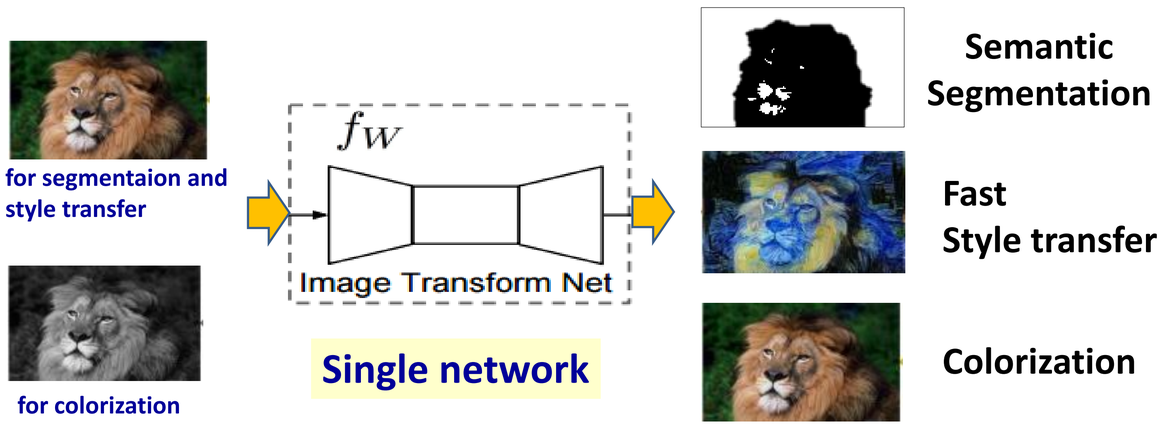
CL-2018

Continual Learning for an Encoder-Decoder CNN Using "Piggyback"

Asato Matsumoto Keiji Yanai
The University of Electro-Communications, Tokyo, Japan

Objective

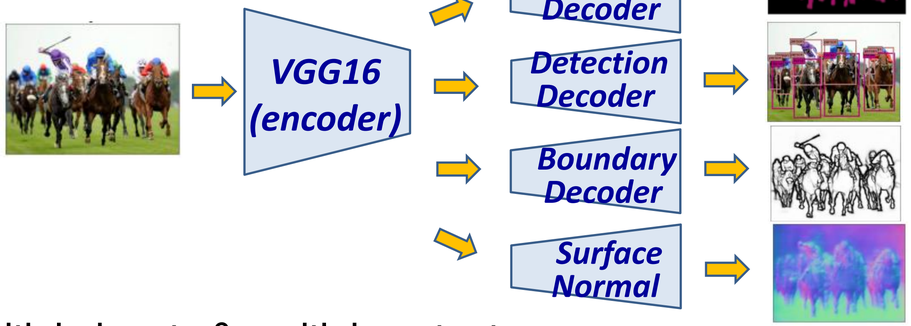
Continual learning of an Encoder-Decoder network for heterogeneous image-to-image tasks



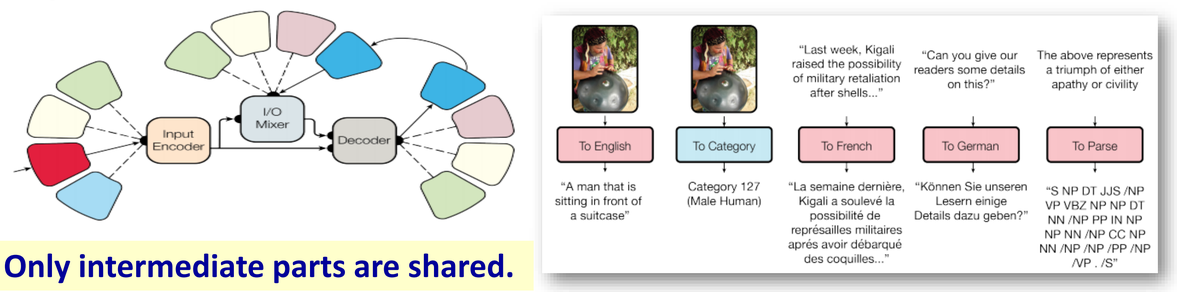
Related work

- **Simultaneous training of multiple tasks**

(1) Single encoder & task-specific decoders
e.g. UberNet [Ikkinos CVPR2017]



(2) Multiple inputs & multiple outputs
e.g. "**One Model To Learn Them All**" [Kaiser et al. arXiv 2017]



Only intermediate parts are shared.

- **Continual learning : major approarches for overcoming "catastrophic forgetting"**

(1) **Rehearsal** [Hetherington et al. 1989]
- Keep (sampled) old training samples and use them for new training as well

(2) **Distillation** (Learning without Forgetting (LfW)) [Li and Hoiem 2016]
- Reproduce training labels of old tasks with trained model and use them for new training

(3) **Elastic Weight Consolidation** (EWC) [Kirkpatrick et al. 2016] **(regularization)**
- Train weights for new tasks according to un-importantness of weights

(4) **Progressive Neural Networks** [Ruse et al. 2016]
- Fix trained weights for the previous tasks, and extend the net and train extended weights.
- "PackNet" [Mallya et al. CVPR2018] : a pruning version of progressive network

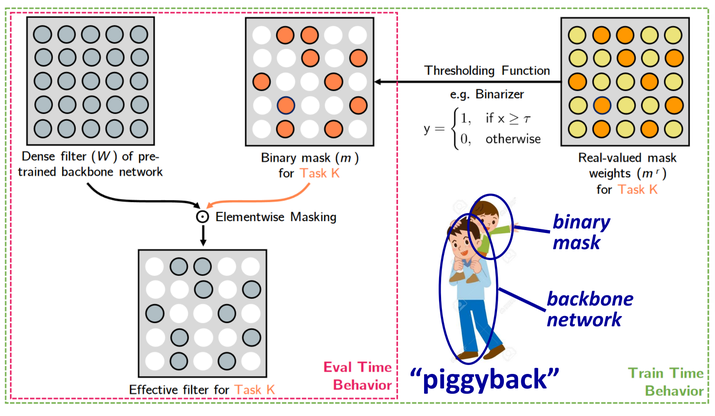
(5) **Weight selection: "Piggyback"** [Mallya et al. ECCV2018]
- Select the weights of the fixed base network with task-specific binary masks

(4) (5) bring "no catastrophic forgetting" with small additional task-specific weights .

In this work, we adopt "Piggyback" (5) as a basic approach.

Weight selection approach: "Piggyback"

(1) For the first task, train a network as a "backbone network."
(2) For the second task or more, fine-tune the network, and obtain the task-specific binary mask by thresholding. [Mallya et al. ECCV2018]
(3) At evaluation time, calculate the task-specific weights by elementwise masking.



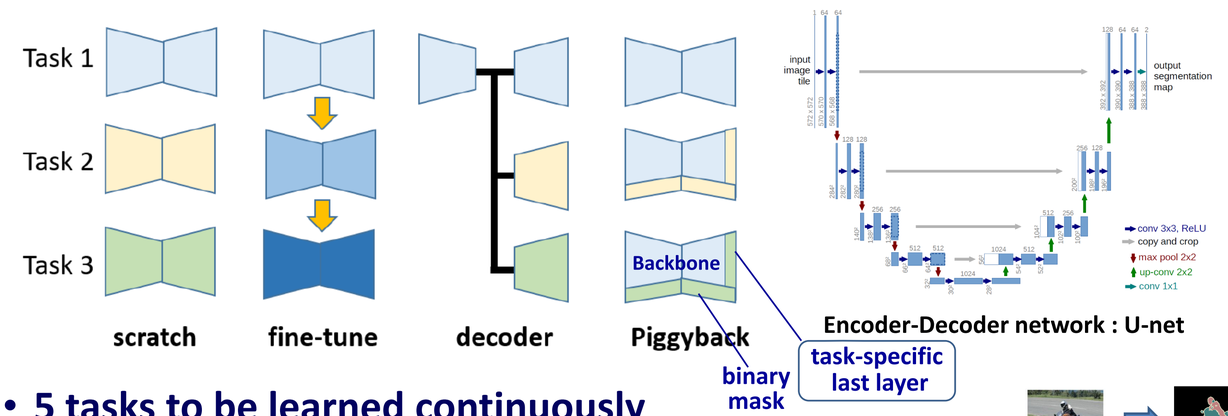
Dataset	Classifier Only	PackNet [7] ↓	PackNet [7] ↑	Piggyback (ours)	Individual Networks
ImageNet	28.42 (9.61)	29.33 (9.99)		28.42 (9.61)	28.42 (9.61)
CUBS	36.49	22.30	29.69	20.99	21.30
Stanford Cars	54.66	15.81	21.66	11.87	12.49
Flowers	20.01	10.33	10.25	7.19	7.35
WikiArt	49.53	32.80	31.48	29.91	29.84
Sketch	58.53	28.62	24.88	22.70	23.54
# Models (Size)	1 (537 MB)	1 (587 MB)		1 (621 MB)	6 (3,222 MB)

Table 2: Errors obtained by starting from an ImageNet-trained VGG-16 network and then using various methods to learn new fine-grained classification

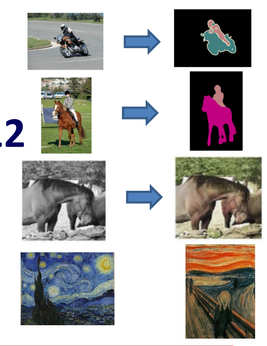
No "catastrophic forgetting" happen with small additional binary masks and task-specific final layers.

Experiments

- **Apply "Piggyback" to 3 kinds of image-to-image tasks.**
  - Baseline 1: independent models (scratch)
  - Baseline 2: copying models and incremental fine-tuning (fine-tune)
  - Baseline 3: shared decoder and task-specific decoder (decoder)



- **5 tasks to be learned continuously**
  - Task1 : Semantic segmentation with MSCOCO
  - Task2 : Semantic segmentation with PASCAL VOC 2012
  - Task3 : Colorization for gray-scaled MSCOCO images
  - Task4 : Fast Style Transfer with "Gogh's Starry Night"
  - Task4': Fast Style Transfer with "Munk's Scream"

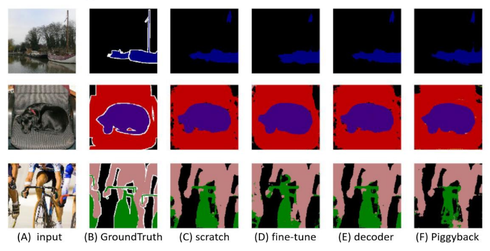


- **Experimental Results** ⇨ **"Piggyback" worked well for Encoder-Decoder net !**

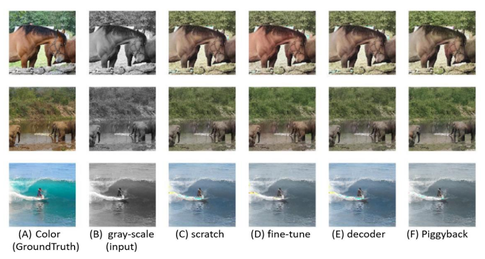
	scratch	fine-tune	decoder	piggyback
Task1 (mIoU(%))	21.47			
Task2 (mIoU(%))	58.59	64.87	61.63	61.45
Task3 (MSE, SSIM)	244.00 0.9138	237.92 0.9148	241.66 0.9121	242.49 0.9058
Task4 (SSIM, total loss(epoch))	0.3678 413833 (200)	0.3555 405893 (200)	0.3595 473723 (200)	0.3501 528587 (100)
Task4' (total loss(epoch))	447480 (6)	490490 (6)	544348 (6)	521476 (6)
Model Size (MB)	282.0 (56.4 × 5)	282.0 (56.4 × 5)	138.4 (56.4 + 20.5 × 4)	63.6 (56.4 + 1.8 × 4)
Task1 after Task2	-	0.70	21.47	21.47
Task2 after Task3	-	1.87	61.63	61.45
Task3 after Task4 (MSE, SSIM)	-	870.18 0.5321	241.66 0.9121	242.49 0.9058

Catastrophic forgetting / No catastrophic forgetting / No catastrophic forgetting

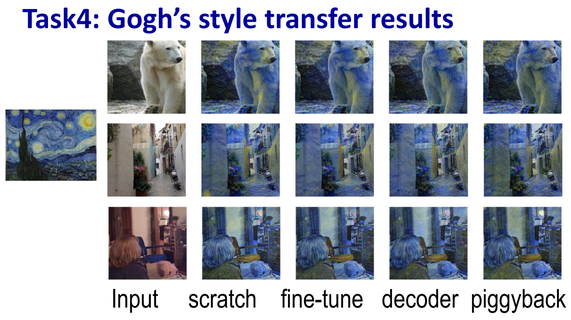
Task2: segmentation results



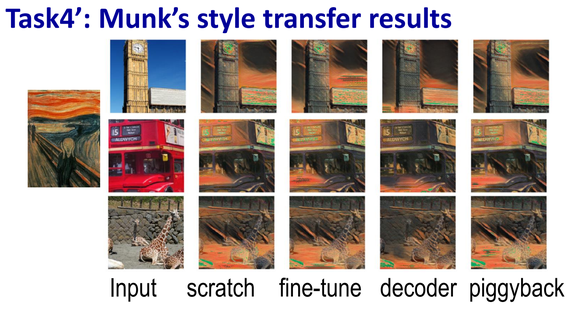
Task3: colorization results



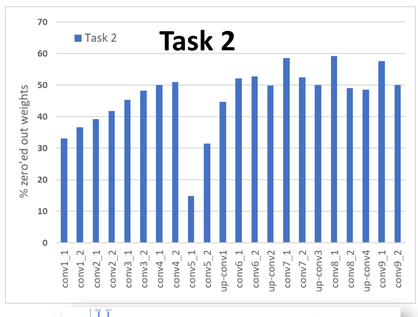
Task4: Gogh's style transfer results

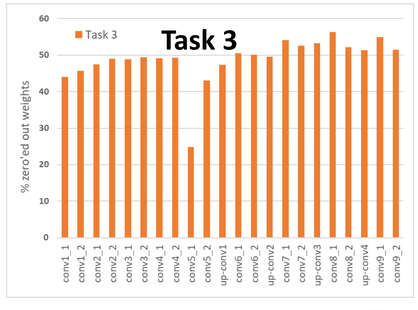


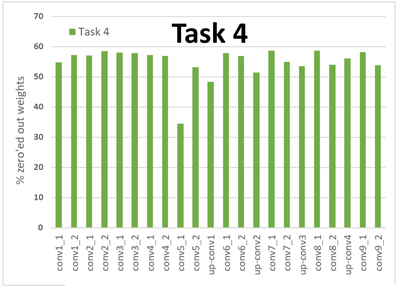
Task4': Munk's style transfer results

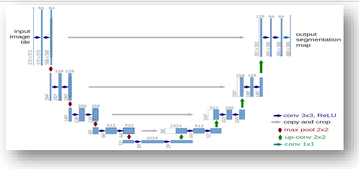


- **The ratio of Zero'ed out weights**

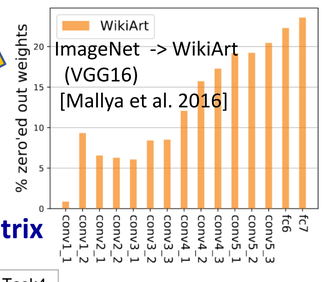








The tendency is totally different from VGG16 with classification tasks.



Binary mask similarity matrix
(ratio of the same values)

	Task1	Task2	Task3	Task4
Task2	0.5075	-	-	-
Task3	0.5042	0.5054	-	-
Task4	0.4326	0.5034	0.5020	-
Task4'	0.4529	0.5029	0.5025	0.5210

- **Feature works**
  - Reducing the size of binary masks
  - Combining pruning
  - Analyzing the trained masks → binary masks are "task features" ??