Supplementary Material

Class-incremental Continual Learning for Instance Segmentation with Image-level Weak Supervision

Yu-Hsing Hsieh[1] Guan-Sheng Chen[1] Shun-Xian Cai[1] Ting-Yun Wei[1]
Huei-Fang Yang[2] Chu-Song Chen[1*]
[1]National Taiwan University, Taiwan [2]National Sun Yat-sen University, Taiwan
{r10922024,r10922052,r11922081,r10922010,chusong}@csie.ntu.edu.tw hfyang@mis.nsysu.edu.tw

As mentioned in the main paper, our training procedure consists of two phases, CL4WSSS and CL4WSIS. More of their details are illustrated in Figs. 1(a) and 1(b), respectively. In Phase 1, we train the semantic branch of the t-th step Segmenter. The learned CL4WSSS network then serves for predicting the semantic segmentation results and producing further the synthetic center & offset maps in Phase 2 to train the instance branch of the Segmenter for CL4WSIS. As demonstrated in the results of the main paper, our method outperforms the method in [2] on CL4WSSS by introducing FLAC, random dropout and peak generator techniques. We develop a selective distillation mechanism that leverages CL4WSSS results to learn the step-t Segmenter from the $(t-1)$-th and obtain a better CL4WSIS solution.

More details and results are given in the following.

1. Additional Implementation Details

In this section, we complement additional details on Peak Generator (PG) & Multi-label Classification Loss.

1.1. Peak generator & multi-label classification loss

As mentioned in the main paper, PAM introduced in [2] does not ensure that the peaks (or hypothesized object center locations) are inside the high-score semantic maps. It is because PAM is trained separately from the WSSS model, and so it does not take into account the semantic knowledge.

To strengthen the relationship between peaks and WSSS, we append the peak generator (PG) after the Decoder. Specifically, PG takes as input the Decoder's output Z^t and produces $Z^{\text{pg}} \in \mathbf{R}^{|\mathcal{Y}^t|\times H\times W}$. We then highlight the core regions and suppress the noisy regions in Z^{pg} as follows. Pixels on channel c are treated as core pixels if their values are greater than the channel-specific threshold τ_c. τ, the threshold vector for all channels, is computed by pixel-wise multiplying a hyper-paramenter γ with $G \in \mathbb{R}^{|\mathcal{Y}^t|\times 1\times 1}$, where G is the global max pooling of Z^{pg} and γ is set to 0.7 in our implementation. Furthermore, unlike PAM that uses global average pooling (GAP), we use the nGWP [1] to aggregate the Z^{pg} because nGWP can highlight the contribution of more correlated pixels to a class. The aggregated score $\hat{y}^{\text{pg}}$ is obtained by:

$$\hat{y}_c^{\text{pg}} = \frac{\sum_{i,j} M_{c,i,j}^{\text{pg}} Z_{c,i,j}^{\text{pg}}}{\epsilon + \sum_{i',j'} M_{c,i',j'}^{\text{pg}}}, \tag{1}$$

where $M^{\text{pg}} = \texttt{softmax}(Z^{\text{pg}})$ and ϵ is a small constant. The aggregated $\hat{y}^{\text{pg}}$ is then trained with Global Image Labels via the multi-label classification loss:

$$\ell_{cls}^{pg}(\hat{y}^{\text{pg}}, y) = -\frac{1}{|\mathcal{Y}^t|}\sum_{c\in\mathcal{Y}^t}(y_c \log(\hat{y}_c^{\text{pg}}) + (1-y_c)\log(1-\hat{y}_c^{\text{pg}})). \tag{2}$$

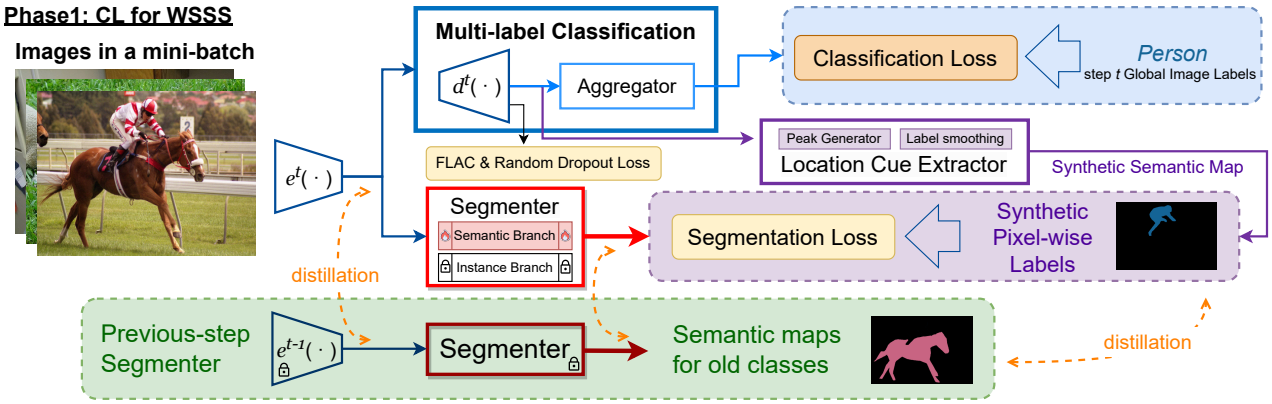
2. Additional Training Details

As illustrated in Fig. 1(a) for Phase 1, the Multi-label Classification module contains a Decoder followed by an Aggregator ($\mathbb{A}$) and is optimized with the binary cross entropy (BCE) using Global Image Labels. This module is mainly for estimating each pixel's "contribution score" to each class by decomposing its intermediate output Z^t, which is then refined by the Location Cue Extractor through label smoothing [3] to provide the synthetic pixel-wise labels to train our Segmenter. The label smoothing is done by first generating a one-hot distribution for each pixel, where the class of the highest score for a pixel is set to one, denoted as $Z^{hard,t}$. The synthetic semantic map is then obtained by

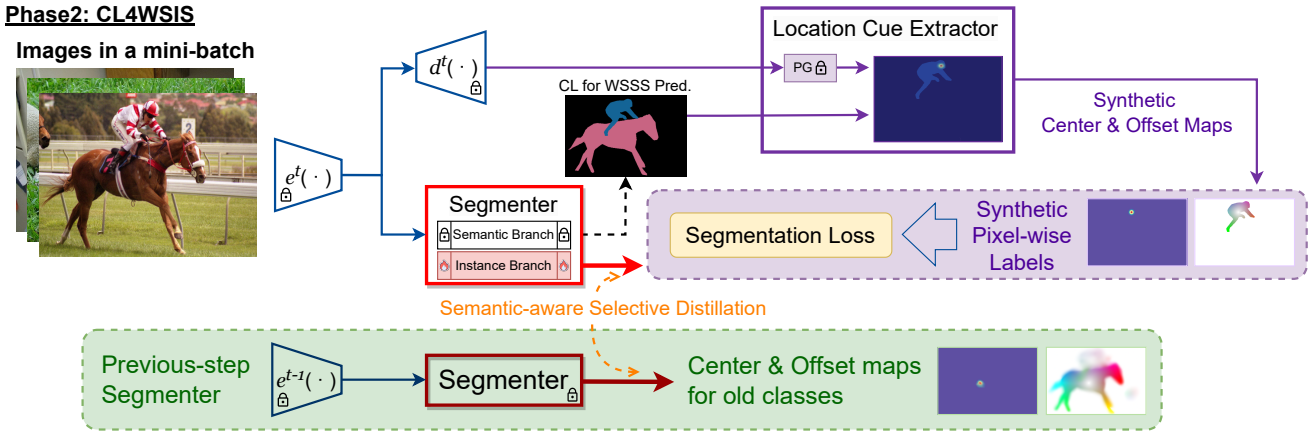
$$S^{syn} = \beta Z^{hard,t} + (1-\beta)Z^t \tag{3}$$

with β a hyper-parameter that controls the smoothness. Feature-level augmentation consistency and Random Dropout are employed to improve the Decoder's performance. Peak Generator (PG) is appended after the Decoder to strengthen the activation of Decoder's outputted semantic map and provide instance cues for Phase 2. The old knowledge is maintained by distilling the semantic maps yielded by the Previous-step Segmenter to both the Decoder and the current Segmenter, and also through feature distillation between the outputs of e^t and e^{t-1}. Note that the instance

*corresponding author.



(a) Phase1: CL for WSSS. Our Decoder $d^t(\cdot)$ followed by an Aggregator is optimized with Global Image Labels. This module's output semantic scores are then refined by the Location Cue Extractor through label smoothing to provide the synthetic pixel-wise labels to train our Segmenter. Feature-level augmentation consistency and Random Dropout are employed to improve the Decoder's performance. Peak Generator (PG) is appended after the Decoder to strengthen the activation of Decoder's output semantic map and provide instance cues for Phase 2. The old knowledge is maintained by distilling the semantic maps yielded by the Previous-step Segmenter and also through feature distillation. The instance branch for generating the center and offset maps is frozen in this phase.



(b) Phase2: CL4WSIS. Only the instance branch is optimized in this phase. Our approach relies on the instance cues outputted by PG and CL for WSSS prediction from the semantic branch in the Segmenter to generate the synthetic center and offset maps through the Location Cue Extractor. At the same time, the CL for WSSS prediction is used for the Semantic-aware Selective Distillation to preserve the old center & offset knowledge from the Previous-step Segmenter.

Figure 1: Training phases. Our method includes improving Phase 1: CL for WSSS and upgrading to Phase 2: CL4WSIS. Suppose that horse is the old class and person is the current class during continual learning.

branch in the Segmenter for generating the center and offset maps is frozen in this phase.

In Phase 2 shown in Fig. 1(b), the whole network, except for the instance branch, is frozen. We rely on the instance cues outputted by PG and CL for WSSS predictions from the semantic branch of the Segmenter to generate the synthetic center and offset maps through Location Cue Extractor. At the same time, the CL for WSSS predictions are also used in the Semantic-aware Selective Distillation to keep the old center and offset knowledge from Previous-step Segmenter.

References

[1] Nikita Araslanov and Stefan Roth. Single-stage semantic segmentation from image labels. In *CVPR*, pages 4252–4261, 2020. 1

[2] Fabio Cermelli, Dario Fontanel, Antonio Tavera, Marco Ciccone, and Barbara Caputo. Incremental learning in semantic segmentation from image labels. In *CVPR*, pages 4361–4371, 2022. 1

[3] Michal Lukasik, Srinadh Bhojanapalli, Aditya Menon, and Sanjiv Kumar. Does label smoothing mitigate label noise? In *ICML*, pages 6448–6458. PMLR, 2020. 1