Chapter 20
Graph Neural Networks in Computer Vision

Siliang Tang, Wenqiao Zhang, Zongshen Mu, Kai Shen, Juncheng Li, Jiacheng Li and Lingfei Wu

Abstract Recently Graph Neural Networks (GNNs) have been incorporated into many Computer Vision (CV) models. They not only bring performance improvement to many CV-related tasks but also provide more explainable decomposition to these CV models. This chapter provides a comprehensive overview of how GNNs are applied to various CV tasks, ranging from single image classification to cross-media understanding. It also provides a discussion of this rapidly growing field from a frontier perspective.

Siliang Tang,
College of Computer Science and Technology, Zhejiang University e-mail: siliang@zju.edu.cn

Wenqiao Zhang,
College of Computer Science and Technology, Zhejiang University, e-mail: wenqiaozhang@zju.edu.cn

Zongshen Mu,
College of Computer Science and Technology, Zhejiang University, e-mail: zongshen@zju.edu.cn

Kai Shen,
College of Computer Science and Technology, Zhejiang University, e-mail: shenkai@zju.edu.cn

Juncheng Li,
College of Computer Science and Technology, Zhejiang University, e-mail: junchengli@zju.edu.cn

Jiacheng Li,
College of Computer Science and Technology, Zhejiang University, e-mail: lijiacheng@zju.edu.cn

Lingfei Wu
JD.COM Silicon Valley Research Center, e-mail: lwu@email.wm.edu

20.1 Introduction

Recent years have seen great success of Convolutional Neural Network (CNN) in Computer Vision (CV). However, most of these methods lack the fine-grained analysis of relationships among the visual data (e.g., relation visual regions, adjacent video frames). For example, an image can be represented as a spatial map while the regions in an image are often spatially and semantically dependent. Similarly, video can be represented as spatio-temporal graphs, where each node in the graph represents a region of interest in the video and the edges capture relationships between such regions. These edges can describe the relations and capture the interdependence between nodes in the visual data. Such fine-grained dependencies are critical to perceiving, understanding, and reasoning the visual data. Therefore, graph neural networks can be naturally utilized to extract patterns from these graphs to facilitate the corresponding computer vision tasks.

This chapter introduces the graph neural network model in various computer vision tasks, including specific tasks for image, video and cross-media (cross-modal) (Zhuang et al, 2017). For each task, this chapter demonstrates how graph neural networks can be adapted to and improve the aforementioned computer vision tasks with representative algorithms.

Ultimately, to provide a frontier perspective, we also introduce some other distinctive GNN modeling methods and application scenarios on the subfield.

20.2 Representing Vision as Graphs

In this section, we introduce the representation of visual graph $\mathscr{G}^V = \{\mathscr{V}, \mathscr{E}\}$. We focus on how to construct nodes $\mathscr{V} = \{v_1, v_2, ..., v_N\}$ and edges (or relations) $\mathscr{E} = \{e_1, e_2, ..., e_M\}$ in the visual graph.

20.2.1 Visual Node representation

Nodes are essential entities in a graph. There are three kinds of methods to represent the node of the image $\mathbf{X} \in \mathbb{R}^{h \times w \times c}$ or the video $\mathbf{X} \in \mathbb{R}^{f \times h \times w \times c}$, where (h, w) is the resolution of the original image, c is the number of channels, and f is the number of frames.

Firstly, it is possible to split the image or the frame of the video into regular grids referring to Fig. 20.1, each of which is the (p, p) resolution of the image patch (Dosovitskiy et al, 2021; Han et al, 2020). Then each grid servers as the vertex of the visual graph and apply neural networks to get its embedding.

Secondly, some pre-processed structures like Fig. 20.2 can be directly borrowed for vertex representation. For example, by object detection framework like Faster R-CNN (Ren et al, 2015) or YOLO (Heimer et al, 2019), visual regions in the first

Fig. 20.1: Split an image into fixed-size patches and view as vertexes

column of the figure, have been processed and can be thought of as vertexes in the graph. We map different regions to the same dimensional features and feed them to the next training step. Like the middle column of the figure, scene graph generation models (Xu et al, 2017a; Li et al, 2019i) not only achieve visual detection but also aim to parse an image into a semantic graph which consists of objects and their semantic relationships, where it is tractable to get vertexes and edges to deploy downstream tasks in the image or video. In the last one, human joints linked by skeletons naturally form a graph and learn human action patterns (Jain et al, 2016b; Yan et al, 2018a)

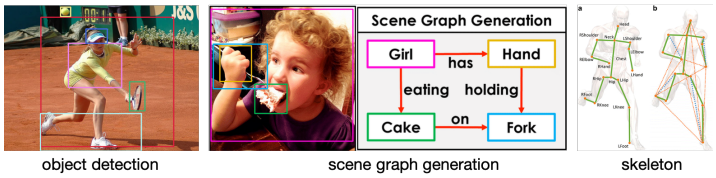

Fig. 20.2: Pre-processed visual graph examples

At last, some works utilize semantic information to represent visual vertexes. Li and Gupta (2018) assigns pixels with similar features to the same vertex, which is soft and likely groups pixels into coherent regions. Pixel features in the group are further aggregated to form a single vertex feature as Fig. 20.3. Using convolutions to learn densely-distributed, low-level patterns, Wu et al (2020a) processes the input image with several convolution blocks and treat these features from various filters as vertexes to learn more sparsely-distributed, higher-order semantic concepts. A point cloud is a set of 3D points recorded by LiDAR scans. Te et al (2018) and Landrieu and Simonovsky (2018) aggregate k-nearest neighbor to form superpoint (or vertex) and build their relations by ConvGNNs to explore the topological structure and 'see' the surrounding environment.

Fig. 20.3: Grouping similar pixels as vertexes (different colors)

20.2.2 Visual Edge representation

Edges depict the relations of nodes and play an important role in graph neural networks.For a 2D image, the nodes in the image can be linked with different spatial relations. For a clip of video stacked by continuous frames, it adds temporal relations between frames besides spatial ones within the frame. On the one hand, these relations can be fixed by predefined rules to train GNNs, referred to as static relations. Learning to learn relations (thought of as dynamic relations) attracts more and more attention on the other hand.

20.2.2.1 Spatial Edges

To capture spatial relations is the key step in the image or video. For static methods, generating scene graphs (Xu et al, 2017a) and human skeletons (Jain et al, 2016b) are natural to choose edges between nodes in the visual graph described in the Fig. 20.2. Recently, some works (Bajaj et al, 2019; Liu et al, 2020g) use fully-connected graph (every vertex is linked with other ones) to model the relations among visual nodes and compute union region of them to represent edge features. Furthermore, self-attention mechanism (Yun et al, 2019; Yang et al, 2019f) are introduced to learn the relations among visual nodes, whose main idea is inspired by transformer (Vaswani et al, 2017) in NLP. When edges are represented, we can choose either spectral-based or spatial-based GNNs for applications (Zhou et al, 2018c; Wu et al, 2021d).

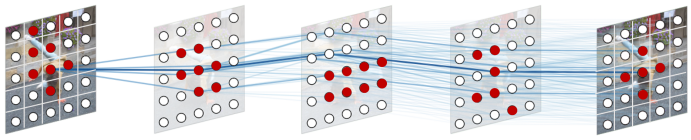

Fig. 20.4: A spatial-temporal graph by extracting nodes from each frame and allowing directed edges between nodes in neighbouring frames

20.2.2.2 Temporal Edges

To understand the video, the model not only builds spatial relations in a frame but also captures temporal connections among frames. A series of methods (Yuan et al, 2017; Shen et al, 2020; Zhang et al, 2020h) compute each node in the current frame with near frames by semantic similarity methods like k-Nearest Neighbors to construct temporal relations among frames. Especially, as you can see in the Fig. 20.4, Jabri et al (2020) represent video as a graph using a Markov chain and learn a random walk among nodes by dynamic adjustment, where nodes are image patches, and edges are affinities (in some feature space) between nodes of neighboring frames. Zhang et al (2020g) use regions as visual vertexes and evaluate the IoU (Intersection of Union) of nodes among frames to represent the weight edges.

20.3 Case Study 1: Image

20.3.1 Object Detection

Object detection is a fundamental and challenging problem in computer vision, which received great and lasting attention in recent years. Given a natural image, the object detection task seeks to locate the visual object instances from certain categories (e.g. humans, animals, or trees). Generally speaking, object detection can be grouped into two categories (Liu et al, 2020b): 1) generic object detection and 2) salient object detection. The first class aims to detect unlimited instances of objects in the digital image and predict their class attributes from some pre-defined categories. The goal of the second type is to detect the most salient instance. In recent years deep learning-based methods have achieved tremendous success in this field, such as Faster-RCNN (Ren et al, 2015), YOLO (Heimer et al, 2019), and etc. Most of the early methods and their follow-ups (Ren et al, 2015; He et al, 2017a) usually adopt the region selection module to extract the region features and predict the active probability for each candidate region. Although they are demonstrated successful, they mostly treat the recognition of each candidate region separately, thus leading to nonnegligible performance drops when facing the nontypical and nonideal occasions, such as heavy long-tail data distributions and plenty of confusing categories (Xu et al, 2019b). The graph neural network (GNN) is introduced to effectively address this troublesome challenge by modeling the correlations between regions explicitly and leveraging them to achieve better performance. In this section, we will present one typical case SGRN (Xu et al, 2019b) to discuss this promising direction.

The SGRN can be simply divided into two modules: 1) sparse graph learner which learns the graph structure explicitly during the training and 2) the spatial-aware graph embedding module which leverages the learned graph structure information and obtains the graph representation. To make it clear, we denote the graph

as $\mathscr{G}(\mathscr{V},\mathscr{E})$, where $\mathscr{V}$ is the vertex set and $\mathscr{E}$ is the edge set. The image is $\mathscr{I}$. And we formulate the regions as $R = \{\mathbf{f}_i\}_{i=1}^{n_\mathscr{I}}, \mathbf{f}_i \in \mathbb{R}^d$ for a specific image $\mathscr{I}$, where d is the region feature's dimension. We will discuss these two parts and omit other details.

Unlike previous attempts in close fields which build category-to-category graph (Dai et al, 2017; Niepert et al, 2016), the SGRN treats the candidate regions R as graph nodes $\mathscr{V}$ and constructs dynamic graph $\mathscr{G}$ on top of them. Technically, they project the region features into the latent space $\mathbf{z}$ by:

$$\mathbf{z}_i = \phi(\mathbf{f}_i) \tag{20.1}$$

where ϕ is the two fully-connected layers with ReLU activation, $\mathbf{z}_i \in \mathbb{R}^l$ and l is the latent dimension.

The region graph is constructed by latent representation $\mathbf{z}$ as follows:

$$S_{i,j} = \mathbf{z}_i \mathbf{z}_j^\top \tag{20.2}$$

where $S \in \mathbb{R}^{n_r \times n_r}$. It is not proper to reserve all relations between region pairs since there are many negative (i.e., background) samples among the region proposals, which may affect the down task's performance. If we use the dense matrix S as the graph adjacency matrix, the graph will be fully-connected, which leads to computation burden or performance drop since most existing GNN methods work worse on fully-connected graphs (Sun et al, 2019). To solve this issue, the SGRN adopt KNN to make the graph sparse (Chen et al, 2020n,o). In other words, for the learned similarity matrix $S_i \in \mathbb{R}^{N_r}$, they only keep the K nearest neighbors (including itself) as well as the associated similarity scores (i.e., they mask off the remaining similarity scores). The learned graph adjacency is denoted as:

$$A = \mathrm{KNN}(S) \tag{20.3}$$

The node's initial embedding is obtained by the pre-trained visual classifier. We omit the details and simply denote it as $X = \{\mathbf{x}_i\}_{i=1}^{n_r}$. The SGRN introduces a spatial-aware graph reasoning module to learn the spatial-aware node embedding. Formally, they introduce a patch of operator adapted by graph convolutional network (GCN) with learnable gaussian kernels, given by:

$$f_k'(i) = \sum_{j \in \mathscr{N}(i)} \omega_k(\mu(i,j))\mathbf{x}_j A_{i,j} \tag{20.4}$$

where $\mathscr{N}(i)$ denotes the neighborhood of node i, $\mu(i,j)$ is the distance of node i,j calculated by the center of them in a polar coordinate system, and $\omega_k()$ is the $k-$th gaussian kernel. Then the K kernels' results are concatenated together and projected to the latent space as follows:

$$\mathbf{h}_i = g([f_1'(i); f_2'(i); ...; f_K'(i)]) \tag{20.5}$$

where $g(\cdot)$ denotes the projection with non-linearity. Finally, $\mathbf{h}_i$ is combined with the original visual region feature $\mathbf{f}_i$ to enhance classification and regression performance.

20.3.2 Image Classification

Inspired by the success of deep learning techniques, significant improvement has been made in the image classification field, such as ResNet (He et al, 2016a). However, the CNN-based models are limited in modeling relations between samples. The graph neural network is introduced to image classification, which aims to model the fine-grained region correlations to enhance classification performance (Hong et al, 2020a), combining labeled and unlabeled image instances for semi-supervised image classification (Luo et al, 2016; Satorras and Estrach, 2018). In this section, we will present a typical case for semi-supervised image classification to show the effectiveness of GNN.

We denote the data samples as $(x_i, y_i) \in \mathscr{T}$, where x_i is the image and $y_i \in \mathbb{R}^K$ is the image label. For semi-supervised setting, the $\mathscr{T}$ is divided into labeled part $\mathscr{T}_{labeled}$ and unlabeled part $\mathscr{T}_{unlabeld}$. We assume that there are N_l labeled samples and N_u unlabeled samples, respectively. The proposed GNN is dynamic and multi-layer, which means for each layer, it will learn the graph topology from the previous layer's the node embedding and learn the new embedding on top of it. Thus, we denote the layer number as M and only present the detailed graph construction and graph embedding techniques of layer k. Technically, they construct the graph for the image set and formulate the posterior prediction task as message passing with graph neural network. They cast the samples as graph $\mathscr{G}(\mathscr{V}, \mathscr{E})$, whose nodes set is the image set consisting of both labeled and unlabeled data. The edge set $\mathscr{E}$ is constructed during training.

First, they denote the initial node representation as $X = \{\mathbf{x}_i\}_{i=1}^{n_l+n_u}$ as follows:

$$\mathbf{x_i}^0 = (\phi(x_i), h(y_i)) \tag{20.6}$$

where $\phi()$ is the convolutional neural network and $h()$ is the one-hot label encoding. Note that for unlabeled data, they replace the $h()$ with the uniform distribution over the K-simplex.

Second, the graph topology is learned by current layer's node embedding denoted as $\mathbf{x}^k$. The distance matrix modeling the distance in the embedding space between nodes is denoted as S given by:

$$S_{i,j}^k = \varphi(\mathbf{x}_i, \mathbf{x}_j) \tag{20.7}$$

where φ is a parametrized symmetric function as follows:

$$\varphi(\mathbf{a}, \mathbf{b}) = MLP(abs(\mathbf{a} - \mathbf{b})) \tag{20.8}$$

where $MLP()$ is a multilayer perceptron network and $abs()$ is the absolute function. Then the adjacency matrix A is calculated by normalizing the row of S using softmax operation.

Then a GNN layer is adapted to encode the graph nodes with learned topology A. The GNN layer receives the node embedding matrix $\mathbf{x}^k$ and outputs the aggregated node representation $\mathbf{x}^{k+1}$ as:

$$\mathbf{x}_l^{k+1} = \rho(\sum_{B \in A} B\mathbf{x}^k \theta_{B,l}^k), l = d_1 ... d_{k+1} \tag{20.9}$$

where $\{\theta_1^k, ..., \theta_{|A|}^k\}$ are trainable parameters, and $\rho()$ is non-linear activate function (leaky ReLU here).

The graph neural network is effective in modeling the unstructured data's correlation. In this work, the GNN explicitly exploits the relation between samples, especially the labeled and unlabeled data, contributing to few-shot image classification challenges.

20.4 Case Study 2: Video

20.4.1 Video Action Recognition

Action recognition in video is a highly active area of research, which plays a crucial role in video understanding. Given a video as input, the task of action recognition is to recognize the action appearing in the video and predict the action category. Over the past few years, modeling the spatio-temporal nature of video has been the core of research in the field of video understanding and action recognition. Early approaches of activity recognition such as Hand-crafted Improved Dense Trajectory(iDT) (Wang and Schmid, 2013), two-Stream ConvNets (Simonyan and Zisserman, 2014a), C3D (Tran et al, 2015), and I3D (Carreira and Zisserman, 2017) have focused on using spatio-temporal appearance features. To better model longer-term temporal information, researchers also attempted to model the video as an ordered frame sequence using Recurrent Neural Networks (RNNs) (Yue-Hei Ng et al, 2015; Donahue et al, 2015; Li et al, 2017b). However, these conventional deep learning approaches only focus on extracting features from the whole scenes and are unable to model the relationships between different object instances in space and time. For example, to recognize the action in the video corresponds to "opening a book", the temporal dynamics of objects and human-object and object-object interactions are crucial. We need to temporally link book regions across time to capture the shape of the book and how it changes over time.

To capture relations between objects across time, several deep models (Chen et al, 2019d; Herzig et al, 2019; Wang and Gupta, 2018; Wang et al, 2018e) have been recently introduced that represent the video as spatial-temporal graph and leverage recently proposed graph neural networks. These methods take dense ob-

ject proposals as graph nodes and learn the relations between them. In this section, we take the framework proposed in (Wang and Gupta, 2018) as one example to demonstrate how graph neural networks can be applied to action recognition task.

As illustrated in Fig 20.5, the model takes a long clip of video frames as input and forwards them to a 3D Convolutional Neural Network to get a feature map $I \in \mathbb{R}^{t\times h\times w\times d}$, where t represents the temporal dimension, $h \times w$ represents the spatial dimensions and d represents the channel number. Then the model adopts the Region Proposal Network (RPN) (Ren et al, 2015) to extract the object bounding boxes followed by RoIAlign (He et al, 2017a) extracting d-dimension feature for each object proposal. The output n object proposals aggregated over t frames are corresponding to n nodes in the building graphs. There are mainly two types of graphs: Similarity Graph and Spatial-Temporal Graph.

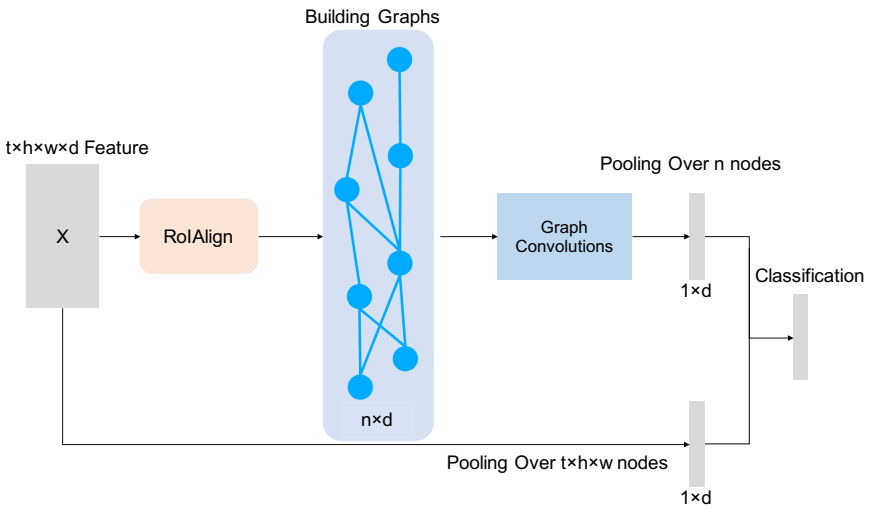

Fig. 20.5: Overview of the GNN-based model for Video Action Recognition.

The similarity graph is constructed to measure the similarity between objects. In this graph, pairs of semantically related objects are connected. Formally, the pairwise similarity between every two nodes can be represented as:

$$F(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)^\top \phi'(\mathbf{x}_j) \tag{20.10}$$

where ϕ and ϕ' represent two different transformations of the original features.

After computing the similarity matrix, the normalized edge values A_{ij}^{sim} from node i to node j can be defined as:

$$A_{ij}^{sim} = \frac{expF(\mathbf{x}_i, \mathbf{x}_j)}{\sum_{j=1}^{n} expF(\mathbf{x}_i, \mathbf{x}_j)} \tag{20.11}$$

The spatial-temporal graph is proposed to encode the relative spatial and temporal relations between objects, where objects in nearby locations in space and time are connected together. The normalized edge values of the spatial-temporal graph can be formulated as:

$$A_{ij}^{front} = \frac{\sigma_{ij}}{\sum_{j=1}^{n} \sigma_{ij}} \tag{20.12}$$

where G^{front} represents the forward graph which connects objects from frame t to frame t + 1, and σ_{ij} represents the value of Intersection Over Unions (IoUs) between object i in frame t and object j in frame $t+1$. The backward graph A^{back} can be computed in a similar way. Then, the Graph Convolutional Networks (GCNs) (Kipf and Welling, 2017b) is applied to update features of each object node. One layer of graph convolutions can be represented as:

$$Z = AXW \tag{20.13}$$

where A represents one of the adjacency matrix (A^{sim}, A^{front}, or A^{back}), X represents the node features, and W is the weight matrix of the GCN.

The updated node features after graph convolutions are forwarded to an average pooling layer to obtain the global graph representation. Then, the graph representation and pooled video representation are concatenated together for video classification.

20.4.2 Temporal Action Localization

Temporal action localization is the task of training a model to predict the boundaries and categories of action instances in untrimmed videos. Most existing methods (Chao et al, 2018; Gao et al, 2017; Lin et al, 2017; Shou et al, 2017, 2016; Zeng et al, 2019) tackle temporal action localization in a two-stage pipeline: they first generate a set of 1D temporal proposals and then perform classification and temporal boundary regression on each proposal individually. However, these methods process each proposal separately, failing to leverage the semantic relations between proposals. To model the proposal-proposal relations in the video, graph neural networks are then adopted to facilitate the recognition of each proposal instance. P-GCN (Zeng et al, 2019) is recently proposed method to exploit the proposal-proposal relations using Graph Convolutional Networks. P-GCN first constructs an action proposal graph, where each proposal is represented as a node and their relations between two proposals as an edge. Then P-GCN performs reasoning over the proposal graph using GCN to model the relations among different proposals and update their representations. Finally, the updated node representations are used to refine their boundaries and classification scores based on the established proposal-proposal dependencies.

20.5 Other Related Work: Cross-media

Graph-structured data widely exists in different modal data (images, videos, texts), and is used in existing cross-media tasks (e.g., *visual caption, visual question answer, cross-media retrieval*). In other words, using of graph structure data and GNN rationally can effectively improve the performance of cross-media tasks.

20.5.1 Visual Caption

Visual caption aims at building a system that automatically generates a natural language description of a given image or video. The problem of image captioning is interesting not only because it has important practical applications, such as helping visually impaired people see, but also because it is regarded as a grand challenge for vision understanding. The typical solutions of visual captioning are inspired by machine translation and equivalent to translating an image to a text. In these methods (Li et al, 2017d; Lu et al, 2017a; Ding et al, 2019b), Convolutional Neural Network (CNN) or Region-based CNN (R-CNN) is usually exploited to encode an image and a decoder of Recurrent Neural Network (RNN) w/ or w/o attention mechanism is utilized to generate the sentence. However, a common issue not fully studied is how visual relationships should be leveraged in view that the mutual correlations or interactions between objects are the natural basis for describing an image.

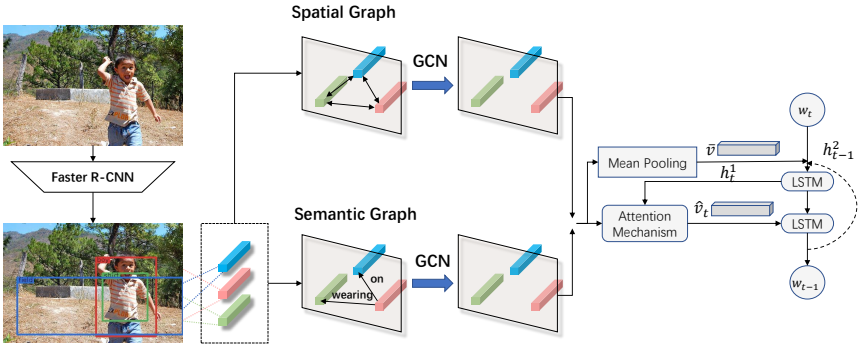

Fig. 20.6: Framework of GCN-LSTM.

In recent years, Yao et al (2018) presented Graph Convolutional Networks plus Long Short-Term Memory (GCN-LSTM) architecture, which explores visual relationship for boosting image captioning. As shown in Fig. 20.6, they study the problem from the viewpoint of modeling mutual interactions between objects/regions to enrich region-level representations that are feed into sentence decoder. Specifically,

they build two kinds of visual relationships, i.e., semantic and spatial correlations, on the detected regions, and devised Graph Convolutions on the region-level representations with visual relationships to learn more powerful representations. Such relation-aware region-level representations are then input into attention LSTM for sentence generation.

Then, Yang et al (2019g) presented a novel Scene Graph Auto-Encoder (SGAE) for image captioning. This captioning pipeline contains two step: 1) extracting the scene graph for an image and using GCN to encode the corresponding scene graph, then decoding the sentence by the recoding representation; 2) incorporating the image scene graph to the captioning model. They also use GCNs to encode the visual scene graph . Given the representation of visual scene graph, they introduce joint visual and language memory to choose appropriate representation to generate image description.

20.5.2 Visual Question Answering

Visual Question Answering (VQA) aims at building a system that automatically answers natural language questions about visual information. It is a challenging task that involves mutual understanding and reasoning across different modalities. In the past few years, benefiting from the rapid developments of deep learning, the prevailing image and video question methods (Shah et al, 2019; Zhang et al, 2019g; Yu et al, 2017a) prefer to represent the visual and linguistic modalities in a common latent subspace, use the encoder-decoder framework and attention mechanism, which has made remarkable progress.

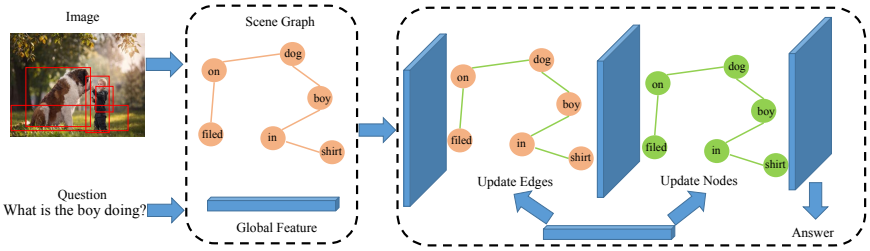

Fig. 20.7: GNN-based Visual QA.

However, the aforementioned methods also have not considered the graph information in the VQA task. Recently, Zhang et al (2019a) investigates an alternative approach inspired by conventional QA systems that operate on knowledge graphs. Specifically, as shown in Fig. 20.7, they investigate the use of scene graphs derived from images, then naturally encode information on graphs and perform structured reasoning for Visual QA. The experimental results demonstrate that scene graphs,

even automatically generated by machines, can definitively benefit Visual QA if paired with appropriate models like GNNs. In other words, leveraging scene graphs largely increases the Visual QA accuracy on questions related to counting, object presence and attributes, and multi-object relationships.

Another work (Li et al, 2019d) presents the Relation-aware Graph Attention Network (ReGAT), a novel framework for VQA, to model multi-type object relations with question adaptive attention mechanism. A Faster R-CNN is used to generate a set of object region proposals, and a question encoder is used for question embedding. The convolutional and bounding-box features of each region are then injected into the relation encoder to learn the relation-aware, question-adaptive, region-level representations from the image. These relation-aware visual features and the question embeddings are then fed into a multimodal fusion module to produce a joint representation, which is used in the answer prediction module to generate an answer.

20.5.3 Cross-Media Retrieval

Image-text retrieval task has become a popular cross-media research topic in recent years. It aims to retrieve the most similar samples from the database in another modality. The key challenge here is how to match the cross-modal data by understanding their contents and measuring their semantic similarity. Many approaches (Faghri et al, 2017; Gu et al, 2018; Huang et al, 2017b) have been proposed. They often use global representations or local to express the whole image and sentence. Then, a metric is devised to measure the similarity of a couple of features in different modalities. However, the above methods lose sight of the relationships between objects in multi-modal data, which is also the key point for image-text retrieval.

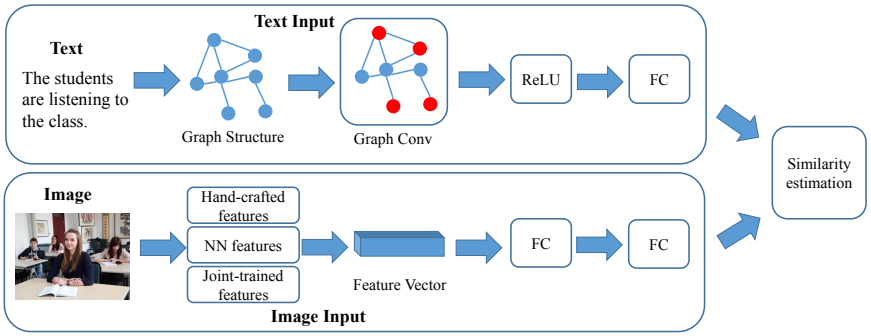

Fig. 20.8: Overview of dual-path neural network for Image-text retrieval.

To utilize the graph data in image and text better, as shown in Fig. 20.8, Yu et al (2018b) proposes a novel cross-modal retrieval model named dual-path neural network with graph convolutional network. This network takes both irregular graph-structured textual representations and regular vector-structured visual representations into consideration to jointly learn coupled feature and common latent semantic space.

In addition, Wang et al (2020i) extract objects and relationships from the image and text to form the visual scene graph and text scene graph, and design a so-called Scene Graph Matching (SGM) model, where two tailored graph encoders encode the visual scene graph and text scene graph into the feature graph. After that, both object-level and relationship-level features are learned in each graph, and the two feature graphs corresponding to two modalities can be finally matched at two levels more plausibly.

20.6 Frontiers for Graph Neural Networks on Computer Vision

In this section, we introduce the frontiers for GNNs on Computer Vision. We focus on the advanced modeling methods of GNN for Computer Vision and their applications in a broader area of the subfield.

20.6.1 Advanced Graph Neural Networks for Computer Vision

The main idea of the GNN modeling method on CV is to represent visual information as a graph. It is common to represent pixels, object bounding boxes, or image frames as nodes and further build a homogeneous graph to model their relations. Despite this kind of method, there are also some new ideas for GNN modeling.

Considering the specific task nature, some works try to represent different forms of visual information in the graph.

- **Person Feature Patches** Yan et al (2019); Yang et al (2020b); Yan et al (2020b) build spatial and temporal graphs for person re-identification (Re-ID). They horizontally partition each person feature map into patches and use the patches as the nodes of the graph. GCN is further used to modeling the relation of body parts across frames.
- **Irregular Clustering Regions** Liu et al (2020h) introduce the bipartite GNN for mammogram mass detection. It first leverages kNN forward mapping to partition an image feature map into irregular regions. Then the features in an irregular region are further integrated as a node. The bipartite node sets are constructed by cross-view images respectively, while the bipartite edge learns to model both inherent cross-view geometric constraints and appearance similarities.

- **NAS Cells** Lin et al (2020c) proposed graph-guided Neural Architecture Search (NAS) algorithms. The proposed models represent an operation cell as a node and apply the GCNs to model the relationship of cells in network architecture search.

20.6.2 Broader Area of Graph Neural Networks on Computer Vision

In this subsection, we introduce some other application scenarios of GNNs on CV, including but not limited to the following:

- **Point Cloud Analysis** Point Cloud Analysis aims to recognize a set of points in a coordinate system. Each point is represented by its three coordinates with some other features. In order to utilize CNN, the early works (Chen et al, 2017; Yan et al, 2018b; Yang et al, 2018a; Zhou and Tuzel, 2018) convert a point cloud to a regular grid such as image and voxel. Recently, a series of works (Chen et al, 2020g; Lin et al, 2020f; Xu et al, 2020e; Shi and Rajkumar, 2020; Shu et al, 2019) use a graph representation to preserve the irregularity of a point cloud. GCN plays a similar role as CNN in image processing for aggregating local information. Chen et al (2020g) develops a hierarchical graph network structure for 3D object detection on point clouds. Lin et al (2020f) proposes a learnable GCN kernel and a 3D graph max pooling with a receptive field of K nearest neighboring nodes. Xu et al (2020e) proposes a Coverage-Aware Grid Query and a Grid Context Aggregation to accelerate 3D scene segmentation. Shi and Rajkumar (2020) designs a Point-GNN with an auto-registration mechanism to detect multiple objects in a single shot.
- **Low Resource Learning** Low-resource learning models the ability of learning from a very small amount of data or transferring from prior. Some works leverage GNN to incorporate structural information for the low-resource image classification. Wang et al (2018f); Kampffmeyer et al (2019) use knowledge graphs as extra information to guide zero-short image classification. Each node corresponds to an object category and takes the word embeddings of nodes as input for predicting the classifier of different categories. Except for the knowledge graph, the similarity between images in the dataset is also helpful for the few-shot learning. Garcia and Bruna (2017); Liu et al (2018e); Kim et al (2019) set up similarity metrics and further modeling the few-shot learning problem as a label propagating or edge-labeling problem.
- **Face Recognition** Wang et al (2019p) formulates the face clustering task as a link prediction problem. It utilizes the GCN to infer the likelihood of linkage between pairs in the face sub-graphs. Yang et al (2019d) proposes a proposal-detection-segmentation framework for face clustering on an affinity graph. Zhang et al (2020b) propose a global-local GCN to perform label cleansing for face recognition.

- **Miscellaneous** We also introduce some distinctive GNN applications on the subfield. Wei et al (2020) proposes a view-GCN to recognizes 3D shape through its projected 2D images. Wald et al (2020) extends the concept of scene graph to the 3D indoor scene. Ulutan et al (2020) leverage GCNs to reason the interactions between humans and objects. Cucurull et al (2019) predicts fashion compatibility between two items by formulating an edge prediction problem. Sun et al (2020b) builds a social behavior graph from a video and uses GNNs to propagate social interaction information for trajectory prediction. Zhang et al (2020i) builds a vision and language relation graph to alleviate the hallucination problem in the grounded video description task.

20.7 Summary

This chapter shows that GNN is a promising and fast-developing research field that offers exciting opportunities in computer vision techniques. Nevertheless, it also presents some challenges. For example, graphs are often related to real scenarios, while the aforementioned GNNs lack interpretability, especially the decision-making problems (e.g., medical diagnostic model) in the computer vision field. However, compared to other black-box models (e.g., CNN), interpretability for graph-based deep learning is even more challenging since graph nodes and edges are often heavily interconnected. Thus, a further direction worth exploring is how to improve the interpretability and robustness of GNN for computer vision tasks.

Editor's Notes: Convolutional Neural Network has achieved huge success in computer vision domain. However, recent years have seen the rise of relational machine learning like GNNs and Transformers to modeling more fine-grained correlations in both images and videos. Certainly, graph structure learning techniques in Chapter 14 becomes very important for constructing an optimized graph from an image or a video and learning node representations on this learnt implicit graph. Dynamic GNNs in Chapter 15 will play an important role when coping with a video. GNN Methods in Chapter 4 and GNN Scalability in Chapter 6 are then another two basic building blocks for the use of GNNs for CV. This chapter is also highly correlated with the Chapter 21 (GNN for NLP) since vision and language is a fast-growing research area and multi-modality data is widely used today.