RETHINKING LONG-TAILED VISUAL RECOGNITION WITH DYNAMIC PROBABILITY SMOOTHING AND FREQUENCY WEIGHTED FOCUSING

Wan Jun Nah[1], *Chun Chet Ng*[1], *Che-Tsung Lin*[2], *Yeong Khang Lee*[3],
Jie Long Kew[1], *Zhi Qin Tan*[1], *Chee Seng Chan*[1]*, *Christopher Zach*[2], *Shang-Hong Lai*[4]

[1]CISiP, Faculty of Comp. Sci. and Info. Tech., Universiti Malaya, Kuala Lumpur, Malaysia,
[2]Dept. of Electrical Engineering, Chalmers University of Technology, Gothenburg, Sweden,
[3]Centre of Excellence, ViTrox Corporation Berhad, Penang, Malaysia,
[4]Dept. of Computer Science, National Tsing Hua University, Hsinchu, Taiwan.

ABSTRACT

Deep learning models trained on long-tailed (LT) datasets often exhibit bias towards head classes with high frequency. This paper highlights the limitations of existing solutions that combine class- and instance-level re-weighting loss in a naive manner. Specifically, we demonstrate that such solutions result in overfitting the training set, significantly impacting the rare classes. To address this issue, we propose a novel loss function that dynamically reduces the influence of outliers and assigns class-dependent focusing parameters. We also introduce a new long-tailed dataset, ICText-LT, featuring various image qualities and greater realism than artificially sampled datasets. Our method has proven effective, outperforming existing methods through superior quantitative results on CIFAR-LT, Tiny ImageNet-LT, and our new ICText-LT datasets. The source code and new dataset are available at `https://github.com/nwjun/FFDS-Loss`.

Index Terms— Long-tailed Classification, Weighted-loss

1. INTRODUCTION

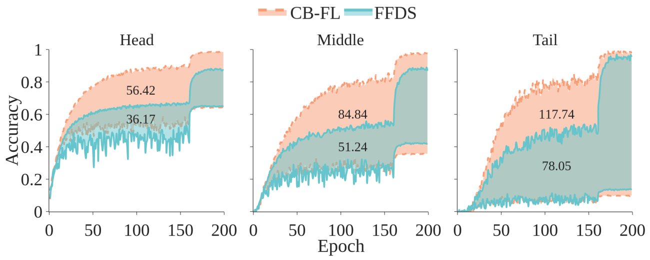


Fig. 1: Overfitting of CB-FL on CIFAR-100-LT, IF $= 100$. The colored areas between the upper bound (training acc.) and lower bound (testing acc.) indicate the difference of acc. on training and testing sets. As the class frequency decreases, the gap gets larger. The smaller colored areas prove the superiority of our method (FFDS).

Common approaches to address long-tailed (LT) classification problems include data re-sampling [1, 2] and cost-sensitive re-weighting [3–6]. Data re-sampling involves either over-sampling by replicating data from minor classes or under-sampling by removing samples from major classes to balance the dataset distribution. This allows the model to train on a balanced dataset and prevents it from favoring certain classes. In contrast, cost-sensitive re-weighting aims to influence the loss by assigning relatively higher penalties to under-represented samples. For instance, class-wise re-weighting methods [4, 7] assign weights based on class frequency. However, the assumption that every sample within a class is equally important is invalid as some samples are more difficult to learn than others. As such, instance-wise re-weighting methods dynamically assign loss weight to each instance during training [6, 8]. For example, [6] re-weights an instance based on its difficulty, determined by the predicted probability from a model. Class-balanced focal loss (CB-FL) [4], one of the most popular methods, combines class- and instance-wise re-weighting methods. Although CB-FL offers some improvements, it is prone to overfit with a large gap between training and testing accuracy and deteriorates as class frequency decreases (Fig. 1). We hypothesize this is due to an unintentional focus on outliers with significant instance-level weights. Furthermore, overfitting is exacerbated when the number of samples decreases as class-level weights amplify the loss of outliers.

Motivated by these issues, this paper makes three main contributions: (i) We identify the limitations of naively combining class-wise and instance-wise re-weighting methods (Fig. 1) and propose a novel loss function, Frequency weighted Focusing with Dynamic Smoothing (FFDS), to dynamically smooth instance-wise weights and assign focusing parameters based on class frequency (Sec. 3); (ii) We introduce a new long-tailed industrial dataset, ICText-LT, with different qualities and collected under more realistic settings (Sec. 4); and (iii) We demonstrate the effectiveness of our proposed method on various LT datasets, outperforming state-of-the-art alternatives (Table 1a).

*Corresponding author - cs.chan@um.edu.my

2. RELATED WORK

Loss Re-weighting. Loss re-weighting methods assign different weights to samples, with commonly used approaches including re-weighting by inverse class frequency or inverse square root of class frequency. Cui et al. [4] noted that as more data becomes available, the marginal benefit that a model can extract from the data diminishes, leading them to re-weight classes by the inverse of the effective number of samples. Additionally, [9] suggests assigning larger class-dependent margins to tail classes. Since minority classes are not always the most under-represented, another line of work [3, 6] dynamically re-weights each sample based on its difficulty. Influenced-Balanced Loss (IB-Loss) [5] assigns a sample's weight based on its influence on the decision boundary.

Two-stage Methods. Two-stage methods involve training a model in two separate stages using different losses. LDAM [9] defers re-weighting to the later stage of training, demonstrating that the model can learn better features with instance-balanced data. In contrast, [10] proposes splitting training into feature learning and classifier learning phases. Specifically, the model is trained using standard Cross-Entropy (CE) in the first phase, while in the second phase, the backbone is frozen and the classifier is re-trained using different re-sampling or re-weighting schemes.

Label Smoothing. Label smoothing is used to mitigate overfitting by softening ground-truth labels. As noted by Müller et al. [11], it calibrates the model so that prediction confidences are more aligned with their accuracies.

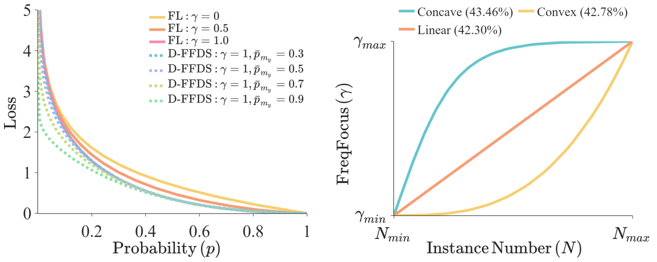
3. PROPOSED METHOD

When a neural network is trained on long-tailed datasets, its gradient can be overwhelmed by samples from head classes, resulting in biased predictions towards the majority. To address this issue, this paper introduces two new modules: (i) dynamic probability smoothing (DynaSmooth) to alleviate overfitting observed in CB-FL, and (ii) frequency-weighted focusing (FreqFocus) to address intra-class imbalance by re-weighting hard examples of each class based on class frequency. We follow [6] in defining an instance's difficulty based on its negative association with predicted probability.

Let $z = [z_1, z_2, \cdots, z_C]^T$ denote the predicted logits of a model for all C classes. The probability distribution over all classes computed with softmax function is written as $p_i = e^{z_i} / \sum_{j=1}^{C} e^{z_j}, \forall i \in \{1, 2, \cdots, C\}$. As such, our proposed loss function L_{FFDS} can be stated as:

$$L_{\text{FFDS}}(z, y) = -w_y(1 - \hat{p}_y)^{\gamma_y} \sum\nolimits_{j}^{C} Q(j) \log(p_j), \tag{1}$$

$$Q(j) = (1 - \epsilon)\mathbb{1}_{\{j=y\}} + (\epsilon/(C-1))\mathbb{1}_{\{j \neq y\}}, \tag{2}$$



(a) (b)

Fig. 2: (a): Smoothing effect of DynaSmooth on loss based on different values of $\bar{p}_{m_y}$. (b): Different forms of curves to map instance number to γ in FreqFocus and respective accuracy on CIFAR-100-LT, IF = 100.

where y is the ground-truth class, $w_y = \frac{1-\beta}{1-\beta^{N_y}}$ is the class-balanced weight [4] for N_y examples, and β controls the rate of growth of the effective number of samples as N_y increases. $\hat{p}_y$ is the smoothed p_y that will be detailed next, while $Q(j)$ is the label of j from label smoothing [12] with a smoothing parameter empirically set as $\epsilon = 0.1$. Finally, γ_y is a class-level frequency-weighted focusing parameter. It controls the contribution of easy/hard examples and is detailed in Sec. 3b.

(a) Dynamic Probability Smoothing (DynaSmooth). DynaSmooth aims to mitigate the impact of outliers on overfitting by dynamically smoothing instance-level weights based on their likelihood of being outliers, where outliers are predictions with extreme probabilities.

To ensure training stability, we sort the classes in descending order by frequency and partition them into M equally sized groups, as they directly represent the underlying distribution of the dataset. We then calculate the square root difference, $d \in [-1, 1]$, between the predicted probability of the ground-truth class, p_y, and the mean predicted probability of the group containing y from the previous epoch, denoted as $\bar{p}_{m_y}$. Thus, we have $d = \sqrt{p_y} - \sqrt{\bar{p}_{m_y}}$. Calculating the square root difference, we prioritize small deviations when both values are small. The impact of $\bar{p}_{m_y}$ is shown in Fig.2a, where the loss decreases as $|d|$ increases. Next, we design a function $f(d) = (kd)^2 \in [0, 1]$ to compute the percentage of smoothing with a positive correlation to $|d|$. Here, k is a hyperparameter adjusting the rate of change of $f(d)$. The smoothing percentage, $f(d)$, is then multiplied by the difference between p_y and $\bar{p}_{m_y}$ to determine the magnitude of smoothing. The final formula for DynaSmooth leading to the smoothed probability, $\hat{p}_y$, is:

$$\hat{p}_y = p_y - f(d)(p_y - \bar{p}_{m_y}), \tag{3}$$

where $\hat{p}_y$ is bounded by p_y and $\bar{p}_{m_y}$. As shown in Fig. 3, DynaSmooth pushes the outliers of p_y for all classes closer to $\bar{p}_{m_y}$, reducing outliers and their weights, leading to $\hat{p}_y$. Although head classes have more outliers, they are less overfitted than tail classes due to their lower class-level weights, making them less influential during training. To balance outlier reduction and the risk of over-smoothing, we introduce a more flexible focusing parameter, detailed in the next section.

(b) Frequency-weighted Focusing (FreqFocus). FreqFocus addresses the issue of instance-level imbalance through a frequency-based per-class focusing parameter. In most cases,

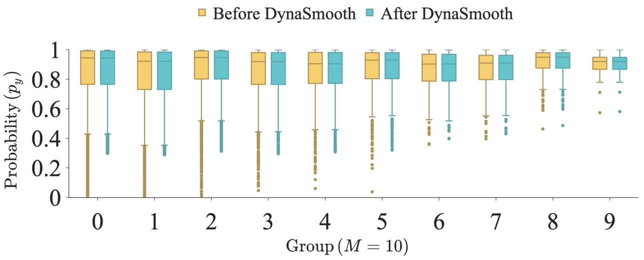


Fig. 3: Boxplot of probability, p_y, before and after applying DynaSmooth for instance-wise re-weighting. Each group contains 10 classes from CIFAR-100-LT with IF = 100, sorted from head (left) to tail (right). After applying DynaSmooth, the number of outliers decreases and their corresponding weights are reduced, resulting in their reduced contribution to the training process.

Algorithm 1 D-FFDS Training Scheme

```
Require: $B = (x_i, y_i)_{i=1}^{N}$: dataset, $f_\theta$: $\theta$ parameterized model, $\delta$: learning rate
Initialize the model with random parameters $\theta$
$t \leftarrow 0$
while $t < T_{\text{phase\_1}}$ do
    $B_m \sim B$    ▷ Sample mini-batch with $m$ samples
    $L(f_\theta) \leftarrow \frac{1}{m}\sum_{(x,y)\in B_m} L_{\text{CE}}(f(\theta), y)$
    $f_\theta \leftarrow f_\theta - \delta\nabla L(f_\theta)$
    $t \leftarrow t + 1$
end while
while $t < T_{\text{phase\_2}}$ do
    $B_m \sim B$    ▷ Sample mini-batch with $m$ samples
    $L(f_\theta) \leftarrow \frac{1}{m}\sum_{(x,y)\in B_m} L_{\text{FFDS}}(f(\theta), y)$
    $f_\theta \leftarrow f_\theta - \delta\nabla L_{\text{FFDS}}(f_\theta)$
    $t \leftarrow t + 1$
end while
```

improvement of head classes is limited due to the presence of hard examples, while tail classes suffer from limited data availability. To address this, we propose a focusing parameter, γ_y, positively correlated with the frequency of the ground-truth class, N_y. This encourages the model to attend to hard examples in head classes while treating examples equally in tail classes. We formulate our idea using three possible forms of curves to define $\gamma_y \in [\gamma_{\text{min}}, \gamma_{\text{max}}]$.

First, we consider a linear form that increases γ_y uniformly with respect to class frequency:

$$\gamma_y = \gamma_{\text{min}} + (\gamma_{\text{max}} - \gamma_{\text{min}}) \left(\frac{N_y - N_{\text{min}}}{N_{\text{max}} - N_{\text{min}}}\right). \tag{4}$$

Next, we explore a convex form that gradually increases the focusing parameter and eventually caps at γ_{max}:

$$\gamma_y = \gamma_{\text{min}} + (\gamma_{\text{max}} - \gamma_{\text{min}}) \left(\frac{N_y - N_{\text{min}}}{N_{\text{max}} - N_{\text{min}}}\right)^3. \tag{5}$$

Finally, we consider a concave form that rapidly increases γ_y for tail classes and slows down the rate of growth as class frequency increases:

$$\gamma_y = \gamma_{\text{min}} + (\gamma_{\text{max}} - \gamma_{\text{min}}) \tanh\left(4 \cdot \frac{N_y - N_{\text{min}}}{N_{\text{max}} - N_{\text{min}}}\right). \tag{6}$$

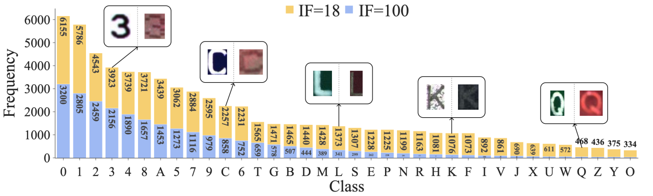


Fig. 4: Distribution of ICText-LT's training set with IF $\in \{18, 100\}$. Images shown are good (left) and bad (right) quality samples.

Fig. 2b shows the growth rate of γ in three forms and their corresponding accuracy on CIFAR-100, IF $= 100$. The concave form demonstrates the highest accuracy, as it better models the increase in the number of instances, making it more suitable for handling the long-tailedness of CIFAR-100-LT sampled with an exponential distribution. Here, IF $= \frac{N_{\text{max}}}{N_{\text{min}}}$ represents the degree of data imbalance, where N_{max} and N_{min} are the largest and smallest number of training instances [4].

Training Scheme. Insights reported in LDAM [9] indicate that training with a cost-sensitive function can result in difficulty converging and instability. Inspired by this, we propose a variant, Deferred-FFDS (D-FFDS), with a two-phase training process: (1) a normal training phase and (2) a fine-tuning phase controlled by $T_{\text{phase\_1}}$. During phase 1, the model is trained using CE, while in phase 2, our proposed method is applied to re-adjust the decision boundary. This ensures the model begins the fine-tuning process with a meaningful and general representation of the target dataset. Pseudocode for the D-FFDS training scheme is shown in Algorithm 1.

4. LT INDUSTRIAL DATASET - ICTEXT-LT

Most popular LT datasets only cover natural objects (e.g., CIFAR10/100-LT [13], Tiny ImageNet-LT [14]) and are collected under sufficient lighting with good image quality. However, LT image classification is more challenging in industry settings. To address this, we introduce a new long-tailed industrial dataset, ICText-LT. Originally, ICText [15] is an industrial-based dataset focused on detecting printed characters on chip components. It comprises 62 classes (A-Z, a-z, 0-9) and exhibits long-tail distribution in both training and testing sets. Herein, we resample and balance the distribution of the testing set by removing lower-case letters. As a result, the new ICText-LT with IF $= 18$ has 36 classes with 68307 training and 6300 testing images. We also sample ICText-LT based on IF $= 100$. Both distributions of ICText-LT's training set are shown in Fig. 4, along with a few samples to illustrate the variation of images and their level of challenge.

5. EXPERIMENTS

5.1. Experiment Settings

Datasets. We conducted experiments on three LT datasets: CIFAR (CIFAR-10/100) [13], Tiny ImageNet [14] and ICText-LT. CIFAR-10 and CIFAR-100 consist of 50,000 training images and 10,000 validation images with 10 and 100 classes,

	Method	CIFAR-10-LT			CIFAR-100-LT			Tiny ImageNet-LT		ICText-LT	
		100	50	10	100	50	10	100	10	100	18
One-Phase	CE	70.47	77.21	86.40	38.85	43.24	56.09	38.19	53.22	74.22	83.51
	FL [6]	70.38	76.71	86.66	38.41	44.32	55.78	38.95	54.02	74.21	83.70
	CB [4]	70.36	74.81	87.03	38.32	43.85	55.71	41.37	54.82	75.29	84.86
	CB-FL [4]	74.57	79.27	85.73	39.60	41.66	57.99	38.71	54.92	75.35	83.70
	LDAM [9]	73.35	78.74	86.96	39.60	44.19	56.91	39.40	54.58	74.24	83.38
	FFDS	**75.60**	**79.82**	**87.46**	**40.74**	**45.67**	**58.66**	**42.34**	**56.11**	**76.89**	**85.40**
Two-Phase	LDAM-DRW [9]	77.03	81.53	88.16	42.04	47.71	58.71	42.78	57.06	77.87	84.52
	IB [5]	78.26	81.70	88.25	42.14	46.22	57.13	42.65	57.22	76.59	85.41
	IB-CB [5]	78.04	81.54	88.09	41.31	46.16	56.78	40.15	55.79	75.86	85.22
	IB-FL [5]	79.76	81.51	88.04	42.06	47.49	58.20	41.04	57.06	77.59	85.05
	D-FFDS	**79.93**	**82.94**	**88.48**	**43.46**	**48.48**	**58.82**	**43.86**	**58.31**	**79.56**	**85.98**

(a)

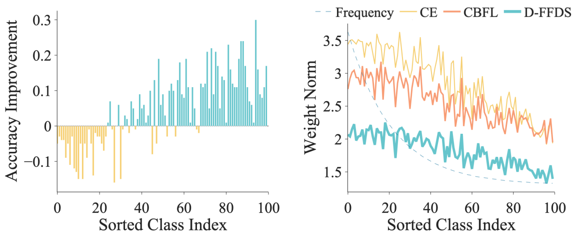
Number of Groups (M)	CIFAR-100-LT	
	100	10
1	42.66	58.24
5	42.07	**58.82**
10	**43.46**	58.29
50	43.17	58.41

(b)

DynaSmooth	FreqFocus	CIFAR-100-LT	
		100	10
-	-	41.80	57.10
-	✓	42.43	57.68
✓	-	42.71	58.21
✓	✓	**43.46**	**58.82**

(c)

Table 1: (a): Comparison of testing accuracy on public CIFAR-10-LT, CIFAR-100-LT, Tiny ImageNet-LT and ICText-LT datasets. (b): Quantitative results for the ablation study of different numbers of groups, M. (c): Ablation study of all proposed modules with deferred CB-FL as the base model. More details are in Section 5.2.



(a) (b)

Fig. 5: Per-class (a) accuracy improvement of D-FFDS compared to CE and (b) classifier weight norm, on CIFAR-100-LT, IF $=$ 100. Our method enhances accuracy and balances weight norms.

respectively. On the other hand, Tiny ImageNet contains 500 training images and 50 testing images for each of its 200 classes. Artificial long-tailed versions are created using an exponential function $N = N_j \mu^j$, where $\mu \in (0, 1)$ and j is the class index [4]. This work conducts all experiments with imbalance factors ranging from 10 to 100.

Implementation Details. All models are trained using ResNet and an SGD optimizer with a momentum of 0.9. For CIFAR-LT, we follow [4, 9] in using the ResNet-32 backbone and setting the multi-step learning rate scheduler to 0.1. ResNet-18 is trained for 100 epochs with a learning rate of 0.1 and weight decay of 2e-4 on Tiny ImageNet-LT and ICText-LT. The learning rate is decreased by a factor of 0.01 after 15 epochs for Tiny ImageNet-LT and after 60 and 80 epochs for ICText-LT. The presented results are based on a search space of $\beta \in \{0.9, 0.99, 0.999, 0.9999\}$ for class-balanced weight, $k \in [0, 1]$ and $M \in \{3, 5, 10\}$ for DynaSmooth, $\gamma \in [0, 5]$ and curve $\in \{linear, convex, concave\}$ for FreqFocus.

5.2. Results

(a) Existing Methods. Table 1a shows the comparisons of our approach to various one-stage [4, 6, 9] and two-stage [5, 9] competing methods. In both training schemes, our methods (FFDS and D-FFDS) outperform the others, with D-FFDS having the best performance across all datasets. As a result, we conduct the rest of the study using D-FFDS. Additionally, we verify that the performance improvement is mainly attributed to increased accuracy in the tail classes, as shown by the per-class accuracy of CIFAR-100-LT when IF $=$ 100 in Fig. 5a. Although performance in head classes declines slightly, the improvement in tail classes outweighs the degradation. In Fig. 5b, we show that our proposed method reduces the magnitude of the weight norm and has less bias in the majority classes.

(b) Effect of Two-Phase Training. Two-phase D-FFDS outperforms FFDS, as shown in Table 1a. We believe this is mainly due to better representation learned in phase 1 [9] and more accurate representation of groups by mean probability.

(c) Effect of Number of Groups, M. In Table 1b, we show test accuracy of models trained on CIFAR-100-LT with IF $\in \{10, 100\}$ and $M \in \{1, 5, 10, 50\}$. Note that the higher the imbalance factor, the greater the difference in class frequency of neighboring classes. The best M is 10 and 5 respectively for CIFAR-100-LT with IF $= \{10, 100\}$. Reasonably, we aim to maintain relatively balanced instances within each group to accurately represent the group by mean probability. In other words, the number of instances for all classes in a group should not vary significantly to allow for an accurate estimation of group properties.

(d) Effectiveness of Proposed Modules. In Table 1c, we show the effectiveness of each module on CIFAR-100-LT with IF $\in \{10, 100\}$. Each module improves accuracy and combining all proposed modules leads to the best result.

6. CONCLUSION

This paper addresses the limitation of naively combining class-wise and instance-wise weights in long-tailed classification by proposing a novel loss function, FFDS. It incorporates DynaSmooth to dynamically reduce weights of outliers based on probabilities and FreqFocus to independently re-weight well-/poorly-classified examples based on class frequency. We also introduce a new LT industrial dataset, ICText-LT, with challenging character representations. Experimental results show that our method outperforms existing works on CIFAR-LT, Tiny ImageNet-LT, and ICText-LT.

7. REFERENCES

[1] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer, "Smote: synthetic minority over-sampling technique," *Journal of artificial intelligence research*, vol. 16, pp. 321–357, 2002.

[2] Mateusz Buda, Atsuto Maki, and Maciej A. Mazurowski, "A systematic study of the class imbalance problem in convolutional neural networks," *Neural Networks*, vol. 106, pp. 249–259, 2018.

[3] Saptarshi Sinha, Hiroki Ohashi, and Katsuyuki Nakamura, "Class-difficulty based methods for long-tailed visual recognition," *International Journal of Computer Vision*, vol. 130, no. 10, pp. 2517–2531, aug 2022.

[4] Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge Belongie, "Class-balanced loss based on effective number of samples," in *CVPR*, 2019.

[5] Seulki Park, Jongin Lim, Younghan Jeon, and Jin Young Choi, "Influence-balanced loss for imbalanced visual classification," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2021, pp. 735–744.

[6] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár, "Focal loss for dense object detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 2, pp. 318–327, 2020.

[7] Yu-Xiong Wang, Deva Ramanan, and Martial Hebert, "Learning to model the tail," in *Advances in Neural Information Processing Systems*, I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds. 2017, vol. 30, Curran Associates, Inc.

[8] Tomasz Malisiewicz, Abhinav Gupta, and Alexei A. Efros, "Ensemble of exemplar-svms for object detection and beyond," in *2011 International Conference on Computer Vision*, 2011, pp. 89–96.

[9] Kaidi Cao, Colin Wei, Adrien Gaidon, Nikos Arechiga, and Tengyu Ma, "Learning imbalanced datasets with label-distribution-aware margin loss," in *Advances in Neural Information Processing Systems*, 2019.

[10] Bingyi Kang, Saining Xie, Marcus Rohrbach, Zhicheng Yan, Albert Gordo, Jiashi Feng, and Yannis Kalantidis, "Decoupling representation and classifier for long-tailed recognition," in *Eighth International Conference on Learning Representations (ICLR)*, 2020.

[11] Rafael Müller, Simon Kornblith, and Geoffrey E Hinton, "When does label smoothing help?," *Advances in neural information processing systems*, vol. 32, 2019.

[12] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna, "Rethinking the inception architecture for computer vision," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 2818–2826.

[13] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," *Master's thesis, Department of Computer Science, University of Toronto*, 2009.

[14] Ya Le and Xuan Yang, "Tiny imagenet visual recognition challenge," *CS 231N*, vol. 7, no. 7, pp. 3, 2015.

[15] Chun Chet Ng, Akmalul Khairi Bin Nazaruddin, Yeong Khang Lee, Xinyu Wang, Yuliang Liu, Chee Seng Chan, Lianwen Jin, Yipeng Sun, and Lixin Fan, "Icdar 2021 competition on integrated circuit text spotting and aesthetic assessment," in *International Conference on Document Analysis and Recognition*. Springer, 2021, pp. 663–677.