G-CAM: Graph Convolution Network Based Class Activation Mapping for Multi-label Image Recognition

Yangtao Wang[†], Yanzhao Xie[‡]*, Yu Liu[‡], Lisheng Fan[†]
[†]School of Computer Science and Cyber Engineering, Guangzhou University, Guangzhou, China
[‡]Huazhong University of Science and Technology, Wuhan, China
{ytaowang@gzhu.edu.cn, yzxie@hust.edu.cn, liu_yu@hust.edu.cn, lsfan@gzhu.edu.cn}
*Corresponding author: Yanzhao Xie (yzxie@hust.edu.cn)

ABSTRACT

In most multi-label image recognition tasks, human visual perception keeps consistent for different spatial transforms of the same image. Existing approaches either learn the perceptual consistency with only image-level supervision or preserve the middle-level feature consistency of attention regions but neglect the (global) label dependencies between different objects over the dataset. To address this issue, we integrate graph convolution network (GCN) and propose G-CAM, which learns visual attention consistency via GCN based class attention mapping (CAM) for multi-label image recognition. G-CAM consists of an image feature extraction module to generate the feature maps of the original image and its transformed one and a GCN module to learn weighted classifiers that capture the label dependencies between different objects. Different from previous works which use fully-connected classification layer, G-CAM first fuses weighted classifiers with the feature vector to generate the predicted labels for each input image, then combines weighted classifiers with the feature maps to respectively obtain the transformed attention heatmaps of the original image and the attention heatmaps of its transformed one. We can compute the attention consistency loss according to the distance between these two attention heatmaps. Finally, this loss is combined with the multi-label classification loss to update the whole network in an end-to-end manner. We conduct extensive experiments on three multi-label image datasets including FLICKR25K, MS-COCO and NUS-WIDE. Experimental results demonstrate G-CAM can achieve better performance compared with the state-of-the-art multi-label image recognition methods.

CCS CONCEPTS

• **Computing methodologies → Image representations**.

KEYWORDS

multi-label image recognition, graph convolutional network, class activation mapping

ICMR '21, August 21–24, 2021, Taipei, Taiwan

ACM ISBN 978-1-4503-8463-6/21/08…$15.00
https://doi.org/10.1145/3460426.3463620

ACM Reference Format:
Yangtao Wang[†], Yanzhao Xie[‡]*, Yu Liu[‡], Lisheng Fan[†]. 2021. G-CAM: Graph Convolution Network Based Class Activation Mapping for Multi-label Image Recognition. In *Proceedings of the 2021 International Conference on Multimedia Retrieval (ICMR '21), August 21–24, 2021, Taipei, Taiwan.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3460426.3463620

1 INTRODUCTION

In recent years, multi-label image recognition has aroused wide attention in computer vision community including various applications such as human attribution recognition [6, 25], multi-object recognition [15, 16], scene understanding [33], *etc.* With the rapid development of deep convolution neural network (CNN) [8, 12, 18, 34], the performance of multi-label image recognition has been greatly boosted in the past few years. However, it still poses a great challenge to multi-label images due to the appearance complexities, intra-label variation, unsatisfactory image qualities [11, 23, 24, 43], *etc.*

In most multi-label image recognition tasks, human visual perception keeps consistent for different spatial transforms (such as flipping, scaling, rotation and translation) of an image. Take Figure 1 for example, the flipped transform will not vary human recognition of "person" and "horse" (red attention regions) in the original image. Data augmentation methods [18, 19] utilize this consistency to increase the samples to train CNN classifiers, which construct new training data by assigning the same ground truth labels of each original image to its corresponding transformed one. However, these methods only learn the perceptual consistency of high-level features at the final classification layer with image-level supervision [28, 43] but neglect the related attention regions of human vision [27] and CNN models [32, 42]. To learn and preserve the middle-level feature consistency of attention regions under different transforms, Guo *et al.* [7] propose VACIT that first uses class activation mapping (CAM) [42] to calculate the attention heatmaps for each label, and then designs an attention consistency loss to limit the distance between the transformed attention heatmaps of the original image and the attention heatmaps of the transformed image. This work trains an effective classifier that well preserves the visual attention consistency under different transforms and achieves good results on multi-label images. Nevertheless, we find that this classifier only focuses on the objects inside an image but neglects the (global) label dependencies between different objects over the dataset. Take Figure 2 for example, from a global perspective, semantically related objects will be more likely to co-occur in an image, which means a large number of combination of labels will hardly occur

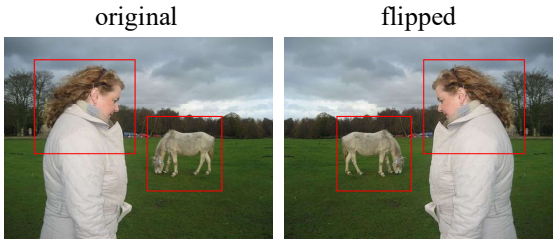


Figure 1: Human visual consistency for different transforms of an image.

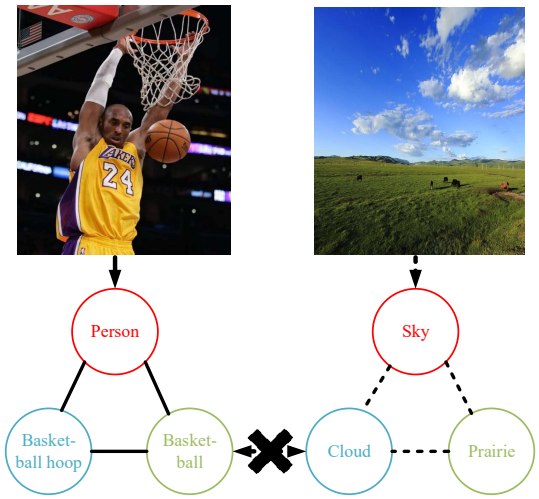


Figure 2: Label dependencies over the dataset. Commonly, "person", "basketball" and "basketball hoop" will co-occur in an image with a high possibility, and so will the combination of "sky", "prairie" and "cloud". However, we hardly see a "basketball" in the "cloud", because there is no direct relationship between these two objects.

in the real world. To model the label dependencies, ML-GCN [2] and A-GCN [22] utilize graph convolution network (GCN) [17, 21] to generate weighted classifiers to capture the label correlations for multi-label image recognition, which improve the precision of classification results. Therefore, we expect better performance on multi-label images by incorporating the label dependencies into visual attention consistency under different transforms.

In this paper, we propose G-CAM, which learns the visual attention consistency via GCN based CAM for multi-label image recognition. Our G-CAM consists of an image feature extraction module and a GCN module. First, the former uses two identical CNN [8] branches to respectively generate the feature maps of the original image and its transformed one while the latter adopts GCN to learn weighted classifiers that capture the label dependencies between different objects. Different from the original fully-connected (fc) classifier layer [7], we then use the dot product to fuse the weighted classifiers with the feature vector of each input image to obtain the predicted labels. At the same time, according to CAM, the feature maps and weighted classifiers (replacing the weights of the original fc layer) are used to generate the transformed attention heatmaps of the original image and the attention heatmaps of the transformed image. We can compute the attention consistency loss according to the distance between these two attention heatmaps. Finally, this loss is combined with the multi-label classification loss to update the whole network in an end-to-end manner. We conduct extensive experiments on three multi-label image datasets including FLICKR25K [13], MS-COCO [26] and NUS-WIDE [3]. Experimental results demonstrate G-CAM can achieve better performance compared with the state-of-the-art multi-label image recognition methods.

2 RELATED WORKS

2.1 Multi-label Image Recognition

Early multi-label image recognition works treat each label in isolation and divide this task into multiple binary classification tasks by training a classifier for each label. Razavian *et al.* [30] use the parameters pre-trained on ImageNet to obtain an SVM classifer to annotate each object for multi-label images. Gong *et al.* [5] propose to combine convolution architectures with an approximate top-k ranking objective function to recognize multi-label images. Different from these label-independent methods, others begin to explore the label correlations between different objects. Wang *et al.* [36] combine CNN with recurrent neural network (RNN) to model the dependencies in a sequential fashion by embedding semantic labels into vectors. Zhu *et al.* [43] propose SRN to generate an attention map for each label by exploiting both semantic and spatial relationships between labels with image-level supervisions. Wang *et al* [40] introduce a recurrent memorized-attention module, comprising a spatial transformer and a long short-term memory units to locate attentional regions and capture the label correlation. In addition to image-level representation learning approaches, Guo *et al.* [7] integrate CAM [42] to learn the middle-level representations to preserve the visual attention consistency under different transforms of an image. These works effectively learn the attention regions within an image but neglect the (global) label dependencies over the dataset. Furthermore, Chen *et al.* [2] propose to utilize GCN to learn the label co-occurrence relationship and achieve good results on multi-label images. Similarly, A-GCN [22] designs an adaptive graph to explore the label dependencies according to the label embeddings.

2.2 Graph Convolution Network

Graph convolution network (GCN) [17, 21] is a deep convolution learning paradigm for graph-structured data which integrates local node features and graph topology structure in convolution layers. A GCN layer takes the node features and the correlation matrix between nodes as input, and outputs new node features after convolution operation on graph. Formally, this process can be described as follows:

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

where H^l, W^l, a^l respectively denote the input features, weights, non-linear activation function of the l-th graph convolution layer and $\hat{A}$ denotes the normalized version of correlation matrix A:

$$\hat{A} = \tilde{D}^{-\frac{1}{2}}\tilde{A}\tilde{D}^{-\frac{1}{2}}, \tag{2}$$

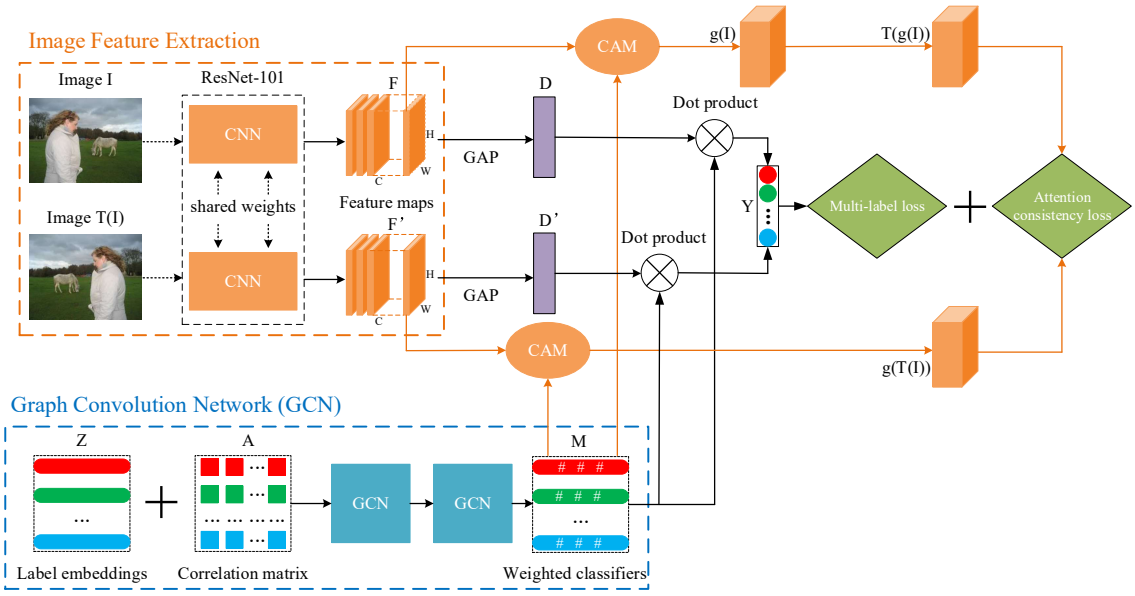


Figure 3: The overall framework of G-CAM.

where I is the identity matrix, $\tilde{A} = A + I$ and $\tilde{D}$ is a diagonal matrix with $\tilde{D}_{ii} = \sum_j \tilde{A}_{ij}$.

In recent years, GCN has been widely applied to relational feature extraction, node classification prediction and information retrieval tasks [10, 35, 41]. Wang *et al.* [37] propose to use GCN in a zero-shot image classification task. Rios *et al.* [31] combine GCN with attention mechanism for natural language classification task. More recently, GCN begins to play an important role in image generation, image classification, emotion learning, *etc.* Johnson *et al.* [14] adopt GCN to analyze scene-graphs with the application of image generation from scene graphs. ML-GCN [2] and A-GCN [22] use GCN to learn the correlation between objects and achieve good results on multi-label images. F-GCN [39] utilizes a cross-modal component to fuse image features and label embeddings, which speeds up the model convergence and achieve comparable results as ML-GCN and A-GCN. In addition, EmotionGCN [9] utilizes GCN to model the correlation between emotions for emotion distribution learning and NRDH [38] adopts GCN to learn the similarity between images for image retrieval.

3 PROPOSED METHODOLOGY

In this section, we elaborate our proposed G-CAM consisting of an image feature extraction module and a GCN module, which will be trained with a multi-label classification loss and an attention consistency loss in an end-to-end manner. The overall framework of G-CAM is shown in Figure 3. In the following, we describe the workflow of our approach in detail.

3.1 Image Feature Extraction Module

In this part, we use two identical CNN branches with shared weights to extract the feature maps of image I and its corresponding transformed one $I' = T(I)$, where T denotes the commonly-used transforms such as flipping, scaling, rotation and translation. As shown in the orange frame of Figure 3, following mainstream methods [2, 22], we employ ResNet-101 [8] to generate the feature maps F and $F' \in R^{C \times H \times W}$ respectively corresponding to I and I' from the "$conv5\_x$" layer of this network, where C, H and W respectively denote the number of channels, height and width of the feature maps. Note that we conduct the global average pooling (GAP) operation on F and F' to generate the feature vector D and $D' \in R^C$, which will be fused with the weighted classifiers to generate the predicted labels $Y \in R^L$ for each input image, where L denotes the number of object categories in a dataset.

3.2 Graph Convolution Network Module

In this part, we use GCN to model the label dependencies to generate the weighted classifiers that reflect the co-occurrence relationships between objects.

Different from the original GCN which was proposed to solve the node classification problem, we treat and design the node-level output of our G-CAM as a classifier corresponding to each label. Specifically, we aim to map the object dependencies of a dataset into label co-occurrence embeddings in our task. The input of GCN calls for the feature vector of each node and the correlation matrix between these nodes. As shown in the blue frame of Figure 3, we adopt the GloVe [29] model to transform each object (totally L object categories in a dataset) into a d-dimensional (*i.e.*, 300-dimensional) word vector. Therefore, we can obtain an $L \times d$ object word embeddings matrix Z.

In addition to obtaining the feature vector of each node (object), another essential issue in G-CAM is to construct the label correlation matrix A between these nodes. In the implementation, we capture the label dependencies and construct matrix A according to the label statistical information over the dataset. Specifically, for $\forall i \in [1, L]$, we collect the occurrence times (*i.e.*, Γ_i) of the i-th object (*i.e.*, o_i) as well as the co-occurrence times (*i.e.*, Γ_{ij}, which

equals Γ_{ji}) of o_i and o_j. Furthermore, the label dependencies can be formulated by the conditional probability as follows:

$$P_{ij} = P(o_i|o_j) = \frac{\Gamma_{ij}}{\Gamma_j}, \tag{3}$$

where P_{ij} denotes the probability that o_i occurs in the conditional of o_j appearing. Note that P_{ij} is not equal to P_{ji} owing that the conditional probability between two objects is asymmetric. Based on this, we can construct the correlation matrix A below:

$$A_{ij} = P_{ij}, \tag{4}$$

where A_{ij} denotes the i-th row and j-th column element of matrix A. However, similar to ML-GCN, if we directly use this correlation matrix to train the model, the rare co-occurrence objects will become some noise that affect the data distribution as well as the model convergence. To filter the noise, we choose to use a threshold ϵ to binarize the above matrix A:

$$A_{ij} = \begin{cases} 0, & \text{if } P_{ji} \leq \epsilon \\ 1, & \text{otherwise} \end{cases} \tag{5}$$

where $\epsilon \in [0, 1]$. Besides, when using GCN to update the node's feature in the propagation process, the binary correlation matrix may lead to the over-smoothing problem which makes the generated nodes' features indistinguishable. Therefore, we adopt the weighted scheme to calculate the final correlation matrix as:

$$A_{ij} = \begin{cases} \frac{\delta}{\sum_{j=1 \cap i \neq j}^{C}} A_{ij}, & \text{if } i \neq j \\ 1 - \delta, & \text{otherwise} \end{cases} \tag{6}$$

where $\delta \in [0, 1]$. In this way, we can use this weighted correlation matrix A to update the node's feature by choosing a suitable δ.

After obtaining both the object word embeddings vectors Z and label correlation matrix A, we design a two-layer GCN to propagate this relationship and each GCN layer can be described as:

$$Z^{l+1} = f^l(\hat{A}Z^lU^l), \quad l \in [0, 2] \tag{7}$$

where $\hat{A}$ (see Equation (2)) denotes the normalized version of correlation matrix A, Z^l denotes the latent features of L nodes in the l-layer, U^l denotes the weights of the l-layer, and $f^l(\cdot)$ denotes the non-linear operation which is a ReLU function. Note that Z is the input of this sub-network, and the output is an $L \times C$ weighted classifiers matrix M. On the one hand, M will be fused with the feature vector D (or D') via the dot product operation to generate the predicted labels Y for image I (or I'). On the other hand, according to CAM, M will be combined with feature maps F and F' to respectively compute the attention heatmaps for I and I'.

3.3 Loss Function

In stead of directly using fc layers to classify multiple labels, we take the (global) label dependencies into consideration and utilize the weighted classifiers M to simultaneously calculate the multi-label classification loss and preserve the middle-level attention consistency under different transforms.

Multi-label loss function. Given the feature vector D (or D') and weighted classifiers matrix M, we can obtain the predicted labels Y for image I (or I'):

$$Y = M \otimes D, \tag{8}$$

where $\otimes$ denotes the dot product operation. Given the ground truth labels $\hat{Y} \in \{0, 1\}^L$ of image I, we adopt the commonly-used multi-label classification loss function[1] to calculate this loss $\mathcal{L}_{\mathcal{M}}$ as follows:

$$\mathcal{L}_{\mathcal{M}} = \sum_{i=1}^{L} \hat{Y}^i log(Sigmoid(Y^i)) + (1 - \hat{Y}^i) log(1 - Sigmoid(Y^i)), \tag{9}$$

where $\hat{Y}^i$ denotes the i-th element of $\hat{Y}$ and $\hat{Y}^i = \{1, 0\}$ denotes whether the label i appears in the image I or not.

Attention consistency loss function. Once we replace the original fc layer with weighted classifiers M, each element of M is actually one of the partial derivatives (*i.e.*, $\frac{\partial Y}{\partial F}$) of the predicted labels Y with respect to the feature maps F. Given F, according to CAM [42], we compute the attention heatmaps Q by linearly weighted sum of all channels:

$$Q_j(m, n) = \sum_{k=1}^{C} M(j, k) F_k(m, n), \; j \in [1, L] \tag{10}$$

where $Q_j(m, n)$ denotes the attention heatmap at spatial location (m, n) for label j, $M(j, k)$ denotes the weight corresponding to label j for channel k of feature maps, and $F_k(m, n)$ denotes the feature maps of channel k from the "$conv5\_x$" layer at spatial location (m, n). Note that, for image I and $T(I)$, we use $Q = g(I)$ and $Q' = g(T(I))$ to represent their attention heatmaps with size $H \times W$.

Specifically, by reshaping F, F' into shape $1 \times C \times H \times W$, and M into shape $L \times C \times 1 \times 1$, we conduct channel-wise multiplication to linearly combine feature maps for each label, and sum along dimension C of combined feature maps. Thus, $Q = g(I)$ and $Q' = g(T(I))$ are both in shape $L \times H \times W$. Note that the attention heatmaps of the original I and its transformed $T(I)$ need to be equivariant under the given transform T, which means:

$$T(g(I)) = g(T(I)). \tag{11}$$

Therefore, to preserve the visual attention consistency, we set the attention consistency loss $\mathcal{L}_{\mathcal{A}}$ between the transformed heatmaps $\hat{Q} = T(Q) = T(g(I))$ of the original image I and the heatmaps Q' of the transformed image I' as follows:

$$\mathcal{L}_{\mathcal{A}} = \frac{1}{LHW} \sum_{j=1}^{L} \|\hat{Q}_j - Q'_j\|_2, \tag{12}$$

where Q_j denotes the attention heatmap for label j.

At last, we combine this attention consistency loss with the multi-label classification loss to update the network in an end-to-end manner:

$$\mathcal{L} = (1 - \lambda)\mathcal{L}_{\mathcal{M}} + \lambda \mathcal{L}_{\mathcal{A}}, \tag{13}$$

where λ is a factor to balance these two losses. Note that we can embed more than one transforms to train this network. For example, if we use both T_1 (flipping) and T_2 (scaling) transforms, the attention consistency loss will be $\mathcal{L}_{\mathcal{A}} = \mathcal{L}_{\mathcal{A}}{}^{T_1} + \mathcal{L}_{\mathcal{A}}{}^{T_2}$.

[1] https://pytorch.org/docs/master/nn.html?highlight=multilabelsoft#torch.nn.MultiLabelSoftMarginLoss

4 EXPERIMENTS

4.1 Datasets

FLICKR25K [13] is a collection of 25,000 multi-label images belonging to 24 unique provided labels, and each image is annotated by 4.7 labels on average. We randomly split the train and test sets with a ratio of 5:5, and evaluate the performance on the test set.

MS-COCO [26] is a popular multi-label dataset for image recognition, segmentation and captioning, which contains 82,783 training images, 40,504 validation images and 40,775 test images, where each image is labeled by some of the 80 semantic concepts except that the labels of test set are not available. Owing that the ground truth labels of the test set are not available, we train our model on the train set and evaluate the performance on the validation set.

NUS-WIDE [3] consists of 269,648 multi-label images with 161,789 training images and 107,859 test images, where each image is annotated with multiple labels based on 81 concepts. We use the train set to train our model, and then evaluate the performance on the test set.

4.2 Implementation Details and Evaluation Metrics

Implementation details. All experiments are conducted with PyTorch. In the image feature extraction module, we adopt ResNet-101 [8] to generate the feature maps for each input image with two transforms: flipping and scaling. Each image is resized into two scales: 224×224 and 192×192. In the GCN module, following ML-GCN [2], we set $\epsilon = 0.4$ (see Equation (5)) and $\delta = 0.2$ (see Equation (6)) to construct the correlation matrix, which will be input to a two-layer GCN with the output dimension of 1024 and 2048. In the loss function $\mathcal{L}$ (see Equation (13)), we set $\lambda = 0.4$. Our model will be trained using stochastic gradient descent (SGD) with a momentum of 0.9, a weight decay of 5×10^{-4}, an initial rate of 10^{-3} which decays by a factor of 2 every 30 epochs, and a batchsize of 32. Note that in the testing process, we will remove the network branch of the transformed image.

Evaluation metrics. We report two important evaluation metrics: the average precision (AP) for each category and mean average precision (mAP) over all categories. In addition, we also record the per-class precision (CP), per-class recall (CR), per-class F1 (C-F1) and the overall precision (OP), overall recall (OR), overall F1 (O-F1). For fair comparisons, we further list these results on top-3 labels.

4.3 Experimental Results

4.3.1 Comparisons with the state-of-art methods. In this section, we record and analyze the experimental results of G-CAM compared with the state-of-the-art multi-label image recognition methods on FLICKR25K, MS-COCO and NUS-WIDE.

Results on FLICKR25K. We compare the performance of G-CAM with ML-GCN [2], A-GCN [22] and VACIT [7] on FLICKR25K. For fair comparisons, we employ the same settings including feature extraction network (ResNet-101), SGD, batchsize, learning rate on these methods. As shown in Table 1 which lists mAP, CP, CR, CF1, OP, OR, OF1 values, except for a lower result on CR and OR than A-GCN, G-CAM outperforms others on all other metrics. Especially for mAP, G-CAM exceeds the state-of-the-art methods by 1.0%-1.5%. In addition, we also record the loss trend and AP results in Figure 4 and Figure 5 . From Figure 4, with the increase of iterations, G-CAM can faster converge to a lower value than VACIT. As for AP values, according to Figure 5, except for a slightly lower result on "lake", G-CAM outperforms VACIT on the remaining 23 objects with a 3.7% higher AP on "bird" and an average 1% higher result (*i.e.*, mAP) on each object. All these results reflect that our G-CAM integrates the global label dependencies into CAM and can better recognize multi-label images.

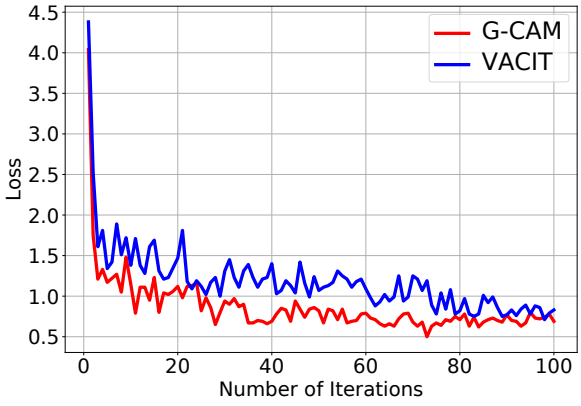


Figure 4: Loss trend with the increase of iterations on FLICKR25K.

Results on MS-COCO. We compare the performance of G-CAM with the state-of-the-art methods including CNN-RNN [36], RNN-Attention [40], Order-Free RNN [1], ML-ZSL [20], SRN [43], Multi-Evidence [4], ResNet-101 [8], ML-GCN [2], A-GCN [22] and VACIT [7] on MS-COCO. As shown in Table 2, G-CAM outperforms other candidates by at least 1% in terms of mAP. As for other metrics, on all labels, G-CAM is inferior to ML-GCN on CP and OP, but outperforms all other methods on all metrics. In addition, on top-3 labels, G-CAM exceeds others on CR, OR and OF1, but produces a relatively lower result on CP, CF1 and OP than ML-GCN. Note that VACIT obtains lower results than both ML-GCN and A-GCN on this dataset. Despite that there are some lower results compared with ML-GCN or A-GCN, we integrate the global label dependencies into visual attention consistency (VACIT) and greatly promote the image recognition performance on most metrics. Table 2 verifies the effectiveness of our method.

Results on NUS-WIDE. We compare the performance of G-CAM with the state-of-the art methods including KNN [3], WARP [5], CNN-RNN [36], ResNet-101 [8], SRN [43], ML-GCN [2], A-GCN [22] and VACIT [7] on NUS-WIDE. As shown in Table 3, except for a lower result on CR, OR on all labels, G-CAM outperforms all other methods on the remaining metrics including a 0.8% higher mAP than VACIT, 1.2% higher mAP than both ML-GCN and A-GCN. As for those results on top-3 labels, G-CAM almost achieves the highest values on all metrics except for a slightly 0.1% lower result on CR than A-GCN. On the whole, G-CAM nearly produces better results in all cases, which brings optimal multi-label image classification

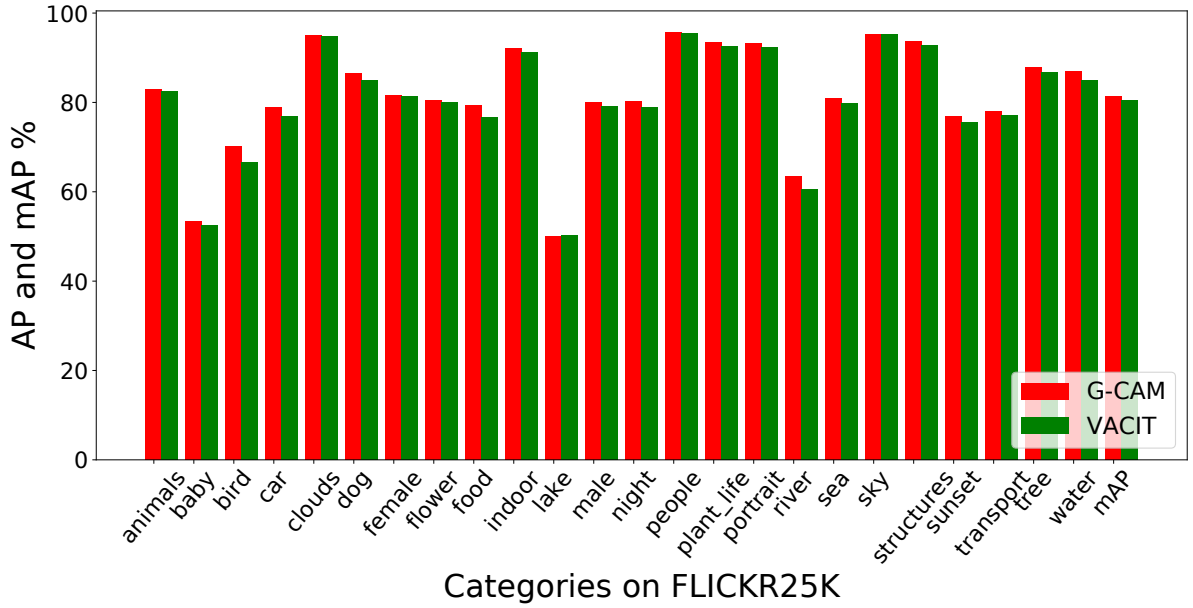


Figure 5: AP and mAP results on FLICKR25K.

Table 1: Performance (%) comparisons of G-CAM with the state-of-the-art methods on FLICKR25K.

Method	All							Top-3					
	mAP	CP	CR	CF1	OP	OR	OF1	CP	CR	CF1	OP	OR	OF1
ML-GCN	79.9	79.2	67.9	73.1	83.5	74.8	78.9	85.1	49.3	62.4	88.6	57.1	69.4
A-GCN	80.0	79.1	**68.0**	73.1	83.3	**75.1**	79.0	85.1	**49.4**	62.5	88.7	57.2	69.5
VACIT	80.4	83.6	64.9	72.5	86.8	73.3	79.5	85.7	48.9	62.3	89.2	57.6	70.0
G-CAM	**81.4**	**84.7**	65.4	**73.3**	**87.5**	73.5	**80.0**	**86.8**	**49.4**	**63.0**	**90.0**	**58.6**	**71.0**

Table 2: Performance (%) comparisons of G-CAM with the state-of-the-art methods on MS-COCO.

Method	All							Top-3					
	mAP	CP	CR	CF1	OP	OR	OF1	CP	CR	CF1	OP	OR	OF1
CNN-RNN	61.2	-	-	-	-	-	-	66.0	55.6	60.4	69.2	66.4	67.8
RNN-Attention	-	-	-	-	-	-	-	79.1	58.7	67.4	84.0	63.0	72.0
Order-Free RNN	-	-	-	-	-	-	-	71.6	54.8	62.1	74.2	62.2	67.7
ML-ZSL	-	-	-	-	-	-	-	74.1	64.5	69.0	-	-	-
SRN	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
Multi-Evidence	-	80.4	70.2	74.9	85.2	72.5	78.4	84.5	62.2	70.6	89.1	64.3	74.7
ResNet-101	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
ML-GCN	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
A-GCN	83.1	84.7	72.3	78.0	85.6	75.5	80.3	89.0	64.2	**74.6**	**90.5**	66.3	76.6
VACIT	77.5	77.4	68.3	72.2	79.8	73.1	76.3	85.2	59.4	68.0	86.6	63.3	73.1
G-CAM	**84.1**	81.6	**75.1**	**78.2**	83.4	**78.1**	**80.7**	85.1	**65.1**	73.7	90.0	**67.1**	**76.9**

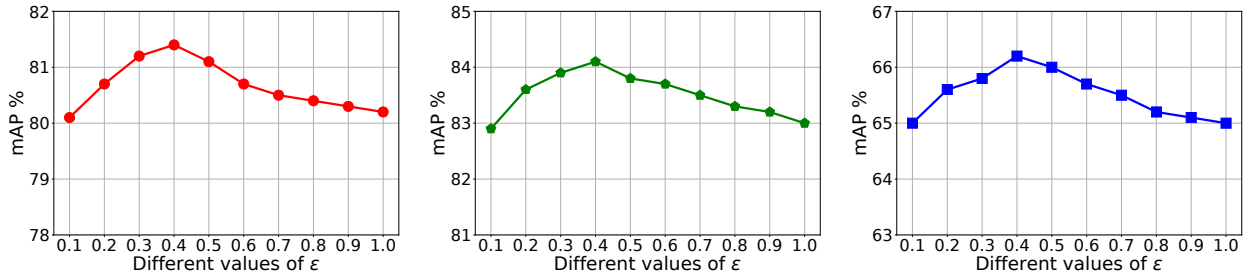
effect by integrating the global label dependencies to preserve the visual attention consistency under different transforms.

4.3.2 Ablation studies. In this section, we perform ablation studies to first observe the performance change by removing GCN or CAM component in GAM, then analyze the influence of key parameter settings on our model including ϵ (see Equation (5)) in the correlation matrix construction, δ (see Equation (6)) in the weighted correlation matrix and λ (see Equation (13)) in the loss function.

GCN and CAM components. In this part, we remove the key components including GCN and CAM to explore the performance change of G-CAM on FLICKR25K, MS-COCO and NUS-WIDE. As

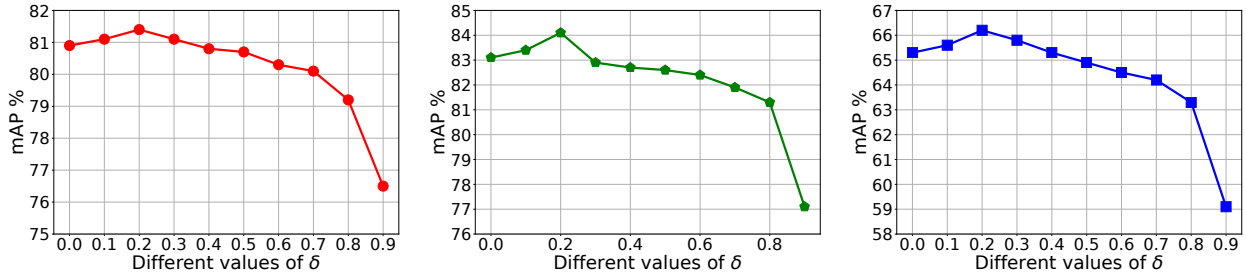
Table 3: Performance (%) comparisons of G-CAM with the state-of-the-art methods on NUS-WIDE.

Method	All							Top-3					
	mAP	CP	CR	CF1	OP	OR	OF1	CP	CR	CF1	OP	OR	OF1
KNN	-	-	-	-	-	-	-	32.6	19.3	24.3	42.9	53.4	47.6
WARP	-	-	-	-	-	-	-	31.7	35.6	33.5	48.6	60.5	53.9
CNN-RNN	-	-	-	-	-	-	-	40.5	30.4	34.7	49.9	61.7	55.2
ResNet-101	59.8	65.8	51.9	55.7	75.9	69.5	72.5	46.9	56.8	47.0	55.8	69.1	61.7
SRN	62.0	65.2	55.8	58.5	75.5	71.5	73.4	48.2	58.9	48.9	56.2	69.6	62.2
ML-GCN	64.9	70.9	58.0	63.8	80.1	73.6	76.7	51.2	61.5	55.8	56.7	73.3	63.9
A-GCN	65.0	70.7	**58.2**	63.8	80.0	**73.7**	76.7	51.1	**61.7**	55.9	56.7	73.3	63.9
VACIT	65.4	73.3	56.7	63.9	82.2	73.1	77.4	51.7	61.3	56.1	57.3	73.7	64.5
G-CAM	**66.2**	**73.7**	57.3	**64.5**	**82.8**	73.2	**77.7**	**52.3**	61.6	**56.6**	**57.5**	**74.2**	**64.8**



(a) FLICKR25K (b) MS-COCO (c) NUS-WIDE

Figure 6: The change of mAP using different values of ϵ.



(a) FLICKR25K (b) MS-COCO (c) NUS-WIDE

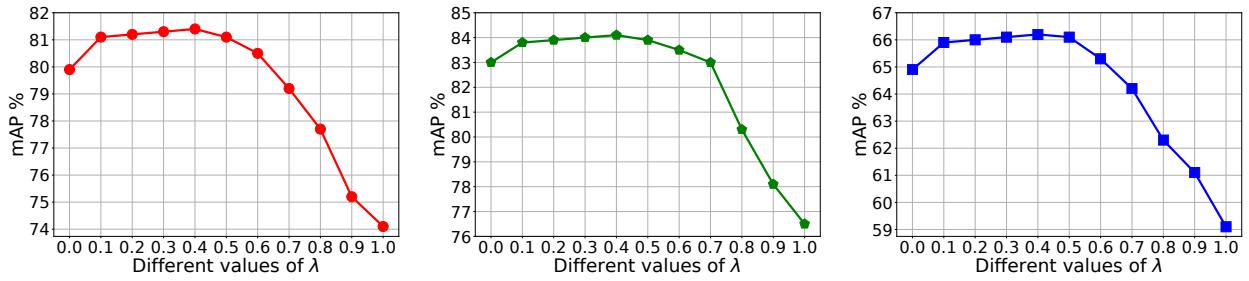
Figure 7: The change of mAP using different values of δ.

Table 4: Performance change with/without GCN and CAM.

Datasets	G-CAM			Metric
	without GCN	without CAM	with GCN and CAM	
FLICKR25K	80.4	79.9	**81.4**	
MS-COCO	77.5	83.0	**84.1**	mAP
NUS-WIDE	65.4	64.9	**66.2**	

shown in Table 4, G-CAM can achieve the highest mAP on these three datasets if keeping GCN and CAM components. However, once we remove GCN or CAM, the performance of G-CAM will directly drop to a certain extent. This phenomenon reflects that GCN and CAM simultaneously play an important role by integrating the global label dependencies into image visual consistency to promote the performance of G-CAM. Therefore, this result also verifies the feasibility and effectiveness of our method.

Parameter ϵ in the correlation matrix construction. In this part, we evaluate the performance of G-CAM by using different values of ϵ to construct the correlation matrix. Parameter ϵ is used to filter the noise data (rare co-occurrence probability) to balance the label dependencies. Note that if we retain all the edges ($\epsilon = 0$), the model will hardly converge. As shown in Figure 6, we vary ϵ from 0.1 to 1 to observe the effect and find that G-CAM achieves the highest mAP on all the three datasets when setting $\epsilon = 0.4$. As we see, a large ϵ will cut off too many edges, thus decreasing the performance. We believe $\epsilon = 0.4$ not only reduces the small-probability data points but also well preserves the correlation between objects.



(a) FLICKR25K (b) MS-COCO (c) NUS-WIDE

Figure 8: The change of mAP using different values of λ.

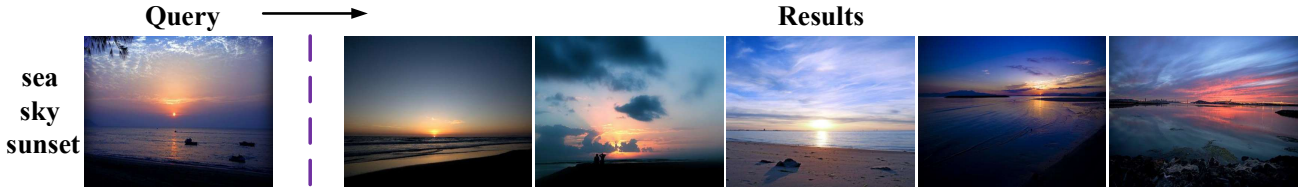


Figure 9: The top-5 retrieval results on FLICKR25K.

Parameters δ in the weighted correlation matrix. As we mentioned, δ is used to avoid the over-smoothing problem which may make the generated nodes' features indistinguishable. We vary δ from 0 to 1 to observe the effect and find G-CAM achieves the highest mAP on all the three datasets when $\delta = 0.2$. Note that G-CAM will not converge when setting $\delta = 1$. If we use a large δ, the feature of the node itself may be ignored in the propagation process. Otherwise, a too small δ will make G-CAM ignore the correlation between a node with its neighbor nodes. Figure 7 demonstrates $\delta = 0.2$ can well balance this correlation on the graph.

Parameter λ in loss function. When training our model, we combine the multi-label loss $\mathcal{L}_{\mathcal{M}}$ and attention consistency loss $\mathcal{L}_{\mathcal{A}}$ to update the network. The weighted factor λ indicates the contribution of $\mathcal{L}_{\mathcal{A}}$ in the whole loss $\mathcal{L}$. In this part, we vary λ from 0 to 1 and record the change of mAP on VOC2007, MS-COCO and FLICKR25K in Figure 8. In fact, when setting $\lambda = 0$, G-GAM will be transformed into ML-GCN, which cannot preserve the visual attention consistency. With the increase of λ, the performance of G-CAM will be promoted, and it can achieve the highest value on these three datasets with $\lambda = 0.4$. This setting well balances global label dependencies and visual attention consistency, which helps obtain the best classification results. However, when λ exceeds 0.4, the performance will drop rapidly, which means that only using attention consistency loss without considering label dependencies cannot achieve the desired result. It is the balance between these two factors that plays a significant role in G-CAM. Both $\mathcal{L}_{\mathcal{M}}$ and $\mathcal{L}_{\mathcal{A}}$ will contribute to achieving better multi-label image classification performance.

4.3.3 Visual retrieval results. In this section, we evaluate G-CAM by illustrating the retrieval results on FLICKR25K. We return the top-5 images by the KNN algorithm for the given query image. As shown in Figure 9, we randomly choose an input image that contains three objects: "sea", "sky" and "sunset". Obviously, the returned 5 images also contain these three objects. The visual retrieval results verify that G-CAM owns a good classification ability to recognize multi-label images.

5 CONCLUSION

In this paper, we propose G-CAM, which learns visual attention consistency via GCN based CAM for multi-label image recognition. G-CAM consists of an image feature extraction module to generate the feature maps of the original image and its transformed one and a GCN module to learn weighted classifiers that capture the label dependencies between different objects. Instead of using the fc classification layer, G-CAM first fuses weighted classifiers with the feature vector to generate the predicted labels for each input image, then combines weighted classifiers with the feature maps to respectively obtain the transformed attention heatmaps of the original image and the attention heatmaps of its transformed one. Our model is trained by the combination of multi-label classification loss and attention consistency loss in an end-to-end manner. Extensive experiments on FLICKR25K, MS-COCO and NUS-WIDE demonstrate G-CAM can achieve better performance compared with the state-of-the-art multi-label image recognition methods. In addition, the ablation studies explore how the key components and parameter settings influence the performance of G-CAM. At last, the visual retrieval results reflect G-CAM can well recognize multi-label images.

ACKNOWLEDGMENTS

This work is supported by the National Natural Science Foundation of China (No.61902135) and the International Science and Technology Cooperation Projects of Guangdong Province (No.2020A050510 0060). Thanks for Jay Chou, a celebrated Chinese singer whose songs have been accompanying the author.

REFERENCES

[1] Shang-Fu Chen, Yi-Chen Chen, Chih-Kuan Yeh, and Yu-Chiang Frank Wang. 2018. Order-Free RNN With Visual Attention for Multi-Label Classification. In *AAAI 2018*.

[2] Zhao-Min Chen, Xiu-Shen Wei, Peng Wang, and Yanwen Guo. 2019. Multi-Label Image Recognition With Graph Convolutional Networks. In *CVPR 2019*.

[3] Tat-Seng Chua, Jinhui Tang, Richang Hong, Haojie Li, Zhiping Luo, and Yantao Zheng. 2009. NUS-WIDE: a real-world web image database from National University of Singapore. In *CIVR 2009*.

[4] Weifeng Ge, Sibei Yang, and Yizhou Yu. 2018. Multi-Evidence Filtering and Fusion for Multi-Label Classification, Object Detection and Semantic Segmentation Based on Weakly Supervised Learning. In *CVPR 2018*.

[5] Yunchao Gong, Yangqing Jia, Thomas Leung, Alexander Toshev, and Sergey Ioffe. 2014. Deep Convolutional Ranking for Multilabel Image Annotation. In *ICLR 2014*.

[6] Hao Guo, Xiaochuan Fan, and Song Wang. 2017. Human attribute recognition by refining attention heat map. *Pattern Recognit. Lett.* 94 (2017), 38–45.

[7] Hao Guo, Kang Zheng, Xiaochuan Fan, Hongkai Yu, and Song Wang. 2019. Visual Attention Consistency Under Image Transforms for Multi-Label Image Classification. In *CVPR 2019*.

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep Residual Learning for Image Recognition. In *CVPR 2016*.

[9] Tao He and Xiaoming Jin. 2019. Image Emotion Distribution Learning with Graph Convolutional Networks. In *ICMR 2019*.

[10] Fenyu Hu, Yanqiao Zhu, Shu Wu, Liang Wang, and Tieniu Tan. 2019. Hierarchical Graph Convolutional Networks for Semi-supervised Node Classification. In *IJCAI 2019*.

[11] Hexiang Hu, Guang-Tong Zhou, Zhiwei Deng, Zicheng Liao, and Greg Mori. 2016. Learning Structured Inference Neural Networks with Label Relations. In *CVPR 2016*.

[12] Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. 2017. Densely Connected Convolutional Networks. In *CVPR 2017*.

[13] Mark J. Huiskes and Michael S. Lew. 2008. The MIR flickr retrieval evaluation. In *MIR 2008*.

[14] Justin Johnson, Agrim Gupta, and Li Fei-Fei. 2018. Image Generation From Scene Graphs. In *CVPR 2018*.

[15] Kai Kang, Hongsheng Li, Junjie Yan, Xingyu Zeng, Bin Yang, Tong Xiao, Cong Zhang, Zhe Wang, Ruohui Wang, Xiaogang Wang, and Wanli Ouyang. 2018. T-CNN: Tubelets With Convolutional Neural Networks for Object Detection From Videos. *IEEE Trans. Circuits Syst. Video Techn.* 28, 10 (2018), 2896–2907.

[16] Kai Kang, Wanli Ouyang, Hongsheng Li, and Xiaogang Wang. 2016. Object Detection from Video Tubelets with Convolutional Neural Networks. In *CVPR 2016*.

[17] Thomas N. Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *ICLR 2017*.

[18] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. 2012. ImageNet Classification with Deep Convolutional Neural Networks. In *NIPS 2012*.

[19] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. 2017. ImageNet classification with deep convolutional neural networks. *Commun. ACM* 60, 6 (2017), 84–90.

[20] Chung-Wei Lee, Wei Fang, Chih-Kuan Yeh, and Yu-Chiang Frank Wang. 2018. Multi-Label Zero-Shot Learning With Structured Knowledge Graphs. In *CVPR 2018*.

[21] Ron Levie, Federico Monti, Xavier Bresson, and Michael M. Bronstein. 2019. CayleyNets: Graph Convolutional Neural Networks With Complex Rational Spectral Filters. *IEEE Trans. Signal Process.* 67, 1 (2019), 97–109.

[22] Qing Li, Xiaojiang Peng, Yu Qiao, and Qiang Peng. 2019. Learning Category Correlations for Multi-label Image Recognition with Graph Networks. *CoRR* abs/1909.13005 (2019).

[23] Qiang Li, Maoying Qiao, Wei Bian, and Dacheng Tao. 2016. Conditional Graphical Lasso for Multi-label Image Classification. In *CVPR 2016*.

[24] Xin Li, Feipeng Zhao, and Yuhong Guo. 2014. Multi-label Image Classification with A Probabilistic Label Enhancement Model. In *UAI 2014*.

[25] Yining Li, Chen Huang, Chen Change Loy, and Xiaoou Tang. 2016. Human Attribute Recognition by Deep Hierarchical Contexts. In *ECCV 2016*.

[26] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. 2014. Microsoft COCO: Common Objects in Context. In *ECCV 2014*.

[27] George R. Mangun. 2007. Neural mechanisms of visual selective attention. *Psychophysiology* 32, 1 (2007), 4–18.

[28] Maxime Oquab, Léon Bottou, Ivan Laptev, and Josef Sivic. 2015. Is object localization for free? - Weakly-supervised learning with convolutional neural networks. In *CVPR 2015*.

[29] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global Vectors for Word Representation. In *EMNLP 2014*.

[30] Ali Sharif Razavian, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson. 2014. CNN Features Off-the-Shelf: An Astounding Baseline for Recognition. In *CVPR Workshops 2014*.

[31] Anthony Rios and Ramakanth Kavuluru. 2018. Few-Shot and Zero-Shot Multi-Label Learning for Structured Label Spaces. In *EMNLP 2018*.

[32] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. 2020. Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization. *Int. J. Comput. Vis.* 128, 2 (2020), 336–359.

[33] Jing Shao, Kai Kang, Chen Change Loy, and Xiaogang Wang. 2015. Deeply learned attributes for crowded scene understanding. In *CVPR 2015*.

[34] Karen Simonyan and Andrew Zisserman. 2015. Very Deep Convolutional Networks for Large-Scale Image Recognition. In *ICLR 2015*.

[35] Jiaxiang Tang, Wei Hu, Xiang Gao, and Zongming Guo. 2019. Joint Learning of Graph Representation and Node Features in Graph Convolutional Neural Networks. *CoRR* abs/1909.04931 (2019).

[36] Jiang Wang, Yi Yang, Junhua Mao, Zhiheng Huang, Chang Huang, and Wei Xu. 2016. CNN-RNN: A Unified Framework for Multi-label Image Classification. In *CVPR 2016*.

[37] Xiaolong Wang, Yufei Ye, and Abhinav Gupta. 2018. Zero-Shot Recognition via Semantic Embeddings and Knowledge Graphs. In *CVPR 2018*.

[38] Yangtao Wang, Jingkuan Song, Ke Zhou, and Yu Liu. 2021. Unsupervised deep hashing with node representation for image retrieval. *Pattern Recognit.* 112 (2021), 107785.

[39] Yangtao Wang, Yanzhao Xie, Yu Liu, Ke Zhou, and Xiaocui Li. 2020. Fast Graph Convolution Network Based Multi-label Image Recognition via Cross-modal Fusion. In *CIKM 2020*.

[40] Zhouxia Wang, Tianshui Chen, Guanbin Li, Ruijia Xu, and Liang Lin. 2017. Multi-label Image Recognition by Recurrently Discovering Attentional Regions. In *ICCV 2017*.

[41] Liang Yao, Chengsheng Mao, and Yuan Luo. 2019. Graph Convolutional Networks for Text Classification. In *AAAI 2019*.

[42] Bolei Zhou, Aditya Khosla, Àgata Lapedriza, Aude Oliva, and Antonio Torralba. 2016. Learning Deep Features for Discriminative Localization. In *CVPR 2016*.

[43] Feng Zhu, Hongsheng Li, Wanli Ouyang, Nenghai Yu, and Xiaogang Wang. 2017. Learning Spatial Regularization with Image-Level Supervisions for Multi-label Image Classification. In *CVPR 2017*.