(B) = -\sum_{i \in B} \log \frac{\exp\left(-D(f_i, p^+)\right)}{\sum_{p^- \in P^-} \exp\left(-D(f_i, p^-)\right)}$$

[6] No fuss distance metric learning using proxies, ICCV 2017

Two Categories of Existing Losses

Pair-based losses



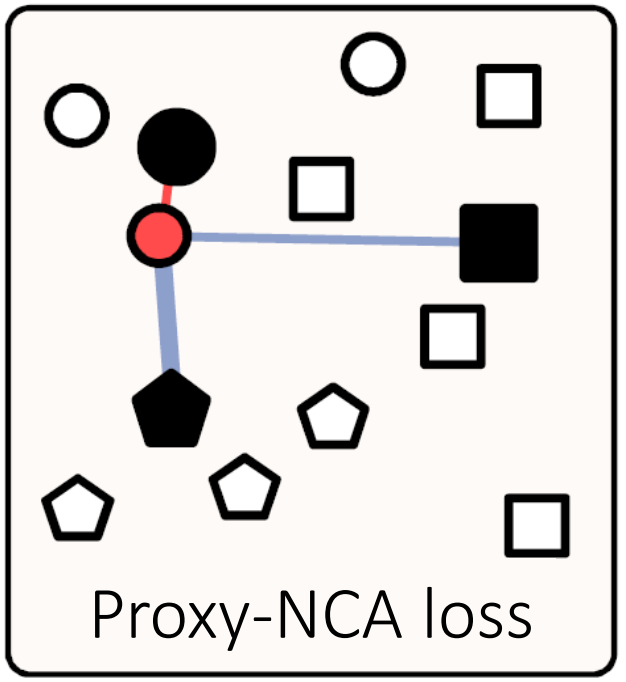


"Data-to-data relations"

Rich and fine-grained

Demanding high training complexity

Proxy-based losses



"Data-to-proxy relations"

Reducing training complexity

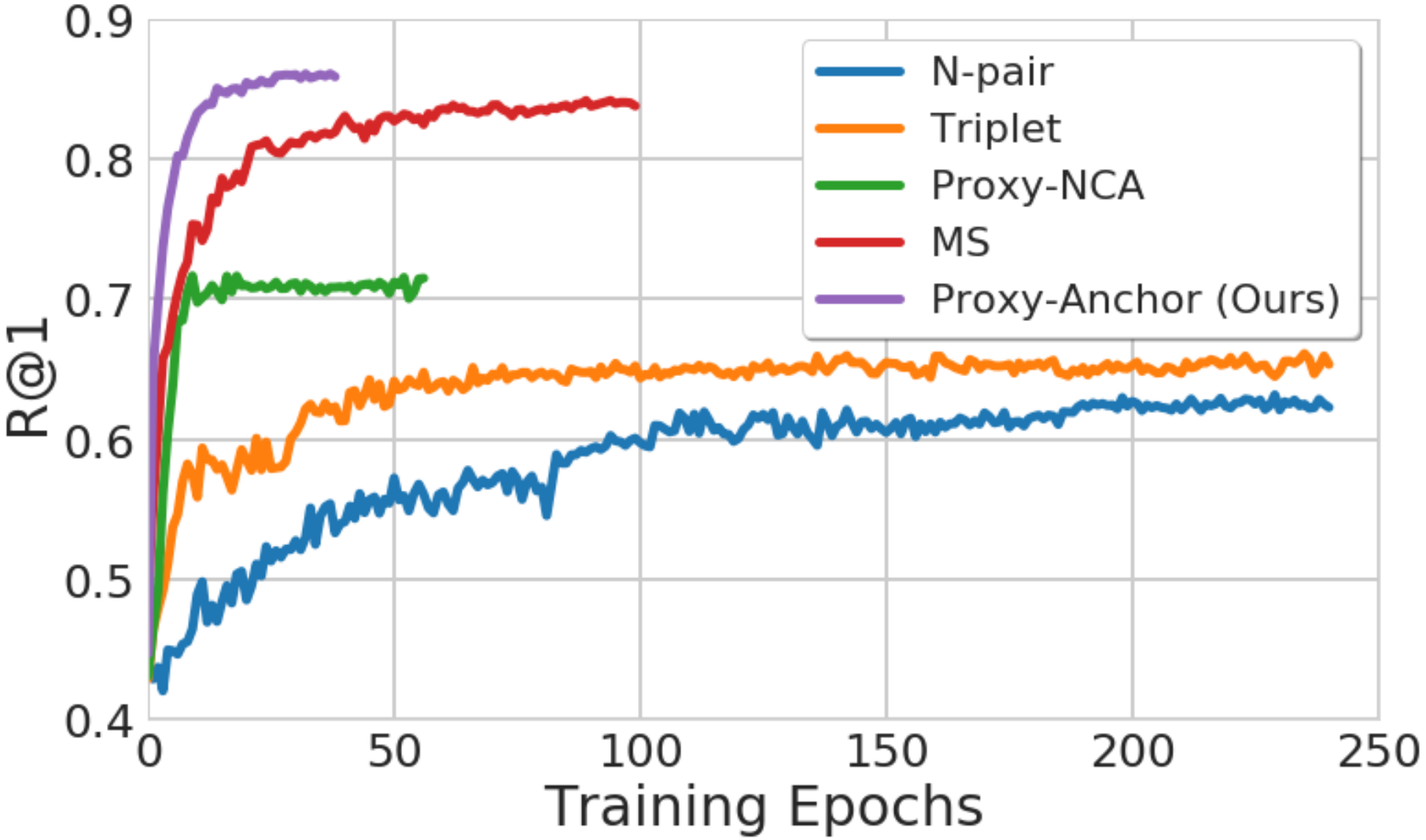
Impoverished information

Our Method

- A new proxy-based loss called *proxy anchor loss*
  - Taking only advantages of both categories
  - Overcoming their limitations
- How it works
  - Using a proxy as an anchor, and associating it with all data in a batch
  - Fast convergence thanks to the use of proxies
  - Taking data-to-data relations into account by allowing data points to interact with each other during training
- Results
  - State-of-the-art performance
  - Fastest convergence (on the Cars-196 dataset)

Our Method

Recall@1 vs. training epochs on the Cars-196 dataset



Details of Proxy Anchor Loss

- Mathematical form and its interpretation

$$\ell(B) = \frac{1}{|P^+|}\sum_{p\in P^+}\log\left(1+\sum_{i\in B_p^+}\exp[-\alpha(S(f_i,p)-\delta)]\right)$$
$$+\frac{1}{|P|}\sum_{p\in P}\log\left(1+\sum_{j\in B_p^-}\exp[\alpha(S(f_j,p)+\delta)]\right)$$
$$=\frac{1}{|P^+|}\sum_{p\in P^+}\left[\mathrm{SoftPlus}\left(\underset{i\in B_p^+}{\mathrm{LSE}}\,-\alpha(S(f_i,p)-\delta)\right)\right]$$
$$+\frac{1}{|P|}\sum_{p\in P}\left[\mathrm{SoftPlus}\left(\underset{j\in B_p^-}{\mathrm{LSE}}\,\alpha(S(f_j,p)+\delta)\right)\right]$$

$S(\cdot,\cdot)$
Cosine similarity

SoftPlus
A smooth approx. of ReLU

LSE
A smooth approx. of MAX

Details of Proxy Anchor Loss

- Mathematical form and its interpretation

$$\ell(B) = \frac{1}{|P^+|}\sum_{p\in P^+}\left[\text{SoftPlus}\left(\underset{i\in B_p^+}{\text{LSE}} -\alpha(S(f_i,p)-\delta)\right)\right]$$
$$+\frac{1}{|P|}\sum_{p\in P}\left[\text{SoftPlus}\left(\underset{i\in B_p^-}{\text{LSE}}\ \alpha\big(S\big(f_j,p\big)+\delta\big)\right)\right]$$

Regarding LSE as MAX: pull p and its hardest positive example together, push p and its hardest negative example apart.

In practice pull/push all embedding vectors in the batch, but with different degrees of strength determined by their relative hardness.

Details of Proxy Anchor Loss

- Analysis on its gradients

$$\frac{\partial \ell(B)}{\partial S(f_i, p)} = \begin{cases} \frac{1}{|P^+|} \frac{-\alpha\, h_p^+(f_i)}{1 + \sum_{j \in B_p^+} h_p^+(f_j)}, & \forall i \in B_p^+, \\ \frac{1}{|P|} \frac{\alpha\, h_p^-(f_i)}{1 + \sum_{k \in B_p^-} h_p^-(f_k)}, & \forall i \in B_p^-, \end{cases}$$

where

$h_p^+(f) = \exp[-\alpha(S(f,p) - \delta)]$: Positive hardness metric

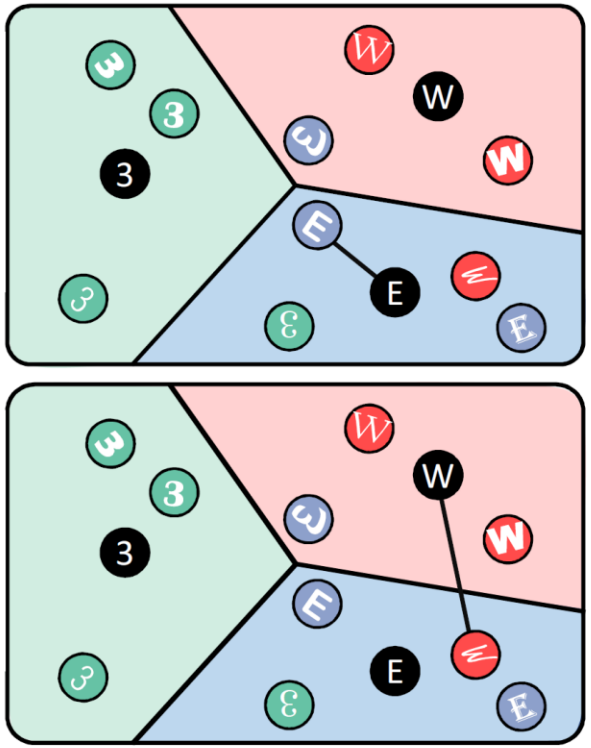
$h_p^-(f) = \exp[\alpha(S(f,p) + \delta)]$: Negative hardness metric

The gradient w.r.t. f_i is affected by other examples in the batch.
(The gradient becomes larger when f_i is harder than others.)

Comparison to Proxy NCA

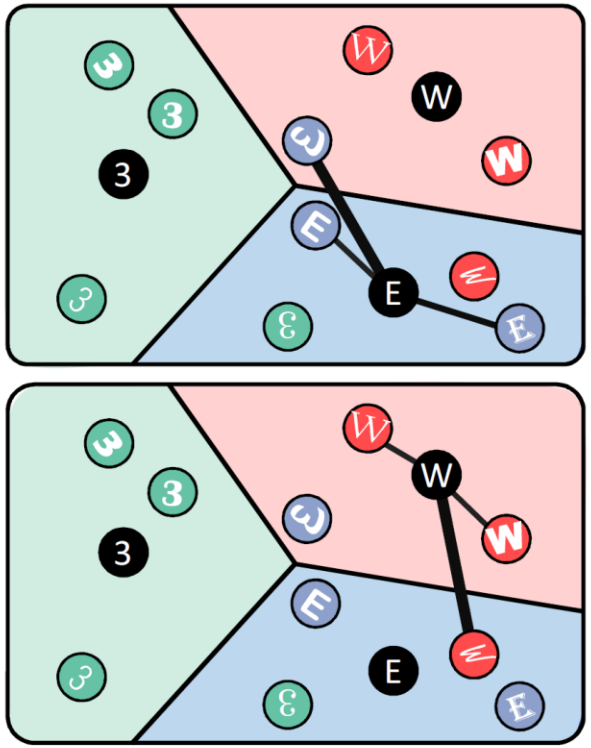
In the case of positive examples

Proxy NCA



Uniform scale
for all gradients

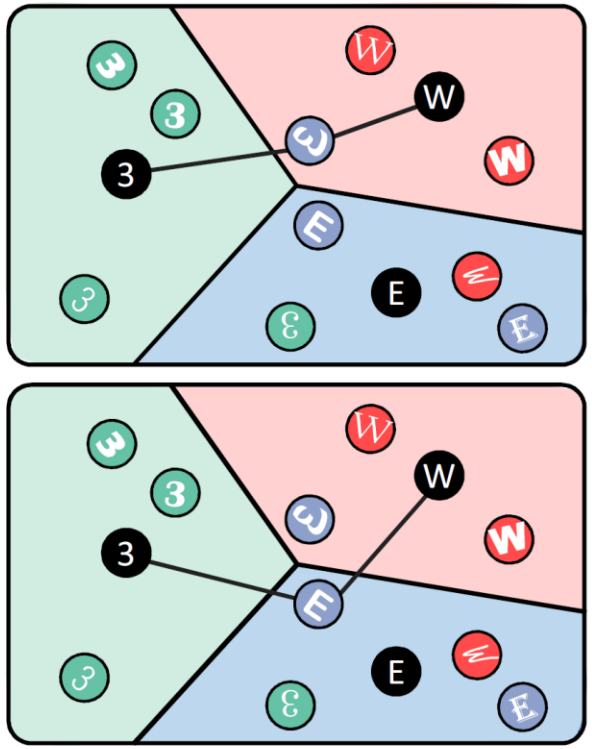
Proxy Anchor



Scales weighted by
relative hardness

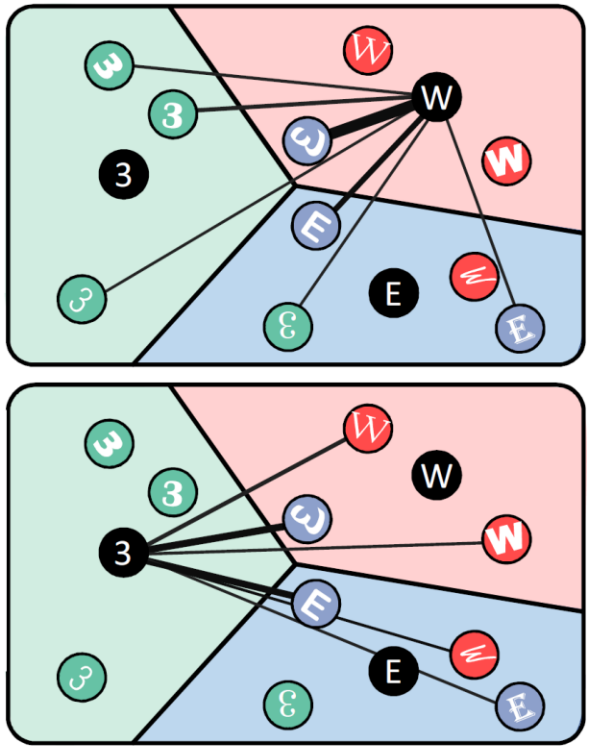
In the case of negative examples

Proxy NCA



Pushing only a small
number of data with
uniform strength

Proxy Anchor



Pushing all data with
consideration of their
distribution

Complexity Analysis

Type	Loss	Training Complexity
Proxy	Proxy Anchor	$O(MC)$
	Proxy NCA[6]	$O(MC)$
	SoftTriplet[8]	$O(MCU^2)$
Pair	Contrastive[4]	$O(M^2)$
	Triplet[1]	$O(M^3)$
	N-pair[5]	$O(M^3)$
	Lifted Structure[7]	$O(M^3)$

The same complexity, but Proxy Anchor converges faster & performs better since it considers relative hardness of data.

M: # of data

C: # of classes ($C \ll M$)

U: # of proxies per class

[1] FaceNet: A unified embedding for face recognition and clustering, CVPR 2015
[4] Learning a similarity metric discriminatively with application to face verification, CVPR 2005
[5] Improved deep metric learning with multi-class N-pair loss objective, NeurIPS 2016
[6] No fuss distance metric learning using proxies, ICCV 2017
[7] Deep metric learning via lifted structured feature embedding, CVPR 2016
[8] Softtriple loss: Deep metric learning without triplet sampling, ICCV 2019

Experiments

- Evaluation on the 4 image retrieval benchmarks
  - Caltech-UCSD Bird 200 (CUB-200-2011)
  - Cars-196
  - Stanford Online Product (SOP)
  - In-Shop Clothes Retrieval (In-Shop)
- Proxy setting: 1 proxy per class
- Image setting
  - Default: 224 X 224 (as in most previous work)
  - Larger: 256 X 256 (for comparison to HORDE[9])
- Hyper-parameters: $\alpha = 32, \delta = 10^{-1}$

[9] High-order regularizer for deep embeddings, ICCV 2019

Experiments

- Quantitative results on the CUB-200-2011 and Cars-196

Recall@K		CUB-200-2011				Cars-196			
		1	2	4	8	1	2	4	8
Clustering[64]	BN	48.2	61.4	71.8	81.9	58.1	70.6	80.3	87.8
Proxy-NCA[64]	BN	49.2	61.9	67.9	72.4	73.2	82.4	86.4	87.8
Smart Mining[64]	G	49.8	62.3	74.1	83.3	64.7	76.2	84.2	90.2
MS[64]	BN	57.4	69.8	80.0	87.8	77.3	85.3	90.5	94.2
SoftTriple[64]	BN	60.1	71.9	81.2	88.5	78.6	86.6	91.8	95.4
Proxy-Anchor[64]	BN	**61.7**	**73.0**	**81.8**	**88.8**	**78.8**	**87.0**	**92.2**	**95.5**
Margin[128]	R50	63.6	74.4	83.1	90.0	79.6	86.5	91.9	95.1
HDC[384]	G	53.6	65.7	77.0	85.6	73.7	83.2	89.5	93.8
A-BIER[512]	G	57.5	68.7	78.3	86.2	82.0	89.0	93.2	96.1
ABE[512]	G	60.6	71.5	79.8	87.4	85.2	90.5	94.0	96.1
HTL[512]	BN	57.1	68.8	78.7	86.5	81.4	88.0	92.7	95.7
RLL-H[512]	BN	57.4	69.7	79.2	86.9	74.0	83.6	90.1	94.1
MS[512]	BN	65.7	77.0	86.3	91.2	84.1	90.4	94.0	96.5
SoftTriple[512]	BN	65.4	76.4	84.5	90.4	84.5	90.7	94.5	96.9
Proxy-Anchor[512]	BN	**68.4**	**79.2**	**86.8**	**91.6**	**86.1**	**91.7**	**95.0**	**97.3**
†Contra+HORDE[512]	BN	66.3	76.7	84.7	90.6	83.9	90.3	94.1	96.3
†Proxy-Anchor[512]	BN	**71.1**	**80.4**	**87.4**	**92.5**	**88.3**	**93.1**	**95.7**	**97.5**

Experiments

- Quantitative results on the SOP (*left*) and In-Shop (*right*)

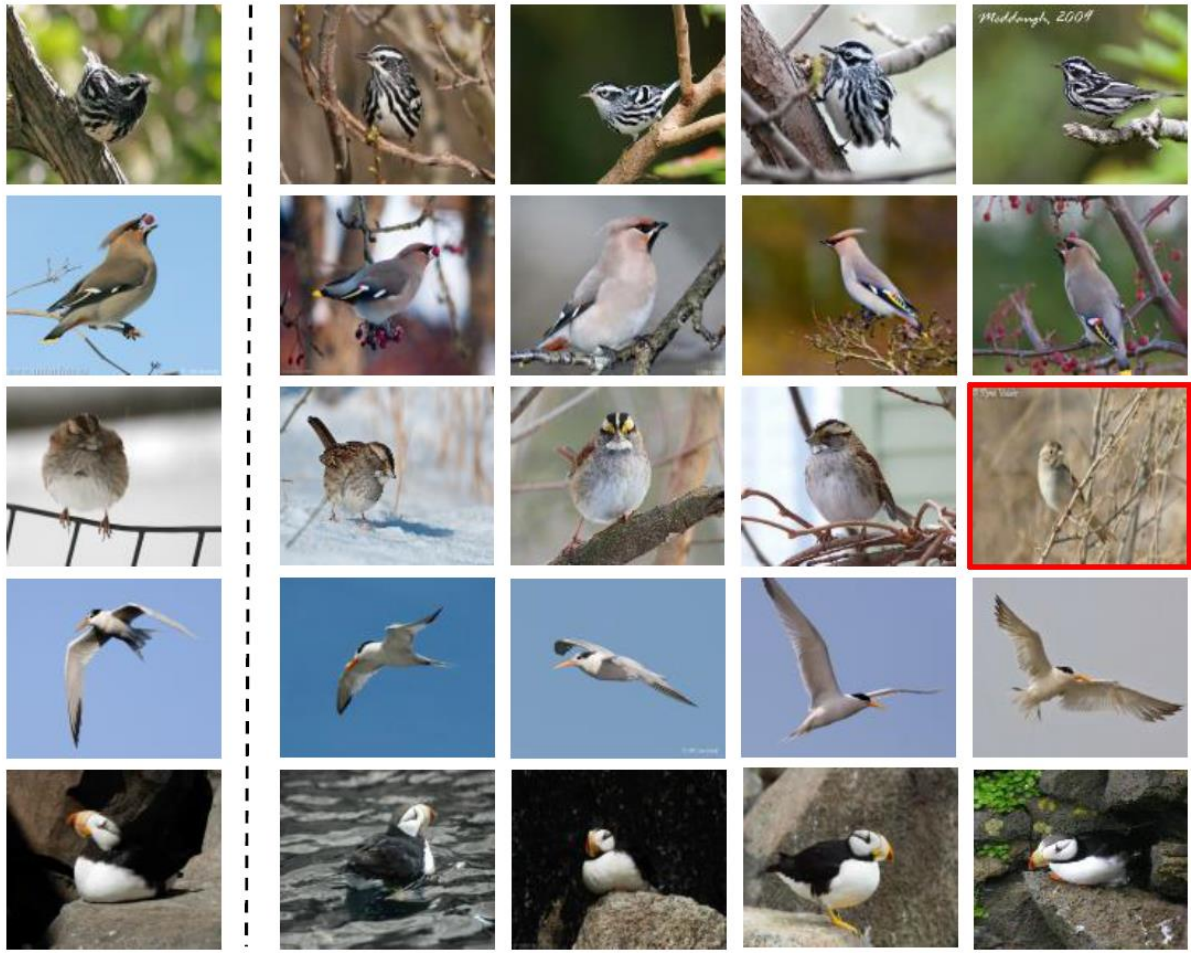
Recall@K	1	10	100	1000
Clustering^{64}	67.0	83.7	93.2	-
Proxy-NCA^{64}	73.7	-	-	-
MS^{64}	74.1	87.8	94.7	**98.2**
SoftTriple^{64}	76.3	**89.1**	**95.3**	-
Proxy-Anchor^{64}	**76.5**	89.0	95.1	**98.2**
Margin^{128}	72.7	86.2	93.8	98.0
HDC^{384}	69.5	84.4	92.8	97.7
A-BIER^{512}	74.2	86.9	94.0	97.8
ABE^{512}	76.3	88.4	94.8	98.2
HTL^{512}	74.8	88.3	94.8	98.4
RLL-H^{512}	76.1	89.1	95.4	-
MS^{512}	78.2	90.5	96.0	**98.7**
SoftTriple^{512}	78.3	90.3	95.9	-
Proxy-Anchor^{512}	**79.1**	**90.8**	**96.2**	**98.7**
†Contra+HORDE^{512}	80.1	91.3	96.2	**98.7**
†Proxy-Anchor^{512}	**80.3**	**91.4**	**96.4**	**98.7**

Recall@K	1	10	20	40
HDC^{384}	62.1	84.9	89.0	92.3
HTL^{128}	80.9	94.3	95.8	97.4
MS^{128}	88.0	97.2	98.1	98.7
Proxy-Anchor^{128}	**90.8**	**97.9**	**98.5**	**99.0**
FashionNet^{4096}	53.0	73.0	76.0	79.0
A-BIER^{512}	83.1	95.1	96.9	97.8
ABE^{512}	87.3	96.7	97.9	98.5
MS^{512}	89.7	97.9	98.5	99.1
Proxy-Anchor^{512}	**91.5**	**98.1**	**98.8**	**99.1**
†Contra+HORDE^{512}	90.4	97.8	98.4	98.9
†Proxy-Anchor^{512}	**92.6**	**98.3**	**98.9**	**99.3**

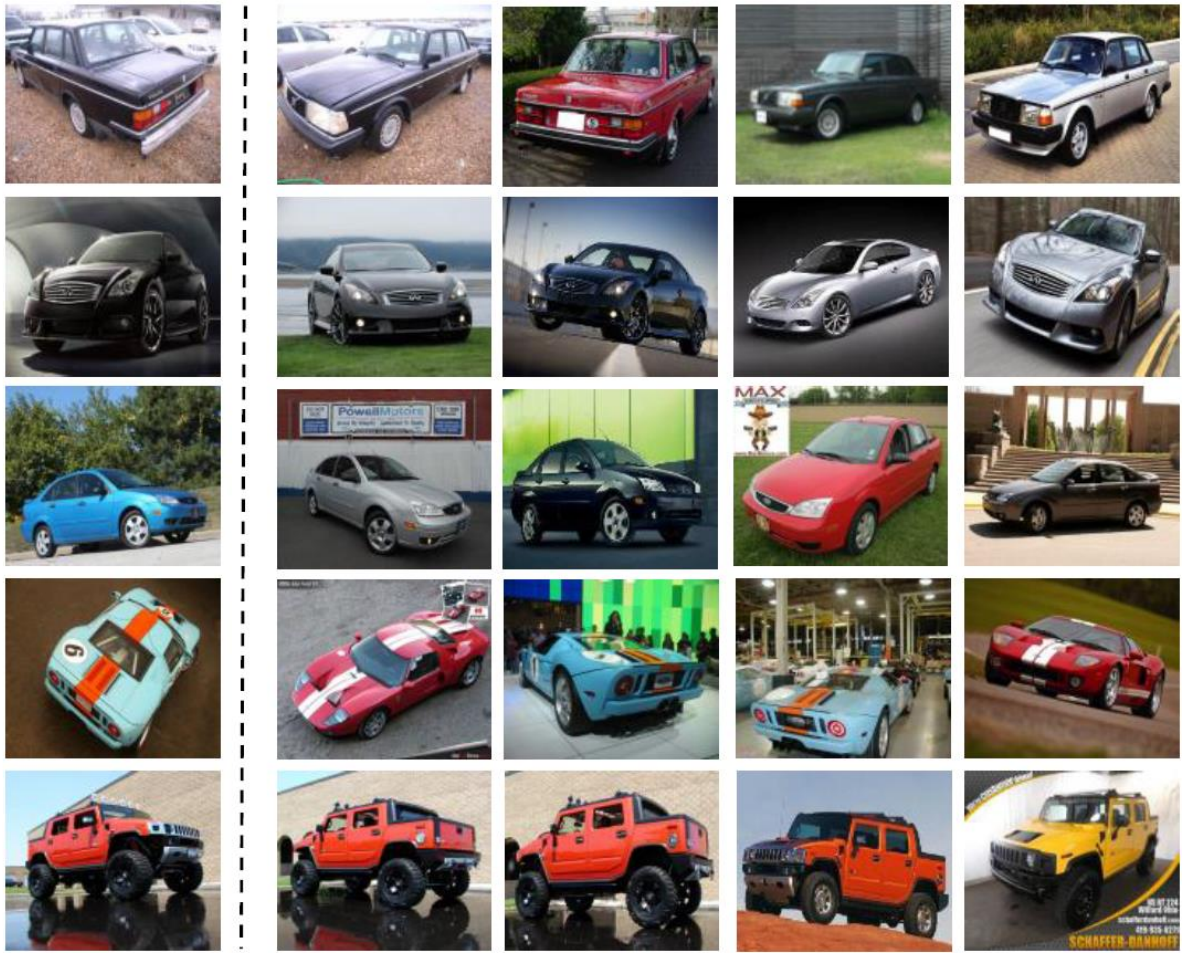
Our method achieves state-of-the-art performance in almost all settings on the all 4 benchmarks.

Experiments

- Qualitative results: Top 4 retrievals



CUB-200-2011

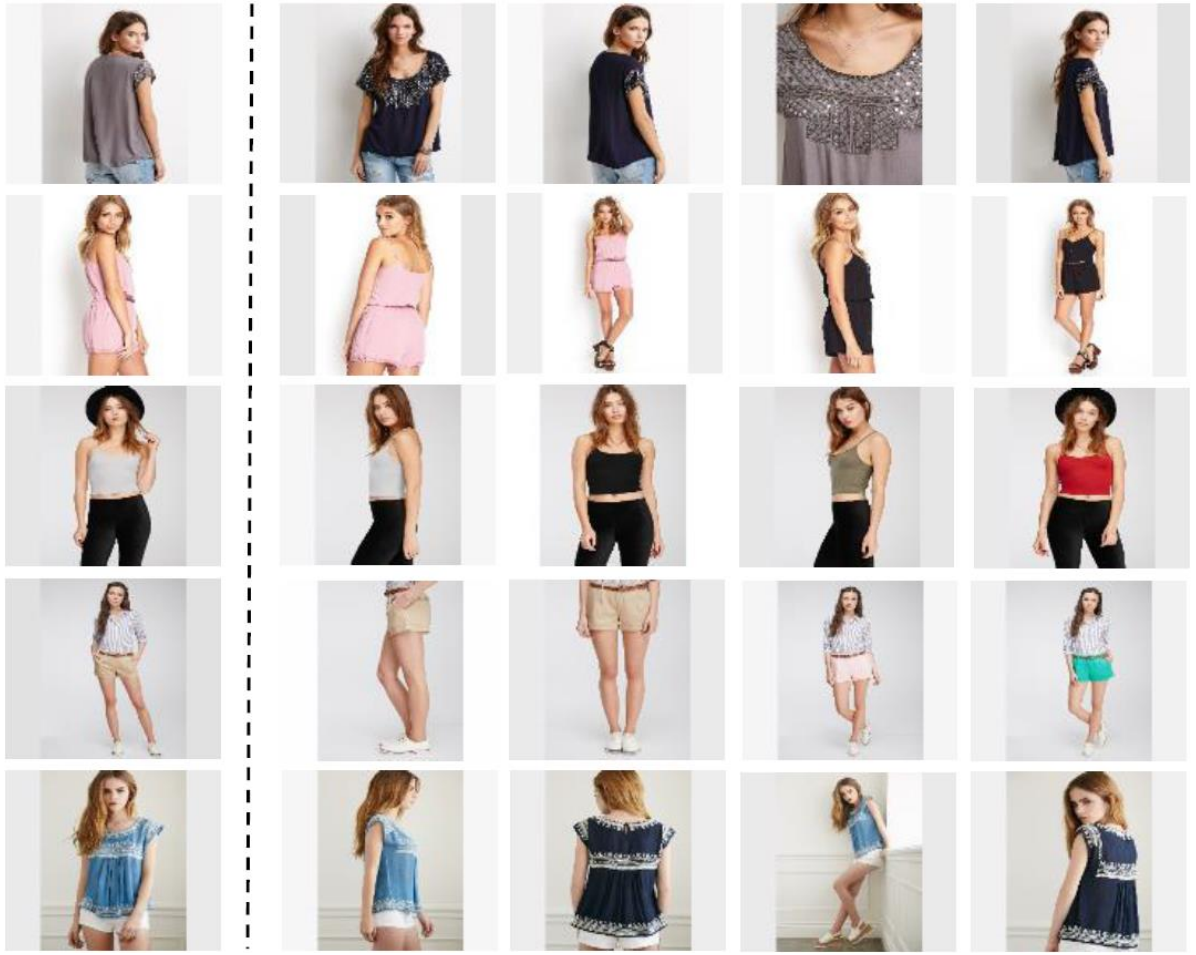


Cars-196

Experiments

- Qualitative results: Top 4 retrievals

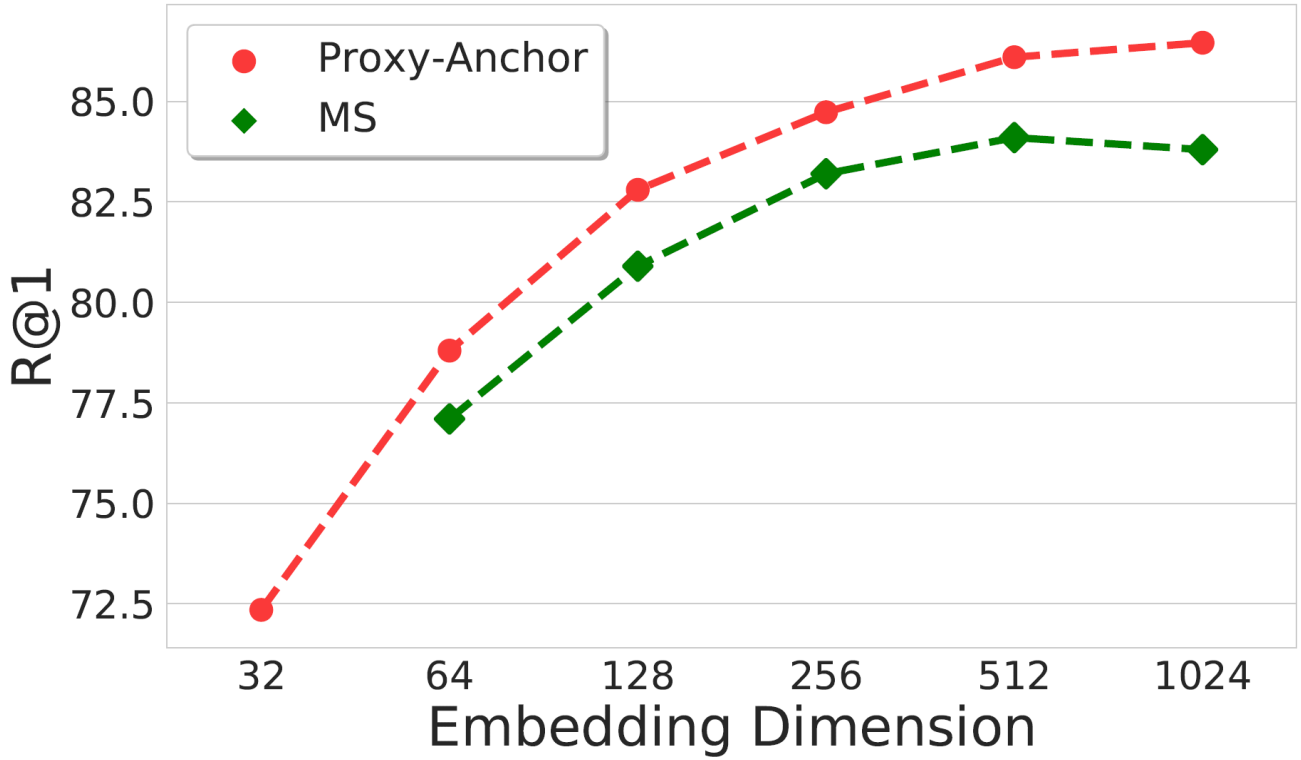
SOP



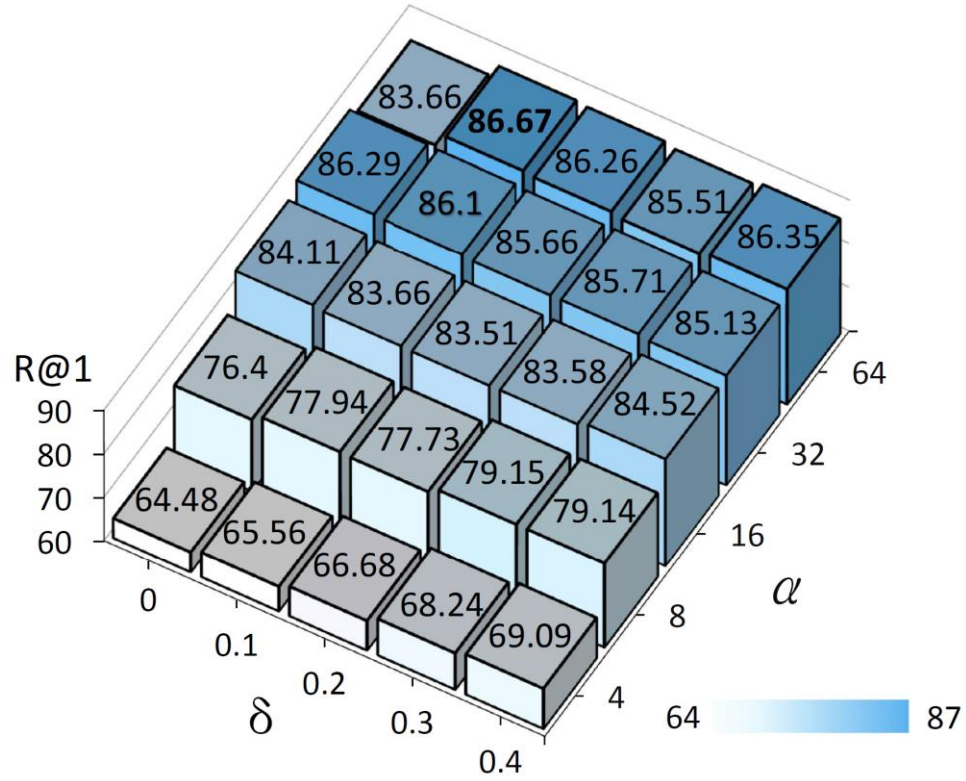
In-Shop

Experiments

- Impact of hyper-parameters



Accuracy vs. embedding dimension



Accuracy vs. α and δ

The performance is stable and high enough
when the embedding dimension ≥ 128 and $\alpha \geq 16$.

Experiments

- Ablation studies

Network	Image Size	CUB-200-2011				Cars-196			
		R@1	R@2	R@4	R@8	R@1	R@2	R@4	R@8
GoogleNet	224 × 224	63.8	74.4	83.6	90.4	84.3	90.4	94.1	96.7
Inception-BN		68.4	79.2	86.8	91.6	86.1	91.7	95.0	97.3
ResNet-50		69.7	80.0	87.0	92.4	87.7	92.9	95.8	97.9
ResNet-101		**70.8**	**81.0**	**88.1**	**93.0**	**87.9**	**93.0**	**96.1**	**97.9**
Inception-BN	256 × 256	71.1	80.4	87.4	92.5	88.3	93.1	95.7	97.5
	324 × 324	74.0	82.9	88.9	93.2	91.1	94.9	96.9	98.3
	448 × 448	**77.3**	**85.6**	**91.1**	**94.2**	**92.9**	**96.1**	**97.7**	**98.7**

Strong backbone and large input improve performance.

Conclusion

- Contributions
  - A new metric learning loss based on proxy
  - State-of-the-art performance
  - Fastest convergence

- Future directions
  - Analysis on generalizability
  - Improving test time efficiency

References

[1] FaceNet: A unified embedding for face recognition and clustering, CVPR 2015

[2] Beyond triplet loss: A deep quadruplet network for person re-identification, CVPR 2017

[3] Learning to compare image patches via convolutional neural networks, CVPR 2015

[4] Learning a similarity metric discriminatively with application to face verification, CVPR 2005

[5] Improved deep metric learning with multi-class N-pair loss objective, NeurIPS 2016

[6] No fuss distance metric learning using proxies, ICCV 2017

[7] Deep metric learning via lifted structured feature embedding, CVPR 2016

[8] Softtriple loss: Deep metric learning without triplet sampling, ICCV 2019

[9] High-order regularizer for deep embeddings, ICCV 2019