Computers, Materials & Continua

Tech Science Press

DOI: 10.32604/cmc.2024.053938

ARTICLE

Multi-Label Image Classification Based on Object Detection and Dynamic Graph Convolutional Networks

Xiaoyu Liu and Yong Hu*

School of Cyber Science and Engineering, Sichuan University, Chengdu, 610207, China

*Corresponding Author: Yong Hu. Email: huyong@scu.edn.cn

Received: 14 May 2024 Accepted: 01 August 2024 Published: 12 September 2024

ABSTRACT

Multi-label image classification is recognized as an important task within the field of computer vision, a discipline that has experienced a significant escalation in research endeavors in recent years. The widespread adoption of convolutional neural networks (CNNs) has catalyzed the remarkable success of architectures such as ResNet-101 within the domain of image classification. However, in multi-label image classification tasks, it is crucial to consider the correlation between labels. In order to improve the accuracy and performance of multi-label classification and fully combine visual and semantic features, many existing studies use graph convolutional networks (GCN) for modeling. Object detection and multi-label image classification exhibit a degree of conceptual overlap; however, the integration of these two tasks within a unified framework has been relatively underexplored in the existing literature. In this paper, we come up with Object-GCN framework, a model combining object detection network YOLOv5 and graph convolutional network, and we carry out a thorough experimental analysis using a range of well-established public datasets. The designed framework Object-GCN achieves significantly better performance than existing studies in public datasets COCO2014, VOC2007, VOC2012. The final results achieved are 86.9%, 96.7%, and 96.3% mean Average Precision (mAP) across the three datasets.

KEYWORDS

Deep learning; multi-label image recognition; object detection; graph convolution networks

1 Introduction

With the continuous and vigorous evolution of deep learning networks, the accuracy of single-label image classification has been significantly improved. However, in reality, the great mass of images often includes multiple objects, and the richness of semantic information and the coexistence of higher-order labels make the classification of multi-label images more necessary [1]. Fig. 1 illustrates some images with multiple categories.

Understanding the interdependencies among labels is crucial in the context of multi-label classification. Recognizing the relationships between labels can significantly enhance the predictive accuracy of classification models. Certain labels exhibit semantic relatedness, as exemplified by the likelihood that an image labeled with both “beach” and “summer” would depict a summer beach scene. Incorporating this semantic association can aid models in discerning the underlying relationships within images. Conversely, there are instances of mutual exclusivity between labels, such as the impossibility of an image being simultaneously labeled as “cow” and “horse”. Accounting for such exclusivity can prevent the model from generating erroneous classifications. Collectively, these aspects of label correlation can profoundly impact the classification accuracy of the employed framework.

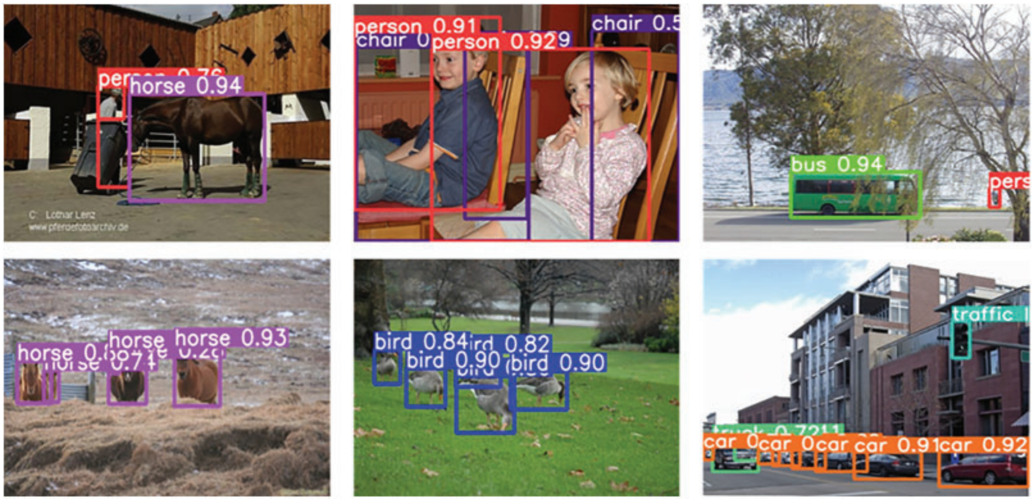

Figure 1: Images with multiple labels

Multi-label image recognition is a fundamental work in the sphere of computer vision, which can help humans have a more comprehensive understanding of the content of images, and plays a crucial role in many applications such as human attribute recognition [2], medical image recognition [3], and recommendation systems [4]. In contrast to single-label image classification, which assigns a single class to each image, multi-label image classification requires the assignment of multiple labels to a single image. This approach necessitates a comprehensive consideration of the interrelationships among labels to refine the accuracy of the classification process.

The original framework transformed the multi-label classification issue into a binary classification problem. Reference [5] approaches binary classification by training individual classifiers for each label, allowing for predictions to be made for every label separately. However, this strategy ignores dependencies between individual labels, since then, many studies have proposed a series of improvements to the dependence between labels. The studies cited in References [6,7] conceptualize the label prediction challenge as a pairwise prediction issue, whereas the works presented in References [8,9] propose an embedding of label vectors into a latent space to explore and learn the inherent correlations among them. With the development of recurrent neural networks, RNN (Recurrent Neural Network) is also widely used to learn correlations between labels. As a result, an array of RNN-based models has been introduced to the field, each designed to enhance the classification process by leveraging these learned correlations. Reference [10] proposes a CNN-RNN framework to learn the image-label joint embedding vector, so as to learn the dependency of semantic labels and the correlation of multiple labels in an image. Building upon the foundational work presented in Reference [10], the study in Reference [11] has enhanced the model by integrating a dynamic attention mechanism. This innovation is designed to direct the model’s focus towards salient regions within the image. However, the methods used in these papers do not adequately incorporate spatial and contextual information of objects.

Graph Convolutional Network (GCN) has a powerful ability to model the nodes of a graph [12], so it has been diffusely used in the sphere of multi-label image recognition since it was proposed. In the multi-label image recognition process based on GCN, formally, we represent the topological graph as $G = (V, E, A)$, where $V = \{v1, v2, \ldots, vc\}$ represents all label nodes, E represents the edges of label nodes, and $A \in R^{C\times C}$ represents the adjacency matrix of the constructed graph. To mine the global dependencies between labels, we need to construct a topological graph adjacency matrix A between label nodes.

Assume that there are two labels li and lj. $P\left(lj|li\right)$ represents the probability of the label lj appearing when the label li appears. And $P\left(li|lj\right)$ represents the probability of the label li appearing when the label lj appears. In Fig. 2, the probability of a horse co-occurring when a person is present is much higher than the probability of a horse co-occurring when a person is present.

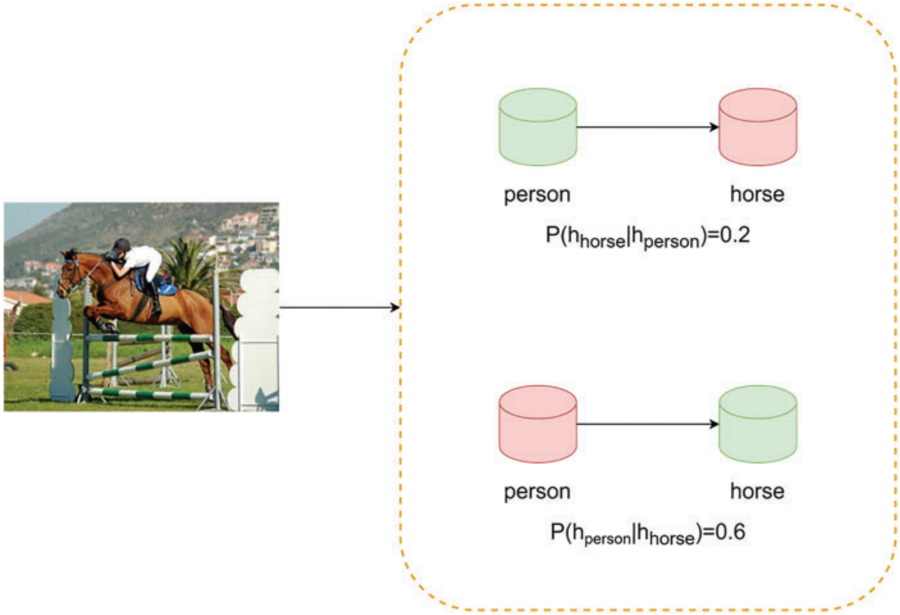

Figure 2: Description of label co-occurrence probability

In recent years, the field of multi-label image classification has witnessed a proliferation of research utilizing Graph Convolutional Network (GCN). Reference [13] introduces an innovative multi-label classification framework grounded in a Graph Convolutional Network (GCN), adept at discerning and modeling the nuanced relationships and interdependencies among label sets. In nearly the same time period, in response to the issue that current methods are unable to accurately pinpoint semantic regions, Reference [14] has proposed the SSGRL framework. Chen et al. make an improvement in the research [13] by introducing a new dynamic graph for multi-label recognition [15]. Reference [16] designs a novel deep learning framework that introduces a graph matching (GM) learning mechanism to explicitly mine the relationship between instances and labels.

Object detection networks are capable of identifying and localizing a diverse array of objects within an image, delineating their respective positions and dimensions. Despite occasional omissions due to factors such as inadequate lighting or the diminutive size of targets, these networks offer valuable contributions as auxiliary tasks in multi-label classification endeavors, thereby enhancing the overall performance of the classification system. Nonetheless, the integration of object detection with multi-label classification remains an underexplored area within the literature, with scant studies having addressed this combination to date.

Building upon the aforementioned considerations, we have developed an Object-GCN framework that integrates a dynamic graph convolutional network with an object detection network. Furthermore, we have incorporated word vector embeddings within the dynamic GCN to enhance the establishment of associations between visual features and semantic information.

In the present study, the key contributions we have made can be encapsulated in the following points:

- We firstly combine object detection network YOLOv5 and graph convolutional networks (GCN) for multi-label image classification tasks;
- We introduce a dynamic graph convolutional network (GCN) into the model and fuse the features of both image and text modalities when constructing graph nodes;
- We conduct a large number of comparative experiments on the datasets COCO and VOC, and obtain 86.9%, 96.7%, and 96.3% mAPs, respectively. These results are superior to the best model. We also performed ablation experiments to prove that both the object detection network we added and the proposed scheme for constructing graph nodes are effective.

2 Related Work

Nowadays, with the improvement of deep learning, CNN has been frequently used in image feature extraction, and it has shown good performance in public datasets for instance MSCOCO [17], VOC [18] and ImageNet [19]. Consequently, an extensive body of literature has introduced numerous frameworks predicated on Convolutional Neural Networks (CNNs) to tackle the multi-label image recognition challenge. Initially, within the research domain, the multi-label classification task has commonly been decomposed into multiple binary classification sub-tasks, where the model prognosticates one label per iteration [5]. Nonetheless, this methodological approach often overlooks the inter-label dependencies, a critical factor that necessitates consideration in the multi-label classification problem. In the context of zero-shot learning, Reference [6] presented a methodology for the simultaneous prediction of multiple image labels. Ji et al. [6] have introduced a versatile framework that encompasses two pivotal components: Visual Semantic Embedding and Zero-Shot Multilabel Prediction. This framework aims to gain a deep understanding of image content through the Visual Semantic Embedding module and to perform multilabel classification without direct samples using a zero-shot learning strategy, thereby enhancing the model's generalization ability and predictive accuracy. Reference [7] identified that the hinge loss function, commonly utilized in prior models, exhibits non-smooth characteristics, posing challenges for optimization. Consequently, Li et al. [7] have devised a novel pairwise ranking classification loss function, which is endowed with smoothness across its entire domain. The aforementioned methodologies transform the multi-label classification problem into a series of pairwise classification predictions.

Deep learning models have become prevalent for their efficacy in identifying and delineating interconnections among disparate labels. Given the proficient performance of Recurrent Neural Networks (RNNs) in textual data processing, numerous scholarly investigations have incorporated RNNs to characterize the interdependencies inherent within label sets. Wang et al. have developed a CNN-RNN hybrid framework [10], which integrates the complementary strengths of Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs). Specifically, the RNN component within this framework is strategically employed to discern and model the inter-label relationships. Subsequently, Reference [11] enhanced this CNN-RNN framework by incorporating a dynamic attention mechanism. This innovation enables the network to selectively focus on interest regions within the image, thereby improving the overall classification performance. Yazici et al. [20] have introduced methods for dynamically arranging the ground truth labels in accordance with the sequence of predicted labels, which speeds up and optimizes the LSTM training. Wang et al. have devised a two-component framework that employs a spatial converter layer to identify key attention areas within convolutional feature maps. Subsequently, an LSTM subnetwork is integrated to comprehend the overarching interdependencies among these regions [21].

With the popularity of graph convolutional networks, a substantial body of research has emerged that harnesses these networks to elucidate the relationships among categorical labels. Chen et al. pioneered a graph convolutional network modeled by graph nodes to predict multiple labels in an image [13]. They use the word vectors of each category word as the vertices of the graph in the graph convolutional network, and finally achieve better experimental results than other methods [13]. To thoroughly investigate the interactions within semantic regions, Chen et al. have introduced a novel framework for Signed Semantic Graph Reasoning Learning (SSGRL) that leverages graph theory [14]. This framework is composed of two integral modules: a semantic decoupling module designed to isolate individual semantic components, and a semantic interaction module that facilitates the analysis of interplays among these components. The ADD-GCN framework [15] introduces a significant enhancement over the ML-GCN model [13] by employing a dynamic Graph Convolutional Network (GCN) to supersede the traditional static graph convolutional approach. Wu et al. introduced a graph matching (GM) scheme to better establish the relationship between instances and labels [16]. In an effort to advance the ADD-GCN [15] framework, Cao et al. have introduced the 2S-DGCN model, presenting a novel iteration that builds upon the foundational strengths of its predecessor [22]. Zheng et al. have innovatively put forward the CGML framework, which leverages a Labels Adaptive Graph Convolutional Network (LAGCN) to effectively model the interdependencies between labels [23]. In order to explore the relationship between different images, Zhou et al. proposed a dual relational graph network framework, which uses a dual-branch structure to mine semantic information from both intra-image and inter-image simultaneously [24]. It is pointed out in Reference [25] that the collaborative relationship between labels in multi-label classification is closely related to the scene of an image. This paper introduce a novel graph learning framework designed to comprehensively discern the co-occurrence relationships among labels within varied imaging contexts.

Recently, transformer is also latterly used in multi-label image recognition. Lanchantin et al. have introduced acomprehensive framework grounded in the Transformer architecture to delineate the connections between visual features and their corresponding labels [26]. Furthermore, many studies have combined CNN and graph convolutional networks with transformers to study multi-label image classification. Zhao et al. combine graph and Transformer to propose a dual relations learning framework [27]. Zhao et al. deeply combine CNN and Transformer, and propose M3TR framework, a transformer model capable of learning intermodal relations and intra-modal triadic relations [28]. In recent years, there has been a significant trend in the literature to incorporate cross-modal features into classification frameworks. Specifically, Reference [29] has presented a pioneering Hierarchical Scale-Aware Vision-Language Transformer (HSVLT) framework. This framework is predicated on the Transformer model and introduces a novel attention mechanism module, which seamlessly integrates cross-modal interactions to enhance feature representation. Zhou et al. pointed out that many current transformer-based models do not mine various potentially useful features hidden by the most salient features in an image, and proposed a FL-Tran framework to solve the problem of difficult recognition of small-scale objects [30].

Despite the advancements, no studies have been conducted that integrate object detection tasks associated with multi-label image classification and the application of dynamic graph convolutional networks. Drawing inspiration from object detection methodologies, our study merges the well-established YOLOv5 deep learning model for object detection with the dynamic graph convolutional network to enhance multi-label image classification. Following extensive experimental analysis, our findings indicate that the YOLOv5 network is effective in increasing the precision of multi-label image classification.

3 Proposed Method

3.1 Overview of Object-GCN

To capture relationships between objects, the related studies of graph convolutional networks often use label co-occurrence. It can model relationships between labels. We use the visual feature $V = \{v1, v2, \ldots, vc\}$ extracted by the object detection network to represent the labels, and the association matrix A to represent the relationship between the labels (edges in the graph). Fig. 3 shows the entire framework Object-GCN we design.

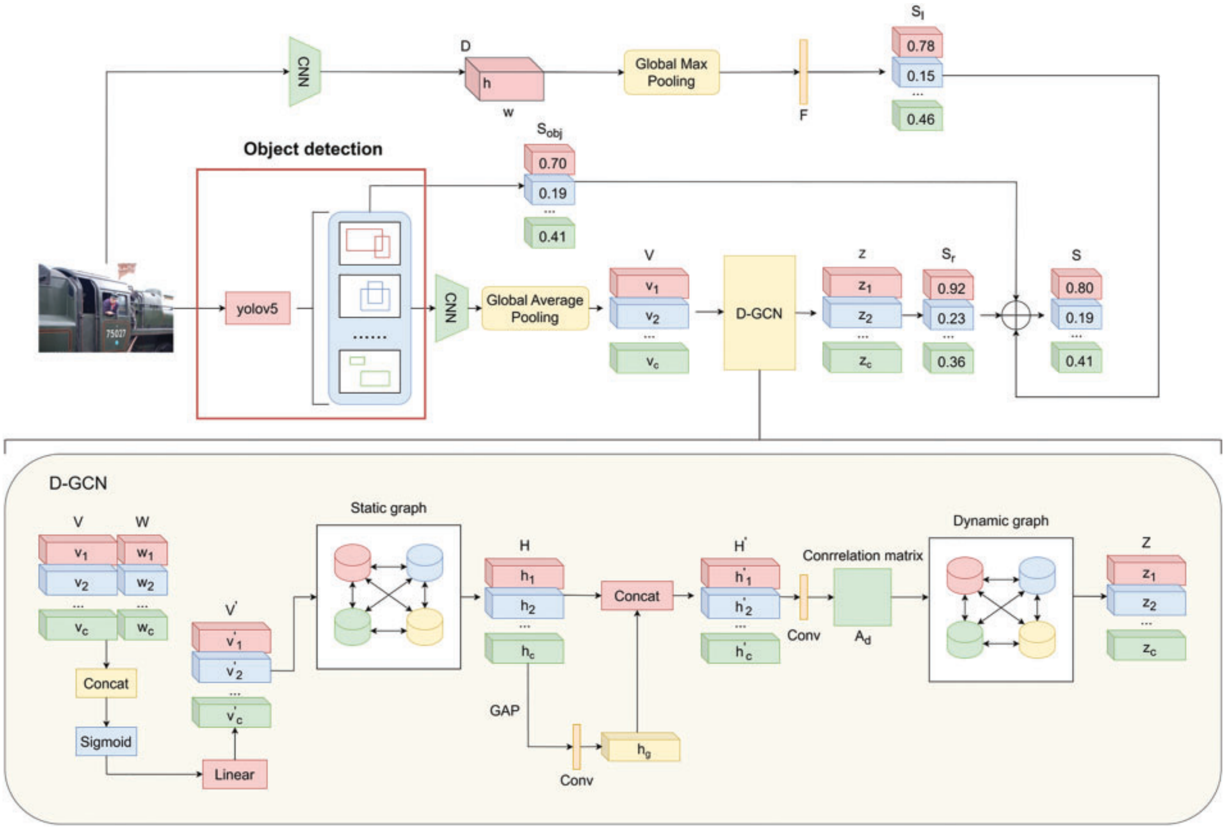

Figure 3: The overall framework of Object-GCN

In Reference [13], the authors boost the effectiveness of multi-label image classification through the creation of a convolutional network that leverages static graph models, but static graph cannot make full use of specific input images. So Reference [15] proposed a dynamic graph convolutional network to improve this problem, but the Class Activation Mapping they use is still less accurate, which will make the extracted features of a particular class mixed with features of other categories, which will have an impact on multi-label classification accuracy. The team led by Cao et al. has introduced a two-stream dynamic graph Convolutional network (2S-DGCN) to advance the accuracy of identifying images that possess multiple labels [22]. However, due to the influence of Class Active Map (CAM) [31], the performance improvement is very limited. Therefore, we use a combination of dynamic convolutional networks and YOLOv5 network to solve this problem.

The framework consists of three modules: the Convolutional network classification prediction module, Object Detection Category Feature Extraction module and the Dynamic GCN. The first module uses ResNet101 to extract the complicated image features and predict the probability of each category. The object detection module YOLOv5 is used to detect the object and extract the content-aware representation of each category c. V is fed into the Dynamic GCN for the final classification. To reinforce the correlation between semantic and visual features, this work integrates category-specific word vector embeddings into the dynamic graph convolutional network, thereby enhancing the network's capability to capture the underlying associations.

3.2 Image Feature Extraction and Classification

In this subsection, we plan to use ResNet101 [32] to extract the complicated features of the image. We adjusted the resolution of the image to 448 × 448, and then input it into the convolutional network. We remove the linear layer of ResNet101 and obtain the output of "conv5_x", generating 2048 × 14 × 14 feature maps. Then we choose global max pooling (GMP) to generate 2048 dimensional feature vector F for each train image.

$$F = fGMP\left(fcnn\left(I; \theta cnn\right)\right) \tag{1}$$

where $fGMP$ denotes the GMP operation, $fcnn$ denotes ResNet101, θcnn denotes the model parameters, and D = 2048. Then we use a fully connected layer to convert the 2048 dimensional vector into a C-dimensions vector, and eventually use the $softmax\left(\cdot\right)$ to obtain a set of predicted scores SI.

3.3 Object Detection Process Module

The Object Detection Process Module (ODPM) is designed to acquire content-aware representation for each category. Unlike ADD-GCN [15] which uses CAM [31], we directly use the object detection network YOLOv5 to extract the area of each object. We then compute a content-aware representation v_c for each category based on the confidence of each bounding box for each category. Assume that there are n bounding boxes each category, each bounding box is defined as b_c, i the confidence of each bounding box is $conf_c, i$ and the feature of the image obtained by cropping each bounding box is $f_{c,i}^D$.

Specifically, v_c is computed as:

$$v_c = Sigmoid\left(\sum_{i=1}^{n} conf_{c,i} f_{c,i}^D\right) \tag{2}$$

Then s_{obj}^c for each category can be computed as follows:

$$s_{obj}^c = 1 - \prod_{i=1}^{n}\left(1 - conf_{c,i}\right) \tag{3}$$

where *confc*, *i* represents the confidence of each bounding box in that category. YOLOv5 is a one-step object detection framework. It adds some new ideas for improvement on YOLOv4 [33], which gives it high speed and high accuracy. Different from the anchor-based Faster-RCNN [34], YOLOv5 directly predicts the center point and bounding box size of the object by transforming the object detection task into a regression problem, so as to achieve faster detection speed and higher accuracy. In comparison with the recently introduced YOLOv7 [35], YOLOv5 exhibits a modest reduction in accuracy. However, it offers superior training and inference velocities. Owing to these advantages, YOLOv5 has been selected as the foundational network for the object detection image processing module.

3.4 Dynamic GCN

Numerous recent studies have demonstrated the effectiveness of graph convolutional Networks (GCN) in computer vision tasks [12]. The static graph convolutional network proposed by [13] uses word vector embedding as the nodes of the graph, and the dynamic graph convolutional network proposed by [15] uses content-aware representation vectors as the nodes of the graph. Different from previous studies, we use a combination of word vector embedding and content-aware representation to represent graph nodes. Fig. 4 illustrates the formation of graph nodes. Given the distinct nature of image and text modalities, the concatenated feature vectors derived from both sources undergo a transformation through the Sigmoid function to ensure compatibility. Subsequently, the resultant fusion vector is refined by a fully connected layer to produce the final feature representation. We leverage a pre-trained GloVe model to procure 300-dimensional word vectors for each categorical term. These word vectors are concatenated with the 1024-dimensional categorical feature vectors, culminating in a comprehensive 1324-dimensional vector. Each dimension of this vector is then scaled to a range between 0 and 1 by the Sigmoid function, facilitating the normalization of feature values. Ultimately, this processed feature vector is conveyed through a fully connected layer, yielding the node features for the graph convolutional network.

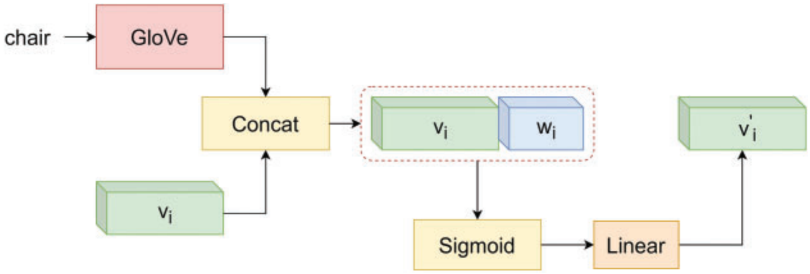

Figure 4: The generation process of graph nodes

Reference [13] compares several diverse word vector embedding methods such as GloVe [36], GoogleNews [37], FastText [38] and the one-hot word vector embedding, and finds that the capability of several algorithms is very approximate. But the capability of GloVe is still slightly higher than other methods. Therefore, we choose GloVe for word vector embedding in this paper.

In a static graph convolutional network, the correlation matrix is shared for all image samples, so the correlation matrix A_s is fixed for each sample. However, in order to improve the classification accuracy, we prefer that the graph convolutional network can capture the global coarse classification dependency of the image. After the static graph convolutional network, we get the output H. We successively input H into the global average pooling layer and the convolution layer to obtain $h_g \in R^{D1}$. This is done in order to fuse the feature vectors corresponding to each label category to obtain the feature vector of the global relationship. Then $H' \in R^{2D_1 \times C}$ is obtained by concatenated fusion using H and the global eigenvector representation h_g. By doing this, we concatenate the global feature vectors to the feature vectors of each original category label. Finally, we compute the dynamic correlation matrix A_d by H'. Different from the static correlation matrix A_s, the dynamic correlation matrix A_d is calculated according to the adaptive estimation of the input feature H, and different sample images have different A_d, which changes according to the input. It is also for this reason that the model has a higher representative power.

Thus V' can be expressed as:

$$V' = f_{linear}\left(Sigmoid\left(\left[(v1; w1), (v2; w2), \ldots, (v_c; w_c)\right]\right)\right) \in R^{C \times D} \tag{4}$$

where v_c stands for category content-aware representation and w_c stands for word vector embedding. V' is fed into the static GCN and the D-GCN in turn, and the resulting H is defined as follows:

$$H = LReLU\,(A_s V' W_s) \in R^{C \times D_1} \tag{5}$$

where A_s represents the correlation matrix of the static graph convolutional network, and $W_s^{D \times D_1}$ represents the weight matrix. LReLU is LeakyReLU [39] in the experiment. The nodes are updated as follows:

$$H^{l+1} = f\left(AH^l W\right) \tag{6}$$

where H^l denotes the node features of lth, and H^{l+1} denotes the node features of $(l+1)th$. The A matrix is generally predefined, the W matrix is obtained during training, and $f(\cdot)$ is the activation function, which is LeakyReLU in this paper. After obtaining H, we concatenate it with the hg obtained by the global average pooling layer(GAP) and the convolutional layer to obtain $H^{'}$. $H^{'}$ is represented as follows:

$$H' = [(h1; hg), (h2; hg), \ldots, (hc; hg)] \tag{7}$$

After that H will be input into the dynamic graph convolutional network, and the correlation matrix A_d of the dynamic graph convolutional network is defined as follows:

$$A_d = Sigmoid\,(W_A H') \tag{8}$$

where $WA = R^{2D_1 \times C}$ is the weight of the convolutional layer constructing the dynamic correlation matrix A_d. After H is fed into the dynamic graph convolutional network, the update of the nodes is represented as follows:

$$Z = LReLU\,(A_d H W_d) \tag{9}$$

where LReLU is LeakyReLU [39] and $W_d^{D_1 \times D_2}$ is the weight of the dynamic graph convolutional network. Different from static graph convolutional networks, dynamic graph convolutional networks have different A_d for each image. Different from those in [15,22], we fuse the content-aware representation and word vector in series and then input it into the graph convolutional network to better fuse the visual features and semantic features.

3.5 Loss and Classification

We end up with a category representation $Z = [z_1, z_2, \ldots, zc]$ from the dynamic neural network, each vector z_i contained in Z has complicated semantic relationships with other vectors and it is aligned with a specific category. Putting each vector z_i into it by a binary classifier yields a prediction score for each class. Where the confidence score vector s_i for each category is represented as follows:

$$s_r = \left[s_r^1, s_r^2, \ldots, s_r^c\right] \tag{10}$$

In [15,22], the Class Activation Map (CAM) is used to calculate the score of each category to obtain another credibility score s_m. We apply the method of object detection when extracting the content-aware representation, so the score s_{obj} we get from the object detection module is represented as follows:

$$s_{obj} = \left[s_{obj}^1, s_{obj}^2, \ldots, s_{obj}^c\right] \tag{11}$$

where s_{obj}^i represents the confidence in object detection for category i. In the image feature extraction module, the score s_I obtained through the ResNet101 classifier is defined as follows:

$$s_I = \left[s_I^1, s_I^2, \ldots, s_I^c\right] \tag{12}$$

We take the average of these three vectors and we get our final vector s. The vector s is defined as follows:

$$s = \left[s^1, s^2, \ldots, s^c\right] = \left[\frac{s_r^1 + s_{obj}^1 + s_I^1}{3}, \frac{s_r^2 + s_{obj}^2 + s_I^2}{3}, \ldots, \frac{s_r^c + s_{obj}^c + s_I^c}{3}\right] \tag{13}$$

Finally, we train the whole Object-GCN model using the traditional multi-label classification training loss function, which is defined as follows:

$$L\left(y, s\right) = \sum_{c=1}^{C} y^c \log\left(\left(s^c\right)\right) + \left(1 - y^c\right) \log\left(1 - s^c\right) \tag{14}$$

4 Experiment

4.1 Datasets

MSOCO 2014 [17] is a versatile dataset that is often used for computer vision tasks. It has also been generally used for image recognition task in recent years. The public dataset includes a total of 122,218 images, of which the training part includes 82,081 images and the validation part includes 40,137 images. The images in the dataset were classified into 80 daily categories. And each image contains an average of 2.9 categories.

VOC 2007 [18] is similarly generally used for multi-label classification tasks. The researchers classified all images in the dataset into 20 common categories, including people, vehicles, animals, and more. Among them, the training and validation sets have 5011 images, and the test set contains 4952 images.

VOC 2012 [18] is an extension of Pascal VOC 2007. The dataset is also divided into 20 common categories. Among them, 11,540 images are the training part and 10,991 images are the test part. There are a lot of new images compared to the previous version.

4.2 Metrics

To facilitate a rigorous comparison with existing baseline models, this study meticulously conducts an extensive experimental evaluation on the aforementioned trio of datasets, adhering to the established metrics and methodologies of prior research. Among them, the evaluation metrics are OP, CP, OR, CR, OF1, CF1 and the mean Average Precision (mAP). In addition, because the COCO2014 dataset is so diverse, we also provide the top-3 data for each metric.

4.3 Baselines

This paper mainly uses graph convolutional networks, so in addition to comparing with classical methods such as CNN-RNN [10], ResNet101 [32], RNN-Attention [21], Multi-evidence [40], this paper focuses on comparing with methods based on graph convolutional networks. These include ML-GCN [13], ADD-GCN [15] and many more. Since Transformers have shown good performance in establishing the relationship between images and labels in recent years, we similarly select the transformer-based models C-Train [26], TDRG [27] and M3TR [28].

4.4 Experiment Detail

The deep learning framework selected in this paper is PyTorch [41], and the code is written based on Python 3.8. The GPU configuration we used was RTX 4090, PyTorch version 1.8.1, and Cuda version 11.1. We use ResNet101 as the backbone network to extract the content-aware representation of each class, and the 300-dimensional GloVe pre-trained model to obtain the word vector representation. In the object detection module, we select the YOLOv5-x pre-trained weight parameter with the highest accuracy for training.

SGD is chosen as the optimizer, reducing momentum by 0.9 and weight by 10^{-4}. The learning rate of dynamic convolution graph network is 0.5, and the learning rate of ResNet101 used for image feature extraction is 0.05. Because in the early stage of deep learning network training, we want our network to converge quickly with a large learning rate. In the later stage of training, we want the learning rate to be small, so that the network can avoid oscillating back and forth when converging to the optimal point, so as to better converge to the optimal solution. We set the training round as 50 and reduced the learning rate of ResNet101 by 0.1 at 30 and 40 epochs, respectively. The GPU has a batch size of 16.

We applied a series of preprocessing to the images. The image is resized to 448 × 448 and 576 × 576, random horizontal flipping and multi-scale cropping are performed, and the image is normalized so that the image data has a similar distribution on each channel. The implemented preprocessing techniques serve to standardize the image data, thereby enhancing its quality and, consequentially, augmenting the model's performance and enhancing its generalization capabilities to a significant extent.

4.5 Results

4.5.1 Results on COCO2014

We contrast the metrics of Object-GCN model with recent methods on COCO dataset, and the data of the baseline model comes from the respective papers. Table 1 illustrates the results of comparison, from which we can see that the mAP of our model is 7.5% higher than ResNet101 [32], and has a significant improvement over ADD-GCN [15] and 2S-DGCN [17], which are also based on dynamic graph convolutional networks. In order to make a fair comparison, we selected the experimental results with two imageSize Settings of 448 and 576, and compared with other models. Finally, the experimental results show that our framework has advantages in both Settings Concurrently, it is observed that the dimensions of the image exert a notable influence on the experimental outcomes. Generally, an increase in image size correlates with an enhancement in the performance of the resultant model.

Table 1: Comparison of Object-GCN and other frameworks on the COCO2014

Method	Resolution	All							Top-3					
		mAP	CP	CR	CF_1	OP	OR	OF_1	CP	CR	CF_1	OP	OR	OF_1
[10] CNN-RNN	–	61.2	–	–	–	–	–	–	66.0	55.6	60.4	69.2	66.4	67.8
[21] RNN-attention	–	–	–	–	–	–	–	–	79.1	58.7	67.4	84.0	63.0	72.0
[32] ResNet101	448 × 448	79.7	82.7	67.4	74.3	86.4	71.8	78.4	85.9	60.5	71.0	90.2	64.2	75.0
[40] Multi-evidence	448 × 448	–	80.4	70.2	74.9	85.2	72.5	78.4	84.5	62.2	70.6	89.1	64.3	74.7
[13] ML-GCN	448 × 448	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
[14] SSGRL	448 × 448	81.9	84.2	70.3	76.6	85.8	72.4	78.6	88.0	63.1	73.5	90.2	64.5	75.2
[15] ADD-GCN	448 × 448	85.2	84.7	75.9	80.1	84.9	79.4	82.0	88.8	66.2	75.8	90.3	68.5	77.9
[22] 2S-DCN	448 × 448	85.6	84.9	75.7	80.0	86.8	78.0	82.2	88.9	66.7	76.3	91.1	68.1	77.9
[24] DRGN	448 × 448	84.9	86.3	73.8	79.6	87.4	76.6	81.6	89.5	65.7	75.8	91.3	67.4	77.6
[25] SALGL	448 × 448	85.8	87.2	74.5	80.4	87.8	77.6	82.4	90.4	65.7	76.1	91.8	67.9	78.1
[27] TDRG	448 × 448	86.0	87.0	74.7	80.4	87.5	77.9	82.4	90.7	65.6	76.2	91.9	68.0	78.1
[25] SALGL	576 × 576	87.3	87.8	76.8	81.9	88.1	79.5	83.6	91.1	66.9	77.2	92.4	69.0	79.0
[24] DRGN	576 × 576	86.4	87.4	75.6	81.1	88.1	78.3	82.9	90.6	66.7	76.8	92.1	68.3	78.5
[26] C-Train	576 × 576	85.1	86.3	74.3	79.9	87.7	76.5	81.7	90.1	65.7	76.0	92.1	**71.4**	77.6
[28] M3TR	576 × 576	**87.5**	88.4	**77.2**	**82.5**	88.3	79.8	83.8	**91.9**	68.1	78.2	92.6	69.6	79.4
Object-GCN	448 × 448	86.2	87.4	73.9	80.1	87.6	78.0	82.5	90.5	65.8	76.2	92.0	67.9	78.1
Object-GCN	576 × 576	86.9	**88.7**	76.8	82.3	**88.5**	**80.1**	**84.1**	91.4	**68.5**	**78.3**	**92.8**	69.4	**79.5**

Although our model does not use transformer, it still has certain advantages over the recent transformer-based models C-Train [26],TDRG [27] and M3TR [28]. Among the fully GCN-based frameworks, our framework obtains the best metrics. The experiment results demonstrate that the object detection module we introduced greatly improves the metrics. It can be discovered from the table that after the introduction of the object detection module, the CP has been significantly improved, and it should be that the object detection module makes some small targets that are not easy to be identified.

4.5.2 Results on VOC

In this paper, the performance of our framework is contrasted with the baseline framework on VOC2007 dataset, and the detection accuracy of 20 categories is obtained. Table 2 demonstrates the experimental results we obtained.

Table 2: Comparison of Object-GCN and other models on the VOC2007 dataset

	Resolution	Aero	Bike	Bird	Boat	Bottle	Bus	Car	Cat	Chair	Cow	Table	Dog	Horse	Motor	Person	Plant	Sheep	Sofa	Train	Tv	mAP
[10]	–	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
[21]	–	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
[32]	448 × 448	99.1	97.3	96.2	94.7	68.3	92.9	95.9	94.6	77.9	89.9	85.1	94.7	96.8	94.3	98.1	80.8	93.1	79.1	98.2	91.1	90.8
[13]	448 × 448	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
[14]	448 × 448	99.7	98.4	98.0	97.6	85.7	96.2	98.2	98.8	82.0	98.1	89.7	98.8	98.7	97.0	99.0	86.9	98.1	85.8	99.0	93.7	95.0
[24]	448 × 448	99.8	98.6	98.3	98.6	81.8	95.5	97.6	98.0	83.9	94.9	87.5	98.4	97.8	97.4	98.8	86.6	96.2	85.6	99.4	94.9	94.5
[25]	448 × 448	99.9	98.8	98.3	98.2	81.6	96.5	98.1	97.8	85.2	97.0	89.6	98.5	98.7	97.1	99.2	86.9	96.4	89.9	99.5	95.2	95.1
[27]	448 × 448	99.9	98.9	98.4	98.7	81.9	95.8	97.8	98.0	85.2	95.6	89.5	98.8	98.6	97.1	99.1	86.2	97.7	87.2	99.1	95.3	95.0
[15]	576 × 576	99.8	99.0	98.4	99.0	86.7	98.1	98.5	98.3	85.8	98.3	88.9	98.8	99.0	97.4	99.2	88.3	98.7	90.7	99.5	97.0	96.0
[28]	576 × 576	99.9	99.3	99.1	99.1	84.0	97.6	98.0	99.0	85.9	99.4	**93.9**	99.5	99.4	98.5	99.2	90.3	99.7	91.6	99.8	96.0	96.5
[25]	576 × 576	**100.0**	99.2	98.8	98.6	**87.1**	98.1	**99.0**	**99.2**	**87.9**	98.9	92.3	98.8	99.1	**98.9**	99.4	89.5	99.0	**93.7**	99.8	97.1	96.7
[24]	576 × 576	99.9	99.1	98.8	98.5	90.0	97.9	98.9	98.8	85.9	98.6	90.6	99.0	99.0	97.8	99.3	90.3	99.6	89.0	99.4	96.8	96.4
Ours	448 × 448	99.9	99.2	98.6	98.8	85.6	97.9	98.4	98.4	85.2	98.2	89.3	99.1	99.3	97.7	98.9	88.5	98.6	91.5	99.5	96.9	96.0
Ours	576 ×576	99.9	**99.5**	**99.2**	**99.2**	87.0	**98.2**	98.7	98.9	86.0	**99.5**	90.1	**99.6**	**99.6**	98.3	**99.4**	**91.1**	99.5	92.6	**99.9**	**97.3**	**96.7**

From the information in the table, it can be seen that the mAP of our model is increased by 5.2% on the basis of ResNet101 [32]. Among the frameworks predicated on graph convolutional networks, our Object-GCN framework demonstrates superior performance. Specifically, in the majority of categories, Object-GCN surpasses the Transformer-based TDRG [27] and M3TR [28], showcasing its efficacy in comparative evaluations.

VOC2012 is an extension of VOC2007 dataset, so this paper also conducts experiments on VOC2012 dataset. The VOC2012 dataset also has 20 categories and Table 3 shows the results of the experiments. We choose RMIC [42], VeryDeep [43], HCP [44], SSGRL [14], ADD-GCN [15] as our baseline models.

Table 3: Comparison of Object-GCN and other models on the VOC2012 dataset

	Resolution	Aero	Bike	Bird	Boat	Bottle	Bus	Car	Cat	Chair	Cow	Table	Dog	Horse	Motor	Person	Plant	Sheep	Sofa	Train	Tv	mAP
[42]	–	98.0	85.5	92.6	88.7	64.0	86.8	82.0	94.9	72.7	83.1	73.4	95.2	91.7	90.8	95.5	58.3	87.6	70.6	93.8	83.0	84.4
[43]	–	99.1	88.7	95.7	93.9	73.1	92.1	84.8	97.7	79.1	90.7	83.2	97.3	96.2	94.3	96.9	63.4	93.2	74.6	97.3	87.9	89.0
[44]	–	99.1	92.8	97.4	94.4	79.9	93.6	89.8	98.2	78.2	94.9	79.8	97.8	97.0	93.8	96.4	74.3	94.7	71.9	96.7	88.6	90.5
[14]	448 × 448	99.7	96.1	97.7	96.5	86.9	95.8	95.0	98.9	88.3	97.6	87.4	99.1	99.2	97.3	99.0	84.8	98.3	85.8	99.2	94.1	94.8
[15]	448 × 448	99.8	97.1	98.6	96.8	89.4	97.1	96.5	99.3	89.0	97.7	87.5	99.2	99.1	97.7	99.1	86.3	98.8	87.0	99.3	95.4	95.5
Ours	448 × 448	99.8	97.0	98.7	97.5	88.6	97.7	96.8	99.3	89.2	97.8	88.2	99.0	98.7	98.0	99.3	87.4	98.5	88.1	98.9	95.8	95.7
Ours	576 × 576	99.8	**97.4**	**99.0**	**98.2**	89.1	**98.4**	**97.5**	**99.6**	**89.8**	**98.3**	**88.9**	**99.5**	99.2	**98.1**	**99.5**	**88.3**	**99.2**	**89.0**	**99.6**	**96.7**	**96.3**

From the data in the table, we can see that our Object-GCN achieves the best results. Specifically, it exhibits a relative enhancement of 0.2% in mean Average Precision (mAP) compared to the baseline ADD-GCN model [15], and a more pronounced improvement of 0.9% over the SSGRL [14].

4.5.3 Limited Annotations Studies

Our experiment uses an object detector to extract the category features in the image, but in reality many images lack complete annotations, so we conduct more experiments to solve this problem. Our experiment uses an object detector to extract the category features in the image, but in reality many images lack complete annotations, so we conduct more experiments to solve this problem.

The VOC2007 dataset contains 9963 labeled images with 24,640 labeled objects. We randomly select 10%–90% of the number of annotations to train our object detector. We separately calculate the mAP of the Object-GCN framework in the case of different Object detectors for comparison. When training the YOLOv5 object detector, we set the confidence *conf* to 0.25. Table 4 shows the performance of the model when the ratio of annotations is different. And we set the image size to 448 × 448.

Table 4: mAP of different annotation ratios in the VOC2007 dataset

Dataset	10%	20%	30%	40%	50%	60%	70%	80%	90%	Ave. mAP
VOC2007	84.3	87.5	89.8	93.9	94.6	95.0	95.4	95.8	95.8	92.5

Fig. 5 illustrates the trend of the model mAP when the ratio of annotations is different. It can be seen from the figure that when the number of annotation boxes is small, the performance of the model is significantly improved with the increase of the number of annotations. The possible reason is that the improved accuracy of the object detector makes the prediction of the whole model more accurate. When the number of comment boxes increases, the performance of the model improves slowly. The performance of Object-GCN is even worse than ResNet101 [32] when there are fewer annotation boxes, and we guess the reason may be that the poor performance of the object detector at this time has a negative impact on the model. When the proportion of annotated boxes reaches more than 30%, the performance of our framework exceeds the baseline model.

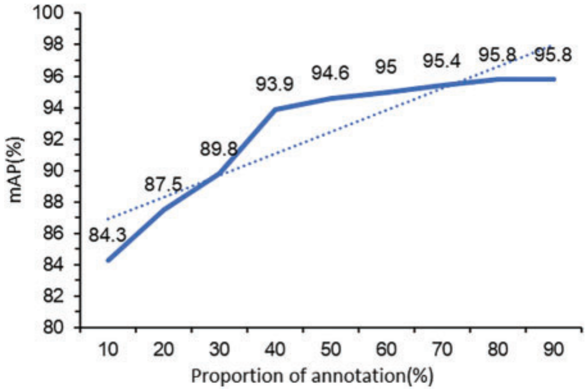

Figure 5: Model performance with different annotation ratios

4.5.4 Results Conclusions

From the data in the above table, we can see that our proposed model shows good performance on the three datasets. Compared with ResNet101, our model uses a dynamic graph convolutional network to take into account the dependency between labels, so it has a large improvement over the baseline model. Compared with other frameworks that use graph convolutional networks, our proposed framework uses an object detector instead of CAM to extract image features of each category, and the image features extracted by the object detector are more accurate than the category activator. Moreover, we also fuse the word vectors and image features generated by GloVe. Therefore, our model has a large performance improvement over the baseline model ResNet101 and the graph convolution framework ADD-GCN.

4.6 *Ablation Studies*

In this subsection, we conduct ablation studies in order to explore the contributions of the Object detection module and Dynamic GCN in Object-GCN to the whole model, respectively. We remove the Object detection module and the Dynamic GCN, respectively, and compare with the baseline model and the Object-GCN framework. To evaluate the effect of the object detection module, we delete the object detection module when performing ablation studies. Fig. 6 presents the results of our ablation studies. Fig. 6 shows that compared with the baseline model ResNet101 [32], the Dynamic GCN and object detection modules we add can improve the performance of the framework. However, compared with the Dynamic GCN, the object detection module has less improvement in performance. The best results can be obtained by adding both the object detection module and the Dynamic GCN. In the MS COCO dataset, the improvements of mAP, OF1 and CF1 are 7.2%, 5.7% and 8.0%, respectively. On the VOC2007 dataset, the performance improvement of the three metrics is 5.9%, 9.3%, 8.3%. Compared with the ablation experimental data in Reference [15], the effect of object detection and category feature extraction module proposed by us is more significant than that of SAM module proposed in [15]. At the same time, our dynamic graph convolutional network also performs better than [15] because we fuse the vectors of both image and text modalities when generating the graph vertices.

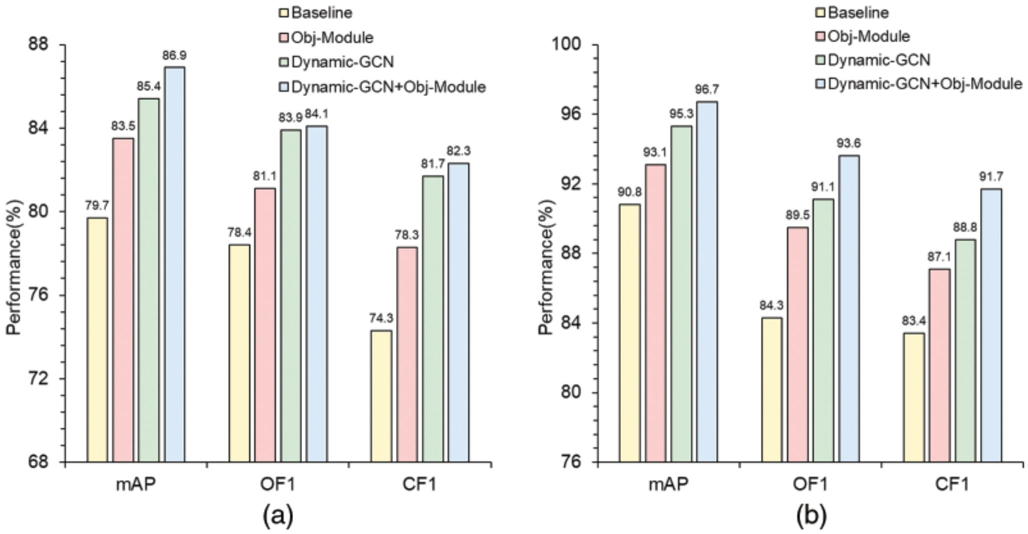

Figure 6: Dynamic-GCN and Obj-Module on MS-COCO and VOC2007 ((a): Comparisons on MS-COCO, (b): Comparisons on VOC2007)

4.7 *Visualization*

In this section, we visualize the object detection module and the dynamic graph convolutional network introduced in this paper.

4.7.1 Visualization of the Object Detection Module

We input the original image into the YOLOv5 network for object detection, which shows that YOLOv5 has the capacity to detect objects in the picture. Some examples are shown in Fig. 7, where the first column represents the input image, the second column represents the image after YOLOv5 object detection, and the last column shows the prediction scores for each category resulting from the whole framework.

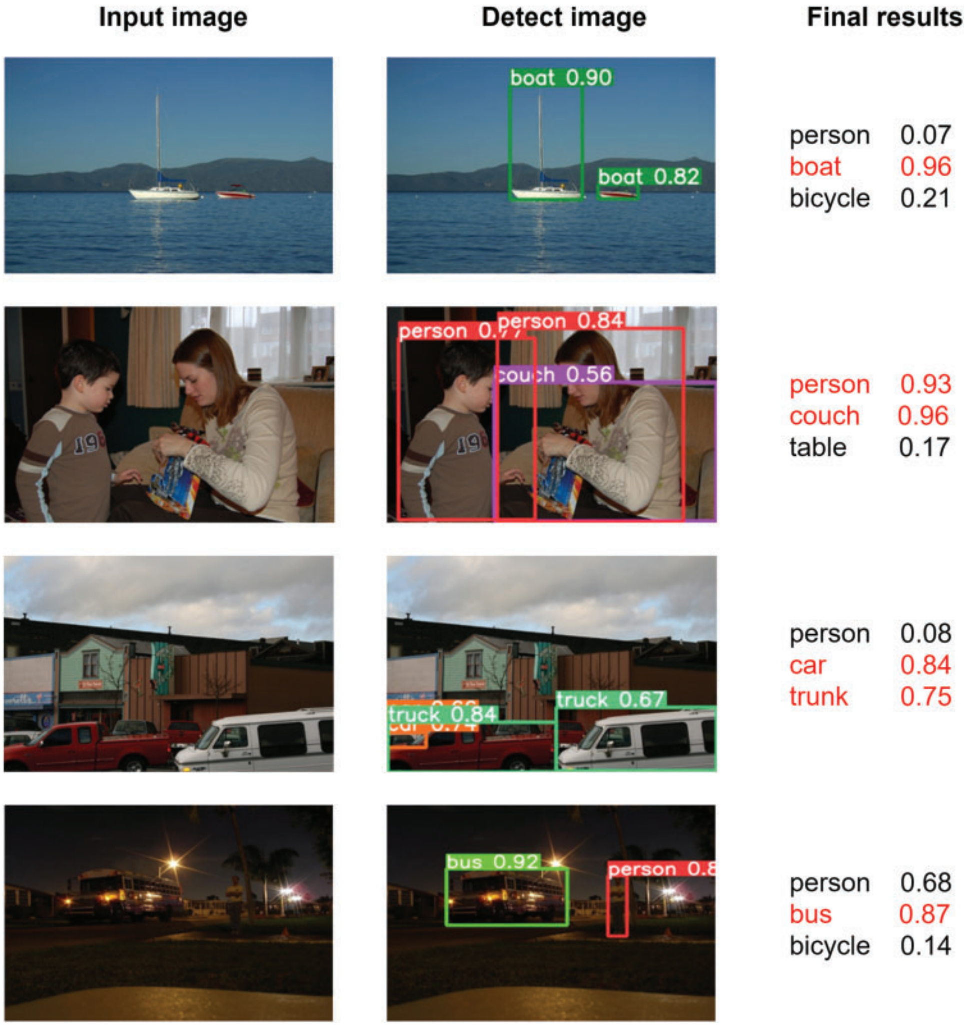

Figure 7: Visualization of the object detection module

As can be seen from the figure, our framework is able to recognize the various types of objects in the image and get their location in the image. When a category does not exist in the image, the YOLOv5 network cannot recognize it. In the fourth example, when the scene in the picture is dimly lit, it is easy to ignore the object in the picture using convolutional neural network, but this shortcoming can be well compensated by using YOLOv5.

4.7.2 Visualization of the Dynamic Graph Convolutional Networks

In our paper, the input of our dynamic graph convolutional network is converted, so the correlation matrix of the dynamic graph convolutional network A_d changed accordingly. We can look at the correlation matrix of the dynamic GCN to observe the contribution of the object detection module and word vector embedding we use.

It can be seen from Fig. 8 that the truth values of the input original image are "person", "dog" and "sofa". Comparing $A_d^{person,sofa}$ and $A_d^{person,dog}$ in (a) and (b) in Fig. 8, it can be seen that "sofa" and "dog" have enhanced correlation with people. Similar consequence can be seen for the two rows "sofa" and "dog". From the visualization of dynamic graph convolutional networks, we can see that our approach has good capability in establishing and capturing label semantic relationships.

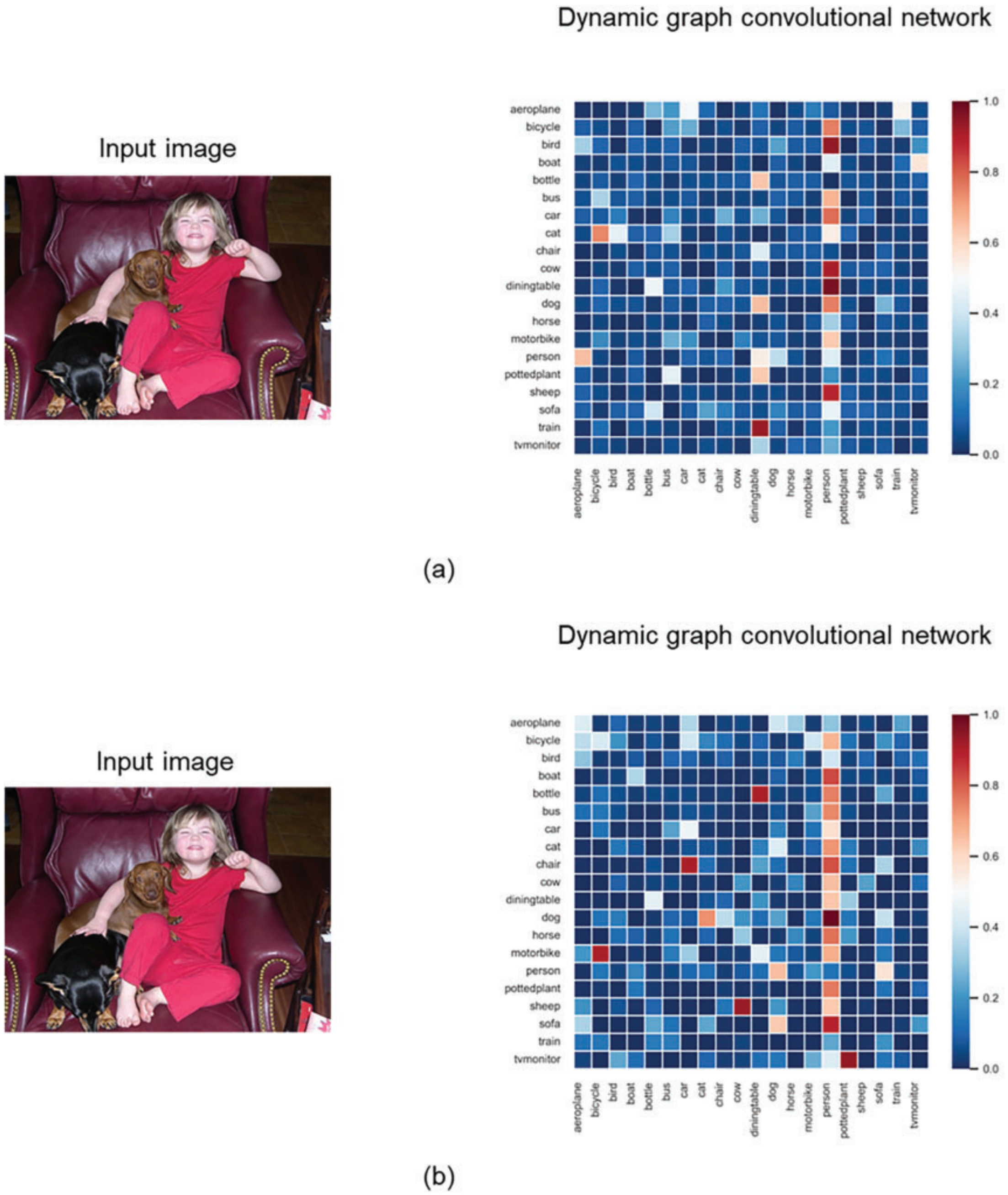

Figure 8: Visualization of the dynamic graph convolutional networks ((a): No change, (b): Object detection module and word embedding)

5 Conclusion

In this work, we introduce a groundbreaking approach by integrating the YOLOv5 object detection model with dynamic graph convolutional networks for multi-label image classification. We present the Object-GCN framework, leveraging the YOLOv5 and ResNet101 architectures to extract visual features, which serve as the content-aware representation for each category. Within the dynamic graph convolutional network, we concatenate GloVe pre-trained word vectors with content-aware vectors to form the vertices of the GCN, thereby enhancing the integration of semantic and visual features. We have conducted rigorous experiments on public datasets and devised ablation studies to validate our methodology. The results of these experiments substantiate the significant contributions of our proposed enhancements.

Our proposed Object-GCN demonstrates commendable performance in the domain of multi-label image classification. However, there remains ample scope for further refinement and advancement. In our proposed model, the trained object detector is used to extract the visual features of each category, which requires additional labeling boxes and a certain training cost. Secondly, we use a concatenation method to fuse text and image features. In the future, we will use a multimodal low-rank bilinear method to fuse visual features and semantic features to generate the nodes of the graph, so that the graph convolutional network can better learn the relationship between vision and semantics. In the following work, we will explore how the combination of category word vectors and content-aware representation vectors can achieve the best results (concatenate or add). Then, we will explore how to combine our research with Transformers to achieve better results.

Acknowledgement: The authors would like to thank the editors and reviewers.

Funding Statement: The authors received no specific funding for this study.

Author Contributions: The authors confirm contribution to the paper as follows: Study conception and design: Xiaoyu Liu; data collection: Xiaoyu Liu; analysis and interpretation of results: Yong Hu; draft manuscript preparation: Xiaoyu Liu, Yong Hu. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: The datasets used to support the findings of this study are publicly available on http://cocodataset.org (accessed on 21 July 2023) and http://host.robots.ox.ac.uk/pascal/VOC/ (accessed on 24 July 2023).

Ethics Approval: Not applicable.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] Z. Guo, B. Dong, Z. Ji, J. Bai, Y. Guo and W. Zuo, “Texts as images in prompt tuning for multi-label image recognition,” in *Proc. IEEE Int. Conf. Comput. Vis.*, Paris, France, 2023, pp. 2808–2817.

[2] H. Guo, X. Fan, and S. Wang, “Visual attention consistency for human attribute recognition,” *Int. J. Comput. Vis.*, vol. 130, no. 4, pp. 1088–1106, 2022. doi: 10.1007/s11263-022-01591-y.

[3] Y. Wu, J. Chen, J. Yan, Y. Zhu, D. Z. Chen and J. Wu, “GCL: Gradient-guided contrastive learning for medical image segmentation with multi-perspective meta labels,” in *Proc. 31st ACM Int. Conf. Multimed.*, Ottawa, ON, Canada, 2023, pp. 463–471.

[4] K. V. Demochkin and A. V. Savchenko, "Multi-label image set recognition in visually-aware recommender systems," *Anal. Images, Social Netw. Texts: 8th Int. Conf.*, Kazan, Russia, 2019, pp. 291–297.

[5] E. Montañés, R. Senge, J. Barranquero, J. R. Quevedo, J. J. del Coz and E. Hüllermeier, "Dependent binary relevance models for multi-label classification," *Pattern Recognit.*, vol. 47, no. 3, pp. 1494–1508, 2014. doi: 10.1016/j.patcog.2013.09.029.

[6] Z. Ji *et al.*, "Deep ranking for image zero-shot multi-label classification," *IEEE Trans. Image Process.*, vol. 29, pp. 6549–6560, 2020. doi: 10.1109/TIP.2020.2991527.

[7] Y. Li, Y. Song, and J. Luo, "Improving pairwise ranking for multi-label image classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Hawaii, HI, USA, 2017, pp. 1837–1845.

[8] C. -K. Yeh, W. -C. Wu, W. -J. Ko, and Y. -C. F. Wang, "Learning deep latent space for multi-label classification," in *Proc. AAAI Conf. Artif. Intell.*, San Francisco, CA, USA, 2017, pp. 2838–2844.

[9] S. Wen *et al.*, "Multilabel image classification via feature/label co-projection," *IEEE Trans. Syst. Man Cybern. Syst.*, vol. 51, no. 11, pp. 7250–7259, 2021. doi: 10.1109/TSMC.2020.2967071.

[10] J. Wang, Y. Yang, J. Mao, Z. Huang, C. Huang and W. Xu, "CNN-RNN: A unified framework for multi-label image classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Las Vegas, NV, USA, 2016, pp. 2285–2294.

[11] F. Lyu, Q. Wu, F. Hu, Q. Wu, and M. Tan, "Attend and imagine: Multi-label image classification with visual attention and recurrent neural networks," *IEEE Trans. Multim.*, vol. 21, no. 8, pp. 1971–1981, 2019. doi: 10.1109/TMM.2019.2894964.

[12] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," 2017, *arXiv:1609.02907*.

[13] Z. -M. Chen, X. -S. Wei, P. Wang, and Y. Guo, "Multi-label image recognition with graph convolutional networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Los Angeles, CA, USA, 2019, pp. 5177–5186.

[14] T. Chen, M. Xu, X. Hui, H. Wu, and L. Lin, "Learning semantic-specific graph representation for multi-label image recognition," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, Seoul, Republic of Korea, 2019, pp. 522–531.

[15] J. Ye, J. He, X. Peng, W. Wu, and Y. Qiao, "Attention-driven dynamic graph convolutional network for multi-label image recognition," in *Comput. Vis. ECCV 2020: 16th European Conf.*, Glasgow, UK, 2020, pp. 649–665.

[16] Y. Wu, H. Liu, S. Feng, Y. Jin, G. Lyu and Z. Wu, "GM-MLIC: Graph matching based multi-label image classification," 2021, *arXiv:2104.14762*.

[17] T. -Y. Lin *et al.*, "Common objects in context," in *Comput. Vis. ECCV 2014: 13th European Conf.*, Zurich, Switzerland, 2014, pp. 740–755.

[18] M. Everingham, L. V. Gool, C. K. I. Williams, J. M. Winn, and A. Zisserman, "The pascal visual object classes (VOC) challenge," *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303–338, 2010. doi: 10.1007/s11263-009-0275-4.

[19] J. Deng, W. Dong, R. Socher, L. J. Li, and F. F. Li, "ImageNet: A large-scale hierarchical image database," in *2009 IEEE Conf. Comput. Vis. Pattern Recognit.*, Miami, FL, USA, 2009, pp. 248–255.

[20] V. O. Yazici, A. Gonzalez-Garcia, A. Ramisa, B. Twardowski, and J. V. D. Weijer, "Orderless recurrent models for multi-label classification," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Seattle, USA, 2020, pp. 13437–13446.

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

[22] P. Cao, P. Chen, and Q. Niu, "Multi-label image recognition with two-stream dynamic graph convolution networks," *Image Vis. Comput.*, vol. 113, 2021, Art. no. 104238.

[23] X. Zheng, X. Liang, and B. Wu, "Capsule graph neural network for multi-label image recognition (Student Abstract)," in *Proc. AAAI Conf. Artif. Intell.*, Vancouver, BC, Canada, 2022, vol. 36, pp. 13117–13118.

[24] W. Zhou, W. Jiang, D. Chen, H. Hu, and T. Su, "Mining semantic information with dual relation graph network for multi-label image classification," *IEEE Trans. Multim.*, vol. 26, pp. 1143–1157, 2024.

[25] X. Zhu, J. Liu, W. Liu, J. Ge, B. Liu and J. Cao, "Scene-aware label graph learning for multi-label image classification," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, Paris, France, 2023, pp. 1473–1482.

[26] J. Lanchantin, T. Wang, V. Ordonez, and Y. Qi, "General multi-label image classification with transformers," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Nashville, TN, USA, 2021, pp. 16478–16488.

[27] J. Zhao, K. Yan, Y. Zhao, X. Guo, F. Huang and J. Li, "Transformer-based dual relation graph for multi-label image recognition," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, Montreal, QC, Canada, 2021, pp. 163–172.

[28] J. Zhao, Y. Zhao, and J. Li, "M3TR: Multi-modal multi-label recognition with transformer," in *Proc. 29th ACM Int. Conf. Multimed.*, Chengdu, China, 2021, pp. 469–477.

[29] S. Ouyang *et al.*, "HSVLT: Hierarchical scale-aware vision-language transformer for multi-label image classification," in *Proc. 31st ACM Int. Conf. Multimed.*, Ottawa, ON, Canada, 2023, pp. 4768–4777.

[30] W. Zhou, P. Dou, T. Su, H. Hu, and Z. Zheng, "Feature learning network with transformer for multi-label image classification," *Pattern Recognit.*, vol. 136, 2023, Art. no. 109203. doi: 10.1016/j.patcog.2022.109203.

[31] B. Zhou, A. Khosla, À. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localizaion," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Las Vegas, NV, USA, 2016, pp. 2921–2929.

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

[33] A. Bochkovskiy, C. -Y. Wang, and H. -Y. M. Liao, "YOLOv4: Optimal speed and accuracy of object detection," 2020, *arXiv:2004.10934*.

[34] S. Ren, K. He, R. B. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, 2017. doi: 10.1109/TPAMI.2016.2577031.

[35] C. -Y. Wang, A. Bochkovskiy, and H. -Y. M. Liao, "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," 2023, *arXiv:2207.02696*.

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

[37] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," 2013, *arXiv:1301.3781*.

[38] A. Joulin, E. Grave, P. Bojanowski, M. Douze, H. Jégou and T. Mikolov, "FastText.zip: Compressing text classification models," 2016, *arXiv:1612.03651*.

[39] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," in *ICML Workshop Deep Learn. Audio Speech Lang. Process.*, Atlanta, GA, USA, 2013.

[40] 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. Vis. Pattern Recognit.*, Salt Lake City, UT, USA, 2018, pp. 1277–1286.

[41] A. Paszke *et al.*, "Automatic differentiation in PyTorch," in *31st Conference on Neural Information Processing Systems (NIPS 2017)*, Long Beach, CA, USA, 2017.

[42] S. He, C. Xu, T. Guo, C. Xu, and D. Tao, "Reinforced multi-label image classification by exploring curriculum," in *Proc. AAAI Conf. Artif. Intell.*, New Orleans, LA, USA, 2018, vol. 32, pp. 3183–3190.

[43] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2015, *arXiv:1409.1556*.

[44] 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, 2016. doi: 10.1109/TPAMI.2015.2491929.