Supplementary material PLS-LSA

Paul Albert[1], Jack Valmadre[1], Eric Arazo[2], Tarun Krishna[3], Noel E. O'Connor[3], Kevin McGuinness[3]

[1] Australian Institute for Machine Learning, University of Adelaide
[2] CeADAR: Ireland's Centre for Applied Artificial Intelligence
[3] Insight Centre for Data Analytics, Dublin City University
paul.albert@adelaide.edu.au

Supplementary material overview. Section 1 details the training hyper-parameters for experiments on the CNWL, mini-Webvision and Webly-fg datasets. Section 2 reports results when training PLS-LSA and PLS-LSA+ on mini-Webivision using a ResNet50 as well as results for related state-of-the-art algorithms. Section 3 studies if PLS-LSA+ and other SOTA co-training alternatives produce individual neural networks that are significantly more accurate than non co-trained strategies. Section 4 studies different strategies for LSA including using a trusted ID/OOD subset, different alternating strategies and computing the linear separation using features at different depth in the network. Section 5 shows the complementarity of the noise retrieval metrics used in PLS-LSA and examples of missed clean examples by one metric but retreived by the other. Section 7 reports top-2 and top-5 accuracy results for the Webly-fg datasets. Finally, Section 8 displays noisy images detected using PLS-LSA for the mini-Webivision or the Webly-fg datasets.

1 Training details

1.1 CNWL

The Controlled Web Noisy Label (CNWL) [8] proposes a controlled web-noise corruption of MiniImageNet [18] where some images of the original dataset are replaced with human curated incorrect samples obtained from web-queries on the corresponding class. We train on the CNWL at a resolution of 32×32 using a pre-activation ResNet18 [5]. We train for 200 epochs using a cosine decay scheduling from a learning rate of 0.1. We optimize the network using stochastic gradient descent (SGD) with a weight decay of 0.0005. For training augmentations, we use random cropping and horizontal flipping, for the strong augmentations, we use a random resize copping strategy followed by RandAugment [4] with parameters 1 and 6. For unsupervised pretraining, we train SimCLR for 1.000 epochs using the solo-learn [17] library.

1.2 mini-Webvision

Webvision [12] is a real world classification web-dataset over the classes of ImageNet [10] the original paper estimates the noise level in Webvision to be between 20% to 34%. As in previous research, we train on the first 50 classes

Table 1: Classification accuracy training on mini-Webvision using ResNet50. We denote with † algorithms using unsupervised initialization. We test on the mini-Webvision valset and ImageNet 1k test set (ILSVRC12). We run PLS and PLS-LSA, other results are from the respective papers. −− denotes that the papers did not report any results. We bold the best results. Accuracy results averaged over 3 random seeds ± one std.

Valset		M	DM	†C2D	TCL	LCI	NCR	†PLS-LSA	†PLS-LSA+
mini-WebVision	top-1	76.0±0.2	76.3±0.36	79.4±0.3	79.1	80.0	80.5	82.5±0.2	83.2±0.2
	top-5	90.0±0.1	90.7±0.16	92.3±0.3	92.3	−−	−−	94.1±0.8	94.8±0.2
ILSVRC12	top-1	72.1±0.4	74.4±0.29	78.6±0.4	75.4	−−	−−	79.9±0.2	80.6±0.3
	top-5	89.1±0.3	91.2±0.12	93.0±0.1	92.4	−−	−−	94.6±0.6	95.2±0.1

(mini-Webvision) which yields 65,944 training images and using an Inception-ResNetV2 [16] or a ResNet50 [9] architecture. We train at a resolution of 224×224 for 130 epochs and otherwise the same optimization regime as for the CNWL dataset (cosine lr decay, SGD, weight decay 0.0005) but with a batch size of 64 and from an initial learning rate of 0.02 (0.01 for ResNet50). The training augmentations are resizing to 256×256 before random cropping to 224×224 and random horizontal flipping. The strong augmentations are first resizing to 256×256 then random resize cropping to 224×224 and applying RandAugment with parameters 1 and 4. For unsupervised pretraining, we train SimCLR for 400 epochs using the solo-learn library.

1.3 Webly-fg

We also evaluate PLS-LSA on the Webly-fine-grained (Webly-fg) datasets [15] which are real world fine-grained classification datasets build from web queries. We train specifically on the web-bird, web-car and web-aircraft subsets that respectively contain 200, 196 and 100 classes. Each dataset contains 18.388, 21.448, 13.503 training, and 5.794, 8.041, 3.333 test images. We train a ResNet50 network with a batch size of 32, at a resolution of 448×448 for 110 epochs. Our intial learning rate is 0.006 and we train using cosine decay, SGD and a weight decay of 0.001). The training augmentations are resizing to 512×512 then cropping to 448×448 and random horizontal flipping. For the strong augmentations, we resize to 512×512 then random resize crop to 448×448 and apply RandAugment with parameters 1 and 4. Unsupervised pretraining is the same as Webvision but at a resolution of 448×448.

2 mini-Webvision results with ResNet50

We report in Table 1 results for noise-robust algorithms training a ResNet50 on the mini-Webvision dataset. We report results for Contrast to Divide (C2D) [22] that trains DivideMix (DM) [11] from a SimCLR initialization, Twin Contrastive Learning (TCL) [6] that trains a two-head contrastive network where the distribution of ID and OOD samples are captured by a two mode Gaussian mixture model, Label Confidence Incorporation (LCI) [1] that uses a teacher network trained on noisy data to supervise a noise-free student model and Neighbor Consistency Regularization (NCR) [7] that regularizes samples close in the feature space to have similar supervised predictions. Similarly to results using InceptionResNetV2 in the main body of the paper, PLS-LSA improves over related work from 1 to 2 accuracy points and PLS-LSA+ further improves the results of PLS-LSA by 0.5 to 1 absolute point.

3 Are co-training benefits only limited to network ensembling at test time ?

Because co-training is now a common strategy for label noise robustness as many newer methods [3, 14, 21] build up on DivideMix (DM) [11], we aim to find out if co-training strategies produces better individual networks or if they are better simply because a network ensemble is used at test time.
We train PLS-LSA+ using the following co-training strategies: an independent approach (Indep) where the only interactions the two networks have is the test time prediction ensemble, the DivideMix co-training strategy (DM) where a network predicts noisy samples for the other and semi-supervised imputation is done using the ensemble prediction of the networks, a naive voting strategy (Vote) where the noisy samples are selected when detected as noisy by both networks (also ensembling for SSL imputation) and our co-training strategy (Ours) where we use the voting strategy but use a co-guessing strategy for the pseudo-loss of PLS (one network validates the SSL imputation for the other).
We report results training PLS-LSA+ using these co-training strategies for noise ratios 0.2 and 0.8 on the CNWL dataset in Table 2. The results are displayed as best accuracy for the ensemble (Ens) and for the individual (Indiv) networks. We additionally report the p-value obtained from a T-test of the current strategy against the independent one to evaluate if the improvement of the current co-train strategy are statistically better than the independent strategy.
We find that our co-training is the strategy producing the most statistically significantly more accurate individual networks ($p < 0.05$) and that the semi-supervised co-validation strategy is important to acheive improved individual networks (Ours vs Vote). We recommend that future label noise research utilizing co-training strategies to conduct similar experiments to prove that the co-training strategy is beneficial beyond network ensembling.

4 Human labeled subset

4.1 Improved noise retrieval using a trusted subset

We visualize here the noise retreival capacities of PLS, RRL, W_{PLS} and W_{RRL} by plotting the Receiver Operating Characteristic Curves (ROC) when identifying noisy samples on the CNWL dataset under 20% and 80% noise. Although

Table 2: Is co-training better than network ensembling ? We report the p-value of each strategy against the independent one. CNWL dataset. We bold the **best accuracy** and underline p-values under 0.05

	20% noise				80% noise			
	Indep	DM	Vote	Ours	Indep	DM	Vote	Ours
Ens.	66.15$\pm$0.23	66.24$\pm$0.22	66.23$\pm$0.18	**66.64**$\pm$0.37	50.61$\pm$0.49	50.83$\pm$0.01	50.76$\pm$0.48	**51.88**$\pm$0.36
p-value	1	0.72	0.74	0.19	1	0.20	0.85	0.20
Indiv.	63.95$\pm$0.42	64.56$\pm$0.46	64.52$\pm$0.13	**64.71**$\pm$0.28	47.50$\pm$0.48	48.34$\pm$0.10	48.38$\pm$0.74	**49.07**$\pm$0.31
p-value	1	0.06	0.015	0.007	1	0.05	0.28	0.03

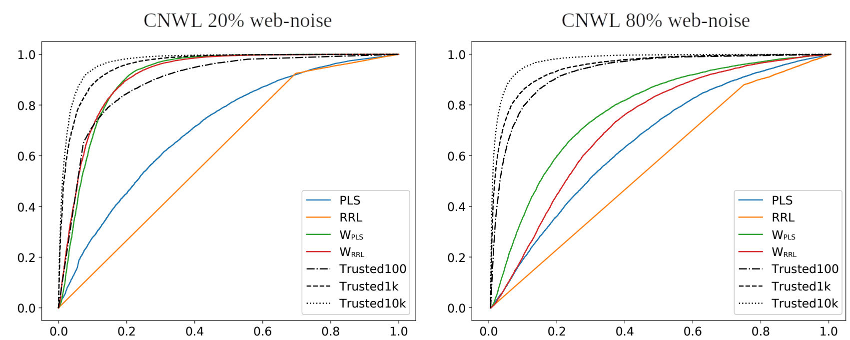

Fig. 1: ROC for different noise-retrieval metrics. We report PLS (loss-based) and RRL (feature-based), the refined detection when they are used as a support set for the logistic regressor (W_{PLS} and W_{RRL} respectively) and results where trusted examples (100, 1k or 10k) are used for training the logistic regressor. Features extracted after the block 2 of a PreAct ResNet18.

this is not a case we study in this paper, we additionally report here the performance of utilizing a human annotated subset (oracle) to compute W. We run experiments where we train the logistic regressor on 10.000 (10k), 1.000 (1k) and 100 randomly selected and ID/OOD-annotated samples. Figure 1 reports the results. We find that W_{PLS} and W_{RRL} import on the metrics that are based on. As little as 100 trusted samples provide a strong noise detection especially in high noise ratios. We leave this information for future work.

4.2 PLS-LSA with a trusted subset

We utlize here the trusted subset computed in the previous subsection to train PLS-LSA/ Results and the comparison against our unsupervised solution can be found in Table 3. We observe that for up to 40% noise corruption, our unsupervised approach performs on par with using 100 to 1.000 trusted samples yet the added supervision of even 100 trusted ID/OOD samples becomes beneficial for the 60% and 80% noise scenarios.

Table 3: Training PLS-LSA using trusted subsets. CNWL dataset with various noise levels. PLS-LSA uses W_{PLS} for the linear separation while others use a trusted subset e.g W_{100}. Results averaged over 3 runs ± one std

Noise ratio	0.2	0.4	0.6	0.8
PLS-LSA	64.43$\pm$0.21	61.14$\pm$0.35	57.18$\pm$0.30	49.53$\pm$0.46
100	64.51$\pm$0.34	61.48$\pm$0.20	58.07$\pm$0.33	50.35$\pm$0.43
1k	64.62$\pm$0.31	61.83$\pm$0.18	58.16$\pm$0.55	50.84$\pm$0.38
10k	64.88$\pm$0.31	62.07$\pm$0.30	58.61$\pm$0.25	51.43$\pm$0.40

Table 4: Different alternating strategies for LSA

Noise ratio	0.2	0.8
modulo 2	64.43$\pm$0.21	49.53$\pm$0.46
random	64.44$\pm$0.02	48.32$\pm$0.17
random sample	63.15$\pm$0.24	46.21$\pm$0.58

4.3 Different strategies to alternate feature and loss detection

We study here different strategies for alternating between Z and W. We propose to compare the approach proposed in the main body of the paper (modulo 2) against a random choice every epoch with a probability of 50% (random) or a random choice for each training sample (random sample) at a given epoch instead of using the same strategy for all samples. Results are displayed for the CNWL dataset under noise perturbations of 0.2 and 0.8 in Table 4. We observe that alternating randomly between Z and W is similarly accurate than regulated alternation every other epoch (modulo 2). Doing a random selection at the sample level (random sample) is however less accurate. These results appear to evidence that maintaining a selection logic (linear separation or small-loss) for a period of time of at least one epoch in our case is beneficial.

4.4 Computing the linear separation at different depth

We study here the influence of computing the linear separation on features at different depth in the network. We run PLS-LSA utilizing features extracted after blocks 0-3 in the ResNet18 architecture as well as utilizing the contrastive projection (block 4 in this case). The results can be found in Table 5. We find that features extracted at block 1 produce the more accurate networks and that the accuracy degrades when using deeper layers. These results are coherent with the observed noise retrieval accuracy using the linear separation in the main body of the paper. For every experiment where we run PLS-LSA, we use average-pooled then L2 normalized features at the end of the 2nd block of our ResNet architecture (feature dimension 128) for W.

Table 5: Using features after different ResNet blocks to compute W_{PLS}, CNWL.

Noise	0.2	0.4	0.6	0.8
0	$64.19_{\pm 0.38}$	$60.66_{\pm 0.41}$	$57.00_{\pm 0.15}$	$48.41_{\pm 0.10}$
1	$64.43_{\pm 0.21}$	$61.14_{\pm 0.35}$	$57.18_{\pm 0.30}$	$49.53_{\pm 0.46}$
2	$63.79_{\pm 0.27}$	$60.35_{\pm 0.29}$	$56.92_{\pm 0.16}$	$49.25_{\pm 0.62}$
3	$63.41_{\pm 0.36}$	$60.19_{\pm 0.14}$	$56.52_{\pm 0.27}$	$49.01_{\pm 0.29}$
4	$63.73_{\pm 0.03}$	$59.92_{\pm 0.03}$	$55.79_{\pm 0.17}$	$48.45_{\pm 0.78}$

Fig. 2: Clean samples missed by our linear separation but retrieved by PLS or RRL. PLS-LSA trained on the CNWL 20%.

5 Missed important samples

5.1 Complementary noise detections

We report how complementary our linear separation retrieval is with a generic small loss approach by plotting every epoch of training PLS-LSA how much of the missed clean samples is retrieved by either PLS or RRL. Figure 2 displays the results where we observe that up to 80% of the missed samples are retrieved and that the further the PLS-LSA training progresses, the less clean samples our linear separation misses (from 40% of the total clean samples at the start of training to less than 20% at the end).

5.2 LSA improves small loss noise detection

Another observation we make of the mutual benefits of our linear separation alternating is the improved noise retrieval of the original PLS metric (small loss) when training PLS-LSA as opposed to PLS alone. Figure 3 reports the AUROC for the PLS noise detection metric retrieving noisy samples in the PLS or PLS-LSA configurations. We observe that the small loss retrieval of PLS is improved

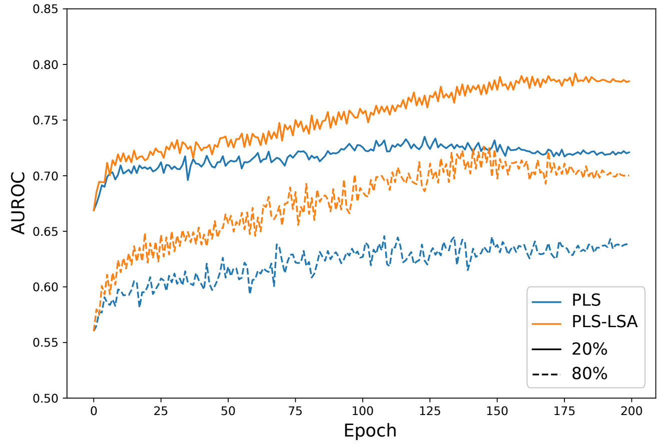

Fig. 3: PLS-LSA improves the small loss noise retrieval of PLS. CNWL under 20% or 80% noise.

when trained with LSA, highlight the complementarity and resulting mutual improvement of each metric.

5.3 Visualizing missed clean samples

Figure 4 displays examples of clean samples missed by our linear separation detection. We display images for classes 0, 15, 17, 25, 36, 45, 59, 61, 95 and 96 of the CNWL dataset (randomly selected). We notice how most of the missed samples are the target object displayed on a uniform background free of distractors. We also report in Figure 5 the opposite scenario: clean images missed by PLS but successfully identified as clean by our linear separation. In this second scenario, we observe that the images missed by PLS but retrieved by our linear separation appear to be more difficult images with a cluttered background or presenting a small instance of the target class.

6 Additional results for CLIP ViT architectures and other noise robust algorithms

6.1 CLIP architectures

We provide here some results on training PLS-LSA on ViT architectures pretrained using a CLIP-like framework [13]. We obtain pretrained weights from the open-clip repository [2] and finetune the ResNet-50 and ViT-B/32 architectures on the CNWL and Webivison datasets. Results are reported in Table 6 where we add non-robust training with mixup and PLS as baselines. We find that LSA scales well when applied to the CNWL dataset but Webvision improvements are less convincing, supposedly because the margin for improvement is small. These

Fig. 4: Examples of clean samples missed by our linear separation W_{PLS} but correctly recovered by PLS (green). 20% noise CNWL. Repeated from the main body for convenience

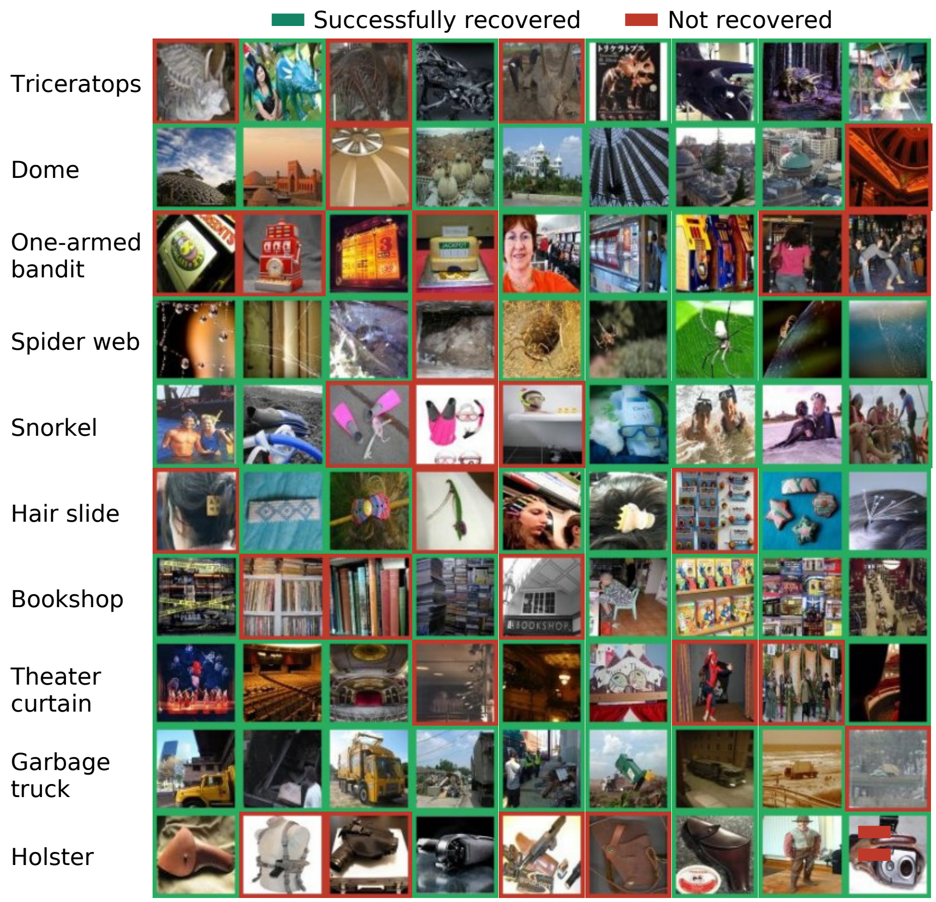

Fig. 5: Examples of clean samples missed by PLS but correctly recovered by our linear separation W_{PLS} (green). 20% noise CNWL.

Table 6: PLS-LSA with CLIP. Top-1 accuracy

	Dataset	CNWL 0.4	CNWL 0.8	Webvision
ViT-B/32	mixup	84.06	78.94	83.80
	PLS	85.72	80.04	85.12
	PLS-LSA	86.70	82.22	85.20
ResNet-50	mixup	77.54	69.56	81.88
	PLS	77.72	70.04	82.36
	PLS-LSA	78.60	71.78	82.64

Table 7: LSA applied to ProMix. Top-1 accuracy with a PreActivation ResNet18.

Dataset	CNWL 0.4	CNWL 0.8	Webvision (32x32)
ProMix	60.70	46.54	64.80
ProMix-LSA	64.00	50.96	68.12
PLS-LSA+	63.42	52.03	69.16

early results suggest that LSA is generalizable to transformer architectures and different manners of contrastive pre-training.

6.2 ProMix-LSA

We report here additional results when training LSA with ProMix [19] the current leader of the CIFAR-N datasets [20] leaderboard [4]. We compare ProMix-LSA with PLS-LSA+ as ProMix utilizes an ensemble of two networks to predict. We report results on the CNWL dataset and Webvision at the resolution of 32×32 as ProMix requires an amount of VRAM too large to train at full resolution with our resources. ProMix-LSA largely improves over ProMix alone in all scenarios.

7 Top-n accuracy web-fg datasets

We report Top-2 and Top-5 accuracy results of PLS-LSA on the Web-fg datasets in Table 8. We observe that top-2 accuracy offers a significantly improvement over top-1 classification which indicates that PLS-LSA rarely catastrophically fails as if the target class is not the most accurate prediction is is often the second best.

8 Example of detected noisy samples

Figure 6 reports examples of training samples we detect as noisy with PLS-LSA and Figures 7, 8 and 9 report detected noisy examples on Web-car/bird/aircraft.

[4] http://noisylabels.com/

Table 8: Top-K classification accuracy of PLS-LSA on the Webly-fg datasets

	Web-Aircraft	Web-bird	Web-car
Top-1	87.82	79.47	86.76
Top-2	95.11	84.73	94.03
Top-5	97.54	93.92	97.57

Fig. 6: Examples of detected noisy samples in Webvision

Fig. 7: Examples of detected noisy samples in Web-car

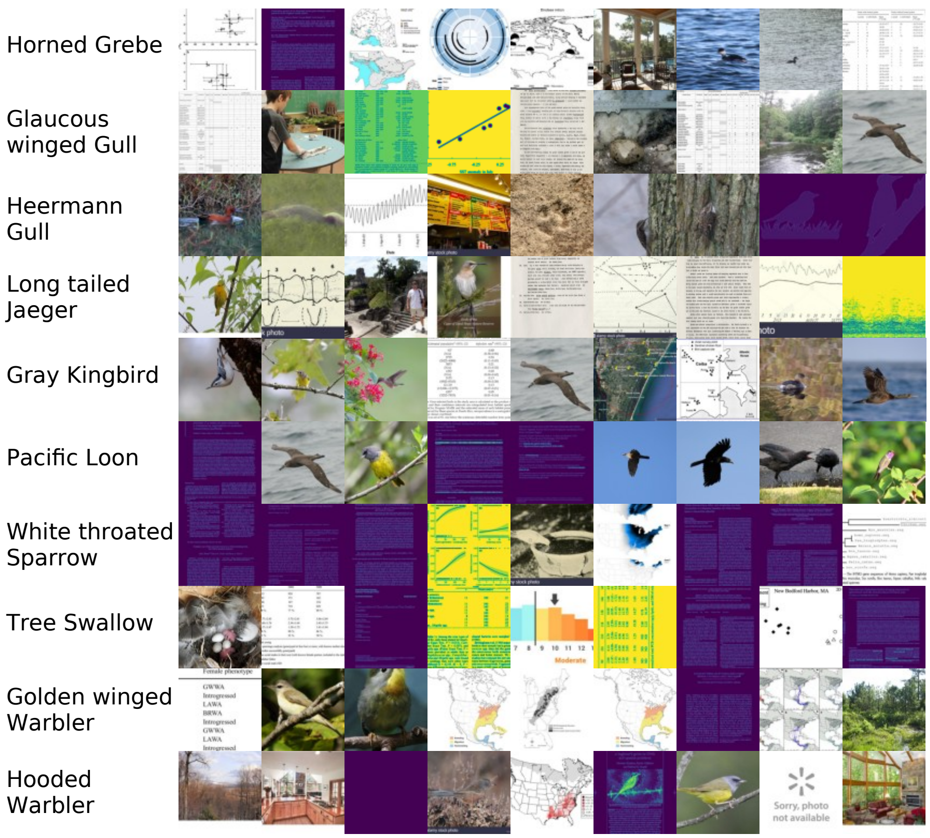

Fig. 8: Examples of detected noisy samples in Web-bird

Fig. 9: Examples of detected noisy samples in Web-aircraft

References

1. Ahn, C., Kim, K., Baek, J.w., Lim, J., Han, S.: Sample-wise Label Confidence Incorporation for Learning with Noisy Labels. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (2023)
2. Cherti, M., Beaumont, R., Wightman, R., Wortsman, M., Ilharco, G., Gordon, C., Schuhmann, C., Schmidt, L., Jitsev, J.: Reproducible scaling laws for contrastive language-image learning. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2023)
3. Cordeiro, F.R., Sachdeva, R., Belagiannis, V., Reid, I., Carneiro, G.: Longremix: Robust learning with high confidence samples in a noisy label environment. Pattern Recognition (2023)
4. Cubuk, E.D., Zoph, B., Shlens, J., Le, Q.V.: Randaugment: Practical automated data augmentation with a reduced search space. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW) (2020)
5. He, K., Zhang, X., Ren, S., Sun, J.: Identity mappings in deep residual networks. In: European Conference on Computer Vision (ECCV) (2016)
6. Huang, Z., Zhang, J., Shan, H.: Twin contrastive learning with noisy labels. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2023)
7. Iscen, A., Valmadre, J., Arnab, A., Schmid, C.: Learning With Neighbor Consistency for Noisy Labels. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2022)
8. Jiang, L., Huang, D., Liu, M., Yang, W.: Beyond Synthetic Noise: Deep Learning on Controlled Noisy Labels. In: International Conference on Machine Learning (ICML) (2020)
9. Kaiming, H., Xiangyu, Z., Shaoqing, R., Jian, S.: Deep Residual Learning for Image Recognition. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016)
10. Krizhevsky, A., Sutskever, I., Hinton, G.: Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems (NeurIPS) (2012)
11. Li, J., Socher, R., Hoi, S.: DivideMix: Learning with Noisy Labels as Semi-supervised Learning. In: International Conference on Learning Representations (ICLR) (2020)
12. Li, W., Wang, L., Li, W., Agustsson, E., Van Gool, L.: WebVision Database: Visual Learning and Understanding from Web Data. arXiv: 1708.02862 (2017)
13. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: International Conference on Machine Learning (ICML) (2021)
14. Sachdeva, R., Cordeiro, F.R., Belagiannis, V., Reid, I., Carneiro, G.: ScanMix: learning from severe label noise via semantic clustering and semi-supervised learning. Pattern Recognition (2023)
15. Sun, Z., Yao, Y., Wei, X.S., Zhang, Y., Shen, F., Wu, J., Zhang, J., Shen, H.T.: Webly Supervised Fine-Grained Recognition: Benchmark Datasets and An Approach. In: IEEE/CVF International Conference on Computer Vision (ICCV) (2021)
16. Szegedy, C., Ioffe, S., Vanhoucke, V., Alemi, A.: Inception-v4, inception-resnet and the impact of residual connections on learning. In: Association for the Advancement of Artificial Intelligence (AAAI) (2016)

17. Victor Guilherme Turrisi da Costa and Enrico Fini and Moin Nabi and Nicu Sebe and Elisa Ricci: solo-learn: A library of self-supervised methods for visual representation learning. Journal of Machine Learning Research (2022)
18. Vinyals, O., Blundell, C., Lillicrap, T., Kavukcuoglu, K., Wierstra, D.: Matching Networks for One Shot Learning. In: Advances in Neural Information Processing Systems (NeuRIPS) (2016)
19. Wang, H., Xiao, R., Dong, Y., Feng, L., Zhao, J.: Promix: Combating label noise via maximizing clean sample utility. In: International Joint Conference on Artificial Intelligence (IJCAI) (2023)
20. Wei, J., Zhu, Z., Cheng, H., Liu, T., Niu, G., Liu, Y.: Learning with Noisy Labels Revisited: A Study Using Real-World Human Annotations. In: International Conference on Learning Representations (ICLR) (2023)
21. Zhang, Z., Chen, W., Fang, C., Li, Z., Chen, L., Lin, L., Li, G.: RankMatch: Fostering Confidence and Consistency in Learning with Noisy Labels. In: IEEE/CVF International Conference on Computer Vision (ICCV) (2023)
22. Zheltonozhskii, E., Baskin, C., Mendelson, A., Bronstein, A.M., Litany, O.: Contrast to divide: Self-supervised pre-training for learning with noisy labels. In: IEEE/CVF Winter Conference on Applications of Computer Vision (WACV) (2022)