Self-Supervised Pyramid Representation Learning for Multi-Label Visual Analysis and Beyond Supplementary Material

Cheng-Yen Hsieh* Chih-Jung Chang* Fu-En Yang Yu-Chiang Frank Wang
National Taiwan University
chengyeh@andrew.cmu.edu, {b06201018, f07942077, ycwang}@ntu.edu.tw

A. Pseudocode of SS-PRL

We provide the pseudocode of our proposed Self-Supervised Pyramid Representation Learning (SS-PRL) in Algorithm 1.

B. Implementation Details

Implementation Details of SS-PRL. We conduct all experiments by training 200 epochs with a batch size of 128 on MSCOCO [10] or 100 epochs with a batch size of 256 on ImageNet [3]. We use Resnet-50 [8] as our backbone network. The projection head for each scale is a 2-layer multi-layer perceptron (MLP) head that embeds the pyramid representations into a 128-dim space ($D = 128$). We use single linear layers as the cross-scale correlation learners (Section 3.2.2). The number of prototypes K_0, K_1 and K_2 are set to 300, 150, and 150, respectively. The temperature parameter τ in Equation (1) is set to 0.1. The weight λ in Equation (4) is set as 1.0. We set α_s (Equation (2)) as 1.0 for $s = 0$ and 0.25 for other scales, and β_s (Equation (3)) is selected as the same values as α_s for simplicity. We adopt the same optimization method as in [1]. We use a weight decay of 10^{-6}, LARS optimizer [20] with an initial learning rate of 0.6, and the cosine learning rate decay with a final value of 0.0006.

Data Augmentations. Unless specified, we construct pyramid views consisting of 3 different spatial scales (*i.e.*, $s = 0, 1, 2$). The image patch groups V_0, V_1 and V_2 contain 1, 4, and 9 non-overlapping augmented patches, respectively. Our image patch groups are generated as follows: First, a random region is cropped with at least 14%, 60%, 60% of the original image and then resized to 224×224, 314×314, and 383×383 for $s = 0, 1, 2$. We directly use the 224×224 image as the single patch in V_0, *i.e.*, at the global image level. To generate V_1 consisting of 4 patches ($M_1 = 4$), we resize the image to 314×314, divide it into 2×2 grids, and then apply the random crop on each grid to get patches of size 112×112; to generate V_2 containing 9 patches ($M_2 = 9$), we divide the 383×383 resized image into 3×3 grids and then apply the random crop on each grid to get patches of size 96×96. Finally, all the patches are applied with the same set of image augmentation, which includes random horizontal flip, color jittering, gray-scale conversion, and Gaussian blur, following the same implementation as [2, 6, 1].

Implementation Details of Downstream Multi-Label Classification with Fine-tuned Linear Classifiers. We train a linear multi-label classifier on top of the fixed pre-trained backbone network (*i.e.*, ResNet-50) on COCO `train2014` [10] and VOC `trainval07` [4], and then report mean average precision (mAP) on COCO `val2014` [10] and VOC `test2007` [4]. All images are first resized to 256 pixels along the shorter side, and then center-cropped to 224×224. We train the linear classifier using stochastic gradient decent for 50 epochs with weight decay of 0.0001, momentum of 0.9, and a batch size of 32/128 on VOC [4]/COCO [10]. Grid search of range $[0.001, 3.0]$ is applied to the initial learning rate of each linear classifier pre-trained with different method. The learning rate reduces by a factor of 10 three times (equally spaced intervals).

Implementation Details of Downstream Multi-Label Classification in Semi-Supervised Settings. We fine-tune the whole network with 1%, 10%, and 100% labeled data from COCO `train2014` [10] and report mAP on COCO `val2014` [10]. The network is fine-tuned for 20 epochs with a batch size of 32 when using 1% labeled data and a batch size of 128 when using 10% or 100% labeled data. We use grid search within the range of $[0.001, 0.1]$ to set distinct learning rates for the backbone network and the final linear layer, and we decay the learning rates by 0.2 at the 12th and 16th epochs. Other hyperparameters are kept the same as in the linear evaluation setting, *i.e.*, fine-tuned

*Authors contributed equally

Algorithm 1: Self-Supervised Pyramid Representation Learning (SS-PRL)

Input: feature extractor f_θ, cross-scale correlation learners $g_\phi = \{g_{\phi,s}\}_{s=1}^{S}$, multi-level semantic prototypes $\{C_s\}_{s=0}^{S}$, batch size B, temperature τ, weights of loss terms $\{\alpha_s\}_{s=0}^{S}$, $\{\beta_s\}_{s=1}^{S}$, and λ

Data: unlabeled dataset $\mathcal{D}_u$

Output: backbone network f_θ

for sampled minibatch x **do**

  for s in $0 : S$ **do**

    generate V_s and V'_s from x

    $Z_s,\ Z'_s = f_\theta(V_s),\ f_\theta(V'_s)$ # Z_s: $(B \times M_s, D)$

    $Q_s,\ Q'_s = S\text{-}K(Z_s, C_s),\ S\text{-}K(Z'_s, C_s)$ # Q_s: $(B \times M_s, K_s)$

    $P_s,\ P'_s = softmax(\frac{1}{\tau} Z_s C_s),\ softmax(\frac{1}{\tau} Z'_s C_s)$ # P_s: $(B \times M_s, K_s)$

  end

  # loss function

  $L_{pyr} = -\sum_s \alpha_s mean(Q_s \log P'_s + Q'_s \log P_s)$

  $L_{cross} = -\sum_{s \neq 0} \beta_s mean(Q_0 \log g_{\phi,s}(\texttt{Avg}(P_s)) + Q'_0 \log g_{\phi,s}(\texttt{Avg}(P'_s)))$

  $L = L_{pyr} + \lambda L_{cross}$

  # optimization step

  update f_θ, g_ϕ, and $\{C_s\}_{s=0}^{S}$ to minimize L

  normalize $\{C_s\}_{s=0}^{S}$

end

def Avg(P_s)**:**

  P_s = P_s.view([B, M_s, K_s])

  return $P_s.mean$(dim=1)

linear classifiers.

Implementation Details of Downstream Object Detection and Instance Segmentation. We use the $1\times$ schedule in Detectron2 [17] to fine-tune a Mask R-CNN [7] detector with FPN [9] backbone (pre-traned with each self-supervised method) on COCO `train2014` [10] and evaluate on COCO `val2014` [10]. Following the settings in [16, 15], synchronized batch normalization is applied in backbone, FPN [9] and prediction heads during the training. We report the result of detector with $15k$ training iterations to compare the transfer ability of each SSL pre-training method.

C. Further Quantitative Comparisons

In this section, we provide further quantitative comparisons of our SS-PRL against existing SSL methods on downstream semantic segmentation tasks. We also conduct experiments to verify the effectiveness of visual pre-training with SS-PRL on downstream state-of-the-art multi-label classification methods. For fair comparisons, all models are pre-trained on COCO `train2014` [10] for 200 epochs with a batch size of 128 or on ImageNet [3] with a batch size of 256.

Downstream Semantic Segmentation on PASCAL VOC. We provide comparisons with previous SSL methods using the downstream semantic segmentation task on PASCAL VOC [4] to further exhibit the effectiveness and robustness of our SS-PRL. We report the mIoU using an FCN backbone network [11] fine-tuned on VOC `train_aug2012` set (10582 images) [4] for $20k$ iterations and then evaluated on `val2012` set [4]. We mostly follow the same settings as [16], where the first 7×7 convolution is kept, the batch size is set to 16, and the crop size is selected as 512, except the learning rate is set to 0.05. As shown in Table S-A, SS-PRL outperforms most SSL methods by achieving 62.6% mIoU and even exceeds SSL methods that are specifically designed for dense prediction tasks [18, 21, 19].

Transfer to SOTA Multi-Label Classification Methods. To examine the impact of visual pre-training with SS-PRL on downstream state-of-the-art methods, we further provide results by fine-tuning the pre-trained backbone network on COCO `train2014` [10] with the asymmetric loss [13]. Following the settings in [13], we train the model with Adam optimizer and 1-cycle policy [14], with a maximal

Method		Semantic Segmentation on VOC (mIoU)
Random Init.		40.7
MoCo v2 [6]	*general-purpose SSL*	57.3
SwAV [1]		56.1
BYOL [5]		54.1
DenseCL [16]	*dense prediction SSL*	<u>63.2</u>
DetCo [18]		47.6
MaskCo [21]		<u>59.8</u>
InsLoc [19]		56.1
SS-PRL (ours)		<u>62.6</u>

Table S-A. **Downstream semantic segmentation task on the PASCAL VOC dataset.** We report the mean IoU (mIoU) on VOC with fine-tuned FCN models. All methods are pre-trained on COCO for 200 epochs. Top-3 best pre-training methods are <u>underlined</u>.

		Multi-Label Classification with ASL [13] on COCO (mAP)	
Method		Pretrained on COCO	Pretrained on ImageNet
Random Init.	without pre-training	50.0	50.0
MoCo v2 [6]	*general-purpose SSL*	63.8	71.1
SwAV [1]		68.3	73.2
BYOL [5]		62.7	68.6
DenseCL [16]	*dense prediction SSL*	68.7	72.7
DetCo [18]		64.7	71.0
MaskCo [21]		65.8	70.8
InsLoc [19]		62.9	71.8
SS-PRL (ours)		**69.3**	**73.8**

Table S-B. **Transfer to a SOTA multi-label classification method.** We report the mAP on COCO with the whole network fine-tuned with asymmetric loss [13] for 40 epochs. All methods are pre-trained on COCO with 200 epochs or ImageNet with 100 epochs, respectively.

learning rate of 2e-4. All the models are fine-tuned for 40 epochs. Note that we use ResNet-50 as our backbone instead of TResNet-L as used in [13]. As shown in Table S-B, SS-PRL again outperforms other SSL methods with 69.3 % mAP when pre-trained on COCO and 73.8 % mAP when pre-trained on ImageNet. The experiment demonstrates the effectiveness of SS-PRL in downstream multi-label image classification, even when applied to state-of-the-art methods.

D. More Visualization Results

Prototypes learned at each scale. Figure A1 visualizes the semantic concepts portrayed by our learned prototypes at different patch levels on the COCO dataset [10]. We observe that prototypes at scale 0 tend to encapsulate the concept of entire scenes that include multiple objects or items (*e.g*. motorcycle with a rider or steam train blowing smoke), while prototypes at scale 1 tend to capture a single object in an image (*e.g*. a motorcycle or a train), and prototypes at scale 2 tend to portray a fine-grained item (*e.g*. a tire or rail). It shows that SS-PRL is able to discover the hierarchical semantic structure of the dataset with our proposed pyramid representation learning.

E. Further Analysis

For the completeness of analysis, we conduct further experiments by training SS-PRL with two spatial scales (*i.e*., $s = 0, 1$). SS-PRL trained with a single spatial scale (*i.e*., $s = 0$) is adopted as the baseline. All the implementation details follow the settings introduced in Section B, except that here we set the number of prototypes K_1 as 300, the weights of loss terms α_1 and β_1 as 0.5.

Further Discussion on Patch-Level Semantic Prototypes. In Table S-C, we report the linear evaluation [12] results of SS-PRL with semantic prototypes C_s trained under different scenarios. Similar to Table 4 in our main paper, we observe that SS-PRL still achieves the best performance when distinct sets of prototypes are applied in different spatial scales, indicating that exploring the hierarchical infor-

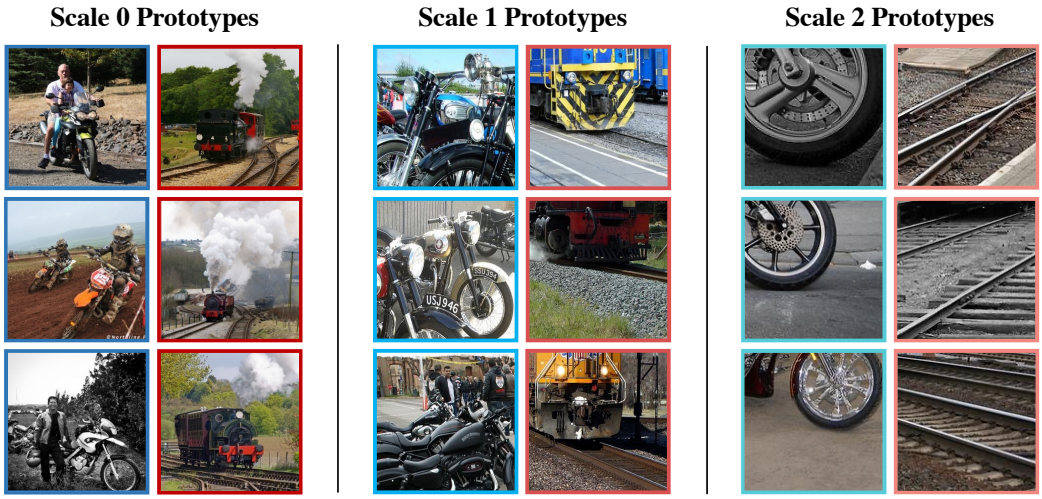

Figure A1. **Visual concepts captured by the learned prototypes at each scale on COCO.** Each column show example images/patches associated with the learned prototypes (in each sub-column) at the corresponding patch scale. Our SS-PRL is shown to exploit hierarchical semantic information from image data by learning patch-level prototypes with semantic practicality (*e.g.*, scenes with motorcycles at scale 0, motorcycle at scale 1, and tires at scale 2).

Prototype	mAP
Baseline	79.2
Shared across all scales	77.8
Learned & correlated across scales	**79.9**

Table S-C. **Ablation Studies on the derived patch-level prototypes with two spatial scales ($s = 0, 1$).** Note that *Shared across all scales* indicates the same prototypes learned across patch scales (*i.e.*, same C_s at different patch scales in Fig. 1). Similar to the results in Table 4, we see that prototypes learned from each scale enforced by our cross-scale correlation would be desirable.

mation from the dataset is crucial for the multi-label classification downstream task.

Parameter Analysis. In Table S-D, we evaluate the influence of the number of semantic prototypes utilized at two spatial scales by reporting the multi-label classification mAP on VOC [4]. Here we only adopt two image scales (*i.e.*, $s = 0, 1$) for network training to better understand the relationships between K_s from two different spatial scales. The results show that the performance greatly degrades when the number of prototypes at the global image level K_0 is selected as a large number (3000). It indicates that K_0 should not be varied considerably from the actual number of labels (80) of the training dataset (*i.e.* COCO [10]), since a large K_0 easily leads to over-clustering or redundant information. We also experiment with different numbers of patch-level prototypes (*e.g.* scale 1 K_1) with a fixed K_0 (300). As K_1 increases from 0 to 300, the performance improves from 79.2 % to 79.9 % mAP. This suggests that the exploitation of fine-grained information from patches indeed facilitates the downstream multi-label classification task. Note that the performance converges and even degrades when K_1 increases from 150 to 1500. The sweet point in the observed reverse U-shape of performance lies at $K_1 = K_0 = 300$, indicating the number of prototypes chosen at each spatial scale should be comparable.

# of Prototypes		
Scale 0 (K_0)	Scale 1 (K_1)	mAP
3000	1500	77.6
300	1500	79.6
300	300	79.9
300	150	79.9
300	0	79.2

Table S-D. **Ablation: Impact of the number of semantic prototypes K_s for multi-label classification on VOC dataset.** With the feature extractor learned and froze with different supervised/self-supervised methods, we report the mAP on VOC with *fine-tuned linear classifiers*. The model shows the best results when trained with comparable K_0 and K_1.

References

[1] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *arXiv preprint arXiv:2006.09882*, 2020.

[2] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020.

[3] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[4] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2):303–338, 2010.

[5] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H. Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Michal Valko. Bootstrap your own latent: A new approach to self-supervised learning. 2020.

[6] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9729–9738, 2020.

[7] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[9] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017.

[10] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[11] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

[12] Ishan Misra and Laurens van der Maaten. Self-supervised learning of pretext-invariant representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6707–6717, 2020.

[13] Tal Ridnik, Emanuel Ben-Baruch, Nadav Zamir, Asaf Noy, Itamar Friedman, Matan Protter, and Lihi Zelnik-Manor. Asymmetric loss for multi-label classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 82–91, 2021.

[14] Leslie N Smith. A disciplined approach to neural network hyper-parameters: Part 1–learning rate, batch size, momentum, and weight decay. *arXiv preprint arXiv:1803.09820*, 2018.

[15] Yonglong Tian, Chen Sun, Ben Poole, Dilip Krishnan, Cordelia Schmid, and Phillip Isola. What makes for good views for contrastive learning? *Advances in Neural Information Processing Systems*, 33:6827–6839, 2020.

[16] Xinlong Wang, Rufeng Zhang, Chunhua Shen, Tao Kong, and Lei Li. Dense contrastive learning for self-supervised visual pre-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3024–3033, 2021.

[17] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. `https://github.com/facebookresearch/detectron2`, 2019.

[18] Enze Xie, Jian Ding, Wenhai Wang, Xiaohang Zhan, Hang Xu, Peize Sun, Zhenguo Li, and Ping Luo. Detco: Unsupervised contrastive learning for object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8392–8401, 2021.

[19] Ceyuan Yang, Zhirong Wu, Bolei Zhou, and Stephen Lin. Instance localization for self-supervised detection pretraining. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3987–3996, 2021.

[20] Yang You, Igor Gitman, and Boris Ginsburg. Scaling sgd batch size to 32k for imagenet training. *arXiv preprint arXiv:1708.03888*, 6:12, 2017.

[21] Yucheng Zhao, Guangting Wang, Chong Luo, Wenjun Zeng, and Zheng-Jun Zha. Self-supervised visual representations learning by contrastive mask prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10160–10169, 2021.