

VITAL: A Visual Interpretation on Text with Adversarial Learning for Image Labeling

Tao Hu^{1,3}, Chengjiang Long^{*2}, Leheng Zhang¹, and Chunxia Xiao¹

¹School of Computer Science, Wuhan University, Wuhan, Hubei, China

²Kitware Inc. Clifton Park, NY, USA

³School of Information Engineering, Hubei University for Nationalities, Enshi, Hubei, China

hutao.es@foxmail.com, chengjiang.long@kitware.com, 375602133@qq.com, cxxiao@whu.edu.cn

Abstract

In this paper, we propose a novel way to interpret text information by extracting visual feature presentation from multiple high-resolution and photo-realistic synthetic images generated by Text-to-image Generative Adversarial Network (GAN) to improve the performance of image labeling. Firstly, we design a stacked Generative Multi-Adversarial Network (GMAN), StackGMAN++, a modified version of the current state-of-the-art Text-to-image GAN, StackGAN++, to generate multiple synthetic images with various prior noises conditioned on a text. And then we extract deep visual features from the generated synthetic images to explore the underlying visual concepts for text. Finally, we combine image-level visual feature, text-level feature and visual features based on synthetic images together to predict labels for images. We conduct experiments on two benchmark datasets, i.e., the Oxford 102 Category Flower Dataset and the Caltech-UCSD Birds-200-2011 Dataset and the experimental results clearly demonstrate the efficacy of our proposed approach.

1. Introduction

Nowadays, images are being taken and shared to be commented at an unprecedented rate among social networks like Facebook, Twitter, and Flickr. To help users efficiently organize and manage such media data from a very huge collection, it is necessary and practical to collect labeled visual datasets at large scale to develop automatic tools with robust machine learning approaches [18, 7, 16, 19, 17, 8]. However, most of the current annotation platforms like Amazon Mechanical Turk [2] and LabelMe [28] do not make full

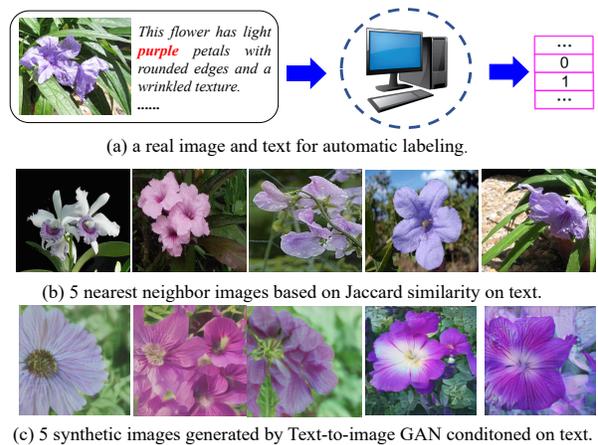


Figure 1: Illustration of two ways to visually interpret text for automatic image labeling (a). One is Johnson *et al.*'s work [9] in which a group of nearest neighbor images (b) defined based on Jaccard similarity between image meta-data especially tags extracted from a text. And the other one is our proposed VITAL method, which makes full use of K high-resolution and photo-realistic synthetic images (c) generated from StackGMAN++ (our modified version on StackGAN++ [37]) conditioned on text. The goal of this paper is to visually interpret text information and extract visual feature representations from synthetic images to boost the accuracy of image labeling on real images.

use of the text information to aid the labeling problems, although most images on the web carry rich text information which includes informative and semantic signals like who took the photo and, where and with whom.

Prior work takes advantage of text context information to improve image classification by various treatments like selecting top frequently used words and user-generated

*This work was co-supervised by Chengjiang Long and Chunxia Xiao.

tags [20, 6, 14], extracting text-level feature representation with Text Convolutional Neural Networks (CNNs) [1], and exploring visual feature representation from a set of neighbor images [9] defined based on Jaccard similarity between image metadata especially tags extracted from a text. The intuition behind is that images with similar text context information tend to depict similar scenes.

Inspired by the development of Text-to-image Generative Adversarial Networks (GANs) [21, 36, 37, 33], which is able to generate high-resolution and photo-realistic synthetic images conditioned on a text, we propose a novel visual interpretation on text with adversarial learning, named as “VITAL”, by interpreting text information with visual concepts extracted from a series of visually plausible synthetic images generated by Text-to-image GANs.

Unlike Johnson *et al.*’s work [9] which uses a set of nearest neighbor images based on Jaccard similarity between the text of query image and texts of training images, our proposed VITAL approach generates synthetic images conditioned on only the text of query image without using text from any other images. As illustrated in Figure 1, the color information of nearest neighbor images to represent text are not well consistent with the real image, while synthetic images generated in our VITAL are able to not only cover most of the content in text information, but also provide much information about the background which is underlying in text information. This is also consistent with our human understanding to text information.

“As there are 1000 Hamlets, there are 1000 readers.” Usually, given a text that describes a specific scene, different readers can image different relevant visual scenes in their brains. Obviously, one synthetic image is not sufficient to simulate what multiple readers can visually interpret from a single text information. In principle, any text-to-image GANs can be extended and incorporated into our VITAL framework and in this paper we just start to extend one current state-of-the-art Text-to-image GAN, StackGAN++ [37], dubbed StackGMAN++ in brief with “M” indicating multiple generators, to extend the number of generators from the original 1 to K with different noise priors at each stage. The intuition behind is that each reader interprets Hamlets based on his/her own prior knowledge, and different prior knowledge leads to a different image of Hamlet in his/her mind.

With K generators, we are able to generate K synthetic images. Since ResNet [5] has demonstrated successes in a vast of vision applications, we apply it to extract visual feature for each synthetic image. It worth emphasizing that we care much more about the common visual representation among these K synthetic images rather than each individual. Therefore, to obtain a compact visual feature representation, we first apply an affine transformation with a ReLU layer to adjust feature maps and reduce the channels before

we apply an element-wise pooling to extract the common feature representation.

Considering that feature fusion has been proved to be able to effectively improve the performance of image labeling, we combine the real image feature extracted from an image-level CNN, text feature extracted from a text-level CNN, and the common feature representation from K synthetic images by concatenation and feed them into a fully connected layer as a classification for image labeling [23].

To sum up, our paper has three contributions:

- We propose a novel way to interpret text information with K visually plausible synthetic images generated via our StackGMAN++ derived by modifying the number of generators at each stage from 1 to K on StackGAN++ associated with different noise priors.
- We extract a common and compact visual feature representation from synthetic images conditioned on a text, and combine it with an image-level feature from real image, and a text-level feature together by concatenation to boost the performance of image labeling.
- The experiments on two benchmark datasets have demonstrated that our proposed approach outperforms the state-of-art method [9].

2. Related work

The related work can be divided into two categories: *text information for image labeling* and *Text-to-image GANs*.

Text information for image labeling. As an important source for multimedia, text information has been studied and combined with image content to improve the accuracy of image labeling, because it provides informative and semantic signals [34] like who took the photo and, where and with whom. McAuley *et al.* [20] selected top-1000 most frequently occurring words from the texts in the training set as tags [4, 13, 30, 29, 31, 6], explored pairwise social relations and applied a CRF-based structural learning approach for multi-label image annotation, which demonstrates impressive results although only metadata is used without any visual features. Following [20], Johnson *et al.* [9] proposed a deep convolutional neural network to combine both the visual information of images and their neighbor images defined based on the shared similar metadata especially tags [14] from a text. With multiple text CNNs [12, 11, 10] emerged, Long *et al.* [1] proposed to extract deep text-level features rather than user-generated tags for text representation in image labeling. It worths mentioning that Johnson *et al.*’s work [9] is a visual interpretation of text, although neighbor images depend too much on the density distribution of training data. Different from [9], we propose to use a

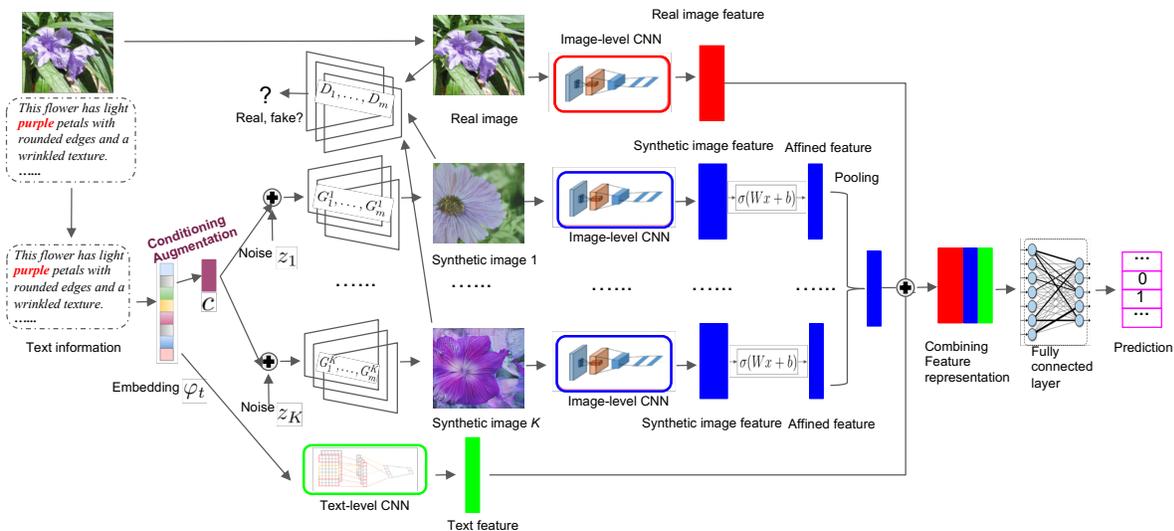


Figure 2: Overview of our proposed framework VITAL for image labeling with text information. At the beginning, we train a Text-to-image GAN, StackGMAN++, a modified version on StackGAN++ [37], to generate K high-resolution and photo-realistic synthetic images. Given an input image with its text information, we apply the trained StackGMAN++ to generate K synthetic images. We apply an image-level CNN to extract visual feature on each synthetic image. To avoid the possible affine transforms between the synthetic images, we apply a linear transformation with ReLU to reduce the dimensions of the visual features. After that, we apply an element-wise pooling to extract visual feature representation in blue from all K synthetic images. Finally, we combine the image-level feature (in red), text feature (in green) and the text corresponding visual feature (in blue) together by concatenation and feed them into a fully connected layer as a classifier to achieve the final prediction of labeling on the given image.

Text-to-image GAN to generate high-resolution and photo-realistic synthetic images to extract visual feature representation for the text of query image only, which we expect to be complementary to both visual feature on real images and textual feature on a text.

Text-to-image GANs are proposed to effectively translate visual concepts from characters to pixels, with the aim to bridge these advances in text and image modeling [24, 15]. Based on text descriptions, Reed *et al.* [27] was able to generate plausible 64×64 images for birds and flowers, and 128×128 images were successfully generated by utilizing additional annotations on object part locations [26]. To generate high-resolution (*e.g.*, 227×227) and photo-realistic images, Nguyen *et al.* [21] proposed to generate images conditioned on a text using an approximate Langevin sampling approach with an iterative optimization process. Zhang *et al.* presented a Stacked GANs (StackGAN) [36] to generate 256×256 photo-realistic synthetic images with two stages, in which low-resolution images are generated at first stage, and then more details are added in the second stage to form high-resolution images with better quality by utilizing an encoder-decoder network before the upsampling layers. And its improved version StackGAN++ [37] extends to multi-stage GANs with multiple generators and multiple discriminators arranged in a tree-

like structure. Based on StackGAN and StackGAN++, Xu *et al.* [33] adopted attention mechanism to generate synthetic images from fine-grained text with Attention Generative Adversarial Networks. We argue that high-resolution and photo-realistic synthetic images should be beneficial for us to extract visual representation for a text. Hence we modify the StackGAN++ to be a Generative Multi-Adversarial Network (GMAN), StackGMAN++, to generate K synthetic images with different generators. Different from Durgkar *et al.*'s [3] GMAN with one generator and multiple discriminators, our StackGMAN++ uses multiple generators with various prior noise vectors and one discriminator at each stage to generate multiple synthetic images, to well represent visual concepts embedded in text information.

3. Proposed method

As illustrated in Figure 2, our proposed framework consists of three key components, *i.e.*, StackGMAN++ to generate K synthetic images, extract visual feature representation from the synthetic images, and combine the synthetic image feature with real image feature and text feature to conduct a classification task for image labeling. We are going to discuss with details in the following subsections.

3.1. StackGMAN++ to generate K synthetic images

Text-to-image Generative Adversarial Networks (GANs) [27, 36, 37] are proved to be able to generate visually-plausible images conditioned on a text. We start on StackGAN++ [37] because it is able to generate high-resolution and photo-realistic images. Note there are two implementation versions for StackGAN++, *i.e.*, StackGAN-v1 and StackGAN-v2. Our StackGMAN++ is derived from StackGAN-v2 which extends StackGAN-v1 from two stages to multiple stages and organizes generators and discriminators in a tree-like structure.

Intuitively, given a text information, different people may imagine a different visual scene, and one text-based synthetic image is not sufficient to cover the underlying information behind the text itself. Therefore, to better explore the visual feature representation of any text information, we proposed to generate K synthetic images for each text information.

Different from StackGAN-v2 which takes only one noise vector z as the input and has a generator G_i and a corresponding discriminator D_i for i -th branch with different scales, our StackGMAN++ takes a conditional vector c which is defined based on the text embedding φ_t in [37] and K different prior noise vectors z_1, \dots, z_K (we sample values for each z_k from a normal distribution) as input, and at each stage i , we design one discriminator D_i and K generators G_i^1, \dots, G_i^K to generate synthetic images at a certain scale. We pass the hidden feature h_i^k for each generator G_i^k by a non-linear transformation,

$$h_i^k = \begin{cases} F_i^k(c, z_k) & i = 0 \\ F_i^k(h_{i-1}^k, c) & i = 1, \dots, m \end{cases} \quad (1)$$

where h_i^k represents hidden features for the k -th generator G_i^k at the i -th branch, m is the total number of branches, and F_i^k is modeled as the corresponding neural network. In order to encourage the generators to draw images with more details according to the conditioning variables, c is concatenated to the hidden features h_{i-1}^k as the inputs of F_i^k for calculating h_i^k . Based on hidden features at different layers (h_1^k, \dots, h_m^k), generators G_1^k, \dots, G_m^k can generate synthetic images of small-to-large scales

$$s_i^k = G_i^k(h_i^k), i = 1, \dots, m. \quad (2)$$

For the training purpose, we define the loss functions with joint conditional and unconditional distribution ap-

proximation at each stage i as following:

$$\begin{aligned} \mathcal{L}_{D_i} = & K \mathbb{E}_{\mathbf{x}_i \sim p_{data_i}} [\log D_i(\mathbf{x}_i)] + \\ & \sum_{k=1}^K \mathbb{E}_{s_i^k \sim p_{G_i^k}} [\log(1 - D_i(s_i^k))] + \\ & + K \mathbb{E}_{\mathbf{x}_i \sim p_{data_i}} [\log D_i(\mathbf{x}_i, c)] + \\ & \sum_{k=1}^K \mathbb{E}_{s_i^k \sim p_{G_i^k}} [\log(1 - D_i(s_i^k, c))], \end{aligned} \quad (3)$$

$$\begin{aligned} \mathcal{L}_{G_i^1, \dots, G_i^K} = & \sum_{k=1}^K \mathbb{E}_{s_i^k \sim p_{G_i^k}} [-\log D_i(s_i^k)] \\ & + \sum_{k=1}^K \mathbb{E}_{s_i^k \sim p_{G_i^k}} [-\log D_i(s_i^k, c)] \end{aligned} \quad (4)$$

where \mathbf{x}_i is from the true image distribution p_{data_i} at the i -th scale, and s_i^k is from the model distribution $p_{G_i^k}$ at the same scale. Then the discriminator D_i and generators G_i^1, \dots, G_i^K at i -th stage can be optimized in a joint form by alternatively maximizing \mathcal{L}_{D_i} and minimizing $\mathcal{L}_{G_i^1, \dots, G_i^K}$ until convergence.

Based on StackGMAN++, we are able to generate K high-resolution and photo-realistic synthetic images $\mathbf{s} = \{s^1, \dots, s^K\}$ with $s^k = G^k(s_{m-1}^k)$, and the size of each image is 256×256 .

Discussion: we extend StackGAN++ to StackGMAN++ and joint learn the model with a united loss function to generate the diverse synthetic images. Note that the K generators in StackGMAN++ share weights. In this way, we can control the training cost when compared to train K StackGAN++ separately. We observe that training a StackGMAN++ with shared weights is less expensive than training K StackGAN++, and the K synthetic images generated by StackGMAN++ are more diverse.

3.2. Visual representation for text Information

Given a text t , we are able to generate K synthetic images $\mathbf{s} = \{s^1, \dots, s^K\}$ with our StackGMAN++. Then we feed K generated synthetic images into ResNet [5] to extract visual feature. Note that we use ResNet as feature extractor ϕ in the pretrained ResNet model [5], and get a feature map of the second last layer of $\phi(s^k)$ of $7 \times 7 \times 2048$ size for each synthetic image s^k , respectively.

In order to fuse visual features for these K synthetic image, we compute an h -dimensional hidden state for each image by applying an affine transformation and an element-wise ReLU nonlinearity $\sigma(\varepsilon) = \max(0, \varepsilon)$ to its feature. To let the model treat hidden states for each synthetic image differently, we apply distinct transformations to $\phi(s^k)$ with parameters $W_k \in \mathbb{R}^{d \times h}$ and $b_k \in \mathbb{R}^h$, and then we arrives at hidden states $\mathbf{v}_{s^k} \in \mathbb{R}^h$ for $s^k \in \mathbf{s}$. To generate as



Figure 3: Plot of the multichannel convolutional neural network, 3Text-CNNs, for text-level feature extraction.

single hidden state $\mathbf{v}_s \in \mathbb{R}^h$ for all the synthetic images s , we apply an element-wise max-pooling on each $\mathbf{v}_{s,k}$ so that $\mathbf{v}_s = \max_k \mathbf{v}_{s,k}$, *i.e.*,

$$\mathbf{v}_s = \max_k (\sigma(W_k \phi(s^k) + b_k)) \quad (5)$$

and pass it to be included into the final combined feature representation with image feature and text feature.

Discussion: we choose synthetic images rather than visual features because we want to visually interpret the text with high quality synthetic images so that the human can view directly, while visual features may miss some details.

3.3. Feature fusion for image labeling

As shown in Figure 2, besides the visual feature representation \mathbf{v}_s on K synthetic images, we also include the text-level feature \mathbf{v}_t and image-level feature \mathbf{v}_x together to better explore both image and text information to improve the quality for image labeling. In this paper, we choose to use the second last layer of ResNet [5] to extract the visual feature \mathbf{v}_x for real image x because ResNet has been proved successful in most visual application tasks.

Regarding text information, a standard deep learning model for text classification and sentiment analysis uses a word embedding layer and one-dimensional convolutional neural network [12]. The model can be expanded by using multiple parallel convolutional neural networks that read the source document using different kernel sizes. This, in effect, creates a multichannel convolutional neural network for text that reads text using different n -gram sizes (groups of words). We follow Kim’s multi-channel model to implement a merged model with 3 text CNNs with kernels of different sizes, denoted as 3Text-CNNs, as illustrated in Fig-

ure 3, to extract 512-dimensional text-level feature \mathbf{v}_t from the second last layer.

All these three kinds of features are combined to form the final feature representation by concatenation as $(\mathbf{v}_x, \mathbf{v}_{\hat{x}}, \mathbf{v}_t)$ and feed them into a fully connected layer to conduct a classification task as image labeling.

3.4. Implementation details

The parameters need to be learned include the parameters in mK generators and m discriminators in StackGMAN++, parameters in ResNet-50s, 3Text-CNNs, the affine transformation parameter $W_{\hat{x}}$ and $b_{\hat{x}}$, and the parameters in the last fully connected layer. Note our training procedure is divided into two phases. At Phase I, we use pairs of text and its corresponding image to train our StackGMAN++ in an alternative optimization procedure until it convergence. Then at Phase II, we apply the trained StackGMAN++ to generate K high-resolution and photo-realistic synthetic images. Then we feed n real images, nK synthetic images and n text to learn the rest of parameters in the entire framework with cross-entropy loss function.

Note that we follow the tricks in StackGAN-v2 to train StackGMAN++ at each stage at Phase I with a batch size of 12 for 350000 iterations. At Phase II, with a minibatch size of 50, we initialize all parameters with pre-trained models (ResNet and 3Text-CNNs) and use stochastic gradient descent with a fixed learning rate, RMSProp, as the optimization.

4. Experiments

Our experiments are conducted on two datasets, *i.e.*, the Oxford 102 Category Flower Dataset [22], and the Caltech-UCSD Birds-200-2011 Dataset [32]. We use accuracy as the metric to measure performance of image labeling.

4.1. Experiments on the Oxford 102 Category Flower Dataset

The Oxford 102 Category Flower Dataset [22] consists of 8,189 images from 102 categories of flowers which commonly occurs in the United Kingdom, and each category has 40 to 258 images. The images have large scales, pose and light variations. The text context information is provided by [25] with 10 descriptions for each image. Due to the limit space, we plot text using only 1 sentence and use the symbol “.....” to represent the rest 9 sentences in this paper. We train our StackGMAN++ with text and the corresponding real images. With the learned StackGMAN++, we are able to generate K visual plausible synthetic images conditioned on a text for experiments.

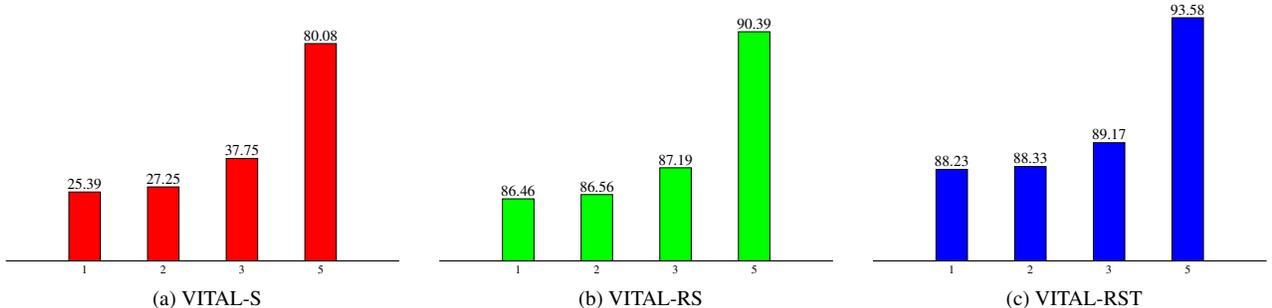


Figure 4: Performance with accuracy (unit: %) for our VITAL-S, VITAL-RS and VITAL-RST using visual feature representation on K synthetic images with different values of K , *i.e.*, $K = 1, 2, 3$, and 5 on the 102 Category Flower Dataset.



Figure 5: Visualization of $K = 5$ high-resolution and photo-realistic synthetic images (blue) conditioned on a text, and compared with the corresponding real images (red) on the Oxford 102 Category Flower Dataset.

4.1.1 Effectiveness of StackGMAN++

To verify the effectiveness of our StackGMAN++, we conduct experiments to check visual concept consistency between the generated synthetic images by StackGMAN++ and the corresponding real images, and the effectiveness of visual feature representation on K synthetic images.

Visual concept consistency between synthetic and real images. We firstly visualize the generated synthetic images conditioned on a text and measure the correlation between our generated synthetic images with the corresponding images in a visual feature space.

As shown in Figure 5, our generated K synthetic images are able to not only cover main content elements in the text, but also provide underlying background and other rich visual information like size, shape and pose variations which are not mentioned in the text, due to various prior noise vector z_k . These observations are consistent with our human’s behavior to interpret a text based on his/her prior knowledge. In other words, even given the same text, different people with different growing backgrounds will imagine different visual pictures in their brains. Such diverse interpretations are complementary to each other and can be

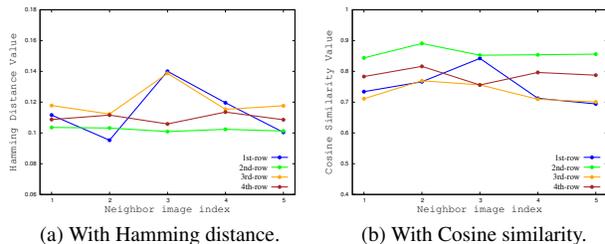


Figure 6: The correlation measured with Hamming distance and Cosine similarity between the synthetic images generated by our StackGMAN++ conditioned on a text and the corresponding real images on the Oxford 102 Category Flower Dataset.

merged to formulate a more representative format.

We also measure the correlation between our generated synthetic images conditioned on a text with the corresponding real images with two distance metrics, *i.e.*, Hamming distance and Cosine similarity, between their visual feature vectors extracted by ResNet [5]. For Hamming distance in the range $[0, 1]$, smaller value means the higher similarity between images. For Cosine similarity in the range $[-1, 1]$, the value closer to 1.0 means two compared images are more correlative to each other in the given feature space. We plot both Hamming distance values and Cosine similarity value between our generated synthetic images and the corresponding real images in Figure 6. As we observe, none of Hamming distances is larger than 0.15 and all Cosine distance values are over 0.70, which indicates high correlation between each synthetic image and the corresponding real image in the visual feature space.

Effectiveness of visual feature representation on K synthetic images. As stated in Section 3.3, our VITAL uses the feature combinations $(\mathbf{v}_x, \mathbf{v}_s, \mathbf{v}_t)$ where \mathbf{v}_x and \mathbf{v}_s indicate the visual feature representation extracted by ResNet [5] on real image, and synthetic images, respectively, and \mathbf{v}_t represents the text-level feature extracted from 3Text-CNNs [35]. We then develop two different baseline algorithms with two different feature combina-

Table 1: Performance comparison on the Oxford 102 Category Flower Dataset. (unit: %)

Methods	Accuracy
JCNN-NN [9]	89.16±0.57
3Text-CNNs [12]	37.63±0.76
ResNet [5]	85.86±1.85
ResNet [5]+3Text-CNNs [12]	88.39±0.44
VITAL-S	79.87±0.31
VITAL-RS	89.68±0.61
VITAL-RST	93.38±0.15

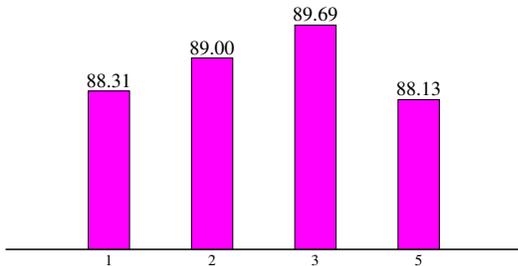


Figure 7: Performance with accuracy (unit: %) for JCNN-NN with different K nearest neighbor images.

tions, *i.e.*, \mathbf{v}_s only and $(\mathbf{v}_x, \mathbf{v}_s)$. For notation simplification, we denote our proposed method to be VITAL-RST where “-RST” represents the feature combinations $(\mathbf{v}_x, \mathbf{v}_s, \mathbf{v}_t)$. We further denote the first to second baseline algorithm to be VITAL-S and VITAL-RS, respectively.

We conduct a group of experiments by setting K to be various values, *i.e.*, 1, 2, 3 and 5. The results are summarized in the Figure 4. Apparently, for any algorithm among VITAL-S, VITAL-RS, and VITAL-RST, the performance accuracy goes up when the value of K increases. This indicates that multiple synthetic images are complementary to be used for extracting visual concepts embedded in text and boosting the accuracy for image labeling.

Note that K is also the number of generators in each stage in our StackGMAN++. Therefore, when $K = 1$, our StackGMAN++ is equivalent to StackGAN++ with its implementation version StackGAN-v2. Obviously, all these three algorithms with K (especially when $K > 1$) synthetic images generated by our StackGMAN++ always work better than using only synthetic images generated by StackGAN++. The observation strongly demonstrates the effectiveness and robustness of our StackGMAN++.

4.1.2 Performance comparison

We compare our proposed VITAL-RST with Johnson *et al.*’s Convolutional Neural Network with Nearest Neighborhood [9], denoted as “JCNN-NN”, which explores the related neighbor images to interpret image metadata especially including tags from text information. To our best ac-

knowledge, JCNN-NN is the most closely related work to our VITAL because it can be interpreted as a visual interpretation of image metadata with the related neighbor images. To make sure the comparison is fair, we utilize the same ResNet [5] as the visual feature extractor in JCNN-NN. In addition, we add two simple baseline algorithms, *i.e.*, ResNet [5] and 3Text-CNNs [35], which indicate using \mathbf{v}_x only and using \mathbf{v}_t only, respectively.

To clarify, we do not compare our VITAL with StackGMAN++ to other adversarial learning methods because this is not our focus and our focus is how to interpret text for image labeling by extending and applying the existing text-to-image GANs. We run experiments on the Oxford 102 Category Flower Dataset. Note that we also run JCNN-NN with five different number of neighbor images, as shown in Figure 7, from which we can find $K = 3$ is the best choice for JCNN-NN.

We conduct the experiments repeatedly for 10 times with 10 different random data split to form the training and testing sets and the results with all six algorithms are summarized in Table 1, from which we can observe: (1) 3Text-CNNs performs much worse than ResNet by a half and this conveys a clue that text on the dataset is a little weaker when compared with image content; (2) with single feature representation, our VITAL-S works much better than 3Text-CNNs and close to ResNet, which indicates visual feature extracted on synthetic images is more representative than text-feature; (3) combining with real image feature, VITAL-RS is able to improve the performance by 4.01% from ResNet, and works better than JCNN-NN [9] and ResNet [5]+3Text-CNNs [12] (*i.e.*, a combination of RT without involving VITAL), which shows that our visual interpretation on text is more effective and robust than using a set of neighbor images defined based on the Jaccard similarity between image metadata especially tags extracted from text; and (4) combining with both real image feature and text feature, VITAL-RST performs the best. Apparently, the visual interpretation in our VITAL is robust and the visual synthetic image feature is complementary to both visual real image feature and text feature.

4.2. Experiments on the Caltech-UCSD Birds-200-2011 Dataset

The Caltech-UCSD Birds-200-2011 Dataset consists of 11,169 bird images from 200 categories and each category has 60 images averagely. We randomly select 9,935 images for training, and use the resting 1,234 images for testing. The dataset is very challenging because it contains images with multiple objects and various backgrounds. We train our StackGMAN++ with $K = 5$ and use it to generate synthetic images on the text with 10 descriptions for each real image to conduct the experimental evaluation.

We repeat the experiments with 10 different random

Table 2: Performance comparison on the resized Caltech-UCSD Birds-200-2011 dataset. (unit: %)

Methods	Accuracy
JCNN-NN [9]	89.91±0.48
3Text-CNNs [12]	5.86±0.69
ResNet [5]	86.81±1.14
ResNet [5]+3Text-CNNs [12]	89.38±1.17
VITAL-S	55.05±0.20
VITAL-RS	93.28±1.25
VITAL-RST	94.49±0.90

training/testing data splitting and summarize the results in Table 2, which can be observed from four aspects. (1) ResNet performs much better than 3Text-CNNs, which indicates image is more representative than text content. (2) Using text information only, our VITAL-S still outperforms 3Text-CNNs. (3) VITAL-RST performs a little better than VITAL-RS which works much better than VITAL-S. Again, this observation confirms the complementary relationship between three kinds of feature representations. (4) Our VITAL-RS and VITAL-RST, performs better than JCNN-NN and ResNet+3Text-CNNs, which suggests that our proposed VITAL is good at visual interpretation on a text by extracting visual concepts from the text for boosting the labeling accuracy.

4.3. VITAL performance with GMAN vs. K GANs

Since we can train K GANs with K different noise levels separately and then use the trained GANs to generate K synthetic images, we also can assess the performance of our VITAL with K GANs. For the purpose of fair comparison, we train $K = 5$ StackGAN++ with different noise levels individually and use the K generated synthetic images to extract visual features for image labeling on the Oxford 102 Category Flower Dataset. The accuracy performance of VITAL-RST with K StackGAN++ is 90.59%, which is 3% lower than VITAL-RST with our StackGMAN++.

We visualize the synthetic images generated by both StackGMAN++ and K StackGAN++ in Figure 8. Obviously, our proposed StackGMAN++ is able to generate more diverse images, and the diversity among the synthetic images benefits the performance of image labeling.

4.4. Analysis on successful and failure cases

To further explain why our proposed VITAL works better than JCNN-NN, we conduct an analysis on the quality of the nearest neighbor (NN) images used in JCNN-NN. As shown in Figure 9, the color of neighbor images are not always consistent with real images, and the background of all the neighbor images are more complicated when compared with the generated synthetic images in Figure 5. Moreover, the Jaccard similarity between the query image and the k -



Figure 8: Visualization of $K = 5$ synthetic images (blue) generated by StackMGAN++ (right bottom) and K StackGAN++ (right top). The input text and the corresponding real image are on the left.



Figure 9: Visualization of JCNN-NN [9]’s $K = 3$ nearest neighbor (NN) images based on the Jaccard similarity of tags extracted from text. From left to right are: text, real image, and 3 nearest-neighbour images, respectively.

th NNs always decreases when the value of k increases.

As a non-parameter method, the quality of nearest neighbor images in JCNN-NN will be perfect if an infinite number of samples is available, but in practice the number of samples is limited. Therefore the quality of neighbor images depends on the density distribution in training data. If the distribution is dense enough, then the quality of neighbor images can be a guarantee. Obviously, in both the 102 Category Flower Dataset and Caltech-UCSD Birds-200-2011 Dataset, the observations indicate that the density is not sufficient to ensure the good quality of neighbor images for JCNN-NN.

On the contrary, our proposed VITAL generates K visual plausible synthetic images by StackGMAN++ conditioned on a text of query image only, without requiring text for any other images, which makes the generated synthetic images are closely correlated to the corresponding real images, displaying key visual elements embedded in text and even providing more details about the underlying background and variations. Therefore, our proposed VITAL is more robust to visually interpret text for improving the performance of image labeling.

We also visualize a successful case and a failure case in Figure 10. As we can observe, if text describes a flower with detailed information about colors hand shapes, then our StackGMAN++ is able to generate a bunch of informative synthetic images for leveraging the performance of image labeling. Otherwise, if a text is hard to understand, ambiguous and even misleading, then the generated synthetic images will not be consistent with each other to represent the visual concepts well.



Figure 10: Visualization of a successful case (top) and a failure case (bottom) for our proposed VITAL. From left to right are: text, real image, and 5 synthetic images, respectively.

5. Conclusion

In this paper, we propose a novel way to visually interpret text with adversarial learning for image labeling. With our StackGMAN++, K high-resolution and photo-realistic synthetic images are generated conditioned on a text to represent the visual concepts. The visual synthetic image feature has been proved to be able to improve accuracy for image labeling, and it is complementary to real image feature and text feature. The experimental results conducted on two datasets well support our claims in the paper.

Our future work includes further exploring our GMAN to incorporate the state-of-the-art GANs like AttnGAN [33] to produce better quality of synthetic images, improving VITAL's performance on image labeling, and extending our VITAL to solve more general multimedia problems.

References

- [1] E. S. Chengjiang Long, Roddy Collins and A. Hoogs. Deep neural networks in fully connected crf for image labeling with social network metadata. In *IEEE Winter Conf. on Applications of Computer Vision (WACV)*. IEEE, 2018. 2
- [2] K. Crowston. Amazon mechanical turk: A research tool for organizations and information systems scholars. In *Shaping the Future of ICT Research. Methods and Approaches*. 2012. 1
- [3] I. Durugkar, I. Gemp, and S. Mahadevan. Generative multi-adversarial networks. *arXiv*, 2016. 3
- [4] M. Guillaumin, J. Verbeek, and C. Schmid. Multimodal semi-supervised learning for image classification. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010. 2
- [5] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 2, 4, 5, 6, 7, 8
- [6] H. Hu, G.-T. Zhou, Z. Deng, Z. Liao, and G. Mori. Learning structured inference neural networks with label relations. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 1, 2
- [7] G. Hua, C. Long, M. Yang, and Y. Gao. Collaborative active learning of a kernel machine ensemble for recognition. In *IEEE International Conference on Computer Vision (ICCV)*, 2013. 1
- [8] G. Hua, C. Long, M. Yang, and Y. Gao. Collaborative active visual recognition from crowds: A distributed ensemble approach. *IEEE transactions on pattern analysis and machine intelligence*, 40(3):582–594, 2018. 1
- [9] J. Johnson, L. Ballan, and L. Fei-Fei. Love thy neighbors: Image annotation by exploiting image metadata. In *IEEE International Conference on Computer Vision (ICCV)*, 2015. 1, 2, 7, 8
- [10] R. Johnson and T. Zhang. Effective use of word order for text categorization with convolutional neural networks. In *The Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2015. 2
- [11] N. Kalchbrenner, E. Grefenstette, and P. Blunsom. A convolutional neural network for modelling sentences. In *The Association for Computational Linguistics (ACL)*, 2014. 2
- [12] Y. Kim. Convolutional neural networks for sentence classification. In *The Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014. 2, 5, 7, 8
- [13] S. Lindstaedt, V. P. adn Roland Morzinger, R. Kern, H. Mullner, and C. Wagne. Recommending tags for pictures based on text, visual content and user context. In *International Conference on Internet and Web Applicants and Services (ICIW)*, 2008. 2
- [14] F. Liu, T. Xiang, T. M. Hospedales, W. Yang, and C. Sun. Semantic regularisation for recurrent image annotation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 1, 2
- [15] N. Liu, J. Han, and M.-H. Yang. Picanet: Learning pixel-wise contextual attention for saliency detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 3
- [16] C. Long and G. Hua. Multi-class multi-annotator active learning with robust gaussian process for visual recognition. In *IEEE International Conference on Computer Vision (ICCV)*, 2015. 1
- [17] C. Long and G. Hua. Correlational gaussian processes for cross-domain visual recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 1
- [18] C. Long, G. Hua, and A. Kapoor. Active visual recognition with expertise estimation in crowdsourcing. In *IEEE International Conference on Computer Vision (ICCV)*, pages 3000–3007, 2013. 1
- [19] C. Long, G. Hua, and A. Kapoor. A joint gaussian process model for active visual recognition with expertise estimation in crowdsourcing. *International journal of computer vision (IJCV)*, 116(2):136–160, 2016. 1
- [20] J. J. McAuley and J. Leskovec. Image labeling on a network: Using social-network metadata for image classification. In *The European Conference on Computer Vision (ECCV)*, 2012. 1, 2
- [21] A. Nguyen, J. Clune, Y. Bengio, A. Dosovitskiy, and J. Yosinski. Plug & play generative networks: Conditional iterative generation of images in latent space. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 2, 3
- [22] M.-E. Nilsback and A. Zisserman. Automated flower classification over a large number of classes. In *Computer Vision, Graphics & Image Processing (ICVGIP)*, 2008. 5

- [23] Y. Niu, Z. Lu, J.-R. Wen, T. Xiang, and S.-F. Chang. Multi-modal multi-scale deep learning for large-scale image annotation. *IEEE Transactions on Image Processing (TIP)*, 28(4):1720–1731, 2019. 2
- [24] B. A. Plummer, P. Kordas, M. Hadi Kiapour, S. Zheng, R. Piramuthu, and S. Lazebnik. Conditional image-text embedding networks. In *The European Conference on Computer Vision (ECCV)*, September 2018. 3
- [25] S. Reed, Z. Akata, H. Lee, and B. Schiele. Learning deep representations of fine-grained visual descriptions. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 5
- [26] S. E. Reed, Z. Akata, S. Mohan, S. Tenka, B. Schiele, and H. Lee. Learning what and where to draw. In *Advances in Neural Information Processing Systems (NIPS)*, 2016. 3
- [27] S. E. Reed, Z. Akata, X. Yan, L. Logeswaran, B. Schiele, and H. Lee. Generative adversarial text to image synthesis. In *The International Conference on Machine Learning (ICML)*, 2016. 3
- [28] B. C. Russell, A. Torralba, K. P. Murphy, and W. T. Freeman. Labelme: a database and web-based tool for image annotation. *International journal of computer vision (IJCV)*, 2008. 1
- [29] N. Sawant, R. Datta, J. Li, and J. Z. Wang. Quest for relevant tags using local interaction networks and visual content. In *The ACM SIGMM International Conference on Multimedia Information Retrieval (MIR)*, 2010. 2
- [30] B. Sigurbjörnsson and R. van Zwol. Flickr tag recommendation based on collective knowledge. In *The International Conference on World Wide Web (WWW)*, 2008. 2
- [31] Z. Stone, T. Zickler, and T. Darrell. Autotagging facebook: Social network context improves photo annotation. In *IEEE Conference on Computer Vision and Pattern Recognition Workshop (CVPRW)*, 2008. 2
- [32] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The Caltech-UCSD Birds-200-2011 Dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011. 5
- [33] T. Xu, P. Zhang, Q. Huang, H. Zhang, Z. Gan, X. Huang, and X. He. Attngan: Fine-grained text to image generation with attentional generative adversarial networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2, 3, 9
- [34] R. A. Yeh, M. N. Do, and A. G. Schwing. Unsupervised textual grounding: Linking words to image concepts. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2
- [35] W. Yin and H. Schütze. Multichannel variable-size convolution for sentence classification. *arXiv*, 2016. 6, 7
- [36] H. Zhang, T. Xu, H. Li, S. Zhang, X. Huang, X. Wang, and D. Metaxas. Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. *arXiv preprint*, 2017. 2, 3
- [37] H. Zhang, T. Xu, H. Li, S. Zhang, X. Wang, X. Huang, and D. Metaxas. Stackgan++: Realistic image synthesis with stacked generative adversarial networks. *arXiv*, 2017. 1, 2, 3, 4