Towards Unbiased Multi-Label Zero-Shot Learning With Pyramid and Semantic Attention

Ziming Liu, Song Guo, Fellow, IEEE, Jingcai Guo, Member, IEEE, Yuanyuan Xu, and Fushuo Huo

Abstract—Multi-label zero-shot learning extends conventional single-label zero-shot learning to a more realistic scenario that aims at recognizing multiple unseen labels of classes for each input sample. Existing works usually exploit attention mechanism to generate the correlation among different labels. However, most of them are usually biased on several major classes while neglect most of the minor classes with the same importance in input samples, and may thus result in overly diffused attention maps that cannot sufficiently cover minor classes. We argue that disregarding the connection between major and minor classes, i.e., correspond to the global and local information, respectively, is the cause of the problem. In this paper, we propose a novel framework of unbiased multi-label zero-shot learning, by considering various class-specific regions to calibrate the training process of the classifier. Specifically, Pyramid Feature Attention (PFA) is proposed to build the correlation between global and local information of samples to balance the presence of each class. Meanwhile, for the generated semantic representations of input samples, we propose Semantic Attention (SA) to strengthen the element-wise correlation among these vectors, which can encourage the coordinated representation of them. Extensive experiments on the large-scale multi-label benchmarks MS-COCO, NUS-WIDE and Open-Images demonstrate that the proposed method surpasses other representative methods by significant margins.

Index Terms—Multi-label zero-shot learning, attention mechanism, semantic feature space, classification, pattern recognition.

I. INTRODUCTION

To accurately recognize and describe the rich information contained within a real-world image has been a hot research topic in recent years. With the popularity of smart computing and machine intelligence, various devices such as mobile phones are usually equipped with photo capture and storage functions, that can produce tons of images and continue spawning the demand for recognition. Concretely, the multi-label classification, i.e., to recognize all labels/classes of objects in an image, is one of the fundamental and essential tasks in image recognition, which is of crucial importance to achieve several downstream tasks such as scene analysis, autonomous driving, intelligent monitoring, and so on.

In the past few years, with the increasing development of deep learning techniques and tons of well-annotated training images, recent multi-label classification models [1], [2], [3], [4], [5], [6], [7] can easily achieve superior performance in both recognition accuracy and generalization ability in a close-set scenario, wherein, both training and testing images fall into the same classes of labels. Notably, existing multi-label classification models usually improve their recognition performance by considering the dependencies among labels, and jointly ensemble separate classifiers of each individual label for better classification. However, in real-world applications such as Flickr and other online apps, billions of users are constantly releasing images containing various seen classes and many more novel unseen classes. For example, an image on Flickr can only contain incomplete tags from users, i.e., keywords that facilitate the description and searching process, which may result in less classes been recognized and degrades the recognition performance dramatically. In a more realistic setting, the model is necessarily to be endowed with certain predictive ability and thus to infer whether a new class exists in an image. However, conventional multi-label classification models fail to handle the task that requires not only recognizing objects that come from seen classes, but also generalizing to objects whose labels have never been seen or observed by the classifier. Despite that one can always label newly generated data and re-train the models to address the problem, however, such trivial and repeated labor works are very expensive and time-consuming.

In contrast, zero-shot learning (ZSL) designs a model to simulate the way humans predict, which differs from traditional machine learning methods. Specifically, a shared semantic feature space between seen and novel unseen classes, e.g., word vector space [8], [9], [10], [11], [12], is exploited as the bridge to fill the gap between them. In practice, the ZSL setting can be regarded as an extreme case of transfer learning, that can generalize well to novel concepts that haven’t been observed during training. The taxonomy of ZSL mainly includes conventional zero-shot
learning (CZSL) and generalized zero-shot learning (GZSL). In CZSL, the testing images are limited to unseen classes. However, this setting is not that practical in real-world application since one may also frequently encounter samples from seen classes during inference. In contrast, the GZSL setting requires the model to generalize well to both seen and unseen classes, which is a more challenging scenario in ZSL due to the following two reasons. First, the training samples only contain seen classes while the testing samples contain both seen and unseen ones, it can greatly increase the difficulty of prediction. Second and more importantly, if high overlap exists in the feature space between seen and unseen classes, the model may easily shift to seen classes, i.e., wrongly classify samples of unseen classes to seen classes, and degrade the recognition performance dramatically.

Existing state-of-the-art ZSL models usually focus on predicting single label for each image [8], [13], [14], [15], [16], [17], [18], [19], [20], which obviously cannot present all useful information within it, and thus may not be that practical for real-world applications where the multi-label information is urgently needed. The challenge of multi-label prediction in ZSL mainly lies in the complex contents and relationships in a single image across various classes. To overcome this challenge and construct a more applicable recognition system, our interest is to investigate the multi-label zero-shot learning (ML-ZSL), that can recognize multiple objects simultaneously from seen and unseen classes in just one image. Recently, a few works have addressed the problem of ML-ZSL by using techniques such as attention-sharing [21], structured knowledge graphs [22], convex combinations [12], GAN-based features, multi-scale features [23] synthesis [24], and global feature generation [25], [26], and obtained some promising results. Specifically, Norouzi et al. [12] designed a new multi-label classifier and fused multi-label features to generate combined features to represent images. The combined image features are then fed into the word vector space to rank the labels. Zhang et al. [26] trained the network to make the input image features have a main direction, wherein, the relevant labels are ranked better than the irrelevant labels, so as to determine the unseen classes present in the image. It is noticed that, the idea of the above two methods, i.e., [12], [26], is to fuse multi-class label features. However, if more classes exist in the images, the recognition accuracy may be greatly reduced due to the blurring of single-class features. Alternatively, Deep0tag [23] captures scene information and local information such as objects and attributes by using multiple layers of features. At the same time, it also has the ability of context information aggregation and semantic information combination.

Then, LESA [21] used multiple spatial attentions to do the feature extraction and sharing, which can better attend to relevant images regions. Differently, Gupta et al. [24] introduced the generative model into the multi-label (generalized) zero-shot learning problem, which can generate multi-label visual features with consistent semantics by synthesizing attributes of different classes, to achieve the purpose of prediction. The above two methods, i.e., [21], [24], can distinguish and fuse multiple features through the attention mechanism and the generative model, respectively, which greatly improves the model’s ability. However, they still ignore constructing effective semantic correlation between features, which may generate less accurate semantic representations for input samples. Meanwhile, none of them have considered improving the recognition accuracy of seen classes. Worse still, most existing methods usually tend to overly concentrate on one or a limited number of labels, which cannot deal with all attended objects in an equal manner. Such a limitation may result in a biased model that can easily recognize a few major classes while neglecting most of the minor classes. Specifically, the major class in our manuscript refers to the class that appears frequently in the image and has a relatively large scale. Such classes can be easily identified (both detected and recognized) by the model, thus may also introduce severe classification bias into the trained model towards such classes. In contrast, the minor class in our manuscript is defined as the classes that usually appear in inconspicuous locations in the image, and the objects themselves are usually smaller and have a fairly small number of samples. Meanwhile, none of them can extensively nor try to make utilization of the semantic diversity, which may suppress the representation ability of generated semantic vectors of input samples.

To deal with the above problems and construct an unbiased ML-ZSL model, we propose to fully make use of various class-specific regions to calibrate the model training process. Specifically, we propose Pyramid Feature Attention (PFA) module to integrate multi-scale features into the extraction process to balance the correlation between minor and major classes (Fig. 1), hence the network can pay equal attention to both of them and improves the recognition accuracy. Conceptually, the major classes refer to the larger objects within an image, and these objects are usually tend to be valued in the “Deep Layers” of the network because of the continuous convolution and pooling operations. Existing multi-label zero-shot learning methods
and Gupta et al. [24], we extend extensive experiments on large-scale ML-ZSL benchmark datasets, i.e., NUS-WIDE and Open-Images, to verify the effectiveness of our method along with state-of-the-art results.

In summary, our contributions are four-fold:

- First, our method elaborates on the equal presence of major and minor classes, which is considered as the major contribution for existing ML-ZSL research. Concretely, previous methods are usually biased on major classes and may thus degrade the recognition of minor classes. To solve such an issue, we consider the multi-scale extraction in a model, where we found that the shallow layers may contain more details on minor classes while the deeper layers may eventually converge to major classes due to the local receptive fields and continuous poolings. Such a consideration is also the first attempt in ML-ZSL.

- Second, our method further elaborates on the element-wise correlation of generated semantic vectors. Previous methods usually neglect such a fine-grained and cross-class consideration on semantic vectors, and may thus generate (or partially) uncorrelated representations of samples. To solve such an issue, we consider building a semantic matrix covering the element-wise and cross-class correlations among semantic vectors in an end-to-end manner. Such a consideration can encourage the model to generate more coordinated representations of samples, which is the main motivation of SA. A related consideration is the attention sharing by considering element-wise semantic vectors.

- Third, the proposed PFA and SA can be regarded as an integrated solution, where PFA pays attention to the balance between classes to reduce bias, and SA generates better semantic information for each class.

- Extensive experiments on large-scale ML-ZSL benchmark datasets, i.e., NUS-WIDE and Open-Images, verified the effectiveness of our method along with state-of-the-art results.

II. RELATED WORK

A. Multi-Label Classification

With the continuous development of deep learning techniques, the multi-label classification has made remarkable achievements. Concretely, existing methods can be roughly divided into three directions: classification with convolutional neural network (CNN) based models such as [1], [2], [3], [4], [5], [7], graph neural network (GNN) based models such as [6], [27], and recurrent neural network (RNN) based models such as [28], [29].

The GNN-based multi-label classification aims at establishing associations between different labels through a graph network based on prior information, which is helpful to infer unseen or rarely observed objects. However, the graph propagation is significantly time-consuming and may result in overly smooth feature representations. The RNN-based methods can establish connections between different labels through the time series, and sorting the labels by the frequency. It can help build the relationship much faster. However, the addition of this prior information often ignores the natural attributes of labels, which eventually leads to biased prediction results. The main-stream of CNN-based methods usually resort to explore the attention distribution of the corresponding image regions of different classes. Such kind of spatial attention mechanism can roughly find the positions corresponding to the classes, and achieve relatively good recognition performance. However, the pure spatial attention lacks the ability to generalize to unseen labels, thus it is difficult to be used to predict unseen classes. Other CNN-based methods, such as multi-label object detection [30], [31], [32], [33], resort to locate objects through region proposal. However, the premise is that the dataset needs to provide a bounding box containing the object as training input to locate the label, which is very time-consuming.

B. Zero-Shot Learning

The emergence of ZSL [1], [2], [3], [4], [5], [6], [7] is to enable the network to predict unseen classes and improve the generalization ability of the network. This is typically achieved by constructing and exploring the correlation between seen and unseen classes, i.e., the visual-label semantic space generated by the word vectors or the attribute vectors. Practically, an image containing one or more unseen classes is first fed into a classifier trained by seen classes, to obtain its semantic vector. Such a semantic vector is then sent into the semantic space to classify the unseen classes through similarity calculation with both seen and unseen classes. Recently, several advanced methods also tried to model the parts of the image using attention mechanism and obtained superior results [34], [35], [36], [37]. Based on the predicted labels/classes, existing ZSL can be divided into single-label zero-shot learning and multi-label zero-shot learning (ML-ZSL).

1) Single-Label Zero-Shot Learning: Single-label prediction is the very first and most widely studied scenario in ZSL. Early works of ZSL usually explore efficient and robust mapping
between visual and semantic feature space to achieve the recognition [8], [16], [38], [39], [40], [41], [42], [43], [44], [45], [46], [47]. Most recently, Rahman et al. [48] developed an effective and unified method for both zero-shot and few-shot learning by inducing the concept of class adapting principal directions that make the embeddings of unseen class images in the semantic space more discriminative. Guo et al. [20] proposed a novel model to align the manifold structures between the visual and semantic feature spaces via expansion of semantic features. Differently, to enhance the discriminability on both seen and unseen domains, Zhang et al. [49] developed a systematical solution via separately learning visual prototypes and proposed an efficient solution. To mitigate biases towards seen classes and accommodate diverse tasks, Li et al. [50] proposed an attribute-aware modulation network to remedy the defect of meta generative approaches, which is devoted to explore the common model shared across task distributions. Similarly, Min et al. [51] proposed dual-cycle consistency and domain division constraints to make obtain the domain similarity and specialities to overcome biases. Although good results have been achieved, these methods generally lack generalization capabilities and cannot directly migrate the methods to multi-label scenarios.

2) Multi-Label Zero-Shot Learning: In recent years, the multi-label prediction has gained increasing attention in ZSL. For example, Norouzi et al. [12] proposed to directly combine the image classifier and the semantic word embedding model into a new model, and the output of the classifier is convexly combined and then sent to the semantic embedding space for prediction. Zhang et al. [26] used a linear mapping and a non-linear deep neural network to approximate the principal direction of images. The principal direction is regarded as the main information contained in the image. The word vector of the relevant labels of the image is ranked ahead of irrelevant labels. Differently, Lee et al. [22] designed a new framework based on the knowledge graph to construct a knowledge graph between multiple labels. At the same time, a new information propagation mechanism is learned from semantic information to model the relationship between seen and unseen labels. Ji et al. [52] proposed a flexible framework, consisting of two modules: a visual-semantic regression unit and multi-label zero-shot prediction unit, which aims to embed the visual features to semantic space and solve the prediction as the ranking problem.

Rahman et al. [23] found that most of the labels in multi-label zero-shot learning are related to multi-scale information such as objects and scenes, so a multi-scale information processing module was introduced into the model. Most recently, LESA [21] developed a new shared attention model, which lets the unseen labels select among a set of shared attentions. Although this model can focus the network on key areas, it lacks the integration of global and local information, so the class balance is affected. In addition, it has a very complicated loss function, which is not conducive to improvements. Gupta et al. [24] proposed a generative method to explore the multi-label feature synthesis problem via integrating three different fusion solutions with two representative generative architectures, to compute reliable attention maps for unseen classes.

3) Our Method: In this paper, we observe that existing methods usually fail or degrade their performance in predicting images that with more minor classes. We argue the global and local information, i.e., corresponds to major and minor classes, should be given the same importance to balance and calibrate the recognition on minor classes and further improve the overall results towards unseen classes. To the best of our knowledge, we are the first to jointly consider the multi-scales of feature extraction and element-wise coordination of generated semantic vectors, which can well mitigate the minor class recognition degradation and increase the semantic diversity. Moreover, our method can achieve state-of-the-art results without a large-scale backbone network.

III. PROPOSED METHOD

A. Problem Setting

We start by formalizing the task of multi-label zero-shot learning, and then introduce our proposed method in details. Let us denote the seen labels of a dataset as $C_s$ and unseen labels as $C_u$, where the seen labels are the label information contained in the image we know, that can be used in the model training process. During testing, the trained model is expected to (jointly) predict unseen labels without observing them before. Hence, the entire label information during testing is defined as $C = C_s \cup C_u$. Given $n$ input samples $\{(I_1, Y_1), \ldots, (I_n, Y_n)\}$ in the training process, where $I_i$ denotes the $i$-th training image and $Y_i$ denotes the corresponding set of seen labels. Based on the recognition setting, we construct a multi-label zero-shot learning problem that predicts the unseen labels $y_u \subset C_u$ by given an image $I_u$, and a generalized multi-label zero-shot learning problem that can predict both seen and unseen labels $y_u \subset C$ by given an image $I_u$, respectively. Moreover, we also follow the widely adopted assumption that each label can be represented by a semantic word vector \{w^c \}_{c \in C}.

B. Pyramid Feature Generation

The framework of our proposed ML-ZSL method is demonstrated in Fig. 2. Since our method does not have any restriction on the backbone networks, so we simply use VGG19 [53] as an example. In view of the problem that the previous method cannot balance the major and minor classes, we believe that the reason for this imbalance is that the current multi-label zero-shot learning methods only use single high-level features. The small objects have less pixel information and are easy to be ignored during the process of sampling, so it needs to be balanced with large-scale feature maps. To solve this problem, Lin et al. [54] proposed Feature Pyramid Network (FPN), which can handle the problem of multi-label object scale changes in object detection with a very small amount of calculation. On the basis of FPN, Li et al. [55] transforms the multi-layer feature maps into the same size as the bottom feature and concatenates them, then sends the concatenate feature map to the detection module, which greatly improves the detection accuracy of small objects.

Inspired by the above two works while different from [55], we scale the bottom feature to the same scale as the top feature and
Fig. 2. The overview of our proposed multi-label zero-shot learning framework. An input image is first extracted by a CNN backbone into several selected multi-scale feature maps, which are further concatenated and sent to the Pyramid Feature Attention and Semantic Attention, respectively. The obtained semantic vectors are then mapped into the semantic feature space to search the corresponding labels of both seen and unseen classes.

then concatenate the multi-scale feature map. The purpose of this approach is to use the sensitivity of large-scale feature maps to small objects to localize minor classes. At the same time, the small-scale feature map will make it easier to locate the major classes, and then scale the feature maps of different levels to the same size for connection, and the obtained new combined features will be able to greatly guarantee the balance between major and minor classes. As can be seen from Fig. 1, there are objects of different sizes in the image, such as the smaller 'cup,' the 'photo' and the larger 'stove.' These labels will all be associated with the label 'kitchen.' Therefore, we can easily find that if only the highest-level feature is used, the model cannot obtain the exact position of the 'cup,' and thus cannot accurately learn the feature. Therefore, the feature maps of lower layers can play the role of localization. In addition, the cascade of the forward pass can reduce the size of the generated feature map compared to [55], reducing the amount of computation required by the model itself. At the same time, smaller-scale feature maps are also beneficial to the extraction of semantic information. This forward feature fusion method can be expressed by the following function:

\[
F_{\text{fixed}} = \text{MaxPooling}(F_i),
\]

where \( F_i \triangleq [f_i^3, f_i^4] \) represents the feature layer \( f_i^v \) that needs to be extracted, and \( v \) represents the set of all feature layers to be extracted, usually the third and fourth different scales and their feature map sizes are \( 28 \times 28 \) and \( 14 \times 14 \), respectively. \( i \) means the input image. \( \text{MaxPooling} \) means the scale matching operation, we choose the max-pooling operation to highlight the local information of the bottom feature map. Then the concatenation is conducted as:

\[
C_i = \text{Concat}(F_{\text{fixed}}),
\]

where \( \text{Concat} \) stands for concatenating different feature maps together to generate a new feature map \( C_i \).

C. Pyramid Feature Attention

After obtaining the multi-scale feature map \( C_i \in \mathbb{R}^{H \times W \times C'} \), we introduce the Pyramid Feature Attention (PFA) to focus on the correlation among different channels. In this step, since the model has obtained a sufficient number of features in the previous cascade operation, the number of channels is also greatly increased. The increase of feature dimension will cause the model itself to be unable to judge the importance between different channels. Therefore, we adopt the feature channel attention mechanism to strengthen the channels with more important information and weaken the channels that are not important, thereby enhancing the learning ability of the model for important information. We hope that the model can automatically learn the importance of different channel features, which is conducive to the subsequent generation of semantic vectors. The structure of this attention helps to enhance the network’s attention to minor classes by extracting features of different scales. The PFA is expressed by the following formula as:

\[
C' = \text{Map}(C_i),
\]

where for the input feature map \( C_i \), we get the new feature map \( C' \) after the mapping operation \( \text{Map} \). The mapping operation \( \text{Map} \) denotes the feature map \( C_i \) generated by Pyramid Feature
Generation in the previous step, and the number of channels is updated to \( M \times d_w \). This approach can reduce the amount of computation for subsequent operations. Then we perform global average pooling \( \text{GAPooling} \) to generate channel-wise information as:

\[
z = \text{GAPooling} \left( C' \right) = \frac{1}{H' \times W'} \sum_{i=1}^{H'} \sum_{j=1}^{W'} c(i,j).
\]  

(4)

After that we can get \( z \), the multi-scale channel attention, next, we reduce the amount of computation by reducing the number of feature layers of the network by using the bottleneck structure. In addition, the bottleneck structure can learn nonlinear interactions between different channels, which helps to improve the correlations among channels, i.e.:

\[
z' = \sigma \left[ W^1_{fc} \left( z \right) \right],
\]  

(5)

where \( W^1_{fc} \in \mathbb{R}^{C' \times C^{\text{ex}}} \) refers to the fully convolutional layer used when \( z \) is channel compressed, and \( z' \) is the compressed attention value. The \( \sigma \) represents the ReLU layer, the purpose is to increase the nonlinear information. Then restore the number of channels to \( C' \):

\[
\alpha = \sigma \left[ W^2_{fc} \left( z' \right) \right],
\]  

(6)

where \( W^2_{fc} \in \mathbb{R}^{C' \times C^{\text{ex}}} \) refers to the fully convolutional layer used to change the number of channels back to \( C' \). Then perform channel-wise multiplication with \( C' \) to generate the PFA feature:

\[
F = \alpha \otimes C' = [\alpha_1(c_1') \ldots \alpha_{C'}(c_{C'})].
\]  

(7)

Finally, we use global average pooling to change \( F \) from \( \mathbb{R}^{H' \times W' \times C'} \) to \( \mathbb{R}^{1 \times 1 \times C'} \). Then we train a linear transformation to make the number of channels become \( M \times d_w \), where \( M \) is the number of generated semantic vectors, and \( d_w \) is the dimension of the semantic vectors, usually the same length as \( \psi \) (which is 300):

\[
A = \delta \left[ \text{GAPooling} \left( F \right) \right],
\]  

(8)

where \( \delta \) is a linear transformation. Then we get the semantic matrix \( A \in \mathbb{R}^{M \times d_w} \). The semantic matrix contains the semantic information of all seen labels.

D. Semantic Attention

For the generated semantic matrix \( A \in \mathbb{R}^{M \times d_w} \), its actual meaning is not obvious because it is only a simple segmentation of the output channels. At the same time, this segmentation operation severely destroys the global information covered by each semantic vector, thus causing troubles for our prediction. Therefore, in this section, we introduce how to add global information to the obtained semantic vectors by using our proposed Semantic Attention (SA).

First, we normalize the generated \( A \) by LayerNorm and convert it into embedding vectors \( Q_a, K_a, V_a \):

\[
Q_a = A W^Q_a,
\]  

(9)

\[
K_a = A W^K_a,
\]  

(10)

\[
V_a = A W^V_a,
\]  

(11)

where \( W^Q_a, W^K_a, W^V_a \) are weights of which the number of channels are \( d_w \). Since the generated \( M \) semantic vectors are almost uncorrelated with each other at the element-wise level, we further resort to building the correlations among them by integrating a self-attention module to strengthen their coordination. Our motivation is that self-attention can reduce the dependence on external information and better capture internal correlations among features. To do it, we first calculate the similarity or correlation between different semantic vectors based on Query \( Q_a \) and Key \( K_a \) as:

\[
r_a = \text{softmax} \left( \frac{Q_a K_a^\top}{\sqrt{d_w}} \right).
\]  

(12)

The second stage normalizes the above obtained scores as:

\[
\theta = r_a V_a.
\]  

(13)

After that, We add the generated \( \theta \) and \( A \) to a multi-layer perceptron (MLP). The purpose of MLP is to increase the non-linearity of the model while being able to perceive the local features of different vectors. Finally we get \( \mathcal{S} \). Semantic Attention is different from the traditional vision transformer [56]. We are not dealing with different image patches, so positional encoding is not needed, and we also abandon the class/category token. We build the Multi-Head Attention operation to enrich the global information of each semantic vector. The generated new semantic matrix \( \mathcal{S} \in \mathbb{R}^{M \times d_w} \) has the same dimension as \( A \).

The specific role of Semantic Attention is that these generated semantic representations (embeddings) of each input data can establish element-wise connections in the semantic feature space with each other, and these connections contain both category connections and global information connections which may facilitate the robustness of the visual-semantic projection/mapping in the learned network, and thus improve the recognition accuracy. Specifically, as demonstrated in (12), we obtain the similarity score between different semantic embeddings by computing \( r_a \). This vector can represent the correlation between semantic embeddings, and the generated \( \theta \) represents the new semantic information generated by the combination of different embeddings.

E. Loss Function

During the training phase, for each input image, we output a semantic matrix \( \mathcal{S} \in \mathbb{R}^{M \times d_w} \), which is used to determine the unseen class contained in the image. Therefore, inspired by [26], when calculating class similarity, we need to provide a higher ranking for the positive class, and a lower-ranking for the negative class, as the basis for the design of our loss function:

\[
m_{jk} = \max \left( \mathcal{S} n_j \right) - \max \left( \mathcal{S} p_k \right),
\]  

(14)

where \( n_j \) is the word vector of the negative class, and \( p_k \) is the word vector of the positive class. We choose the \( \max \) result for calculation, the purpose is to make all classes output high scores while contributing to semantic diversity. Similar ideas are also used in multi-label support vector machines (SVM) [57]. Next, inspired by [26], we design the following modified ranking loss.
as:
\[
    \mathcal{L}_{\text{rank}} = \beta \sum_{j} \sum_{k} \log \left(1 + e^{p_{jk}}\right),
\]
where \( \beta = (|Y|/|\bar{Y}|)^{-1} \). \(|Y|\) and \(|\bar{Y}|\) denote the number of seen and unseen classes, respectively. The hyper-parameter \( \beta \) is used to normalize the ranking loss.

In addition, in order to reduce the difficulty of classifying images with high label diversity, we propose to add a new weight to the ranking loss as:
\[
    \omega = 1 + \sum_{i} \text{var}(Y_i).
\]
As to the weight in (16). It is known that in the practice of multi-label learning, the number of labels (classes) contained in each image is usually not the same. Consequently, it is difficult to completely identify all the labels from those images with large number of labels. To mitigate such an issue, the motivation is to treat images with different number of labels unequally in the training process, i.e., by adding a weight to the loss function, and as such, to encourage the model to pay more attention to difficult data samples. Moreover, in order to balance the semantic diversity between different vectors and encourage the network to learn relevant information between different labels, we further construct a regularization loss as:
\[
    \mathcal{L}_{\text{reg}} = \left\| \sum_{n} \text{var}(S_n) \right\|_1.
\]
In summary, our final loss function for each image is defined as:
\[
    \mathcal{L}_{\text{final}} = \frac{1}{N} \sum_{i=1}^{N} (w \cdot (1 - \lambda) \mathcal{L}_{\text{rank}}(S_i, Y_i) + \lambda \mathcal{L}_{\text{reg}}(S_i)),
\]
where \( N \) is the value of batch size, and \( \lambda \) is a hyper-parameter that regularize the weights.

IV. EXPERIMENTS

A. Experimental Setup

1) Datasets: First, we evaluate our experiments on NUS-WIDE [58]. The NUS-WIDE dataset contains 270$k images which are annotated by human into 81 classes. These 81 classes are also called ‘ground-truth’ labels. Besides, each image also has 925 labels which are extracted from Flickr user tags. Similar to the previous works [21], [26], we choose the 925 labels as the seen labels, and the other 81 human-annotated labels as unseen. Next, We also test the ZSL and GZSL tasks separately on the larger multi-label dataset Open-Images-V4 [59], which contains nearly 9 million training images and 125,436 test images, as well as 41,620 images as the validation set. The training set contains 7186 visible labels, and each label contains at least 100 samples for training. In the test set, we select the top 400 most frequent classes not present in the training data, each unseen label has at least 75 samples for detection. Due to the large number of classes, each image has at least one unannotated label. Finally, is the MS COCO dataset [60], as another and the most popular multi-label dataset, it contains 82,783 images for training, and the validation set contains 40,504 images. The images including 80 categories.

2) Evaluation Metrics: In order to measure the performance of our proposed model under the multi-label zero-shot classification task, we use the mean Average Precision (mAP) [61] and F1 score at top-K predictions to evaluate. Our evaluation metric top-K F1 score requires the output of top-3 and top-5 categories, according to the formulas of Precision (P) and Recall (R):
\[
    P = \frac{TP}{TP + FP},
\]
\[
    R = \frac{TP}{TP + TN},
\]
Among them, \( TP \) refers to ‘True Positive,’ \( FP \) refers to ‘False Positive,’ and \( TN \) denotes ‘True Negative’. The \( P \) is the correct ratio of the true values in the data predicted to be correct, and the \( R \) denotes that how many data with correct true values can be predicted correctly. Notably, the calculation formula of F1-Score is directly affected by both \( P \) and \( R \):
\[
    F1-Score = \frac{2 \times P \times R}{P + R}.
\]
The top-K F1 score is used to measure the accuracy of the model for predicting the labeling of the image, while the mAP measurement is based on the accuracy of label retrieval, showing the model’s ranking accuracy for each label of the image. Specifically, the mAP needs to calculate the area enclosed by each class in the PR curve and average over all classes to get the final result.

3) Implementation Details: Unlike [21] and [26], this is an end-to-end training model. We use the backbone network VGG19 pre-trained on the ImageNet dataset [62]. During the cascade operation of multi-scale feature maps, we extract feature maps with sizes of \( 28 \times 28, 14 \times 14 \), and \( 7 \times 7 \) respectively, and use max-pooling to directly sample the bottom feature maps to \( 7 \times 7 \). Compared with FSSD [55], this operation greatly reduces the amount of calculation. The feature map size for pyramid feature attention is \( 2048 \times 7 \times 7 \). We choose the Adam optimizer [63] for the model training, and the maximum learning rate is set as \( 1e^{-5} \), which is reduced to \( \frac{1}{10} \) of the previous after 5 epochs. The weight decay is \( 4e^{-3} \). In the loss function, the value of the regularization term \( \lambda \) is 0.4, and the number \( M \) of semantic vectors generated by the model is 8. The number of epochs for experiments in the NUS-WIDE dataset is 10, and batch_size is set to 64.

4) Baselines: We compare our method with several representative ML-ZSL methods including CONSE [12], LabelEM [10], Fast0Tag [26], Kim et al. [64], Deep0Tag [23], LESA [21], and Gupta et al. [24]. The selection of our competitors are based on the following criteria: 1) All competitors are published or presented in the most recent years; 2) A wide range of techniques are covered; 3) They have stated or demonstrated the state-of-the-art in ML-ZSL; and 4) All competitors are evaluated fairly under the same criteria with their best (reported) results. Moreover, we also measure and analyze our method. Specifically, we not only adjust the number of \( M \) but also choose the best regularization.
TABLE I
STATE-OF-THE-ART COMPARISON FOR MULTI-LABEL ZSL AND GZSL TASKS ON THE NUS-WIDE DATASET IN DETAIL.

| Method        | Task | P (K = 3) | R (K = 3) | F1 (K = 3) | P (K = 5) | R (K = 5) | F1 (K = 5) | mAP |
|---------------|------|-----------|-----------|------------|-----------|-----------|------------|-----|
| CONSE [12]    | ZSL  | 17.5      | 28.0      | 21.6       | 13.9      | 37.0      | 20.2       | 9.4 |
|               | GZSL | 11.5      | 5.1       | 7.0        | 9.6       | 7.1       | 8.1        | 2.1 |
| LabelEM [10]  | ZSL  | 15.6      | 25.0      | 19.2       | 13.4      | 35.7      | 19.5       | 7.1 |
|               | GZSL | 15.5      | 6.8       | 9.5        | 13.4      | 9.8       | 11.3       | 2.2 |
| Fast0Tag [26] | ZSL  | 22.6      | 36.2      | 27.8       | 18.2      | 48.4      | 26.4       | 15.1|
|               | GZSL | 18.8      | 8.3       | 11.5       | 15.9      | 11.7      | 13.5       | 3.7 |
| Attention per Label [65] | ZSL  | 20.9      | 33.5      | 25.8       | 16.2      | 43.2      | 23.6       | 10.4|
|               | GZSL | 17.9      | 7.9       | 10.9       | 15.6      | 11.5      | 13.2       | 3.7 |
| Deep0Tag [23] | ZSL  | 25.7      | 43.8      | 32.4       | 17.2      | 48.9      | 25.5       | -   |
|               | GZSL | 33.8      | 13.1      | 18.9       | 23.7      | 15.3      | 18.5       | -   |
| Attention per Cluster [21] | ZSL  | 20.0      | 31.9      | 24.6       | 15.7      | 41.9      | 22.9       | 12.9|
|               | GZSL | 10.4      | 4.6       | 6.4        | 9.1       | 6.7       | 7.7        | 2.6 |
| LESA (M = 10) [21] | ZSL  | 25.7      | 41.1      | 31.6       | 19.7      | 52.5      | 28.7       | 19.4|
|               | GZSL | 23.6      | 10.4      | 14.4       | 19.8      | 14.6      | 16.8       | 5.6 |
| Gupta et al. [24] | ZSL  | 26.6      | 42.8      | 32.8       | 20.1      | 53.6      | 29.3       | 25.7|
|               | GZSL | 30.9      | 13.6      | 18.9       | 26.0      | 19.1      | 22.0       | 8.9 |
| Our Approach  | ZSL  | 34.0      | 42.3      | 37.7       | 26.7      | 55.3      | 36.0       | 28.0|
|               | GZSL | 31.2      | 13.9      | 19.2       | 26.4      | 19.6      | 22.5       | 9.3 |

We show the indicators of F1-score in the case of k ∈ {3, 5} and mAP. We can clearly see that our method has surpassed all state-of-the-art methods, whether it is under ZSL or GZSL conditions. The “P(K)” refer to “precision” and “R(K)” refers to “recall.” Best results are shown in bold.

B. State-of-The-Art Comparison

1) Result on NUS-WIDE Dataset: Table I shows the comparison between our model and other current state-of-the-art methods on the NUS-WIDE dataset. Not only the traditional zero-shot learning (ZSL), but also the results of generalized zero-shot learning (GZSL) are also reported. Due to the limitation of space, we only show the most important Top-K (K = 3, 5) F1 Score and mAP results.

For the ZSL task, the performance of CONSE [12] and LabelEM [10] on the NUS-WIDE dataset, whether it is mAP or F1-Score, cannot be compared with the current methods. Because Fast0Tag [26] learns to choose the main direction of the image in the word vector space, the mAP is improved by 5.7% compared to CONSE. For the newly proposed method LESA [21], which introduces a spatial attention mechanism, predicts unseen labels by splitting the image into patches and sharing the attention between different them. It also achieved the best results before. But our approach is now the best, leading LESA by 8.6% in mAP and at least 6.1% improvement in F1-Score. In subsequent comparisons with Generative ZSL, i.e., Gupta et al. [24], our method achieves the best results in all evaluation metrics. In GZSL, as the task becomes more difficult, the gap between the various methods is not particularly obvious. In the mAP evaluation metric, our method also achieved the best results, which was 0.4% higher than that of Gupta et al. [24].

2) Results on Open-Images Dataset: Table II shows the performance of our method and all current state-of-the-art comparison methods on the Open-Images dataset. Like NUS-WIDE, we compare on two tasks, ZSL and GZSL. We show the most important Top-K (K = 10, 20) F1 Score and mAP results. Each image in Open-Images has at least one class, and because the total number of classes is too large, the semantic space will also become more complex, making model prediction more difficult.

On the ZSL task, all methods perform poorly in precision. The mAP results of CONSE [12] and LabelEM [10] are only 0.2%, while other attention-based comparison models, such as Fast0Tag [26] and LESA [21] are not more than 1%. And Gupta et al. [24] based on generative models is the only one that exceeds 1%. However, our method achieves the best results at the precision of K = 10, 20. In addition, the recall of our model has also been greatly improved compared to Gupta et al. [24]. The final mAP also proves the effectiveness of our method on the largest dataset from another dimension.

In the GZSL task, due to the huge increase in the number of classes in the test session, the number of predictable labels in each image is increased by at least 40 times. Therefore, the numerical results of all methods are greatly improved in the case of K = 10, 20. Compared with the current state-of-the-art
TABLE II
STATE-OF-THE-ART COMPARISON FOR MULTI-LABEL ZSL AND GZSL TASKS ON THE OPEN-IMAGES DATASET IN DETAIL.

| Method       | Task  | P (K = 10) | R (K = 10) | F1 (K = 10) | P (K = 20) | R (K = 20) | F1 (K = 20) | mAP |
|--------------|-------|------------|------------|-------------|------------|------------|-------------|-----|
| CONSE [12]   | ZSL   | 0.2        | 7.3        | 0.4         | 0.2        | 11.3       | 0.3         | 40.4|
|              | GZSL  | 2.4        | 2.8        | 2.6         | 1.7        | 3.9        | 2.4         | 43.5|
| LabelEM [10] | ZSL   | 0.2        | 8.7        | 0.5         | 0.2        | 15.8       | 0.4         | 40.5|
|              | GZSL  | 4.8        | 5.6        | 5.2         | 3.7        | 8.5        | 5.1         | 45.2|
| FastNet [26] | ZSL   | 0.3        | 12.6       | 0.7         | 0.3        | 21.3       | 0.6         | 41.2|
|              | GZSL  | 14.8       | 17.3       | 16.0        | 9.3        | 21.5       | 12.9        | 45.2|
| Attention per Cluster [21] | ZSL   | 0.6        | 22.9       | 1.2         | 0.4        | 32.4       | 0.9         | 40.7|
|              | GZSL  | 15.7       | 18.3       | 16.9        | 9.6        | 22.4       | 13.5        | 44.9|
| LESA (M = 10) [21] | ZSL   | 0.7        | 25.6       | 1.4         | 0.5        | 37.4       | 1.0         | 41.7|
|              | GZSL  | 16.2       | 18.9       | 17.4        | 10.2       | 23.9       | 14.3        | 45.4|
| Gupta et al. [24] | ZSL   | 1.3        | 42.4       | 2.5         | 1.1        | 52.1       | 2.2         | 43.0|
|              | GZSL  | 33.6       | 38.9       | 36.1        | 22.8       | 52.8       | 31.9        | 49.7|
| Our Approach | ZSL   | 1.5        | 49.9       | 2.9         | 1.1        | 64.0       | 2.2         | 45.4|
|              | GZSL  | 33.9       | 40.1       | 36.7        | 23.1       | 53.1       | 32.2        | 51.5|

We can clearly see that our method has surpassed all state-of-the-art methods, whether it is under ZSL or GZSL conditions. Best results are shown in bold.

Fig. 3. An illustration of the impact of choosing different hyper-parameters for ZSL and GZSL tasks. All ZSL and GZSL tasks are performed on the NUS-WIDE test set.

C. Effect of Hyper-Parameters

Fig. 3(a), Fig. 3(b), and Fig. 3(c) demonstrate the effect of hyper-parameters in our method. Specifically, there are two hyper-parameters in our method, namely, $M$ and $\lambda$, where $M$ is the number of generated semantic vectors and its number determines the breadth of the model’s prediction. Theoretically, the larger $M$ is, the more abundant types of predicted labels will be generated. However, since the amount of information contained in a single image is limited, an increase in $M$ will obscure the information contained in each semantic vector.

From Fig. 3(c), we can conclude that in the ZSL curve, when the value of $M$ increases from 1 to 10, it will first increase the value of mAP, and reach the highest value when $M = 8$. Then as the value of $M$ increases, the mAP begins to fall gently. The GZSL curve is relatively stable than the ZSL curve. It also reaches its peak at $M = 8$, and then drops slightly, but the overall curve is very stable. This phenomenon also better confirms our previous assumptions about the hyper-parameter $M$. Therefore, our model outputs the best result when $M = 8$. At the same time, the change of $M$ will not greatly affect the accuracy of the model itself. Compared with LESA, when $M = 1$ and $M = 10$, the changes brought by mAP are significant.

Next is the hyper-parameter $\lambda$. As a regularization weight, $\lambda$ is very important for the loss function. A good and stable loss function is the ‘teacher’ who guides the model to perform well during training process. First we can see from Fig. 3(a) that the effect of the change of $\lambda$ on F1-Score is very obvious. Especially when $\lambda = 0$ and $\lambda = 1$. When $\lambda = 0$, the loss function of the model loses the regularization term, resulting in the lack of constraints of the generated semantic vectors, resulting in a decrease in accuracy. When $\lambda = 1$, the loss function only contains the regularization term, which is meaningless to the model’s loss, so the F1-Score approaches 0. Fig. 3(b) shows the effect of changes in $\lambda$ on the results of mAP. We can clearly see through that, when $\lambda = 0.4$, the model achieved the best mAP and F1-Score. Except for the two extreme conditions of $\lambda = 0$ and $\lambda = 1$.
and $\lambda = 1$, when $\lambda$ becomes other values, the change in mAP is not obvious. Therefore, the change of $\lambda$ will not affect the model’s ability to recognize labels, but will affect the model’s ability for prediction.

### D. Ablation Study

To better understand the contribution of each component in our method and measure their importance, we demonstrate the ablation study of our method in Table III. In this results, ‘None’ denotes the scenario that we only use VGG19 to generate semantic vectors for unseen classes. It can be seen that after the model uses the pyramid structure, the prediction ability in the ZSL task has been improved by 1.6%, and the GZSL has also been improved by 0.8%, which shows that the rational use of features at different levels has greatly exceeded the efficiency. At the same time, it also proves the validity of our judgment on the existence of major and minor classes. In the use of PFA structure and SA structure, based on the pyramid feature, both PFA and SA can play a positive role in the prediction ability of the model. However, compared with SA, PFA is better matched with pyramid feature. When the two act on the model at the same time, the effect is improved by 1.4% on the ZSL task and 0.7% on the GZSL task compared with only PFA. The change of $M$ and regularization weight in the ablation experiment is to reflect whether different hyper-parameters will have an impact on the prediction results of the model for unseen classes. The PFA and SA have significantly improved the capabilities of the model, and finally achieved the best results when the model reached the most ideal state.

| Module       | M Reg | mAP ZSL | mAP GZSL |
|--------------|-------|---------|----------|
| None         | 8 0.4 | 24.3    | 7.4      |
| Pyramid      | 8 0.4 | 25.9    | 8.2      |
| Pyramid+PFA  | 8 0.4 | 26.6    | 8.6      |
| Pyramid+SA   | 8 0.4 | 26.3    | 8.4      |
| Pyramid+PFA+SA | 3 0.4 | 27.1    | 8.6      |
|              | 8 0.8 | 28.0    | 9.3      |

The baseline methods are performed on the NUS-WIDE test set.

### E. Multi-Label Learning

In this section, in order to explore the performance of this model in the traditional multi-label learning problem, we design a new experiment. Each label contains training images, but the number of samples among labels is very different. Our comparison methods include Logistic Regression [65], WSABIE [4], WARP [2], Fast0Tag [26], CNN-RNN [1], One Attention per Label using Bilinear Attention Network [64], and LESA [21].

Table V shows the F1-Score at $K \in \{3, 5\}$ and mAP values at 81 ‘ground truth’ labels on NUS-WIDE. Obviously, compared with the multi-label zero-shot learning method Fast0Tag [26] and LESA [21], our mAP value is 13.6% higher than that of LESA [21] on the NUS-WIDE dataset, becoming the best performing multi-label zero-shot learning method. In addition, our mAP is also 12.5% higher than the second place One Attention per Label [64], which is the highest mAP among all comparison methods. The success of this experiment shows that our model not only has strong predictive ability, but also has strong competitiveness in the field of classification and recognition.

In order to better verify that our model has strong multi-label recognition ability, we also tested it on the MS-COCO dataset. Following the setting of [22], [23], we choose 40,137 images in the validation set after removing the images without labels. The results shown in Table VI. We compare with some multi-label learning models [2], [4] and multi-label zero-shot learning models [22], [23], [26]. We can see from the experimental results that, both our method and Deep0Tag [23] use multi-scale feature maps for prediction, and we both have achieved very competitive results. We can also find that multi-scale features can greatly improve the multi-label learning task.

### F. Discussions About Using Pre-Trained Model

Does using a model pretrained on ImageNet affect the purity of multi-label zero-shot learning for unseen labels? We elaborate on this issue with the following two facts. First, as summarized in [18], the widely adopted logic of zero-shot learning is to use a pre-trained feature extraction network combined with input seen classes as the training-set to construct a projection or mapping function between visual and semantic feature spaces. Afterwards, the trained model is expected to perform domain adaptation from seen to unseen classes, with only certain kind of side information among classes (e.g., shared attributes or word vectors). Thus, the pre-trained feature extraction network has been an essential component for various zero-shot learning methods.

Second, recent works of zero-shot learning all used the pre-trained CNN backbones as their initial models. Specifically, in the setting of multi-label zero-shot learning, the models such as Huynh et al. [21] and Zhang et al. [26] took the same approach as we did, i.e., using the pre-trained VGG19 as the base feature extraction network, and conducted experiments on both NUS-WIDE and Open-Images datasets to verify the effectiveness of methods.

In addition, we also designed an experiment to verify the impact of the pre-train model on the experimental results. Table VII
is the experimental results of our verification. We can see through the experimental results that the pre-train model can indeed improve the prediction ability of the model to a certain extent, but the improvement is not obvious. Besides, our model can still achieve state-of-the-art performance even without using the pre-trained model.

G. Discussion About Using Different Pyramid Structure

To compare with pyramid models, we have selected two representative works, i.e., FSSD [55] and M2Det [66]. The test set of NUS-WIDE is used in the comparative experiment, and the results are shown in Table VIII. It can be observed from the results that our method can outperform these competitors with large margins, which implies that the existence of max-pooling (utilized in our method) is of crucial importance when recognizing multi-label objects. The complex pyramid structure of M2Det [66] and FSSD [55] is resulted from the fact that it must accurately find the position information of the target object. However, in the recognition task, the overly complex convolutional structure can easily suppress or filter out the semantic features of the object itself.

H. Comparison on Major and Minor Classes

Next, we use the NUS-WIDE dataset as an example to list the major classes and the minor classes, respectively, in Table IX. Specifically, we mainly consider the frequency of class occurrence and the size of the class we usually observe. The frequency of each unseen class in the NUS-WIDE dataset is demonstrated in existing literature [58]. However, there is no statistical explanation for the size and variance of each class in the pixel range. Alternatively, we explain the size from an intuitive point of view combined with the visualization results (i.e., some examples are demonstrated in Fig. 5).

Then, we provide a set of comparative experiments to verify that our method can improve the recognition performance of the minor classes, while still maintains the recognition performance of the major classes as well. The results are demonstrated in Table X. We can observe that our model outperforms the current state-of-the-art (SOTA) attention sharing model LESA [21] with large margins in predicting both major and minor classes. Notably, this improvement is especially successful in minor classes such as ‘flags,’ ‘flowers,’ ‘military,’ ‘toys,’ etc. For the classes that reflect the detail texture of images, such as ‘flowers,’ ‘grass,’ ‘tattoo,’ etc., our model also far outperforms LESA [21].

I. Qualitative Results

In this paper, our method generates a set of semantic vectors that represent many different principal orientations of test images in the visual label semantic space. These vectors are calculated with the semantics in the space, and the closest top 10 classes are obtained as prediction results, so that our model has the ability to predict seen and unseen labels. These labels include not only nouns, but also abstract labels that are difficult to classify in recognition tasks, such as adjectives and verbs. For example, in Fig. 4, in the first picture, we can see that we have successfully predicted ‘bird’ and ‘animal’.
TABLE VII
COMPARISON WITH/WITHOUT PRE-TRAINING ON NUS-WIDE TEST SET

| Model       | Task | P (K = 3) | R (K = 3) | F1 (K = 3) | P (K = 5) | R (K = 5) | F1 (K = 5) | mAP |
|-------------|------|-----------|-----------|------------|-----------|-----------|------------|-----|
| Without pre-training | ZSL  | 32.7      | 41.1      | 36.4       | 24.3      | 53.7      | 33.5       | 26.5|
|             | GZSL | 29.5      | 12.9      | 18.0       | 25.1      | 18.6      | 21.4       | 8.8 |

| With pre-training | ZSL  | 34.0      | 42.3      | 37.7       | 26.7      | 55.3      | 36.0       | 28.0|
|                  | GZSL | 31.2      | 13.9      | 19.2       | 26.4      | 19.6      | 22.5       | 9.3 |

TABLE VIII
COMPARISON USING DIFFERENT PYRAMID MODELS

| Pyramid Model | Task | P (K = 3) | R (K = 3) | F1 (K = 3) | P (K = 5) | R (K = 5) | F1 (K = 5) | mAP |
|---------------|------|-----------|-----------|------------|-----------|-----------|------------|-----|
| FSSD [56]     | ZSL  | 27.9      | 37.7      | 32.1       | 23.1      | 50.6      | 31.7       | 25.5|
|               | GZSL | 25.9      | 13.3      | 17.6       | 23.5      | 17.2      | 19.9       | 8.3 |
| M2Det [67]    | ZSL  | 29.1      | 36.2      | 32.3       | 23.7      | 49.2      | 32.0       | 26.1|
|               | GZSL | 26.7      | 12.1      | 16.7       | 24.2      | 16.9      | 19.9       | 8.5 |
| Our Approach  | ZSL  | 34.0      | 42.3      | 37.7       | 26.7      | 55.3      | 36.0       | 28.0|
|               | GZSL | 31.2      | 13.9      | 19.2       | 26.4      | 19.6      | 22.5       | 9.3 |

Fig. 4. Qualitative results of our method, which shows the top-10 predicted labels. Asterisks mark the unseen labels, while bold text represents the correct labels according to the provided ground truth in NUS-WIDE test set.

TABLE IX
MAJOR CLASSES AND MINOR CLASSES

| Major Class  | Minor Class  |
|--------------|--------------|
| animal       | birds        |
| buildings    | coral        |
| clouds       | flags        |
| grass        | flowers      |
| person       | fish         |
| plants       | leaf         |
| road         | military     |
| sky          | soccer       |
| water        | tattoo       |
| window       | toy          |

TABLE X
ZERO-SHOT COMPARISON ON MAJOR CLASSES AND MINOR CLASSES

| Major Classes | Minor Classes | mAP  | mAP  |
|---------------|---------------|------|------|
|               | LESA          | Ours | LESA | Ours |
| animal        | 80.9          | 89.8 | birds | 34.5 | 51.4 |
| buildings     | 33.3          | 48.3 | coral | 34.4 | 43.7 |
| clouds        | 64.5          | 64.9 | flags | 2.5  | 26.3 |
| grass         | 50.9          | 52.5 | flowers | 49.3 | 64.4 |
| person        | 76.1          | 77.1 | fish  | 13.1 | 28.4 |
| plants        | 28.2          | 32.0 | leaf  | 5.2  | 8.5  |
| road          | 12.1          | 16.7 | military | 18.3 | 37.1 |
| sky           | 74.3          | 76.7 | soccer | 6.9  | 9.2  |
| water         | 51.8          | 59.7 | tattoo | 1.0  | 3.0  |
| window        | 16.4          | 31.1 | toy    | 12.9 | 38.1 |

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
In the seventh image, our model is able to make good use of the smaller feature maps to obtain blur information to recognize fog. In the last image, the model accurately predicted ‘flower’ and ‘butterfly’ while also being able to associate them with the adjective ‘bravo’. It shows that our model can accurately find the best matching classes in the semantic space when generating semantic vectors. In summary, we use PFA to balance the major and minor classes, and SA can establish a relationship with other generated vectors while ensuring that the generated semantic vectors retain their original information. This enables each individual semantic vector with more accurate approximation on the predicted classes in the semantic feature space. Last, we provide the quantitative comparison of the SOTA LEA [21] and our method on Grad-CAM.

Specifically, Fig. 5 shows the attention visualization comparisons in 7 sets of images. For image (a), our model can distinguish ‘fish’ and ‘water’ regions very well. For image (b), the response to ‘window’ and ‘house’ is different. Specifically, the window’s response is more concentrated in small regions, while the house’s attention area is wider. In addition, for image (d), the attentions of major and minor classes are more obvious, for example, there exist large discrimination between ‘person’ and ‘soccer’. When our model predicts the class ‘person’, it does not pay attention to the ball, while the class ‘soccer’ successfully incorporates the ball into its attention area.

V. CONCLUSION

In this paper, we proposed a novel unbiased multi-label zero-shot learning framework involving the proposed Pyramid Feature Attention and Semantic Attention, to jointly address the problem of imbalance between major classes and minor classes by extracting the attention between multi-scale feature layers. Extensive experiments on the large-scale NUS-WIDE and Open-Images datasets show that our framework has achieved state-of-the-art results and significantly improved the classification accuracy in the multi-label environment.

REFERENCES

[1] J. Wang et al., “Cnn-rnn: A unified framework for multi-label image classification,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2016, pp. 2265–2274.
[2] Y. Gong, Y. Jia, T. Leung, A. Toshev, and S. Ioffe, “Deep convolutional ranking for multilabel image annotation,” 2013, arXiv:1312.4994.
[3] H.-F. Yu, P. Jain, P. Kar, and I. Dhillon, “Large-scale multi-label learning with missing labels,” in Proc. Int. Conf. Mach. Learn., 2014, pp. 593–601.
[4] J. Weston, S. Bengio, and N. Usunier, “WSABIE: Scaling up to large vocabulary image annotation,” in Proc. 22nd Int. Joint Conf. Artif. Intell., 2011, pp. 2764–2770.
[5] T. Durand, N. Mehrasa, and G. Mori, “Learning a deep convnet for multi-label classification with partial labels,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2019, pp. 647–657.
[6] Z.-M. Chen, X.-S. Wei, P. Wang, and Y. Guo, “Multi-label image recognition with graph convolutional networks,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., 2019, pp. 5177–5186.
[7] L. Feng, B. An, and S. He, “Collaboration based multi-label learning,” in Proc. AAAI Conf. Artif. Intell., 2019, vol. 33, no. 01, pp. 3550–3557.
[8] A. Frome et al., “Device: A deep visual-semantic embedding model,” in Proc. 26th Int. Conf. Neural Inf. Process. Syst., 2013, pp. 2121–2129.
[9] R. Socher et al., “Zero-shot learning through cross-modal transfer,” Adv. Neural Inf. Process. Syst., pp. 935–943, 2013.
[10] Z. Akata, F. Perronnin, Z. Harchaoui, and C. Schmid, “Label-embedding for image classification,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 38, no. 7, pp. 1425–1438, Jul. 2016.
[11] Y. Xian et al., “Latent embeddings for zero-shot classification,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2016, pp. 69–77.
[12] M. Norouzi et al., “Zero-shot learning by convex combination of semantic embeddings,” in Proc. 2nd Int. Conf. Learn. Representations, 2014, pp. 1–9.
[13] Z. Akata, M. Malinowski, M. Fritz, and B. Schiele, “Multi-cue zero-shot learning with strong supervision,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2016, pp. 59–68.
[14] S. Deutsch, S. Kolouri, K. Kim, Y. Owechko, and S. Soatto, “Zero shot learning via multi-scale manifold regularization,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2017, pp. 7112–7119.
[15] P. Morgado and N. Vasconcelos, “Semantically consistent regularization for zero-shot recognition,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2017, pp. 6060–6069.
[16] E. Kodirov, T. Xiang, and S. Gong, “Semantic autoencoder for zero-shot learning,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2017, pp. 3174–3183.
[17] Y. Li, D. Wang, H. Hu, Y. Lin, and Y. Zhuang, “Zero-shot recognition using dual visual-semantic mapping paths,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2017, pp. 3279–3287.
Song Guo (Fellow, IEEE) received the Ph.D. degree in computer science from the University of Ottawa, Ottawa, ON, Canada. He was a Professor with the University of Aizu, Aizuwakamatsu, Japan. He is currently a Full Professor with the Department of Computing, The Hong Kong Polytechnic University, Hong Kong SAR, China. He also holds a Changjiang Chair Professorship awarded by the Ministry of Education of China. He is also the Editor-in-Chief of IEEE Open Journal of the Computer Society and the Chair of IEEE Communications Society Space and Satellite Communications Technical Committee. His research interests mainly include edge AI, Big Data and machine learning, mobile computing, and distributed systems. He was the recipient of the 2019 IEEE TECHNICAL COMMITTEE ON BIG DATA Best Conference Paper Award, 2018 TECHNICAL COMMITTEE ON GREEN COMMUNICATIONS AND COMPUTING Best Magazine Paper Award, 2019 & 2017 IEEE Systems Journal Annual Best Paper Award, and other 8 Best Paper Awards from IEEE/ACM conferences. His work was also recognized by the 2016 Annual Best of Computing, Notable Books and Articles in Computing in ACM Computing Reviews. He was an IEEE ComSoc Distinguished Lecturer, Director of ComSoc Membership Services, and the Member of IEEE ComSoc Board of Governors. He has also worked for IEEE Computer Society on Fellow Evaluation Committee, Transactions Operations Committee, Editor-in-Chief Search Committee. He was named on Editorial Board of a number of prestigious international journals like IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, IEEE TRANSACTIONS ON CLOUD COMPUTING, IEEE TRANSACTIONS ON EMERGING TOPICS IN COMPUTING, IEEE TRANSACTIONS ON SUSTAINABLE COMPUTING, IEEE TRANSACTIONS ON GREEN COMMUNICATIONS AND NETWORKING, IEEE INTERNET OF THINGS JOURNAL. He was also the Chair of Organizing and Technical Committees of numerous international conferences. He is frequently invited expert for various national and international grant reviews. He is currently on RGC engineering panel. Prof. Guo is a fellow of the Canadian Academy of Engineering, and ACM Distinguished Member.

Jingcai Guo (Member, IEEE) received the B.Eng. degree in computer science from Sichuan University, Chengdu, China, in 2013, the M.Eng. degree in computer science from Waseda University, Tokyo, Japan, in 2015, and the Ph.D. degree from The Hong Kong Polytechnic University, Hong Kong SAR, China, in 2021. He is currently a Research Assistant Professor with the Department of Computing, The Hong Kong Polytechnic University, Hong Kong SAR. From 2021 to 2022, he was an Associate Professor with School of Computing and Artificial Intelligence, Southwestern University of Finance and Economics, China, and from 2019 to 2020, he was a Visiting Researcher with School of Computer Science, The University of Sydney, Camperdown, NSW, Australia. He research interests include machine learning and edge AI, with a particular focus on zero/few-shot learning, federated learning, representation learning, and model compression. He has regularly worked as the PC member and/or reviewer for various venues such as ACM-MM, AAAI, CVPR, ICMR, NeurIPS, ICME, ECCV, TMM, TNNLS, TKDE, IoTJ. He was the recipient of the RGC-HKPFs in 2017.

Yuanyuan Xu received the B.E. degree from Electronic Information Science and Technology, Jiangsu University of Science and Technology, Zhenjiang, China, in 2017. She is currently working toward the master’s degree in electronic information engineering with the University of Electronic Science and Technology of China, Chengdu, China. Her research interests include machine learning and computer vision.

Fushuo Huo received the M.S. degree in electrical engineering with Chongqing University, Chongqing, China. He is currently working toward the Ph.D. degree with the Department of Computing, The Hong Kong Polytechnic University, Hong Kong. His research interests include salient object detection and multi-modal learning.

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.