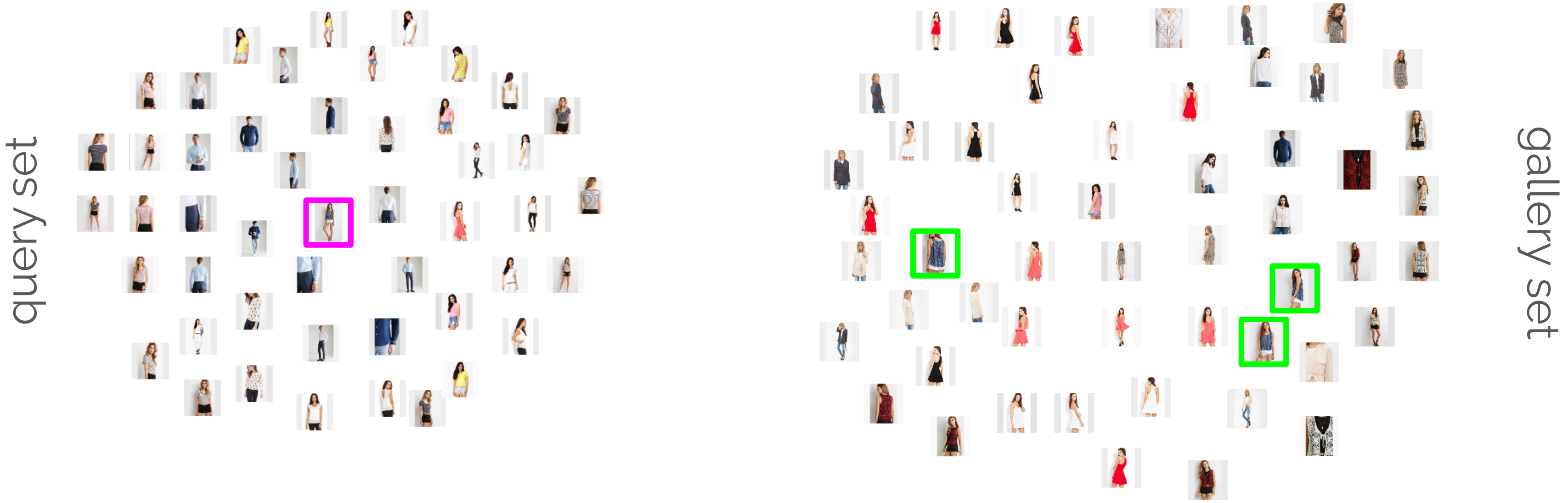
Best Practice DML

Evaluation

Evaluation Protocols



[35] Liu, Z. et al. "Deepfashion: Powering robust clothes recognition and retrieval with rich annotations" (CVPR 2016)

Evaluation

Retrieval performance: Recall@k (R@k)[30]

- at least one sample of same class among top k neighbors: R@k = 1

- Different k for different datasets

[35] Liu, Z. et al. “Deepfashion: Powering robust clothes recognition and retrieval with rich annotations.” (CVPR 2016)
[30] Jegou, H et al. “Product quantization for nearest neighbor search.” (tPAMI 2011)

Evaluation Protocols

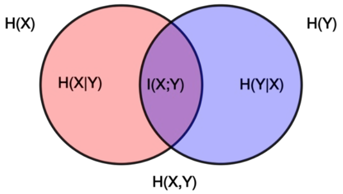
[35] Liu, Z. et al. "Deepfashion: Powering robust clothes recognition and retrieval with rich annotations" (CVPR 2016)

Evaluation

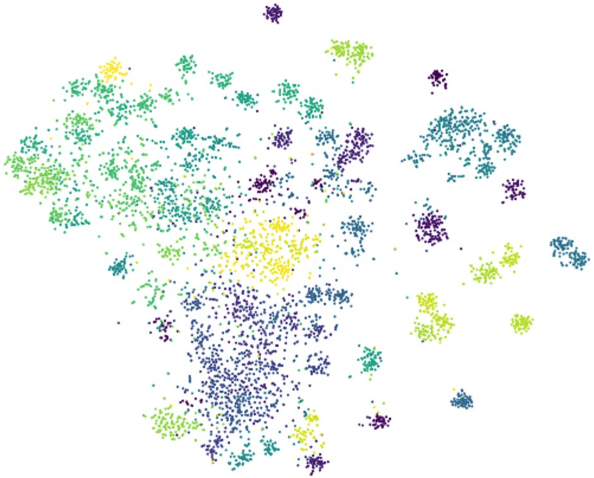
Clustering performance: NMI[31] (only for same query and gallery set)

- k-means clustering on embedding vectors
- Normalized Mutual Information between ground truth Y and clustering $\tilde{Y}$

$$NMI(Y,\tilde{Y}) = \frac{2I(Y,\tilde{Y})}{H(Y)H(\tilde{Y})}$$



[5]



[35] Liu, Z. et al. “Deepfashion: Powering robust clothes recognition and retrieval with rich annotations” (CVPR 2016)
[31]]McDaid, A. et al. “Normalized mutual information to evaluate overlapping community finding algorithms.” (arxiv 2011)

Datasets

Most Common Datasets

CUB-200-2011[32]

11,788 images
200 classes (avg 58 /class)

Cars196[33]

16,185 images
196 classes (avg 82 /class)

Stanford Online Products[34]

120,053 images
120,053 classes (avg 5 /class)

Inshop[35]

52,712 images
7,982 classes (avg 5 /class)

- first half for training, last half for testing
- evaluation set = test set

[32] Wah, C. et al."The Caltech-UCSD Birds-200-2011 Dataset." (Technical Report 2011)
[33] Krause, J. et al. "3d object representations for fine-grained categorization." (Workshop on 3D Representation and Recognition, 2013.)
[34] Song, H. et al. "Deep metric learning via lifted structured feature embedding." (CVPR 2016)
[35] Liu, Z. et al. "Deepfashion: Powering robust clothes recognition and retrieval with rich annotations" (CVPR 2016)

Current SOTA Performance

[10]

		CUB-200-2011					CARS196					Stanford Online Products			
Method	BB	**R@1**	**R@2**	**R@4**	**R@8**	**NMI**	**R@1**	**R@2**	**R@4**	**R@8**	**NMI**	**R@1**	**R@10**	**R@100**	**NMI**
Triplet^{64} (Schroff et al., 2015) *CVPR15*	G	42.5	55	66.4	77.2	55.3	51.5	63.8	73.5	82.4	53.4	66.7	82.4	91.9	89.5
Npairs^{64} (Sohn, 2016) *NeurIPS16*	G	51.9	64.3	74.9	83.2	60.2	68.9	78.9	85.8	90.9	62.7	66.4	82.9	92.1	87.9
Deep Spectral^{512} (Law et al., 2017) *ICML17*	BNI	53.2	66.1	76.7	85.2	59.2	73.1	82.2	89.0	93.0	64.3	67.6	83.7	93.3	89.4
Angular Loss^{512} (Wang et al., 2017) *ICCV17*	G	54.7	66.3	76	83.9	61.1	71.4	81.4	87.5	92.1	63.2	70.9	85.0	93.5	88.6
Proxy-NCA^{64} (Movshovitz-Attias et al., 2017) *ICCV17*	BNI	49.2	61.9	67.9	72.4	59.5	73.2	82.4	86.4	88.7	64.9	73.7	-	-	90.6
Margin Loss^{128} (Manmatha et al., 2017) *ICCV17*	R50	63.6	74.4	83.1	90.0	69.0	79.6	86.5	91.9	95.1	69.1	72.7	86.2	93.8	90.7
Hierarchical triplet^{512} (Ge et al., 2018) *ECCV18*	BNI	57.1	68.8	78.7	86.5	-	81.4	88.0	92.7	95.7	-	74.8	88.3	94.8	-
ABE^{512} (Kim et al., 2018) *ECCV18*	G	60.6	71.5	79.8	87.4	-	85.2	90.5	94.0	96.1	-	76.3	88.4	94.8	-
Normalized Softmax^{512} (Zhai & Wu, 2019) *BMVC19*	R50	61.3	73.9	83.5	90.0	69.7	84.2	90.4	94.4	96.9	**74.0**	78.2	90.6	96.2	91.0
RLL-H^{512} (Wang et al., 2019b) *CVPR19*	BNI	57.4	69.7	79.2	86.9	63.6	74	83.6	90.1	94.1	65.4	76.1	89.1	95.4	89.7
Multi-similarity^{512} (Wang et al., 2019a) *CVPR19*	BNI	65.7	77.0	86.3	91.2	-	84.1	90.4	94.0	96.5	-	78.2	90.5	96.0	-
Relational Knowledge^{512} (Park et al., 2019a) *CVPR19*	G	61.4	73.0	81.9	89.0	-	82.3	89.8	94.2	96.6	-	75.1	88.3	95.2	-
Divide and Conquer^{1028} (Sanakoyeu et al., 2019) *CVPR19*	R50	65.9	76.6	84.4	90.6	69.6	84.6	90.7	94.1	96.5	70.3	75.9	88.4	94.9	90.2
SoftTriple Loss^{512} (Qian et al., 2019) *ICCV19*	BNI	65.4	76.4	84.5	90.4	69.3	84.5	90.7	94.5	96.9	70.1	78.3	90.3	95.9	**92.0**
HORDE^{512} (Jacob et al., 2019) *ICCV19*	BNI	66.3	76.7	84.7	90.6	-	83.9	90.3	94.1	96.3	-	**80.1**	**91.3**	96.2	-
MIC^{128} (Brattoli et al., 2019) *ICCV19*	R50	66.1	76.8	85.6	-	69.7	82.6	89.1	93.2	-	68.4	77.2	89.4	95.6	90.0
Easy triplet mining^{512} (Xuan et al., 2020b) *WACV20*	R50	64.9	75.3	83.5	-	-	82.7	89.3	93.0	-	-	78.3	90.7	**96.3**	-
Group Loss^{1024} (Elezi et al., 2020) *ECCV20*	BNI	65.5	77.0	85.0	91.3	69.0	85.6	91.2	94.9	97.0	**72.7**	75.1	87.5	94.2	**90.8**
Proxy NCA++^{512} (Teh et al., 2020) *ECCV20*	R50	66.3	77.8	**87.7**	91.3	**71.3**	84.9	90.6	94.9	97.2	71.5	79.8	**91.4**	**96.4**	-
DiVA^{512} (Milbich et al., 2020) *ECCV20*	R50	**69.2**	**79.3**	-	-	**71.4**	**87.6**	**92.9**	-	-	72.2	79.6	-	-	90.6
PADS^{128} (Roth et al., 2020) *CVPR20*	R50	67.3	78.0	85.9	-	69.9	83.5	89.7	93.8	-	68.8	76.5	89.0	95.4	89.9
Proxy Anchor^{512} (Kim et al., 2020) *CVPR20*	BNI	68.4	79.2	86.8	**91.6**	-	86.1	91.7	95.0	**97.3**	-	79.1	90.8	96.2	-
Proxy Anchor^{512} (Kim et al., 2020) *CVPR20*	R50	**69.7**	**80.0**	**87.0**	**92.4**	-	**87.7**	**92.9**	**95.8**	**97.9**	-	**80.0**	**91.7**	**96.6**	-
Proxy Few^{512} (Zhu et al., 2020) *NeurIPS20*	BNI	66.6	77.6	86.4	-	69.8	85.5	**91.8**	**95.3**	-	72.4	78.0	90.6	96.2	90.2
Intra-Batch^{512}	R50	**70.3**	**80.3**	**87.6**	**92.7**	**74.0**	**88.1**	**93.3**	**96.2**	**98.2**	**74.8**	**81.4**	**91.3**	95.9	**92.6**

[10] Seidenschwarz, J. et al. "Learning Intra-Batch Connections for Deep Metric Learning." ICML (2021).

Current SOTA Performance

Method	BB	CUB-200-2011 R@1	R@2	R@4	R@8	NMI	CARS196 R@1	R@2	R@4	R@8	NMI	Stanford Online Products R@1	R@10	R@100	NMI
Triplet^{64} (Schroff et al., 2015) *CVPR15*	G	42.5	55	66.4	77.2	55.3	51.5	63.8	73.5	82.4	53.4	66.7	82.4	91.9	89.5
Npairs^{64} (Sohn, 2016) *NeurIPS16*	G	51.9	64.3	74.9	83.2	60.2	68.9	78.9	85.8	90.9	62.7	66.4	82.9	92.1	87.9
Deep Spectral^{512} (Law et al., 2017) *ICML17*	BNI	53.2	66.1	76.7	85.2	59.2	73.1	82.2	89.0	93.0	64.3	67.6	83.7	93.3	89.4
Angular Loss^{512} (Wang et al., 2017) *ICCV17*	G	54.7	66.3	76	83.9	61.1	71.4	81.4	87.5	92.1	63.2	70.9	85.0	93.5	88.6
Proxy-NCA^{64} (Movshovitz-Attias et al., 2017) *ICCV17*	BNI	49.2	61.9	67.9	72.4	59.5	73.2	82.4	86.4	88.7	64.9	73.7	-	-	90.6
Margin Loss^{128} (Manmatha et al., 2017) *ICCV17*	R50	63.6	74.4	83.1	90.0	69.0	79.6	86.5	91.9	95.1	69.1	72.7	86.2	93.8	90.7
Hierarchical triplet^{512} (Ge et al., 2018) *ECCV18*	BNI	57.1	68.8	78.7	86.5	-	81.4	88.0	92.7	95.7	-	74.8	88.3	94.8	-
ABE^{512} (Kim et al., 2018) *ECCV18*	G	60.6	71.5	79.8	87.4	-	85.2	90.5	94.0	96.1	-	76.3	88.4	94.8	-
Normalized Softmax^{512} (Zhai & Wu, 2019) *BMVC19*	R50	61.3	73.9	83.5	90.0	69.7	84.2	90.4	94.4	96.9	**74.0**	78.2	90.6	96.2	91.0
RLL-H^{512} (Wang et al., 2019b) *CVPR19*	BNI	57.4	69.7	79.2	86.9	63.6	74	83.6	90.1	94.1	65.4	76.1	89.1	95.4	89.7
Multi-similarity^{512} (Wang et al., 2019a) *CVPR19*	BNI	65.7	77.0	86.3	91.2	-	84.1	90.4	94.0	96.5	-	78.2	90.5	96.0	-
Relational Knowledge^{512} (Park et al., 2019a) *CVPR19*	G	61.4	73.0	81.9	89.0	-	82.3	89.8	94.2	96.6	-	75.1	88.3	95.2	-
Divide and Conquer^{1028} (Sanakoyeu et al., 2019) *CVPR19*	R50	65.9	76.6	84.4	90.6	69.6	84.6	90.7	94.1	96.5	70.3	75.9	88.4	94.9	90.2
SoftTriple Loss^{512} (Qian et al., 2019) *ICCV19*	BNI	65.4	76.4	84.5	90.4	69.3	84.5	90.7	94.5	96.9	70.1	78.3	90.3	95.9	**92.0**
HORDE^{512} (Jacob et al., 2019) *ICCV19*	BNI	66.3	76.7	84.7	90.6	-	83.9	90.3	94.1	96.3	-	**80.1**	**91.3**	96.2	-
MIC^{128} (Brattoli et al., 2019) *ICCV19*	R50	66.1	76.8	85.6	-	69.7	82.6	89.1	93.2	-	68.4	77.2	89.4	95.6	90.0
Easy triplet mining^{512} (Xuan et al., 2020b) *WACV20*	R50	64.9	75.3	83.5	-	-	82.7	89.3	93.0	-	-	78.3	90.7	**96.3**	-
Group Loss^{1024} (Elezi et al., 2020) *ECCV20*	BNI	65.5	77.0	85.0	91.3	69.0	85.6	91.2	94.9	97.0	**72.7**	75.1	87.5	94.2	**90.8**
Proxy NCA++^{512} (Teh et al., 2020) *ECCV20*	R50	66.3	77.8	**87.7**	91.3	**71.3**	84.9	90.6	94.9	97.2	71.5	79.8	**91.4**	**96.4**	-
DiVA^{512} (Milbich et al., 2020) *ECCV20*	R50	**69.2**	**79.3**	-	-	**71.4**	**87.6**	**92.9**	-	-	72.2	79.6	-	-	90.6
PADS^{128} (Roth et al., 2020) *CVPR20*	R50	67.3	78.0	85.9	-	69.9	83.5	89.7	93.8	-	68.8	76.5	89.0	95.4	89.9
Proxy Anchor^{512} (Kim et al., 2020) *CVPR20*	BNI	68.4	79.2	86.8	**91.6**	-	86.1	91.7	95.0	**97.3**	-	79.1	90.8	96.2	-
Proxy Anchor^{512} (Kim et al., 2020) *CVPR20*	R50	**69.7**	**80.0**	**87.0**	**92.4**	-	**87.7**	**92.9**	**95.8**	**97.9**	-	**80.0**	**91.7**	**96.6**	-
Proxy Few^{512} (Zhu et al., 2020) *NeurIPS20*	BNI	66.6	77.6	86.4	-	69.8	85.5	**91.8**	**95.3**	-	72.4	78.0	90.6	96.2	90.2
Intra-Batch^{512}	R50	**70.3**	**80.3**	**87.6**	**92.7**	**74.0**	**88.1**	**93.3**	**96.2**	**98.2**	**74.8**	**81.4**	**91.3**	95.9	**92.6**

[10]

Method	BB	R@1	R@10	R@20	R@40
FashionNet^{4096} (Liu et al., 2016) *CVPR16*	V	53.0	73.0	76.0	79.0
A-BIER^{512} (Opitz et al., 2020) *PAMI20*	G	83.1	95.1	96.9	97.8
ABE^{512} (Kim et al., 2018) *ECCV18*	G	87.3	96.7	97.9	98.5
Multi-similarity^{512} (Wang et al., 2019a) *CVPR19*	BNI	89.7	**97.9**	98.5	**99.1**
Learning to Rank^{512} (Çakir et al., 2019)	R50	90.9	97.7	98.5	**98.9**
HORDE^{512} (Jacob et al., 2019) *ICCV19*	BNI	90.4	97.8	98.4	**98.9**
MIC^{128} (Brattoli et al., 2019) *ICCV19*	R50	88.2	97.0	98.0	98.8
Proxy NCA++^{512} (Teh et al., 2020) *ECCV20*	R50	90.4	**98.1**	**98.8**	**99.2**
Proxy Anchor^{512} (Kim et al., 2020) *CVPR20*	BNI	**91.5**	**98.1**	**98.8**	**99.1**
Proxy Anchor^{512} (Kim et al., 2020) *CVPR20*	R50	**92.1**	**98.1**	**98.7**	**99.2**
Intra-Batch^{512}	R50	**92.8**	**98.5**	**99.1**	**99.2**

[10]

➡ Fewer works on Inshop dataset as other evaluation protocol

[10] Seidenschwarz, J. et al. “Learning Intra-Batch Connections for Deep Metric Learning.” ICML (2021).

New Transformer-Based Works

	CUB-200-2011					Cars196					SOP				InShop			
	R@1	R@2	R@4	R@8	NMI	R@1	R@2	R@4	R@8	NMI	R@1	R@10	R@100	NMI	R@1	R@10	R@20	R@40
IntraBtach (R50)	70.3	80.3	87.6	92.7	74.0	88.1	93.3	96.2	98.2	74.8	81.4	91.3	95.9	92.6	92.8	98.5	99.1	99.2

[38]

Method	Dim	CUB-200-2011 (K)				Cars-196 (K)				SOP (K)				In-Shop (K)			
		1	2	4	8	1	2	4	8	1	10	100	1000	1	10	20	30
ResNet-50 [18] †	2048	41.2	53.8	66.3	77.5	41.4	53.6	66.1	76.6	50.6	66.7	80.7	93.0	25.8	49.1	56.4	60.5
DeiT-S [53] †	384	70.6	81.3	88.7	93.5	52.8	65.1	76.2	85.3	58.3	73.9	85.9	95.4	37.9	64.7	72.1	75.9
DINO [3] †	384	70.8	81.1	88.8	93.5	42.9	53.9	64.2	74.4	63.4	78.1	88.3	96.0	46.1	71.1	77.5	81.1
ViT-S [48] † §	384	83.1	90.4	94.4	96.5	47.8	60.2	72.2	82.6	62.1	77.7	89.0	96.8	43.2	70.2	76.7	80.5
Sph-DeiT	384	76.2	84.5	90.2	94.3	81.7	88.6	93.4	96.2	82.5	92.9	97.2	99.1	89.6	97.2	98.0	98.4
Sph-DINO	384	78.7	86.7	91.4	94.9	86.6	91.8	95.2	97.4	82.2	92.1	96.8	98.9	90.1	97.1	98.0	98.4
Sph-ViT §	384	85.1	90.7	94.3	96.4	81.7	89.0	93.0	95.8	82.1	92.5	97.1	99.1	90.4	97.4	98.2	98.6
Hyp-DeiT	384	77.8	86.6	91.9	95.1	86.4	92.2	95.5	97.5	83.3	93.5	97.4	99.1	90.5	97.8	98.5	98.9
Hyp-DINO	384	80.9	87.6	92.4	95.6	**89.2**	**94.1**	**96.7**	**98.1**	85.1	94.4	97.8	99.3	92.4	**98.4**	**98.9**	**99.1**
Hyp-ViT §	384	**85.6**	**91.4**	**94.8**	**96.7**	86.5	92.1	95.3	97.3	**85.9**	**94.9**	**98.1**	**99.5**	**92.5**	98.3	98.8	**99.1**

† pretrained encoders without training on the target dataset. § pretrained on the larger ImageNet-21k [6].

[39]

	Method	dim	SOP			CUB			
			1	10	100	1	2	4	8
DeiT	IRT_R [7]	384	84.2	93.7	97.3	76.6	85.0	91.1	94.3
DeiT	ROADMAP (ours)	384	**86.0**	**94.4**	**97.6**	**77.4**	**85.5**	**91.4**	**95.0**

[40]

Method	Arch.$^{\text{dim}}$	SOP [39]				Cars196 [27]			
		r@k							
		10^0	10^1	10^2	10^3	1	2	4	8
RS@k†	R_{50}^{512}	**82.8**	**92.9**	**97.0**	99.0	80.7	88.3	92.8	95.7
RS@k† +SiMix	R_{50}^{512}	82.1	92.8	**97.0**	**99.1**	**88.2**	**93.0**	**95.9**	97.4
		+11%	+5.3%	+12%	+10%	+13%	+6.7%	+4.7%	−13%
SAP† [6]	ViT-B/32^{512}	83.7	94.0	97.8	99.3	78.1	85.7	91.0	94.8
RS@k†	ViT-B/32^{512}	85.1	94.6	98.0	99.3	78.1	86.4	92.3	95.6
SAP† [6]	ViT-B/16^{512}	86.6	95.4	98.4	99.5	86.2	92.1	95.1	97.2
RS@k†	ViT-B/16^{512}	88.0	96.1	98.6	99.6	89.5	94.2	96.6	98.3

[38] Ermolov, A.et al. “Hyperbolic Vision Transformers: Combining Improvements in Metric Learning” (CVPR 2022)
[39] Ramzi, E. et al. “Robust and Decomposable Average Precision for Image Retrieval” (NeurIPS 2021)

Standard Protocol

Standard Protocol - Data Augmentation

Training[3]: Crop (scale, aspect ratio, 227) and Random horizontal flip

Testing[3]: Resize (smaller side *256*) CenterCrop (to *227*)

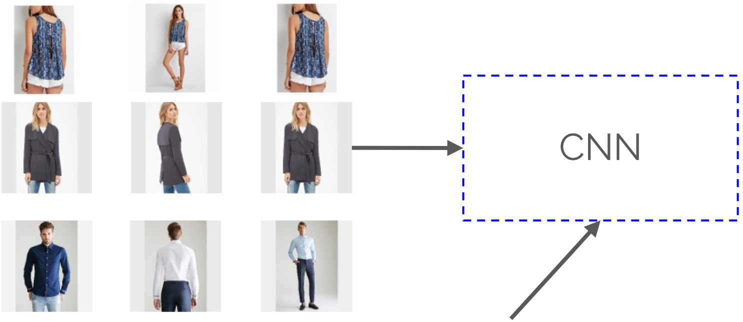
[3] Kim, S. et al. "Proxy Anchor Loss for Deep Metric Learning." CVPR (2020)

Standard Protocol - Training Pipeline

Class-Balanced
Sampling[36], ...

[36] Zhai, A. and Wu, H. Classification is a strong baseline for deep metric learning. (BMVC 2019).

Standard Protocol - Training Pipeline



BN-Inception / ResNet

Class-Balanced Sampling[36], ...

Network	GN	IBN	R50
CUB200, R@1	45.41	48.78	43.77
CARS196, R@1	35.31	43.36	36.39
SOP, R@1	44.28	49.05	48.65

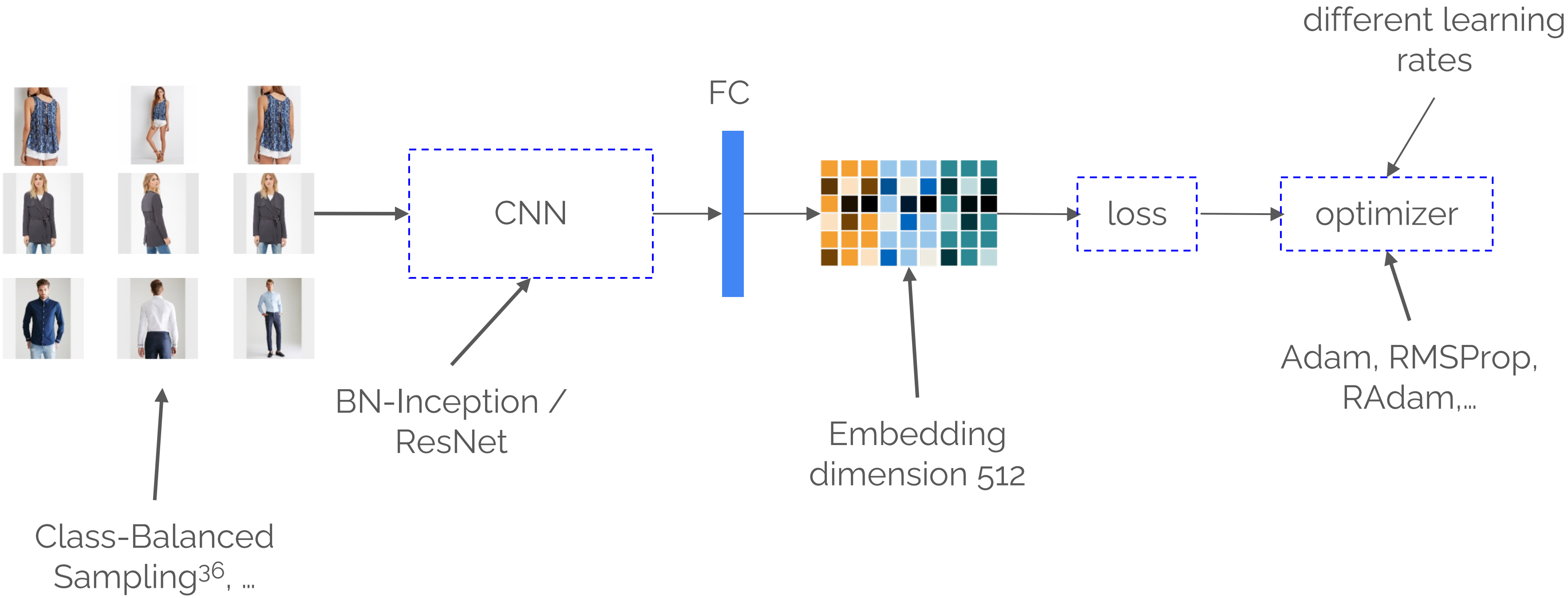
[12]

[10]

		CUB-200-2		
Method	**BB**	**R@1**	**R@2**	**R@4**
Triplet^{64} (Schroff et al., 2015) *CVPR15*	G	42.5	55	66.4
Npairs^{64} (Sohn, 2016) *NeurIPS16*	G	51.9	64.3	74.9
Deep Spectral^{512} (Law et al., 2017) *ICML17*	BNI	53.2	66.1	76.7
Angular Loss^{512} (Wang et al., 2017) *ICCV17*	G	54.7	66.3	76
Proxy-NCA^{64} (Movshovitz-Attias et al., 2017) *ICCV17*	BNI	49.2	61.9	67.9
Margin Loss^{128} (Manmatha et al., 2017) *ICCV17*	R50	63.6	74.4	83.1
Hierarchical triplet^{512} (Ge et al., 2018) *ECCV18*	BNI	57.1	68.8	78.7
ABE^{512} (Kim et al., 2018) *ECCV18*	G	60.6	71.5	79.8
Normalized Softmax^{512} (Zhai & Wu, 2019) *BMVC19*	R50	61.3	73.9	83.5
RLL-H^{512} (Wang et al., 2019b) *CVPR19*	BNI	57.4	69.7	79.2
Multi-similarity^{512} (Wang et al., 2019a) *CVPR19*	BNI	65.7	77.0	86.3
Relational Knowledge^{512} (Park et al., 2019a) *CVPR19*	G	61.4	73.0	81.9
Divide and Conquer^{1028} (Sanakoyeu et al., 2019) *CVPR19*	R50	65.9	76.6	84.4
SoftTriple Loss^{512} (Qian et al., 2019) *ICCV19*	BNI	65.4	76.4	84.5
HORDE^{512} (Jacob et al., 2019) *ICCV19*	BNI	66.3	76.7	84.7
MIC^{128} (Brattoli et al., 2019) *ICCV19*	R50	66.1	76.8	85.6
Easy triplet mining^{512} (Xuan et al., 2020b) *WACV20*	R50	64.9	75.3	83.5

[10] Seidenschwarz, J. et al "Learning Intra-Batch Connections for Deep Metric Learning." ICML (2021).
[12] Roth, K. et al. "Revisiting Training Strategies and Generalization Performance in Deep Metric Learning." ICML (2020)

Standard Protocol - Training Pipeline



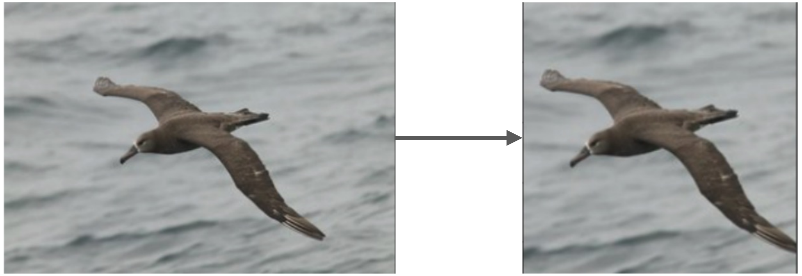
➡ Ensure fair comparison especially backbone and embedding dimension

Tricks to improve performance

Tricks to improve performance

Larger corp[6]

227 x 227 → 256 x 256



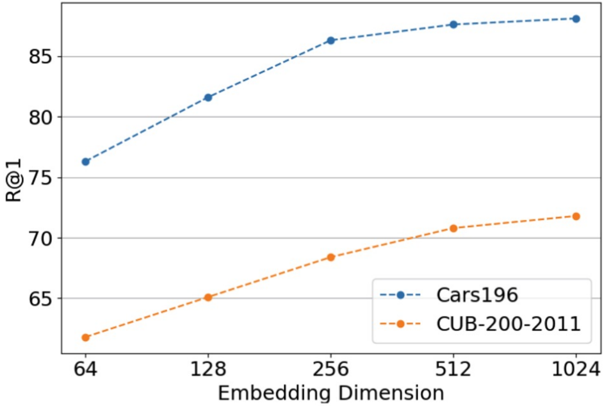
[6] Teh, E. W. et al. "ProxyNCA++: Revisiting and Revitalizing Proxy Neighborhood Component Analysis." ECCV (2020)

Tricks to improve performance

Larger corp[6]

227 x 227 → 256 x 256

Larger embedding dimension



[10] Seidenschwarz, J. et al "Learning Intra-Batch Connections for Deep Metric Learning." ICML (2021).

Tricks to improve performance

Larger corp[6]

227 x 227 → 256 x 256

Larger embedding dimension

[10]

		CUB-200-2011				
Method	BB	**R@1**	**R@2**	**R@4**	**R@8**	**NMI**
Triplet[64] (Schroff et al., 2015) *CVPR15*	G	42.5	55	66.4	77.2	55.3
Npairs[64] (Sohn, 2016) *NeurIPS16*	G	51.9	64.3	74.9	83.2	60.2
Deep Spectral[512] (Law et al., 2017) *ICML17*	BNI	53.2	66.1	76.7	85.2	59.2
Angular Loss[512] (Wang et al., 2017) *ICCV17*	G	54.7	66.3	76	83.9	61.1
Proxy-NCA[64] (Movshovitz-Attias et al., 2017) *ICCV17*	BNI	49.2	61.9	67.9	72.4	59.5
Margin Loss[128] (Manmatha et al., 2017) *ICCV17*	R50	63.6	74.4	83.1	90.0	69.0
Hierarchical triplet[512] (Ge et al., 2018) *ECCV18*	BNI	57.1	68.8	78.7	86.5	-
ABE[512] (Kim et al., 2018) *ECCV18*	G	60.6	71.5	79.8	87.4	-
Normalized Softmax[512] (Zhai & Wu, 2019) *BMVC19*	R50	61.3	73.9	83.5	90.0	69.7
RLL-H[512] (Wang et al., 2019b) *CVPR19*	BNI	57.4	69.7	79.2	86.9	63.6
Multi-similarity[512] (Wang et al., 2019a) *CVPR19*	BNI	65.7	77.0	86.3	91.2	-
Relational Knowledge[512] (Park et al., 2019a) *CVPR19*	G	61.4	73.0	81.9	89.0	-
Divide and Conquer[1028] (Sanakoyeu et al., 2019) *CVPR19*	R50	65.9	76.6	84.4	90.6	69.6
SoftTriple Loss[512] (Qian et al., 2019) *ICCV19*	BNI	65.4	76.4	84.5	90.4	69.3
HORDE[512] (Jacob et al., 2019) *ICCV19*	BNI	66.3	76.7	84.7	90.6	-
MIC[128] (Brattoli et al., 2019) *ICCV19*	R50	66.1	76.8	85.6	-	69.7

[10] Seidenschwarz, J. et al "Learning Intra-Batch Connections for Deep Metric Learning." ICML (2021).

Tricks to improve performance

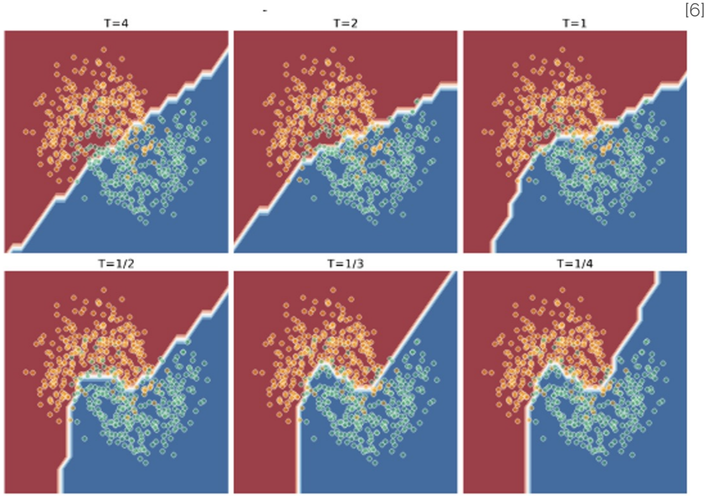
Larger corp[6]

227 x 227 → 256 x 256

Larger embedding dimension

512 → 1024

Temperature scaling



[6] Teh, E. W. et al. “ProxyNCA++: Revisiting and Revitalizing Proxy Neighborhood Component Analysis.” ECCV (2020)

Tricks to improve performance

Larger corp[6]

227 x 227 → 256 x 256

Larger embedding dimension

512 → 1024

Temperature scaling

[6]

R@1	without scale	with scale
ProxyNCA (Emb: 2048)	59.3 ± 0.4	62.9 ± 0.4
+cbs	54.8 ± 6.2	64.0 ± 0.4
+prob	59.0 ± 0.4	63.4 ± 0.6
+norm	60.2 ± 0.6	65.3 ± 0.7
+max	61.3 ± 0.7	65.1 ± 0.3
+fast	56.3 ± 0.8	64.3 ± 0.8
+max +fast	60.3 ± 0.5	67.2 ± 0.5
+norm +prob +cbs	60.4 ± 0.7	69.1 ± 0.5
+norm +prob +cbs +max	61.2 ± 0.7	70.3 ± 0.9
+norm +prob +cbs +max +fast	61.4 ± 0.4	72.2 ± 0.8

[6] Teh, E. W. et al. “ProxyNCA++: Revisiting and Revitalizing Proxy Neighborhood Component Analysis.” ECCV (2020)

Tricks to improve performance

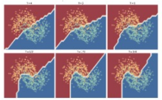
Larger corp[6]

227 x 227 → 256 x 256

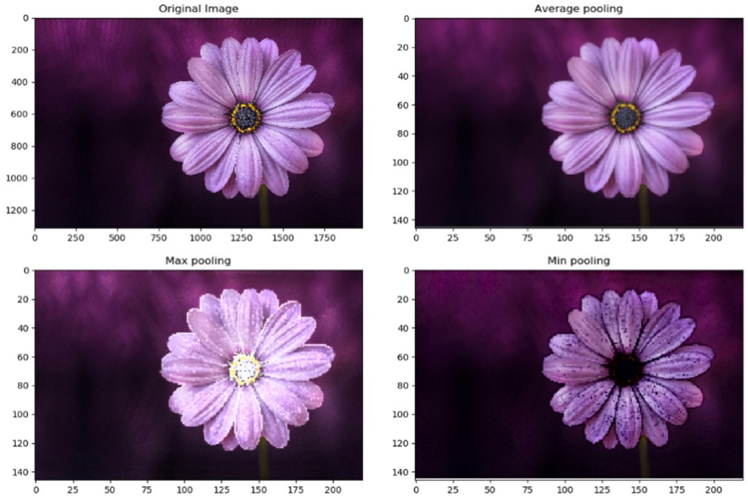
Larger embedding dimension

512 → 1024

Temperatur scaling



Average vs. Max Pooling



[8]

[8] https://medium.com/@bdhuma/which-pooling-method-is-better-maxpooling-vs-minpooling-vs-average-pooling-95fb03f45a9

Tricks to improve performance

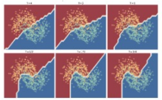
Larger corp[6]

227 x 227 → 256 x 256

Larger embedding dimension

512 → 1024

Temperatur scaling



Average vs. Max Pooling

[6]

Method	Pool	R@1	Arch	Emb
WithoutTraining	avg	45.0	R50	2048
	max	**53.1**	R50	2048
Margin [33]	avg	63.3	R50	128
	max	**64.3**	R50	128
Triplet-Semihard sampling [22]	avg	60.5	R50	128
	max	**61.6**	R50	128
MS [32]	avg	64.9	R50	512
	max	**68.5**	R50	512
MS [32]	avg	65.1	I3	512
	max	**66.1**	I3	512
Horde (Contrastive Loss) [13]	avg	**65.1**	I3	512
	max	63.1	I3	512

[6] Teh, E. W. et al. "ProxyNCA++: Revisiting and Revitalizing Proxy Neighborhood Component Analysis." ECCV (2020)

Tricks to improve performance

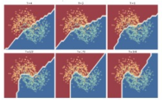
Larger corp[6]

227 x 227 →
256 x 256

Larger embedding dimension

512 → 1024

Temperatur scaling



Average vs.
Max Pooling

[9]

	CUB-200-2011		CARS196		SOP		In-Shop
Inference	**R@1**	**NMI**	**R@1**	**NMI**	**R@1**	**NMI**	**R@1**
GL	65.5	69.0	85.6	72.7	75.1	90.8	86.8
mixed	67.5	69.5	88.2	72.9	78.1	91.2	89.1

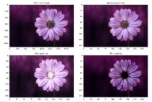
[9] Elezi, I. et al. "The Group Loss++: A deeper look into group loss for deep metric learning", PAMI (2022/03)

Tricks to improve performance

Larger corp[6]

227 x 227 → 256 x 256

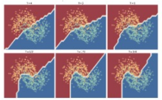
Average vs. Max Pooling



Larger embedding dimension

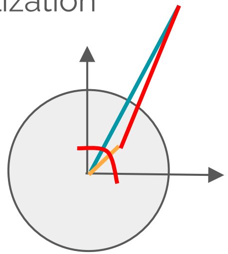
512 → 1024

Temperatur scaling



***β*-normalization**

$$\phi(\boldsymbol{I}_i)_\beta = \frac{\phi(\boldsymbol{I}_i)}{||\phi(\boldsymbol{I}_i)||_2} + \beta\phi(\boldsymbol{I}_i).$$



[9]

	CUB-200-2011		CARS196		SOP		In-Shop
Inference	**R@1**	**NMI**	**R@1**	**NMI**	**R@1**	**NMI**	**R@1**
GL	65.5	69.0	85.6	72.7	75.1	90.8	86.8
β	66.8	69.0	87.1	72.2	75.9	91.3	87.1

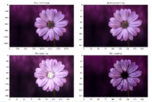
[9] Elezi, I. et al. "The Group Loss++: A deeper look into group loss for deep metric learning", PAMI (2022/03)

Tricks to improve performance

Larger corp[6]

227 x 227 →
256 x 256

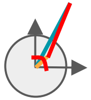
Average vs.
Max Pooling



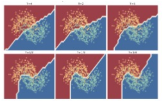
Larger
embedding
dimension

512 → 1024

***β*-**
normalization



Temperatur
scaling



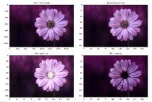
[9] Elezi, I. et al. "The Group Loss++: A deeper look into group loss for deep metric learning", PAMI (2022/03)

Tricks to improve performance

Larger corp[6]

227 x 227 → 256 x 256

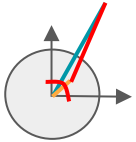
Average vs. Max Pooling



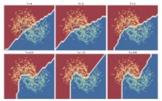
Larger embedding dimension

512 → 1024

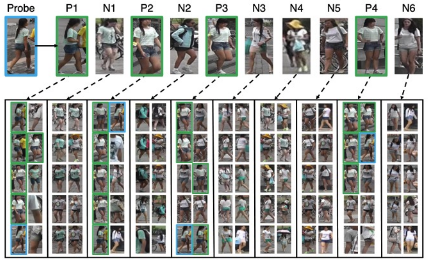
β-normalization



Temperatur scaling



Re-Ranking



[37]

[9]

	CUB-200-2011		CARS196		SOP		In-Shop
Inference	**R@1**	**NMI**	**R@1**	**NMI**	**R@1**	**NMI**	**R@1**
GL	65.5	69.0	85.6	72.7	75.1	90.8	86.8
R	68.3	69.1	87.4	71.8	75.7		87.8

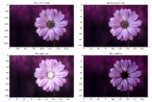
[9] Elezi, I. et al. "The Group Loss++: A deeper look into group loss for deep metric learning", PAMI (2022/03)
[37] Zhong, Z. et al. "Re-ranking Person Re-identification with k-reciprocal Encoding" (CVPR 2017)

Tricks to improve performance

Larger corp[6]

227 x 227 → 256 x 256

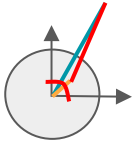
Average vs. Max Pooling



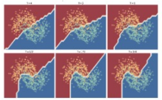
Larger embedding dimension

512 → 1024

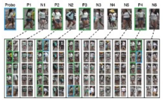
β-normalization



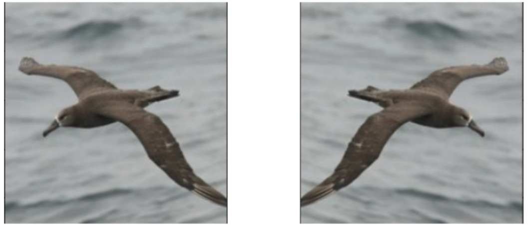
Temperatur scaling



Re-Ranking



Flip-Evaluation



	CUB-200-2011		CARS196		SOP		In-Shop
Inference	**R@1**	**NMI**	**R@1**	**NMI**	**R@1**	**NMI**	**R@1**
GL	65.5	69.0	85.6	72.7	75.1	90.8	86.8
F	68.0	70.9	88.1	73.2	76.1	91.3	87.4

[9]

[9] Elezi, I. et al. "The Group Loss++: A deeper look into group loss for deep metric learning", PAMI (2022/03)
[37] Zhong, Z. et al. "Re-ranking Person Re-identification with k-reciprocal Encoding" (CVPR 2017)

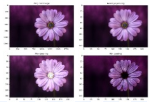
Tricks to improve performance

Technical University of Munich

Larger corp[6]

227 x 227 → 256 x 256

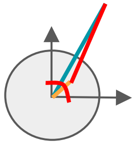
Average vs. Max Pooling



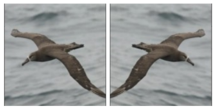
Larger embedding dimension

512 → 1024

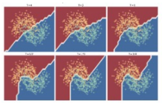
β-normalization



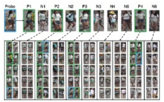
Flip-Evaluation



Temperatur scaling



Re-Ranking



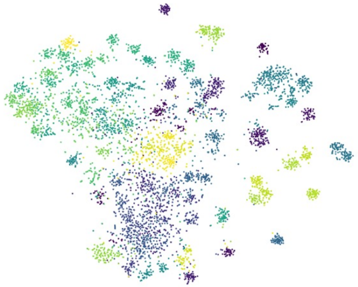
→ Ensure fair evaluation!

[9] Elezi, I. et al. "The Group Loss++: A deeper look into group loss for deep metric learning", PAMI (2022/03)
[37] Zhong, Z. et al. "Re-ranking Person Re-identification with k-reciprocal Encoding" (CVPR 2017)

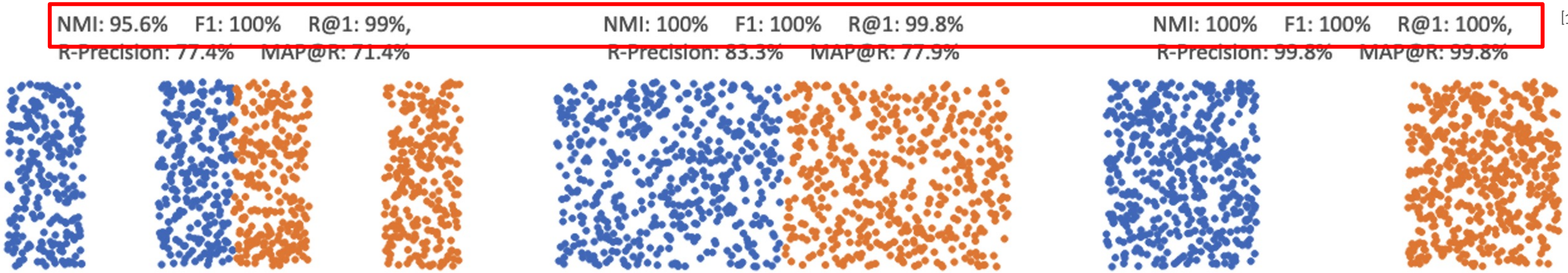
Questioning evaluation protocol

Are current evaluation metrics good?

Varying results NMI
(clustering and seeds)



NMI and R@k not robust



[11]

➡ Are there better evaluation metrics?

[11] Musgrave, K. et al "A Metric Learning Reality Check." ECCV (2020).

R-Presicion

3/5

3/5

$$\frac{r}{R}$$

R = total number of references for given query

r = number of references of same class in R-NN set

[11] Musgrave, K. et al "A Metric Learning Reality Check." ECCV (2020).

MAP@R

P@1 = 1 P@2 = 0

P@1 = 0 P@2 = 0

$$\text{MAP@R} = \frac{1}{R}\sum_{i=1}^{R} P(i)$$

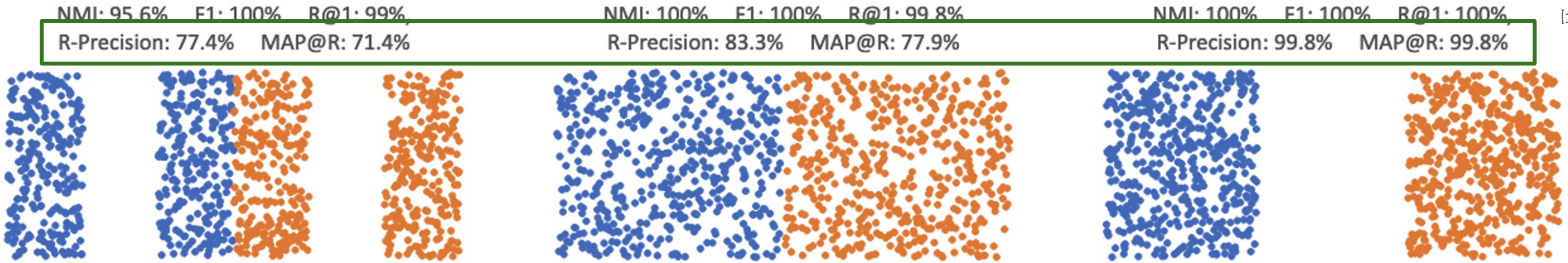
$$P(i) = \begin{cases} \text{precision at } i, & \text{if the ith retrieval is correct} \\ 0, & \text{otherwise} \end{cases}$$

[11] Musgrave, K. et al "A Metric Learning Reality Check." ECCV (2020).

Current vs. new evaluation metrics

R-Precision: $\frac{r}{R}$

MAP@R: $\frac{1}{R}\sum_{i=1}^{R} P(i)$

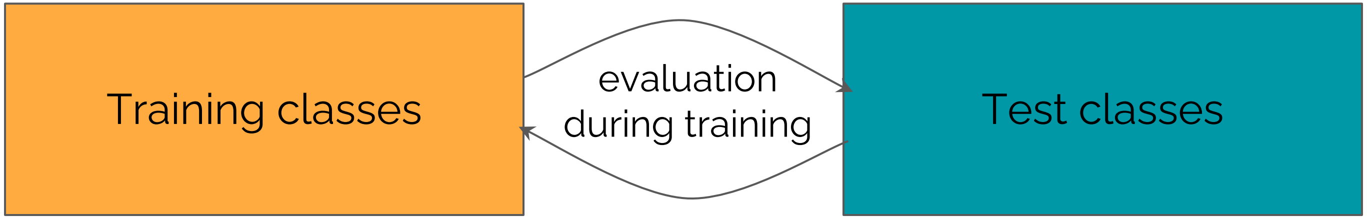


[11]

R-Precision and MAP@R more robust

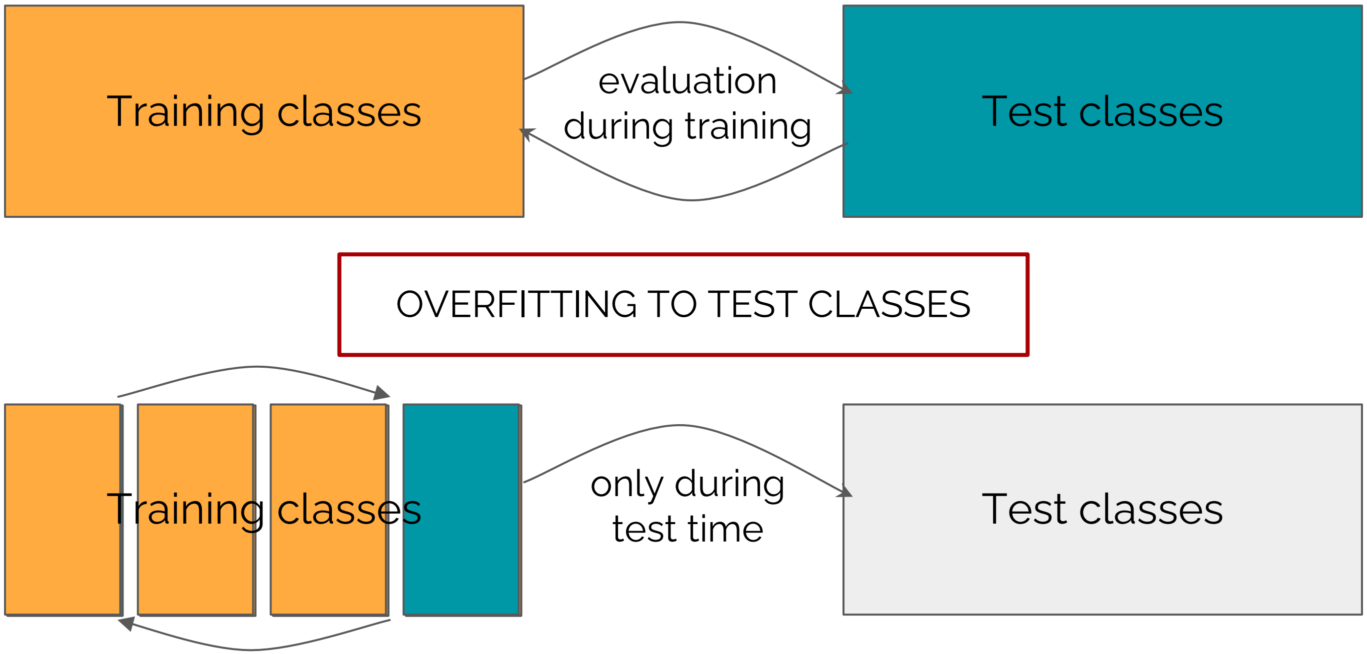
[11] Musgrave, K. et al "A Metric Learning Reality Check." ECCV (2020).

Training with Test set Feedback



OVERFITTING TO TEST CLASSES

Training with Test set Feedback



➡ Don't use test set feedback!

Metric Learning Reality Check

[11]

- The trunk model is an ImageNet [45] pretrained BN-Inception network [21], with output embedding size of 128. BatchNorm parameters are frozen during training, to reduce overfitting.
- The batch size is set to 32. Batches are constructed by first randomly sampling C classes, and then randomly sampling M images for each of the C classes. We set $C = 8$ and $M = 4$ for embedding losses, and $C = 32$ and $M = 1$ for classification losses.
- During training, images are augmented using the random resized cropping strategy. Specifically, we first resize each image so that its shorter side has length 256, then make a random crop that has a size between 40 and 256, and aspect ratio between 3/4 and 4/3. This crop is then resized to 227x227, and flipped horizontally with 50% probability. During evaluation, images are resized to 256 and then center cropped to 227.
- All network parameters are optimized using RMSprop with learning rate 1e-6. We chose RMSprop because it converges faster than SGD, and seems to generalize better than Adam, based on a small set of experiments. For loss functions that include their own learnable weights (e.g. ArcFace), we use RMSprop but leave the learning rate as a hyperparameter to be optimized.
- Embeddings are L2 normalized before computing the loss, and during evaluation.

[11] Musgrave, K. et al "A Metric Learning Reality Check." ECCV (2020).

Metric Learning Reality Check

CUB-200-2011

[11]

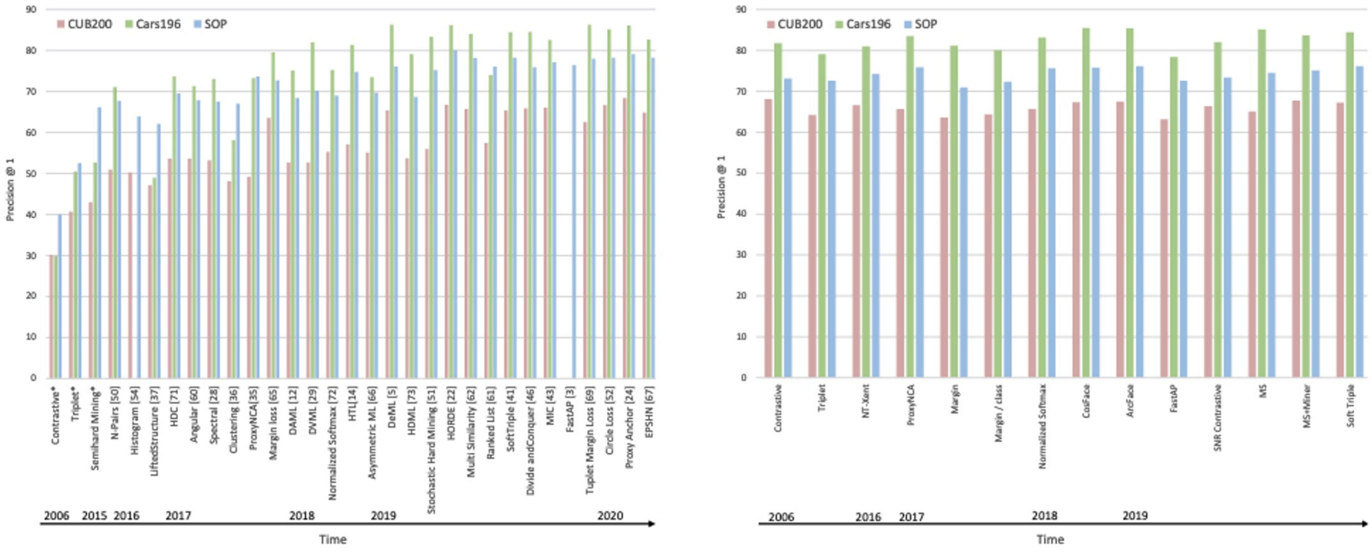
	Concatenated (512-dim)			Separated (128-dim)				
	P@1	**RP**	**MAP@R**	**P@1**	**RP**	**MAP@R**	year	loss
Pretrained	51.05	24.85	14.21	50.54	25.12	14.53		
Contrastive	**68.13 ± 0.31**	37.24 ± 0.28	26.53 ± 0.29	59.73 ± 0.40	31.98 ± 0.29	21.18 ± 0.28	2006	Embedding
Triplet	64.24 ± 0.26	34.55 ± 0.24	23.69 ± 0.23	55.76 ± 0.27	29.55 ± 0.16	18.75 ± 0.15	2006	Embedding
NT-Xent	66.61 ± 0.29	35.96 ± 0.21	25.09 ± 0.22	58.12 ± 0.23	30.81 ± 0.17	19.87 ± 0.16	2016	Embedding
ProxyNCA	65.69 ± 0.43	35.14 ± 0.26	24.21 ± 0.27	57.88 ± 0.30	30.16 ± 0.22	19.32 ± 0.21	2017	Classification
Margin	63.60 ± 0.48	33.94 ± 0.27	23.09 ± 0.27	54.78 ± 0.30	28.86 ± 0.18	18.11 ± 0.17	2017	Embedding
Margin/class	64.37 ± 0.18	34.59 ± 0.16	23.71 ± 0.16	55.56 ± 0.16	29.32 ± 0.15	18.51 ± 0.13	2017	Embedding
N. Softmax	65.65 ± 0.30	35.99 ± 0.15	25.25 ± 0.13	58.75 ± 0.19	31.75 ± 0.12	20.96 ± 0.11	2017	Classification
CosFace	67.32 ± 0.32	**37.49 ± 0.21**	**26.70 ± 0.23**	59.63 ± 0.36	31.99 ± 0.22	21.21 ± 0.22	2018	Classification
ArcFace	67.50 ± 0.25	37.31 ± 0.21	26.45 ± 0.20	**60.17 ± 0.32**	**32.37 ± 0.17**	**21.49 ± 0.16**	2019	Classification
FastAP	63.17 ± 0.34	34.20 ± 0.20	23.53 ± 0.20	55.58 ± 0.31	29.72 ± 0.16	19.09 ± 0.16	2019	Embedding
SNR	66.44 ± 0.56	36.56 ± 0.34	25.75 ± 0.36	58.06 ± 0.39	31.21 ± 0.28	20.43 ± 0.28	2019	Embedding
MS	65.04 ± 0.28	35.40 ± 0.12	24.70 ± 0.13	57.60 ± 0.24	30.84 ± 0.13	20.15 ± 0.14	2019	Embedding
MS+Miner	67.73 ± 0.18	37.37 ± 0.19	26.52 ± 0.18	59.41 ± 0.30	31.93 ± 0.15	21.01 ± 0.14	2019	Embedding
SoftTriple	67.27 ± 0.39	37.34 ± 0.19	26.51 ± 0.20	59.94 ± 0.33	32.12 ± 0.14	21.31 ± 0.14	2019	Classification

[11] Musgrave, K. et al "A Metric Learning Reality Check." ECCV (2020).

Metric Learning Reality Check

CUB-200-2011

[11]



(a) The trend according to papers

(b) The trend according to reality

[11] Musgrave, K. et al "A Metric Learning Reality Check." ECCV (2020).

Are standardized training strategies fair?

- Does every method require the same learning rate, weight decay, and batch size to perform best?

- Should we not use current best performing optimizers and augmentation techniques but stick with “old” stuff?

➡ Optimally: report standard protocol as well as the best you can get!

➡ Take current SOTA results with a grain of salt

https://github.com/KevinMusgrave/powerful-benchmarker

https://github.com/KevinMusgrave/pytorch-metric-learning

References

[1] Goldberger, Jacob, Sam T. Roweis, Geoffrey E. Hinton and Ruslan Salakhutdinov. “Neighbourhood Components Analysis.” NIPS (2004).
[2] Movshovitz-Attias, Yair, Alexander Toshev, Thomas Leung, Sergey Ioffe and Saurabh Singh. “No Fuss Distance Metric Learning Using Proxies.” ICCV (2017)
[3] Kim, Sungyeon, Dongwon Kim, Minsu Cho and Suha Kwak. “Proxy Anchor Loss for Deep Metric Learning.” CVPR (2020)
[4] Zhu, Yuehua, Muli Yang, Cheng Deng and Wei Liu. “Fewer is More: A Deep Graph Metric Learning Perspective Using Fewer Proxies.” NeurIPS (2020)
[5] https://en.wikipedia.org/wiki/Mutual_information#/media/File:Entropy-mutual-information-relative-entropy-relation-diagram.svg
[6] Teh, Eu Wern, Terrance Devries and Graham W. Taylor. “ProxyNCA++: Revisiting and Revitalizing Proxy Neighborhood Component Analysis.” ECCV (2020)
[7] Yu, Dingjun, Hanli Wang, Peiqiu Chen and Zhihua Wei. “Mixed Pooling for Convolutional Neural Networks.” RSKT (2014).
[8] https://medium.com/@bdhuma/which-pooling-method-is-better-maxpooling-vs-minpooling-vs-average-pooling-95fb03f45a9
[9] Elezi, Ismail, Seidenschwarz, Jenny, Wagner, Laurin, Vascon, Sebastiano, Torcinovich, Alessandro, Pelillo, Marcello, Leal-Taixe, Laura. “The Group Loss++: A deeper look into group loss for deep metric learning”, PAMI (2022/03)
[10] Seidenschwarz, Jenny, Ismail Elezi and Laura Leal-Taix'e. “Learning Intra-Batch Connections for Deep Metric Learning.” ICML (2021).
[11] Musgrave, Kevin, Serge J. Belongie and Ser-Nam Lim. “A Metric Learning Reality Check.” ECCV (2020).
[12] Roth, Karsten, Timo Milbich, Samarth Sinha, Prateek Gupta, Bjoern Ommer and Joseph Paul Cohen. “Revisiting Training Strategies and Generalization Performance in Deep Metric Learning.” ICML (2020)
[13] Kemelmacher-Shlizerman, Ira, Steven M. Seitz, Daniel Miller and Evan Brossard. “The MegaFace Benchmark: 1 Million Faces for Recognition at Scale.” CVPR (2016)
[14] Sun, Yifan, Liang Zheng, Yi Yang, Qi Tian and Shengjin Wang. “Beyond Part Models: Person Retrieval with Refined Part Pooling.” ECCV (2018).
[15] Chen, Tianlong, Shaojin Ding, Jingyi Xie, Ye Yuan, Wuyang Chen, Yang Yang, Zhou Ren and Zhangyang Wang. “ABD-Net: Attentive but Diverse Person Re-Identification.” 2019 ICCV) (2019)
[16] Tay, Chiat-Pin, Sharmili Roy and Kim-Hui Yap. “AANet: Attribute Attention Network for Person Re-Identifications.” CVPR (2019)
[17] Zhu, Zhihui, Xinyang Jiang, Feng Zheng, Xiao-Wei Guo, Feiyue Huang, Weishi Zheng and Xing Sun. “Viewpoint-Aware Loss with Angular Regularization for Person Re-Identification.” AAAI (2020).
[18] Luo, Haowen, Youzhi Gu, Xingyu Liao, Shenqi Lai and Wei Jiang. “Bags of Tricks and A Strong Baseline for Deep Person Re-identification.” (2019).
[19] Vinyals, Oriol, Charles Blundell, Timothy P. Lillicrap, Koray Kavukcuoglu and Daan Wierstra. “Matching Networks for One Shot Learning.” NIPS (2016).
[20] Chen, Ting, Simon Kornblith, Mohammad Norouzi and Geoffrey E. Hinton. “A Simple Framework for Contrastive Learning of Visual Representations.” ArXiv abs/2002.05709 (2020): n. pag.
[21] Matching networks for one shot learning. In NIPS, 2016.
[22] Prototypical networks for few-shot learning. In NIPS, 2017.
[23] Liu, W. et al. “SphereFace: Deep Hypersphere Embedding for Face Recognition.” (CVPR, 2017)
[24] Wang, H. et al. “CosFace: Large Margin Cosine Loss for Deep Face Recognition.” (CVPR 2018)
[25] Deng, J. et al. “ArcFace: Additive Angular Margin Loss for Deep Face Recognition.” (CVPR 2019)
[26] Li, W. et al. “DeepReID: Deep Filter Pairing Neural Network for Person Re-identification” (CVPR 2014)
[27] Leal-Taixé, L., Milan, A., Reid, I., Roth, S. & Schindler, K. MOTChallenge 2015: Towards a Benchmark for Multi-Target Tracking. arXiv:1504.01942 [cs], 2015., (arXiv: 1504.01942).
[28] Chen, Ting et al. “A Simple Framework for Contrastive Learning of Visual Representations.” (arxiv 2020)
[29] Toker, Aysim et al. “Coming Down to Earth: Satellite-to-Street View Synthesis for Geo-Localization.” (CVPR, 2021)
[30] Jegou, H et al. “Product quantization for nearest neighbor search.” (tPAMI 2011)
[31]]McDaid, A. et al. “Normalized mutual information to evaluate overlapping community finding algorithms.” (arxiv 2011)
[32] Wah, C. et al.”The Caltech-UCSD Birds-200-2011 Dataset.” (Technical Report 2011)
[33] Krause, J. et al. “3d object representations for fine-grained categorization.” (Workshop on 3D Representation and Recognition, 2013.)
[34] Song, H. et al. “Deep metric learning via lifted structured feature embedding.” (CVPR 2016)
[35] Liu, Z. et al. “Deepfashion: Powering robust clothes recognition and retrieval with rich annotations” (CVPR 2016)
[36] Zhai, A. and Wu, H. Classification is a strong baseline for deep metric learning. (BMVC 2019).
[37] Zhong, Z. et al. “Re-ranking Person Re-identification with k-reciprocal Encoding” (CVPR 2017)
[38] Ermolov, A.et al. “Hyperbolic Vision Transformers: Combining Improvements in Metric Learning” (CVPR 2022)
[39] Ramzi, E. et al. “Robust and Decomposable Average Precision for Image Retrieval” (NeurIPS 2021)
[40] Patel, Y. et al. “Recall@k Surrogate Loss with Large Batches and Similarity Mixup” (CVPR 2022)