Multi-Label Image Recognition with Graph Convolutional Networks*

Zhao-Min Chen[1,2] Xiu-Shen Wei[2] Peng Wang[3] Yanwen Guo[1]
[1]National Key Laboratory for Novel Software Technology, Nanjing University, China
[2]Megvii Research Nanjing, Megvii Technology, China
[3]School of Computer Science, The University of Adelaide, Australia
{chenzhaomin123, weixs.gm}@gmail.com, peng.wang@adelaide.edu.au, ywguo@nju.edu.cn

Abstract

The task of multi-label image recognition is to predict a set of object labels that present in an image. As objects normally co-occur in an image, it is desirable to model the label dependencies to improve the recognition performance. To capture and explore such important dependencies, we propose a multi-label classification model based on Graph Convolutional Network (GCN). The model builds a directed graph over the object labels, where each node (label) is represented by word embeddings of a label, and GCN is learned to map this label graph into a set of inter-dependent object classifiers. These classifiers are applied to the image descriptors extracted by another sub-net, enabling the whole network to be end-to-end trainable. Furthermore, we propose a novel re-weighted scheme to create an effective label correlation matrix to guide information propagation among the nodes in GCN. Experiments on two multi-label image recognition datasets show that our approach obviously outperforms other existing state-of-the-art methods. In addition, visualization analyses reveal that the classifiers learned by our model maintain meaningful semantic topology.

1. Introduction

Multi-label image recognition is a fundamental and practical task in Computer Vision, where the aim is to predict a set of objects present in an image. It can be applied to many fields such as medical diagnosis recognition [7], human attribute recognition [19] and retail checkout recognition [8, 30]. Comparing to multi-class image classification [21], the multi-label task is more challenging due to the combinatorial nature of the output space. As the objects normally co-occur in the physical world, a key for multi-label image recognition is to model the label dependencies, as shown in Fig. 1.

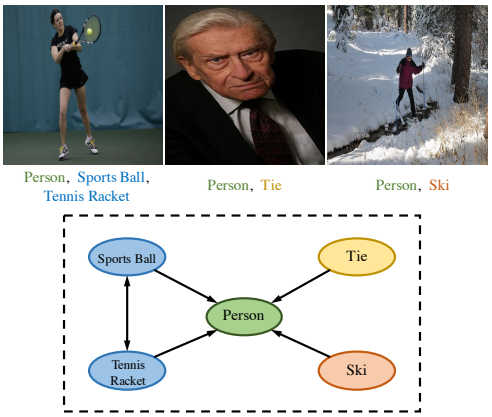


Figure 1. We build a directed graph over the object labels to model label dependencies in multi-label image recognition. In this figure, "$\text{Label}_\text{A} \rightarrow \text{Label}_\text{B}$", means when Label_A appears, Label_B is likely to appear, but the reverse may not be true.

A Naïve way to address the multi-label recognition problem is to treat the objects in isolation and convert the multi-label problem into a set of binary classification problems to predict whether each object of interest presents or not. Benefited from the great success of single-label image classification achieved by deep Convolutional Neural Networks (CNNs) [10, 26, 27, 12], the performance of the binary solutions has been greatly improved. However, these methods are essentially limited by ignoring the complex topology structure between objects. This stimulates research for approaches to capture and explore the label correlations in various ways. Some approaches, based on probabilistic graph model [18, 17] or Recurrent Neural Networks (RNNs) [28], are proposed to explicitly model label dependencies. While the former formulates the multi-label classification problem as a structural inference problem which may suffer from a scalability issue due to high computational complexity, the

*Z.-M. Chen's contribution was made when he was an intern in Megvii Research Nanjing. X.-S. Wei and Y. Guo are the corresponding authors. Yanwen Guo is also with the Science and Technology on Information Systems Engineering Laboratory, China, and the 28th Research Institute of China Electronics Technology Group Corporation, Nanjing 210007, China. This research was supported by National Key R&D Program of China (No. 2017YFA0700800), the National Natural Science Foundation of China under Grants 61772257 and 61672279.

latter predicts the labels in a sequential fashion, based on some orders either pre-defined or learned. Another line of works implicitly model the label correlations via attention mechanisms [36, 29]. They consider the relations between attended regions of an image, which can be viewed as *local correlations*, but still ignore the *global correlations* between labels which require to be inferred from knowledge beyond a single image.

In this paper, we propose a novel GCN based model (*aka* ML-GCN) to capture the label correlations for multi-label image recognition, which properties with scalability and flexibility impossible for competing approaches. Instead of treating object classifiers as a set of independent parameter vectors to be learned, we propose to learn inter-dependent object classifiers from prior label representations, *e.g.*, word embeddings, via a GCN based mapping function. In the following, the generated classifiers are applied to image representations generated by another sub-net to enable end-to-end training. As the embedding-to-classifier mapping parameters are shared across all classes (*i.e.*, image labels), the gradients from all classifiers impact the GCN based classifier generation function. This implicitly models the label correlations. Furthermore, to explicitly model the label dependencies for classifier learning, we design an effective label correlation matrix to guide the information propagation among nodes in GCN. Specifically, we propose a re-weighted scheme to balance the weights between a node and its neighborhood for node feature update, which effectively alleviates overfitting and over-smoothing. Experiments on two multi-label image recognition datasets show that our approach obviously outperforms existing state-of-the-art methods. In addition, visualization analyses reveal that the classifiers learned by our model maintain meaningful semantic structures.

The main contributions of this paper are as follows:

- We propose a novel end-to-end trainable multi-label image recognition framework, which employs GCN to map label representations, *e.g.*, word embeddings, to inter-dependent object classifiers.
- We conduct in-depth studies on the design of correlation matrix for GCN and propose an effective re-weighted scheme to simultaneously alleviate the over-fitting and over-smoothing problems.
- We evaluate our method on two benchmark multi-label image recognition datasets, and our proposed method consistently achieves superior performance over previous competing approaches.

2. Related Work

The performance of image classification has recently witnessed a rapid progress due to the establishment of large-scale hand-labeled datasets such as ImageNet [4], MS-COCO [20] and PASCAL VOC [5], and the fast development of deep convolutional networks [10, 11, 35, 3, 32]. Many efforts have been dedicated to extending deep convolutional networks for multi-label image recognition.

A straightforward way for multi-label recognition is to train independent binary classifiers for each class/label. However, this method does not consider the relationship among labels, and the number of predicted labels will grow exponentially as the number of categories increase. For instance, if a dataset contains 20 labels, then the number of predicted label combination could be more than 1 million (*i.e.*, 2^{20}). Besides, this baseline method is essentially limited by ignoring the topology structure among objects, which can be an important regularizer for the co-occurrence patterns of objects. For example, some combinations of labels are almost impossible to appear in the physical world.

In order to regularize the prediction space, many researchers attempted to capture label dependencies. Gong *et al.* [9] used a ranking-based learning strategy to train deep convolutional neural networks for multi-label image recognition and found that the weighted approximated-ranking loss worked best. Additionally, Wang *et al.* [28] utilized recurrent neural networks (RNNs) to transform labels into embedded label vectors, so that the correlation between labels can be employed. Furthermore, attention mechanisms were also widely applied to discover the label correlation in the multi-label recognition task. In [36], Zhu *et al.* proposed a spatial regularization network to capture both semantic and spatial relations of these multiple labels based on weighted attention maps. Wang *et al.* [29] introduced a spatial transformer layer and long short-term memory (LSTM) units to capture the label correlation.

Compared with the aforementioned structure learning methods, the *graph* was proven to be more effective in modeling label correlation. Li *et al.* [18] created a tree-structured graph in the label space by using the maximum spanning tree algorithm. Li *et al.* [17] produced image-dependent conditional label structures base on the graphical Lasso framework. Lee *et al.* [15] incorporated knowledge graphs for describing the relationships between multiple labels. In this paper, we leverage the graph structure to capture and explore the label correlation dependency. Specifically, based on the graph, we utilize GCN to propagate information between multiple labels and consequently learn inter-dependent classifiers for each of image labels. These classifiers absorb information from the label graph, which are further applied to the global image representation for the final multi-label prediction. It is a more explicit way for evaluating label co-occurrence. Experimental results validate our proposed approach is effective and our model can be trained in an end-to-end manner.

3. Approach

In this part, we elaborate on our ML-GCN model for multi-label image recognition. Firstly, we introduce the moti-

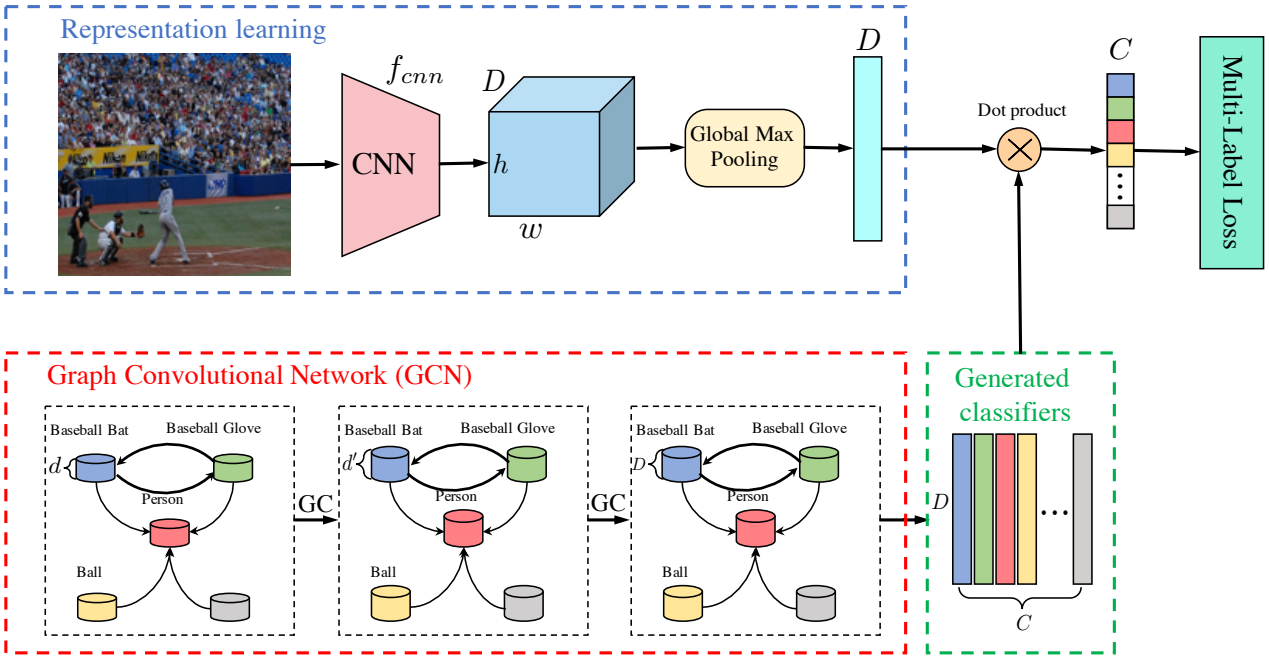


Figure 2. Overall framework of our ML-GCN model for multi-label image recognition. The object labels are represented by word embeddings $\boldsymbol{Z} \in \mathbb{R}^{C\times d}$ (C is the number of categories and d is the dimensionality of word-embedding vector). A directed graph is built over these label representations, where each node denotes a label. Stacked GCNs are learned over the label graph to map these label representations into a set of inter-dependent object classifiers, *i.e.*, $\boldsymbol{W} \in \mathbb{R}^{C\times D}$, which are applied to the image representation extracted from the input image via a convolutional network for multi-label image recognition.

vation for our method. Then, we introduce some preliminary knowledge of GCN, which is followed by the detailed illustration of the proposed ML-GCN model and the re-weighted scheme for correlation matrix construction.

3.1. Motivations

How to effectively capture the correlations between object labels and explore these label correlations to improve the classification performance are both important for multi-label image recognition. In this paper, we use a graph to model the inter dependencies between labels, which is a flexible way to capture the topological structure in the label space. Specifically, we represent each node (label) of the graph as word embeddings of the label, and propose to use GCN to directly map these label embeddings into a set of inter-dependent classifiers, which can be directly applied to an image feature for classification. Two factors motivated the design of our GCN based model. Firstly, as the embedding-to-classifier mapping parameters are shared across all classes, the learned classifiers can retain the weak semantic structures in the word embedding space, where semantic related concepts are close to each other. Meanwhile, the gradients of all classifiers can impact the classifier generation function, which implicitly models the label dependencies. Secondly, we design a novel label correlation matrix based on their co-occurrence patterns to explicitly model the label dependencies by GCN, with which the update of node features will absorb information from correlated nodes (labels).

3.2. Graph Convolutional Network Recap

Graph Convolutional Network (GCN) was introduced in [14] to perform semi-supervised classification. The essential idea is to update the node representations by propagating information between nodes.

Unlike standard convolutions that operate on local Euclidean structures in an image, the goal of GCN is to learn a function $f(\cdot,\cdot)$ on a graph $\mathcal{G}$, which takes feature descriptions $\boldsymbol{H}^l \in \mathbb{R}^{n\times d}$ and the corresponding correlation matrix $\boldsymbol{A} \in \mathbb{R}^{n\times n}$ as inputs (where n denotes the number of nodes and d indicates the dimensionality of node features), and updates the node features as $\boldsymbol{H}^{l+1} \in \mathbb{R}^{n\times d'}$. Every GCN layer can be written as a non-linear function by

$$\boldsymbol{H}^{l+1} = f(\boldsymbol{H}^l, \boldsymbol{A}). \tag{1}$$

After employing the convolutional operation of [14], $f(\cdot,\cdot)$ can be represented as

$$\boldsymbol{H}^{l+1} = h(\widehat{\boldsymbol{A}}\boldsymbol{H}^l\boldsymbol{W}^l), \tag{2}$$

where $\boldsymbol{W}^l \in \mathbb{R}^{d\times d'}$ is a transformation matrix to be learned and $\widehat{\boldsymbol{A}} \in \mathbb{R}^{n\times n}$ is the normalized version of correlation matrix $\boldsymbol{A}$, and $h(\cdot)$ denotes a non-linear operation, which is

acted by LeakyReLU [22] in our experiments. Thus, we can learn and model the complex inter-relationships of the nodes by stacking multiple GCN layers. For more details, we refer interested readers to [14].

3.3. GCN for Multi-label Recognition

Our ML-GCN is built upon GCN. GCN was proposed for semi-supervised classification, where the node-level output is the prediction score of each node. Different from that, we design the final output of each GCN node to be the classifier of the corresponding label in our task. In addition, the graph structure (*i.e.*, the correlation matrix) is normally pre-defined in other tasks, which, however, is not provided in the multi-label image recognition task. Thus, we need to construct the correlation matrix from scratch. The overall framework of our approach is shown in Fig. 2, which is composed of two main modules, *i.e.*, the image representation learning and GCN based classifier learning modules.

Image representation learning We can use any CNN base models to learn the features of an image. In our experiments, following [36, 1, 15, 6], we use ResNet-101 [10] as the base model in experiments. Thus, if an input image $\boldsymbol{I}$ is with the 448×448 resolution, we can obtain $2048 \times 14 \times 14$ feature maps from the "`conv5_x`" layer. Then, we employ global max-pooling to obtain the image-level feature $\boldsymbol{x}$:

$$\boldsymbol{x} = f_{\text{GMP}}(f_{\text{cnn}}(\boldsymbol{I}; \theta_{\text{cnn}})) \in \mathbb{R}^D, \tag{3}$$

where θ_{cnn} indicates model parameters and $D = 2048$.

GCN based classifier learning We learn inter-dependent object classifiers, *i.e.*, $\boldsymbol{W} = \{\boldsymbol{w}_i\}_{i=1}^C$, from label representations via a GCN based mapping function, where C denotes the number of categories. We use stacked GCNs where each GCN layer l takes the node representations from previous layer ($\boldsymbol{H}^l$) as inputs and outputs new node representations, *i.e.*, $\boldsymbol{H}^{l+1}$. For the first layer, the input is the $\boldsymbol{Z} \in \mathbb{R}^{C \times d}$ matrix, where d is the dimensionality of the label-level word embedding. For the last layer, the output is $\boldsymbol{W} \in \mathbb{R}^{C \times D}$ with D denoting the dimensionality of the image representation. By applying the learned classifiers to image representations, we can obtain the predicted scores as

$$\hat{\boldsymbol{y}} = \boldsymbol{W}\boldsymbol{x}. \tag{4}$$

We assume that the ground truth label of an image is $\boldsymbol{y} \in \mathbb{R}^C$, where $y^i = \{0, 1\}$ denotes whether label i appears in the image or not. The whole network is trained using the traditional multi-label classification loss as follows

$$\mathcal{L} = \sum_{c=1}^{C} y^c \log(\sigma(\hat{y}^c)) + (1 - y^c) \log(1 - \sigma(\hat{y}^c)), \tag{5}$$

where $\sigma(\cdot)$ is the sigmoid function.

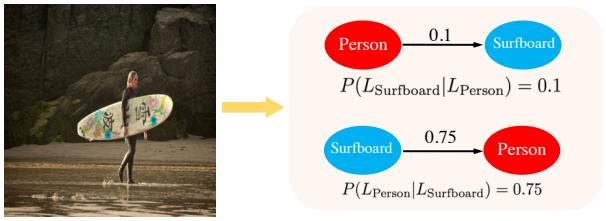


Figure 3. Illustration of conditional probability between two labels. As usual, when "`surfboard`" appears in the image, "`person`" will also occur with a high probability. However, in the condition of "`person`" appearing, "`surfboard`" will not necessarily occur.

3.4. Correlation Matrix of ML-GCN

GCN works by propagating information between nodes based on the correlation matrix. Thus, how to build the correlation matrix $\boldsymbol{A}$ is a crucial problem for GCN. In most applications, the correlation matrix is pre-defined, which, however, is not provided in any standard multi-label image recognition datasets. In this paper, we build this correlation matrix through a data-driven way. That is, we define the correlation between labels via mining their co-occurrence patterns within the dataset.

We model the label correlation dependency in the form of conditional probability, *i.e.*, $P(L_j|L_i)$ which denotes the probability of occurrence of label L_j when label L_i appears. As shown in Fig. 3, $P(L_j|L_i)$ is not equal to $P(L_i|L_j)$. Thus, the correlation matrix is asymmetrical.

To construct the correlation matrix, firstly, we count the occurrence of label pairs in the training set and get the matrix $\boldsymbol{M} \in \mathbb{R}^{C \times C}$. Concretely, C is the number of categories, and M_{ij} denotes the concurring times of L_i and L_j. Then, by using this label co-occurrence matrix, we can get the conditional probability matrix by

$$\boldsymbol{P}_i = \boldsymbol{M}_i / N_i, \tag{6}$$

where N_i denotes the occurrence times of L_i in the training set, and $P_{ij} = P(L_j|L_i)$ means the probability of label L_j when label L_i appears.

However, the simple correlation above may suffer from two drawbacks. Firstly, the co-occurrence patterns between a label and the other labels may exhibit a long-tail distribution, where some rare co-occurrences may be noise. Secondly, the absolute number of co-occurrences from training and test may not be completely consistent. A correlation matrix over-fitted to the training set can hurt the generalization capacity. Thus, we propose to binarize the correlation $\boldsymbol{P}$. Specifically, we use the threshold τ to filter noisy edges, and the operation can be written as

$$\boldsymbol{A}_{ij} = \begin{cases} 0, & \text{if } \boldsymbol{P}_{ij} < \tau \\ 1, & \text{if } \boldsymbol{P}_{ij} \geq \tau \end{cases}, \tag{7}$$

where $\boldsymbol{A}$ is the binary correlation matrix.

Over-smoothing problem From Eq. (2), we can conclude that after GCN, the feature of a node will be the weighted sum of its own feature and the adjacent nodes' features. Then, a direct problem for the binary correlation matrix is that it can result in over-smoothing. That is, the node features may be over-smoothed such that nodes from different clusters (*e.g.*, kitchen related *vs*. living room related) may become indistinguishable [16]. To alleviate this problem, we propose the following re-weighted scheme,

$$\boldsymbol{A}'_{ij} = \begin{cases} p / \sum_{\substack{j=1 \\ i \neq j}}^{C} \boldsymbol{A}_{ij}, & \text{if } i \neq j \\ 1 - p, & \text{if } i = j \end{cases}, \tag{8}$$

where $\boldsymbol{A}'$ is the re-weighted correlation matrix, and p determines the weights assigned to a node itself and other correlated nodes. By doing this, when updating the node feature, we will have a fixed weight for the node itself and the weights for correlated nodes will be determined by the neighborhood distribution. When $p \to 1$, the feature of a node itself will not be considered. While, on the other hand, when $p \to 0$, neighboring information tends to be ignored.

4. Experiments

In this section, we first describe the evaluation metrics and implementation details. Then, we report the empirical results on two benchmark multi-label image recognition datasets, *i.e.*, MS-COCO [20] and VOC 2007 [5]. Finally, visualization analyses are presented.

4.1. Evaluation Metrics

Following conventional settings [28, 6, 36], we report the average per-class precision (CP), recall (CR), F1 (CF1) and the average overall precision (OP), recall (OR), F1 (OF1) for performance evaluation. For each image, the labels are predicted as positive if the confidences of them are greater than 0.5. For fair comparisons, we also report the results of top-3 labels, cf. [36, 6]. In addition, we also compute and report the mean average precision (mAP). Generally, average overall F1 (**OF1**), average per-class F1 (**CF1**) and **mAP** are relatively more important for performance evaluation.

4.2. Implementation Details

Without otherwise stated, our ML-GCN consists of two GCN layers with output dimensionality of 1024 and 2048, respectively. For label representations, we adopt 300-dim GloVe [25] trained on the Wikipedia dataset. For the categories whose names contain multiple words, we obtain the label representation as average of embeddings for all words. For the correlation matrix, without otherwise stated, we set τ in Eq. (7) to be 0.4 and p in Eq. (8) to be 0.2. In the image representation learning branch, we adopt LeakyReLU [22] with the negative slope of 0.2 as the non-linear activation function, which leads to faster convergence in experiments. We adopt ResNet-101 [10] as the feature extraction backbone, which is pre-trained on ImageNet [4]. During training, the input images are random cropped and resized into 448×448 with random horizontal flips for data augmentation. For network optimization, SGD is used as the optimizer. The momentum is set to be 0.9. Weight decay is 10^{-4}. The initial learning rate is 0.01, which decays by a factor of 10 for every 40 epochs and the network is trained for 100 epochs in total. We implement the network based on PyTorch.

4.3. Experimental Results

In this part, we first present our comparisons with state-of-the-arts on MS-COCO and VOC 2007, respectively. Then, we conduct ablation studies to evaluate the key aspects of the proposed approach.

4.3.1 Comparisons with State-of-the-Arts

Results on MS-COCO Microsoft COCO [20] is a widely used benchmark for multi-label image recognition. It contains 82,081 images as the training set and 40,504 images as the validation set. The objects are categorized into 80 classes with about 2.9 object labels per image. Since the ground-truth labels of the test set are not available, we evaluate the performance of all the methods on the validation set. The number of labels of different images also varies considerably, which makes MS-COCO more challenging.

Quantitative results are reported in Table 1. We compare with state-of-the-art methods, including CNN-RNN [28], RNN-Attention [29], Order-Free RNN [1], ML-ZSL [15], SRN [36], Multi-Evidence [6], etc. For the proposed ML-GCN, we report the results based on the binary correlation matrix ("ML-GCN (Binary)") and the re-weighted correlation matrix ("ML-GCN (Re-weighted)"), respectively. It is obvious to see that our ML-GCN method based on the binary correlation matrix obtains worse classification performance, which may be largely due to the over-smoothing problem discussed in Sec. 3.4. The proposed re-weighted scheme can alleviate the over-smoothing issue and consequently obtains superior performance. Comparing with state-of-the-art methods, our approach with the proposed re-weighted scheme consistently performs better under almost all metrics, which shows the effectiveness of our proposed ML-GCN as well as its corresponding re-weighted scheme.

Results on VOC 2007 PASCAL Visual Object Classes Challenge (VOC 2007) [5] is another popular dataset for multi-label recognition. It contains 9,963 images from 20 object categories, which is divided into train, val and test sets. Following [2, 29], we use the *trainval* set to train our model, and evaluate the recognition performance on the test set. In order to compare with other state-of-the-art methods,

Table 1. Comparisons with state-of-the-art methods on the MS-COCO dataset. The performance of the proposed ML-GCN based on two types of correlation matrices are reported."Binary" denotes that we use the binary correlation matrix, cf. Eq. (7). "Re-weighted" means the correlation matrix generated by the proposed re-weighted scheme is used, cf. Eq. (8).

Methods	All							Top-3					
	mAP	CP	CR	CF1	OP	OR	OF1	CP	CR	CF1	OP	OR	OF1
CNN-RNN [28]	61.2	–	–	–	–	–	–	66.0	55.6	60.4	69.2	66.4	67.8
RNN-Attention [29]	–	–	–	–	–	–	–	79.1	58.7	67.4	84.0	63.0	72.0
Order-Free RNN [1]	–	–	–	–	–	–	–	71.6	54.8	62.1	74.2	62.2	67.7
ML-ZSL [15]	–	–	–	–	–	–	–	74.1	**64.5**	69.0	–	–	–
SRN [36]	77.1	81.6	65.4	71.2	82.7	69.9	75.8	85.2	58.8	67.4	87.4	62.5	72.9
ResNet-101 [10]	77.3	80.2	66.7	72.8	83.9	70.8	76.8	84.1	59.4	69.7	89.1	62.8	73.6
Multi-Evidence [6]	–	80.4	70.2	74.9	85.2	72.5	78.4	84.5	62.2	70.6	89.1	64.3	74.7
ML-GCN (Binary)	80.3	81.1	70.1	75.2	83.8	74.2	78.7	84.9	61.3	71.2	88.8	65.2	75.2
ML-GCN (Re-weighted)	**83.0**	**85.1**	**72.0**	**78.0**	**85.8**	**75.4**	**80.3**	**89.2**	64.1	**74.6**	**90.5**	**66.5**	**76.7**

Table 2. Comparisons of AP and mAP with state-of-the-art methods on the VOC 2007 dataset. The meanings of "Binary" and "Re-weighted" are the same as Table 1.

Methods	*aero*	*bike*	*bird*	*boat*	*bottle*	*bus*	*car*	*cat*	*chair*	*cow*	*table*	*dog*	*horse*	*motor*	*person*	*plant*	*sheep*	*sofa*	*train*	*tv*	mAP
CNN-RNN [28]	96.7	83.1	94.2	92.8	61.2	82.1	89.1	94.2	64.2	83.6	70.0	92.4	91.7	84.2	93.7	59.8	93.2	75.3	**99.7**	78.6	84.0
RLSD [34]	96.4	92.7	93.8	94.1	71.2	92.5	94.2	95.7	74.3	90.0	74.2	95.4	96.2	92.1	97.9	66.9	93.5	73.7	97.5	87.6	88.5
VeryDeep [26]	98.9	95.0	96.8	95.4	69.7	90.4	93.5	96.0	74.2	86.6	**87.8**	96.0	96.3	93.1	97.2	70.0	92.1	80.3	98.1	87.0	89.7
ResNet-101 [10]	99.5	97.7	97.8	96.4	65.7	91.8	96.1	97.6	74.2	80.9	85.0	**98.4**	96.5	95.9	98.4	70.1	88.3	80.2	98.9	89.2	89.9
FeV+LV [33]	97.9	97.0	96.6	94.6	73.6	93.9	96.5	95.5	73.7	90.3	82.8	95.4	97.7	95.9	98.6	77.6	88.7	78.0	98.3	89.0	90.6
HCP [31]	98.6	97.1	98.0	95.6	75.3	**94.7**	95.8	97.3	73.1	90.2	80.0	97.3	96.1	94.9	96.3	78.3	94.7	76.2	97.9	91.5	90.9
RNN-Attention [29]	98.6	97.4	96.3	96.2	75.2	92.4	96.5	97.1	76.5	92.0	87.7	96.8	97.5	93.8	98.5	81.6	93.7	82.8	98.6	89.3	91.9
Atten-Reinforce [2]	98.6	97.1	97.1	95.5	75.6	92.8	96.8	97.3	78.3	92.2	87.6	96.9	96.5	93.6	98.5	81.6	93.1	83.2	98.5	89.3	92.0
VGG (Binary)	98.3	97.1	96.1	96.7	75.0	91.4	95.8	95.4	76.7	92.1	85.1	96.7	96.0	95.3	97.8	77.4	93.1	79.7	97.9	89.3	91.1
VGG (Re-weighted)	99.4	97.4	98.0	97.0	77.9	92.4	96.8	97.8	80.8	93.4	87.2	98.0	97.3	95.8	98.8	79.4	95.3	82.2	99.1	91.4	92.8
ML-GCN (Binary)	99.6	98.3	97.9	97.6	78.2	92.3	97.4	97.4	79.2	94.4	86.5	97.4	97.9	97.1	98.7	84.6	95.3	83.0	98.6	90.4	93.1
ML-GCN (Re-weighted)	99.5	**98.5**	**98.6**	**98.1**	80.8	94.6	**97.2**	98.2	**82.3**	**95.7**	86.4	98.2	**98.4**	**96.7**	**99.0**	**84.7**	**96.7**	**84.3**	98.9	**93.7**	**94.0**

we report the results of average precision (AP) and mean average precision (mAP).

The results of VOC 2007 are presented in Table 2. Because the results of many previous works on VOC 2007 are based on the VGG model [26]. For fair comparisons, we also report the results using VGG models as the base model. It is apparent to see that, our proposed method observes improvements upon the previous methods. Concretely, the proposed ML-GCN with our re-weighted scheme obtains 94.0% mAP, which outperforms state-of-the-art by 2%. Even using VGG model as the base model, we can still achieve better results ($+0.8\%$). Also, consistent with the results on MS-COCO, the re-weighed scheme enjoys better performance than the binary correlation matrix on VOC as well.

4.3.2 Ablation Studies

In this section, we perform ablation studies from four different aspects, including the sensitivity of ML-GCN to different types of word embeddings, effects of τ in correlation matrix binarization, effects of p for correlation matrix re-weighting, and the depths of GCN.

ML-GCN under different types of word embeddings By default, we use Glove [25] as label representations, which serves as the inputs of the stacked GCNs for learning the object classifiers. In this part, we evaluate the performance of ML-GCN under other types popular word representations. Specifically, we investigate four different word embedding methods, including GloVe [25], GoogleNews [24], FastText [13] and the simple one-hot word embedding. Fig. 4 shows the results using different word embeddings on MS-COCO and VOC 2007. As shown, we can see that when using different word embeddings as GCN's inputs, the multi-label recognition accuracy will not be affected significantly. In addition, the observations (especially the results of one-hot) justify that the accuracy improvements achieved by our method do not absolutely come from the semantic meanings derived from word embeddings. Furthermore, using powerful word embeddings could lead to better performance. One possible reason may be that the word embeddings [25, 24, 13] learned from large text corpus maintain some semantic topology. That is, for semantic related concepts, their embeddings are close in the embedding space. Our model can employ these implicit dependencies, and further benefit multi-label image recognition.

Effects of different threshold values τ We vary the values of the threshold τ in Eq. (7) for correlation matrix binarization, and show the results in Fig. 5. Note that, if we do

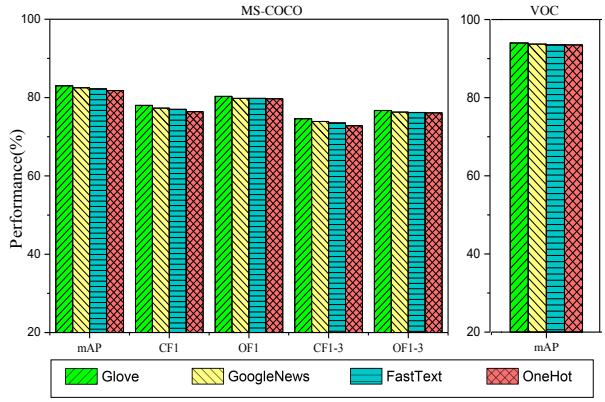
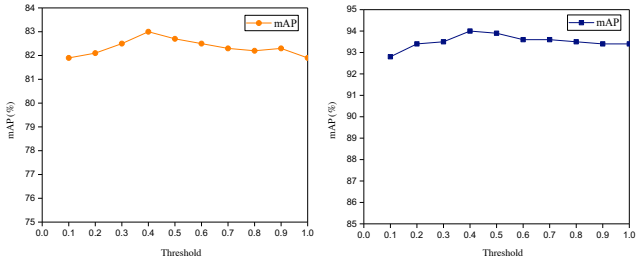


Figure 4. Effects of different word embedding approaches. It is clear to see that, different word embeddings will hardly affect the accuracy, which reveals our improvements do not absolutely come from the semantic meanings derived from word embeddings, rather than our ML-GCN.



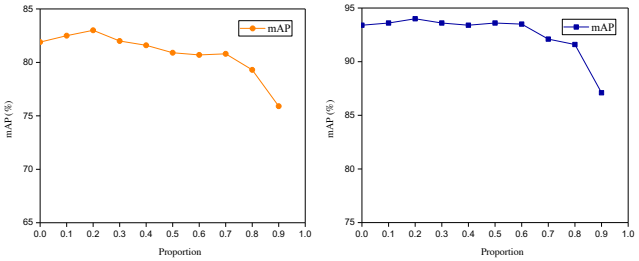
(a) Comparisons on MS-COCO. (b) Comparisons on VOC 2007.

Figure 5. Accuracy comparisons with different values of τ.

not filter any edges, the model will not converge. Thus, there is no result for $\tau = 0$ in that figure. As shown, when filtering out the edges of small probabilities (*i.e.*, noisy edges), the multi-label recognition accuracy is boosted. However, when too many edges are filtered out, the accuracy drops since correlated neighbors will be ignored as well. The optimal value of τ is 0.4 for both MS-COCO and VOC 2007.

Effects of different p for correlation matrix re-weighting To explore the effects of different values of p in Eq. (8) on multi-label classification accuracy, we change the values of p in a set of $\{0, 0.1, 0.2, \ldots, 0.9, 1\}$, as depicted in Fig. 6. Generally, this figure shows the importance of balancing the weights between a node itself and the neighborhood when updating the node feature in GCN. In experiments, we choose the optimal value of p by cross-validations. We can see that when $p = 0.2$, it can achieve the best performance on both MS-COCO and VOC 2007. If p is too small, nodes (labels) of the graph can not get sufficient information from correlated nodes (labels). While, if p is too large, it will lead to over-smoothing.

Another interesting observation is that, when $p = 0$, we



(a) Comparisons on MS-COCO. (b) Comparisons on VOC 2007.

Figure 6. Accuracy comparisons with different values of p. Note that, when $p = 1$, the model does not converge.

Table 3. Comparisons with different depths of GCN in our model.

♯ Layer	MS-COCO					VOC
	All			Top-3		All
	mAP	CF1	OF1	CF1	OF1	mAP
2-layer	**83.0**	**78.0**	**80.3**	**74.6**	**76.7**	**94.0**
3-layer	82.1	76.9	79.7	73.7	76.2	93.6
4-layer	81.1	76.4	79.4	72.5	75.8	93.0

can obtain mAPs of 81.67% on MS-COCO and 93.15% on VOC 2007, which still outperforms existing methods. Note that when $p = 0$, we essentially do not explicitly incorporate the label correlations. The improvement is benefited from that our ML-GCN model learns the object classifiers from the prior label representations through a shared GCN based mapping function, which implicitly models label dependencies as discussed in Sec. 3.1

The deeper, the better? We show the performance results with different numbers of GCN layers for our model in Table 3. For the three-layer model, the output dimensionalities are 1024, 1024 and 2048 for the sequential layers, respectively. For the four-layer model, the dimensionalities are 1024, 1024, 1024 and 2048. As shown, when the number of graph convolution layers increases, multi-label recognition performance drops on both datasets. The possible reason for the performance drop may be that when using more GCN layers, the propagation between nodes will be accumulated, which can result in over-smoothing.

4.4. Classifier Visualization

The effectiveness of our approach has been quantitatively evaluated through comparisons to existing methods and detailed ablation studies. In this section, we visualize the learned inter-dependent classifiers to show if meaningful semantic topology can be maintained.

In Fig. 8, we adopt the t-SNE [23] to visualize the classifiers learned by our proposed ML-GCN, as well the classifiers learned through vanilla ResNet (*i.e.*, parameters of the last fully-connected layer). It is clear to see that, the classifiers learned by our method maintain meaningful semantic

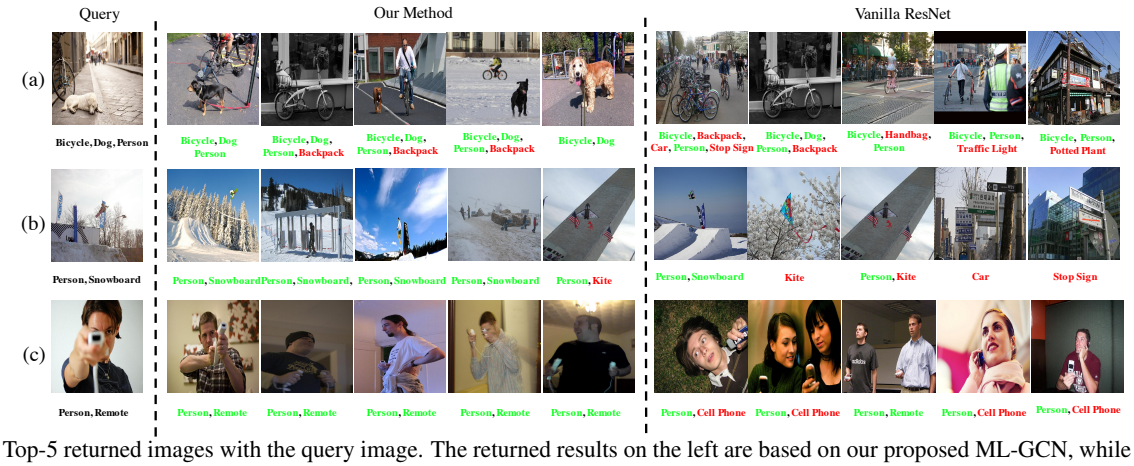


Figure 7. Top-5 returned images with the query image. The returned results on the left are based on our proposed ML-GCN, while the results on the right are vanilla ResNet. All results are sorted in the ascending order according to the distance from the query image.

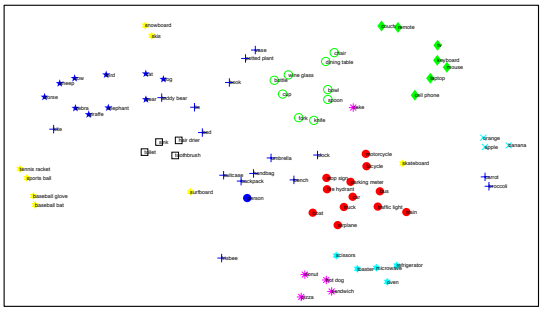
topology. Specifically, the learned classifiers exhibit cluster patterns. Classifiers (of "`car`" and "`truck`") within one super concept ("`transportation`"), tend to be close in the classifier space. This is consistent with common sense, which indicates that the classifiers learned by our approach may not be limited to the dataset where the classifiers are learned, but may enjoy generalization capacities. On the contrary, the classifiers learned through vanilla ResNet uniformly distribute in the space and do not shown any meaningful topology. This visualization further shows the effectiveness of our approach in modeling label dependencies.

4.5. Performance on image retrieval

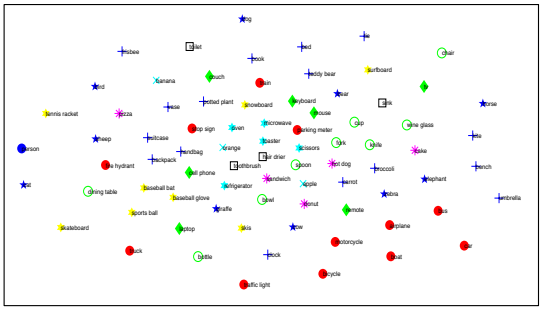
Apart from analyzing the learned classifiers, we further evaluate if our model can learn better image representations. We conduct an image retrieval experiment to verify this. Specifically, we use the k-NN algorithm to perform content-based image retrieval to validate the discriminative ability of image representations learned by our model. Still, we choose the features from vanilla ResNet as the baseline. We show the top-5 images returned by k-NN. The retrieval results are presented in Fig. 7. For each query image, the corresponding returned images are sorted in the ascending order according to the distance to the query image. We can clearly observe that our retrieval results are obviously better than the vanilla ResNet baseline. For example, in Fig. 7 (c), the labels of the images returned by our approach almost exactly match the labels of the query image. It can demonstrate that our ML-GCN can not only effectively capture label dependencies to learn better classifiers, but can benefit image representation learning as well in multi-label recognition.

5. Conclusion

Capturing label dependencies is one crucial issue for multi-label image recognition. In order to model and explore this important information, we proposed a GCN based model



(a) t-SNE on the learned inter-dependent classifiers by our model.



(b) t-SNE on the classifiers by the vanilla ResNet.

Figure 8. Visualization of the learned inter-dependent classifiers by our model and vanillia classifiers of ResNet on MS-COCO.

to learn inter-dependent object classifiers from prior label representations, *e.g.*, word embeddings. To explicitly model the label dependencies, we designed a novel re-weighted scheme to construct the correlation matrix for GCN by balancing the weights between a node and its neighborhood for node feature update. This scheme can effectively alleviate over-fitting and over-smoothing, which are two key factors hampering the performance of GCN. Both quantitative and qualitative results validated the advantages of our ML-GCN.

References

[1] Shang-Fu Chen, Yi-Chen Chen, Chih-Kuan Yeh, and Yu-Chiang Frank Wang. Order-Free RNN with visual attention for multi-label classification. In *AAAI*, pages 6714–6721, 2018. 4, 5, 6

[2] Tianshui Chen, Zhouxia Wang, Guanbin Li, and Liang Lin. Recurrent attentional reinforcement learning for multi-label image recognition. In *AAAI*, pages 6730–6737, 2018. 5, 6

[3] François Chollet. Xception: Deep learning with depthwise separable convolutions. In *CVPR*, pages 1251–1258, 2017. 2

[4] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. ImageNet: A large-scale hierarchical image database. In *CVPR*, pages 248–255, 2009. 2, 5

[5] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (VOC) challenge. *IJCV*, 88(2):303–338, 2010. 2, 5

[6] Weifeng Ge, Sibei Yang, and Yizhou Yu. Multi-evidence filtering and fusion for multi-label classification, object detection and semantic segmentation based on weakly supervised learning. In *CVPR*, pages 1277–1286, 2018. 4, 5, 6

[7] Zongyuan Ge, Dwarikanath Mahapatra, Suman Sedai, and Rajib Chakravorty. Chest X-rays classification: A multi-label and fine-grained problem. *arXiv preprint arXiv:1807.07247*, 2018. 1

[8] Marian George and Christian Floerkemeier. Recognizing products: A per-exemplar multi-label image classification approach. In *ECCV*, pages 440–455, 2014. 1

[9] Yunchao Gong, Yangqing Jia, Thomas Leung, Alexander Toshev, and Sergey Ioffe. Deep convolutional ranking for multi-label image annotation. *arXiv preprint arXiv:1312.4894*, 2013. 2

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 1, 2, 4, 5, 6

[11] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *CVPR*, pages 7132–7141, 2018. 2

[12] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *CVPR*, pages 4700–4708, 2017. 1

[13] Armand Joulin, Edouard Grave, Piotr Bojanowski, Matthijs Douze, Hérve Jégou, and Tomas Mikolov. Fasttext.zip: Compressing text classification models. *arXiv preprint arXiv:1612.03651*, 2016. 6

[14] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *ICLR*, pages 1–10, 2017. 3

[15] Chung-Wei Lee, Wei Fang, Chih-Kuan Yeh, and Yu-Chiang Frank Wang. Multi-label zero-shot learning with structured knowledge graphs. In *CVPR*, pages 1576–1585, 2018. 2, 4, 5, 6

[16] Qimai Li, Zhichao Han, and Xiao-Ming Wu. Deeper insights into graph convolutional networks for semi-supervised learning. In *AAAI*, pages 3538–3545, 2018. 5

[17] Qiang Li, Maoying Qiao, Wei Bian, and Dacheng Tao. Conditional graphical lasso for multi-label image classification. In *CVPR*, pages 2977–2986, 2016. 1, 2

[18] Xin Li, Feipeng Zhao, and Yuhong Guo. Multi-label image classification with a probabilistic label enhancement model. In *UAI*, pages 1–10, 2014. 1, 2

[19] Yining Li, Chen Huang, Chen Change Loy, and Xiaoou Tang. Human attribute recognition by deep hierarchical contexts. In *ECCV*, pages 684–700, 2016. 1

[20] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO: Common objects in context. In *ECCV*, pages 740–755, 2014. 2, 5

[21] L. Liu, P. Wang, C. Shen, L. Wang, A. v. d. Hengel, C. Wang, and H. T. Shen. Compositional model based fisher vector coding for image classification. *IEEE TPAMI*, 39:2335–2348, 2017. 1

[22] Andrew L Maas, Awni Y Hannun, and Andrew Y Ng. Rectifier nonlinearities improve neural network acoustic models. In *ICML*, pages 1–6, 2013. 3, 5

[23] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *JMLR*, 9(11):2579–2605, 2008. 7

[24] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. In *ICLR*, pages 1–12, 2013. 6

[25] Jeffrey Pennington, Richard Socher, and Christopher Manning. GloVe: Global vectors for word representation. In *EMNLP*, pages 1532–1543, 2014. 5, 6

[26] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, pages 1–8, 2015. 1, 6

[27] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *CVPR*, pages 2818–2826, 2016. 1

[28] Jiang Wang, Yi Yang, Junhua Mao, Zhiheng Huang, Chang Huang, and Wei Xu. CNN-RNN: A unified framework for multi-label image classification. In *CVPR*, pages 2285–2294, 2016. 1, 2, 5, 6

[29] Zhouxia Wang, Tianshui Chen, Guanbin Li, Ruijia Xu, and Liang Lin. Multi-label image recognition by recurrently discovering attentional regions. In *ICCV*, pages 464–472, 2017. 2, 5, 6

[30] Xiu-Shen Wei, Quan Cui, Lei Yang, Peng Wang, and Lingqiao Liu. RPC: A large-scale retail product checkout dataset. *arXiv preprint arXiv:1901.07249*, 2019. 1

[31] Yunchao Wei, Wei Xia, Min Lin, Junshi Huang, Bingbing Ni, Jian Dong, Yao Zhao, and Shuicheng Yan. HCP: A flexible cnn framework for multi-label image classification. *IEEE TPAMI*, 38(9):1901–1907, 2016. 6

[32] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *CVPR*, pages 5987–5995, 2017. 2

[33] Hao Yang, Joey Tianyi Zhou, Yu Zhang, Bin-Bin Gao, Jianxin Wu, and Jianfei Cai. Exploit bounding box annotations for multi-label object recognition. In *CVPR*, pages 280–288, 2016. 6

[34] Junjie Zhang, Qi Wu, Chunhua Shen, Jian Zhang, and Jianfeng Lu. Multi-label image classification with regional latent semantic dependencies. *IEEE TMM*, 20(10):2801–2813, 2018. 6

[35] Xiangyu Zhang, Xinyu Zhou, Mengxiao Li, and Jian Sun. ShuffleNet: an extremely efficient convolutional neural network for mobile devices. In *CVPR*, pages 6848–6856, 2018. 2

[36] Feng Zhu, Hongsheng Li, Wanli Ouyang, Nenghai Yu, and Xiaogang Wang. Learning spatial regularization with image-level supervisions for multi-label image classification. In *CVPR*, pages 5513–5522, 2017. 2, 4, 5, 6