Joint Input and Output Space Learning for Multi-Label Image Classification

Jiahao Xu, Hongda Tian, Zhiyong Wang, *Member, IEEE*, Yang Wang, *Senior Member, IEEE*, Wenxiong Kang, *Member, IEEE*, and Fang Chen

***Abstract*—Multi-label image classification aims to predict the labels associated with a given image. While most existing methods utilize unified image representations, extracting label-specific features through input space learning would improve the discriminative power of the learned features. On the other hand, most feature learning studies often ignore the learning in the output label space, although taking advantage of label correlations can boost the classification performance. In this paper, we propose a deep learning framework that incorporates flexible modules which can learn from both input and output spaces for multi-label image classification. For the input space learning, we devise a label-specific feature pooling method to refine convolutional features for obtaining features specific to each label. For the output space learning, we design a Two-Stream Graph Convolutional Network (TSGCN) to learn multi-label classifiers by mapping spatial object relationships and semantic label correlations. More specifically, we build object spatial graphs to characterize the spatial relationships among objects in an image, which supplements the label semantic graphs modelling the semantic label correlations. Experimental results on two popular benchmark datasets (i.e., Pascal VOC and MS-COCO) show that our proposed method achieves superior performance over the state-of-the-arts.**

***Index Terms*—Multi-label image classification, label-specific feature, label correlations, graph convolutional network, deep learning.**

I. Introduction

MULTI-LABEL image classification deals with images associated with multiple labels which are generally correlated with each other. The variety of label combinations makes it a challenging task, as the output space would grow exponentially as the number of class labels increases. In recent years, multi-label image classification has been intensively investigated and applied to various scenarios such as automatic image annotation [1] and emotion recognition [2].

Despite its advances, there have been two long-standing challenges that limit the performance of multi-label image classification. First, most existing studies simply utilize the same set of visual features as unified image representations for all class labels. However, each class label only correlates with certain visual features rather than the entire feature set. Moreover, shared visual features tend to be high-dimensional as they aim to cover as many labels or concepts as possible. Therefore, unified representations for all labels are not suitable for multi-label image classification. Existing feature selection studies mainly focus on dimension reduction [3]–[5], while some studies [6]–[9] explore extracting features specific to each label, which is necessary and promising due to concurrent occurrences of different objects or concepts in images. Recently, Convolutional Neural Networks (ConvNets) have achieved great success in various image classification tasks [10]–[12]. However, it has been rarely investigated to extract label-specific features by leveraging the strong representation power of Deep Neural Networks (DNNs).

Second, it is still an open problem to conduct effective output space learning which needs to address two issues: 1) the exponentially growing size of the output label space due to the increased number of labels; and 2) the complexity of label correlations. Early multi-label learning approaches handle each label independently and generally ignore the correlations among labels [13], [14]. In some studies, label correlations were taken into consideration to adapt conventional single-label classification algorithms for multi-label learning, such as ML-KNN [15], ML-DT [16] and RankSVM [17]. Recently, studies have been carried out to model the complex label correlations with the help of DNNs, such as probabilistic graph model [18], Recurrent Neural Networks (RNNs) [19], and attention mechanisms [20], [21], to implicitly model label correlations.

Overall, almost all the existing methods have only addressed one of the two challenges without taking the other issue into account. That is, most label-specific feature learning methods have rarely explored the correlation among labels in the output space, whereas the studies focusing on label correlations often utilize unified features for all class labels. In this paper, we propose a novel deep learning framework that enables concurrent image label-specific features learning and label correlations modelling for the first time. Within the framework, we propose a new pooling strategy, named label-specific feature pooling(LSP), for input space learning. LSP can generate label-specific features by imposing a constraint of label co-occurrence probability on the convolutional features, which are extracted from a backbone

Manuscript received October 23, 2019; revised March 4, 2020; accepted May 26, 2020. Date of publication June 12, 2020; date of current version May 26, 2021. This work was supported by Data61-CISRO Postgraduate Scholarship. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Engin Erzin. *(Corresponding author: Jiahao Xu.)*

Jiahao Xu and Zhiyong Wang are with the School of Computer Science, The University of Sydney, Sydney NSW 2006, Australia (e-mail: jiahao.xu@sydney.edu.au; zhiyong.wang@sydney.edu.au).

Hongda Tian, Yang Wang, and Fang Chen are with the University of Technology Sydney, Sydney NSW 2007, Australia (e-mail: hongda.tian@uts.edu.au; yang.wang@uts.edu.au; fang.chen@uts.edu.au).

Wenxiong Kang is with the School of Automation Science and Engineering, South China University of Technology, Guangzhou 510641, China (e-mail: auwxkang@scut.edu.cn).

Color versions of one or more of the figures in this article are available online at https://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TMM.2020.3002185

network such as ResNet and VGG. For output space learning, we devise a Two-Stream Graph Convolutional Network (TSGCN), which incorporates object spatial graph and label semantic graph to learn multi-label classifiers. With the label semantic graph capturing the overall semantic correlations in the label space, the object spatial graph aims to model the spatial relationships among object regions in images which are generated by the underlying network such as Mask R-CNN [12] and YOLO [22]. By combining the pooled features and the TSGCN learned classifiers, we can learn from both input and output spaces for multi-label image classification in an end-to-end setting. Experimental results on two popular benchmark datasets show that our proposed method outperforms the state-of-the-arts.

In summary, the key contributions of our work are as follows:

1) We propose a novel deep learning framework which undertakes joint learning in both input space and output label space for multi-label image classification.
2) We design a new pooling strategy, namely label-specific pooling, for ConvNets to generate label-specific features. Our pooling method can enhance the discriminative power of convolutional features while maintaining a high down-sampling ratio.
3) We design a novel Two-Stream GCN (TSGCN) for the learning of label-wise classifiers, which coordinates an object spatial graph and a label semantic graph. Our TSGCN can simultaneously map the spatial relationships among object regions in an image and latent label semantic correlations in the output label space.
4) We further study the generalization capability of our proposed framework through comprehensive experiments. We validate multiple ConvNet variants as the backbone convolutional feature extractors and discover consistent performance improvements over the baseline.

The rest of this paper is organized as follows. The relevant studies are reviewed in Section II. The proposed method is described in detail in Section III. The experimental results on the two benchmark datasets are reported in Section IV, followed by conclusions and future work in Section V.

II. Related Work

In general, the existing approaches to multi-label image classification can be categorized into two types: input space learning and output space learning. Input space learning mainly focuses on extracting low-dimensional discriminant representations of images for the subsequent classification task, while output space learning aims to learn effective classifiers by exploring the correlation among output labels. Therefore, in this section, we review the literature in terms of these two types as well the hybrid type.

A. Input Space Learning

Conventional visual feature extraction algorithms generally produce a unified set of features to be commonly shared by all labels. In order to produce discriminant features specific for each label, four types of studies have been conducted, namely, feature selection, label-specific feature learning, multi-instance learning, and multi-task learning.

1) Feature Selection: Feature selection is a straightforward approach that selects a subset of features from the common set of features for each class label. In some studies [3]–[5], the single-label multi-class setting was adapted to directly handle the multi-label cases. For example, MReliefF and MF-statistic were extended from the traditional ReliefF and F-statistic filter algorithms [4] to address the multi-label classification task. Yan *et al.* [5] proposed a graph-margin based multi-label feature selection algorithm, which represents multi-label data with a graph and applies the large margin theory to the learned features. MLNB (Multi-Label Naive Bayes) [3] is a multi-label version of the wrapper feature selection approach, of which the core search strategy is a genetic algorithm. It consists of a multi-label Naive Bayes classifier to supervise the selection of the most effective features. Ma *et al.* [23] employed subspace-sparsity to supervise feature selection. However, these approaches mainly focus on selecting a subset from the unified feature set, which limits the discriminative power of the selected features. Some even do not consider the correlations among labels in the feature learning process.

2) Label-Specific Feature Learning: Instead of selecting a subset from the unified feature set, some studies aim to learn label-specific features. Zhang and Wu [6] proposed LIFT (multi-label learning with label-specific features) which extracts label-specific features by clustering instances sharing the same labels as final representations. LIFT in fact can be regarded as a feature mapping technique which ignores label correlations. Therefore, variants of the LIFT framework have been proposed to utilize label correlations in the feature extraction process. Some researchers further extended LIFT features through label-specific feature dimension reduction way, such as selecting a subset of LIFT features with the fuzzy rough set technique [9]. LLSF (Learning Label-Specific Features) [7] exploited second-order label correlations in the process of learning label-specific features. JFSC (Joint Feature Selection and Classification) [8] incorporated feature selection and classification for multi-label learning. In JFSC, both shared features and label-specific features were learned by exploring pairwise label correlations.

3) Multi-Instance Learning: Multi-label image classification can also be perceived as a multi-instance learning task where at least one instance (e.g., an object region) in a given image corresponds or is related to each label. The studies of this category are generally known as Multi-Instance Multi-Label Learning (MIML) [24]. MIMLBoost [25] was proposed to segment training samples to bags of instances and utilize traditional single instance boosting algorithms as a transformation of the problem. Huang *et al.* proposed MIMLfast [26] which constructs a low-dimensional subspace shared by all labels and employs stochastic gradient descent to train specific linear models. Ding *et al.* [1] devised a context-aware MIML model for image annotation, which uses a graph to model the contextual relationships among instances in bags. Some other studies have also been inspired by the idea of MIL [27]–[29]. Note that the representations used in these studies are unified for all labels, which limits its discriminative power.

4) Multi-Task Learning: Multi-task learning has also been utilized for multi-label image classification by treating predicting each label as one task. DirtyLasso [30] utilized different normalization strategies when feature extraction is undertaken to differentiate shared features and task-specific ones ($\ell_{1,\infty}$-norm and ℓ_1-norm, respectively). However, it does not take the correlations among different tasks into consideration. GFLasso [31] was proposed to compute the distances between task coefficient vectors in order to constrain related tasks from sharing common features. It also employed ℓ_1-norm for task-specific features extraction. While all the other methods are built on feature-vector instance learning, the authors in [32] first proposed a multi-task graph classification method to deal with structured graph data.

B. Output Space Learning

1) Conventional Methods: Early approaches for output space learning based multi-label classification trained independent binary classifiers and predicted label occurrence independently. One typical approach is binary relevance [13], which treats the existence of each label as a binary classification task. Similar studies include label power set (LP) [14] and pruned problem transformation (PPT) [33]. These methods overlook label relationships and leave the number of predicted labels not regularized. To overcome this issue, researchers have taken label correlations into consideration and adapted conventional algorithms under the multi-label setting. This kind of approaches are known as algorithm adaptation methods, including ML-KNN [15], ML-DT [16], and RankSVM [17].

2) Deep Learning-Based Methods: More recently, deep learning techniques have been explored to model label dependencies [19], [20], [34], [35]. Gong *et al.* [34] trained a CNN with a ranking-based learning strategy and devised a weighted approximated-ranking loss optimal for multi-label image classification. CNN-RNN [19] maps both image convolutional features and label vectors into a joint embedded space, where RNN was employed to model correlations among labels. Attention mechanisms has also been utilized to handle the label correlations. Zhu *et al.* [20] utilized a spatial regularization network under a multi-label classification setting, which models semantic and spatial relations among labels. Chen *et al.* [35] introduced reinforcement learning into a RNN to model label dependencies among salient regions. Similarly, Huang *et al.* [26] proposed a fast MIML algorithm which captures label correlations as sub-concepts. The DeepMIML network [36] models the sub-concepts in a network layer, and achieves superior performance on multi-label learning across different modalities.

3) Graph-Based Methods: As a topology structure, graph representation could effectively model the complex relationships under the multi-label setting. Therefore, graph models have been widely used as a practical regularization for label co-occurrence and dependencies [11], [18], [37]–[39]. A label space tree-structured graph [38] was proposed to model label dependencies with the maximum spanning tree algorithm. A graphical Lasso-based conditional label structure was proposed in [18]. Similarly, structure knowledge graphs were employed by Lee *et al.* [39] to model the correlations across different labels. Besides, Chen *et al.* [11] proposed to apply Graph Convolutional Network (GCN) to simultaneously learn the classifiers and model label dependencies, which achieved the state-of-the-art performance on multi-label image classification.

In a word, these output space learning based methods simply use unified representations for all labels, which limits the discriminative capability and eventually the classification performance. Although the effectiveness of GCN has been demonstrated, the correlations modelled by far are still limited as only the label semantic relationships have been considered. In our study, we design TSGCN which consists of two streams of GCN to model label semantic correlations and object spatial relationships, respectively.

C. Hybrid Approaches

Very few studies have explored joint learning of input space and output space [8], [40]. JFSC [8] was proposed to take pairwise label correlations into consideration for learning label-specific features and classifiers simultaneously. Jiang *et al.* [41] proposed a multi-label metric transfer learning (MLMTL) method, which is a distribution-adaptation-based method to address the potential distribution differences between the training domain and the test domain in the instance space and the label space. MLMTL extends the traditional maximum mean discrepancy method for the multi-label setting which can learn variable weights for training instances. Similarly, Xu *et al.* [40] dealt with the complicated cross-covariance operator in multi-label feature extraction by utilizing a linear kernel for the input space and a delta kernel for the output space to approximate the cross-covariance. With the joint consideration of both spaces, it can lead to an effective approximated and symmetrized representation.

Overall, these studies have not taken advantage of deep learning techniques as they are working on conventional features. Moreover, the mapping of label correlations during classifier learning has been either implicit or ignored. By contrast, we propose an end-to-end deep learning framework which can jointly learn from both input space and output space for multi-label image classification. Our framework refines the powerful convolutional features for label-specific representation and explicitly maps both label semantic correlations and object spatial relationships to learn classifiers.

III. Proposed Method

As shown in Fig. 1, our proposed deep learning framework for multi-label image classification consists of two major components: the image representation learning network and TSGCN for classifier learning. After problem formulation, we first describe the details of the proposed label-specific pooling strategy, and then briefly revisit the concept of GCN, followed by the explanation on extending the conventional GCN to map both label semantic correlations and object spatial relationships for multi-label classifier learning.

A. Problem Formulation

For a multi-label image dataset $\mathcal{D} = (\mathbf{X}, \mathbf{Y})$ with N images, let the matrix $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots \mathbf{x}_N]^{\mathsf{T}} \in \mathbb{R}^{N \times M}$ denote the

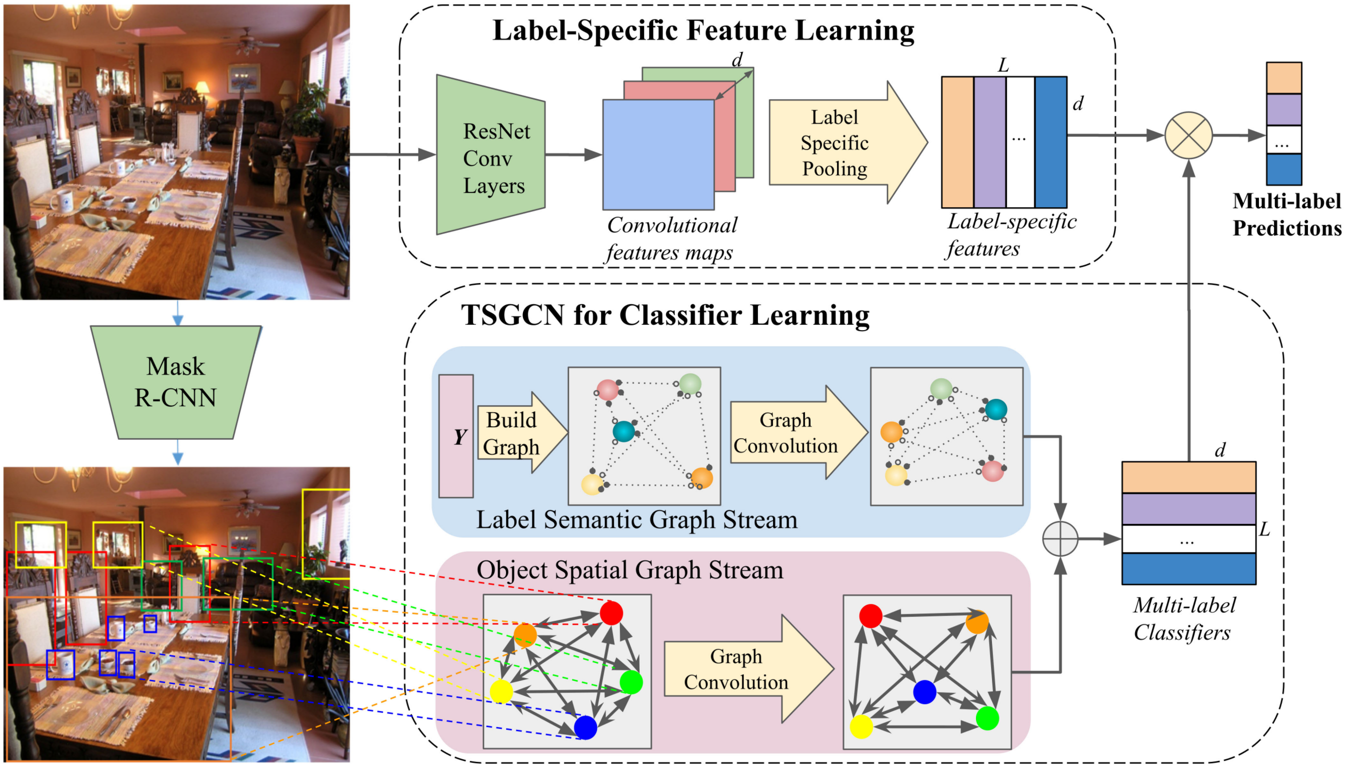


Fig. 1. Illustration of our proposed deep learning framework for multi-label image classification. Overall our model consists of two major components: representation learning and classifier learning. The upper part is image representation learning with label-specific feature pooling of ResNet features, aiming to learn label-specific features with improved discriminative power. The lower part is the Two-Stream GCN designed to learn multi-label classifiers, which consists of a semantic graph and a spatial graph. The upper stream models the label correlations using the ground truth labels from the training set, and the lower stream utilizes R-CNN output to construct an object spatial relation graph for each image. The nodes in the graph are for illustration purpose only.

training samples as the input space, and $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \ldots \mathbf{y}_N]^\mathsf{T} \in \{0,1\}^{N\times L}$ as the output label space, where L is the total number of labels and M is the feature dimension of each image. For the i-th sample $\mathbf{x}_i \in \mathbf{X}$, the ground truth label would be a L-dimensional vector, $\mathbf{y}_i = [y_{i,1}, y_{i,2}, \ldots y_{i,L}]$. For each class label, if the j-th label is associated with $\mathbf{x}_i$ then $y_{i,j} = 1$; otherwise $y_{i,j} = 0$. Our final objective is to learn a set of parameters θ to be used in our network for accurate multi-label image classification, as well as the refinement of label-specific features. To achieve this objective, we would expect the parameters learned to have the following functions:

1) mapping the samples from the input space $\mathbf{X}$ to the output space $\mathbf{Y}$ appropriately;
2) discriminating features specific to each label from the entire feature set accounting for label correlations;
3) mapping label semantic correlations and object spatial relationships to learn inter-dependent classifiers for multi-label classification.

Therefore, we can formulate the objective function $\mathcal{J}$ of our task as

$$\min_{\theta} \mathcal{J}(\theta) = \mathcal{L}(\theta) + \tau\mathcal{S}(\theta) + \lambda\|\theta\|_2, \tag{1}$$

where $\mathcal{L}()$ is the loss function for multi-label classification, a binary cross-entropy loss on the designed classifier which we will describe in detail later. Additionally, $\mathcal{S}()$ is a constraint to guide the extraction of the label-specific features. The last term is the ℓ^2 norm regularization of the parameters. While τ and λ are the hyper-parameters used to control the coefficients of the regularization terms. Based on the framework structure shown in Fig. 1 and the overall objective function, we can divide it into two parts: label-specific feature learning and TSGCN for multi-label classifier learning, described in III-B and III-C, respectively.

B. Label-Specific Feature Learning

We use pre-trained ResNet as preliminary feature extractor. Let $\mathbf{X}^{conv} \in \mathbb{R}^{w\times h\times d}$ denote the extracted convolutional features, in which w and h are the width and height of the final feature maps, and d is the number of feature maps in the last convolutional layer. The objective of our label-specific learning process is to build a function $P()$ that can learn the label-specific features $\mathbf{X}^{lsp}$ from the convolutional features $\mathbf{X}^{conv}$ as

$$\mathbf{X}^{lsp} = P(\mathbf{X}^{conv}). \tag{2}$$

In order to implement label-specific feature learning in a neural network, we need to represent the process as a matrix operation. We apply matrix multiplication on the convolutional features to compute a feature vector for each label. The feature vectors corresponding to all the labels will be concatenated together as the pooled features. The features generated by this pooling process would be a much smaller matrix compared to the input features. Our pooling process is noticeably different from the widely-used pooling methods such as max-pooling and

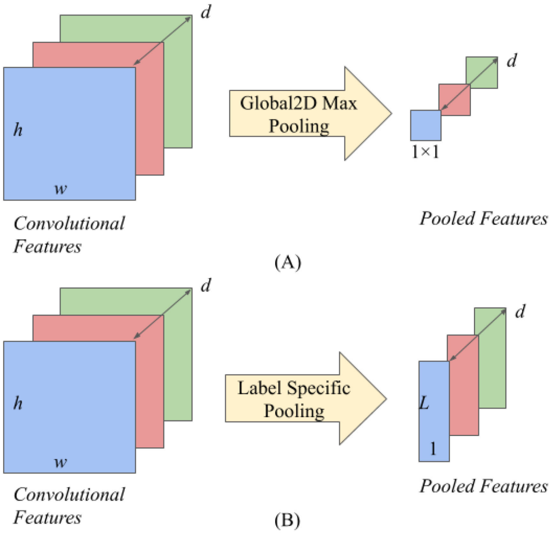


Fig. 2. Comparison of our proposed pooling method and the global max-pooling method. Global max-pooling simply selects the maximum value for each feature map, while our proposed pooling strategy would utilize matrix multiplication to compute a representative value for each label from each feature map.

average-pooling, which aims for the maximum value or the mean of all elements in each feature map.

We illustrate the difference between our proposed pooling method and global max-pooling in Fig. 2. Note the label-specific pooling strategy can be implemented with more than one fully-connected layer. For the sake of simplicity, we explain label specific pooling with only one fully-connected layer. Mathematically, the label-specific pooling process can be expressed as

$$\mathbf{X}^{lsp} = P(\mathbf{X}^{conv}) = \mathbf{W}^{lsp} \cdot R(\mathbf{X}^{conv}), \tag{3}$$

where $\mathbf{W}^{lsp} \in \mathbb{R}^{L \times wh}$ denotes the parameters to be learned for the label-specific pooling layer. We now have $\mathbf{W}^{lsp} = [\mathbf{w}_1^{lsp}, \mathbf{w}_2^{lsp}, \ldots \mathbf{w}_L^{lsp}]^{\mathsf{T}}$ as a subset of all parameters θ, with each vector $\mathbf{w}_j^{lsp} \in \mathbb{R}^{wh}$ representing the pooling parameters for the j-th label. $R()$ denotes the operation of projecting the convolutional features into the dimension of $wh \times d$, which is necessary to align the dimensions for subsequent calculations.

Now the label-specific features $\mathbf{X}^{lsp} \in \mathbb{R}^{L \times d}$ can be computed as

$$\mathbf{X}^{lsp} = \mathbf{W}^{lsp} \cdot R(\mathbf{X}^{conv}) = \begin{bmatrix} {\mathbf{w}_1^{lsp}}^{\mathsf{T}} \\ {\mathbf{w}_2^{lsp}}^{\mathsf{T}} \\ \cdots \\ {\mathbf{w}_L^{lsp}}^{\mathsf{T}} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{x}_1^{conv} \\ \mathbf{x}_2^{conv} \\ \cdots \\ \mathbf{x}_d^{conv} \end{bmatrix}^{\mathsf{T}}, \tag{4}$$

where $\mathbf{x}_i^{conv} \in \mathbb{R}^{1 \times wh}$ is the projected vector from the i-th convolutional feature map. Therefore, our label-specific pooling would generate pooled features at the same down-sampling ratio as the commonly used Global2D Max-Pooling for each label.

Although the weights in a neural network can be eventually learned through error back-propagation and gradient descent, the learning process could be oscillated. In order to accelerate the learning and obtain discriminative label-specific features, we revise the loss function to better supervise the learning by considering the following two aspects.

Firstly, for each label, only a portion of the entire feature set represents the most discriminative features. Therefore, we can treat the sparse parameters $\mathbf{W}^{lsp}$ as masks to be applied to the convolutional features, and the non-zero elements indicate the most discriminative features for different labels. An ideal label-specific feature learning method is expected to identify the useful subsets for all the class labels. Therefore, the label-specific features learned should be sparser than the entire set of features. Thus we revise the overall objective function Equation (1) as

$$\min_{\theta} \mathcal{J}(\boldsymbol{\theta}) = \mathcal{L}(\theta) + \tau \mathcal{S}(\mathbf{W}^{lsp}) + \lambda \|\theta\|_2 + \gamma \left\|\mathbf{W}^{lsp}\right\|_2, \tag{5}$$

in which we utilize an extra ℓ^2 norm of $\mathbf{W}^{lsp}$ as a commonly used regularizer in DNNs to ensure the sparsity of pooling parameters.

Secondly, in multi-label image classification, different labels are usually correlated with each other. To learn the label-specific features, we hold an assumption that the strongly correlated labels should share more features than those that are uncorrelated or weakly correlated. For example, a sample associated with a specific label y_a could be more likely to have another label y_b attached, while being hardly possible to have label y_c associated at the same time. In this example, labels y_a and y_b are highly correlated; therefore, the features discriminative to these two labels should be similar. In contrast, for less correlated or unrelated label pairs such as y_a and y_c, the features discriminative to one label may not be applicable for the other one.

As aforementioned, the vector $\mathbf{w}_j^{lsp} \in \mathbb{R}^{wh}$ denotes the parameters to calculate the label-specific features for the j-th label. In this vector, the i-th non-zero element $w_{i,j}^{lsp}$ would represent the discriminative power of the i-th feature dimension for label y_j. In other words, a greater $w_{i,j}^{lsp}$ value means that the i-th feature dimension would be more useful to distinguish label y_j, and would be retained as part of the label-specific features for label y_j. Thus, to ensure that the features learned from label-specific pooling are discriminative, we incorporate the label correlations using a constraint term $\mathcal{S}(\mathbf{W}^{lsp})$ in the objective function. For a specific label y_i, this term is defined as $\sum_{j=0}^{L}(1 - C_{i,j}){\mathbf{w}_i^{lsp}}^{\mathsf{T}}\mathbf{w}_j^{lsp}$, where $\mathbf{C} \in \mathbb{R}^{L \times L}$ is the correlation matrix of the labels constructed through a data-driven approach, as described in [11]. To be more specific, the label correlation is reflected by their co-occurrence in the dataset, and we utilize conditional probability to represent label correlation dependency. For example, $P(y_j|y_i)$ denotes the probability of the existence of label class j when label class i exists. Note $P(y_j|y_i)$ is different from $P(y_i|y_j)$, therefore the correlation matrix $\mathbf{C}$ would be asymmetrical. Since $C_{i,j}$ is the measure of the correlations between label class i and j, we use $(1 - C)$ to restrict unrelated labels from sharing the same feature dimensions, which hence enhances the disciminative power of the label-specific features learned. When considering all the different labels simultaneously, this equation would be generalized as

$$\mathcal{S}(\mathbf{W}^{lsp}) = \mathrm{Tr}\left((1 - \mathbf{C})\mathbf{W}^{lsp}{\mathbf{W}^{lsp}}^{\mathsf{T}}\right), \tag{6}$$

where $\mathrm{Tr}\,()$ is the trace of a matrix. And the overall objective function Equation (1) can be rewritten as

$$\min_{\theta} \mathcal{J}(\boldsymbol{\theta}) = \mathcal{L}(\theta) + \tau(\mathrm{Tr}\,((1-\mathbf{C})\mathbf{W}^{lsp}\mathbf{W}^{lsp^{\mathsf{T}}})) + \lambda\|\theta\|_2 + \gamma\left\|\mathbf{W}^{lsp}\right\|_2. \tag{7}$$

With the label-specific constraints introduced, our proposed network can learn more discriminative label-specific features with the supervision of label correlations. The coefficient of this constraint term τ is a hyper-parameter and we would explore its optimal setting in the experiment section.

C. TSGCN for Classifier Learning

1) Graph Convolutional Network: The overall objective of a GCN is to learn a non-linear function $f()$ for each layer. The input of the function $f()$ for layer l consists of two components: the node representations $\mathbf{H}^l$ and the adjacency matrix $\mathbf{A}$ needed for graph convolution. In our study, the dimensions of $\mathbf{H}^l$ and $\mathbf{A}$ would be determined by the number of class labels. We can denote them as $\mathbf{H}^l \in \mathbb{R}^{L\times D}$ and $\mathbf{A} \in \mathbb{R}^{L\times L}$, where D is the dimension of a node's feature. Furthermore, GCN layers are stackable and the node features would be updated as $\mathbf{H}^{l+1} \in \mathbb{R}^{L\times D'}$, in which D' denotes the feature dimension of the updated node at the $l+1$-th layer. Therefore, the non-linear function for the l-th layer of a GCN can be formulated as

$$\mathbf{H}^{l+1} = f(\mathbf{H}^l, \mathbf{A}). \tag{8}$$

By applying the convolutional operation, the equation above can be rewritten as

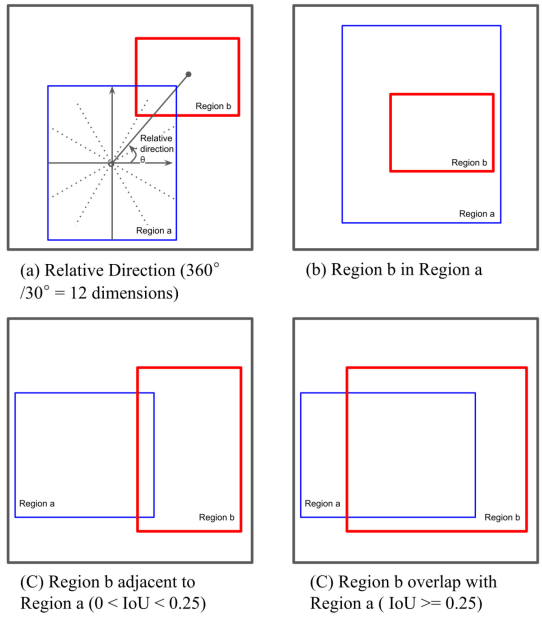
$$\mathbf{H}^{l+1} = \sigma(\hat{\mathbf{A}}\mathbf{H}^l\mathbf{W}^l), \tag{9}$$

where $\mathbf{W}^l \in \mathbb{R}^{D\times D'}$ is the transformation matrix to be learned for each layer, $\hat{\mathbf{A}}$ is the normalized $\mathbf{A}$, and $\sigma()$ is a non-linear activation. With GCN, we can stack several layers to map and model complicated relationships between class labels.

Based on the original GCN, we propose a two-stream GCN as shown in the lower part of Fig. 1. The first stream aims to model label semantic relationships, while the second stream works on object spatial relationships within images. The details on both streams are described in the following two subsections.

2) Label Semantic Graph Stream: Despite that graph structure is generally pre-defined in many tasks, there is no ready-to-use graph (the adjacency matrix, to be specific) available for our multi-label image classification task. Thus, we need to build our own graph structure first.

Specifically, we follow the method in [11] to construct the adjacency matrix through a data-driven approach. This approach exploits label co-occurrence within the dataset with conditional probability to define the adjacency matrix. Conditional probability is considered when mapping the label correlations. For example, $P(y_a \mid y_b)$ represents the conditional probability of label y_a based on the occurrence of label y_b. Therefore, $P_{b,a} = P(y_b \mid y_a)$ would be different from $P_{a,b} = P(y_a \mid y_b)$, where $a \neq b$. As a result, the adjacency matrix $\mathbf{A}^{ls}$ for the label semantic graph stream would be asymmetrical. In order to minimize the impact of possible noisy co-occurrence pairs, a threshold has been set to binarize the adjacency matrix $\mathbf{A}^{ls}$. For more information on the construction of the label semantic graph, interested readers could refer to [11] for more details.



(a) Relative Direction (360° /30° = 12 dimensions)

(b) Region b in Region a

(C) Region b adjacent to Region a (0 < IoU < 0.25)

(C) Region b overlap with Region a (IoU >= 0.25)

Fig. 3. Different spatial object relationships defined in our study to establish the object region spatial graph (blue bounding box denotes the predicted region of object a, red bounding box denotes the predicted region of object b).

3) Object Spatial Graph Stream: We apply a pre-trained Mask R-CNN network [12] to preliminarily recognize and localize the objects in an image. The R-CNN sub-network would generate a set of regions possibly containing label objects, noted as $R = r_1, r_2, \ldots r_K$ where K is the total number of regions in the R-CNN output. For each of the regions, R-CNN would predict a bounding box $\hat{B}(x, y, w, h)$ and a class label $\hat{c}$ with a confidence score $\hat{s}$. Within the bounding box prediction, (x, y) denotes the location (x-axis, y-axis coordinates) and (w, h) denotes the size (width and height). To simplify the graph and have a consistent final output with the other stream, we only select the region with the highest confidence score for each class as the node in the graph. With the nodes selected, we can construct the adjacency matrix $\mathbf{A}^{sr}$ with the directional spatial relation between object regions.

For two regions r_a and r_b whose corresponding labels are y_a and y_b, their normalized centroids coordinates can be denoted as (x_a, y_a) and (x_b, y_b). We can compute the IoU between r_a and r_b, as well as the relative direction $\beta_{a,b}$ to define the spatial relations. Some particular cases have also been taken into consideration when defining the spatial relations between r_a and r_b. More specifically, we define a total number of 15 spatial relations between pairs of regions, as shown in Fig. 3. We first define relationships between the regions based on the relative direction γ from the centroid of r_a and to that of r_b. We divide 360 degrees around a region to 12 sectors of 30 degrees each, where each sector represents a type of spatial relation. Note that we would only use the relative direction relations if the region pair is not falling in the redefined relationships. Then we define the special relationships between region pairs. If one region is

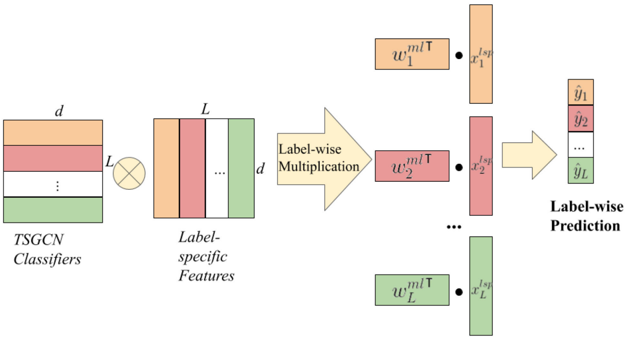


Fig. 4. Illustration of TSGCN for multi-label classification. For each label, there is a feature vector to be multiplied with the corresponding TSGCN output, which results in a predicted confidence score for the specific label in an image.

entirely contained within the other region, this would be another relationship. Moreover, for region pairs with an IoU greater than 0.25, we define this relationship as overlap pairs. We also define the relationship of pairs of regions bordering with each other, when their IoU is greater than 0 but less than 0.25. With the 15 spatial relationships defined above, we can construct the spatial relation graph by connecting the edges of corresponding relationships between region pairs. Note that the order of the pairs would impact the relationship, as the defined spatial relations are directional. For example, region r_a is fully contained in region r_b is totally different from region r_b in region r_a. Therefore, the correlation matrix will also be asymmetric, which is the same as the semantic graph stream. For the labels which do not have candidate regions generated by R-CNN, no edges would be connected to the nodes in the graph.

4) Multi-Label Classification With TSGCN: Different from the semi-supervised classification task where GCN is originally introduced, we aim to use a GCN network to train a series of inter-dependent classifiers to perform multi-label image classification. To this end, our GCN is designed to have the same number of nodes in the graph as that of class labels in datasets. We further make the dimension of the final nodes equal to that of the pooled features to enable a dot product calculation. As a result, the final output of each node can be regarded as a binary classifier for a single label. As noticed, the learned classifiers are a set of weights to be applied on the features extracted by the label-specific pooling, denoted as $\mathbf{W}^{ml} = [\mathbf{w}_1^{ml}, \mathbf{w}_2^{ml}, \ldots \mathbf{w}_L^{ml}]^\top$, where L is the total number of classes in the dataset. In our GCN, the number of nodes in both streams is equal, while the inputs are slightly different. The input of the GCN for the label semantic stream is $\mathbf{Z} \in \mathbb{R}^{L \times E}$, where E is the dimension of the label word embeddings, while the input for the object spatial stream is $\mathbf{Z}' \in \mathbb{R}^{L \times Q}$, where Q is the dimension of the region features extracted by R-CNN. And the last layer of both streams would produce $\mathbf{W}^{ml} \in \mathbb{R}^{L \times d}$, where d represents the dimension of the label-specific features learned. We combine the final output from each stream together and use their average as the final weights.

The classification process is illustrated in Fig. 4. By multiplying the learned weights to the extracted label-specific features, we can compute the confidence score for the i-th label of a given image as

$$\hat{y}_i = \mathbf{w}_i^{ml\top} \mathbf{x}_i^{lsp}, \tag{10}$$

where $\mathbf{x}_i^{lsp} \in \mathbb{R}^d$ denotes the features specific to the i-th label for the given image, $\mathbf{w}_i^{ml} \in \mathbb{R}^d$ is the average of the i-th final node representations from both streams of TSGCN. The confidence score $\hat{y}_i$ denotes the possibility of the i-th label is associated with the given image. While the ground truth label is $\mathbf{y} \in \mathbb{R}^L$, we adopt binary cross-entropy loss as multi-label classification loss during training. Therefore, the loss function $\mathcal{L}()$ in Equation (1) would be

$$\mathcal{L}(\theta) = \frac{1}{L} \sum_{i=1}^{L} (y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)). \tag{11}$$

In the training stage, we will build the label semantic graph by following the method described in [11]. For each training image, we will feed it into R-CNN and use the output to build the object spatial graph. With error back-propagation and gradient descent optimization, both GCN streams would be updated in each batch to learn the classifiers for each label. At the same time, the image representation learning branch would also be updated to learn the optimal mask for label-specific features. The pre-trained ResNet convolutional layers would also be fine-tuned in the training process.

During the testing process, the label semantic graph would remain the same for all test instances. Each test image will be fed into the fine-tuned ResNet for convolutional feature extraction and into R-CNN for the possible generation of object regions. The GCN for the object spatial stream will carry out corresponding graph convolution and update the classifiers with preliminary recognition and localization of objects for each test image. The label-specific features would be fed into updated multi-label classifiers for final prediction.

IV. Experiments

A. Datasets

We chose MS-COCO [45] and PASCAL VOC [46] as the evaluation datasets for our study, since they are widely used for benchmarking multi-label image classification methods. Both datasets have detailed annotations with multiple objects in images. MS-COCO consists of a training set with more than 80,000 images and a validation set with more than 40,000 images. While the ground-truth labels are not available for the test set, we utilize the validation set to evaluate the performance of our method. There are 80 different classes of objects in the images. The positive labels indicate the existence of objects in images. As a multi-label image dataset, MS-COCO 2014 dataset has an average of 2.9 labels for each image. However, the number of co-existing labels is not evenly distributed across the images, making it even more challenging.

PASCAL VOC (Visual Object Classes) is a computer vision challenge hosted from 2005 to 2012. Among the years, the VOC 2007 dataset has been the most widely used one. There are only 20 different classes of objects in the VOC dataset, resulting in a smaller output space. In total there are 9,963 images equally

TABLE I
EXPERIMENTAL RESULTS ON THE MS-COCO DATASET IN TERMS OF OVERALL AND CLASS-WISE PERFORMANCE

	All							Top-3					
Methods	mAP	CP	CR	CF1	OP	OR	OF1	CP	CR	CF1	OP	OR	OF1
CNN-RNN [19]	61.2	-	-	-	-	-	-	66.0	55.6	60.4	69.2	66.4	67.8
RNN-Attention [42]	-	-	-	-	-	-	-	79.1	58.7	67.4	84.0	63.0	72.0
Order-Free RNN [43]	-	-	-	-	-	-	-	71.6	54.8	62.1	74.2	62.2	67.7
ML-ZSL [39]	-	-	-	-	-	-	-	74.1	64.5	69.0	-	-	-
SRN [20]	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 [44]	-	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) [11]	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) [11]	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
Ours (Binary)	83.5	81.5	72.3	76.7	84.9	75.3	79.8	84.1	**67.1**	74.6	89.5	**69.3**	**78.1**
Ours (Re-weighted)	**83.7**	83.9	**73.1**	**78.1**	85.6	**75.9**	**80.5**	85.4	66.9	**75.0**	89.9	68.6	77.8

split into training and test sets annotated with a total of 24,640 objects.

B. Experimental Settings

We use a pre-trained ResNet-101 network as the feature extractor which produces 2,048 feature maps in the last convolutional layer. Our label-specific pooling can down-sample the convolutional features to a 2,048-dimensional feature vector for each label. In the process of feature learning, we adopt ReLU activation function for its effectiveness in faster convergence as demonstrated in other studies. We have applied data augmentation on the training images through random cropping and horizontal flipping to alleviate the impact of over-fitting, along with batch normalization on the convolutional features and pooled features.

For the classifiers, we train the same number of classifiers as the total number of labels in the dataset, with one classifier for each class label. Two GCN layers are used for both streams of our TSGCN. We utilize a trained GloVe word embedding [47] to build our label semantic graph, similar to what has been done in ML-GCN [11]. We use a pre-trained Mask R-CNN [12] to extract the object regions and build the object spatial graph. The GCN's final nodes from both streams will be aggregated in average as the final classifier weights for all the class labels.

We utilize SGD optimizer for network optimization with the momentum set as 0.9. We set the initial learning rate as $1e^{-3}$, with a decay factor of 0.01 for each epoch. We would also halve the learning rate once the training has been on a plateau to accelerate the convergence. Our model was trained for a total of 200 epochs, with early stopping strategy to avoid over-fitting the training set.

C. Evaluation Metrics

Following other studies on multi-label learning [11], [44], we adopt several commonly used metrics for both class-wise and overall performance evaluation. For each class, the metrics include class-wise precision(CP), class-wise recall (CR), and class-wise F1 score (CF1). We also report the overall average precision (OP), overall recall (OR) and overall F1 score(OF1). To provide a direct comparison with other methods, we include the evaluation of top-3 labels by confidence score. We apply the same set of metrics for top-3 labels. Additionally, mean average precision (mAP) has been reported, as it is a widely-accepted metric for multi-label learning tasks. Among the different metrics, the F1 Scores (OF1 and CF1) and the mean average precision (mAP) are generally regarded as the most important ones [11], [19].

D. Experimental Results

1) Overall Performance: We first report the classification performance on the MS-COCO dataset in Table I, where we compare our proposed method with various state-of-the-art methods, including CNN-RNN [19], RNN-Attention [42], Order-Free RNN [42], ML-ZSL [39], SRN [20], ResNet-101 [10], Multi-Evidence [44] and ML-GCN [11] (both Binary and Re-weighted schemes). As mentioned before, there are no ground-truth labels for the test set. Thus we follow the common practice as seen in other studies [11], [20], [42], which use the validation set provided as test set, and withhold part of the training set as a validation split to train the model.

As noticed in Table I, our proposed method consistently outperforms the state-of-the-art methods in most metrics, which validates the effectiveness of our proposed framework on joint learning from both input and output spaces. It is worth noting that ML-GCN [11] has a variant which utilizes a re-weighted scheme when constructing the adjacency matrix instead of the binary matrix. We report the results of our methods using both the binary and the re-weighted matrix to provide a fair comparison. When compared with the state-of-the-art method ML-GCN with the binary adjacency matrix, our proposed method improves both recall score and F1 score. These improvements can be attributed to the label-specific feature pooling and the object spatial graph in TSGCN included in our model. In other words, the effective extraction of label-specific features enhances the representative power, and the employment of the spatial graph reduces the possibility of missing objects in images. However, the improvements are not significant when the re-weighted matrix is involved in both methods. The narrower performance gap also reflects the effectiveness of the object spatial graph in TSGCN from another aspect, as it has played a similar but better role in modelling the label correlations.

We also report the results on the PASCAL VOC dataset. As the test ground-truth labels are available here, we train our model

TABLE II
EXPERIMENTAL RESULTS ON THE PASCAL VOC 2007 DATASET IN TERMS OF CLASS-WISE PRECISION AND MEAN AVERAGE PRECISION (MAP)

Methods	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	motor	person	plant	sheep	sofa	train	tv	mAP
CNN-RNN [19]	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 [48]	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
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 [49]	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 [27]	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 [42]	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 [35]	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
ML-GCN(Binary) [11]	**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) [11]	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
Ours(Binary)	99.3	98.1	96.3	95.5	86.4	94.6	97.2	97.1	**85.4**	92.1	**89.7**	98.2	96.5	96.9	98.9	**85.1**	**96.7**	**87.5**	98.6	93.8	94.2
Ours(Re-weighted)	98.9	**98.5**	96.8	97.3	**87.5**	94.2	**97.4**	97.7	84.1	92.6	89.3	**98.4**	98.0	96.1	98.7	84.9	96.6	87.2	98.4	93.7	**94.3**

TABLE III
EXPERIMENTAL RESULTS OF USING DIFFERENT NUMBER OF LAYERS IN THE LABEL-SPECIFIC POOLING PROCESS

Dataset	MS-COCO						VOC
# of Layers		All		Top-3			
in LSP	mAP	CF1	OF1	CF1	OF1		mAP
No LSP	81.2	74.7	79.2	72.8	76.5		92.5
1 layer	82.4	75.9	79.3	73.1	77.6		**94.2**
2 layers	**83.5**	**76.7**	**79.8**	**74.6**	**78.1**		93.6
3 layers	79.4	74.2	78.4	71.7	72.9		92.7

TABLE IV
MEAN AVERAGE PRECISION (MAP) BASED ON LABEL-SPECIFIC POOLING USING DIFFERENT CONVNETS AS FEATURE EXTRACTOR

# of	MS-COCO				VOC
Layers	ResNet-101	ResNet-50	InceptionV3	VGG	VGG
No LSP	81.2	73.8	78.4	71.4	92.5
1 layer	82.4	**77.3**	79.3	**75.1**	**94.2**
2 layers	**83.5**	76.9	**80.1**	74.6	93.6

TABLE V
EXPERIMENTAL RESULTS OF INDIVIDUAL STREAMS IN TSGCN

Dataset	MS-COCO					VOC
TSGCN Model	mAP	CP	CR	CF1		mAP
Label Semantic Graph Only	81.1	81.7	71.3	76.2		93.1
Object Spatial Graph Only	82.2	79.4	73.6	76.4		93.4
TSGCN-Both Graph	83.5	81.5	72.3	76.7		94.2

with the train/validation set, then evaluate the trained model with the test set. To compare with the other methods, we use a similar set of metrics, including class-wise average precision (AP) and mean average precision (mAP). As shown in Table II, our proposed method outperforms the best existing model, ML-GCN as described in [11], by 1.1% in terms of mAP when both use the binary matrix. With the re-weighted adjacency matrix in place, our method is still able to achieve slightly better performance than ML-GCN model by 0.3% mAP. The performance improvements on the PASCAL VOC dataset are not as significant as those on MS-COCO, which is also observed in existing studies. Considering that the dataset is less complicated and its size is relatively small, it is likely that performance on this dataset has almost reached the saturation point.

2) Ablation Studies: Despite the overall classification performance improves, we also carry out ablation studies on the different components of our proposed framework. The following experiments aim to study how individual modules of our proposed model would affect the classification performance. We vary the number of layers in label-specific pooling and TSGCN to study the effectiveness of both modules while observing the overall model performance. We also test different convolutional networks as the backbone network to investigate the generalization capacity of proposed label-specific pooling.

We first investigate how the label-specific pooling would affect the classification performance by changing the number of pooling layers in the model. As shown in Table III, compared with directly using convolutional features(NO LSP), label-specific pooling can effectively improve multi-label classification performance. However, it is not simply *'the deeper the better'*. In our study, we use fully connected layers to implement the label specific pooling. Having too many network layers not only increases the computational cost but also compromises the classification performance as error accumulates and training difficulty increases. For the dataset with a smaller output space such as PASCAL VOC (20 class labels versus 80 for MS-COCO), a small number of layers would be sufficient to obtain the optimal label-specific features.

In addition, we carry out more experiments to study the generalization capability of the proposed label-specific pooling strategy, by replacing the ResNet with other well-known ConvNets. As shown in Table IV, compared to plain convolutional features, we can always achieve better multi-label classification results with the help of the label-specific proposed pooling strategy. Another interesting finding is that the relatively simple ConvNets(i.e. VGG, ResNet-50) usually witness better results when using fewer label-specific pooling layers. On the contrary, more complicated ConvNets would require extra label-specific pooling layers to achieve optimal results.

Secondly, we study the effectiveness of each stream in TSGCN by varying the number of layers in each stream. The results are reported in Table V and Fig. 5. For the GCN layers, the output node dimensions are always 1,024 except for the last layer, which would have a 2,048-dimensional output. From Table V we can notice that combining two streams using both graphs would result in the best performance. When looking into details, we can see how individual streams affect the performance: using label semantic graph would result in higher precision but lower recall score than using object spatial graph solely. In other words, introducing object spatial graph would reduce the chance of missing objects in images. Combining the two convoluted graphs in the final TSGCN would further improve the precision as each stream supplements each other. Referring to the results reported in Table II, it is also noticed that our method improves the class-wise precision for indoor objects, such as *bottle*, *chair*, *sofa* and *TV*. This finding reflects the effectiveness of the

TABLE VI
SAMPLE IMAGES IN THE MS-COCO VALIDATION SET AND PREDICTED LABELS USING DIFFERENT MODELS

Sample Images	Ground Truth Labels	Predicted Labels			
		LSP Only	Semantic Graph Only	Spatial Graph Only	LSP+TSGCN
	person, car, frisbee, chair	**person, car, frisbee, chair,** potted plant	**person, car, frisbee, chair**	**person, car,** *frisbee*, **chair,** potted plant, sandwich	**person, car, frisbee, chair,** potted plant
	person, bottle, cup fork, knife, spoon bowl, cake, table	**person,** *bottle*, **cup fork,** *knife*, **spoon, bowl, cake, table** sandwich	**person,** *bottle*, **cup fork, knife, spoon, bowl, cake, table** sandwich	**person,** *bottle*, **cup fork, knife, spoon, bowl, cake, table** sandwich, toothbrush	**person,** *bottle*, **cup fork, knife, spoon, bowl, cake, table** sandwich
	person, bottle, glass, cup, knife, spoon, bowl, orange, oven, sink, refrigerator	**person, bottle, glass, cup,** *knife*, **spoon, bowl,** *orange*, **oven, sink,** *refrigerator*	**person, bottle, glass, cup,** *knife*, **spoon, bowl,** *orange*, **oven, sink, refrigerator**	**person, bottle, glass, cup, knife, spoon, bowl,** *orange*, **oven, sink, refrigerator,** fork	**person, bottle, glass, cup, knife, spoon, bowl,** *orange*, **oven, sink, refrigerator**
	bottle, toilet, sink, potted plant	**bottle,** *toilet*, **sink, potted plant**	**bottle, toilet, sink, potted plant**	**bottle, toilet, sink, potted plant,** cup	**bottle, toilet, sink, potted plant**

Note: **labels in bold** denote correct predictions; *italic labels* denote missed labels in prediction; underlined labels indicate false positive labels in prediction.

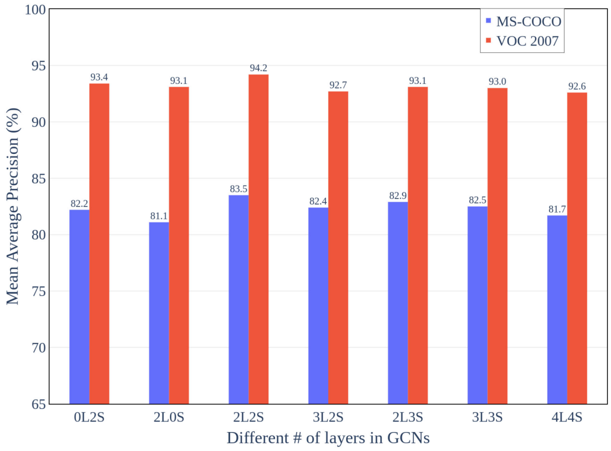


Fig. 5. Mean average precision on both datasets using different number of layers in each stream of TSGCN, L denotes the number of label semantic graph layers while S denotes the number of object spatial graph layers.

classifiers trained with object spatial graph, as they are capable of robustly mapping the related objects in a complex environment and therefore improving the overall classification performance. This finding can also be observed from the predicted labels of the samples shown in Table VI, which we will give more details below. We then explore how the depth of GCN would affect model performance. While using both graphs in TSGCN, from Fig. 5, it can be noticed that stacking more graph convolution layers would compromise the classification performance decreases on both datasets. The decrease in performance could be caused by the increasing accumulated propagation when more GCN layers are used.

Finally, to visually demonstrate the effectiveness of the modules in our proposed framework, we include several sample images from the MS-COCO validation set and the predicted results using varied models in Table VI. We tested three different sets of models: (1) using LSP only; (2) using each graph in TSGCN only; and (3) combining LSP and TSGCN as the optimal model. From the results, it is noticed that LSP can help distinguish highly similar objects, such as *frisbee* versus *bowl* in the first example. This is because LSP learns from the input space aiming for high discriminative power while not being impacted by the complicated correlation in the output space. While only a single stream of TSGCN is involved, relationship between objects or labels could be effectively modelled, leading to different results. Semantic graph utilizes the prior knowledge on label occurrence and is helpful for better understanding a scene. For example, in the last sample image, using semantic graph can help identify *toilet* which is missed in the LSP Only model. On the other hand, spatial graph which models object spatial relationship can further reduce the possibility of missing tiny objects. However, over-confidence could lead to false positive in some cases. Taking the second sample image as an example, the Spatial Graph Only model predicts *toothbrush* probably due to misunderstanding the straw in the cup. The optimal model combining both LSP and TSGCN would help balance the two streams, reducing missing labels and false positive as shown in the predicted labels for the third example image. As a result, LSP+TSGCN model achieves the best overall performance over the entire dataset.

3) Impact of Hyper-Parameters: In this subsection, we study the performance of our proposed method given different parameter settings. We vary the dimension of the pooled features (the

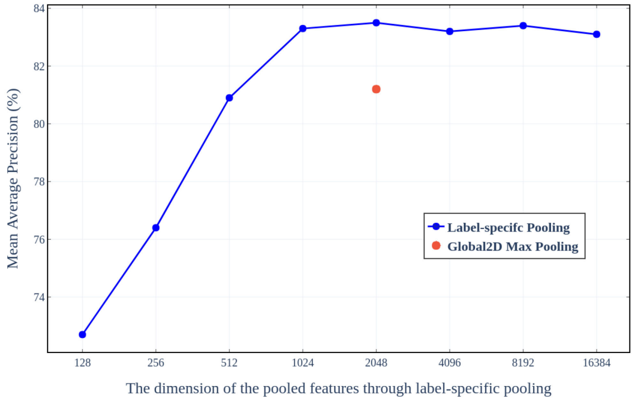


Fig. 6. Mean average precision on the MS-COCO dataset using label-specific features with different dimensions.

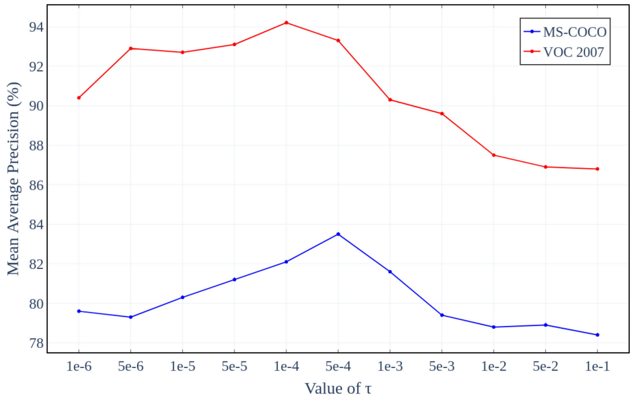


Fig. 7. Mean average precision on both datasets using different values of τ.

down-sampling ratio of the label-specific pooling) to investigate its impact. The different choices of parameter τ for the label-specific constraint have also been studied to obtain the optimal setting.

We first change the down-sampling ratio to explore how effective the label-specific pooling strategy retains critical information for classification. We carry out multi-label classification on the MS-COCO dataset using the pooled features of various dimensions, and report the results in Fig. 6. Compared with the conventional global max-pooling method generating 2,048-dimensional features, our label-specific pooling can generate more discriminative features with the same dimension. Moreover, label-specific pooling can achieve similar results at a higher down-sampling ratio (e.g., 512-dimensional pooled features). The performance decrease in term of mAP with very low dimensional features is inevitable due to the information loss in the pooling process.

The effects of using different label-specific constraint coefficient τ are shown in Fig. 7. Selecting different values of τ would affect the representation power of the label-specific features. It is noticed that either a large or small τ would lead to a decreased overall performance. The range between $1e^{-3}$ and $1e^{-4}$ would be suitable for both MS-COCO and PASCAL VOC. As MS-COCO contains 80 class labels while PASCAL VOC only has 20, a larger τ would be required in the case of MS-COCO for its more complicated label space to achieve an optimal performance.

V. Conclusion

In this paper, we present a novel deep learning framework for multi-label image classification, which jointly learns from both input and output spaces. Aiming for learning discriminative representation from the input space, we propose a novel pooling strategy named LSP. LSP is able to model label co-occurrence and generate discriminative features specific to each label. For the output space, we devise a Two-Stream Graph Convolutional Network (TSGCN) to train the multi-label classifiers with both label semantic graph and object spatial graph. Experimental results on two popular benchmark datasets demonstrate that our proposed model can achieve superior performance over the state-of-the-arts. In the future, we aim to integrate the label-specific pooling and object regions detection in a single network to achieve better performance in terms of both effectiveness and efficiency.

References

[1] X. Ding *et al.*, "Multi-instance multi-label learning combining hierarchical context and its application to image annotation," *IEEE Trans. Multimedia*, vol. 18, no. 8, pp. 1616–1627, Aug. 2016.

[2] S. Wang, J. Wang, Z. Wang, and Q. Ji, "Multiple emotion tagging for multimedia data by exploiting high-order dependencies among emotions," *IEEE Trans. Multimedia*, vol. 17, no. 12, pp. 2185–2197, Dec. 2015.

[3] M.-L. Zhang, J. M. Peña, and V. Robles, "Feature selection for multi-label naive Bayes classification," *Inf. Sci.*, vol. 179, no. 19, pp. 3218–3229, 2009.

[4] D. Kong, C. Ding, H. Huang, and H. Zhao, "Multi-label ReliefF and F-statistic feature selections for image annotation," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2012, pp. 2352–2359.

[5] P. Yan and Y. Li, "Graph-margin based multi-label feature selection," in *Proc. Joint Eur. Conf. Mach. Learn. Knowl. Discovery Databases*, 2016, pp. 540–555.

[6] M.-L. Zhang and L. Wu, "LIFT: Multi-label learning with label-specific features," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 1, pp. 107–120, Jan. 2015.

[7] J. Huang, G. Li, Q. Huang, and X. Wu, "Learning label specific features for multi-label classification," in *Proc. IEEE Int. Conf. Data Mining*, 2015, pp. 181–190.

[8] J. Huang, G. Li, Q. Huang, and X. Wu, "Joint feature selection and classification for multilabel learning," *IEEE Trans. Cybern.*, vol. 48, no. 3, pp. 876–889, Mar. 2018.

[9] Y. Yao and X. Zhang, "Class-specific attribute reducts in rough set theory," *Inf. Sci.*, vol. 418, pp. 601–618, 2017.

[10] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2016, pp. 770–778.

[11] Z.-M. Chen, X.-S. Wei, P. Wang, and Y. Guo, "Multi-label image recognition with graph convolutional networks," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2019, pp. 5177–5186.

[12] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proc. IEEE Int. Conf. Comput. Vision*, 2017, pp. 2961–2969.

[13] M. R. Boutell, J. Luo, X. Shen, and C. M. Brown, "Learning multi-label scene classification," *Pattern Recognit.*, vol. 37, no. 9, pp. 1757–1771, 2004.

[14] G. Tsoumakas, I. Katakis, and I. Vlahavas, "Mining multi-label data," in *Data Mining and Knowledge Discovery Handbook*. Berlin, Germany: Springer, 2009, pp. 667–685.

[15] M.-L. Zhang and Z.-H. Zhou, "ML-KNN: A lazy learning approach to multi-label learning," *Pattern Recognit.*, vol. 40, no. 7, pp. 2038–2048, 2007.

[16] C. Vens, J. Struyf, L. Schietgat, S. Džeroski, and H. Blockeel, "Decision trees for hierarchical multi-label classification," *Mach. Learn.*, vol. 73, no. 2, pp. 185–214, 2008.

[17] A. Elisseeff and J. Weston, "A kernel method for multi-labelled classification," in *Proc. Advances Neural Inf. Process. Syst.*, 2002, pp. 681–687.

[18] Q. Li, M. Qiao, W. Bian, and D. Tao, "Conditional graphical lasso for multi-label image classification," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2016, pp. 2977–2986.

[19] J. Wang *et al.*, "CNN-RNN: A unified framework for multi-label image classification," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2016, pp. 2285–2294.

[20] F. Zhu, H. Li, W. Ouyang, N. Yu, and X. Wang, "Learning spatial regularization with image-level supervisions for multi-label image classification," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2017, pp. 5513–5522.

[21] Y. Luo, M. Jiang, and Q. Zhao, "Visual attention in multi-label image classification," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2019, pp. 820–827.

[22] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2016, pp. 779–788.

[23] Z. Ma, F. Nie, Y. Yang, J. R. Uijlings, and N. Sebe, "Web image annotation via subspace-sparsity collaborated feature selection," *IEEE Trans. Multimedia*, vol. 14, no. 4, pp. 1021–1030, Aug. 2012.

[24] Z.-H. Zhou, M.-L. Zhang, S.-J. Huang, and Y.-F. Li, "Multi-instance multi-label learning," *Artif. Intell.*, vol. 176, no. 1, pp. 2291–2320, 2012.

[25] Z.-H. Zhou and M.-L. Zhang, "Multi-instance multi-label learning with application to scene classification," in *Proc. Advances Neural Inf. Process. Syst.*, 2007, pp. 1609–1616.

[26] S.-J. Huang, W. Gao, and Z.-H. Zhou, "Fast multi-instance multi-label learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, pp. 2614–2627, Nov. 2019.

[27] Y. Wei *et al.*, "HCP: A flexible CNN framework for multi-label image classification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 9, pp. 1901–1907, Sep. 2015.

[28] M. Wang, C. Luo, R. Hong, J. Tang, and J. Feng, "Beyond object proposals: Random crop pooling for multi-label image recognition," *IEEE Trans. Image Process.*, vol. 25, no. 12, pp. 5678–5688, Dec. 2016.

[29] S. Lu, Z. Wang, T. Mei, G. Guan, and D. D. Feng, "A bag-of-importance model with locality-constrained coding based feature learning for video summarization," *IEEE Trans. Multimedia*, vol. 16, no. 6, pp. 1497–1509, Oct. 2014.

[30] A. Jalali, S. Sanghavi, C. Ruan, and P. K. Ravikumar, "A dirty model for multi-task learning," in *Proc. Advances Neural Inf. Process. Syst.*, 2010, pp. 964–972.

[31] S. Kim, K.-A. Sohn, and E. P. Xing, "A multivariate regression approach to association analysis of a quantitative trait network," *Bioinformatics*, vol. 25, no. 12, pp. i204—-i212, 2009.

[32] S. Pan, J. Wu, X. Zhu, G. Long, and C. Zhang, "Task sensitive feature exploration and learning for multitask graph classification," *IEEE Trans. Cybern.*, vol. 47, no. 3, pp. 744–758, Mar. 2017.

[33] J. Read, B. Pfahringer, and G. Holmes, "Multi-label classification using ensembles of pruned sets," in *Proc. IEEE Int. Conf. Data Mining*, 2008, pp. 995–1000.

[34] Y. Gong, Y. Jia, T. Leung, A. Toshev, and S. Ioffe, "Deep convolutional ranking for multilabel image annotation," in *Proc. Int. Conf. Learn. Representations*, 2014.

[35] T. Chen, Z. Wang, G. Li, and L. Lin, "Recurrent attentional reinforcement learning for multi-label image recognition," in *Proc. AAAI Conf. Artif. Intell.*, 2018, pp. 6730–6737.

[36] J. Feng and Z.-H. Zhou, "Deep MIML network," in *Proc. AAAI Conf. Artif. Intell.*, 2017, pp. 1884–1890.

[37] D. Liu, S. Yan, X.-S. Hua, and H.-J. Zhang, "Image retagging using collaborative tag propagation," *IEEE Trans. Multimedia*, vol. 13, no. 4, pp. 702–712, Aug. 2011.

[38] X. Li, F. Zhao, and Y. Guo, "Multi-label image classification with a probabilistic label enhancement model," in *Proc. Conf. Uncertainty Artif. Intell.*, 2014, pp. 430–439.

[39] C.-W. Lee, W. Fang, C.-K. Yeh, and Y.-C. Frank Wang, "Multi-label zero-shot learning with structured knowledge graphs," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2018, pp. 1576–1585.

[40] J. Xu and Z.-H. Mao, "Multilabel feature extraction algorithm via maximizing approximated and symmetrized normalized cross-covariance operator," *IEEE Trans. Cybern.*, to be published, doi: 10.1109/TCYB.2019.2909779.

[41] S. Jiang *et al.*, "Multi-label metric transfer learning jointly considering instance space and label space distribution divergence," *IEEE Access*, vol. 7, pp. 10 362–10 373, 2019.

[42] Z. Wang, T. Chen, G. Li, R. Xu, and L. Lin, "Multi-label image recognition by recurrently discovering attentional regions," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2017, pp. 464–472.

[43] S.-F. Chen, Y.-C. Chen, C.-K. Yeh, and Y.-C. F. Wang, "Order-free RNN with visual attention for multi-label classification," in *Proc. AAAI Conf. Artif. Intell.*, 2018, pp. 6714–6721.

[44] W. Ge, S. Yang, and Y. Yu, "Multi-evidence filtering and fusion for multi-label classification, object detection and semantic segmentation based on weakly supervised learning," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2018, pp. 1277–1286.

[45] T.-Y. Lin *et al.*, "Microsoft COCO: Common objects in context," in *Proc. Eur. Conf. Comput. Vision*, 2014, pp. 740–755.

[46] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (VOC) challenge," *Int. J. Comput. Vision*, vol. 88, no. 2, pp. 303–338, 2010.

[47] J. Pennington, R. Socher, and C. Manning, "GloVe: Global vectors for word representation," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2014, pp. 1532–1543.

[48] J. Zhang, Q. Wu, C. Shen, J. Zhang, and J. Lu, "Multilabel image classification with regional latent semantic dependencies," *IEEE Trans. Multimedia*, vol. 20, no. 10, pp. 2801–2813, Oct. 2018.

[49] H. Yang *et al.*, "Exploit bounding box annotations for multi-label object recognition," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2016, pp. 280–288.