Proxy Anchor-based Unsupervised Learning for Continuous Generalized Category Discovery

Hyungmin Kim[1,2] Sungho Suh[3] Daehwan Kim[2] Daun Jeong[2] Hansang Cho[2]
Junmo Kim[1]
[1]Korea Advanced Institute of Science and Technology, Daejeon, Korea
[2]Samsung Electro-Mechanics, Suwon, Korea
[3]German Research Center for Artificial Intelligence, Kaiserslautern, Germany
{hyungmin83, junmo.kim}@kaist.ac.kr, sungho.suh@dfki.de
{daehwan85.kim, du33.jeong, hansang.cho}@samsung.com

Algorithm A Pseudo-code of the one-step incremental novel category discovering

```
Initial step:
    Given labeled dataset $\mathcal{D}^0 = \{(x, y)\}$
    Train network $f^0(\cdot)$ and proxy anchors $g^0(\cdot)$
    Calculate $\sigma$ of $\mathcal{D}^0$ for exemplar $\mathcal{E}^0$
Discovering novel category step:
    Given unlabeled joint dataset $\mathcal{D}^1 = \{x\}$
    Extract embedding vectors $z_i$ on $f^0(x_i)$
    Get initial separated datasets on measuring cosine similarity score $s(z_i, p)$ using the initial split
    Get fine separated datasets, $\mathcal{D}^1_{old} = \{x^1_{old}\}$ and $\mathcal{D}^1_{new} = \{x^1_{new}\}$ on Noisy labeling and Gaussian mixture model using the initial separated dataset
    Get pseudo-labels $\mathcal{D}^1_{old}$ using previous network $f^0$, $\hat{\mathcal{D}}^1_{old} = \{(x^1_{old}, \hat{y}^1_{old} = \text{argmax}(f^0(x^1_{old}))\}$
    Get pseudo-labels $\mathcal{D}^1_{new}$ using a non-parametric clustering approach $c(\cdot)$, affinity propagation, $\hat{\mathcal{D}}^1_{new} = \{(x^1_{new}, \hat{y}^1_{new} = \text{argmax}(c(x^1_{new}))\}$
Category incremental step:
    Add new proxy anchors as the estimated number of novel categories based on $c(\cdot)$ results
    Assign initial means of newly added proxy anchors based on $c(\cdot)$ results
    Generate old embedding vectors $\tilde{z}^0$ using $\mathcal{E}^0$
    Train network $f^1(\cdot)$ and modified Proxy anchors $g^1(\cdot)$ using pseudo-labeled dataset $\hat{\mathcal{D}}^1$ and $\tilde{z}^0$
    Distill knowledge between networks, $f^0(\cdot)$ and $f^1(\cdot)$
    Calculation $\sigma$ of $\hat{\mathcal{D}}^1$ for exemplar $\mathcal{E}^1$
```

A. Pseudo Code

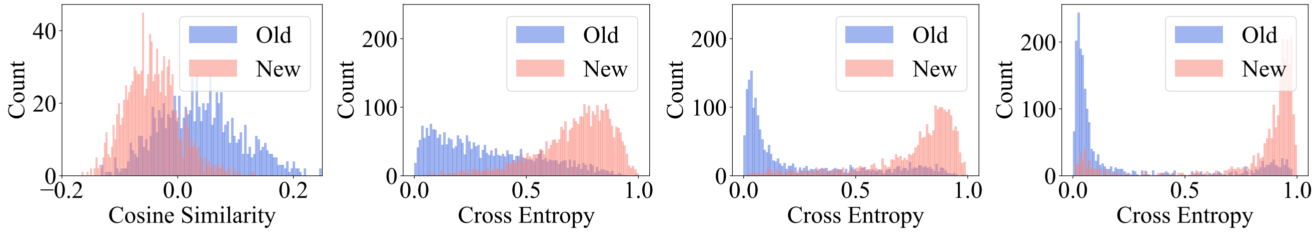
Pseudo-code of our proposed method is represented in Algorithm A in detail. The code is written for the one-time step procedure and is comprised of three steps: the initial step, the discovering novel category step, and the category incremental step. The initial step is fine-tuning the network on the dataset and training proxy anchors using the labeled dataset. Then, the following given dataset for incremental category learning is the unlabeled joint set, which includes the old and novel classes. In the discovering novel category step, we separate the dataset into old and novel categories using the initial split and the fine split, and pseudo-label the separated datasets. In the last step, exploiting the pseudo-labeled dataset, we add new proxy anchors and fine-tune the network and proxy anchors. Alleviating the catastrophic forgetting, we utilize proxy anchor-based exemplar and feature distillation.

For extending to continuously incremental novel categories, the initial step is trained only once, then the discovering novel category step and the category incremental step are trained iteratively and sequentially. Specifically, let us assume that the novel category discovery continually increases until the n^{th} step. In the discovering novel category step, the given dataset notation is changed from $\mathcal{D}^1$ to $\mathcal{D}^n$, and the previous network f^0 is also replaced to f^{n-1}. Also, the pseudo-labeled dataset $\hat{\mathcal{D}^1}$ is modified to $\hat{\mathcal{D}^n}$. In the category incremental step, the generated vector $\tilde{z}^0$, the exploited exemplar $\mathcal{E}^0$ for the vector, and the pseudo-labeled dataset $\hat{\mathcal{D}^1}$ are notated $\tilde{z}^{n-1}$, $\mathcal{E}^{n-1}$, and $\hat{\mathcal{D}^n}$, respectively. $f^1(\cdot)$ and $g^1(\cdot)$ are replaced $f^n(\cdot)$ and $g^n(\cdot)$. Lastly, the new exemplar $\mathcal{E}^1$ is substituted with $\mathcal{E}^n$.

Our code is made available at https://github.com/Hy2MK/CGCD.

B. Fine Split

To acquire a more clearly separated dataset without noisy data, we design a simple network for the fine split. The net-



(a) Initial Split (b) Fine Split: the first epoch (c) Fine Split: the second epoch (d) Fine Split: the last epoch

Figure A. The initial split and the fine split results using CUB-200 dataset on ResNet-18. The given dataset is joint unlabeled, and we separate the dataset into old and novel categories without any prior knowledge

Method	Step	Novel Classes	$\mathcal{M}_{all} \uparrow$	$\mathcal{M}_o^0 \uparrow$	$\mathcal{M}_f \downarrow$	$\mathcal{M}_n^1 \uparrow$	$\mathcal{M}_n^2 \uparrow$	$\mathcal{M}_n^3 \uparrow$	$\mathcal{M}_d \uparrow$
GM-CI [3]	0th	$0 \sim 139$	59.51	59.51	-	-	-	-	-
	1st	$140 \sim 159$	13.55	13.48	46.03	13.99	-	-	13.99
	2nd	$160 \sim 179$	37.32	41.36	18.15	25.43	21.62	-	23.53
	3rd	$180 \sim 199$	35.74	40.44	19.07	27.13	19.93	28.06	25.04
	$*$		-	-	19.48	-	-	-	24.97
	0th	$0 \sim 159$	60.34	60.34	-	-	-	-	-
	1st	$160 \sim 199$	7.13	7.28	53.06	6.53	-	-	6.53
GM-MI [3]	0th	$0 \sim 139$	26.54	26.54	-	-	-	-	-
	1st	$140 \sim 159$	18.70	20.48	6.06	7.51	-	-	7.51
	2nd	$160 \sim 179$	19.90	23.24	3.30	6.83	10.14	-	8.49
	3rd	$180 \sim 199$	17.22	20.80	5.74	6.66	8.28	12.24	6.89
	0th	$0 \sim 159$	46.39	46.39	-	-	-	-	-
	1st	$160 \sim 199$	**6.43**	**6.57**	**39.82**	**5.92**	-	-	**5.92**
Ours-CGCD	0th	$0 \sim 159$	74.27	74.27	-	-	-	-	-
	1st	$160 \sim 200$	**54.75**	**58.80**	**15.47**	**40.90**	-	-	**40.90**

Table A. Comparison of GM [3] method evaluation results on the two different scenarios, which are Class incremental scenario (CI) and Mixed Incremental Scenario (MI) are proposed originally in GM paper. The experiment results exploited the CUB-200 dataset on ResNet-18. For a fair comparison, we followed the hyperparameters on their report. Nevertheless, GM is underperformed significantly. $*$ denotes results reported in the original GM paper and the bold results indicated the reported in the main paper.

work comprises a series of the Fully Connected layer (FC) - Batch Normalization (BN) - sigmoid - FC - BN - sigmoid - FC. The data is selected on both ends of the spectrum for training the network since we assume the data in the region is clean (*i.e.* lower than 5% and over 95% based on the result of GMM). The simple perceptron model aims to predict whether the noisy data belongs to the old or novel categories and is trained on three epochs. Figure A (a) is depicted the initial split result using cosine similarity score measurement between the embedding vectors and proxy anchors. There is an overlapped region between the old and novel categories, which represents the initial split using previous knowledge is unclear to divide novel and old categories. Therefore, we adopt the noisy labeling scheme to fine separation into old and novel classes, and confirmed the results of every epoch of the fine split from Figure A (b) to Figure A (d).

C. GM Results Analysis

In the paper, we conducted the comparison experiment of our proposed method with state-of-the-art approaches. Among the compared methods, GM [3] is recorded as significantly underperforming. Therefore, we should confirm to clear that our experiments are conducted in a fair comparison following the hyperparameters reported in the original paper without any modifications.

GM proposed four different scenarios, which are Class Incremental scenario (CI), Data Incremental Scenario (DI), Mixed Incremental scenario (MI), and Semi-supervised Mixed Incremental Scenario (SMI). Among the scenarios, CI and MI are the most similar to ours, Continuous Generalized Category Discovery (CGCD). We evaluated these two scenarios using CUB-200 dataset on ResNet-18. As presented in Table A, GM conducted experiments using fine-grained datasets only in the CI scenario, and in the other scenario, the CIFAR-100 dataset was utilized to evaluate the

Method	Dataset	1^{st}		2^{nd}		3^{rd}	
		$\mathcal{M}_f \downarrow$	$\mathcal{M}_d \uparrow$	$\mathcal{M}_f \downarrow$	$\mathcal{M}_d \uparrow$	$\mathcal{M}_f \downarrow$	$\mathcal{M}_d \uparrow$
GM	CUB-200	30.99	12.21	19.86	15.99	21.18	16.56

Table B. GM step-wise results following GM's original process.

Method	CUB-200			
	$\mathcal{M}_{all} \downarrow$	$\mathcal{M}_o \uparrow$	$\mathcal{M}_f \downarrow$	$\mathcal{M}_d \uparrow$
ArcFace [1]	53.13±2.76	60.21±3.80	13.52±3.81	28.02±1.39
ProxyNCA++ [2]	53.72±0.85	**65.62±0.52**	**8.85±0.52**	25.00±1.77
Ours	**54.75±0.64**	58.80±0.99	15.47±0.99	**40.90±1.07**

Table C. Ablation study for adopting various deep metric learning.

performance of the method. In this sense, we confirmed the reproducibility in the CI scenario using CUB-200 dataset since the $\mathcal{M}_d$ and $\mathcal{M}_f$ results were equivalent. The MI scenario is the closest to our proposed scenario CGCD. However, GM presented significant underperformance in the MI scenarios, and $\mathcal{M}_d$ performances were recorded at under 10%, particularly in both three-time incremental and one-time incremental scenarios.

On the other hand, we confirmed outstanding performance compared to GM in this experiment, and our proposed method also recorded improved performance in the two-step incremental scenario reported in the paper.

Following the dataset policy of GM, their given distribution is $7:1:1:1$, and the results using the original parameters reported in the paper are shown in Table B. As the step increases, $\mathcal{M}_f$ and $\mathcal{M}_d$ recorded mean values of 24.01 and 14.92, respectively. But, in our ablations, we evaluated the policy, $7:3$. This means that we classify into joint unlabeled novel data more than three times as GM at once. Through experiments, we implicitly showed superior performance.

D. Adopting other deep metric learning

Table C shows the results of replacing PAs with others. [1, 2]. We assumed PAs is trained in more valuable features for utilizing to split novel and old category since having both proxy- and anchor-based merits. As well-separated novel data samples increased, the result showed $\mathcal{M}_f$ decreased, but $\mathcal{M}_d$ improved. And we confirmed PAs is to fit our framework.

E. Ours Qualitative Results

To evaluate the proposed method qualitatively, we clustered the evaluation dataset using the CUB-200 dataset. As shown in Figure B, our method well-discovered old categories and clustered them correctly. Each row is clustered into the same category, and the classes are old categories on the evaluation dataset. The left five columns are well-clustered, while the last two are not. The sixth-column images are still reasonable, but the last-column images are the worst cases. Figure C depicted the clustered results of the novel categories discovery. Like the former result, each row image belongs to the same category. In contrast, the Figure D presents failure cases. The images on each row indicated that they were clustered into the same class. However, there are no images with the same label. Nevertheless, the images on each row have similar features, such as the colors of wings and feathers, and the behaviors. In this sense, we are hard to recognize that the categorized results failed without the specialized knowledge of the birds' species.

References

[1] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin for deep face recognition. In *CVPR*, 2019. 3

[2] Eu Wern Teh, Terrance DeVries, and Graham W Taylor. Proxynca++: Revisiting and revitalizing proxy neighborhood component analysis. In *ECCV*, 2020. 3

[3] Xinwei Zhang, Jianwen Jiang, Yutong Feng, Zhi-Fan Wu, Xibin Zhao, Hai Wan, Mingqian Tang, Rong Jin, and Yue Gao. Grow and merge: A unified framework for continuous categories discovery. In *NeurIPS*, 2022. 2

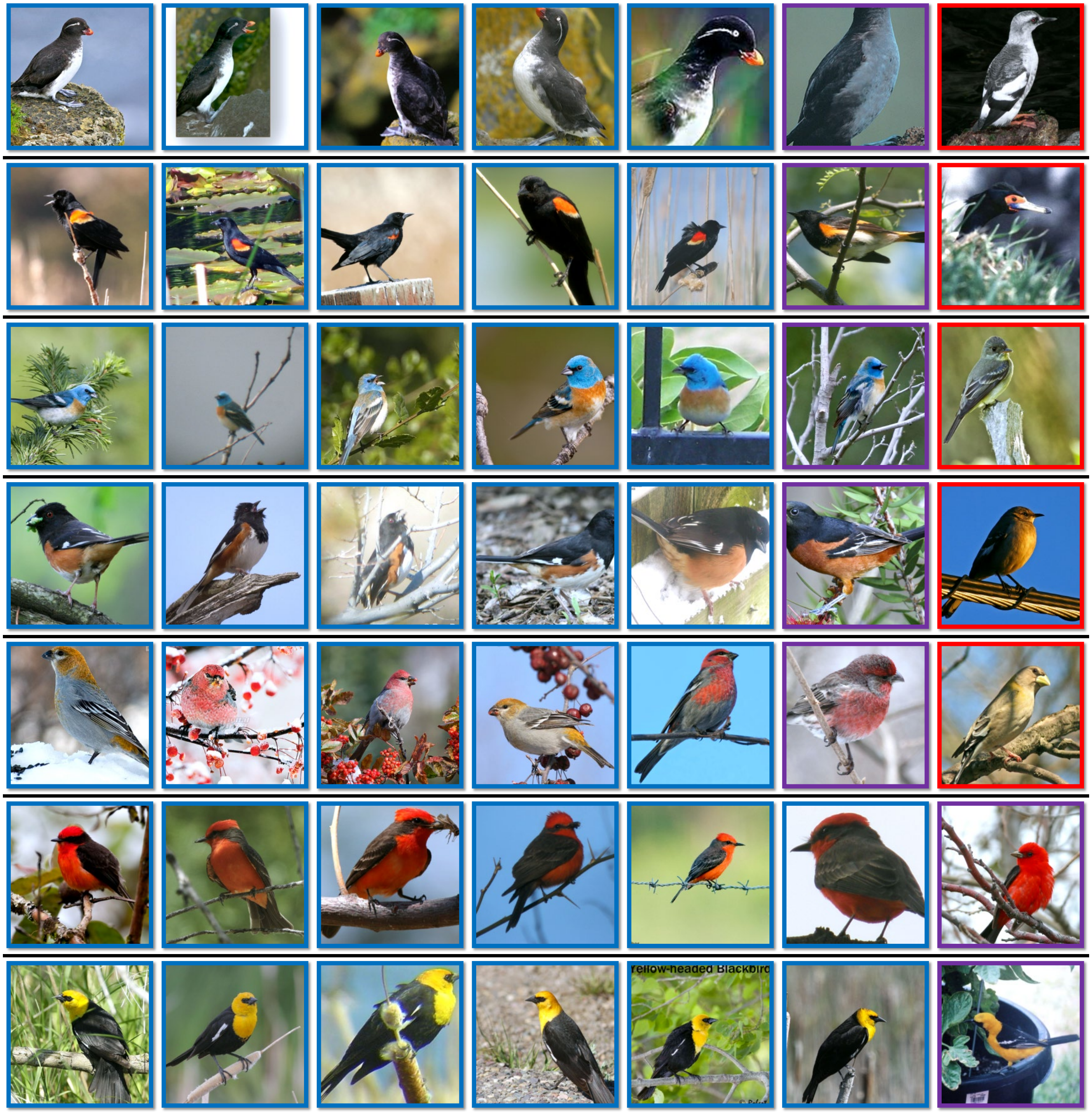


Figure B. Qualitative experiment results of the proposed method. The evaluation is conducted using the CUB-200 dataset on ResNet-18, and the images belong to the old categories and are also clustered in the old categories. The first five columns with blue boxes denote well-clustered examples. The last two columns represent failed prediction results, including example images with purple boxes denoting hard negatives and those with red boxes indicating incorrect categorization.

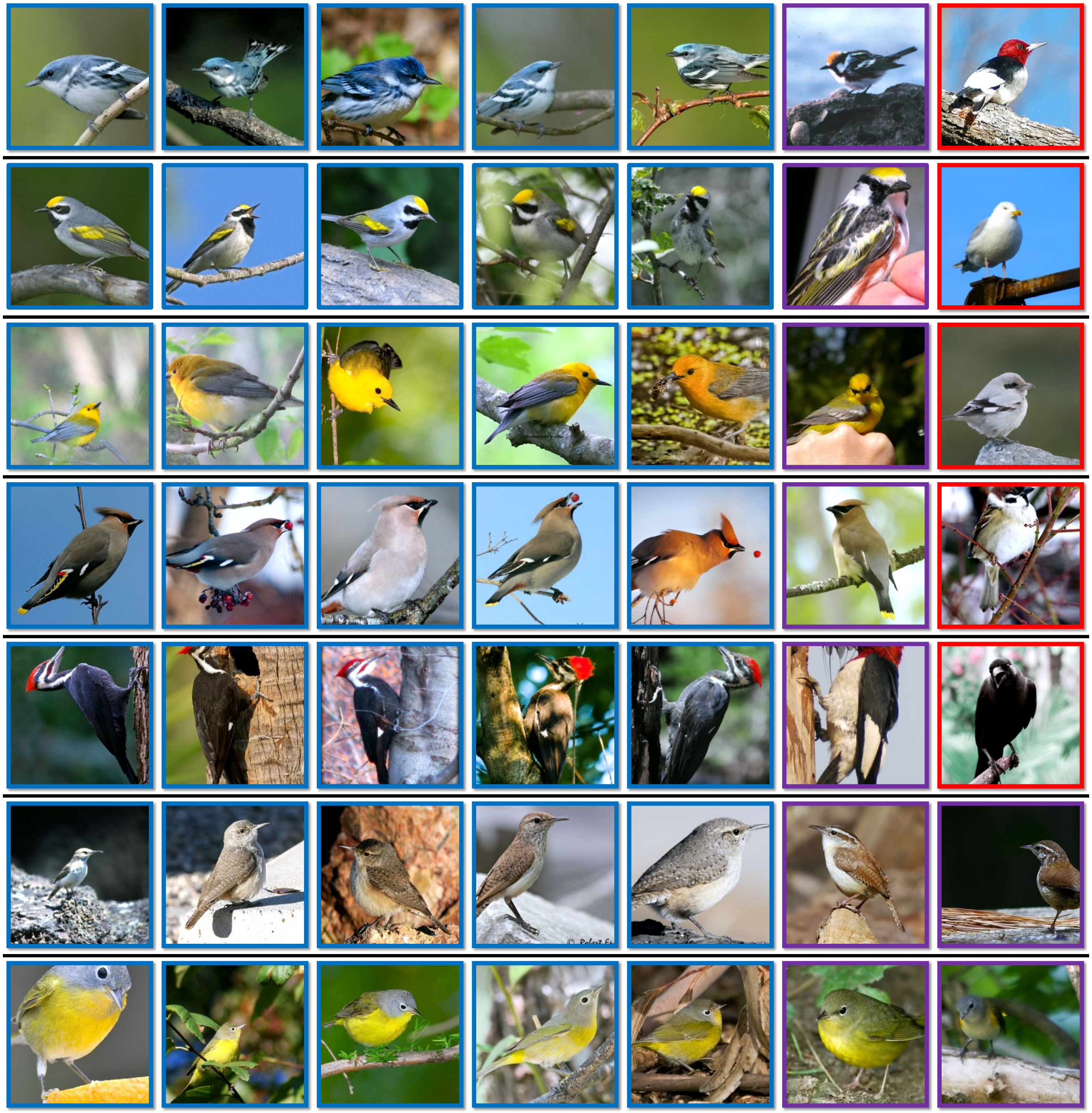


Figure C. Qualitative experiment results of the proposed method. The evaluation is conducted using the CUB-200 dataset on ResNet-18, and the images belong to the novel categories and are also clustered in the novel categories. The first five columns with blue boxes denote well-clustered examples. The last two columns represent failed prediction results, including example images with purple boxes denoting hard negatives and those with red boxes indicating incorrect categorization.

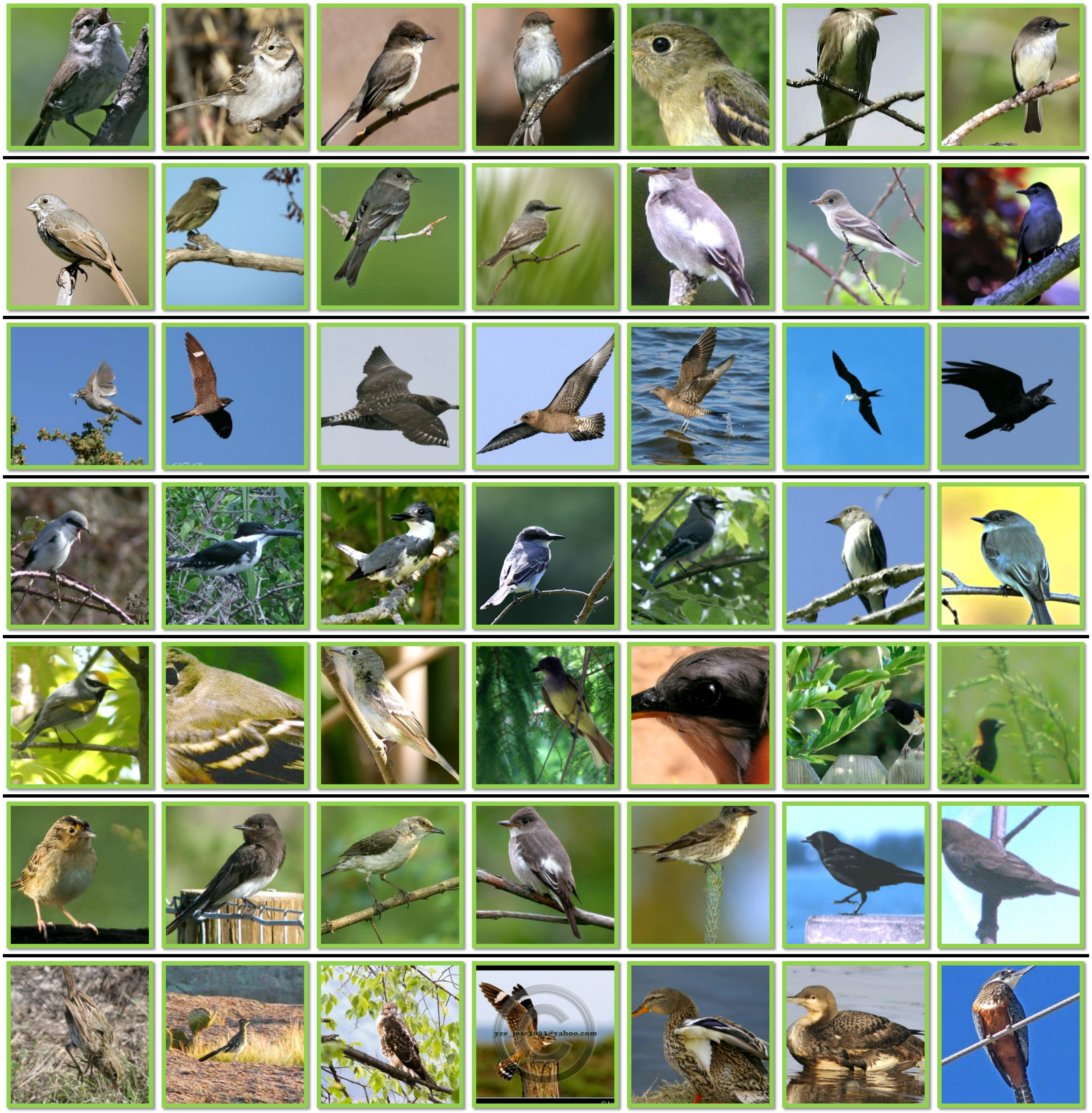


Figure D. Qualitative experiment results of the proposed method. The evaluation is conducted using the CUB-200 dataset on ResNet-18. The images on each row have been categorized into the same classes, but it is not true. Nevertheless, the images on each row are hard to recognize the specific spices without the knowledge of experts. For that reason, we treat the images are hard negative samples.