Artificial Intelligence

Lecture 12：Deep Learning II

Xiaojin Gong

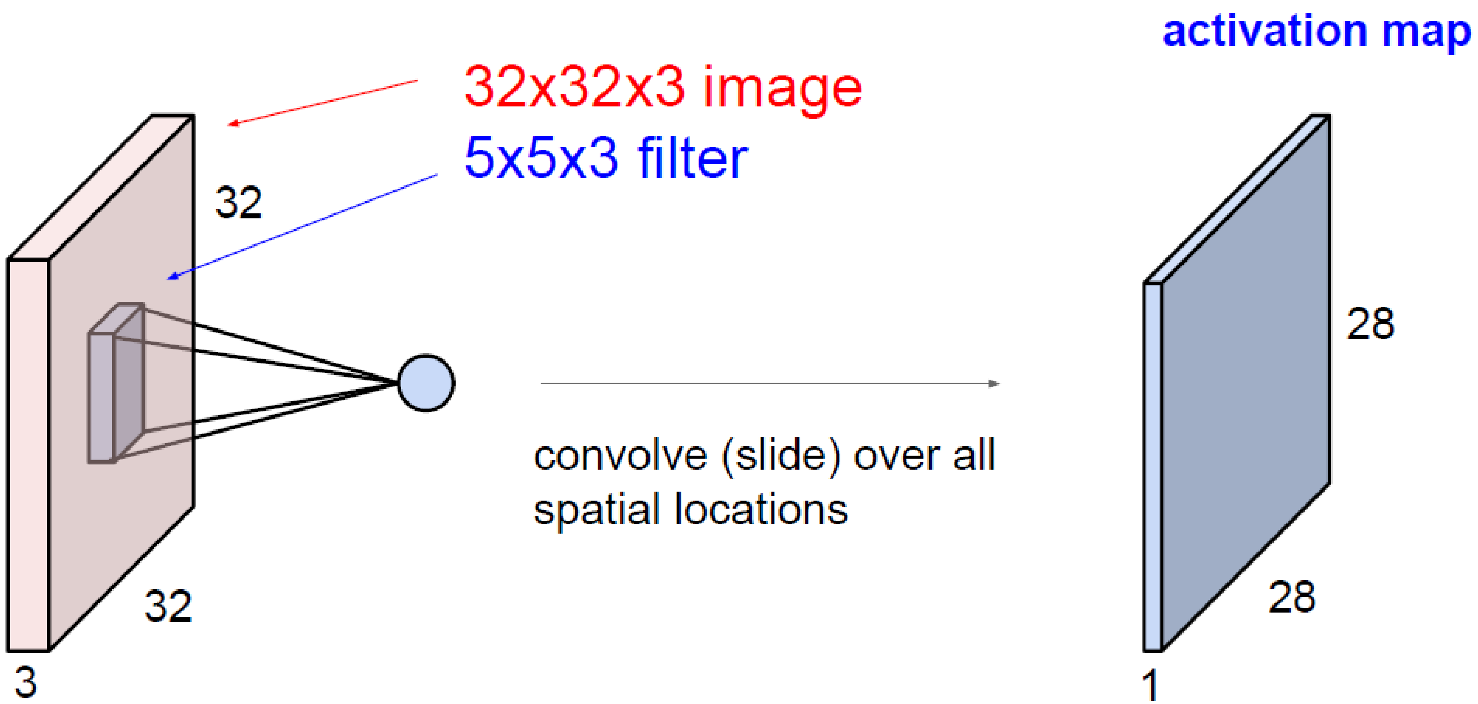
2022-05-23

Outline

- Attentions
  - Channel attention
  - Spatial attention
  - Self-attention / Transformer
- Unsupervised Learning
  - Unsupervised feature representation learning
  - Unsupervised person re-identification

Review: CNN

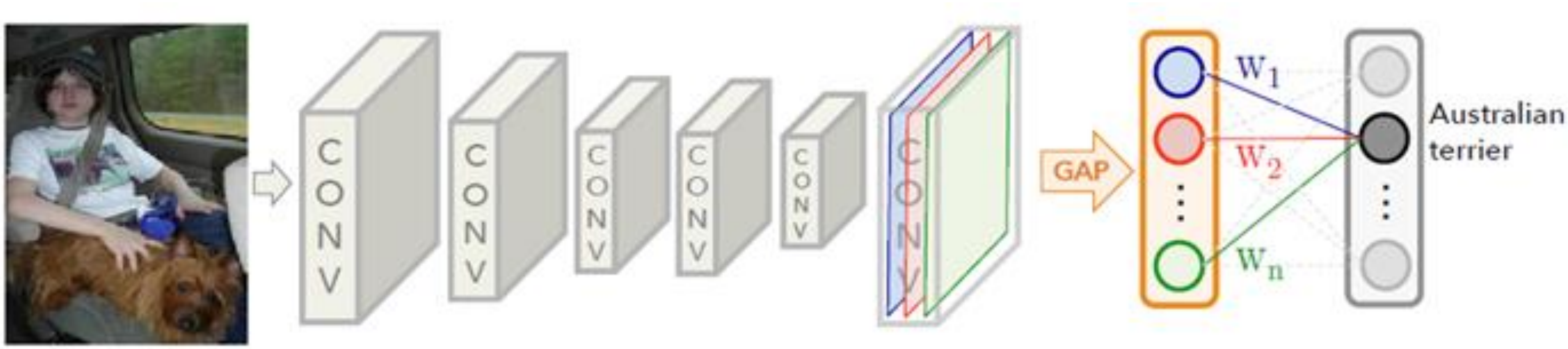
- The convolutional layer



- Cons:
  - Channel dependencies and spatial correlations are entangled
  - Process one local neighborhood at a time

Review: CNN

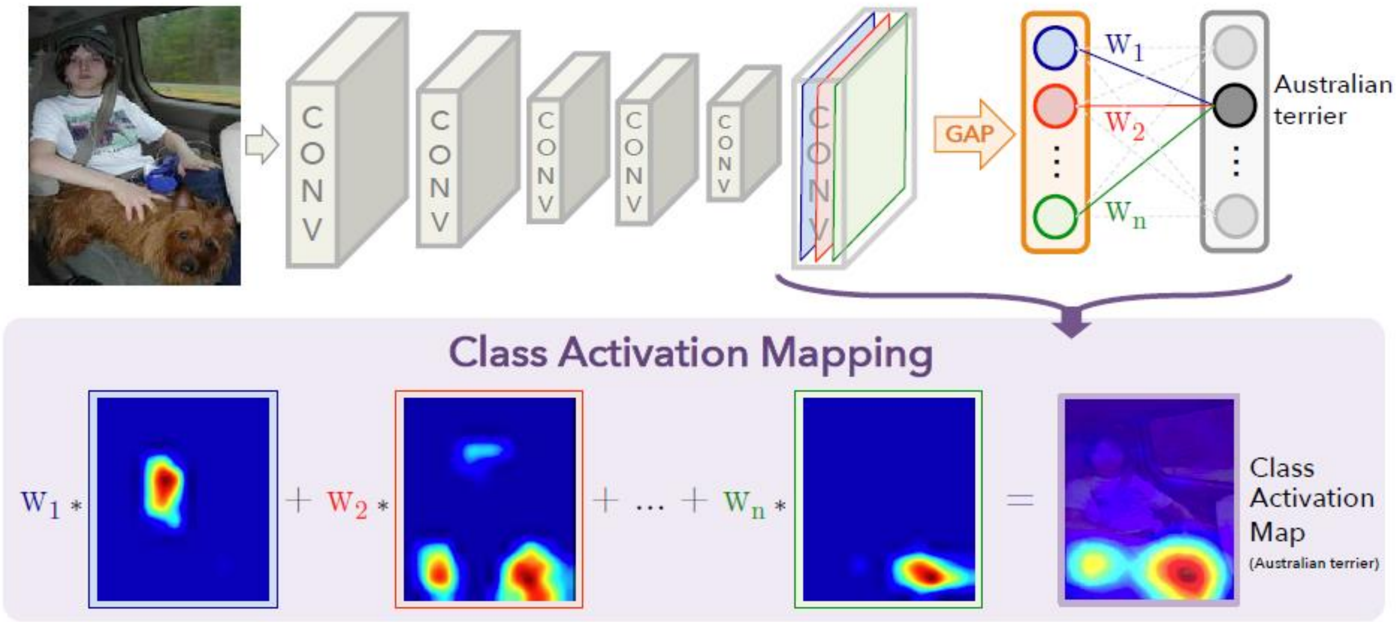
- The convolutional neural network



$$S_c = \sum_k w_k^c F_k = \sum_k w_k^c \sum_{x,y} f_k(x,y)$$

Class Activation Maps

- CAM – to estimate class activation maps using global average pooling



- Class score:

$$S_c = \sum_k w_k^c \sum_{x,y} f_k(x,y) = \sum_{x,y} \sum_k w_k^c f_k(x,y)$$

$$= \sum_{x,y} M_c(x,y)$$

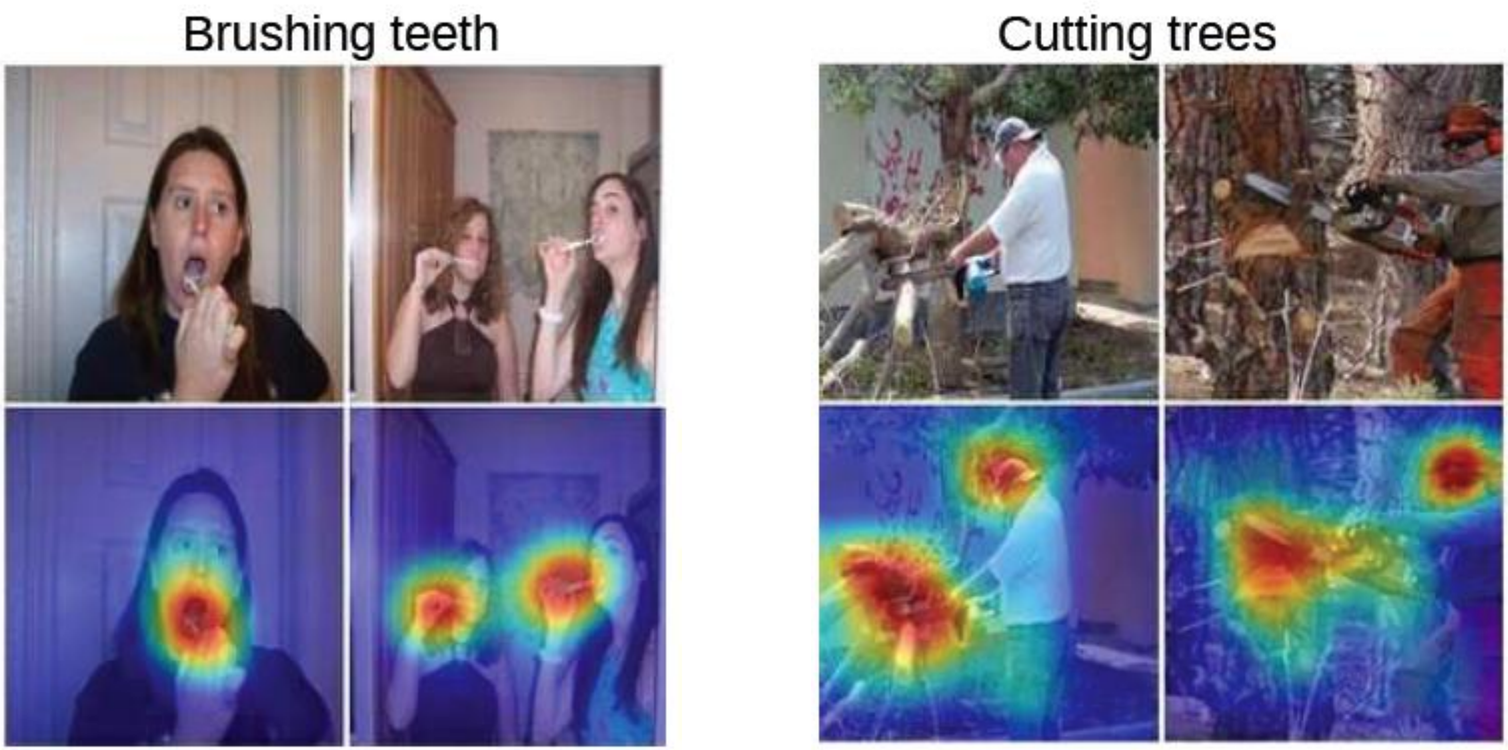
- Class activation map:

$$M_c(x,y) = \sum_k w_k^c f_k(x,y)$$

Zhou, et al. Learning Deep Features for Discriminative Localization, CVPR 2016.

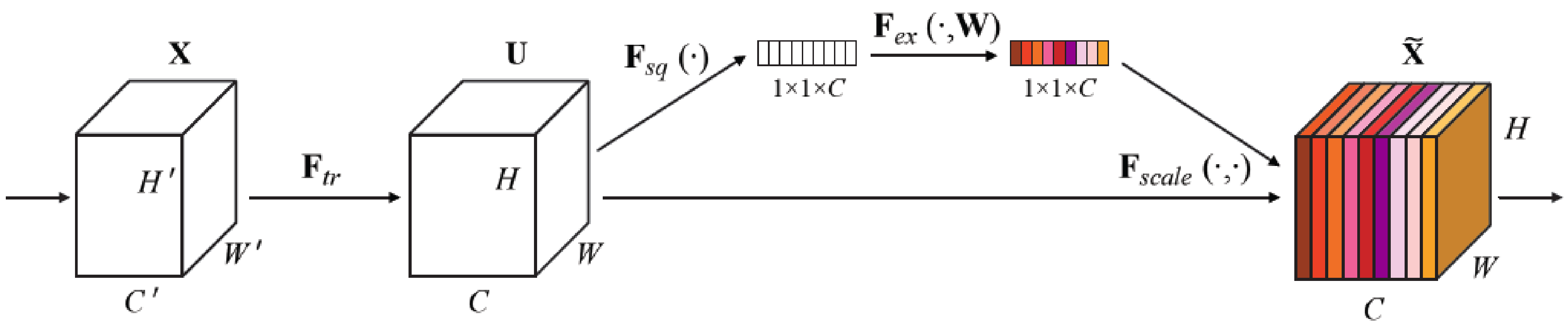
Class Activation Maps

- CAM – to estimate class activation maps using global average pooling



Squeeze and Excitation Networks

- Convolution – channel dependencies and spatial correlations are entangl
- SENet – to explicitly model the interdependencies between the channels
- – won the 1st place in ILSVRC 2017 classification competition.

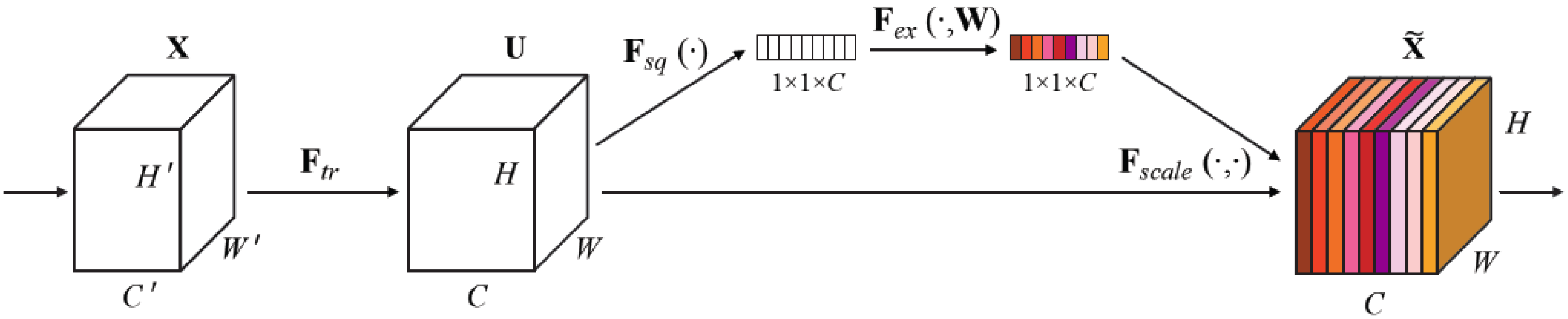


SE block

J. Hu, et al. Squeeze and Excitation Networks, CVPR 2018.

Squeeze and Excitation Networks

SE block



- Squeeze: global information embedding

$$z_c = \mathbf{F}_{sq}(\mathbf{u}_c) = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} u_c(i,j)$$

- Excitation: reduce dimensionality to prevent overfitting & reduce complexity

$$\mathbf{s} = \mathbf{F}_{ex}(\mathbf{z}, \mathbf{W}) = \sigma(g(\mathbf{z}, \mathbf{W})) = \sigma(\mathbf{W}_2 \cdot \mathrm{ReLU}(\mathbf{W}_1 \mathbf{z}))$$

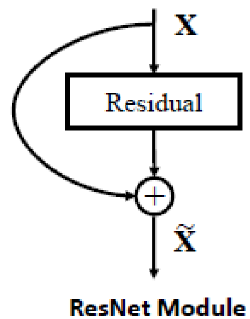
$\mathbf{W}_1 \in \mathbb{R}^{\frac{C}{r} \times C}$

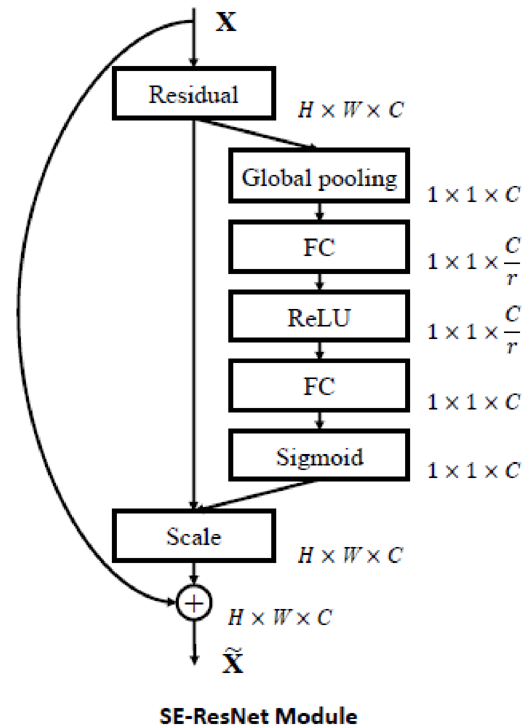
$\mathbf{W}_2 \in \mathbb{R}^{C \times \frac{C}{r}}$

- Scaling: to enhance features

$$\widetilde{\mathbf{x}}_c = \mathbf{F}_{scale}(\mathbf{u}_c, s_c) = s_c \cdot \mathbf{u}_c$$

Squeeze and Excitation Networks

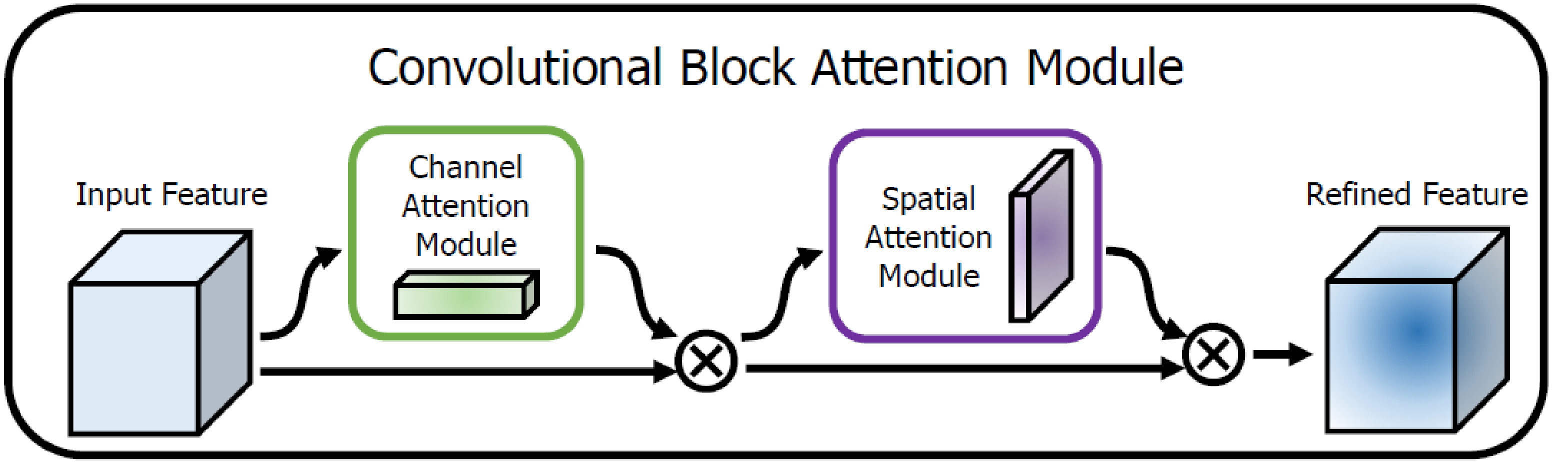




	original		re-implementation			SENet		
	top-1 err.	top-5 err.	top-1err.	top-5 err.	GFLOPs	top-1 err.	top-5 err.	GFLOPs
ResNet-50 [10]	24.7	7.8	24.80	7.48	3.86	$23.29_{(1.51)}$	$6.62_{(0.86)}$	3.87
ResNet-101 [10]	23.6	7.1	23.17	6.52	7.58	$22.38_{(0.79)}$	$6.07_{(0.45)}$	7.60
ResNet-152 [10]	23.0	6.7	22.42	6.34	11.30	$21.57_{(0.85)}$	$5.73_{(0.61)}$	11.32
ResNeXt-50 [47]	22.2	-	22.11	5.90	4.24	$21.10_{(1.01)}$	$5.49_{(0.41)}$	4.25
ResNeXt-101 [47]	21.2	5.6	21.18	5.57	7.99	$20.70_{(0.48)}$	$5.01_{(0.56)}$	8.00
VGG-16 [39]	-	-	27.02	8.81	15.47	$25.22_{(1.80)}$	$7.70_{(1.11)}$	15.48
BN-Inception [16]	25.2	7.82	25.38	7.89	2.03	$24.23_{(1.15)}$	$7.14_{(0.75)}$	2.04
Inception-ResNet-v2 [42]	$19.9^{\dagger}$	$4.9^{\dagger}$	20.37	5.21	11.75	$19.80_{(0.57)}$	$4.79_{(0.42)}$	11.76

Convolutional Block Attention Module

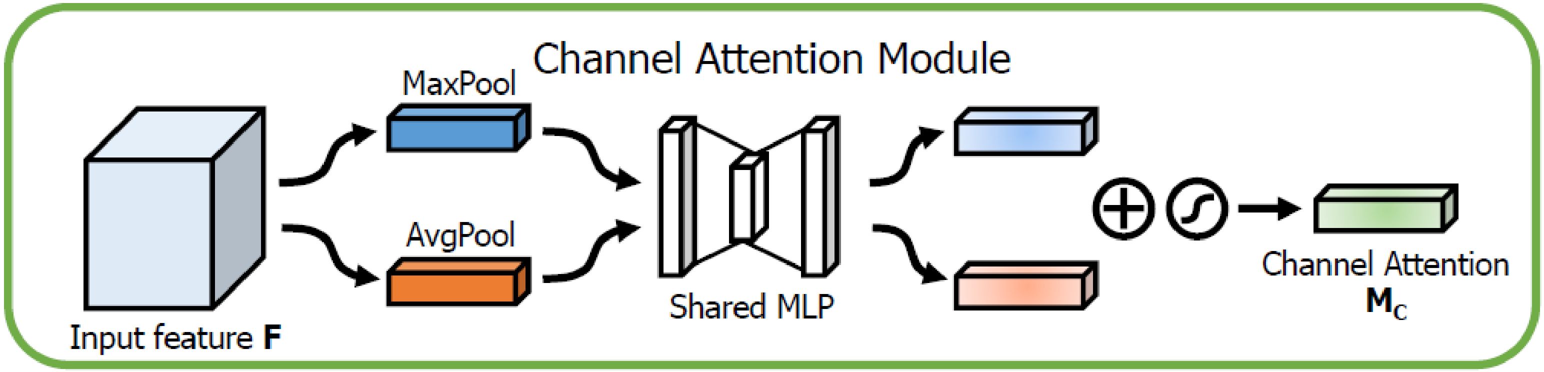
- CBAM – Channel attention + spatial attention



Woo, et al. CBAM: Convolutional Block Attention Module, ECCV 2018.

Convolutional Block Attention Module

- CBAM – Channel attention + spatial attention

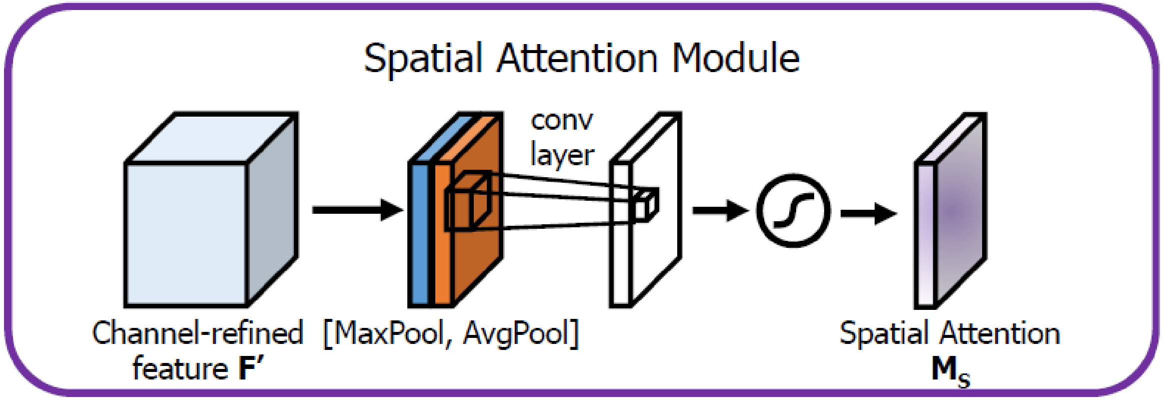


$$\mathbf{M_c}(\mathbf{F}) = \sigma(MLP(AvgPool(\mathbf{F})) + MLP(MaxPool(\mathbf{F}))) = \sigma(\mathbf{W_1}(\mathbf{W_0}(\mathbf{F^c_{avg}})) + \mathbf{W_1}(\mathbf{W_0}(\mathbf{F^c_{max}}))),$$

Spatial pooling

Convolutional Block Attention Module

- CBAM – Channel attention + spatial attention



$$
\begin{aligned}
\mathbf{M_s}(\mathbf{F}) \quad &\in \mathbf{R}^{H \times W} \\
&= \sigma(f^{7\times 7}([AvgPool(\mathbf{F}); MaxPool(\mathbf{F})])) \\
&= \sigma(f^{7\times 7}([\mathbf{F}^{\mathbf{s}}_{\mathbf{avg}}; \mathbf{F}^{\mathbf{s}}_{\mathbf{max}}])),
\end{aligned}
$$

Channel pooling

Convolutional Block Attention Module

- CBAM – Channel attention + spatial attention

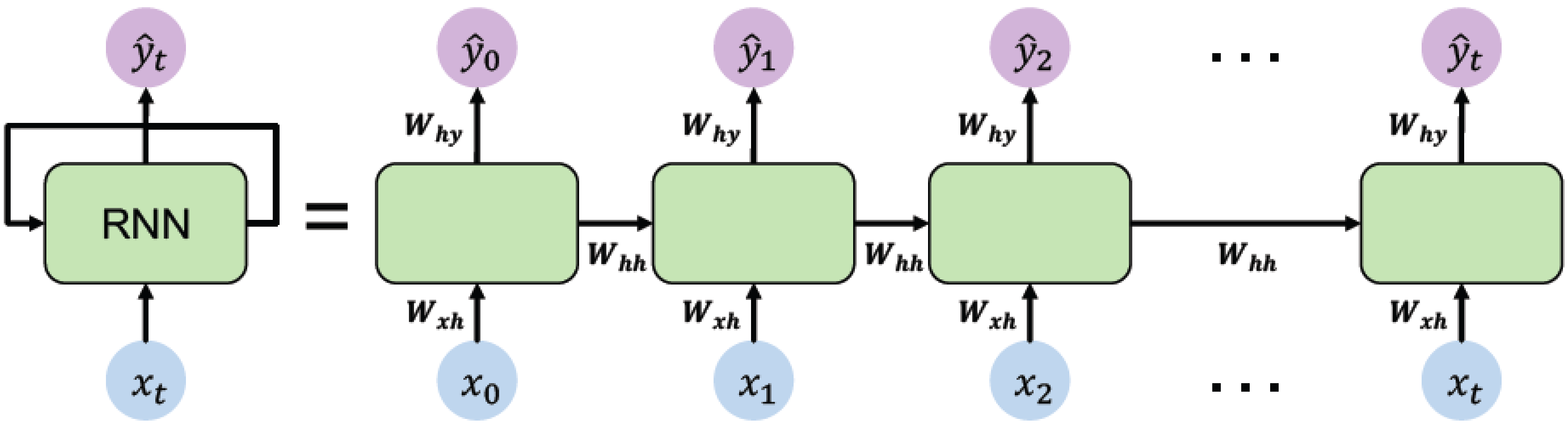
Architecture	Param.	GFLOPs	Top-1 Error (%)	Top-5 Error (%)
ResNet18 [5]	11.69M	1.814	29.60	10.55
ResNet18 [5] + SE [28]	11.78M	1.814	29.41	10.22
ResNet18 [5] + CBAM	11.78M	1.815	**29.27**	**10.09**
ResNet34 [5]	21.80M	3.664	26.69	8.60
ResNet34 [5] + SE [28]	21.96M	3.664	26.13	8.35
ResNet34 [5] + CBAM	21.96M	3.665	**25.99**	**8.24**
ResNet50 [5]	25.56M	3.858	24.56	7.50
ResNet50 [5] + SE [28]	28.09M	3.860	23.14	6.70
ResNet50 [5] + CBAM	28.09M	3.864	**22.66**	**6.31**
ResNet101 [5]	44.55M	7.570	23.38	6.88
ResNet101 [5] + SE [28]	49.33M	7.575	22.35	6.19
ResNet101 [5] + CBAM	49.33M	7.581	**21.51**	**5.69**

Outline

- Attentions
  - Channel attention
  - Spatial attention
  - Self-attention / Transformer
- Unsupervised Learning
  - Unsupervised feature representation learning
  - Unsupervised person re-identification

Review: RNN

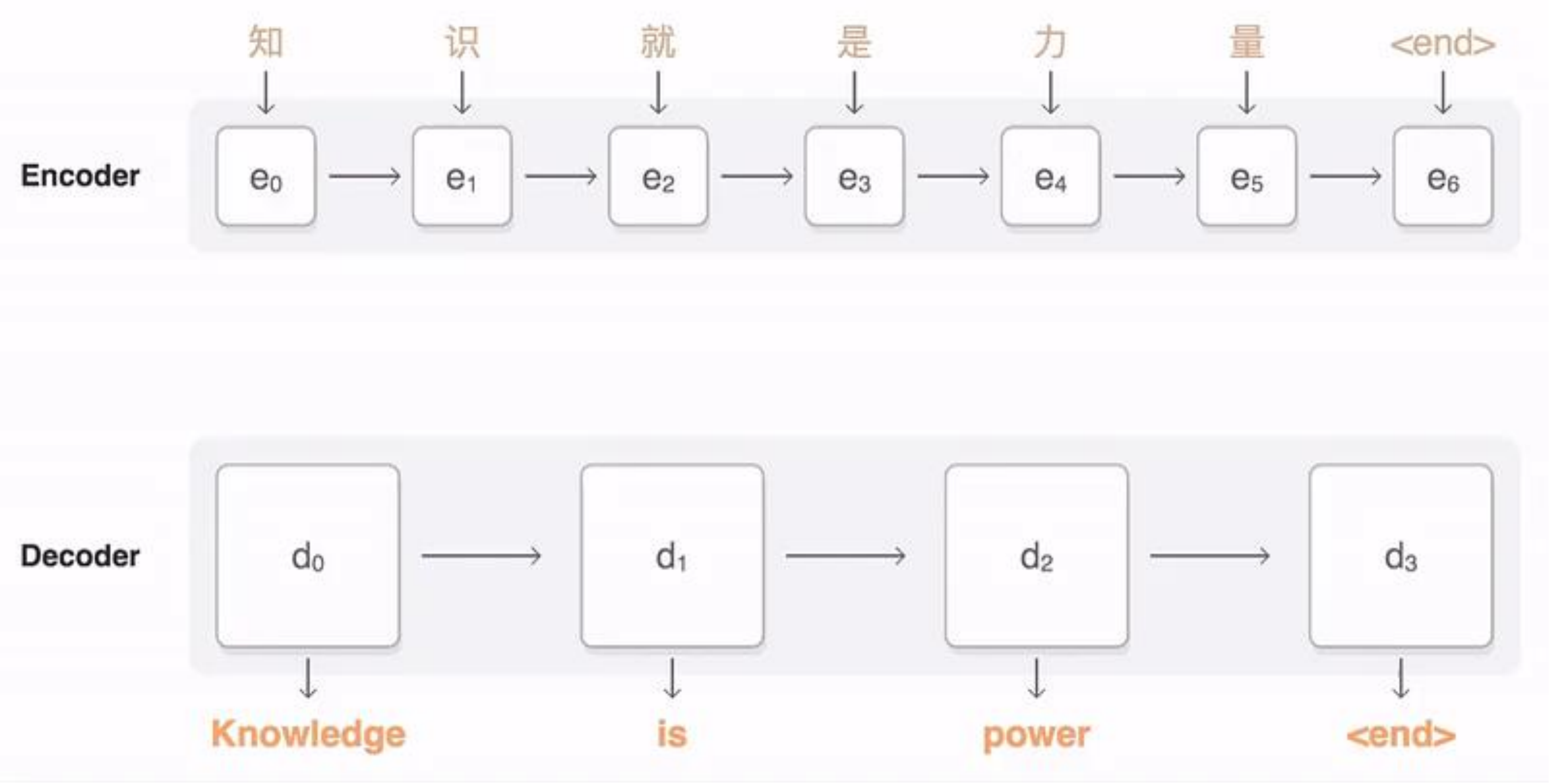
Re-use the **same weight matrices** at every time step



- Cons:
  - Vanishing gradients
  - The inherently sequential nature precludes parallelization
  - Ineffective to capture long-term dependencies

Attention is All You Need

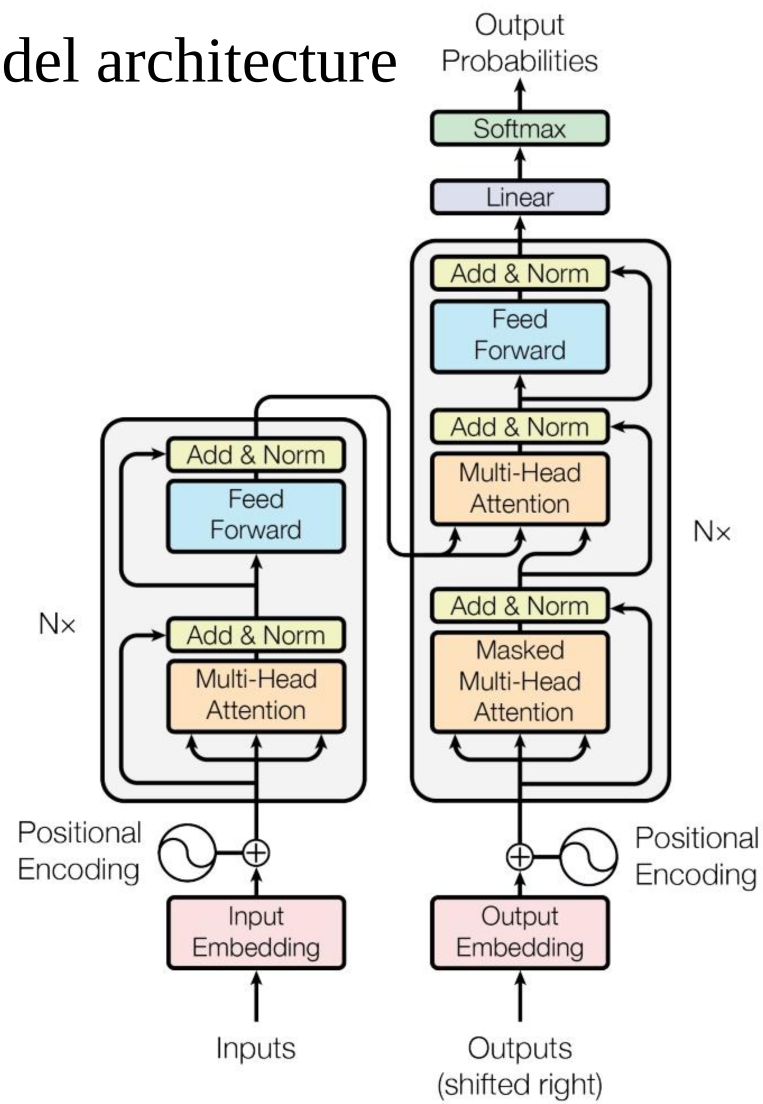
- Example



Vaswani, et al. Attention Is All You Need, NIPS 2017.

Attention is All You Need

- The transformer model architecture

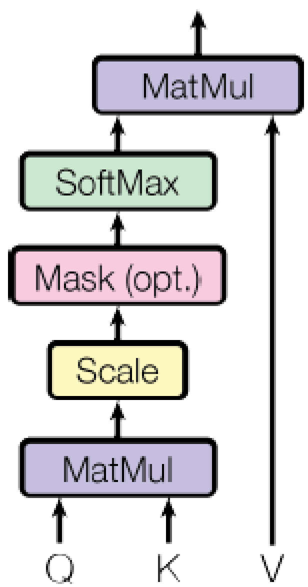


Vaswani, et al. Attention Is All You Need, NIPS 2017.

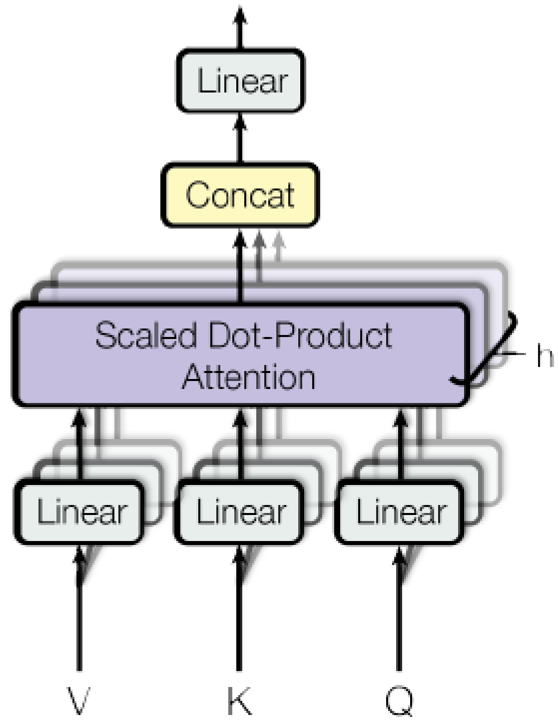
Attention is All You Need

- Self-attention modules

Scaled Dot-Product Attention



Multi-Head Attention



$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V$$

- Q = the current position-word vector in the input sequence
- K = all the position-word vectors in the input sequence
- V = all the position-word vectors in the input sequence

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, ..., \text{head}_h)W^O$$
$$\text{where head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V)$$

https://medium.com/@b.terryjack/deep-learning-the-transformer-9ae5e9c5a190

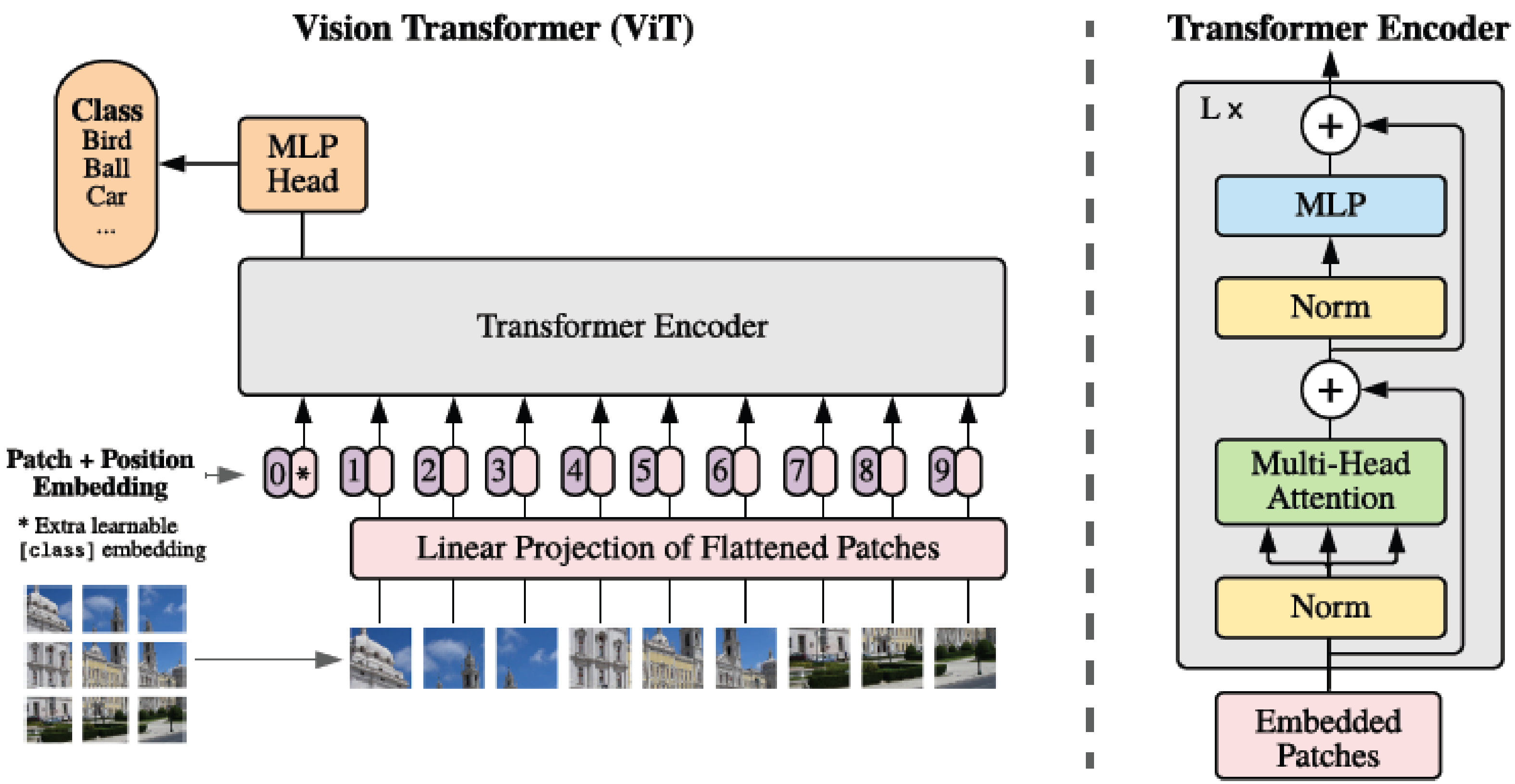
Attention is All You Need

Table 2: The Transformer achieves better BLEU scores than previous state-of-the-art models on the English-to-German and English-to-French newstest2014 tests at a fraction of the training cost.

Model	BLEU		Training Cost (FLOPs)	
	EN-DE	EN-FR	EN-DE	EN-FR
ByteNet [15]	23.75			
Deep-Att + PosUnk [32]		39.2		$1.0 \cdot 10^{20}$
GNMT + RL [31]	24.6	39.92	$2.3 \cdot 10^{19}$	$1.4 \cdot 10^{20}$
ConvS2S [8]	25.16	40.46	$9.6 \cdot 10^{18}$	$1.5 \cdot 10^{20}$
MoE [26]	26.03	40.56	$2.0 \cdot 10^{19}$	$1.2 \cdot 10^{20}$
Deep-Att + PosUnk Ensemble [32]		40.4		$8.0 \cdot 10^{20}$
GNMT + RL Ensemble [31]	26.30	41.16	$1.8 \cdot 10^{20}$	$1.1 \cdot 10^{21}$
ConvS2S Ensemble [8]	26.36	**41.29**	$7.7 \cdot 10^{19}$	$1.2 \cdot 10^{21}$
Transformer (base model)	27.3	38.1	$\mathbf{3.3 \cdot 10^{18}}$	
Transformer (big)	**28.4**	**41.0**	$2.3 \cdot 10^{19}$	

Transformers for Image Recognition

- Vision transformer



Dosovitskiy, et al. An Image is Worth 16*16 Words: Transformers for Image Recognition at Scale, ICLR 2021..

Transformers for Image Recognition

- Vision transformer

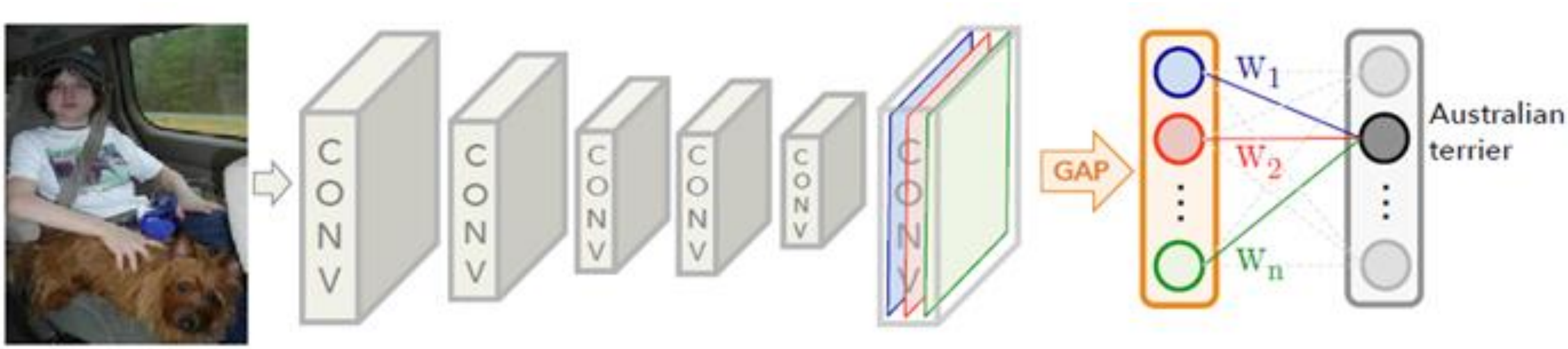
	Ours-JFT (ViT-H/14)	Ours-JFT (ViT-L/16)	Ours-I21k (ViT-L/16)	BiT-L (ResNet152x4)	Noisy Student (EfficientNet-L2)
ImageNet	**88.55** $\pm$ 0.04	87.76 $\pm$ 0.03	85.30 $\pm$ 0.02	87.54 $\pm$ 0.02	88.4/88.5*
ImageNet ReaL	**90.72** $\pm$ 0.05	90.54 $\pm$ 0.03	88.62 $\pm$ 0.05	90.54	90.55
CIFAR-10	**99.50** $\pm$ 0.06	99.42 $\pm$ 0.03	99.15 $\pm$ 0.03	99.37 $\pm$ 0.06	–
CIFAR-100	**94.55** $\pm$ 0.04	93.90 $\pm$ 0.05	93.25 $\pm$ 0.05	93.51 $\pm$ 0.08	–
Oxford-IIIT Pets	**97.56** $\pm$ 0.03	97.32 $\pm$ 0.11	94.67 $\pm$ 0.15	96.62 $\pm$ 0.23	–
Oxford Flowers-102	99.68 $\pm$ 0.02	**99.74** $\pm$ 0.00	99.61 $\pm$ 0.02	99.63 $\pm$ 0.03	–
VTAB (19 tasks)	**77.63** $\pm$ 0.23	76.28 $\pm$ 0.46	72.72 $\pm$ 0.21	76.29 $\pm$ 1.70	–
TPUv3-core-days	2.5k	0.68k	0.23k	9.9k	12.3k

Outline

- Attentions
  - Channel attention
  - Spatial attention
  - Self-attention / Transformer
- Unsupervised Learning
  - Unsupervised feature representation learning
  - Unsupervised person re-identification

Review: CNN

- The convolutional neural network

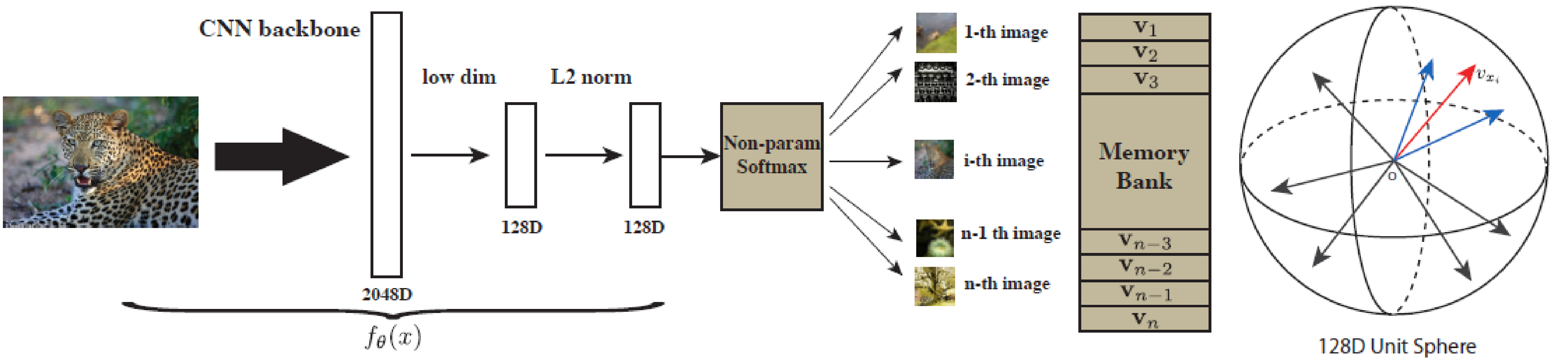


$$J(\boldsymbol{\theta}) = -\sum_{i=1}^{n} \log P(i|f_{\boldsymbol{\theta}}(x_i)).$$

$$P(i|\mathbf{v}) = \frac{\exp\left(\mathbf{w}_i^T\mathbf{v}\right)}{\sum_{j=1}^{n}\exp\left(\mathbf{w}_j^T\mathbf{v}\right)} \qquad \mathbf{v} = f_{\boldsymbol{\theta}}(x).$$

Non-Parametric Instance Discrimination

- Model



- Non-Parametric classifier

$$J(\theta) = -\sum_{i=1}^{n} \log P(i|f_{\boldsymbol{\theta}}(x_i)).$$

$$P(i|\mathbf{v}) = \frac{\exp\left(\mathbf{v}_i^T \mathbf{v}/\tau\right)}{\sum_{j=1}^{n} \exp\left(\mathbf{v}_j^T \mathbf{v}/\tau\right)}$$

- Memory bank

- Noise contrastive estimation

Wu et al. Unsupervised Feature Learning via Non-Parametric Instance Discrimination, CVPR 2018

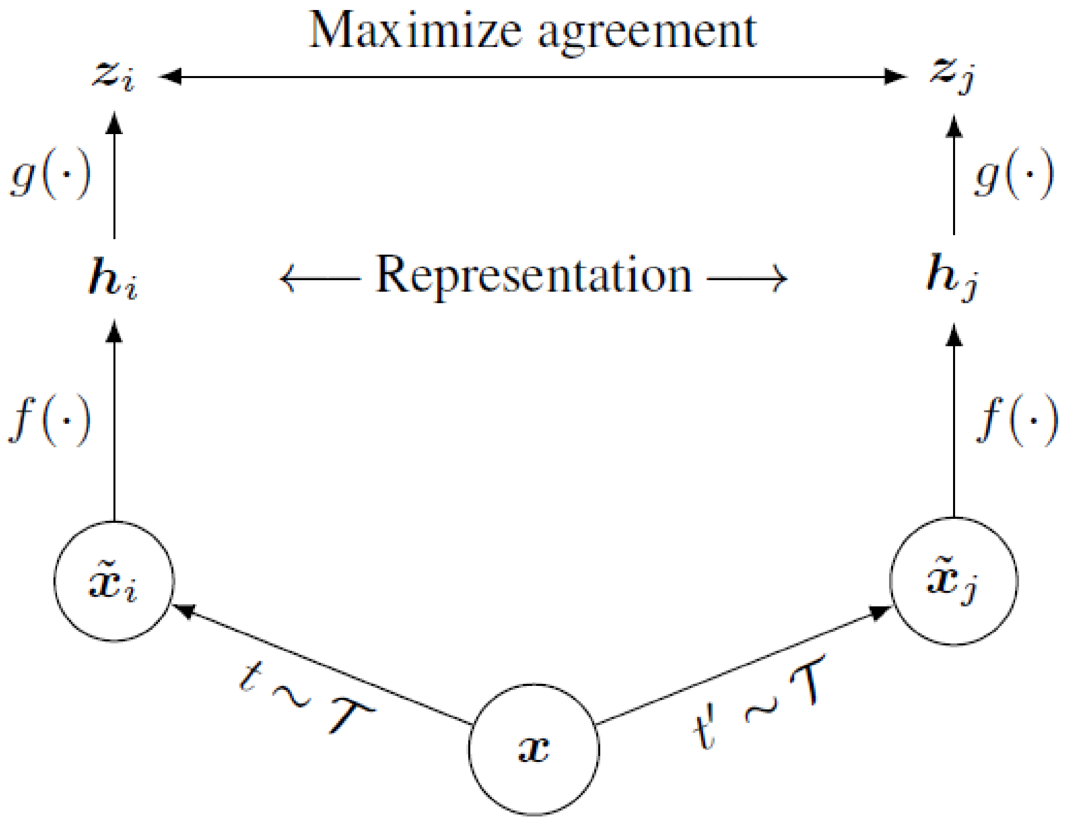
Non-Parametric Instance Discrimination

Image Classification Accuracy on ImageNet							
method	conv1	conv2	conv3	conv4	conv5	kNN	#dim
Random	11.6	17.1	16.9	16.3	14.1	3.5	10K
Data-Init [16]	17.5	23.0	24.5	23.2	20.6	-	10K
Context [2]	16.2	23.3	30.2	31.7	29.6	-	10K
Adversarial [4]	17.7	24.5	31.0	29.9	28.0	-	10K
Color [47]	13.1	24.8	31.0	32.6	31.8	-	10K
Jigsaw [27]	19.2	30.1	34.7	33.9	28.3	-	10K
Count [28]	18.0	30.6	34.3	32.5	25.7	-	10K
SplitBrain [48]	17.7	29.3	35.4	35.2	32.8	11.8	10K
Exemplar[3]			31.5			-	4.5K
Ours Alexnet	16.8	26.5	31.8	34.1	**35.6**	31.3	128
Ours VGG16	16.5	21.4	27.6	33.1	**37.2**	33.9	128
Ours Resnet18	16.0	19.9	26.3	35.7	**42.1**	**40.5**	128
Ours Resnet50	15.3	18.8	24.4	35.3	**43.9**	**42.5**	128

Table 2: Top-1 classification accuracies on ImageNet.

SimCLR

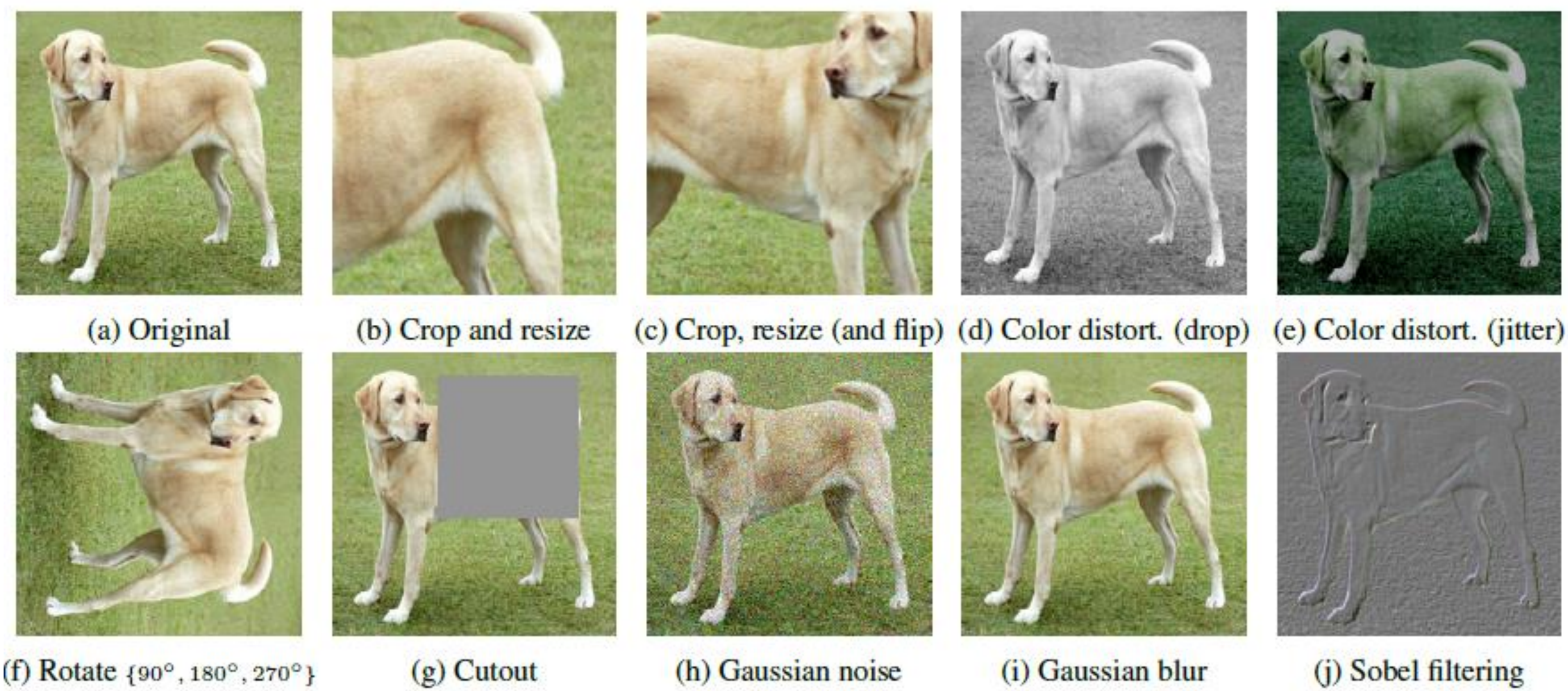
- A simple framework for contrastive learning



Chen et al. A Simple Framework for Contrastive Learning of Visual Representations, ICML 2020.

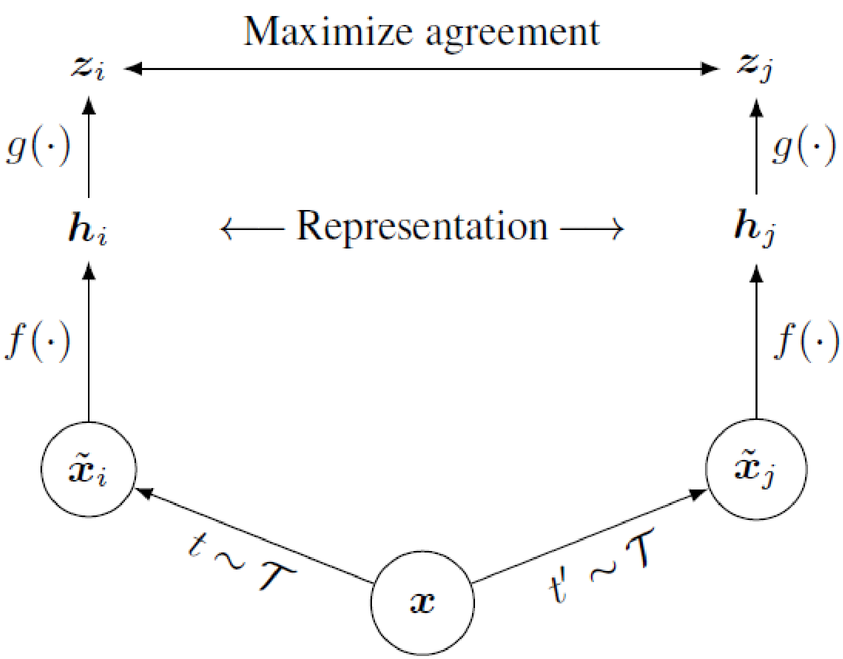
SimCLR

- Data augmentation



SimCLR

- Contrastive learning



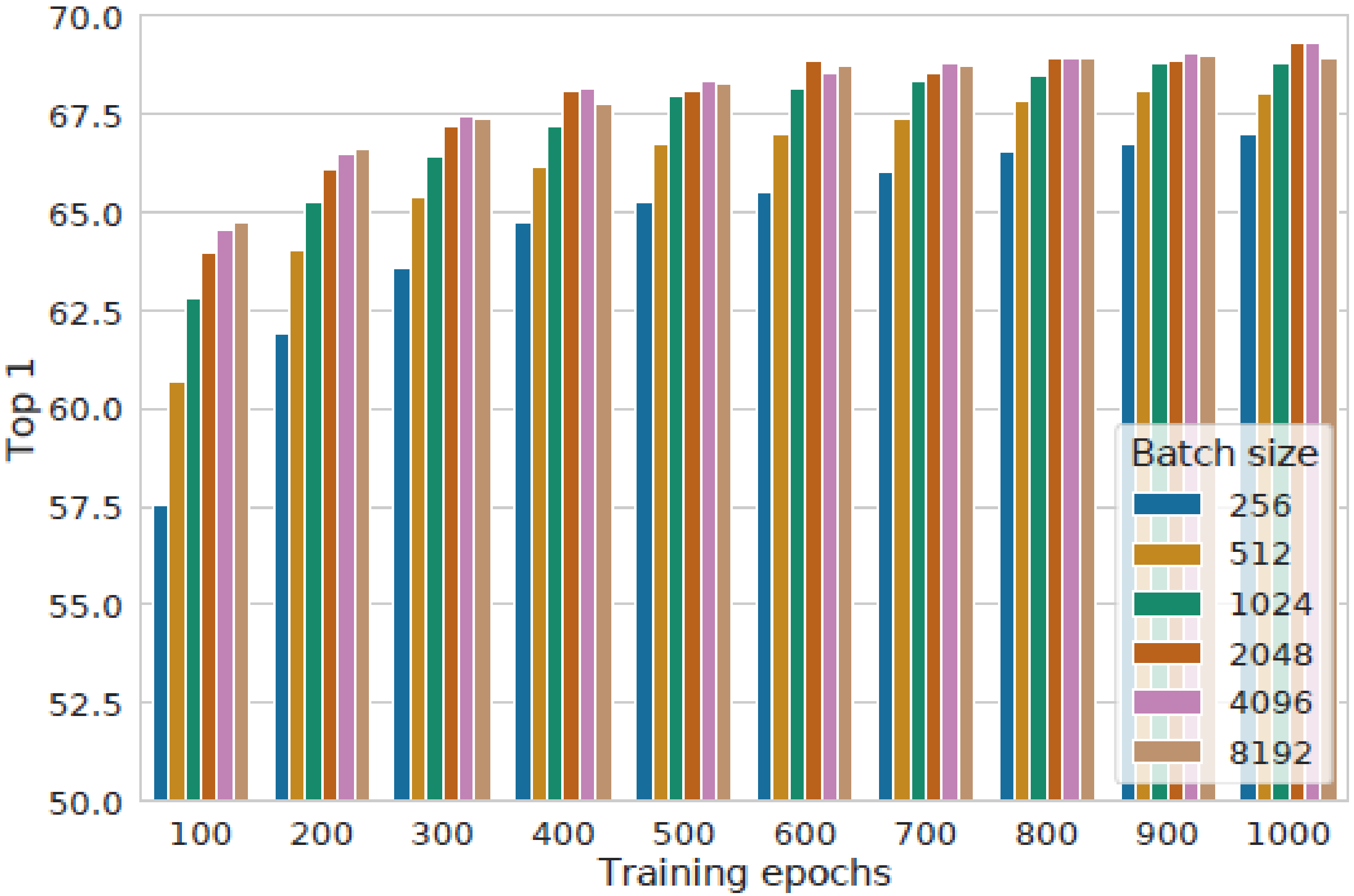
$$\ell_{i,j} = -\log \frac{\exp(\mathrm{sim}(\boldsymbol{z}_i, \boldsymbol{z}_j)/\tau)}{\sum_{k=1}^{2N} \mathbb{1}_{[k \neq i]} \exp(\mathrm{sim}(\boldsymbol{z}_i, \boldsymbol{z}_k)/\tau)}$$

$$\mathrm{sim}(\boldsymbol{u}, \boldsymbol{v}) = \boldsymbol{u}^\top \boldsymbol{v} / \|\boldsymbol{u}\| \|\boldsymbol{v}\|$$

Chen et al. A Simple Framework for Contrastive Learning of Visual Representations, ICML 2020.

SimCLR

- Large batch size



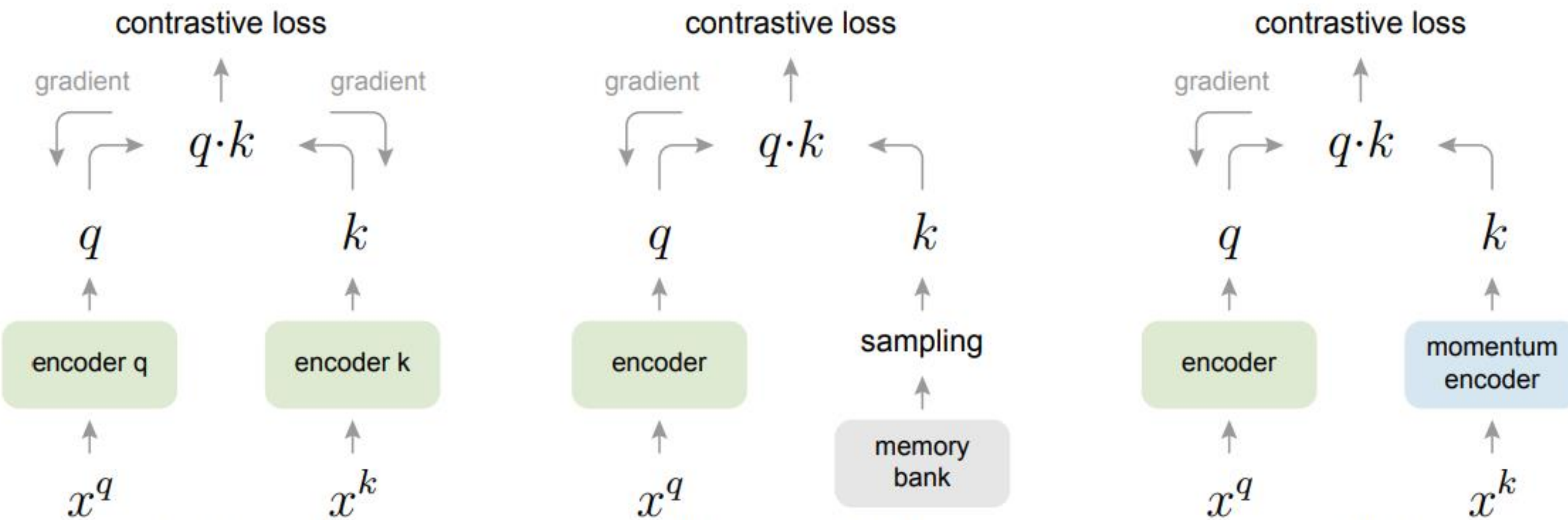
SimCLR

Method	Architecture	Param.	Top 1	Top 5
Methods using ResNet-50:				
Local Agg.	ResNet-50	24	60.2	-
MoCo	ResNet-50	24	60.6	-
PIRL	ResNet-50	24	63.6	-
CPC v2	ResNet-50	24	63.8	85.3
SimCLR (ours)	ResNet-50	24	**69.3**	**89.0**
Methods using other architectures:				
Rotation	RevNet-50 (4×)	86	55.4	-
BigBiGAN	RevNet-50 (4×)	86	61.3	81.9
AMDIM	Custom-ResNet	626	68.1	-
CMC	ResNet-50 (2×)	188	68.4	88.2
MoCo	ResNet-50 (4×)	375	68.6	-
CPC v2	ResNet-161 (∗)	305	71.5	90.1
SimCLR (ours)	ResNet-50 (2×)	94	74.2	92.0
SimCLR (ours)	ResNet-50 (4×)	375	**76.5**	**93.2**

Table 6. ImageNet accuracies of linear classifiers trained on representations learned with different self-supervised methods.

Momentum Contrast

- MOCO

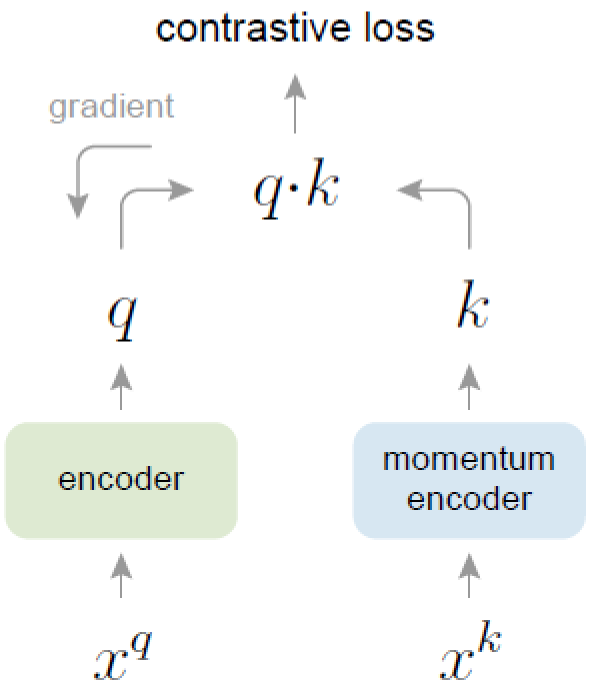


(a) end-to-end (b) memory bank (c) MoCo

He et al. Momentum Contrast for Unsupervised Visual Representation Learning, CVPR 2020.

Momentum Contrast

- MOCO



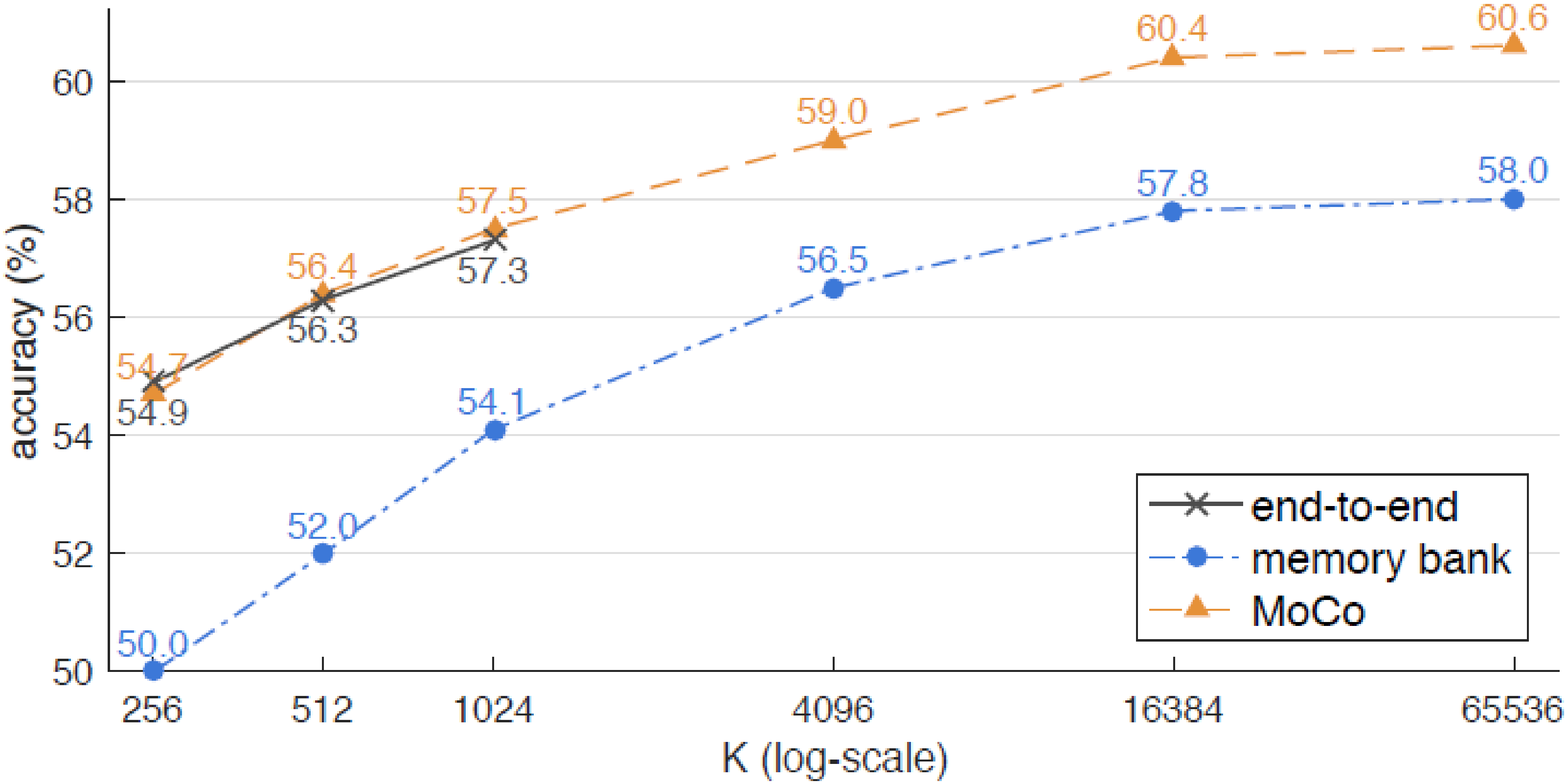
(c) MoCo

- Momentum update

$$\theta_{\mathrm{k}} \leftarrow m\theta_{\mathrm{k}} + (1-m)\theta_{\mathrm{q}}$$

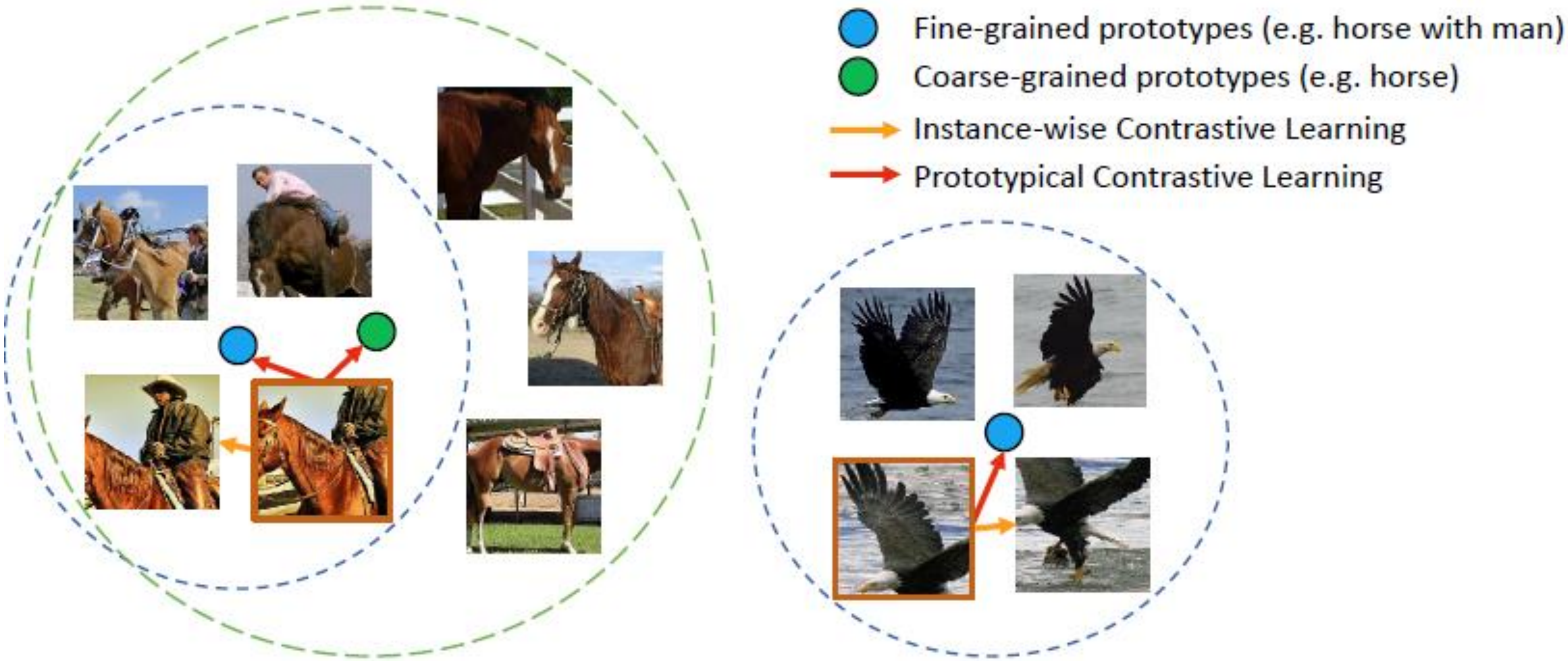
- Queue

Momentum Contrast



Prototypical Contrastive Learning

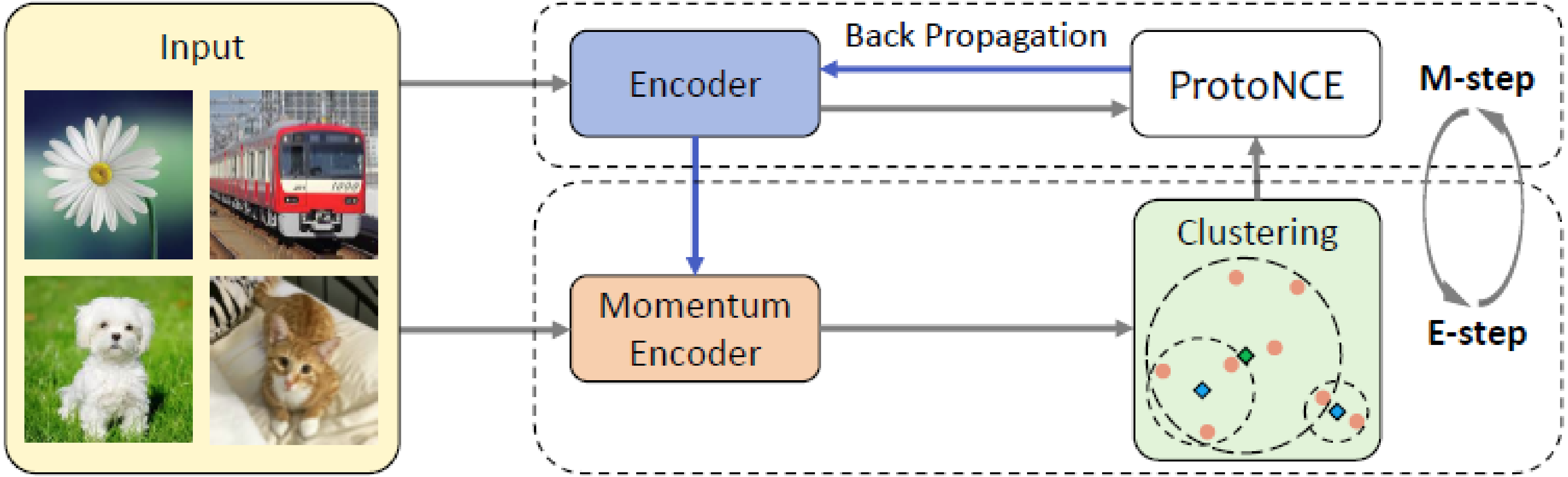
- Instance => Cluster



Li et al. Prototypical Contrastive Learning of Unsupervised Representations, ICLR 2021.

Prototypical Contrastive Learning

- Instance => Cluster



Prototypical Contrastive Learning

- Instance => Cluster

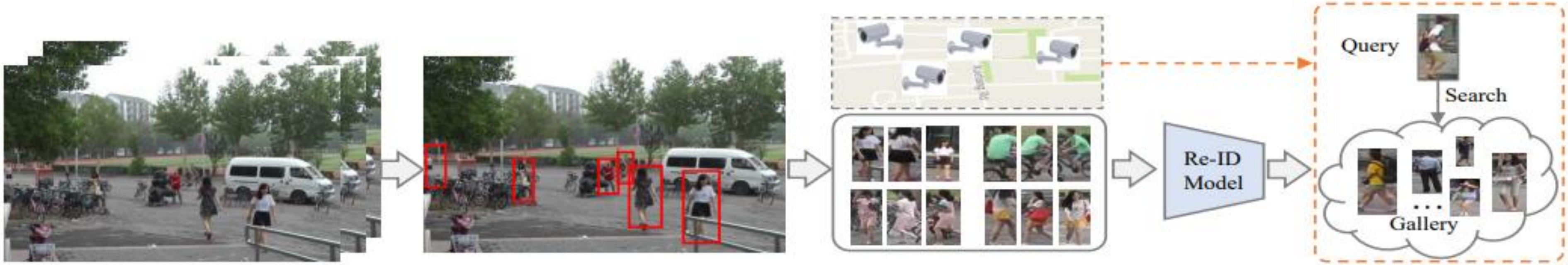
Method	architecture	VOC07					Places205				
		k=1	k=2	k=4	k=8	k=16	k=1	k=2	k=4	k=8	k=16
Random	ResNet-50	8.0	8.2	8.2	8.2	8.5	0.7	0.7	0.7	0.7	0.7
Supervised		54.3	67.8	73.9	79.6	82.3	14.9	21.0	26.9	32.1	36.0
Jigsaw	ResNet-50	26.5	31.1	40.0	46.7	51.8	4.6	6.4	9.4	12.9	17.4
MoCo		31.4	42.0	49.5	60.0	65.9	8.8	13.2	18.2	23.2	28.0
PCL (ours)		**46.9**	**56.4**	**62.8**	**70.2**	**74.3**	**11.3**	**15.7**	**19.5**	**24.1**	**28.4**
SimCLR	ResNet-50-MLP	32.7	43.1	52.5	61.0	67.1	9.4	14.2	19.3	23.7	28.3
MoCo v2		46.3	58.3	64.9	72.5	76.1	10.9	16.3	20.8	26.0	30.1
PCL v2 (ours)		**47.9**	**59.6**	**66.2**	**74.5**	**78.3**	**12.5**	**17.5**	**23.2**	**28.1**	**32.3**

Outline

- Attentions
  - Channel attention
  - Spatial attention
  - Self-attention
- Unsupervised Learning
  - Unsupervised feature representation learning
  - Unsupervised person re-identification

Person Re-ID

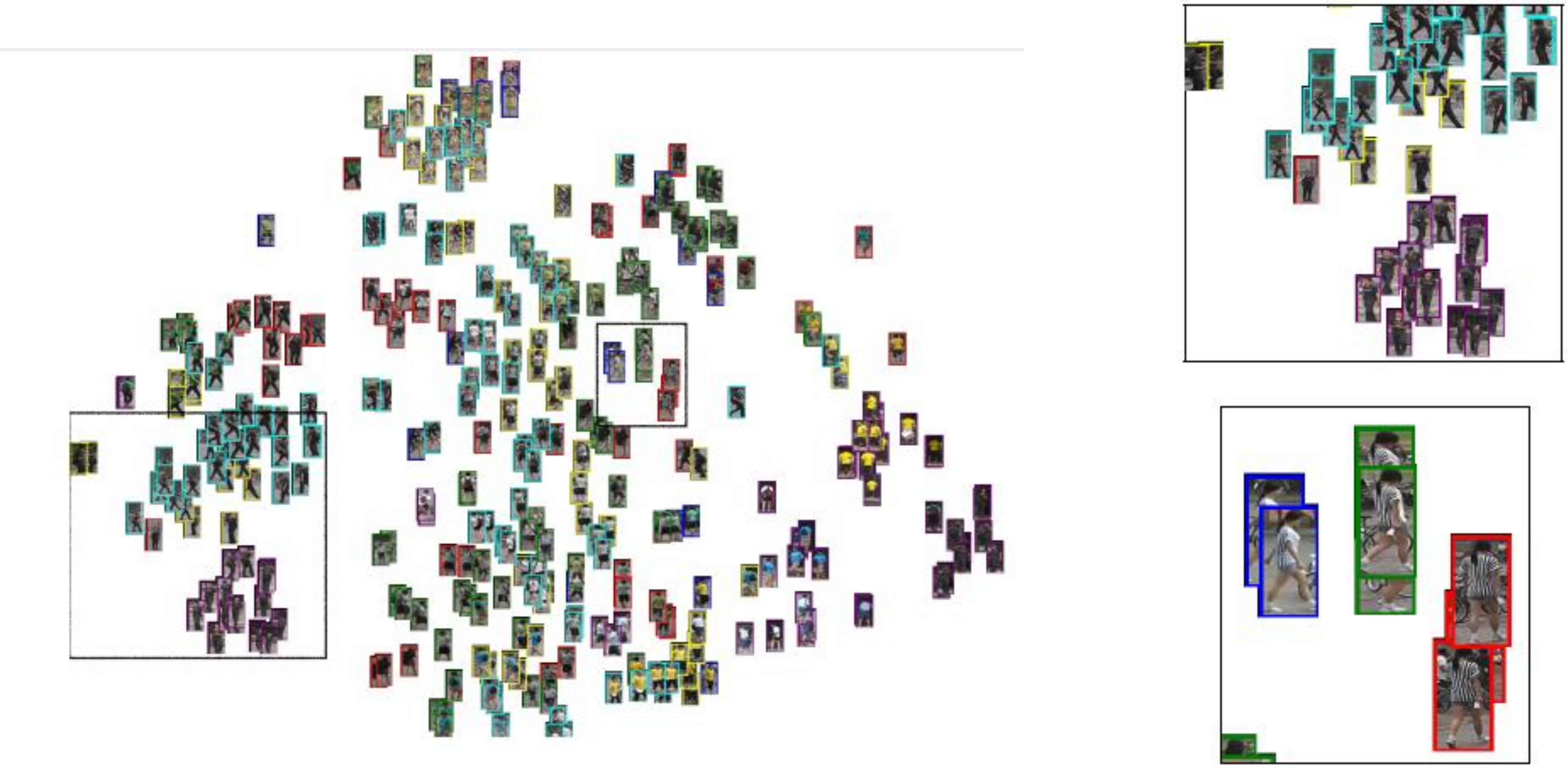
- The flow of a practical person Re-ID system



Ye, et al. Deep Learning for Person Re-identification, TPAMI 2021..

Unsupervised Person Re-ID

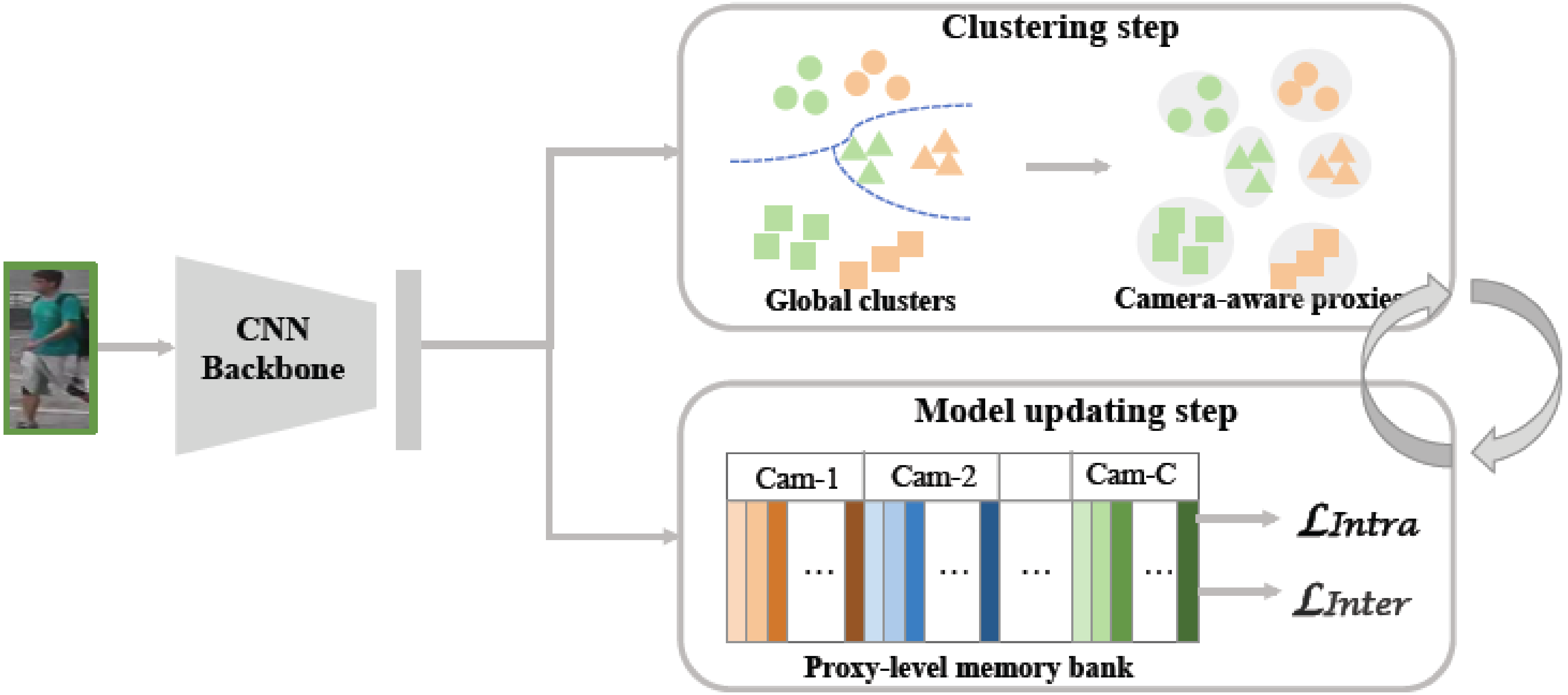
- Observations



Wang, et al. Camera-aware Proxies for Unsupervised Person Re-identification, AAAI 2021..

Unsupervised Person Re-ID

- Model



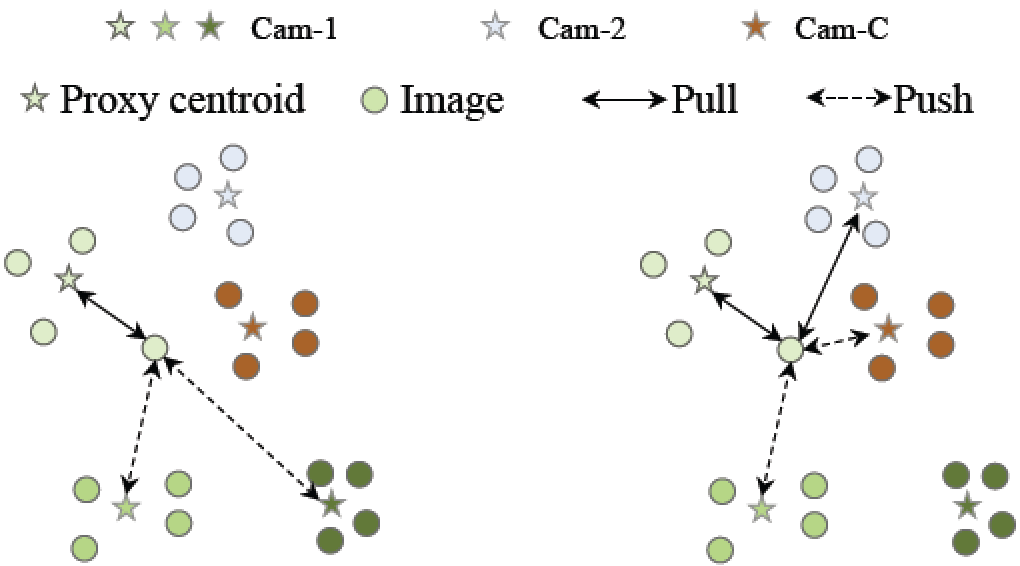
Unsupervised Person Re-ID

- Loss

$$\mathcal{L} = \mathcal{L}_{Intra} + \lambda \mathcal{L}_{Inter}$$

$$\mathcal{L}_{Intra} = -\sum_{c=1}^{C} \frac{1}{N_c} \sum_{x_i \in \mathcal{D}_c} \log \frac{exp(\mathcal{K}'[j]^T f(x_i)/\tau)}{\sum_{k=A+1}^{A+Z_{c_i}} exp(\mathcal{K}'[k]^T f(x_i)/\tau)}$$

$$\mathcal{L}_{Inter} = -\sum_{i=1}^{N'} \frac{1}{|\mathcal{P}|} \sum_{p \in \mathcal{P}} \log \frac{S(p, x_i)}{\sum\limits_{u \in \mathcal{P}} S(u, x_i) + \sum\limits_{q \in \mathcal{Q}} S(q, x_i)}$$



(a) Intra-camera Loss (b) Inter-camera Loss

Unsupervised Person Re-ID

Methods	Reference	Market-1501				DukeMTMC-ReID				MSMT17			
		R1	R5	R10	mAP	R1	R5	R10	mAP	R1	R5	R10	mAP
Purely Unsupervised													
BUC (Lin et al. 2019)	AAAI19	66.2	79.6	84.5	38.3	47.4	62.6	68.4	27.5	-	-	-	-
UGA (Wu et al. 2019)	ICCV19	87.2	-	-	70.3	75.0	-	-	53.3	49.5	-	-	21.7
SSL (Lin et al. 2020)	CVPR20	71.7	83.8	87.4	37.8	52.5	63.5	68.9	28.6	-	-	-	-
MMCL† (Wang and Zhang 2020)	CVPR20	80.3	89.4	92.3	45.5	65.2	75.9	80.0	40.2	35.4	44.8	49.8	11.2
HCT (Zeng et al. 2020)	CVPR20	80.0	91.6	95.2	56.4	69.6	83.4	87.4	50.7	-	-	-	-
CycAs (Wang et al. 2020b)	ECCV20	84.8	-	-	64.8	77.9	-	-	60.1	50.1	-	-	26.7
SpCL† (Ge et al. 2020)	NeurIPS20	88.1	95.1	97.0	73.1	-	-	-	-	42.3	55.6	61.2	19.1
CAP	This paper	91.4	96.3	97.7	79.2	81.1	89.3	91.8	67.3	67.4	78.0	81.4	36.9
Unsupervised Domain Adaptation													
PUL (Fan et al. 2018)	TOMM18	45.5	60.7	66.7	20.5	30.0	43.4	48.5	16.4	-	-	-	-
SPGAN (Deng et al. 2018b)	CVPR18	51.5	70.1	76.8	22.8	41.1	56.6	63.0	22.3	-	-	-	-
ECN (Zhong et al. 2019)	CVPR19	75.1	87.6	91.6	43.0	63.3	75.8	80.4	40.4	30.2	41.5	46.8	10.2
pMR (Wang et al. 2020a)	CVPR20	83.0	91.8	94.1	59.8	74.5	85.3	88.7	55.8	-	-	-	-
MMCL (Wang and Zhang 2020)	CVPR20	84.4	92.8	95.0	60.4	72.4	82.9	85.0	51.4	43.6	54.3	58.9	16.2
AD-Cluster (Zhai et al. 2020)	CVPR20	86.7	94.4	96.5	68.3	72.6	82.5	85.5	54.1	-	-	-	-
MMT (Ge, Chen, and Li 2020)	ICLR20	87.7	94.9	96.9	71.2	78.0	88.8	92.5	65.1	50.1	63.9	69.8	23.3
SpCL (Ge et al. 2020)	NeurIPS20	90.3	96.2	97.7	76.7	82.9	90.1	92.5	68.8	53.1	65.8	70.5	26.5
Fully Supervised													
PCB (Sun et al. 2018)	ECCV18	93.8	-	-	81.6	83.3	-	-	69.2	68.2	-	-	40.4
ABD-Net (Chen et al. 2019)	ICCV19	95.6	-	-	88.3	89.0	-	-	78.6	82.3	90.6	-	60.8
CAP's Upper Bound	This paper	93.3	97.5	98.4	85.1	87.7	93.7	95.4	76.0	77.1	87.4	90.8	53.7

Summary

- Attentions
- Unsupervised Learning