Paper ID #462

Semi-Supervised Domain Adaptation via Selective Pseudo Labeling and Progressive Self-Training

Yoonhyung Kim[1] and Changick Kim[1]

[1] Korea Advanced Institute of Science and Technology (KAIST), Republic of Korea

Introduction

- A fundamental limitation of deep convolutional neural network (DCNN)
  - ✓ Due to the strong dependency on training data, DCNN is fragile to domain shifts.

Introduction

- Definitions
  - ✓ **Image domain** indicates the contextual characteristic of an image set, which is affected by the process of image production.
  - ✓ **Domain shift** is statistical difference of data distributions between two domains.
    - It degrades the performance of DCNN.

- Naïve Solution to Overcome Domain Shifts
  - ✓ Preparing a large number of images & labels in the domain of interest.
    - Very costly and sometimes impossible.

- Alternative: Algorithmic Approach
  - ✓ To overcome domain shifts via domain adaptive learning schemes.
    - Called “Domain Adaptation” (DA).

Related Work

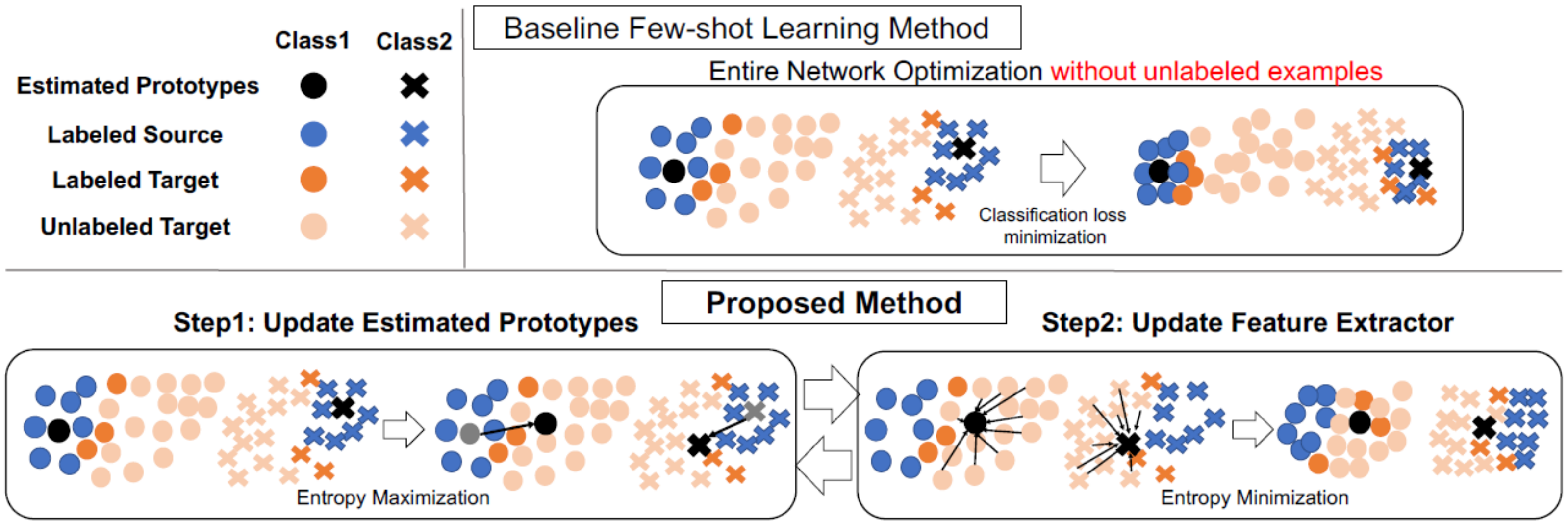
- Unsupervised Domain Adaptation (UDA) ← Majority of previous methods
  - ✓ Feature-level adaptation approach [5, 8, 9, 10, 12, 13].
  - ✓ Pixel-level adaptation approach [6, 7, 11].

- Limitations of UDA approaches
  - ✓ Discriminative power in target domain is not guaranteed.
  - ✓ Not robust to large domain shifts (e.g., shape variations).

Related Work

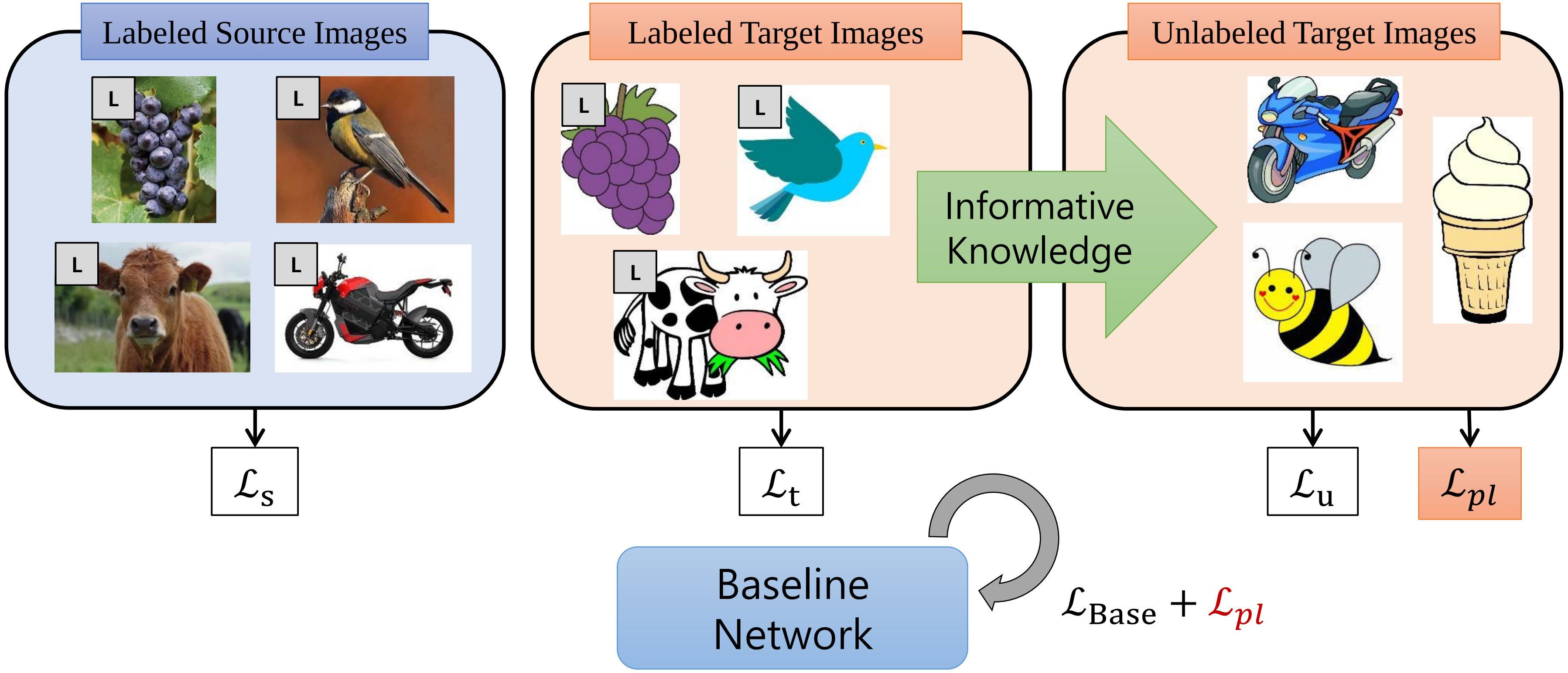
- Semi-Supervised Domain Adaptation (SSDA)
  - ✓ In comparison with UDA, a few labeled images are additionally given for training

- SSDA via Minimax Entropy (MME) [14]
  - ✓ Update to increase entropy w.r.t. Classifier
  - ✓ Update to decrease entropy w.r.t. Feature extractor



Proposed Method

- Motivation of the proposed SSDA method
  - ✓ Exploit the labeled target images to assign pseudo labels to unlabeled target images

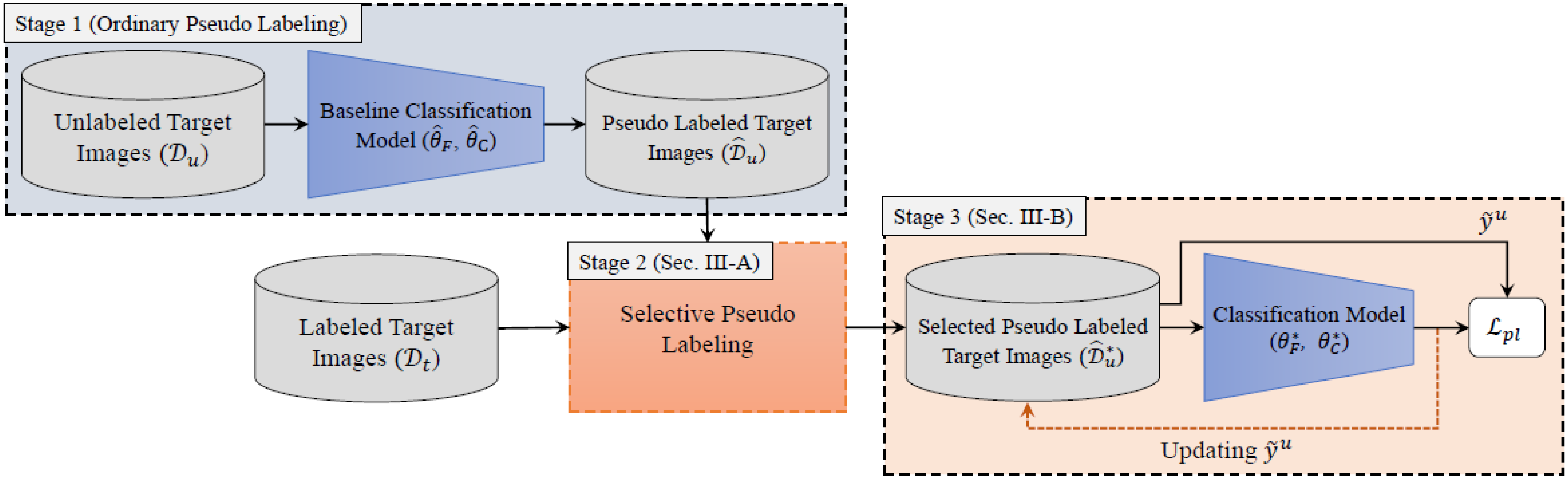


Proposed Method

- Step #1: Training a Baseline Network for Generating Pseudo Labels
  - ✓ MME [14] for baseline SSDA
  - ✓ Pseudo label is assigned to every unlabeled target image
- Step #2: Selective Pseudo Labeling
  - ✓ Selecting restricted amount of pseudo labels for high reliability
- Step #3: Label-Noise Robust Training via Progressive Self-Training
  - ✓ Pseudo labels are inevitably noisy
  - ✓ Apply a learning scheme that is robust to noisy labels

Proposed Method

- Overview of the proposed method

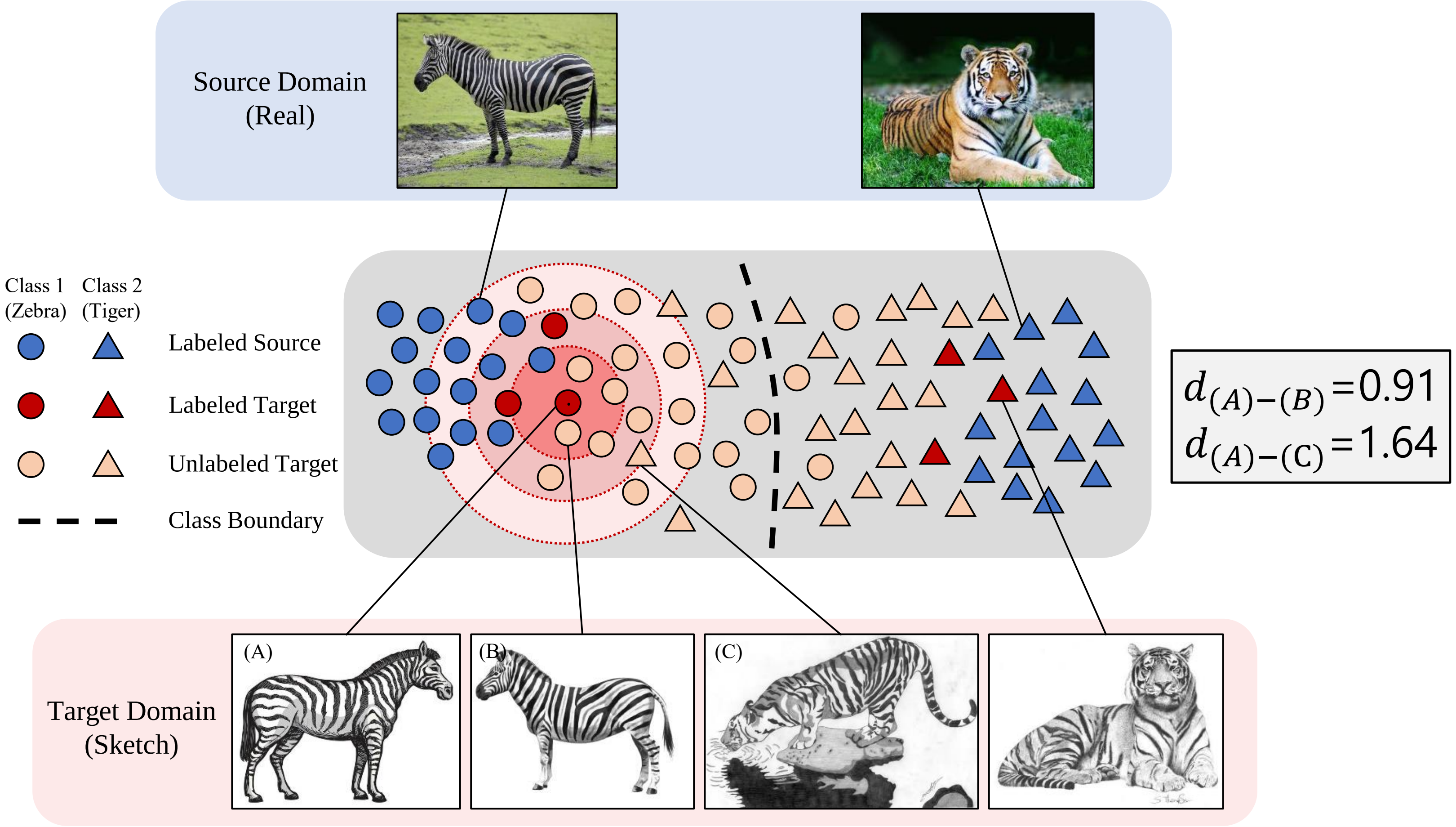


Proposed Method

- Step #1: Training a Baseline Network for Generating Pseudo Labels
  - ✓ Image sets
    - Labeled source images: $\mathcal{D}_s = \{(\mathbf{x}_i^s, y_i^s)\}_{i=1}^{n_s}$
    - Labeled target images: $\mathcal{D}_t = \{(\mathbf{x}_i^t, y_i^t)\}_{i=1}^{n_t}$
    - Unlabeled target images: $\mathcal{D}_u = \{\mathbf{x}_i^u\}_{i=1}^{n_u}$
  - ✓ Generate two kinds of pseudo label for $\mathbf{x}_i^u$
    - Hard label: $\hat{y}_\mathrm{i}^u = \operatorname*{argmin}_{k \in \{1,\ldots,K\}} p(y = k|\mathbf{x}_i^u)$
    - Soft label: $\tilde{\mathbf{y}}_\mathrm{i}^u = [p(y = 1|\mathbf{x}_i^u), \ldots, p(y = K|\mathbf{x}_i^u)]$
    - Pseudo labeled target images: $\widehat{\mathcal{D}}_u = \{(\mathbf{x}_i^u, \tilde{\mathbf{y}}_\mathrm{i}^u, \hat{y}_\mathrm{i}^u)\}_{i=1}^{n_u}$

Proposed Method

- Step #2: Selective Pseudo Labeling



Proposed Method

- Step #2: Selective Pseudo Labeling
  - ✓ Applied independently for each image category
  - ✓ For the j-th unlabeled image, a pairwise distance d_j is calculated

$$d_j = \frac{1}{n_t'} \sum_{i=1}^{n_t'} \left\| \mathbf{f}(\mathbf{x}_i^t) - \mathbf{f}(\mathbf{x}_j^u) \right\|_1$$

  - ✓ Sort the unlabeled images based on the magnitude of d_j, and determine $n_u^{'}$ samples as pseudo labels

$$n_u' = \left\lceil r_u \frac{n_u}{K} \right\rceil$$

  - ✓ $\mathrm{r_u} = 0.2$ as default
  - ✓ Pseudo labeled target images: $\widehat{\mathcal{D}}_u^* = \{(\mathbf{x}_i^u, \tilde{\mathbf{y}}_i^u, \hat{y}_i^u)\}_{i=1}^{n_u}$

Proposed Method

- Step #2: Selective Pseudo Labeling
  - ✓ Reliability of pseudo labels is enhanced after the selective pseudo labeling stage

Reliability of pseudo labels in terms of correctness [%]
"Before→After" the selective pseudo labeling stage

Net	Clipart to Sketch (C to S)		Painintg to Real (P to R)	
	1-shot	3-shot	1-shot	3-shot
AlexNet	35.2→**61.6**	41.0→**64.8**	57.7→**83.8**	60.7→**85.8**
VGG-16	51.2→**72.5**	54.6→**76.4**	72.2→**88.6**	75.0→**92.3**

Proposed Method

- Step #3: Label-Noise Robust Training via Progressive Self-Training
  - ✓ The selected pseudo labels are still not completely reliable (Noisy)
  - ✓ Apply a label noise-robust learning scheme

- Joint Optimization Framework for Learning with Noisy Labels [17]
  - ✓ Alternately updating 1) the network and 2) the noisy label set
  - ✓ Utilize the output prediction (soft labels) for training the network
  - ✓ Comparison of [17] and this work
    - [17]: Noisy labels are manually generated via simulation
    - This work: Directly employ noisy pseudo labels

Proposed Method

- Final Objective Function

Baseline (MME [14]) Proposed

$$\mathcal{L}_F = \boxed{\mathcal{L}_l + \lambda H} + \boxed{\mathcal{L}_{pl}}$$

$$\mathcal{L}_C = \boxed{\mathcal{L}_l - \lambda H} + \boxed{\mathcal{L}_{pl}}$$

$$\mathcal{L}_l = \mathbb{E}_{(\mathbf{x},y)\in\mathcal{D}_s,\mathcal{D}_t}\,\mathcal{L}_{ce}(\mathbf{p}(\mathbf{x}), y)$$

$$H = \mathbb{E}_{\mathbf{x}\in\mathcal{D}_\mathrm{u}}\sum_{i=1}^{K} p(y=i|\mathbf{x})\log(p(y=i|\mathbf{x}))$$

$\tilde{\mathbf{y}}^u$ is updated for every validation phase

$$\mathcal{L}_{pl} = \mathbb{E}_{(\mathbf{x},\tilde{\mathbf{y}})\in\widehat{\mathcal{D}}_u^*}\,\mathcal{L}_{ce}(\mathbf{p}(\mathbf{x}), \tilde{\mathbf{y}})$$

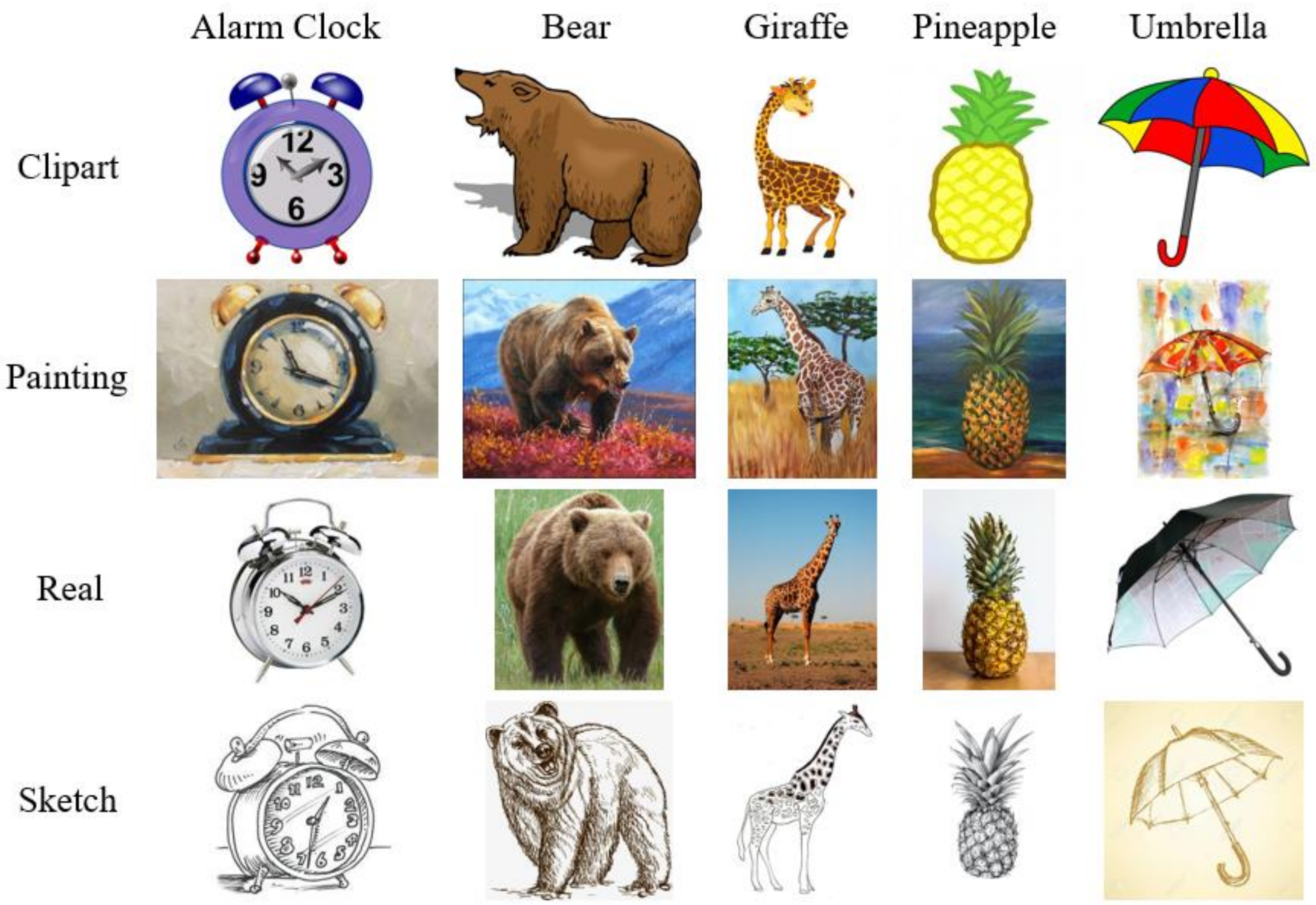
Experiments

- Datasets

-	**LSDAC [3]**	**Office-Home [18]**	**Office [19]**
Number of image categories	**126** classes	**65** classes	**31** classes
Domains	4 (Real, Clipart, Painting, Sketch)	4 (Real, Clipart, Art, Product)	3 (Amazon, Webcam, DSLR)
DA scenarios	**7**	**12**	**2**

Experiments

- Image samples in the LSDAC dataset [3]



Experiments

- Experimental Setups
  - ✓ Trained on $\mathcal{D}_s$, $\mathcal{D}_t$ (1-shot or 3-shot), and $\mathcal{D}_u$
  - ✓ Tested on: $\mathcal{D}_u$

- Baseline Networks
  - ✓ AlexNet [27], VGG-16 [28], and ResNet-34 [29]
  - ✓ To demonstrate the robustness across various network architectures

- Implementation
  - ✓ Framework: PyTorch [32]
  - ✓ GPU: A single NVIDIA Titan-X (Pascal Archit.)

Experiments

- DA Methods in Comparison
  - ✓ S+T
    - Trained on $\mathcal{D}_s$ and $\mathcal{D}_t$
  - ✓ DANN [4], ADR [20], CDAN [5], ENT [21]
    - UDA methods that are trained with the SSDA setup (+ $\mathcal{D}_t$)
  - ✓ MME [14]
    - The baseline SSDA method

Experiments

TABLE II
QUANTITATIVE EVALUATION RESULTS ON LSDAC DATASET IN TERMS OF ACCURACY (%).

Net	Method	R to C		R to P		P to C		C to S		S to P		R to S		P to R		MEAN	
		1-shot	3-shot	1-shot	3-shot	1-shot	3-shot	1-shot	3-shot	1-shot	3-shot	1-shot	3-shot	1-shot	3-shot	1-shot	3-shot
AlexNet	S+T	43.3	47.1	42.4	45.0	40.1	44.9	33.6	36.4	35.7	38.4	29.1	33.3	55.8	58.7	40.0	43.4
	DANN	43.3	46.1	41.6	43.8	39.1	41.0	35.9	36.5	36.9	38.9	32.5	33.4	53.6	57.3	40.4	42.4
	ADR	43.1	46.2	41.4	44.4	39.3	43.6	32.8	36.4	33.1	38.9	29.1	32.4	55.9	57.3	39.2	42.7
	CDAN	46.3	46.8	45.7	45.0	38.3	42.3	27.5	29.5	30.2	33.7	28.8	31.3	56.7	58.7	39.1	41.0
	ENT	37.0	45.5	35.6	42.6	26.8	40.4	18.9	31.1	15.1	29.6	18.0	29.6	52.2	60.0	29.1	39.8
	MME	48.9	55.6	48.0	49.0	46.7	51.7	36.3	39.4	39.4	43.0	33.3	37.9	56.8	60.7	44.2	48.2
	Proposed	**54.2**	**58.3**	**48.8**	**51.7**	**49.0**	**55.1**	**38.9**	**43.5**	**44.7**	**48.4**	**37.5**	**41.2**	**60.2**	**63.3**	**47.6**	**51.6**
VGG-16	S+T	49.0	52.3	55.4	56.7	47.7	51.0	43.9	48.5	50.8	55.1	37.9	45.0	69.0	71.7	50.5	54.3
	DANN	43.9	56.8	42.0	57.5	37.3	49.2	46.7	48.2	51.9	55.6	30.2	45.6	65.8	70.1	45.4	54.7
	ADR	48.3	50.2	54.6	56.1	47.3	51.5	44.0	49.0	50.7	53.5	38.6	44.7	67.6	70.9	50.2	53.7
	CDAN	57.8	58.1	57.8	59.1	51.0	57.4	42.5	47.2	51.2	54.5	42.6	49.3	71.7	74.6	53.5	57.2
	ENT	39.6	50.3	43.9	54.6	26.4	47.4	27.0	41.9	29.1	51.0	19.3	39.7	68.2	72.5	36.2	51.1
	MME	60.6	64.1	63.3	63.5	57.0	60.7	50.9	55.4	60.5	60.9	50.2	54.8	72.2	**75.3**	59.2	62.1
	Proposed	**64.5**	**68.0**	**63.7**	**64.9**	**60.5**	**64.4**	**53.7**	**57.4**	**62.5**	**63.4**	**52.7**	**57.5**	**73.0**	74.9	**61.5**	**64.4**
ResNet-34	S+T	55.6	60.0	60.6	62.2	56.8	59.4	50.8	55.0	56.0	59.5	46.3	50.1	71.8	73.9	56.9	60.0
	DANN	58.2	59.8	61.4	62.8	56.3	59.6	52.8	55.4	57.4	59.9	52.2	54.9	70.3	72.2	58.4	60.7
	ADR	57.1	60.7	61.3	61.9	57.0	60.7	51.0	54.4	56.0	59.9	49.0	51.1	72.0	74.2	57.6	60.4
	CDAN	65.0	69.0	64.9	67.3	63.7	68.4	53.1	57.8	63.4	65.3	54.5	59.0	73.2	78.5	62.5	66.5
	ENT	65.2	71.0	65.9	69.2	65.4	71.1	54.6	60.0	59.7	62.1	52.1	61.1	75.0	78.6	62.6	67.6
	MME	70.0	72.2	67.7	69.7	69.0	71.7	56.3	61.8	64.8	66.8	61.0	61.9	76.1	78.5	66.4	68.9
	Proposed	**72.4**	**73.9**	**69.4**	**71.5**	**71.6**	**73.9**	**61.7**	**63.3**	**66.7**	**69.0**	**62.5**	**65.1**	**78.8**	**80.4**	**69.0**	**71.0**

Experiments

- Ablation studies
  - ✓ Ratio of selecting pseudo labels (r_u)
    - Selecting moderate amounts of pseudo labels is encouraged

r_u	0.01	0.05	0.20	0.50	1.00
C to S	41.7	42.8	**43.5**	43.1	42.4
P to R	60.5	62.1	**63.3**	62.9	62.0

  - ✓ ‘With’ vs. ‘Without’ applying label noise-robust learning scheme
    - ‘Without’ indicates: Training with fixed hard pseudo labels

Whether applied	C to S		P to R	
	1-shot	3-shot	1-shot	3-shot
Yes	**38.9**	**43.5**	**60.2**	**63.3**
No	37.7	42.1	58.3	61.9

Conclusion

- In this paper, we have introduced a novel SSDA method for image classification.
- The key idea is to exploit labeled target images to find out reliable pseudo labels for the unlabeled target images.
- The proposed SSDA method outperforms previous state-of-the-art method by 2.7%, 0.9%, and 2.2% for LSDAC, Office-Home, and Office datasets, respectively.

Acknowledgement

- This work was supported by Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. 2019-0-00524, Development of precise content identification technology based on relationship analysis for maritime vessel/structure).

References [1/4]

[1] S. J. Pan and Q. Yang, “A survey on transfer learning,” IEEE Transactions on knowledge and data engineering, vol. 22, no. 10, pp. 1345–1359, 2009.

[2] M. Wang and W. Deng, “Deep visual domain adaptation: A survey,” Neurocomputing, vol. 312, pp. 135–153, 2018.

[3] X. Peng, Q. Bai, X. Xia, Z. Huang, K. Saenko, and B. Wang, “Moment matching for multi-source domain adaptation,” in Proc. IEEE International Conference on Computer Vision (ICCV), 2019, pp. 1406–1415.

[4] Y. Ganin and V. Lempitsky, “Unsupervised domain adaptation by backpropagation,” in Proc. International Conference on Machine Learning (ICML), 2015.

[5] M. Long, Z. Cao, J. Wang, and M. Jordan, “Conditional adversarial domain adaptation,” in Proc. Neural Information Processing Systems (NeurIPS), 2018.

[6] R. Volpi, P. Morerio, S. Savarese, and V. Murino, “Adversarial feature augmentation for unsupervised domain adaptation,” in Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018.

[7] L. Hu, M. Kan, S. Shan, and X. Chen, “Duplex generative adversarial network for unsupervised domain adaptation,” in Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018.

[8] K. Saito, K. Watanabe, Y. Ushiku, and T. Harada, “Maximum classifier discrepancy for unsupervised domain adaptation,” in Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018.

[9] G. French, M. Mackiewicz, and M. Fisher, “Self ensembling for visual domain adaptation,” arXiv preprint arXiv:1706.05208, 2017.

[10] V. Kurmi, S. Kumar, and V. Namboodiri, “Attending to discriminative certainty for domain adaptation,” in Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018.

References [2/4]

[11] R. Gong, W. Li, Y. Chen, and L. Gool, “DLOW: Domain flow for adaptation and generalization,” in Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2019.

[12] X. Ma, T. Zhang, and C. Xu, “GCAN: Graph convolutional adversarial network for unsupervised domain adaptation,” in Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2019.

[13] J. Choi, M. Jeong, T. Kim, and C. Kim, “Pseudo-labeling curriculum for unsupervised domain adaptation,” in Proc. British Machine Vision Conference (BMVC), 2019.

[14] K. Saito, D. Kim, S. Sclaroff, T. Darrell, and K. Saenko, “Semisupervised domain adaptation via minimax entropy,” in Proc. IEEE Conference on International Conference on Computer Vision (ICCV), 2019.

[15] W.-Y. Chen, Y.-C. Liu, Z. Kira, Y.-C. F. Wang, and J.-B. Huang, “A closer look at few-shot classification,” arXiv preprint arXiv:1904.04232, 2019.

[16] D.-H. Lee, “Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks,” in Workshop on challenges in representation learning (ICML), vol. 3, 2013, p. 2.

[17] D. Tanaka, D. Ikami, T. Yamasaki, and K. Aizawa, “Joint optimization framework for learning with noisy labels,” in Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018, pp. 5552–5560.

[18] H. Venkateswara, J. Eusebio, S. Chakraborty, and S. Panchanathan, “Deep hashing network for unsupervised domain adaptation,” in Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 5018–5027.

[19] K. Saenko, B. Kulis, M. Fritz, and T. Darrell, “Adapting visual category models to new domains,” in Proc. European Conference on Computer Vision (ECCV). Springer, 2010, pp. 213–226.

[20] K. Saito, Y. Ushiku, T. Harada, and K. Saenko, “Adversarial dropout regularization,” arXiv preprint arXiv:1711.01575, 2017.

References [3/4]

[21] Y. Grandvalet and Y. Bengio, "Semi-supervised learning by entropy minimization," in Proc. Advances in neural information processing systems (NeurIPS), 2005, pp. 529–536.

[22] Y. Kim, J. Yim, J. Yun, and J. Kim, "NLNL: Negative learning for noisy labels," in Proc. IEEE International Conference on Computer Vision (ICCV), 2019, pp. 101–110.

[23] A. Ghosh, H. Kumar, and P. Sastry, "Robust loss functions under label noise for deep neural networks," in Proc. Thirty-First AAAI Conference on Artificial Intelligence, 2017.

[24] C. G. Northcutt, T. Wu, and I. L. Chuang, "Learning with confident examples: Rank pruning for robust classification with noisy labels," arXiv preprint arXiv:1705.01936, 2017.

[25] Z. Zhang and M. Sabuncu, "Generalized cross entropy loss for training deep neural networks with noisy labels," in Proc. Advances in Neural Information Processing Systems (NeurIPS), 2018, pp. 8778–8788.

[26] A. Vahdat, "Toward robustness against label noise in training deep discriminative neural networks," in Proc. Advances in Neural Information Processing Systems (NeurIPS), 2017, pp. 5596–5605.

[27] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in Proc. Neural Information Processing Systems (NeurIPS), 2012.

[28] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," arXiv preprint arXiv:1409.1556, 2014.

[29] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770–778.

[30] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 4700–4708.

References [4/4]

[31] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018, pp. 4510–4520.

[32] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in pytorch," 2017.

[33] R. Ranjan, C. D. Castillo, and R. Chellappa, "L2-constrained softmax loss for discriminative face verification," arXiv preprint arXiv:1703.09507, 2017.

[34] Y. Chen, W. Li, C. Sakaridis, D. Dai, and L. Van Gool, "Domain adaptive faster r-cnn for object detection in the wild," in Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018, pp. 3339–3348.

[35] J. Hoffman, E. Tzeng, T. Park, J.-Y. Zhu, P. Isola, K. Saenko, A. A. Efros, and T. Darrell, "CYCADA: Cycle-consistent adversarial domain adaptation," arXiv preprint arXiv:1711.03213, 2017.

Contact Information

- Yoonhyung Kim: yhkim1127@kaist.ac.kr
  - ✓ First author

- Changick Kim: changick@kaist.ac.kr
  - ✓ Corresponding author