腾讯优图

FoPro: Few-Shot Guided Robust Webly-Supervised Prototypical Learning

*Yulei Qin, * Xingyu Chen, * Chao Chen, Yunhang Shen, Bo Ren, Yun Gu, Jie Yang, Chunhua Shen*

AAAI 2023

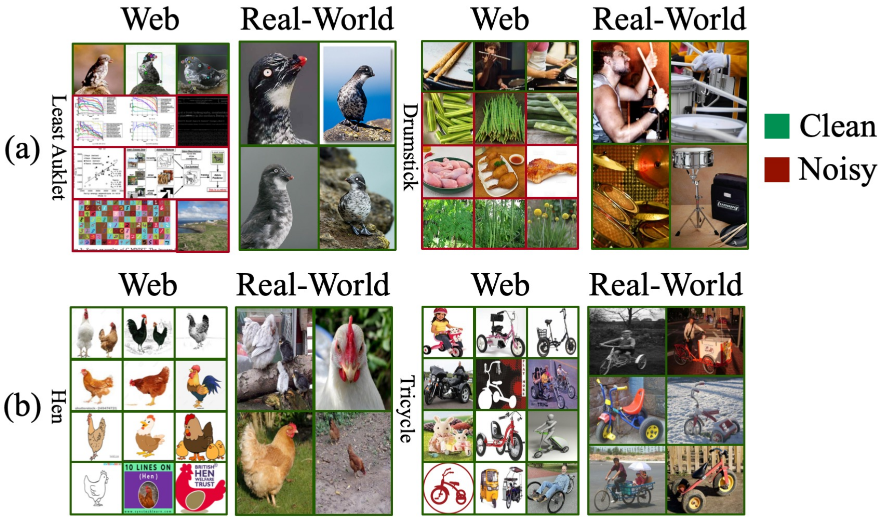
Tencent YouTu Lab

*: Equal Contribution

Webly-Supervised Learning (WSL)

腾讯优图

- How to learn robust representations from **abundant, weakly-labeled** web images?
  - Challenge 1: label noise
  - Challenge 2: domain gap

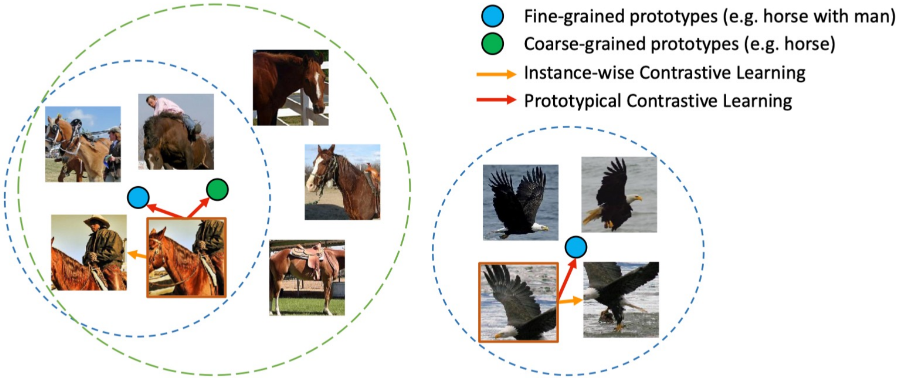


- Generalizability on real-world testing sets is not emphasized.

Prototypical & Contrastive Learning

腾讯优图

- Prototypes
  - A representative embedding for a group of **semantically similar** instances
- Contrastive Learning
  - A self-supervised learning method that brings samples from the **same** instance **closer**, and separates samples from **different** instances **farther**
- Why Prototype + Contrastive Learning?
  - Instance-wise contrast push two different instances of the same class
  - Prototypical-instance contrast encourages formulation of semantic structure



1) Prototypical Contrastive Learning of Unsupervised Representations, ICLR 2021
2) Learning from Noisy Data with Robust Representation Learning, ICCV 2021
3) MoPro: Webly-Supervised Learning with momentum Prototypes. ICLR 2021

FoPro for A Brand-new Setting of WSL

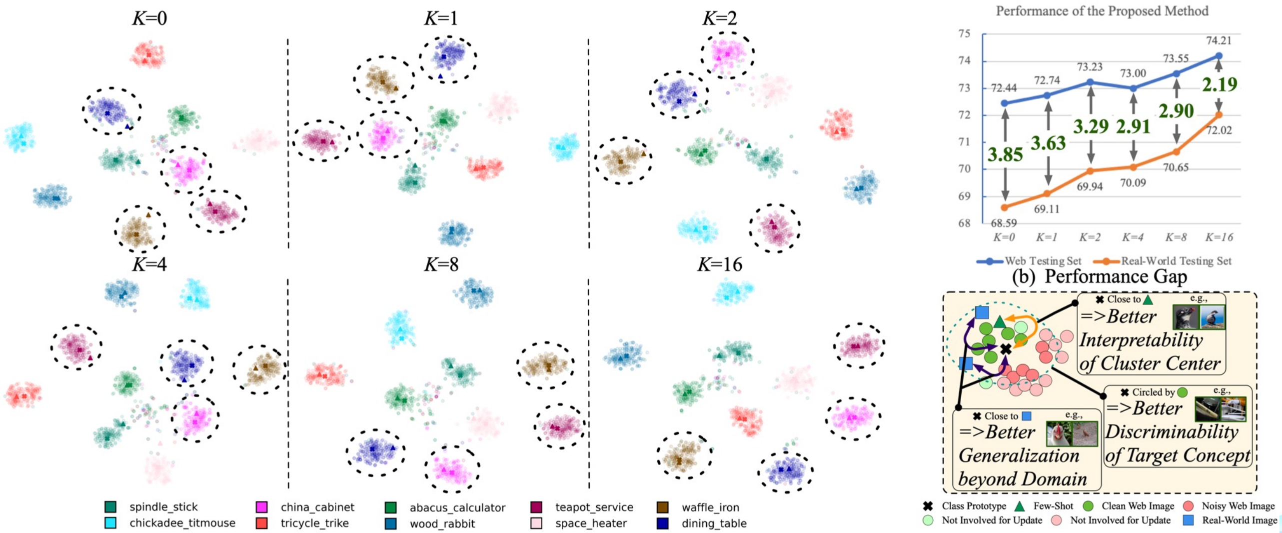
腾讯优图

Objective

- Learn from web data for real-world applications
- Two key problems: 1) whom to learn from; 2) what to learn

Contribution

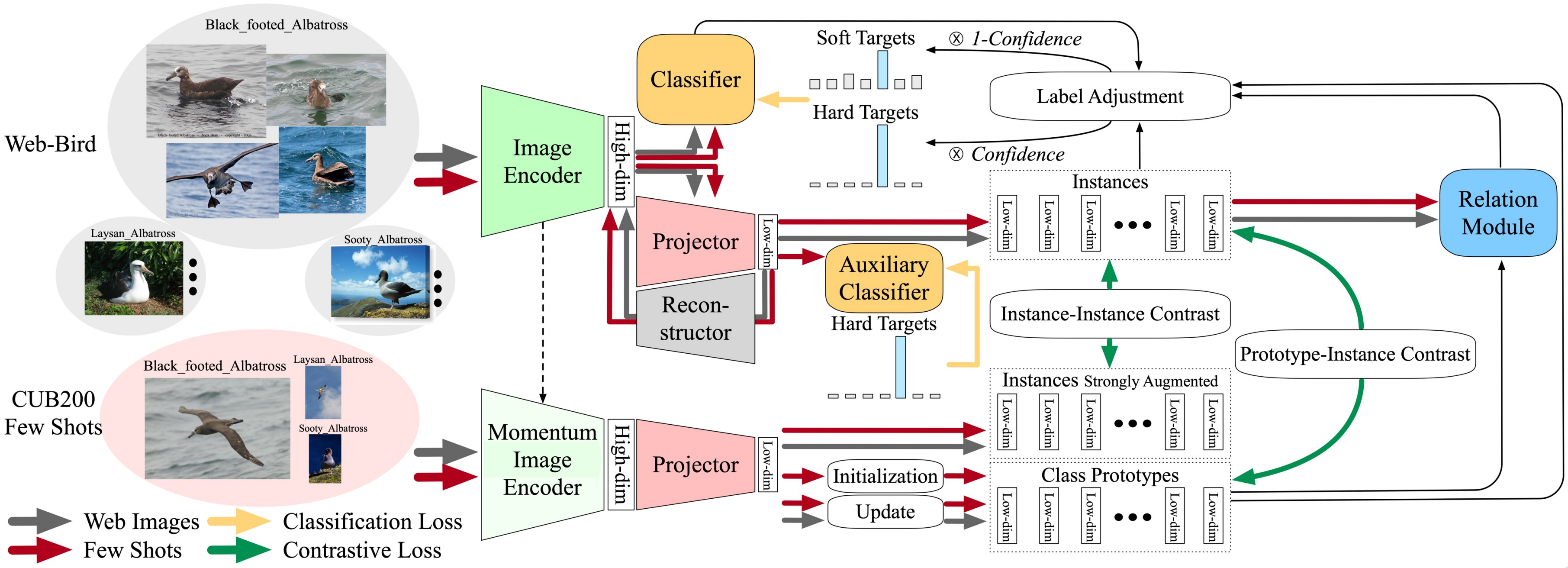
- A new few-shot setting
- A new prototypical contrastive representation learning method
- A new flexible relation module



(a) Effect of Few-Shots for Guidance (c) Mechanism of the Proposed Method

FoPro Model Architecture

- Two siamese encoders, one classifier, one projector, one reconstructor, one auxiliary classifier, and one relation module



FoPro Stage 1: Preparation

腾讯优图

- Learn common, regular patterns for the encoder via:

$$\mathcal{L}_i^{cls} = -\log(\mathbf{p}_{i(y_i)}^{\{w;t\}}).$$

- Extract principal, distinguishable low-dimensional embeddings via:

$$\mathcal{L}_i^{prj} = \|\tilde{\mathbf{v}}_i^{\{w;t\}} - \mathbf{v}_i^{\{w;t\}}\|_2^2 - \log(\mathbf{q}_{i(y_i)}^t)$$

FoPro Stage 2: Incubation

腾讯优图

- Initialize prototypes with labeled, real-world fewshots via:

$$\hat{\mathbf{c}}_k = \frac{1}{K}\sum_{y_i=k}\mathbf{z}_i^t, \mathbf{c}_k = \frac{\hat{\mathbf{c}}_k}{\|\hat{\mathbf{c}}_k\|_2}.$$

- Pull instances closer to their prototypes via:

$$\mathcal{L}_i^{pro} = -\log\frac{\exp((\mathbf{z}_i^{w;t}\cdot\mathbf{c}_{y_i} - \delta^{w;t})/\phi_{y_i})}{\sum_{k=1}^{C}\exp((\mathbf{z}_i^{w;t}\cdot\mathbf{c}_k - \delta^{w;t})/\phi_k)},$$

- Pull instances from the same sample closer via:

$$\mathcal{L}_i^{ins} = -\log\frac{\exp(\mathbf{z}_i^{w;t}\cdot\mathbf{z}_i^{\prime w;t}/\tau)}{\sum_{j=1}^{Q}\exp(\mathbf{z}_i^{w;t}\cdot\mathbf{z}_j^{\prime w;t}/\tau)},$$

- Tighten/loosen class cluster distribution adaptively via:

$$\phi_k = \frac{\sum_{y_i=k}\|\mathbf{z}_i^{w;t} - \mathbf{c}_k\|_2}{N_k^{w;t}\log(N_k^{w;t}+\alpha)},$$

FoPro Stage 3: Illumination

腾讯优图

- Select clean sample for relation module via:

$$D^r = D^t \cup \{(\mathbf{x}_i^w, y_i^w) | \sum_{j=1}^{C} |(\mathbf{z}_i^w - \mathbf{c}_{y_i}) \cdot \mathbf{c}_j| \le \sigma\},$$

- Learn the metric on instance-prototype similarity via:

$$\mathcal{L}_i^{rel} = -\log \frac{\exp(r_{iy_i})}{\sum_{k=1}^{C} \exp(r_{ik})}.$$

FoPro Stage 4: Verification

腾讯优图

- Complete wrong label correction, out-of-distribution sample removal via:

$$\mathbf{s}_i^w = \beta \mathbf{p}_i^w + (1-\beta)[\mathbf{c}_1, ..., \mathbf{c}_C]^T \cdot \mathbf{z}_i^w$$

$$\hat{y}_i^w = \begin{cases} y_i^w & \text{if } r_{iy_i} > \gamma, \\ \arg\max_k \mathbf{s}_{i(k)}^w & \text{else if } \max_k \mathbf{s}_{i(k)}^w > \gamma, \\ y_i^w & \text{else if } \mathbf{s}_{i(y_i)}^w > 1/C, \\ \text{Null } (OOD) & \text{otherwise,} \end{cases}$$

- Leverage self-knowledge with model prediction and self-contained confidence via:

$$\begin{aligned} \mathcal{L}_i^{cls} = & -\log(\mathbf{p}_{i(y_i)}^t) - \mathbf{s}_{i(\hat{y}_i)}^w \log(\mathbf{p}_{i(\hat{y}_i)}^w) \\ & -(1-\mathbf{s}_{i(\hat{y}_i)}^w) \sum_{k=1}^C \mathbf{p}_{i(k)}^w \log \mathbf{p}_{i(k)}^w. \end{aligned}$$

- Maintain noise-robust prototypes via:

$$\hat{\mathbf{c}}_k = m_p \mathbf{c}_k + (1-m_p)\mathbf{z}_i^{w;t}, \mathbf{c}_k = \frac{\hat{\mathbf{c}}_k}{\|\hat{\mathbf{c}}_k\|_2}.$$

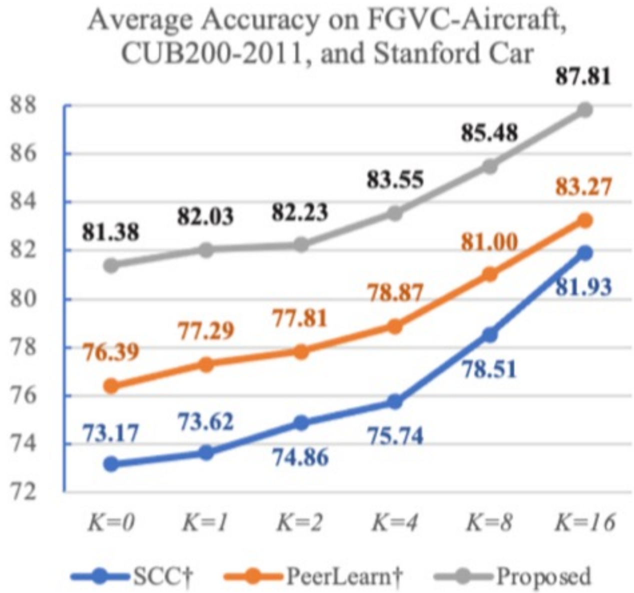
FoPro Results on Fine-Grained Datasets

腾讯优图

- FoPro boosts performance of vanilla backbones more significantly than SOTA methods.
- FoPro achieves consistent performance with an increasing K-shot.

Method	Back-bone	WebFG496 Bird	Air	Car	Avg.
Vanilla	R50	64.43	60.79	60.64	61.95
MoPro†	R50	71.16	76.85	79.68	75.90
SCC†	R50-D	61.10	74.92	83.49	73.17
Vanilla	B-CNN	66.56	64.33	67.42	66.10
Decouple	B-CNN	70.56	75.97	75.00	73.84
CoTeach	B-CNN	73.85	72.76	73.10	73.24
PeerLearn	B-CNN	76.48	74.38	78.52	76.46
PeerLearn†	B-CNN	76.57	74.35	78.26	76.39
FoPro(K=0)	B-CNN	77.79	79.37	86.99	81.38
FoPro(K=1)	B-CNN	78.07	79.87	88.01	82.03
FoPro(K=16)	B-CNN	**85.54**	**86.40**	**91.51**	**87.81**

† Results are reproduced by ourselves with the official codes.



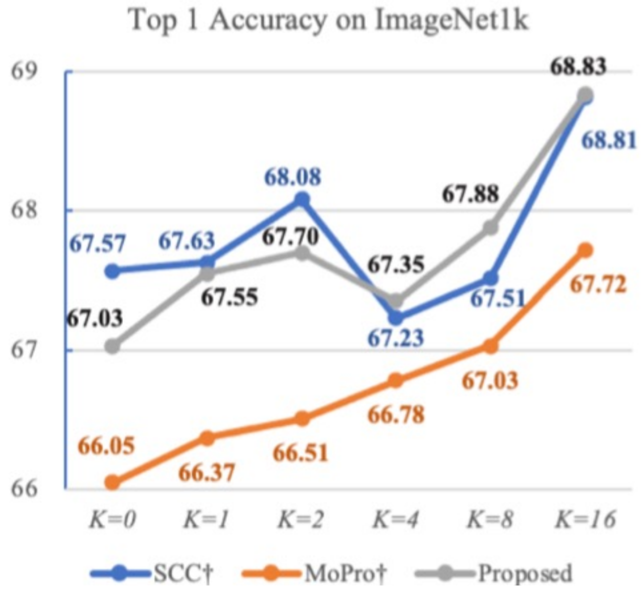
FoPro Results on Large-scale Datasets

腾讯优图

- FoPro is initially preceded by SCC and MoPro, but rises steadily after efficiently exploiting a few real-world examples.
- When K=0, the prototypes are solely initialized by web examples randomly. The relatively higher percentage of noise in WebVision/Google500 causes lower performance.

Method†	Backbone	ImageNet1k Top 1	ImageNet1k Top 5	ImageNet500 Top 1	ImageNet500 Top 5
MentorNet	Inception ResNetV2	64.20	84.80	–	–
Curriculum-Net	Inception V2	64.80	83.40	–	–
Vanilla	R50-D	67.23	84.09	–	–
SCC	R50-D	67.93	84.77	68.84	84.62
SCC†	R50-D	67.57	85.74	64.40	81.56
Vanilla	R50	65.70	85.10	61.54	78.89
CoTeach	R50	–	–	62.18	80.98
CleanNet	R50	63.42	84.59	–	–
MoPro	R50	67.80	87.00	–	–
MoPro†	R50	66.05	85.66	58.68	78.39
PeerLearn†	R50	52.57	73.35	42.04	61.71
FoPro(K=0)	R50	67.03	85.57	68.59	86.03
FoPro(K=1)	R50	67.55	86.31	69.11	86.19
FoPro(K=16)	R50	**68.83**	**87.83**	**72.02**	**89.38**

† Results are reproduced by ourselves with the official codes.



Reduced Gap of Web & Real-word Performance

腾讯优图

- The abnormal case of K=4 is due to sampling jittering where atypica, unrealistic images of certain classes can be sampled from ImageNet1k.
- The reduced gap reflects that FoPro bridges the noisy web domain and real-world domain with limited *K* shots.

K	WebFG496 Avg.		ImageNet1k		ImageNet500	
	Top 1	Gap	Top 1	Gap	Top 1	Gap
0	81.38	–	67.03	5.57	68.59	3.85
1	+0.65	–	+0.52	5.22	+0.52	3.63
2	+0.85	–	+0.67	5.20	+1.35	3.29
4	+2.17	–	+0.32	4.60	+1.50	2.91
8	+4.10	–	+0.85	4.64	+2.06	2.90
16	+6.43	–	+1.80	3.91	+3.43	2.19
16	87.81	–	68.83	–	72.02	–
Ref.	87.16†	–	76.15‡	–	76.22‡	–

† Official results of the B-CNN trained on FGVC-Aircraft, CUB200-2011, and Stanford Car are averaged.

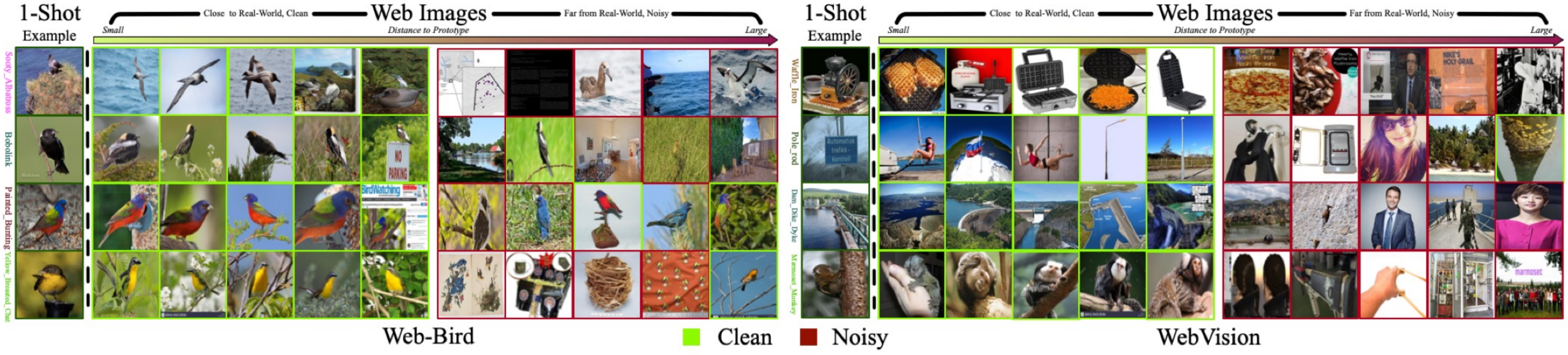
‡ Official results of the R50 trained on ImageNet1k by PyTorch are quoted respectively for 500 and 1k classes.

Ablation Study & Qualitative Results

- Compared with pre-defined, fixed similarity metrics such as cosine distance, our Relation Module (RM) discovers clean examples more precisely and efficiently.

K=1	WebFG496 Avg.	ImageNet1k	ImageNet500
w/o RM	81.59	65.22	64.69
w RM	82.03	67.55	69.11

- Visualization on the sorted web examples confirm that the prototypes we polished can be used to indicate clean, web images.



Conclusion

腾讯优图

- We propose **FoPro**, the first few-shot guided method for learning from web data, that tackles both **noise** and **domain gap** with a large quantity of web images and a few real-world images.

- Our contribution
  - We propose a new **few-shot** learning setting in WSL with abundant noisy web images and **a few real-world images**, which aims to improve the performance of WSL for real-world applications in a cost-efficient way.
  - We present **FoPro** to solve noise and data bias in an end-to-end manner, which relies on the formulation of **class-representative** and **domain-generalized prototypes**.
  - We propose **relation module** for label noise correction. It outperforms fixed metrics (cos distance) by evaluating instance-prototype similarity with a learnable metric.
  - Performance under the increasing K-shot settings demonstrates that **FoPro** utilizes few shots wisely to bridge the gap towards **real-world applications**.

腾讯优图

Thanks for your attention!