Enriching Visual Features via Text-driven Manifold Augmentation

Moon Ye-Bin
Dept. of EE, POSTECH
ybmoon@postech.ac.kr

Jisoo Kim
Ewha Womans University
genniferk@ewhain.net

Hongyeob Kim
Nuvilab
heimish.kyma@gmail.com

Kilho Son
Microsoft Azure
kilho.son@gmail.com

Tae-Hyun Oh
Dept. of EE and GSAI, POSTECH
taehyun@postech.ac.kr

Abstract

Recent label mix-based augmentation methods have shown their effectiveness in generalization despite their simplicity, and their favorable effects are often attributed to semantic-level augmentation. However, we found that they are vulnerable to highly skewed distribution, because scarce data classes are rarely sampled for inter-class perturbation. We propose a text-driven manifold augmentation that semantically enriches visual feature spaces, regardless of data distribution. Our method augments visual data with intra-class semantic perturbation by exploiting easy-to-understand visually mimetic words, i.e., attributes. To this end, we bridge between the text representation and a target visual feature space, and propose an efficient vector augmentation. Our experiments demonstrate that the proposed method is powerful in scarce samples with class imbalance. Note that this research is a work in progress.

1. Introduction

The effectiveness of label mix-based approaches, such as Mixup [14], CutMix [13], and manifold Mixup [11], is attributed to semantic perturbation by label mixing. However, we found that the performance of mix-based augmentation methods is noticeably degraded when training with skewed class distribution having scarce samples for non-major classes. This motivates us to seek a semantically rich data augmentation effective for long-tailed distribution.

In this work, we propose a text-driven manifold augmentation, effective for long-tail datasets. We hypothesize that general language models, *e.g.*, BERT [3] or GPT [9], learn visual information to some extent. With this hypothesis, we semantically enrich the target visual feature space by leveraging visually mimetic texts, encoded with general language models and transferred to the target space. Specifically, our method encodes meaningful attributes such as "red" and "large" to vectors by computing the difference between text embeddings with and without attributes. We add the attribute embeddings to target visual features to mimic those attributes on the visual feature space. the augmentation process of the proposed augmentation is illustrated in Fig. 1. The input feature (*e.g.*, the visual feature of "bull") is manipulated by adding the attribute vector induced by the attribute text (*e.g.*, "red"), which yields the augmented visual feature (*e.g.*, "red bull"). Thanks to the text modality properties, the augmentations by our method are symbolic, human-interpretable, and easily controllable. Also, our method perturbs data in an *intra-class* way. It means our method can densify around the training samples by extrapolating the class semantics along augmented semantic attribute axes.

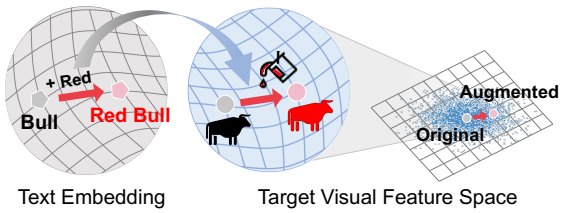

Figure 1. Illustration of the proposed method. Our method augments the target visual feature by leveraging text embedding of the visually mimetic words, which are comprehensible and semantically rich. For example, when the text of the existing class "bull" is manipulated as "red bull" by adding the attribute "red," we can get augmented visual features by reflecting the difference of text embeddings. In this way, our method densifies sparse visual feature space using various attributes text.

To empirically support that our attribute vector estimation with text embedding is reasonably designed, we analyze the embedding space with t-SNE, which demonstrates that attribute vectors lead to visually interpretable manifold augmentation of input. We also evaluate our method image classification with long-tail datasets. Our experiments demonstrate that our method is an effective and model-agnostic data augmentation method, especially in scarce data cases,

by exploiting the favors of zero-shot attributes. Our key contributions are summarized as:

- We propose an augmentation method that enriches the visual features by conveying attribute information from the text embedding to the target visual feature space.
- We demonstrate that our method is especially helpful in augmenting sparse samples in long-tail class cases.

2. Related Work

Image Data Augmentation. Data augmentation having semantic perturbation, such as Mixup [14], CutMix [13], and manifold Mixup [11], execute semantic perturbation along with label mixing. While the mixed label is known to be effective for generalization and model calibration effects [4], we found that the mix-based methods are heavily affected by class distribution due to sampling from two sources. Our method, on the other hand, is applied to all of the given samples uniformly regardless of class distribution. The proposed method densifies around the sample features by perturbing and enriching the semantic meaning of them at an intra-class level, which does not change the label.

Foundation Models. Recent foundation models have shown a successful case of reflecting human nuances with visually imitated word composition. Particularly, language models, *e.g.*, BERT [3] and GPT [9], show their ability not only in language tasks but also in vision-language multimodal tasks [10]. CLIP [8] also achieves huge success in various tasks [12] even in zero-shot recognition. In our method, we focus on estimating attribute features by exploiting BERT, GPT-2, or CLIP text encoder alone. Differ from knowledge distillation of foundation models [2], we only transfer the estimated attribute feature to augment visual features in a different space.

Long-tail Classification. In real world, visual data follow a long-tailed distribution which induces class imbalance and leads to performance degrading. The rebalancing methods [1] resample data or reweights the loss for tail classes, which have improvement in performance of the tail classes comes with the sacrifice of head class performance. Note that our method densifies all the given samples regardless of the class imbalance, which improves the performance while minimizing sacrifice of the head class.

3. Text-driven Manifold Augmentation

In image classification, the class label is typically utilized only as a supervision for measuring the loss. We, instead, propose to treat the class label as additional text information and derive semantic information from it. However, class label as a text description itself is too coarse to represent rich semantics within a class. To enrich the detailed semantics over the given coarse class texts, we leverage the attribute

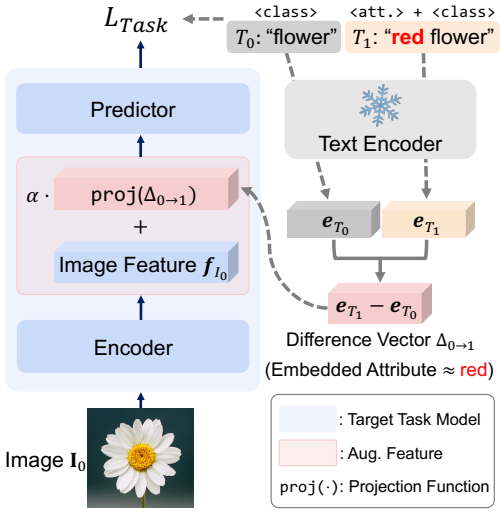

Figure 2. Overview of Text-driven Manifold Augmentation. Given flower image $\mathbf{I}_0$ and class "flower" T_0, we construct the variant text T_1 by adding the attribute "red" on T_0. $\mathbf{e}_{T_0}$ and $\mathbf{e}_{T_1}$ are computed with text encoders, and their difference vector $\mathbf{\Delta}_{0\to1} = \mathbf{e}_{T_1} - \mathbf{e}_{T_0}$ is added to the image feature $\mathbf{f}_{I_0}$ after projection $\texttt{proj}(\cdot)$ and weight α. We make the target feature space semantically rich and plausible by adding the difference vector, which embeds interpretable information.

words, such as "small size" and "brown colored," that can visually modify objects in images at the semantic level.

Main Idea. The main idea of our text-driven manifold augmentation is to densify distribution around sparse training samples on the target feature space, making it semantically rich through the difference vectors having plausible attribute information, as in Fig. 1.

Figure 2 illustrates how our method augment data. Suppose we have an image $\mathbf{I}_0$ and corresponding class label T_0. The model generally learns the target task using the image $\mathbf{I}_0$ as an input and the class label T_0 as supervision. In this work, we also consider the class label T_0 as text information and extract the embedding vector $\mathbf{e}_{T_0} \in \mathbb{R}^{d_c}$ using text encoder, *e.g.*, CLIP [8], BERT [3], or GPT-2 [9], where d_c is the text embedding dimension. The text input T_0 is formed with class name and pre-defined prompts. We also synthesize text input variant T_1 by adding color or size attribute words (*e.g.*, "red" and "big") and compute the embedding vector $\mathbf{e}_{T_1} \in \mathbb{R}^{d_c}$. Based on the word vector analogy[1] [7], we hypothesize that the relationship between T_0 and T_1 is maintained in the text embedding space, *i.e.*, the difference vector $\mathbf{\Delta}_{0\to1} = \mathbf{e}_{T_1} - \mathbf{e}_{T_0}$ would contain the information of added attributes. To exploit the difference vector from text embeddings, we design our method on the manifold.

To bridge the gap between attribute embedding and visual feature, we project the attribute embedding to the target fea-

[1]It was shown that simple algebraic operations can be performed on the word vectors, *e.g.*, king - man + woman ≈ queen on the embedding space.

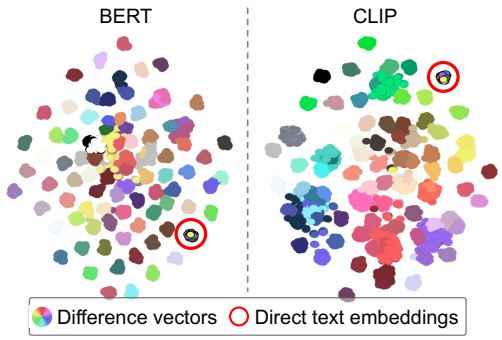

Figure 3. The t-SNE plot of difference vectors (*e.g.*, "brown dog" – "dog") projected to visual feature space. The colors of the points represent color attributes used for computing the difference vector, and we use all the classes in CIFAR-100 for this plot. As a comparison, the colored points in the red circle show direct color-text embedding (*e.g.*, "brown") projected to the visual feature space.

ture space with a learnable linear layer $\texttt{proj}(\cdot)$. A mixing weight $\alpha \in \mathbb{R}$ is introduced and randomly sampled from the clamped Normal distribution in the range over 0.1 to inject the stochasticity. The augmented feature vector $\hat{\mathbf{f}}_{I_0}$ is

$$\hat{\mathbf{f}}_{I_0} = \mathbf{f}_{I_0} + \alpha \cdot \texttt{proj}(\boldsymbol{\Delta}_{0\to 1}). \quad (1)$$

For the cases having $d_t = d_c$, we can set $\texttt{proj}(\cdot)$ operation to be an identity mapping without any learnable parameter. The class label of the augmented feature vector is still T_0.

Different from knowledge distillation [2], our method does not transfer-learn the text embeddings directly. Instead, the difference vector projected onto the target domain is injected into the target model, allowing our method to be applied to arbitrary target models. Since the visual feature augmentation is solely controlled by text, our method is human-interpretable and easily controllable.

Compared to label mix-based augmentations [11, 13, 14], our method has advantages in imbalanced data distribution. If we apply a mix-based method in the long-tailed class distribution, *i.e.*, notably skewed distribution, the class imbalance is further aggravated, and augmentation is more biased toward major classes. In contrast, our method can equally densify all the given samples since it augments each sample independently. Thus, ours can be used in general regardless of the imbalance factor of class distribution.

Difference Vector vs. Direct Text Embedding. When guessing the difference between two texts, *e.g.*, "brown X" – "X," it would be "brown." Someone may think of using the text embedding directly obtained from "brown" instead of our attribute embedding from "brown X" – "X." To understand the difference between the two representations, we visualize the difference vectors and text embeddings with BERT and CLIP text encoder in Fig. 3. While the direct text embeddings in the red circle of Fig. 3 are clustered no matter with different color-texts, the difference vectors are well clustered dependent on the color. This observation indicates that the difference vector is more effective in augmenting the visual feature space than text embedding. In addition, the difference vectors obtained from the same attribute word are similarly clustered regardless of the class "X" but slightly different. It may imply our attribute embedding has subtle difference awareness on granularity according to class.

Note that Fig. 3 presents difference vectors in the visual feature space, and we also observe similar distributions of difference vectors in the original text embedding space. This observation supports our hypothesis that general language models, *e.g.*, BERT or GPT, have learned visual information to some extent. It, also, demonstrates the visual information is properly transferred to the target visual feature space.

(a) Augmentation	Imbalance Factor (IF)		
	100	50	10
Baseline	38.39	43.33	59.29
Ours (CLIP)	40.65 (+2.26)	46.48 (+3.15)	60.17 (+0.88)
Ours (BERT)	41.10 (+2.71)	**47.17 (+3.84)**	60.67 (+1.38)
Ours (GPT-2)	**41.20 (+2.81)**	46.93 (+3.60)	60.94 (+1.65)
Cutmix [13]	37.93	43.34	59.30
Cutmix + Ours	40.22 (+2.29)	45.36 (+2.02)	**61.30 (+2.00)**
Mixup [14]	36.75	40.77	57.50
Mixup + Ours	38.40 (+1.65)	43.33 (+2.56)	59.80 (+2.30)
ManiMixup [11]	35.72	40.51	55.26
ManiMixup + Ours	38.60 (+2.88)	43.22 (+2.71)	59.35 (+4.09)

(b) Augmentation	Set of Classes (IF=100)		
	Many	Medium	Few
Baseline	71.11	38.42	3.00
Ours (CLIP)	71.14 (+0.03)	40.28 (+1.86)	7.53 (+4.53)
Ours (BERT)	70.22 (-0.89)	40.73 (+2.31)	9.41 (+6.41)
Ours (GPT-2)	70.60 (-0.51)	40.61 (+2.19)	**9.93 (+6.93)**
Cutmix	72.02	37.17	0.90
Cutmix + Ours	72.37 (+0.35)	**40.80 (+3.63)**	3.90 (+3.00)
Mixup	71.97	33.62	0.36
Mixup + Ours	71.97 (+0.00)	36.77 (+3.15)	1.83 (+1.47)
ManiMixup	72.97	29.51	0.70
ManiMixup + Ours	**73.20 (+0.23)**	36.80 (+7.29)	0.76 (+0.06)

Table 1. Long-tail classification results (%) on CIFAR-100-LT with ResNet18. (a) The accuracy with respect to the different imbalance factors, *i.e.*, IF=$\{100, 50, 10\}$. (b) The accuracy of each class set with IF=100. Baseline contains random horizontal flip, random crop and rotation, and normalization, applied in all experiments. Ours without parenthesis uses CLIP for the text encoder.

4. Experiments

Experimental Setting. We compare our method with the mix-based augmentations on CIFAR-100-LT [1] and ImageNet-LT [6], where LT stands for long-tailed distribution. Long-tail datasets usually have three sets of classes: Many-shot (more than 100 images), Medium-shot (20-100 images), and Few-shot (less than 20 images). For CIFAR-100-LT, we control the imbalance factor (IF), the ratio of samples in the head to tail class, N_1/N_K, where $N_k = |\mathcal{D}_k|$, and $\mathcal{D}_k$ is the set of samples in class $k \in \{1, \cdots, K\}$. A

Aug.	CBS	All	Many	Medium	Few
Baseline		38.39	71.11	38.42	3.00
Cutmix	✓	38.23	**71.77**	37.79	1.90
Mixup	✓	38.73	71.60	37.64	3.16
ManiMixup	✓	38.56	71.25	37.88	2.80
Ours		**40.65**	71.14	**40.28**	**7.53**

Table 2. Comparison to label mix-based augmentations with class-balanced sampling (CBS) on CIFAR-100-LT with IF=100. CBS samples two classes first and then samples data in each classes.

Method	Many	Medium	Few	All
LWS [5]	63.34	48.08	27.19	51.14
cRT [5]	61.80	46.20	27.40	49.60
cRT+Ours	62.74	48.60	29.67	51.47

Table 3. Long-tail classification accuracy(%) on ImageNet-LT with ResNext50. We compare with LWS, cRT, and ours on cRT, and color the value as best, second best, and third best.

larger value of the IF represents a more severe imbalance in data. Note that we apply each augmentation on all the samples without carefully selecting a set of classes.

Results. In Table 1 for long-tail classification on CIFAR-100-LT, the results show consistent improvement with our method. Also, our method with various text encoders achieves analogous improvement trend regardless of the IF but marginal degradation on Many class (IF=100) when using BERT or GPT-2. Compared to single usage of mix-based augmentations, ours shows higher accuracy because of uniform effects on samples regardless of class imbalance. The mix-based methods, on the other hand, sample two data points from the total dataset, where the probability that a tail class sample contributes to a resulting augmented sample is very low. Even with class-balanced sampling on mixed-based augmentation in Table 2, ours performs better, further demonstrating our effectiveness.

Particularly in Table 1(b), the mix-based methods have degraded performance in the Medium and Few-shot classes, while our method improves performance. Combining the mix-based methods with ours improves overall performance, but the tendency to sacrifice the Medium and Few-shot classes is the same as before combining.

In Table 3 for ImageNet-LT, we compare with Learnable Weight Scaling (LWS) [5] and classifier Re-Training (cRT) [5], and our method on cRT. The results show that our augmentation on cRT achieves the best performance compared to the counterparts in all classes except for the Many class, wherefrom ours achieves second best. The overall results indicate that our method is a scalable method not only effective in neural network training with skewed class distribution but also in scaling factor learning.

5. Conclusion

We propose a text-driven visual feature manifold augmentation method. Our method densifies around all the given individual visual features by adding a difference vector stem from the text embedding. While the mix-based augmentations inflict semantic perturbation in an inter-class way by label mixing, our method perturbs the semantic meaning of the visual features at an intra-class level, *i.e.*, having semantic perturbation while maintaining its class. The intra-class semantic perturbation is achieved by transferring the attribute-embedded vectors to visual feature space.

To scrutinize the design of our estimated attribute embedding, we conduct analysis with t-SNE plot. The results empirically demonstrate that our method readily enriches the sparse samples with comprehensible manipulation, since the general language models also reflect some extent of visual information. The experiment on the long-tail classification validates the effectiveness of our method, especially on the highly skewed class distribution. In this work, note that we only use color and size as attributes; thus, there would be room for further investigation of other effective attributes.

Acknowledgement. This work was supported by Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. 2020-0-00004, Development of Previsional Intelligence based on Long-term Visual Memory Network) and Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (No.2021-0-02068, Artificial Intelligence Innovation Hub).

References

[1] Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge Belongie. Class-balanced loss based on effective number of samples. In *CVPR*, 2019. 2, 3

[2] Wenliang Dai, Lu Hou, Lifeng Shang, Xin Jiang, Qun Liu, and Pascale Fung. Enabling multimodal generation on clip via vision-language knowledge distillation. *ACL*, 2022. 2, 3

[3] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *ACL*, 2018. 1, 2

[4] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *ICML*, 2017. 2

[5] Bingyi Kang, Saining Xie, Marcus Rohrbach, Zhicheng Yan, Albert Gordo, Jiashi Feng, and Yannis Kalantidis. Decoupling representation and classifier for long-tailed recognition. In *ICLR*, 2020. 4

[6] Ziwei Liu, Zhongqi Miao, Xiaohang Zhan, Jiayun Wang, Boqing Gong, and Stella X Yu. Large-scale long-tailed recognition in an open world. In *CVPR*, 2019. 3

[7] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. In *ICLR*, 2013. 2

[8] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *ICML*, 2021. 2

[9] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019. 1, 2

[10] Fawaz Sammani, Tanmoy Mukherjee, and Nikos Deligiannis. Nlx-gpt: A model for natural language explanations in vision and vision-language tasks. In *CVPR*, 2022. 2

[11] Vikas Verma, Alex Lamb, Christopher Beckham, Amir Najafi, Ioannis Mitliagkas, David Lopez-Paz, and Yoshua Bengio. Manifold mixup: Better representations by interpolating hidden states. In *ICML*, 2019. 1, 2, 3

[12] Kim Youwang, Kim Ji-Yeon, and Tae-Hyun Oh. Clip-actor: Text-driven recommendation and stylization for animating human meshes. *ECCV*, 2022. 2

[13] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *ICCV*, 2019. 1, 2, 3

[14] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *ICLR*, 2018. 1, 2, 3