TextManiA: Enriching Visual Feature by Text-driven Manifold Augmentation

Moon Ye-Bin[1] Jisoo Kim[2] Hongyeob Kim[3] Kilho Son[4] Tae-Hyun Oh[1,5,6*]

[1]Dept. of Electrical Engineering and [5]Grad. School of Artificial Intelligence, POSTECH
[2]Columbia University [3]Sungkyunkwan University [4]Microsoft Azure
[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.*

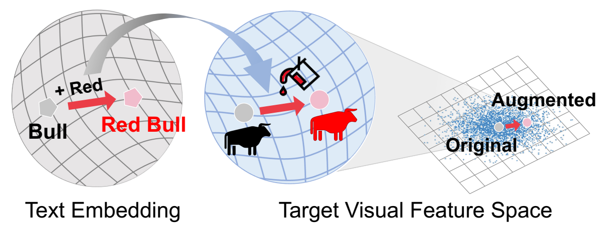

Figure 1. Illustration of `TextManiA`. Our method augments the target visual feature by leveraging text embedding of the visually mimetic words, which are comprehensible and semantically rich. For example, when the text of the existing class "bull" is manipulated as "red bull" by adding the attribute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

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

*Acknowledgment. This work was partially supported by 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).

distribution, scarce data, and few-shot cases.

In this work, we propose `TextManiA`, a text-driven manifold augmentation, which is effective for long-tail classes and scarce data. `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. Our key contributions are summarized as:

- We propose `TextManiA`, which enriches the visual features by conveying attribute information from the text embedding to the target visual feature space.
- We validate our hypothesis of the existence of embedded visual knowledge in pre-trained language encoders despite no training on visual data.
- We demonstrate that `TextManiA` is especially helpful in augmenting sparse samples in long-tail class cases.
- 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-2 [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.

[1] For 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.

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 pre-trained 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.

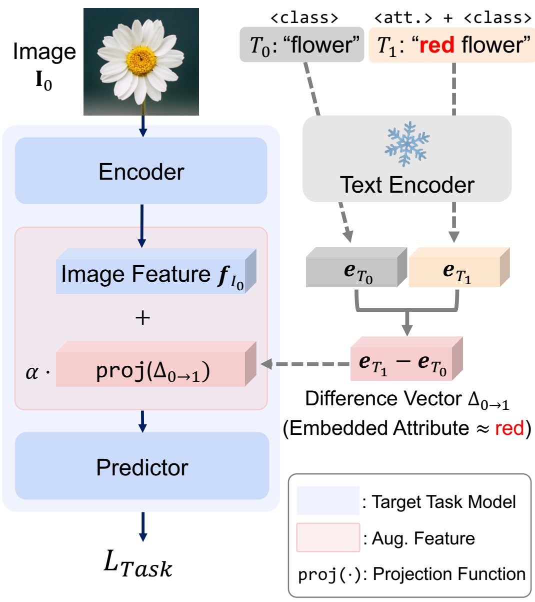

Figure 2. Overview of TextManiA. Given flower image $\mathbf{I}_0$ and class "flower" T_0, we construct the variant text T_1 by adding the attribute "red" on T_0. Embeddings of T_0 and T_1 are computed with text encoders, *e.g.* CLIP [54], BERT [17], or GPT-2 [55] and their difference vector, $\mathbf{\Delta}_{0\rightarrow1} = \mathbf{e}_{T_1} - \mathbf{e}_{T_0}$ 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.

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 $\mathbf{I}_0$ 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 [54], BERT [17], or GPT-2 [55], where d_c is the text embedding dimension. For obtaining an embedding vector $\mathbf{e}_{T_0}$, 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 $\mathbf{e}_{T_1} \in \mathbb{R}^{d_c}$. 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[2] [48], we hypothesize that the relationship between T_0 and T_1 is maintained in the text embedding space, *i.e.*, the difference vector $\mathbf{\Delta}_{0\to 1} = \mathbf{e}_{T_1} - \mathbf{e}_{T_0}$ 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 $\texttt{proj}(\cdot)$. A linear layer would be sufficient to transfer cross-modal information, referring to our experimental results and the cross-modal transferability of the multi-modal contrastive model [87]. Then, we add the projected difference vector to the target image feature $\mathbf{f}_{I_0} \in \mathbb{R}^{d_t}$ obtained from the target task encoder with the input image $\mathbf{I}_0$, where d_t is the target feature space dimension.

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 $\hat{\mathbf{f}}_{I_0}$ as,

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

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

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 $\{\mathbf{\Delta}\}$ and store them in a look-up table because class names and attributes can be pre-determined and unchanged during training.

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

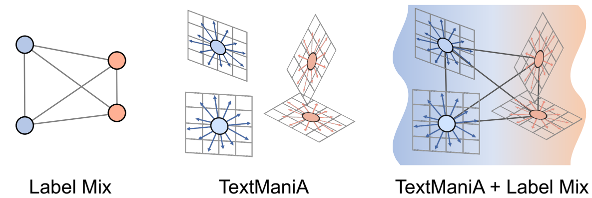

Figure 3. Comparison of a typical label mix-based augmentation, `TextManiA`, and their combination, in a sparse data case. Given two samples for two classes, the augmented samples by a label mix-based method are at points lying on six lines resulting from the combinations of samples, *i.e.*, inter-class perturbation. `TextManiA` densifies the sample distribution along semantic attribute axes in an intra-class. The combination of them yields synergy of their respective advantages.

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 `TextManiA` 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 and text variant T_1 and the attribute embedding $\mathbf{\Delta}_{0\to 1}$, 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

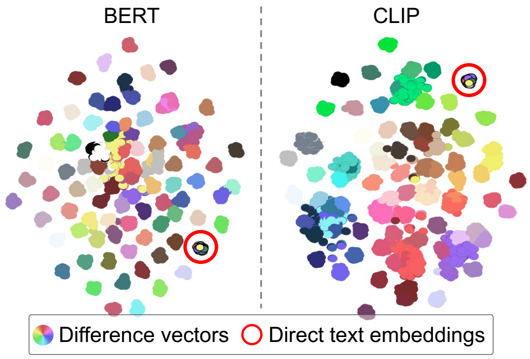

Figure 4. The t-SNE plot of difference vectors (*e.g.*, "brown dog" – "dog") projected to visual feature space. The colors of the points represent color attributes used for computing the difference vector, and we use all the classes in CIFAR-100 for this plot. As a comparison, the colored points in the 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 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. 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 $\mathbf{\Delta}_{0\to1} = \mathbf{e}_{T_1} - \mathbf{e}_{T_0}$, *i.e.*, we want to visualize the change effect between $\mathbf{f}_{\mathbf{I}_0}$ and $\mathbf{e}_{a_1} = \mathbf{f}_{\mathbf{I}_0} + \alpha\mathbf{\Delta}_{0\to1}$ in image domain. To see the effect in image domain, we need to invert the change from $\mathbf{e}_{\mathbf{I}_0}$ to $\mathbf{e}_{a_1}$ in image domain, which can be formulated as the following optimization problem,

$$\arg\min_{\mathbf{I}} \ \|E_i(\mathbf{I}) - \mathbf{e}_{a_1}\|_1, \tag{2}$$

where $E_i(\cdot) : \mathbf{I} \to \mathbf{f}$ 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_θ with a latent code $\mathbf{z}$, *i.e.*, $\mathbf{I}(\theta) = G_\theta(\mathbf{z})$, which is known to ease the optimization [71]. Then, we can obtain the visualization by the following optimization over θ

$$\arg\min_{\theta} \ \|E_i(\mathbf{I}(\theta)) - \mathbf{e}_{a_1}\|_1. \tag{3}$$

Since the goal is to see the move from $\mathbf{f}_{\mathbf{I}_0} = E_i(\mathbf{I}_0)$ to $\mathbf{e}_{a_1}$, we initialize θ and $\mathbf{z}$ such that $G_\theta(\mathbf{z}) = \mathbf{I}'_0 \simeq \mathbf{I}_0$ by the GAN inversion technique [88]. Note that the latent vector $\mathbf{z}$, the encoders $E_i(\cdot)$, and the augmented visual embedding vector $\mathbf{e}_{a_1}$ are frozen during the optimization. In this work, we use IC-GAN [52] for the image generator and the text embeddings are obtained from the CLIP text 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 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.

(a) Augmentation	Imbalance Factor (IF)		
	100	50	10
Baseline	38.39	43.33	59.29
TextManiA (CLIP)	40.65 (+2.26)	46.48 (+3.15)	60.17 (+0.88)
TextManiA (BERT)	41.10 (+2.71)	**47.17 (+3.84)**	60.67 (+1.38)
TextManiA (GPT-2)	**41.20 (+2.81)**	46.93 (+3.60)	60.94 (+1.65)
Cutout [18]	37.51	42.28	59.26
Cutout + TextManiA	40.35 (+2.84)	45.48 (+3.20)	**61.31 (+2.05)**
Cutmix [85]	37.93	43.34	59.30
Cutmix + TextManiA	40.22 (+2.29)	45.36 (+2.02)	61.30 (+2.00)
Mixup [86]	36.75	40.77	57.50
Mixup + TextManiA	38.40 (+1.65)	43.33 (+2.56)	59.80 (+2.30)
ManiMixup [73]	35.72	40.51	55.26
ManiMixup + TextManiA	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
TextManiA (CLIP)	71.14 (+0.03)	40.28 (+1.86)	7.53 (+4.53)
TextManiA (BERT)	70.22 (-0.89)	40.73 (+2.31)	9.41 (+6.41)
TextManiA (GPT-2)	70.60 (-0.51)	40.61 (+2.19)	**9.93 (+6.93)**
Cutout	71.54	35.94	1.06
Cutout + TextManiA	71.94 (+0.83)	**40.97 (+2.55)**	4.03 (+3.03)
Cutmix	72.02	37.17	0.90
Cutmix + TextManiA	72.37 (+0.35)	40.80 (+3.63)	3.90 (+3.00)
Mixup	71.97	33.62	0.36
Mixup + TextManiA	71.97 (+0.00)	36.77 (+3.15)	1.83 (+1.47)
ManiMixup	72.97	29.51	0.70
ManiMixup + TextManiA	**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. TextManiA without parenthesis uses CLIP for the text encoder.

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
TextManiA		**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.

4. Experiments

In this section, we demonstrate the effectiveness of TextManiA in scarce data regimes. We evaluate our method in various cases presenting sparse data with different tasks: long-tail classification in Sec. 4.1, scarce data classification in Sec. 4.2, and few-shot object detection (FSOD) in Sec. 4.3. We also conduct additional experiments demonstrating the compatibility of TextManiA and ablation study on attributes in Sec. 4.4. Additional 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 $N_k = |\mathcal{D}_k|$, and $\mathcal{D}_k$ is the set of samples belonging to the class $k \in \{1, \cdots, K\}$. 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. 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 the mix-based methods with TextManiA improves overall performance, but the tendency to sacrifice the Medium and Few-shot classes is the same as before combining. Additionally, while Cutout has performance degradation due to the information loss [85], it is not affected by skewness due to no mix between inter-classes; thus, the performance is higher than the mix-based one in the long-tailed distribution.

In Table 3 for ImageNet-LT, we compare with LWS [34], cRT [34], and TextManiA on cRT. The result shows that our TextManiA on cRT achieves the best results compared to the counterparts in all classes except for the Many class, wherefrom ours achieves second best. This is consistent with the result in Table 3 in the main paper regardless of different

Method	Many	Medium	Few	All
LWS [34]	63.34	48.08	27.19	51.14
cRT [34]	61.80	46.20	27.40	49.60
cRT+TextManiA	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 TextManiA on cRT, and color the value as best, second best, and third best. The models are trained with a batch size of 512.

Augmentation	Acc. Top-1	Acc. Top-5
Baseline	31.10	59.14
Cutout	32.03	60.53
Cutmix	32.43	61.04
Mixup	32.72	62.47
ManiMixup	33.74	63.29
TextManiA	**34.52 (+3.42)**	**65.74 (+6.60)**
Cutout + TextManiA	33.91 (+2.81)	61.58 (+2.44)
Cutmix + TextManiA	35.61 (+4.51)	63.82 (+4.68)
Mixup + TextManiA	37.97 (+6.87)	66.75 (+7.61)
ManiMixup + TextManiA	**38.02 (+6.92)**	**67.28 (+8.14)**

Table 4. Classification results on CIFAR-100-10% with ResNet18. Baseline represents random horizontal flip, random crop, and normalization, which are basically applied in all experiments. The parentheses stands for the improvement compared to the Baseline.

Augmentation	Acc. Top-1	Acc. Top-5
Baseline	65.37	89.82
Cutout	69.17	91.12
Cutmix	69.82	91.76
Mixup	67.54	90.23
TextManiA	**70.81 (+5.44)**	**92.37 (+2.55)**
Cutout + TextManiA	69.71 (+4.34)	91.32 (+1.50)
Cutmix + TextManiA	**71.05 (+5.68)**	**92.22 (+2.40)**
Mixup + TextManiA	70.56 (+5.19)	91.58 (+1.76)

Table 5. Classification results on CIFAR-100-10% with VIT-Tiny. The configuration follows Table 4. The parentheses stands for the improvement compared to the Baseline.

batch sizes. The overall results show that our TextManiA improves well-established works, *e.g.*, LWS, and cRT.

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

Augmentation	Acc. Top-1	Acc. Top-5
Baseline	25.94	50.53
Cutout	26.41	50.28
Cutmix	25.94	49.67
Mixup	29.34	**54.10**
ManiMixup	28.43	53.25
TextManiA	**29.37 (+3.43)**	52.37 (+1.84)
Cutout + TextManiA	29.14 (+3.20)	52.60 (+2.07)
Cutmix + TextManiA	29.86 (+3.92)	54.31 (+3.78)
Mixup + TextManiA	31.15 (+5.21)	56.71 (+6.18)
ManiMixup + TextManiA	**32.39 (+6.35)**	**58.25 (+7.72)**

Table 6. Classification results on Tiny-ImageNet-10% with ResNet18. The configuration follows Table 4. The parentheses represents the improvement compared to the Baseline.

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 4 for CIFAR-100-10%, TextManiA outperforms other methods when a single augmentation is used. Furthermore, the effect is amplified when our method and mix-based methods are combined, with particularly good compatibility with Manifold Mixup. The results demonstrate the importance of intra-class semantic perturbation along with inter-class in scarce data settings. This tendency is also observed with another baseline architecture in Table 5, and datasets in Table 6, 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 divison 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

Split	Aug.	K- shot				
		1	2	3	5	10
All	Baseline	12.82	16.65	20.04	20.64	23.19
	`TextManiA`	**17.74** (+4.92)	**22.40** (+5.75)	**23.37** (+3.33)	**25.09** (+4.45)	**24.22** (+1.03)
1	Baseline	15.11	18.82	22.61	21.97	23.74
	`TextManiA`	**21.94** (+6.83)	**26.44** (+7.62)	**23.66** (+1.05)	**25.88** (+3.91)	**25.14** (+1.40)
2	Baseline	10.86	14.22	18.67	19.34	22.49
	`TextManiA`	**14.64** (+3.78)	**18.49** (+4.27)	**23.28** (+4.61)	**23.06** (+3.72)	**24.44** (+1.95)
3	Baseline	12.49	16.90	18.84	20.61	23.35
	`TextManiA`	**16.65** (+4.16)	**22.26** (+5.36)	**23.16** (+4.32)	**26.33** (+5.72)	**25.08** (+1.73)

Table 7. Few-shot object detection results (AP50) on VOC. The value in the parentheses indicates the improvement compared to the Baseline of each split set.

Aug.	K- shot				
	1	3	5	10	30
Baseline	3.43	4.66	6.10	9.11	12.78
`TextManiA`	**5.39** (+1.96)	**6.47** (+1.81)	**7.80** (+1.70)	**10.03** (+0.92)	**13.60** (+0.82)

Table 8. Few-shot object detection results (mAP) on COCO. The configuration follows Table 7.

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 7 for VOC and Table 8 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. Additional Experiments

Model	LP-Full
VL-LTR	61.04
+`TextManiA`	**61.82**

Table 9. Comparison between the SOTA model with and without `TextManiA` during linear probing on CIFAR-100.

Linear Probing with Advanced Models. Further demonstrating the compatibility of `TextManiA`, we apply our method during linear probing of the model. In Table 9, we test VL-LTR [70], the state-of-the-art model in long-tail classification, on CIFAR-100. In Table 10, 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.

CLIP Arch.	Aug.	ZS	LP-10%	LP-Full
ResNet50	Baseline	39.47	50.18	63.64
	`TextManiA`	-	**52.83**	**64.17**
ResNet101	Baseline	45.17	57.37	68.60
	`TextManiA`	-	**59.49**	**69.12**
ViT-B	Baseline	58.21	73.30	**79.99**
	`TextManiA`	-	**73.35**	79.58

Table 10. 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.

Color	Size	Acc.
		31.10
✓		33.48
	✓	33.89
✓	✓	**34.52**

Table 11. Ablation study on the attributes with CIFAR-100-10%.

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 11. 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.

References

[1] Alexander A Alemi, Ian Fischer, Joshua V Dillon, and Kevin Murphy. Deep variational information bottleneck. *ICLR*, 2017. 2

[2] Guozhong An. The effects of adding noise during backpropagation training on a generalization performance. *Neural computation*, 8(3):643–674, 1996. 2

[3] Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Dan Klein. Neural module networks. In *CVPR*, 2016. 3

[4] Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Machine learning*, 79(1):151–175, 2010. 1

[5] Chris M Bishop. Training with noise is equivalent to tikhonov regularization. *Neural computation*, 7(1):108–116, 1995. 2

[6] Ali Borji and Seyed Mehdi Iranmanesh. Empirical upper bound in object detection and more. *arXiv preprint arXiv:1911.12451*, 2019. 8

[7] Mateusz Buda, Atsuto Maki, and Maciej A Mazurowski. A systematic study of the class imbalance problem in convolutional neural networks. *Neural networks*, 2018. 3

[8] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *ICML*, 2020. 5

[9] Tung-I Chen, Yueh-Cheng Liu, Hung-Ting Su, Yu-Cheng Chang, Yu-Hsiang Lin, Jia-Fong Yeh, Wen-Chin Chen, and Winston Hsu. Dual-awareness attention for few-shot object detection. *IEEE TMM*, 2021. 3

[10] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *CVPR*, 2021. 5

[11] Peng Chu, Xiao Bian, Shaopeng Liu, and Haibin Ling. Feature space augmentation for long-tailed data. In *ECCV*, 2020. 6

[12] Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. Autoaugment: Learning augmentation policies from data. In *CVPR*, 2019. 2

[13] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *CVPRW*, 2020. 1, 2

[14] Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge Belongie. Class-balanced loss based on effective number of samples. In *CVPR*, 2019. 3, 6

[15] 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. 3, 4

[16] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 6

[17] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *ACL*, 2018. 2, 3, 4

[18] Terrance DeVries and Graham W Taylor. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*, 2017. 1, 2, 6

[19] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021. 7

[20] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *IJCV*, 2010. 7

[21] Kevin Frans, Lisa B Soros, and Olaf Witkowski. Clipdraw: Exploring text-to-drawing synthesis through language-image encoders. *arXiv preprint arXiv:2106.14843*, 2021. 3

[22] Rinon Gal, Yuval Alaluf, Yuval Atzmon, Or Patashnik, Amit H Bermano, Gal Chechik, and Daniel Cohen-Or. An image is worth one word: Personalizing text-to-image generation using textual inversion. *ICLR*, 2023. 3

[23] Rinon Gal, Or Patashnik, Haggai Maron, Amit H Bermano, Gal Chechik, and Daniel Cohen-Or. Stylegan-nada: Clip-guided domain adaptation of image generators. *ACM TOG*, 2022. 3

[24] Caglar Gulcehre, Marcin Moczulski, Misha Denil, and Yoshua Bengio. Noisy activation functions. In *ICML*, pages 3059–3068. PMLR, 2016. 2

[25] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *ICML*, 2017. 1, 2

[26] Tanmay Gupta and Aniruddha Kembhavi. Visual programming: Compositional visual reasoning without training. In *CVPR*, 2023. 3

[27] Bart LM Happel and Jacob MJ Murre. Design and evolution of modular neural network architectures. *Neural networks*, 7(6-7):985–1004, 1994. 3

[28] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 7

[29] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. 3, 4

[30] Lasse Holmstrom and Petri Koistinen. Using additive noise in back-propagation training. *IEEE transactions on neural networks*, 3(1):24–38, 1992. 2

[31] Hanzhe Hu, Shuai Bai, Aoxue Li, Jinshi Cui, and Liwei Wang. Dense relation distillation with context-aware aggregation for few-shot object detection. In *CVPR*, 2021. 3

[32] Ajay Jain, Ben Mildenhall, Jonathan T Barron, Pieter Abbeel, and Ben Poole. Zero-shot text-guided object generation with dream fields. In *CVPR*, 2022. 3

[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

[34] 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. 6, 7

[35] Gwanghyun Kim, Taesung Kwon, and Jong Chul Ye. Diffusionclip: Text-guided diffusion models for robust image manipulation. In *CVPR*, 2022. 3

[36] Youmin Kim, Jinbae Park, YounHo Jang, Muhammad Ali, Tae-Hyun Oh, and Sung-Ho Bae. Distilling global and local logits with densely connected relations. In *ICCV*, 2021. 4

[37] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009. 6, 7

[38] Ashish Kumar, Zipeng Fu, Deepak Pathak, and Jitendra Malik. Rma: Rapid motor adaptation for legged robots. In *Robotics: Science and Systems*, 2021. 3

[39] Gihyun Kwon and Jong Chul Ye. Clipstyler: Image style transfer with a single text condition. In *CVPR*, 2022. 3

[40] Ya Le and Xuan Yang. Tiny imagenet visual recognition challenge. *CS 231N*, 2015. 7

[41] Boyi Li, Felix Wu, Ser-Nam Lim, Serge Belongie, and Kilian Q Weinberger. On feature normalization and data augmentation. In *CVPR*, 2021. 1, 2

[42] Bohao Li, Boyu Yang, Chang Liu, Feng Liu, Rongrong Ji, and Qixiang Ye. Beyond max-margin: Class margin equilibrium for few-shot object detection. In *CVPR*, 2021. 3

[43] Liunian Harold Li, Pengchuan Zhang, Haotian Zhang, Jianwei Yang, Chunyuan Li, Yiwu Zhong, Lijuan Wang, Lu Yuan, Lei Zhang, Jenq-Neng Hwang, et al. Grounded language-image pre-training. In *CVPR*, 2022. 3

[44] Pan Li, Da Li, Wei Li, Shaogang Gong, Yanwei Fu, and Timothy M Hospedales. A simple feature augmentation for domain generalization. In *ICCV*, 2021. 1, 2

[45] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014. 7

[46] 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. 6

[47] Oscar Michel, Roi Bar-On, Richard Liu, Sagie Benaim, and Rana Hanocka. Text2mesh: Text-driven neural stylization for meshes. In *CVPR*, 2022. 3

[48] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. In *ICLR*, 2013. 4

[49] Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. In *ICML*, 2022. 3

[50] Tae-Hyun Oh, Tali Dekel, Changil Kim, Inbar Mosseri, William T Freeman, Michael Rubinstein, and Wojciech Matusik. Speech2face: Learning the face behind a voice. In *CVPR*, 2019. 3

[51] Or Patashnik, Zongze Wu, Eli Shechtman, Daniel Cohen-Or, and Dani Lischinski. Styleclip: Text-driven manipulation of stylegan imagery. In *ICCV*, 2021. 3

[52] Guim Perarnau, Joost van de Weijer, Bogdan Raducanu, and Jose M. Álvarez. Invertible conditional gans for image editing. In *CVPR*, 2016. 5

[53] Limeng Qiao, Yuxuan Zhao, Zhiyuan Li, Xi Qiu, Jianan Wu, and Chi Zhang. Defrcn: Decoupled faster r-cnn for few-shot object detection. In *ICCV*, 2021. 3

[54] 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, 3, 4, 8

[55] 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. 2, 3, 4

[56] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *ICML*, 2021. 3

[57] Russell Reed and Robert J MarksII. *Neural smithing: supervised learning in feedforward artificial neural networks*. Mit Press, 1999. 2

[58] Mengye Ren, Wenyuan Zeng, Bin Yang, and Raquel Urtasun. Learning to reweight examples for robust deep learning. In *ICML*, 2018. 3

[59] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NeurIPS*, 2015. 7

[60] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, 2022. 3

[61] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *IJCV*, 2015. 7

[62] Fawaz Sammani, Tanmoy Mukherjee, and Nikos Deligiannis. Nlx-gpt: A model for natural language explanations in vision and vision-language tasks. In *CVPR*, 2022. 3

[63] Gyungin Shin, Weidi Xie, and Samuel Albanie. Namedmask: Distilling segmenters from complementary foundation models. *arXiv preprint arXiv:2209.11228*, 2022. 3, 4

[64] Patrice Y Simard, Yann A LeCun, John S Denker, and Bernard Victorri. Transformation invariance in pattern recognition—tangent distance and tangent propagation. In *Neural networks: tricks of the trade*, pages 239–274. Springer, 1998. 1, 2

[65] Kihyuk Sohn, David Berthelot, Nicholas Carlini, Zizhao Zhang, Han Zhang, Colin A Raffel, Ekin Dogus Cubuk, Alexey Kurakin, and Chun-Liang Li. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. In *NeurIPS*, 2020. 5

[66] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 2014. 2

[67] Bo Sun, Banghuai Li, Shengcai Cai, Ye Yuan, and Chi Zhang. Fsce: Few-shot object detection via contrastive proposal encoding. In *CVPR*, 2021. 3

[68] Kim Sung-Bin, Arda Senocak, Hyunwoo Ha, Andrew Owens, and Tae-Hyun Oh. Sound to visual scene generation by audio-to-visual latent alignment. In *CVPR*, 2023. 3

[69] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *ICLR*, 2014. 1

[70] Changyao Tian, Wenhai Wang, Xizhou Zhu, Jifeng Dai, and Yu Qiao. Vl-ltr: Learning class-wise visual-linguistic representation for long-tailed visual recognition. In *ECCV*, 2022. 8

[71] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Deep image prior. In *CVPR*, 2018. 5

[72] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *JMLR*, 9(11), 2008. 2

[73] 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, 4, 6

[74] Pascal Vincent, Hugo Larochelle, Isabelle Lajoie, Yoshua Bengio, Pierre-Antoine Manzagol, and Léon Bottou. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of machine learning research*, 11(12), 2010. 1, 2

[75] Xin Wang, Thomas E Huang, Trevor Darrell, Joseph E Gonzalez, and Fisher Yu. Frustratingly simple few-shot object detection. In *ICML*, 2020. 3, 7

[76] Zhecan Wang, Noel Codella, Yen-Chun Chen, Luowei Zhou, Xiyang Dai, Bin Xiao, Jianwei Yang, Haoxuan You, Kai-Wei Chang, Shih-fu Chang, et al. Multimodal adaptive distillation for leveraging unimodal encoders for vision-language tasks. *arXiv preprint arXiv:2204.10496*, 2022. 3, 4

[77] Mitchell Wortsman, Gabriel Ilharco, Jong Wook Kim, Mike Li, Simon Kornblith, Rebecca Roelofs, Raphael Gontijo Lopes, Hannaneh Hajishirzi, Ali Farhadi, Hongseok Namkoong, et al. Robust fine-tuning of zero-shot models. In *CVPR*, 2022. 3

[78] Chien-Sheng Wu, Steven Hoi, Richard Socher, and Caiming Xiong. Tod-bert: Pre-trained natural language understanding for task-oriented dialogue. *EMNLP*, 2020. 3

[79] Yang Xiao and Renaud Marlet. Few-shot object detection and viewpoint estimation for objects in the wild. In *ECCV*, 2020. 3

[80] Xiaopeng Yan, Ziliang Chen, Anni Xu, Xiaoxi Wang, Xiaodan Liang, and Liang Lin. Meta r-cnn: Towards general solver for instance-level low-shot learning. In *ICCV*, 2019. 7

[81] Lu Yang, He Jiang, Qing Song, and Jun Guo. A survey on long-tailed visual recognition. *IJCV*, 2022. 3

[82] Moon Ye-Bin, Dongmin Choi, Yongjin Kwon, Junsik Kim, and Tae-Hyun Oh. Eninst: Enhancing weakly-supervised low-shot instance segmentation. *arXiv preprint arXiv:2302.09765*, 2023. 3

[83] Kim Youwang, Kim Ji-Yeon, and Tae-Hyun Oh. Clip-actor: Text-driven recommendation and stylization for animating human meshes. *ECCV*, 2022. 3

[84] Lu Yuan, Dongdong Chen, Yi-Ling Chen, Noel Codella, Xiyang Dai, Jianfeng Gao, Houdong Hu, Xuedong Huang, Boxin Li, Chunyuan Li, Ce Liu, Mengchen Liu, Zicheng Liu, Yumao Lu, Yu Shi, Lijuan Wang, Jianfeng Wang, Bin Xiao, Zhen Xiao, Jianwei Yang, Michael Zeng, Luowei Zhou, and Pengchuan Zhang. Florence: A new foundation model for computer vision. *arXiv preprint arXiv:2111.11432*, 2021. 3

[85] 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

[86] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *ICLR*, 2018. 1, 2, 4, 6

[87] Yuhui Zhang, Jeff Z HaoChen, Shih-Cheng Huang, Kuan-Chieh Wang, James Zou, and Serena Yeung. Diagnosing and rectifying vision models using language. In *ICLR*, 2023. 4

[88] Jiapeng Zhu, Yujun Shen, Deli Zhao, and Bolei Zhou. In-domain gan inversion for real image editing. In *ECCV*, 2020. 5