TextManiA: Enriching Visual Feature by Text-driven Manifold Augmentation

Moon Ye-Bin\textsuperscript{1} Jisoo Kim\textsuperscript{2} Hongyeob Kim\textsuperscript{3} Kilho Son\textsuperscript{4} Tae-Hyun Oh\textsuperscript{1,5,6}

\textsuperscript{1}Dept. of Electrical Engineering and \textsuperscript{5}Grad. School of Artificial Intelligence, POSTECH
\textsuperscript{2}Columbia University \textsuperscript{3}Sungkyunkwan University \textsuperscript{4}Microsoft Azure
\textsuperscript{6}Institute for Convergence Research and Education in Advanced Technology, Yonsei University

https://textmania.github.io/

Abstract

We propose TextManiA, a text-driven manifold augmentation method that semantically enriches visual feature spaces, regardless of class distribution. TextManiA augments visual data with intra-class semantic perturbation by exploiting easy-to-understand visually mimetic words, i.e., attributes. This work is built on an interesting hypothesis that general language models, e.g., BERT and GPT, encompass visual information to some extent, even without training on visual training data. Given the hypothesis, TextManiA transfers pre-trained text representation obtained from a well-established large language encoder to a target visual feature space being learned. Our extensive analysis hints that the language encoder indeed encompasses visual information at least useful to augment visual representation. Our experiments demonstrate that TextManiA is particularly powerful in scarce samples with class imbalance as well as even distribution. We also show compatibility with the label mix-based approaches in evenly distributed scarce data.

1. Introduction

Learning models, e.g., neural networks, are known to perform well on visual recognition tasks when training and testing datasets present similar distributions [4]. However, their performance often degrades considerably when evaluated in subtly different distributions [69]. One effective way to enhance the generalization ability of a model against such data distribution shifts would be data augmentation [18, 86, 85, 44, 41, 73]. Augmenting data enlarges the support of the training distribution formed by given samples and yields the effect of increasing the amount of data even without additional laborious data collection. By training on augmented data, decision boundaries are smoothed, and the generalization ability of the model is improved [73].

There has been a distinctive and successful line of research for label mix-based data augmentation, such as Mixup [86], CutMix [85], and manifold Mixup [73], which are effective for model generalization and calibration [25]. The effectiveness of those label mix-based approaches is attributed to semantic perturbation by label mixing [86, 73, 85]. This is a distinctive property from other lines of data augmentation methods, e.g., [74, 44, 41, 64, 13], where they synthesize diverse virtual data that appear differently but retain class semantics of original contents. 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, i.e., long-tailed distribution. In real-world, data often exhibit long-tailed class distribution (e.g., Pareto distribution), which cannot be dealt with the prevalent mix-based approaches. This motivates us to seek a semantically rich data augmentation effective for limited data regimes, including long-tailed distribution, scarce data, and few-shot cases.

In this work, we propose TextManiA, a text-driven manifold augmentation for visual features, which is effective for long-tail classes and scarce data. Our TextManiA is based
on an interesting hypothesis that general language models, e.g., BERT [17] and GPT [55], have learned visual information to some extent that can be transferred to visual feature spaces even with no visual training data. With this hypothesis, we semantically enrich the target visual feature space to be trained by leveraging visually mimetic texts, encoded with general language models and transferred to the target space. Specifically, TextManiA 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 target visual feature space. Figure 1 illustrates the augmentation process of TextManiA. 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 generated by TextManiA are symbolic, human-interpretable, and easily controllable.

Our approach applies semantic perturbation on a different level to that of the label mix-based methods [86, 73, 85]. The mix-based methods augment a sample from a combination of two different class samples, i.e., applying semantic perturbation in an inter-class way. This further aggravates the class imbalance problem in the long-tailed (skewed) class distribution cases.1 Our TextManiA, whereas, perturbs data in an intra-class way. A sample per each class is selected, and we enrich the semantic granularity of the class using the sample, thus enabling us to better maintain the amount of augmentation balances in the long-tailed class distribution cases. Moreover, TextManiA can densify around the training samples by extrapolating the class semantics along augmented semantic attribute axes. With this, our method can be combined with the label mix-based methods to further improve performance in evenly distributed sparse data cases because they are complementary.

To empirically support that our attribute vectors transformed from text embeddings are reasonably designed, we devise two visualization-based analyses: with t-SNE [72] and a latent inversion technique. These demonstrate that attribute vectors lead to visually interpretable manifold augmentation of input. We also evaluate our method with two different tasks in scarce data regimes: few-shot object detection and image classification with deficient datasets and long-tail datasets. Our experiments demonstrate that TextManiA is an effective and model-agnostic data augmentation method, especially in scarce data cases, by exploiting the favors of zero-shot attributes. Also, additional studies show the versatility and compatibility of the design of TextManiA. Our key contributions are summarized as:

1. We propose TextManiA, which enriches the visual features by conveying attribute information from the text embedding to the target visual feature space.
2. We validate our hypothesis of the existence of embedded visual knowledge in pre-trained language encoders despite no training on visual data.
3. We demonstrate that TextManiA is especially helpful in augmenting sparse samples in long-tail class cases.
4. We show that our TextManiA is complementary to other augmentation methods, and in particular, the combination of our TextManiA and manifold Mixup [73] noticeably improves the performance in deficient data cases.

2. Related Work

We brief the related work in the following three perspectives: image data augmentation, foundation models, and target application tasks. In this work, our TextManiA augments data by leveraging the text encoder of CLIP [54], BERT [17], or GPT-22 [55]. For main target applications, we focus on long-tail and small data classification and few-shot object detection tasks in the data-scarce regimes.

Image Data Augmentation. Image data augmentation can be largely divided into whether semantic perturbation exists. Semantic perturbation, in specific, can be further split into methods with or without label mixing. Methods [18, 66, 57, 2, 24, 5, 30, 74, 44, 1, 41, 64, 13, 12] without semantic perturbation, which have no label change, contain primitive image processing and transformation operations. This includes photometric (e.g., color jitter, contrast, blur, noise, etc.) and geometric (e.g., horizontal reflection, rotation, etc.) operations, and advanced augmentations, including Cutout [18] and adaptive combinations [13, 12].

In contrast, Mixup [86], CutMix [85], and manifold Mixup [73] execute semantic perturbation along with label mixing. Mixup interpolates two whole input images pixel-wisely, CutMix interpolates a partial region of an image with another, and manifold Mixup mixes features from the images. These mix-based methods also augment labels of samples by an inter-class semantic perturbation, where labels of two different class samples are mixed. While the mixed label is known to be effective for generalization and model calibration effects [25], we found that the mix-based methods are heavily affected by class distribution due to sampling from two sources; thus, their effect is restricted to evenly distributed datasets. For datasets with skewed class distributions with tails, the sampling probabilities between major and minor classes would significantly differ, which can exaggerate biased sampling to major classes and makes minor classes more minor.

---

1For example, if data size of major classes is 10 times larger than minor classes, the probability of choosing a pair of source samples from the major classes is approximately 100 times more than that of minor classes.

2GPT-2 is the decoder-only architecture, but we use it as a text embedding extractor, so we call it text-encoder.
Our TextManiA, on the other hand, is applied to all of the given samples uniformly regardless of class distribution. TextManiA 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. Moreover, because of the different semantic granularity of perturbation between TextManiA (intra-class) and mix-based methods (inter-class), two methods can be used complementarily when class imbalance does not exist.

**Foundation Models.** Recent foundation models [84, 54, 43, 33, 56, 17, 55] have shown a successful case of reflecting human nuances with visually imitated word composition. Particularly, language models, e.g., BERT [17] and GPT [55], show their ability not only in language tasks [78] but also in vision-language multi-modal tasks [62, 22]. Contrastive Language-Image Pretraining (CLIP) [54] also achieves huge success in various tasks even in zero-shot recognition. Follow-up studies show that CLIP representation is effective in conducting other visual tasks by bridging vision and language, e.g., 2D image generation [23, 39, 35, 49], image manipulation [51, 35] and synthesis [21], and even 3D domain tasks [83, 32, 47].

In TextManiA, we focus on estimating attribute features by exploiting BERT, GPT-2, or CLIP text encoder alone. Distinctively, we only transfer the estimated attribute feature to augment visual features in a different space, which makes our work different from knowledge distillation [29] of foundation models [15, 76, 63]. Rather, our design is an instance of the module neural network structure [27, 3], where recent module-based designs procedurally train the whole model module-by-module with the guidance of the well pretrained module, e.g., [38, 60, 50, 68, 26]. Also, our work is applicable agnostically to architectures; thus, more flexibly applicable than fine-tuning of foundation models [77].

**Long-tail Classification.** In real world, visual data follow a long-tailed distribution which induces class imbalance and leads to performance degrading [81]. A representative line of the methods for long-tail classification is rebalancing [7, 14, 58], which resamples data or reweights the loss for tail classes. However, improvement in performance of the tail classes comes with the sacrifice of head class performance. Note that TextManiA densifies all the given samples regardless of the class imbalance, and whereby the model is trained with reasonable variations of training samples for every class at least, which improves the performance while minimizing sacrifice of the head class.

**Few-Shot Object Detection (FSOD).** We tackle FSOD, one of the sparse sample problems, to demonstrate the effectiveness of TextManiA and its model architecture agnostic property. FSOD handles novel object classes after the base training for object detection tasks. The model rapidly adapts to novel classes using few data by matching-based [42, 9, 79] or fine-tuning based [31, 67, 75, 53, 82] methods. TextManiA is evaluated with the fine-tuning-based FSOD approach [75], which facilitates to use general model architectures.

3. **TextManiA**

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 information, the text describing the class, and derive semantic information from it. However, class label as a text description itself is too coarse to represent rich semantics within a class. For example, a class label “dog” does not represent all the details of the description such as “small size of the brown colored dog.” To enrich the detailed semantics over the given coarse class texts, we leverage the attribute words, such as “small size” and “brown colored,” that can visually modify objects in images at the semantic level.

3.1. **Main Idea**

The main idea of TextManiA 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 depicted in Fig. 1.
Figure 2 illustrates how TextManiA augment data. Suppose we have an image \( I_0 \) and corresponding class label \( T_0 \). The model generally learns the target task using the image \( I_0 \) 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 \( e_{r_0} \in \mathbb{R}^{d_e} \) using text encoder, e.g., CLIP [54], BERT [17], or GPT-2 [55], where \( d_e \) is the text embedding dimension. For obtaining an embedding vector \( e_{r_0} \), we use the text embedding of the encoder output directly when using CLIP text encoder, or use the average vector of all the embeddings of the sentence when using other language models such as BERT or GPT-2.

Specifically, text input \( T_0 \) is formed with class name and pre-defined prompts, such as “a photo of,” “a picture of,” and “a sketch of.” We also synthesize another text input variant \( T_1 \) by adding color or size attribute words, such as “red” and “big,” and compute the embedding vector \( e_{r_1} \in \mathbb{R}^{d_e} \). Numerous variants can be created with various attribute words and their combinations, but we explain the case of one variant for convenience. Based on the word vector analogy\(^3\) [48], we hypothesize that the relationship between \( T_0 \) and \( T_1 \) is maintained in the text embedding space, i.e., the difference vector \( \Delta_{0 \rightarrow 1} = e_{r_1} - e_{r_0} \) would contain the information of added attributes (this hypothesis is validated in Sec. 3.2). To exploit the difference vector from text embeddings, we design our method on the manifold.

We can obtain such diverse attribute vectors from various attribute text templates; however, their representation space is not directly related to the visual feature space of the target model we are interested in. To bridge the gap, we project the difference vector to the target feature space with a learnable linear projection layer \( \text{proj}(\cdot) \). Then, we add the projected difference vector to the target image feature \( \hat{f}_{I_0} \in \mathbb{R}^{d_f} \) obtained from the target task encoder with the input image \( I_0 \), where \( d_f \) is the target feature space dimension. A linear layer for \( \text{proj}(\cdot) \) would be sufficient to transfer cross-modal information, referring to the cross-modal transferability under the contrastive learning case [87] and our experiments.

To inject the stochasticity, a mixing weight \( \alpha \in \mathbb{R} \) is introduced and randomly sampled from the clamped Normal distribution in the range over 0.1. Then, we have the augmented feature vector \( \hat{f}_{I_0} \) as,

\[
\hat{f}_{I_0} = f_{I_0} + \alpha \cdot \text{proj}(\Delta_{0 \rightarrow 1}). \tag{1}
\]

For the cases having \( d_e = d_c \), we can set \( \text{proj}(\cdot) \) operation to be an identity mapping without any learnable parameter.

We train the target task model with this augmented feature vector, whose class label is still \( T_0 \). We note that computing difference attribute vectors with text encoder is computationally expensive. For efficient training, we pre-compute all possible combinations of difference vectors \( \{ \Delta \} \) and store them in a look-up table because class names and attributes can be pre-determined and unchanged during training.

Different from knowledge distillation [15, 76, 63, 29, 36], TextManiA 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, TextManiA is human-interpretable and easily controllable.

Compared to label mix-based augmentations [86, 73, 85], TextManiA has advantages in imbalanced data distribution. We suppose a scenario where few samples are in one class and many samples are in another class. The augmented points by a mix-based method would be located only on the interpolation lines between the given samples, which limits the augmentation effects, as depicted in Fig. 3. If we apply a mix-based method in the long-tailed class distribution cases, i.e., notably skewed distribution, the class imbalance problem is further aggravated, and augmentation is more biased toward major classes. In contrast, TextManiA can equally densify all the given samples since it augments each sample independently. Thus, TextManiA can be used in general regardless of the imbalance factor of class distribution.

On the other hand, in another scenario with small training data but with uniform class distribution, both TextManiA and mix-based methods would increase diverse combinations of samples by augmentation in respective aspects, which leads to complementary performance improvement. This will be empirically demonstrated in Sec. 4.

### 3.2. Characteristics of Attribute Embedding

To scrutinize the relationship between the text \( T_0 \), text variant \( T_1 \) and the attribute embedding \( \Delta_{0 \rightarrow 1} \), we visualize their distribution and discuss the characteristics. We also visualize the difference vector to verify the hypothesis that the difference vector embeds its corresponding attribute.
Embedding Difference 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. 4. While the direct text embeddings in the red circle of Fig. 4 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. 4 presents difference vectors in the visual feature space, and we also observe similar distributions of difference vectors in the original text embedding space, i.e., t-SNE is not employed for clustering the direct text embeddings. 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.

Do We Need to Rule out Unrealistic Attributes? One can be curious about how TextManiA handles the unrealistic attribute, such as “blue cow.” We intentionally include such unrealistic attributes, motivated by other contexts in self- and semi-supervised learning [10, 65, 8], where they showed the strong benefit of unnatural strong augmentations to train neural networks. This observation regarding strong augmentation is consistent with the design of TextManiA containing unrealistic attributes.

Does Difference Vector Embed Attribute? To visually understand whether attribute editing is reflected while maintaining class information, we attempt to manipulate images by changing their features with the difference vectors $\Delta_{0 \to 1} = e_{f_0} - e_{f_1}$, i.e., we want to visualize the change effect between $f_0$ and $e_{a_1} = f_0 + \alpha \Delta_{0 \to 1}$ in image domain. To see the effect in image domain, we need to invert the change from $e_{a_0}$ to $e_{a_1}$ in image domain, which can be formulated as the following optimization problem,

$$\arg\min_{I_0} \| E_i(I_0) - e_{a_1} \|_1,$$

where $E_i(\cdot): I \to f$ denotes the image encoder in Fig. 2. Direct optimization in Eq. (2) is known to be difficult [88]; thus, we parameterize a given image with an image generator $G_0$ with a latent code $z$, i.e., $I(\theta) = G_0(z)$, which is known to ease the optimization [71]. Then, we can obtain the visualization by the following optimization over $\theta$

$$\arg\min_{\theta} \| E_i(G_\theta(z)) - e_{a_1} \|_1,$$

where $E_i(\cdot)$, $z$, and $a_{a_1}$ are frozen during the optimization. Since our goal is to see the move from $f_0 = E_i(I_0)$ to $e_{a_1}$ for the query $I_0$, we initialize $\theta$ and $z$ such that $G_\theta(z) \simeq I_0$ by the GAN inversion [88]. We use IC-GAN [52] for the image generator and the text embeddings are obtained from the CLIP text encoders. Note that $E_i(\cdot)$ is trained with random perturbation of transformed attributes; thus, the visualization through $E_i(\cdot)$ is different from that of CLIP encoders. Details can be found in the supplementary material.

Figure 5 shows that the manipulated image reflects the added attribute, i.e., the size of the dog is reduced by the size attribute “small,” and the bird becomes yellow by injecting

Figure 4. 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.

Figure 5. The attribute embedding visualization through image manipulation examples. We analyze how an image is manipulated when a difference vector, containing specific attribute information, is injected to the original image feature. (Top) We visualize an example of generated image given a specific class and its manipulated pair by size and color attribute, respectively. (Bottom) From left to right, we visualize the real image of “Butterfly” and its manipulated image pair with gigantic, green, and dot pattern, respectively.
the color attribute “yellow.” The manipulated results imply that 1) the difference vector indeed embeds the attributes while preserving its semantics, and 2) our augmentation on the feature space may have analogous effects to an image-level augmentation but without implementing complicated image perturbation operations. Note that these visualizations are for analysis purposes but not for competing with any existing image manipulation methods.

4. Experiments

We evaluate TextManiA in various cases presenting sparse data with different tasks: long-tail classification in Sec. 4.1, evenly distributed scarce data classification in Sec. 4.2, and few-shot object detection in Sec. 4.3. We also conduct additional studies demonstrating the effectiveness of the design of our method and the versatility of TextManiA in Sec. 4.4. Additional experimental results and details can be found in the supplementary material.

4.1. Long-tail Classification

Experimental Setting. We compare TextManiA with the mix-based augmentations on the CIFAR-100-LT [14] and ImageNet-LT [46] datasets, where LT stands for long-tailed distribution. They are artificially truncated to have a long-tail from each original dataset, CIFAR-100 [37] and ImageNet-2012 [16]. 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) [11] computed as the ratio of samples in the head to tail class, \( N_1/N_K \), where \( N_k = |D_k| \), and \( D_k \) is the set of samples belonging to the class \( k \in \{1, \ldots, K\} \). A larger value of the IF represents a more severe imbalance in data, which is more challenging. We evaluate the performance according to different IFs of 100, 50, and 10.

We utilize ResNet18 as the baseline on CIFAR-100-LT and ResNext50 on ImageNet-LT. We use the validation set of the original datasets to measure the Top-1 accuracy. Note that we apply each augmentation on all the samples without carefully selecting a set of classes in Table 1.

Results. Table 1 presents the long-tail classification results on CIFAR-100-LT, which show consistent improvement with TextManiA. Also, TextManiA with various text encoders achieves analogous improvement trend regardless of the imbalance factor but marginal degradation on Many class of IF=100 when using general language model, BERT and GPT-2. While the performance gain is from leaking pre-trained language information, it is surprising and a virtue that the language models never exposed to any image can improve the visual recognition performance. In comparison to single usage of mix-based augmentations, our method shows higher accuracy because of uniform effects of TextManiA 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, TextManiA 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 TextManiA improves performance. Combining
Table 3. Long-tail classification results (%) on ImageNet-LT with ViT, and color the value as best and second best. Baseline contains random horizontal flip, random resize crop, color jitter, and normalization, applied in all experiments.

<table>
<thead>
<tr>
<th>Augmentation</th>
<th>Set of Classes</th>
<th></th>
<th></th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>Many</td>
<td>85.34</td>
<td>70.47</td>
<td>42.80</td>
</tr>
<tr>
<td>TextManiA</td>
<td>Medium</td>
<td>85.40</td>
<td>71.75</td>
<td>48.49</td>
</tr>
<tr>
<td></td>
<td>Few</td>
<td>85.02</td>
<td>70.32</td>
<td>42.91</td>
</tr>
<tr>
<td></td>
<td>Total</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cutout [18]</td>
<td>Many</td>
<td>85.33</td>
<td>71.70</td>
<td>48.54</td>
</tr>
<tr>
<td>+ TextManiA</td>
<td>Medium</td>
<td>85.35</td>
<td>71.70</td>
<td>48.54</td>
</tr>
<tr>
<td></td>
<td>Few</td>
<td>85.02</td>
<td>70.32</td>
<td>42.91</td>
</tr>
<tr>
<td></td>
<td>Total</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cutmix [85]</td>
<td>Many</td>
<td>84.85</td>
<td>69.90</td>
<td>35.82</td>
</tr>
<tr>
<td>+ TextManiA</td>
<td>Medium</td>
<td>85.30</td>
<td>71.93</td>
<td>47.04</td>
</tr>
<tr>
<td></td>
<td>Few</td>
<td>85.30</td>
<td>71.93</td>
<td>47.04</td>
</tr>
<tr>
<td></td>
<td>Total</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mixup [86]</td>
<td>Many</td>
<td>84.49</td>
<td>70.27</td>
<td>40.20</td>
</tr>
<tr>
<td>+ TextManiA</td>
<td>Medium</td>
<td>84.55</td>
<td>69.95</td>
<td>35.72</td>
</tr>
<tr>
<td></td>
<td>Few</td>
<td>84.85</td>
<td>69.97</td>
<td>38.83</td>
</tr>
<tr>
<td></td>
<td>Total</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ManiMixup [73]</td>
<td>Many</td>
<td>85.42</td>
<td>71.98</td>
<td>46.65</td>
</tr>
<tr>
<td>+ TextManiA</td>
<td>Medium</td>
<td>85.30</td>
<td>71.93</td>
<td>47.04</td>
</tr>
<tr>
<td></td>
<td>Few</td>
<td>85.30</td>
<td>71.93</td>
<td>47.04</td>
</tr>
<tr>
<td></td>
<td>Total</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 4. Long-tail classification results (%) on ImageNet-LT with ResNet50. We compare with LWS, cRT, and TextManiA on cRT, and color the value as best and second best.

<table>
<thead>
<tr>
<th>Method</th>
<th>Many</th>
<th>Medium</th>
<th>Few</th>
<th>All</th>
</tr>
</thead>
<tbody>
<tr>
<td>LWS [34]</td>
<td>63.34</td>
<td>48.68</td>
<td>27.19</td>
<td>51.14</td>
</tr>
<tr>
<td>cRT [34]</td>
<td>61.80</td>
<td>46.20</td>
<td>27.40</td>
<td>49.60</td>
</tr>
<tr>
<td>cRT + TextManiA</td>
<td>62.74</td>
<td>48.60</td>
<td>29.67</td>
<td>51.47</td>
</tr>
</tbody>
</table>

We also evaluate our TextManiA on the large-scale dataset ImageNet-LT. In Table 3, the best and second best results are with TextManiA, which demonstrate that our augmentation method is also effective in the large-scale long-tailed data distribution, consistent with the CIFAR-100-LT results in Table 1. The improvement with TextManiA implies the importance of intra-class perturbation, which can uniformly affect the samples regardless of the skewness of the class distribution.

In Table 4, we compare with LWS [34], cRT [34], and TextManiA on cRT. LWS and cRT are one of effective methods in recent long-tailed recognition. The result shows that TextManiA on cRT achieves the best results compared to the counterparts in all classes except for the Many class, wherefrom ours achieves second best. Overall, TextManiA improves well-established works, e.g., LWS, and cRT, and it demonstrates the compatibility of our method.

4.2. Evenly Distributed Scarce Data Classification

Experimental Setting. For evaluating the effectiveness of TextManiA on the scarce dataset, we use 10% data of the CIFAR-100 [37] and Tiny-ImageNet [40] datasets, named CIFAR-100-10% and Tiny-ImageNet-10%, respectively. CIFAR-100 has 100 classes with 500 training images per class, but we only use randomly sampled 50 images per class. Tiny-ImageNet is a subset of ImageNet-1k [61] with 100k images and 200 classes, but we use 10k images (50 images per class) for simulating a small dataset. Note that the evaluation set is same with those of the original datasets.

The baseline models of scarce data classification are ResNet18 [28] and ViT-Tiny [19]. Due to the space limit, details of training can be found in the supplementary material.

Results. We demonstrate the effectiveness of TextManiA compared to mix-based augmentations on evenly distributed scarce datasets. As in Table 5 for CIFAR-100-10%, TextManiA outperforms other methods when a single augmentation is used. Furthermore, the effect is amplified when
As shown in Table 8 for VOC and Table 9 for COCO, VOC2007 and VOC2012 train/val set. For COCO, the base with AP50 metrics and train with the combination of the

This tendency is also observed with another baseline architecture in Table 6, and datasets in Table 7, implying that TextManiA is model-agnostic to be applied. The overall results demonstrate the potential of TextManiA to enrich the visual feature space using text modalities and develop more accurate and robust models in scarce data regimes.

### 4.3. Few-shot Object Detection

**Experimental Setting.** We evaluate TextManiA on the PASCAL VOC [20] and MS-COCO [45] datasets with a few-shot division following Wang et al. [75]. For VOC, we have three random splits, which have different divisions into 15 base classes and 5 novel classes among the 20 total classes, and $K = 1, 2, 3, 5, 10$ objects are sampled from the novel classes. We utilize the VOC2007 test set for evaluation with AP50 metrics and train with the combination of the VOC2007 and VOC2012 train/val set. For COCO, the base classes are disjoint with VOC classes while the remaining classes are used as novel classes, and $K = 1, 3, 5, 10, 30$ objects are sampled from the novel classes for few-shot fine-tuning. We use 5k images from the validation set in COCO for evaluation with mAP metrics and the rest for training.

The baseline [80] is the Faster R-CNN [59] trained with the base classes first and then fine-tuned with the novel classes. TextManiA is applied to the novel class samples during the fine-tuning stage. Following the prior studies, all the reported results are averaged over 10 repeated runs.

**Results.** Note that we apply TextManiA only on the classification head; thus, the quality of the regressed bounding boxes will remain similar as before applying TextManiA. As shown in Table 8 for VOC and Table 9 for COCO, TextManiA improves the AP by improving only the classification accuracy, where the result is in a similar line to the analysis [6] that classification error weighs more than localization error. The improvement is clearer when $K$ is low. The results demonstrate the applicability of TextManiA to enhance the classification accuracy of detection models.

### 4.4. Further Analyses

**Random Baseline.** In Table 10-(a), we compare our method with the Random baseline. We randomly sample a vector from a Normal distribution $\mathcal{N}(0, 1)$ and use it instead of the difference vector, i.e., augmenting visual features with random perturbations on the same manifold of visual features.

The result shows that the Random baseline improves performance by serving as intra-perturb, but marginal compared to our method considering semantics additionally, which implies that semantic information embedded in the difference vector guides the augmentation more effective direction rather than random.

### Table 7. Classification results on Tiny-ImageNet-10% with ResNet18. The configuration follows Table 5. The parentheses represents the improvement compared to the Baseline.

<table>
<thead>
<tr>
<th>Augmentation</th>
<th>Acc.</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Top-1</td>
<td>Top-5</td>
<td>Top-1</td>
<td>Top-5</td>
<td>Top-5</td>
</tr>
<tr>
<td>Baseline</td>
<td>25.94</td>
<td>50.53</td>
<td>25.94</td>
<td>50.28</td>
<td>25.94</td>
</tr>
<tr>
<td>Cutout</td>
<td>26.41</td>
<td>50.28</td>
<td>26.41</td>
<td>49.67</td>
<td>26.41</td>
</tr>
<tr>
<td>Cutmix</td>
<td>25.94</td>
<td>49.67</td>
<td>25.94</td>
<td>49.67</td>
<td>25.94</td>
</tr>
<tr>
<td>Mixup</td>
<td>29.34</td>
<td>54.10</td>
<td>29.34</td>
<td>54.10</td>
<td>29.34</td>
</tr>
<tr>
<td>ManiMixup</td>
<td>28.43</td>
<td>53.25</td>
<td>28.43</td>
<td>53.25</td>
<td>28.43</td>
</tr>
<tr>
<td><strong>TextManiA</strong></td>
<td><strong>29.37 (+3.43)</strong></td>
<td><strong>52.37 (+1.84)</strong></td>
<td><strong>29.37 (+3.43)</strong></td>
<td><strong>52.37 (+1.84)</strong></td>
<td><strong>29.37 (+3.43)</strong></td>
</tr>
<tr>
<td>Cutout + TextManiA</td>
<td>29.14</td>
<td>52.60</td>
<td>29.14</td>
<td>52.60</td>
<td>29.14</td>
</tr>
<tr>
<td>Cutmix + TextManiA</td>
<td>29.86</td>
<td>54.31</td>
<td>29.86</td>
<td>54.31</td>
<td>29.86</td>
</tr>
<tr>
<td>Mixup + TextManiA</td>
<td>31.15</td>
<td>56.71</td>
<td>31.15</td>
<td>56.71</td>
<td>31.15</td>
</tr>
<tr>
<td>ManiMixup + TextManiA</td>
<td><strong>32.39 (+6.35)</strong></td>
<td><strong>58.25 (+7.72)</strong></td>
<td><strong>32.39 (+6.35)</strong></td>
<td><strong>58.25 (+7.72)</strong></td>
<td><strong>32.39 (+6.35)</strong></td>
</tr>
</tbody>
</table>

**Table 8. Few-shot object detection results (AP50) on VOC. The value in the parentheses indicates the improvement compared to the Baseline of each split set.**

<table>
<thead>
<tr>
<th>Aug.</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>5</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>12.82</td>
<td>16.65</td>
<td>20.04</td>
<td>20.64</td>
<td>23.19</td>
</tr>
<tr>
<td><strong>TextManiA</strong></td>
<td><strong>17.74 (+4.92)</strong></td>
<td><strong>22.40 (+4.75)</strong></td>
<td><strong>23.37 (+3.33)</strong></td>
<td><strong>25.09 (+4.45)</strong></td>
<td><strong>24.22 (+1.03)</strong></td>
</tr>
<tr>
<td>1</td>
<td>Baseline</td>
<td>15.11</td>
<td>18.82</td>
<td>22.61</td>
<td>21.97</td>
</tr>
<tr>
<td><strong>TextManiA</strong></td>
<td><strong>21.94 (+6.83)</strong></td>
<td><strong>26.44 (+7.62)</strong></td>
<td><strong>23.66 (+1.05)</strong></td>
<td><strong>25.88 (+3.91)</strong></td>
<td><strong>25.14 (+1.40)</strong></td>
</tr>
<tr>
<td>2</td>
<td>Baseline</td>
<td>10.86</td>
<td>14.22</td>
<td>18.67</td>
<td>19.34</td>
</tr>
<tr>
<td><strong>TextManiA</strong></td>
<td><strong>14.64 (+3.78)</strong></td>
<td><strong>18.49 (+4.27)</strong></td>
<td><strong>23.06 (+4.61)</strong></td>
<td><strong>24.44 (+3.72)</strong></td>
<td><strong>24.44 (+1.95)</strong></td>
</tr>
<tr>
<td>3</td>
<td>Baseline</td>
<td>12.49</td>
<td>16.90</td>
<td>18.84</td>
<td>20.61</td>
</tr>
<tr>
<td><strong>TextManiA</strong></td>
<td><strong>16.65 (+4.16)</strong></td>
<td><strong>22.26 (+5.36)</strong></td>
<td><strong>23.16 (+4.32)</strong></td>
<td><strong>25.33 (+5.72)</strong></td>
<td><strong>25.08 (+1.73)</strong></td>
</tr>
</tbody>
</table>

**Table 9. Few-shot object detection results (mAP) on COCO. The configuration follows Table 8.**

<table>
<thead>
<tr>
<th>Aug.</th>
<th>Many</th>
<th>Medium</th>
<th>Few</th>
<th>IF=100</th>
<th>IF=50</th>
<th>IF=10</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>71.11</td>
<td>38.42</td>
<td>3.00</td>
<td>38.39</td>
<td>43.33</td>
<td>59.29</td>
</tr>
<tr>
<td><strong>TextManiA</strong></td>
<td><strong>70.22 (+0.96)</strong></td>
<td><strong>40.73 (+1.81)</strong></td>
<td><strong>9.41 (+1.70)</strong></td>
<td><strong>41.10 (+0.92)</strong></td>
<td><strong>47.17 (+0.82)</strong></td>
<td></td>
</tr>
<tr>
<td>(a) Random</td>
<td>71.37</td>
<td>38.55</td>
<td>2.90</td>
<td>38.43</td>
<td>43.28</td>
<td>60.39</td>
</tr>
<tr>
<td><strong>TextManiA</strong></td>
<td><strong>70.22 (+0.96)</strong></td>
<td><strong>40.73 (+1.81)</strong></td>
<td><strong>9.41 (+1.70)</strong></td>
<td><strong>41.10 (+0.92)</strong></td>
<td><strong>47.17 (+0.82)</strong></td>
<td></td>
</tr>
<tr>
<td>Direct</td>
<td>71.34</td>
<td>38.64</td>
<td>4.32</td>
<td>38.66</td>
<td>43.44</td>
<td>59.82</td>
</tr>
<tr>
<td>Concat.</td>
<td>68.02</td>
<td>35.82</td>
<td>5.35</td>
<td>36.98</td>
<td>42.68</td>
<td>59.44</td>
</tr>
</tbody>
</table>

**Table 10. Comparison to (a) random perturbation, and (b) direct text and concatenated embeddings on CIFAR-100-LT.**
we apply our method during linear probing of the model. In Table 11, we test VL-LTR [70], the state-of-the-art model in long-tail classification, on CIFAR-100. In Table 12, we use a CLIP image encoder [54] with various architectures as the baseline model and linear-probe the model on both 10% and full data of CIFAR-100. The results demonstrate that TextManiA is compatible with linear-probed CLIP and VL-LTR models.

### Effectiveness of Difference Vectors.
While we use the difference vectors by subtracting the embeddings with and without attribute words for TextManiA, there could be another way to extract the attribute information. In Table 10-(b), we compare with counterparts, direct text embedding (Direct.) and concatenated embeddings (Concat.). For the Direct method, we use the text embedding computed from the attribute word directly instead of the difference vector. For the Concat method, we concatenate the text embeddings from with and without attribute words, e.g., ["bull"]|"red bull"], and use it instead of the difference vector.

The results show that using difference vector (TextManiA) outperforms using direct text embedding or concatenated embeddings, and imply that remaining contextual information after subtraction plays an important role in doing intra-perturbation in a semantic way. Although the word “blue” can function as both an adjective and a noun, its exact role in a sentence cannot be determined solely based on the word itself. Our intention of subtraction is for attribute words to act as a modifier in the sentence motivated by word analogy. When we computed the cosine similarity, embeddings derived directly from “red” and those obtained by word analogy. When we computed the cosine similarity, embeddings derived directly from “red” and those obtained from the difference exhibited low similarity because they contain different contextual information despite the same origin of a word.

### Linear Probing with Advanced Models.
Further demonstrating the compatibility of TextManiA, we apply our method during linear probing of the model. In Table 11, we test VL-LTR [70], the state-of-the-art model in long-tail classification, on CIFAR-100. In Table 12, we use a CLIP image encoder [54] with various architectures as the baseline model and linear-probe the model on both 10% and full data of CIFAR-100. The results demonstrate that TextManiA is compatible with linear-probed CLIP and VL-LTR models.

### Ablation Study on Attributes.
In TextManiA, we have considered color and size attributes. To confirm the effect of each attribute, we conduct an ablation study on attributes in Table 13. The result shows that while each attribute brings non-trivial gain, using both brings more gain. We believe that there are additional attributes we could use and a more effective method for selecting appropriate attributes, but leave it for future work.

### 5. Conclusion
To mitigate the scarce data problem in long-tailed data distribution, small dataset, and few-shot cases, we propose a text-driven visual feature manifold augmentation method, TextManiA. 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, TextManiA 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 visualization-based analyses: t-SNE plot and simple manipulation tests. The results empirically demonstrate that TextManiA 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. We additionally show the compatibility of TextManiA with other augmentation methods or other models in scarce data cases and during linear probing. 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.

### Acknowledgment.
This work was partially supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (No.2021-0-02068, Artificial Intelligence Innovation Hub; No.2022-0-00124, Development of Artificial Intelligence Technology for Self-Improving Competency-Aware Learning Capabilities; No. 2020-0-00004, Development of Previsional Intelligence based on Long-term Visual Memory Network).

<table>
<thead>
<tr>
<th>CLIP Arch.</th>
<th>Aug.</th>
<th>ZS</th>
<th>LP-10%</th>
<th>LP-Full</th>
</tr>
</thead>
<tbody>
<tr>
<td>ResNet50</td>
<td>Baseline</td>
<td>39.47</td>
<td>50.18</td>
<td>63.64</td>
</tr>
<tr>
<td></td>
<td>TextManiA</td>
<td>-</td>
<td>52.83</td>
<td>64.17</td>
</tr>
<tr>
<td>ResNet101</td>
<td>Baseline</td>
<td>45.17</td>
<td>57.37</td>
<td>68.60</td>
</tr>
<tr>
<td></td>
<td>TextManiA</td>
<td>-</td>
<td>59.49</td>
<td>69.12</td>
</tr>
<tr>
<td>ViT-B</td>
<td>Baseline</td>
<td>58.21</td>
<td>73.30</td>
<td>79.99</td>
</tr>
<tr>
<td></td>
<td>TextManiA</td>
<td>-</td>
<td>73.35</td>
<td>79.58</td>
</tr>
</tbody>
</table>

Table 12. Classification results (%) of CLIP with zero-shot (ZS) and linear-probe (LP) on Full and 10% CIFAR-100. We apply our TextManiA to the linear-probed CLIP.

<table>
<thead>
<tr>
<th>Model</th>
<th>LP-Full</th>
</tr>
</thead>
<tbody>
<tr>
<td>VL-LTR</td>
<td>61.04</td>
</tr>
<tr>
<td>+TextManiA</td>
<td>61.82</td>
</tr>
</tbody>
</table>

Table 11. Comparison between the SOTA model with and without TextManiA during linear probing on CIFAR-100.

<table>
<thead>
<tr>
<th>Color Size Acc.</th>
<th>31.10</th>
<th>33.48</th>
<th>33.89</th>
<th>34.52</th>
</tr>
</thead>
</table>

Table 13. Ablation study on the attributes with CIFAR-100-10%.
References


[33] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *ICML*, 2021. 3


transferrable visual models from natural language supervision. In *ICML*, 2021. 2, 3, 4, 9
[80] Lu Yang, He Jiang, Qing Song, and Jun Guo. A survey on long-tailed visual recognition. IJCV, 2022. 3
[81] 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, 4, 6, 7