DER : Dynamically Expandable Representation for Class Incremental Learning

Paper : https://arxiv.org/pdf/2103.16788.pdf

- Young Jo Choi
- Department of Digital Analytics

Incremental learning

- Human can easily accumulate visual knowledge from past experiences and incrementally learn novel concepts. Inspired by this, the problem of class incremental learning aims to design algorithms that can learn novel concepts in a sequential manner and eventually perform well on all observed classes.
- = Model learning 이후 Class가 증가하는 상황에 대해 추가적인 학습을 하고자 함

Incremental learning

- 추가된 모든 데이터셋을 이용해 학습을 하면 학습시간과 계산비용 소모가 심하고 단순하게 전이학습을 하면 이전 class의 특징을 모델이 기억하지 못할 수 있음
- Stability-plasticity dilemma
- 이를 위해 다양한 incremental learning method가 나왔고 DER은 그중 가장 월등한 score를 보여준다.

DER

DER: Dynamically Expandable Representation for Class Incremental Learning

Shipeng Yan[1,3,4*] Jiangwei Xie[1*] Xuming He[1,2]
[1]School of Information Science and Technology, ShanghaiTech University
[2]Shanghai Engineering Research Center of Intelligent Vision and Imaging
[3]Shanghai Institute of Microsystem and Information Technology, Chinese Academy of Sciences
[4]University of Chinese Academy of Sciences
{yanshp, xiejw, hexm}@shanghaitech.edu.cn

DER

Incremental Learning

212 papers with code • 17 benchmarks • 8 datasets

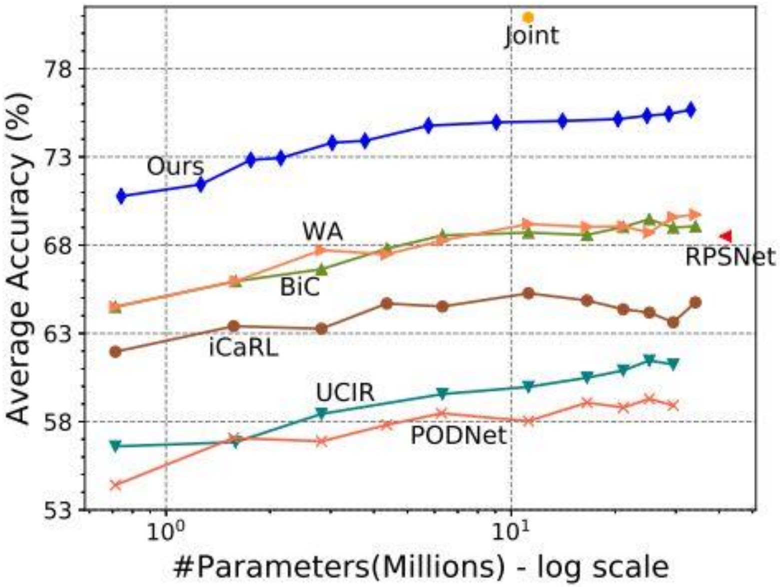
Incremental learning aims to develop artificially intelligent systems that can continuously learn to address new tasks from new data while preserving knowledge learned from previously learned tasks.

Benchmarks

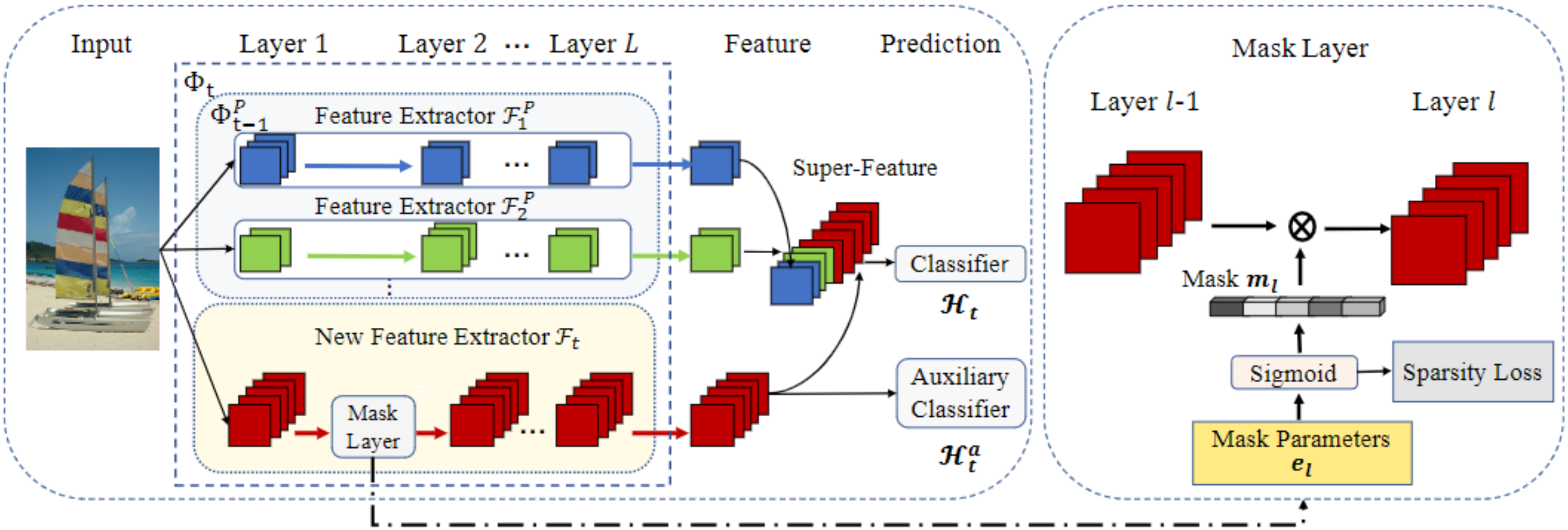
Add a Result

These leaderboards are used to track progress in Incremental Learning

Trend	Dataset	Best Model	Paper	Code	Compare
	CIFAR-100 - 50 classes + 10 steps of 5 classes	DER(Standard ResNet-18)			See all
	CIFAR-100 - 50 classes + 5 steps of 10 classes	DER(Standard ResNet-18)			See all
	ImageNet100 - 10 steps	RMM (ResNet-18)			See all
	ImageNet - 10 steps	DyTox			See all
	ImageNet-100 - 50 classes + 10 steps of 5 classes	RMM (ResNet-18)			See all



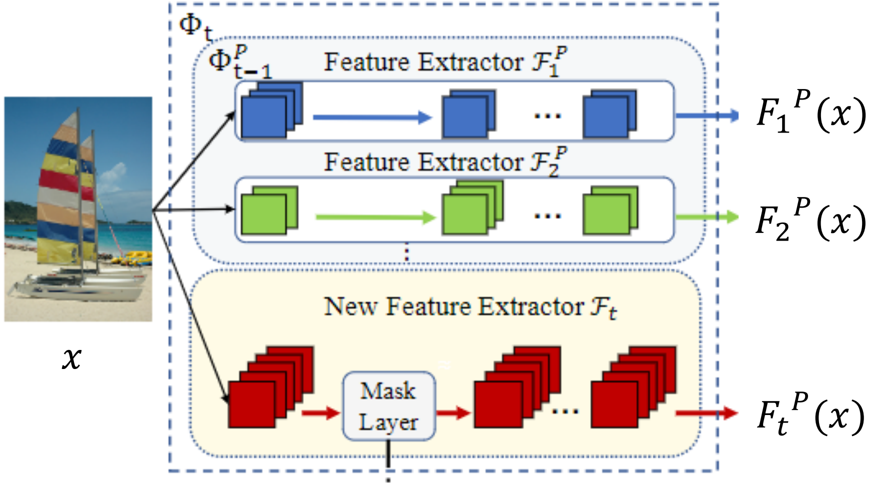
DER structure



- 1) Representation Learning Stage
- 2) Classifier Learning Stage
- 3) Channel Level Mask

DER structure

- 1) Representation Learning Stage



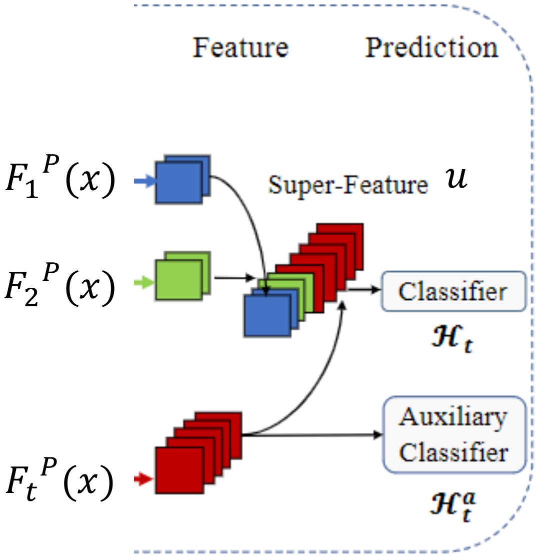
$$\boldsymbol{u} = \Phi_t(\boldsymbol{x}) = [\Phi_{t-1}(\boldsymbol{x}), \mathcal{F}_t(\boldsymbol{x})]$$

x : $image \in \widetilde{D}_t$, $\widetilde{D}_t = D_t \cup M_t$
D_t : $training\ data\ at\ step\ t$
M_t : $rehersal\ memory\ from\ D_{t-1}$
t : $step$
u : $Super\ Feature$
F_t : $Feature\ Extractor\ at\ step\ t$
Φ_t : $Super\ Feature\ Extractor$

When training F_t ,
we freeze the learned function Φ_{t-1} to reduce catastrophic forgetting and F_t are encouraged to learn only novel aspect of new classes.

DER structure

- 2) Classifier Learning Stage



Authors
~~We~~ propose an auxiliary loss to promote the newly added feature module to learn novel classes effectively. (distinguishing new and old classes)

$\tilde{y}_t = \cup_{i=1}^{t} y_i$
y_i : *label set at step t*

$\mathcal{H}_t(\boldsymbol{u})$: *Classifier at step t*

$$p_{\mathcal{H}_t}(\boldsymbol{y}|\boldsymbol{x}) = \mathrm{Softmax}(\mathcal{H}_t(\boldsymbol{u}))$$

$$\hat{y} = \arg\max p_{\mathcal{H}_t}(\boldsymbol{y}|\boldsymbol{x}),\ \hat{y} \in \tilde{\mathcal{Y}}_t.$$

➔ Loss of Classifier

$$\mathcal{L}_{\mathcal{H}_t} = -\frac{1}{|\tilde{\mathcal{D}}_t|}\sum_{i=1}^{|\tilde{\mathcal{D}}_t|}\log(p_{\mathcal{H}_t}(y = y_i|\boldsymbol{x_i})))$$

$\mathcal{H}_t^a(\mathcal{F}_t(\boldsymbol{x}))$: *Auxiliary Classifier at step t*

$$p_{\mathcal{H}_t^a}(\boldsymbol{y}|\boldsymbol{x}) = \mathrm{Softmax}(\mathcal{H}_t^a(\mathcal{F}_t(\boldsymbol{x})))$$

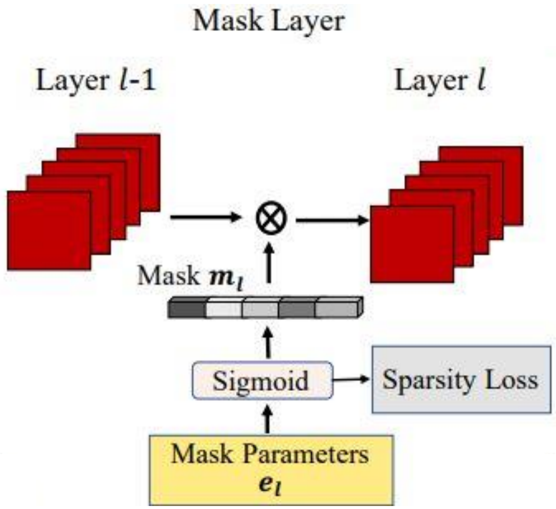
➔ Loss of Auxiliary Classifier : $\mathcal{L}_{\mathcal{H}_t^a}$

➔ Loss of expandable representation

$$\mathcal{L}_{\mathrm{ER}} = \mathcal{L}_{\mathcal{H}_t} + \lambda_a \mathcal{L}_{\mathcal{H}_t^a}$$

DER structure

- 3) Channel Level Mask



to remove the model redundancy and learn the compact features for novel classes, we apply a differentiable channel-level mask-based pruning method that dynamically prunes the network according to the difficulty of novel concepts

$l \in$ convolution layer (feature extractor)

$$\boldsymbol{f}_l' = \boldsymbol{f}_l \odot \boldsymbol{m}_l$$

f_l : representation of image

$m_l^i \in [0,1]$

$$\boldsymbol{m}_l = \sigma(s\boldsymbol{e}_l)$$

e_l : learnable mask parameters

s : scaling factor to control the sharpness of the function

→ Channel level mask by sigmoid

$$s = \frac{1}{s_{\max}} + \left(s_{\max} - \frac{1}{s_{\max}}\right)\frac{b-1}{B-1} \qquad \boldsymbol{g}_{\boldsymbol{e}_l}' = \frac{\sigma(\boldsymbol{e}_l)[1-\sigma(\boldsymbol{e}_l)]}{s\sigma(s\boldsymbol{e}_l)[1-\sigma(s\boldsymbol{e}_l)]}\boldsymbol{g}_{\boldsymbol{e}_l}$$

based on the ratio of used weight in all available weight

$$\mathcal{L}_S = \frac{\sum_{l=1}^{L} K_l \|\boldsymbol{m}_{l-1}\|_1 \|\boldsymbol{m}_l\|_1}{\sum_{l=1}^{L} K_l c_{l-1} c_l}$$

L : the number of layers

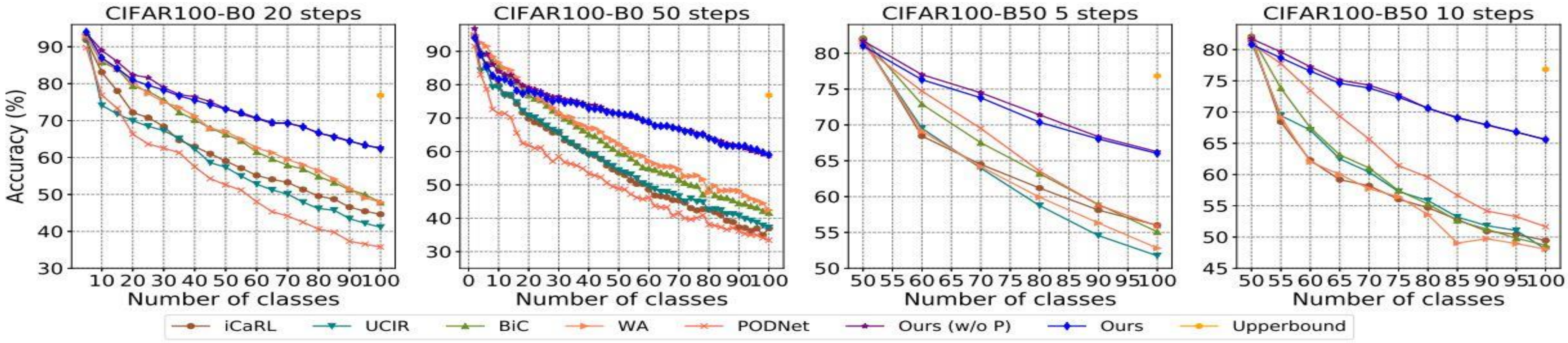
K_l : the kernel size of convolution layer l

→ encourage the model to maximally reduce the number of parameters

$$\mathcal{L}_{\mathrm{DER}} = \mathcal{L}_{\mathcal{H}_t} + \lambda_a \mathcal{L}_{\mathcal{H}_t^a} + \lambda_s \mathcal{L}_S$$

DER result

- Data set : CIFAR100
- Implementation
  - CIFAR100-B0 : trains all 100 classes in several splits including 5,10,20,50 incremental steps
  - CIFAR100-B50 : starts from a model trained on 50 classes, and the remaining 50 classes are divided into split 2,5, and 10 steps with 20 examples as memory per class.



DER result

CIFAR100-B0)

Methods	5 steps		10 steps		20 steps		50 steps	
	#Paras	**Avg**	**#Paras**	**Avg**	**#Paras**	**Avg**	**#Paras**	**Avg**
Bound	11.2	80.40	11.2	80.41	11.2	81.49	11.2	81.74
iCaRL[27]	11.2	71.14$_{\pm0.34}$	11.2	65.27$_{\pm1.02}$	11.2	61.20$_{\pm0.83}$	11.2	56.08$_{\pm0.83}$
UCIR[12]	11.2	62.77$_{\pm0.82}$	11.2	58.66$_{\pm0.71}$	11.2	58.17$_{\pm0.30}$	11.2	56.86$_{\pm3.74}$
BiC[12]	11.2	73.10$_{\pm0.55}$	11.2	68.80$_{\pm1.20}$	11.2	66.48$_{\pm0.32}$	11.2	62.09$_{\pm0.85}$
WA[39]	11.2	72.81$_{\pm0.28}$	11.2	69.46$_{\pm0.29}$	11.2	67.33$_{\pm0.15}$	11.2	64.32$_{\pm0.28}$
PODNet[6]	11.2	66.70$_{\pm0.64}$	11.2	58.03$_{\pm1.27}$	11.2	53.97$_{\pm0.85}$	11.2	51.19$_{\pm1.02}$
RPSNet[26]	60.6	70.5	56.5	68.6	-	-	-	-
Ours(w/o P)	33.6	**76.80**$_{\pm0.79}$(+3.7)	61.6	**75.36**$_{\pm0.36}$(+5.9)	117.6	**74.09**$_{\pm0.33}$(+6.76)	285.6	**72.41**$_{\pm0.36}$(+8.09)
Ours	**2.89**	**75.55**$_{\pm0.65}$(+2.45)	**4.96**	**74.64**$_{\pm0.28}$(+5.18)	**7.21**	**73.98**$_{\pm0.36}$(+6.65)	**10.15**	**72.05**$_{\pm0.55}$(+7.73)

ImageNet)

Methods	ImageNet100-B0					ImageNet1000-B0				
	#Paras	top-1		top-5		#Paras	top-1		top-5	
		Avg	Last	Avg	Last		Avg	Last	Avg	Last
Bound	11.2	-	-	-	95.1	11.2	89.27	-	-	-
iCaRL[27]	11.2	-	-	83.6	63.8	11.2	38.4	22.7	63.7	44.0
BiC[12]	11.2	-	-	90.6	84.4	11.2	-	-	84.0	73.2
WA[39]	11.2	-	-	91.0	84.1	11.2	65.67	55.6	86.6	81.1
RPSNet[26]	-	-	-	87.9	74.0	-	-	-	-	-
Ours(w/o P)	61.6	**77.18**	**66.70**	**93.23**	**87.52**	61.6	**68.84**	**60.16**	**88.17**	**82.86**
Ours	**7.67**	**76.12**	**66.06**	**92.79**	**88.38**	14.52	**66.73**	**58.62**	**87.08**	**81.89**

Methods	ImageNet100-B50				
	#Paras	top-1		top-5	
		Avg	Last	Avg	Last
Bound	11.2	81.20	81.5	-	-
UCIR[12]	11.2	68.09	57.3	-	-
PODNet[6]	11.2	74.33	-	-	-
TPCIL[34]	11.2	74.81	66.91	-	-
Ours(w/o P)	67.20	**78.20**	**74.92**	**94.20**	**91.30**
Ours	**8.87**	**77.73**	**72.06**	**94.01**	**91.64**

Ablation Study

- The effect of each component

Components E.R.	Components Aux.	Avg	Last
✗	✗	61.84	40.81
✓	✗	73.26	63.07
✓	✓	75.36	65.34

Table 4: The contribution of each component. *E.R.* means expandable representation. *Aux.* means using auxiliary loss.

Ablation Study

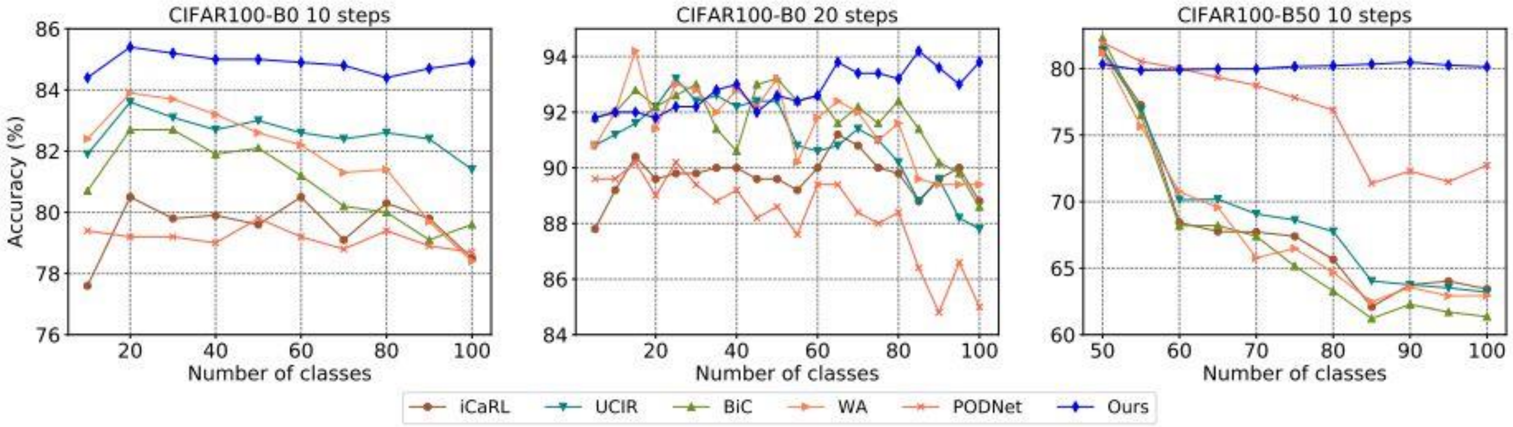


Figure 4: **Analysis**. The backward transfer of representation by observing the changes of $A^t_{\mathcal{Y}_1}$ for different splits.

$$\mathrm{BWT} = \frac{1}{T-1}\sum_{i=2}^{T}\frac{1}{i}\sum_{j=1}^{i} A^i_{\mathcal{Y}_j} - A^j_{\mathcal{Y}_j}$$

$$\mathrm{FWT} = \frac{1}{T-1}\sum_{i=2}^{T} A^i_{\mathcal{Y}_i} - \bar{A}^i_{\mathcal{Y}_i}$$

$A^t_{y_k}$: acc at step T on the test images of class set y_k

$\bar{A}^i_{y_i}$: test acc obtained by model trained on available data $\widetilde{D}_T$ with only cross-entropy loss at random initialization

Methods	iCaRL	UCIR	BiC	WA	PODNet	Ours
BWT (%)	−4.14	−8.52	−3.40	−3.18	−16.27	**+1.36**
FWT (%)	−4.91	−5.56	−0.17	+0.82	−5.58	**+1.49**

Table 5: Backward transfer and Forward transfer (FWT) for representation.

Conclusion

- 각 단계에서 이전에 학습된 representation을 freeze하고 새 task data에 대해서만 학습시킴
- 새 data의 novel한 특징을 더 잘 학습할 수 있도록 auxiliary loss를 추가
- We also introduce channel-level mask-based pruning to dynamically expand representation according to the difficulty of novel concepts.
- 다른 incremental method들에 비해 지속적으로 더 잘 수행된다는 것을 보였을 뿐 아니라, positive한 backward transfer 및 forward transfer을 달성한다는 것을 보였음