

Out-of-Distribution Detection for Long-tailed and Fine-grained Skin Lesion Images

MICCAI 2022

Deval Mehta, Yaniv Gal, Adrian Bowling, Paul Bonnington,
and Zongyuan Ge

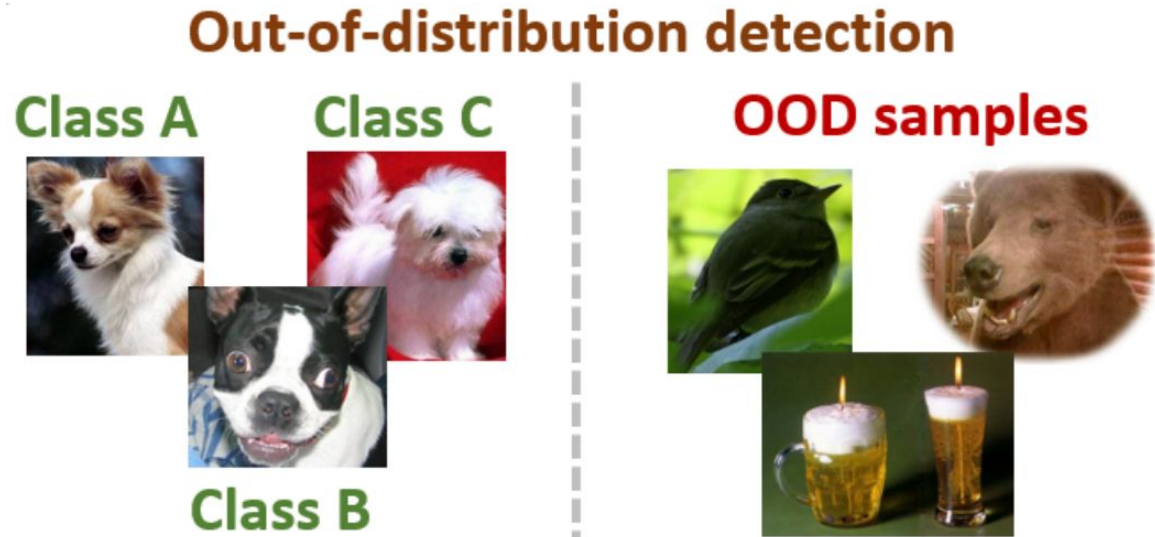
Reporter: Yu-Chen Lai

Date: 2022/07/28

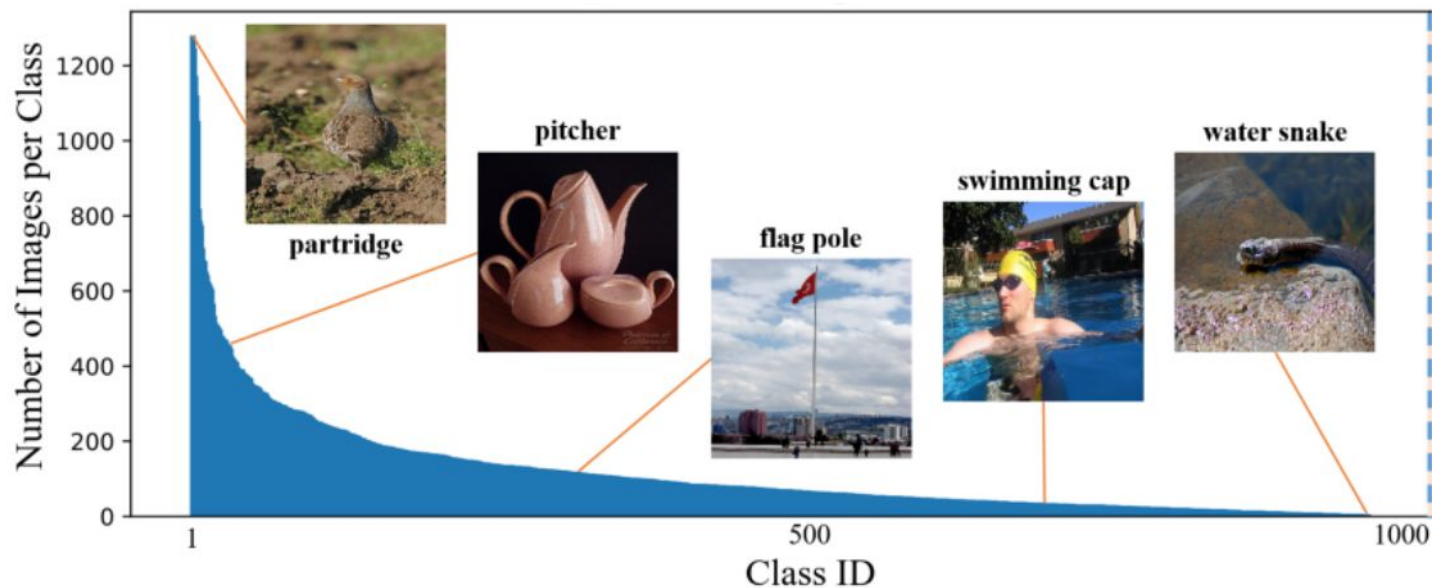
Outline

- Introduction
 - OOD, long-tailed, fine-grained
 - Motivation
- Proposed Method
 - Inter-subset Mixup Strategies
 - Integration of Mixup with Prototype Learning
- Convolutional Prototype Learning
 - Architecture of the framework
 - Loss function
- Experimental Results
 - Dataset Settings and Evaluation Metrics
 - Ablation Study
 - Benchmarking with other methods
 - Confidence Scores Visualization

Out-of-Distribution Detection (OOD)



Long-tailed Images



Fine-grained Images

Rusty
Blackbird



Brewer
Blackbird



Red Winged
Blackbird



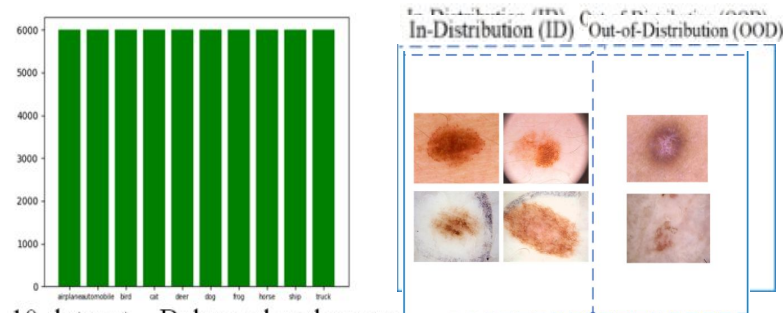
Dataset: CUB200

Motivation

- OOD samples
 - unknown skin conditions
 - hardware device variations
 - different clinical settings
- Different problem setting

- commonly used datasets (A) v.s clinical deployment purposes

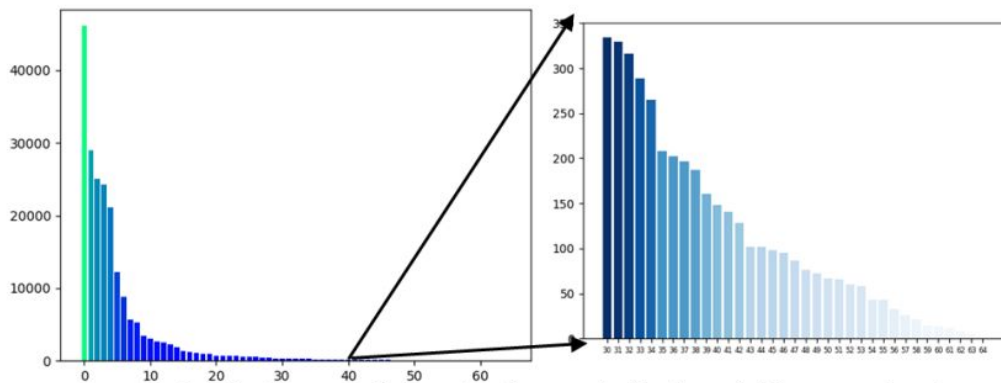
- (A) is **well balanced** and **coarse-grained**
- (A) is **not long-tailed**



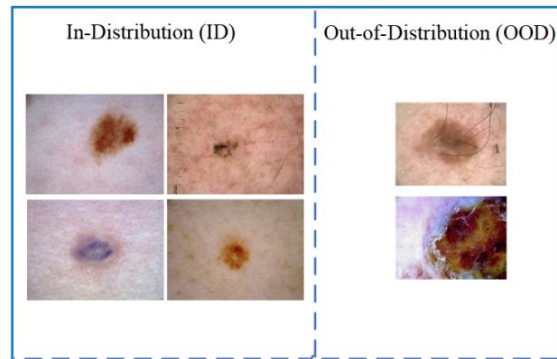
(c) CIFAR-10 dataset – Balanced and coarse-grained
(10 balanced categories with visually distinguishing features)
(8 categories with high invariance and having visually similar features)

Motivation

- The setting for real-world application scenario
 - **fine-grained** categories
 - **long-tailed** distribution



(a) In-house dataset – Long-tailed and Fine-grained
(65 categories with high imbalance and having visually similar features)



Proposed Method

- Although random **oversampling/undersampling** and techniques like SMOTE [3] can be used to tackle this problem, repeating/removing samples of classes **does not help** the classifier **learn any better decision boundaries**.

Proposed Method

- Although random **oversampling/undersampling** and techniques like SMOTE [3] can be used to tackle this problem, repeating/removing samples of classes **does not help** the classifier **learn any better decision boundaries**.
- Our proposed approach employs a combination of **data augmentation** using **mixup** and **better feature space learning** using **prototype loss** specifically targeted to middle and tail classes.
- This enables us to improve the classification performance for those **middle** and **tail** categories which also increases the OOD detection performance.

Proposed Method

- Inter-subset mixup strategy
 - Target the middle and tail classes
- Convolutional Prototype learning
 - Tackle the fine-grained aspect

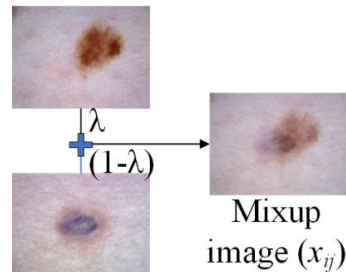
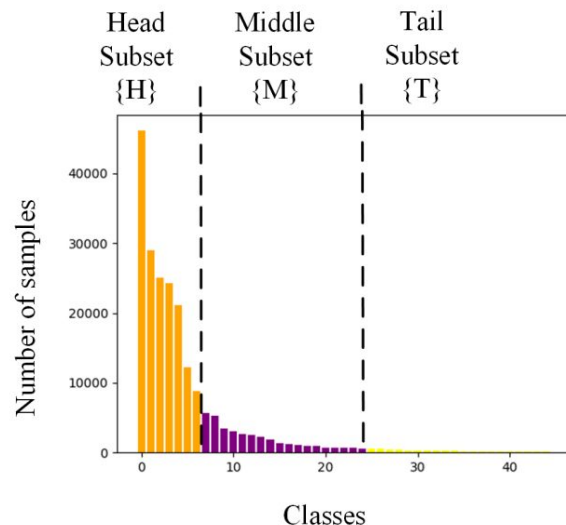
Inter-subset Mixup Strategies (1)

- **Partition** the total categories set C

- Head ($H \subset C$)
- Middle ($M \subset C$)
- Tail ($T \subset C$)

- **Mixup**

- (x_i, y_i) and (x_j, y_j) are two examples drawn at random from our training data.
- $\lambda \in [0, 1]$, where $\lambda \sim \text{Beta}(\alpha, \alpha)$, for $\alpha \in (0, \infty)$.
- $\tilde{x} = \lambda x_i + (1 - \lambda)x_j$, where x_i, x_j are raw input vectors
- $\tilde{y} = \lambda y_i + (1 - \lambda)y_j$, where y_i, y_j are one-hot label encodings

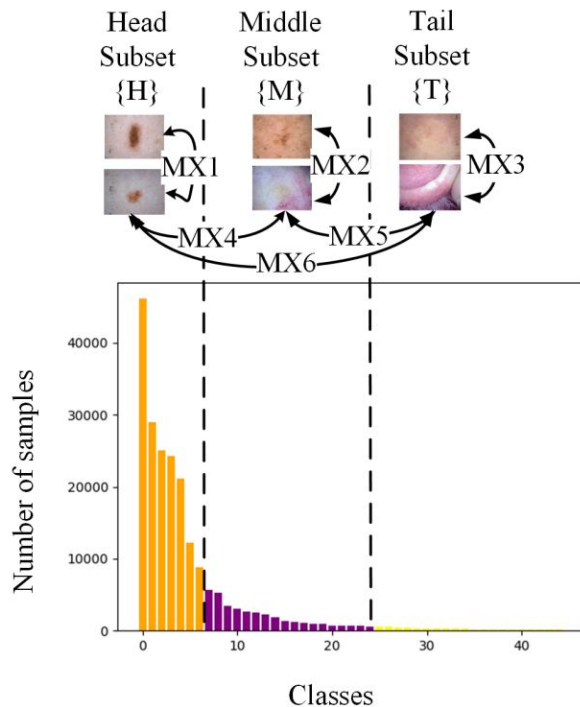


[Zhang, H., Cisse, M., Dauphin, Y.N. and Lopez-Paz, D., 2017. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*.](#)

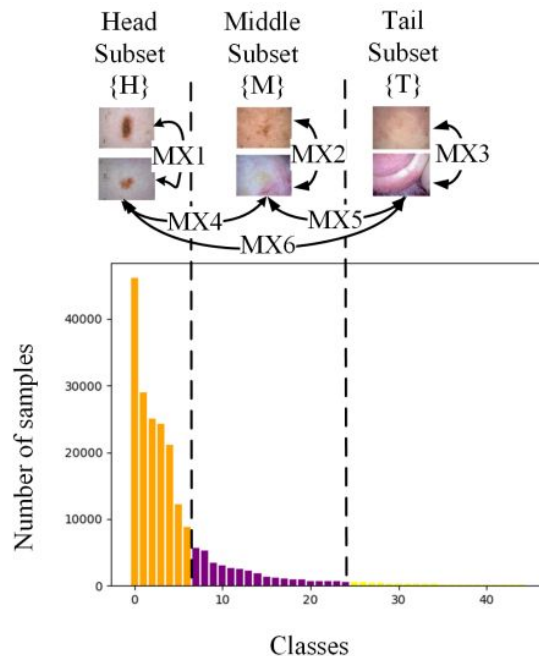
Inter-subset Mixup Strategies (2)

- Mixup strategy
 - intra-subset mixup : MX1 ~ MX3
 - inter-subset mixup : MX4 ~ MX6
- **Mixup loss** for the specific subset selected for mixup.

$$\mathcal{L}_{mixup} = \lambda \mathcal{L}_{CE}(f(x_i), y_i) + (1 - \lambda) \mathcal{L}_{CE}(f(x_j), y_j)$$



Inter-subset Mixup Strategies (3)



Intra-subset Mixup Strategies

MX1 – Head-Head Mixup

$$\mathcal{L}_{total} = \lambda_H \mathcal{L}_{mixup\{H\}} + \lambda_M \mathcal{L}_{CE\{M\}} + \lambda_T \mathcal{L}_{CE\{T\}}$$

MX2 – Middle-Middle Mixup

$$\mathcal{L}_{total} = \lambda_H \mathcal{L}_{CE\{H\}} + \lambda_M \mathcal{L}_{mixup\{M\}} + \lambda_T \mathcal{L}_{CE\{T\}}$$

MX3 – Tail-Tail Mixup

$$\mathcal{L}_{total} = \lambda_H \mathcal{L}_{CE\{H\}} + \lambda_M \mathcal{L}_{CE\{M\}} + \lambda_T \mathcal{L}_{mixup\{T\}}$$

Inter-subset Mixup Strategies

MX4 – Head-Middle Mixup

$$\mathcal{L}_{total} = \lambda_{HM} \mathcal{L}_{mixup\{H-M\}} + \lambda_T \mathcal{L}_{CE\{T\}}$$

MX5 – Middle-Tail Mixup

$$\mathcal{L}_{total} = \lambda_H \mathcal{L}_{CE\{H\}} + \lambda_{MT} \mathcal{L}_{mixup\{M-T\}}$$

MX6 – Head-Tail Mixup

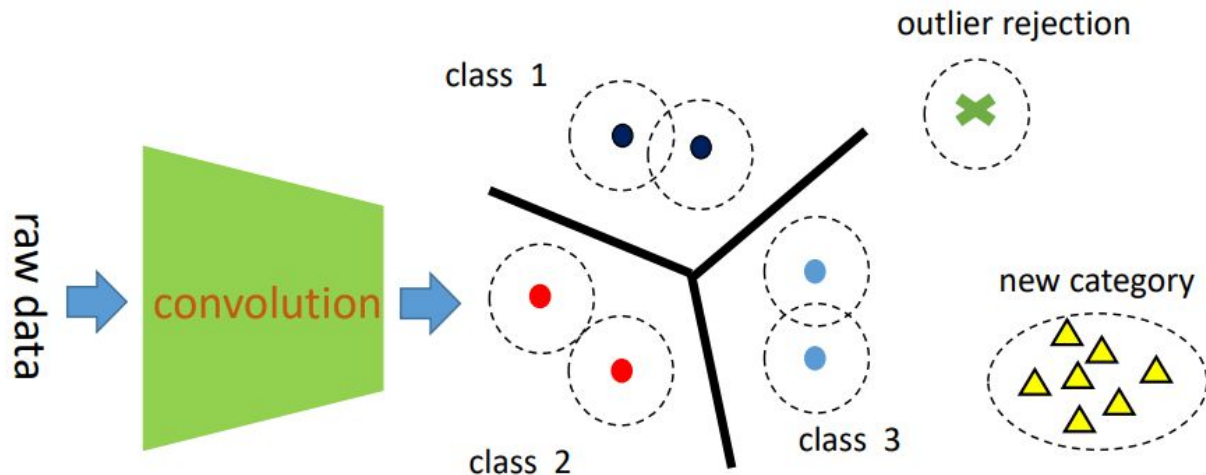
$$\mathcal{L}_{total} = \lambda_M \mathcal{L}_{CE\{M\}} + \lambda_{HT} \mathcal{L}_{mixup\{H-T\}}$$

Proposed Method

- Inter-subset mixup strategy
 - Target the middle and tail classes (MX5)
- Convolutional Prototype learning
 - Tackle the fine-grained aspect

Integration of Mixup with Prototype Learning

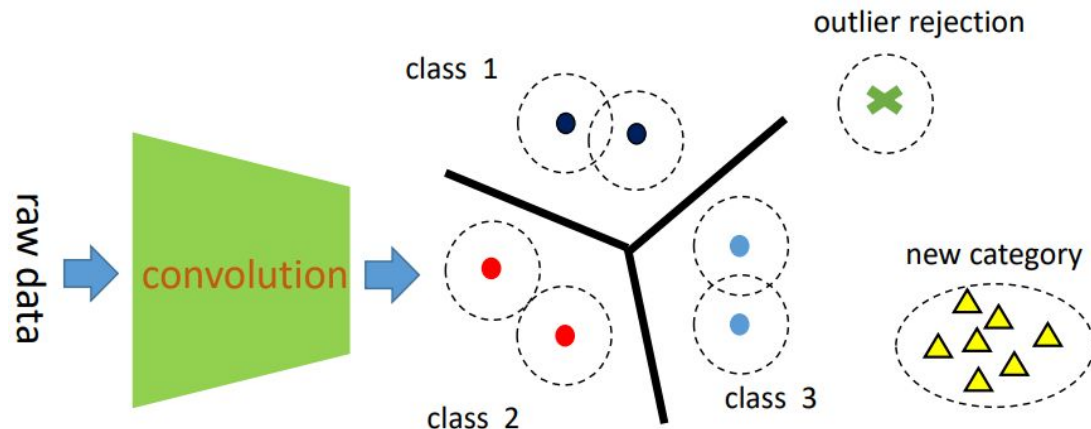
- Convolutional Prototype Learning



H. Yang, X. Zhang, F. Yin and C. Liu, "Robust Classification with Convolutional Prototype Learning," 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018, pp. 3474-3482, doi: 10.1109/CVPR.2018.00366.

Convolutional Prototype Learning (CPL)

- Components in CPL
 - **Convolutional layers** : Extract discriminative features
 - **Multiple prototypes** : Represent different classes
 - **The classification** : Finding the nearest prototype (using Euclidean distance) in the feature space.



Convolutional Prototype Learning

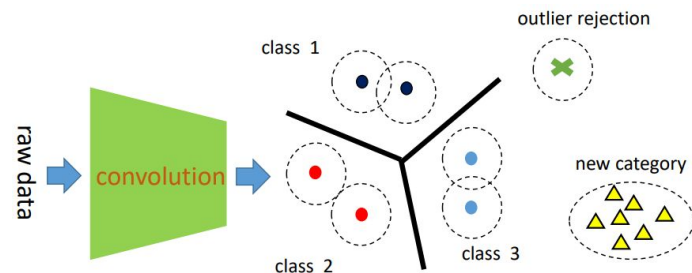
- Components in CPL

- **Convolutional layers**

- Feature extractor(CNN) is denoted as $f(x; \theta)$
 - x and θ denote the raw input and parameters of the CNN

- **Multiple prototype** m_{ij}

- $i \in \{1, 2, \dots, C\}$ represents the index of the classes
 - $j \in \{1, 2, \dots, K\}$ represents the index of the prototypes in each class
 - The final learned representation is **intra-class compact** and **inter-class separable**.

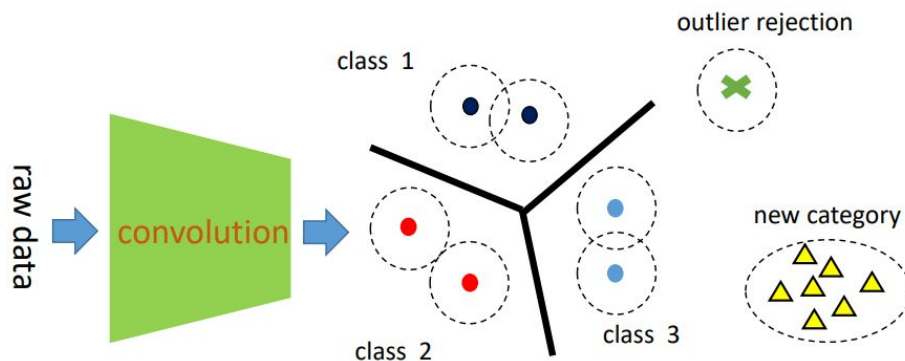


Architecture of the framework (1)

- Feedforward for prediction

- Given an input pattern x , $x \in \text{class } \arg \max_{i=1}^C g_i(x)$
- where $g_i(x)$ is the discriminant function for class i

$$g_i(x) = -\min_{j=1}^K \|f(x; \theta) - m_{ij}\|_2^2$$



Architecture of the framework (2)

- Backward for training
 - The trainable parameters
 - θ : the parameters of the CNN extractor
 - $M = \{ m_{ij} \mid i = 1, \dots, C; j = 1, \dots, K \}$: the prototypes in each class
 - Loss function
 - Intra-class compact
 - Inter-class separable representations
 - Should be derivable with respect to θ and M as well.

Architecture of the framework (3)

- Classification loss
 - Minimum classification error loss (MCE)
 - Margin based classification loss (MCL)
 - **Distance based cross entropy loss (DCE)**

Distance based cross entropy loss (DCE) (1)

- The distance can be used to measure the similarity between the samples and the prototypes

$$p(x \in m_{ij}|x) \propto -\|f(x) - m_{ij}\|_2^2.$$

- To satisfy the non-negative and sum-to-one properties of the probability, we further define the probability as

$$p(x \in m_{ij}|x) = \frac{e^{-\gamma d(f(x), m_{ij})}}{\sum_{k=1}^C \sum_{l=1}^K e^{-\gamma d(f(x), m_{kl})}}$$

where $d(f(x), m_{ij}) = \|f(x) - m_{ij}\|_2^2$

Distance based cross entropy loss (DCE) (2)

- Given the definition of $p(x \in m_{ij}|x)$, we can further define the probability of $p(y|x)$ as:

$$p(y|x) = \sum_{j=1}^K p(x \in m_{yj}|x)$$

- Based on the probability of $p(y|x)$, we can define the cross entropy (CE) loss under our framework as:

$$l((x, y); \theta, M) = -\log p(y|x)$$

Generalized CPL with prototype loss(GCPL)

- Directly minimizing the classification loss may lead to over-fitting.
- Add **prototype loss (PL)** as a **regularization**, which acts like a generative model to improve the generalization performance of CPL.

$$pl((x, y); \theta, M) = \|f(x) - m_{yj}\|_2^2$$

- The total loss :

$$loss((x, y); \theta, M) = \underbrace{l((x, y); \theta, M)}_{\text{DCE Classification Loss}} + \lambda \underbrace{pl((x, y); \theta, M)}_{\text{PL Loss}}$$

DCE

Classification
Loss

PL Loss

Generalized CPL with prototype loss(GCPL)

- **PL** pull the features of samples close to their corresponding prototypes, implicitly increase the distance between the classes.



- The **classification loss** stresses the **separation** property.
The **prototype loss** stresses the **compactness** property.
- More robust and more appropriate for **rejection** and **open set problems**.

Experiments and analysis on MNIST

GCPL

CPL

$$l((x, y); \theta, M) + \lambda pl((x, y); \theta, M)$$

Classification
Loss

PL Loss

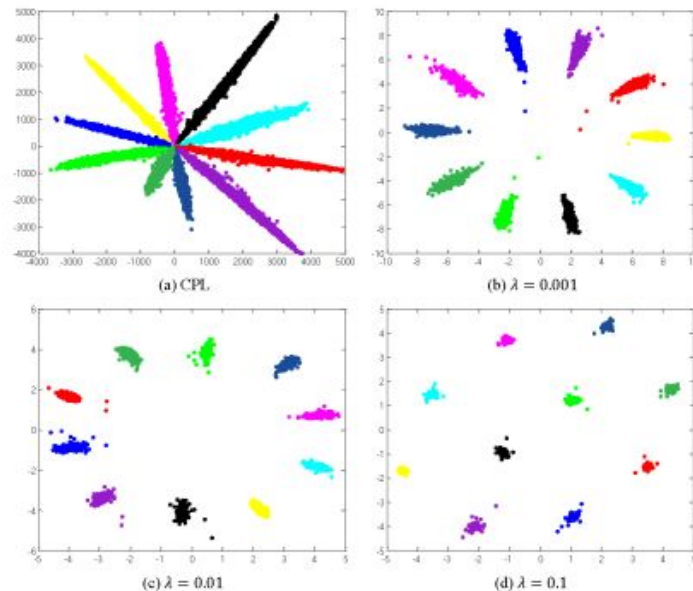




Figure 3. The learned representations of CPL and GCPL on MNIST. Different colors represent different classes

Integration of Mixup with Prototype Learning

- Prototype learning  learn fine-grained features
- Best performing mixup strategy  long-tailed aspects

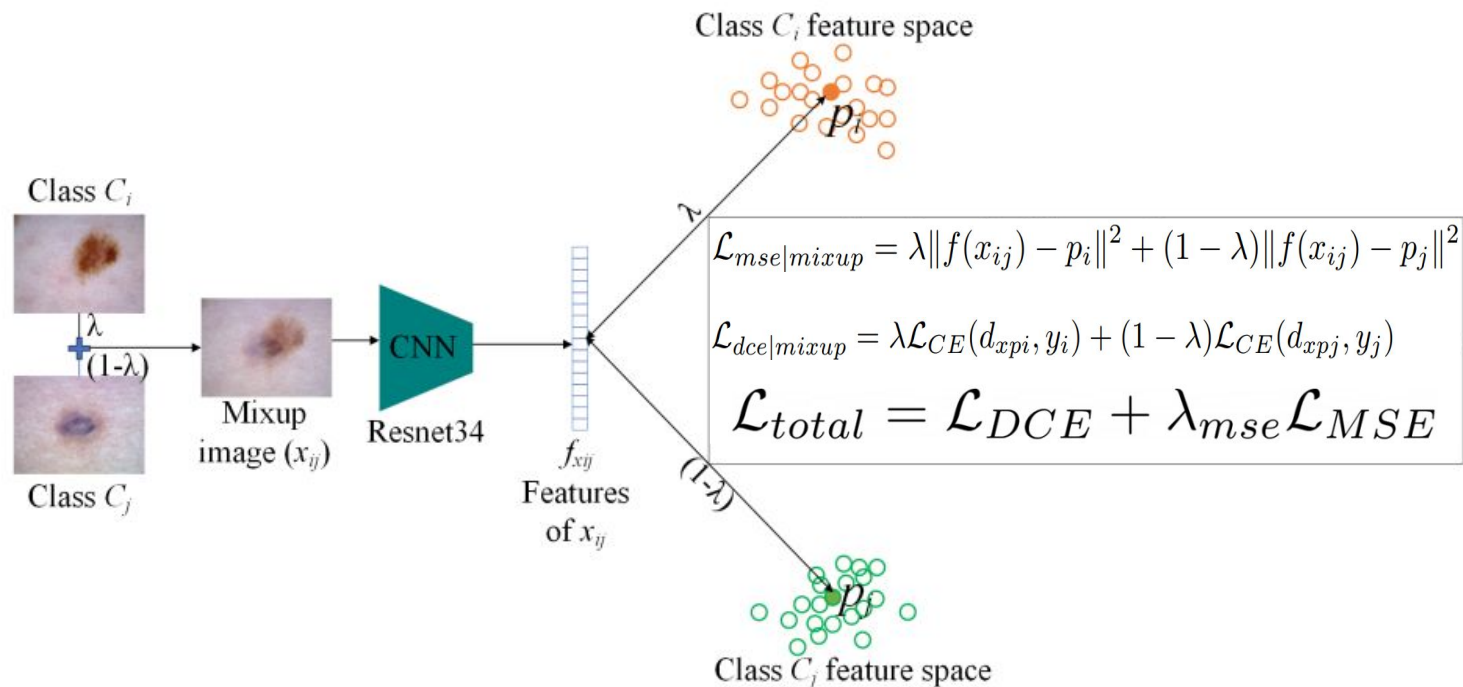
$$\mathcal{L}_{mse|mixup} = \lambda \|f(x_{ij}) - p_i\|^2 + (1 - \lambda) \|f(x_{ij}) - p_j\|^2$$

$$\mathcal{L}_{dce|mixup} = \lambda \mathcal{L}_{CE}(d_{xpi}, y_i) + (1 - \lambda) \mathcal{L}_{CE}(d_{xpj}, y_j)$$

$$-\log p(y|x)$$

- where $\{d_{xpi}, d_{xpj}\}$ is the square of the distance between the feature f_{xij} from the class specific prototypes $\{p_i, p_j\}$

Integration of Mixup with Prototype Learning



GCPL : $loss((x, y); \theta, M) = (-\log p(y|x)) + \lambda_{mse} \|f(x) - m_{y_j}\|_2^2$ 27

Dataset Settings and Evaluation Metrics (1)

- In-house dataset
 - 65 categories
 - 6 Head (more than 10,000 samples)
 - 17 Middle (500 to 10,000 samples)
 - 22 Tail (less than 500)
 - the rest 20 are reserved as OOD categories
 - 45 ID categories : 85%-15% train-test split.
 - Train set : 80%-20% for training and validation.

Dataset Settings and Evaluation Metrics (2)

- ISIC 2019 dataset
 - 8 categories
 - 2 Head (NV-12875 & MEL-4522)
 - 2 Middle (BCC-3323 & BKL-2624)
 - 2 Tail (AK-867 & SCC-628)
 - DF-239 & VASC-253 are reserved as OOD categories
 - 6 ID categories : 85%-15% train-test split.
 - Train set : 80%-20% for training and validation.

Dataset Settings and Evaluation Metrics (3)

- CIFAR-10 dataset (60000 images)
 - 10 classes categories
 - 6000 ID samples
 - 1000 more OOD samples
 - **unusual images** in a clinic, such as blurred images of skin lesions and ones that are completely covered by hair, ear, etc.
 - 50000 train images and 10000 test images.

Ablation Study of Mixup Strategies (1)

- Standard Mixup
 - Increases the overall closed set performance as well as OOD performance by 3.3% and 0.7%.
 - Decrease the closed set accuracy of tail classes.

Table 1. Performance evaluation of proposed mixup strategies on our in-house dataset

Mixup Strategy	Closed set (ID) (Acc%)				OOD (AUROC%)
	Head	Middle	Tail	Total	
Baseline	66.67	38.26	36.98	60.56	65.67
Standard Mixup	67.23	45.18	33.89	63.90	66.35
H-H Intrasubset (MX1)	70.11	34.74	25.14	60.84	64.18
M-M Intrasubset (MX2)	63.12	55.36	31.54	61.80	66.47
T-T Intrasubset (MX3)	64.29	47.96	39.49	61.06	66.25
H-M Intersubset (MX4)	66.92	44.97	22.21	62.31	64.33
M-T Intersubset (MX5)	63.67	55.14	38.76	60.97	68.78
H-T Intersubset (MX6)	66.95	36.32	36.67	59.24	64.45

Ablation Study of Mixup Strategies (2)

- MX2, MX3
 - Help to increase the corresponding subset closed set accuracies.
- MX5
 - Significantly increases the OOD performance by 3%.
 - The overall accuracy only increases slightly when compared to the baseline.

Table 1. Performance evaluation of proposed mixup strategies on our in-house dataset

Mixup Strategy	Closed set (ID) (Acc%)				OOD (AUROC%)
	Head	Middle	Tail	Total	
Baseline	66.67	38.26	36.98	60.56	65.67
Standard Mixup	67.23	45.18	33.89	63.90	66.35
H-H Intrasubset (MX1)	70.11	34.74	25.14	60.84	64.18
M-M Intrasubset (MX2)	63.12	55.36	31.54	61.80	66.47
T-T Intrasubset (MX3)	64.29	47.96	39.49	61.06	66.25
H-M Intersubset (MX4)	66.92	44.97	22.21	62.31	64.33
M-T Intersubset (MX5)	63.67	55.14	38.76	60.97	68.78
H-T Intersubset (MX6)	66.95	36.32	36.67	59.24	64.45

Dataset Settings and Evaluation Metrics

- Training Implementation
 - use Resnet34[12] as backbone architecture : Resnet34
 - Adam optimizer
 - batch size of 32
 - initial learning rate of $1e-4$ with exponential decay for 45 epochs
 - Resize the input image to a size of 224x224
 - Standard data augmentation of random crop and horizontal flip
- Evaluation Metrics
 - closed set performance
 - precision (pre), recall (rec), and f1-score (f1)
 - OOD detection performance
 - Area Under Receiver Operator Characteristic (AUROC)
 - which are the standard metrics for measuring the performance of a model for OOD detection task [10].

[10] Geng, C., Huang, S.j., Chen, S.: Recent advances in open set recognition: A survey. IEEE transactions on pattern analysis and machine intelligence (2020)

[12] He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition (2016)

Benchmarking with other methods (1)

- The existing OOD techniques are capable of detecting relatively easy OOD samples coming from a completely different domain.

Table 2. Benchmarking of OOD techniques on both In-house and ISIC2019 dataset. ID metrics - {Precision (pre), Recall (rec), and f1-score(f1)}; OOD metrics - {AUROC(%)} (best viewed in zoom).

Method	In-house dataset						ISIC2019				
	ID(pre)	ID(rec)	ID(f1)	OOD(20cl)	OOD(unk)	OOD(Cifar)	ID(pre)	ID(rec)	ID(f1)	OOD(2cl)	OOD(Cifar)
Baseline	0.58	0.59	0.585	65.67	52.90	73.24	0.86 ± 0.03	0.86 ± 0.02	0.86 ± 0.02	68.15 ± 0.9	76.43 ± 0.4
Baseline+LS+RandAug+LRS [29]	0.62	0.63	0.625	66.19	63.13	96.34	0.87 ± 0.015	0.86 ± 0.017	0.865 ± 0.015	69.41 ± 0.5	94.87 ± 0.6
ODIN [18]	0.61	0.59	0.60	64.92	62.79	96.48	0.83 ± 0.03	0.81 ± 0.02	0.82 ± 0.025	66.21 ± 1.3	95.60 ± 1.2
OLTR [20]	0.63	0.62	0.625	67.42	70.72	98.00	0.85 ± 0.01	0.86 ± 0.02	0.855 ± 0.015	71.66 ± 0.6	98.45 ± 0.5
MC-Dropout [8]	0.59	0.58	0.585	66.07	68.83	97.57	0.84 ± 0.023	0.84 ± 0.02	0.84 ± 0.02	72.18 ± 0.3	96.41 ± 0.3
ARPL [4]	0.64	0.63	0.635	68.55	80.61	99.42	0.85 ± 0.01	0.86 ± 0.016	0.855 ± 0.012	74.16 ± 0.7	97.20 ± 0.4
Mixup [32]	0.63	0.62	0.625	66.35	66.70	97.10	0.87 ± 0.02	0.88 ± 0.01	0.875 ± 0.013	71.72 ± 0.7	96.65 ± 0.6
Prototype [30]	0.63	0.62	0.625	68.82	74.54	98.04	0.85 ± 0.02	0.86 ± 0.02	0.855 ± 0.02	72.84 ± 0.6	97.02 ± 0.5
M-T Mixup (Ours)	0.61	0.60	0.605	68.78	70.81	99.29	0.85 ± 0.03	0.85 ± 0.02	0.85 ± 0.022	73.86 ± 0.6	97.10 ± 0.6
M-T Mixup + Prototype (Ours)	0.62	0.61	0.615	71.10	82.71	99.59	0.85 ± 0.01	0.86 ± 0.02	0.855 ± 0.015	76.37 ± 0.5	98.46 ± 0.4

Benchmarking with other methods (2)

- The performance of all the techniques drops drastically when the OOD samples are from the same domain. Specifically, this is more evident for a long-tailed nature dataset such as our in-house dataset.

Table 2. Benchmarking of OOD techniques on both In-house and ISIC2019 dataset. ID metrics - {Precision (pre), Recall (rec), and f1-score(f1)}; OOD metrics - {AUROC(%)} (best viewed in zoom).

Method	In-house dataset						ISIC2019				
	ID(pre)	ID(rec)	ID(f1)	OOD(20cl)	OOD(unk)	OOD(Cifar)	ID(pre)	ID(rec)	ID(f1)	OOD(2cl)	OOD(Cifar)
Baseline	0.58	0.59	0.585	65.67	52.90	73.24	0.86 \pm 0.03	0.86 \pm 0.02	0.86 \pm 0.02	68.15 \pm 0.9	76.43 \pm 0.4
Baseline+LS+RandAug+LRS [29]	0.62	0.63	0.625	66.19	63.13	96.34	0.87 \pm 0.015	0.86 \pm 0.017	0.865 \pm 0.015	69.41 \pm 0.5	94.87 \pm 0.6
ODIN [18]	0.61	0.59	0.60	64.92	62.79	96.48	0.83 \pm 0.03	0.81 \pm 0.02	0.82 \pm 0.025	66.21 \pm 1.3	95.60 \pm 1.2
OLTR [20]	0.63	0.62	0.625	67.42	70.72	98.00	0.85 \pm 0.01	0.86 \pm 0.02	0.855 \pm 0.015	71.66 \pm 0.6	98.45 \pm 0.5
MC-Dropout [8]	0.59	0.58	0.585	66.07	68.83	97.57	0.84 \pm 0.023	0.84 \pm 0.02	0.84 \pm 0.02	72.18 \pm 0.3	96.41 \pm 0.3
ARPL [4]	0.64	0.63	0.635	68.55	80.61	99.42	0.85 \pm 0.01	0.86 \pm 0.016	0.855 \pm 0.012	74.16 \pm 0.7	97.20 \pm 0.4
Mixup [32]	0.63	0.62	0.625	66.35	66.70	97.10	0.87 \pm 0.02	0.88 \pm 0.01	0.875 \pm 0.013	71.72 \pm 0.7	96.65 \pm 0.6
Prototype [30]	0.63	0.62	0.625	68.82	74.54	98.04	0.85 \pm 0.02	0.86 \pm 0.02	0.855 \pm 0.02	72.84 \pm 0.6	97.02 \pm 0.5
M-T Mixup (Ours)	0.61	0.60	0.605	68.78	70.81	99.29	0.85 \pm 0.03	0.85 \pm 0.02	0.85 \pm 0.022	73.86 \pm 0.6	97.10 \pm 0.6
M-T Mixup + Prototype (Ours)	0.62	0.61	0.615	71.10	82.71	99.59	0.85 \pm 0.01	0.86 \pm 0.02	0.855 \pm 0.015	76.37 \pm 0.5	98.46 \pm 0.4

Benchmarking with other methods (3)

- M-T mixup (MX5) strategy combined with prototype learning performs the best for OOD detection while maintaining the overall ID performance compared to the baseline on both datasets.

Table 2. Benchmarking of OOD techniques on both In-house and ISIC2019 dataset. ID metrics - {Precision (pre), Recall (rec), and f1-score(f1)}; OOD metrics - {AUROC(%)} (best viewed in zoom).

Method	In-house dataset						ISIC2019				
	ID(pre)	ID(rec)	ID(f1)	OOD(20cl)	OOD(unk)	OOD(Cifar)	ID(pre)	ID(rec)	ID(f1)	OOD(2cl)	OOD(Cifar)
Baseline	0.58	0.59	0.585	65.67	52.90	73.24	0.86 ± 0.03	0.86 ± 0.02	0.86 ± 0.02	68.15 ± 0.9	76.43 ± 0.4
Baseline+LS+RandAug+LRS [29]	0.62	0.63	0.625	66.19	63.13	96.34	0.87 ± 0.015	0.86 ± 0.017	0.865 ± 0.015	69.41 ± 0.5	94.87 ± 0.6
ODIN [18]	0.61	0.59	0.60	64.92	62.79	96.48	0.83 ± 0.03	0.81 ± 0.02	0.82 ± 0.025	66.21 ± 1.3	95.60 ± 1.2
OLTR [20]	0.63	0.62	0.625	67.42	70.72	98.00	0.85 ± 0.01	0.86 ± 0.02	0.855 ± 0.015	71.66 ± 0.6	98.45 ± 0.5
MC-Dropout [8]	0.59	0.58	0.585	66.07	68.83	97.57	0.84 ± 0.023	0.84 ± 0.02	0.84 ± 0.02	72.18 ± 0.3	96.41 ± 0.3
ARPL [4]	0.64	0.63	0.635	68.55	80.61	99.42	0.85 ± 0.01	0.86 ± 0.016	0.855 ± 0.012	74.16 ± 0.7	97.20 ± 0.4
Mixup [32]	0.63	0.62	0.625	66.35	66.70	97.10	0.87 ± 0.02	0.88 ± 0.01	0.875 ± 0.013	71.72 ± 0.7	96.65 ± 0.6
Prototype [30]	0.63	0.62	0.625	68.82	74.54	98.04	0.85 ± 0.02	0.86 ± 0.02	0.855 ± 0.02	72.84 ± 0.6	97.02 ± 0.5
M-T Mixup (Ours)	0.61	0.60	0.605	68.78	70.81	99.29	0.85 ± 0.03	0.85 ± 0.02	0.85 ± 0.022	73.86 ± 0.6	97.10 ± 0.6
M-T Mixup + Prototype (Ours)	0.62	0.61	0.615	71.10	82.71	99.59	0.85 ± 0.01	0.86 ± 0.02	0.855 ± 0.015	76.37 ± 0.5	98.46 ± 0.4

Confidence Scores Visualization

- In Fig 4, we analyse the performance results in more detail by showing the **probability density of the confidence scores** for different subsets.
- The larger the separation between the distribution of OOD {O, U} from ID{H, M, T}, the better the technique.
- Specifically, it is to be noted that this is achieved by **making the {M,T} subsets more confident** which justifies our targeted strategy

Confidence Scores Visualization

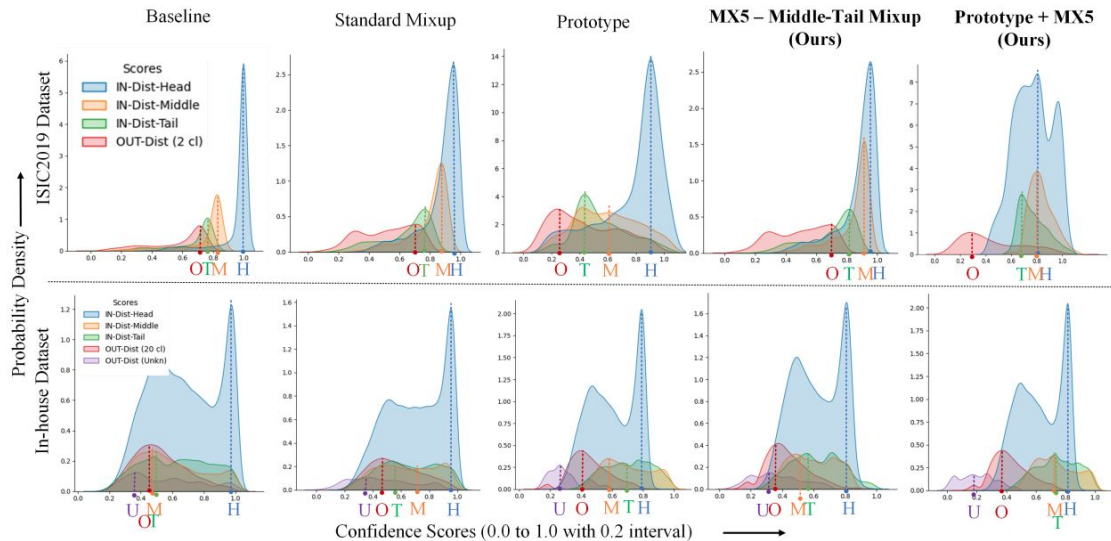


Fig. 4. Confidence Scores visualization for different methods on our In-house dataset and ISIC dataset. {H,M,T} refer to Head, Middle, and Tail subsets. {O} refers to OOD(2cl) and OOD(20cl) for ISIC and In-house dataset. {U} refers to the OOD(unk) for In-house dataset. (best viewed in zoom).

Thank You For Listening