Journal of Computer Science and Cybernetics, V.40, N.4 (2024), 327-346
DOI: 10.15625/1813-9663/20929

OD-VR-CAP: IMAGE CAPTIONING BASED ON DETECTING AND PREDICTING RELATIONSHIPS BETWEEN OBJECTS

NGUYEN VAN THINH[1,2,3], TRAN VAN LANG[4,*], VAN THE THANH[3]

[1] *Institute of Mechanics and Applied Informatics, Vietnam Academy of Science and Technology (VAST), 291 Dien Bien Phu Street, 3 District, Ho Chi Minh City, Viet Nam*
[2] *Graduate University of Science and Technology, Vietnam Academy of Science and Technology (VAST), 18 Hoang Quoc Viet Street, Cau Giay District, Ha Noi, Viet Nam*
[3] *Faculty of Information Technology, HCMC University of Education (HCMUE), 280 An Duong Vuong, 5 District, Ho Chi Minh City, Viet Nam*
[4] *Journal Editorial Department, HCMC University of Foreign Languages and Information Technology (HUFLIT), 828 Su Van Hanh, 10 District, Ho Chi Minh City, Viet Nam*

Abstract. Recent image captioning works often focus on global features or individual object regions within the image without exploiting the relational information between them, resulting in limited accuracy. In this paper, the proposed image captioning model leverages the relationships between objects in the image to fully understand the content and improve accuracy. The approach goes through the following steps: First, objects in the image are detected using an object detection model combined with a graph convolutional network (GCN). From this, a relationship prediction model based on relational context information and knowledge is proposed to classify relationships between objects to create a relationship graph to represent the image. Subsequently, a dual attention mechanism is built to enable the model to focus on relevant parts of both object regions and vertices in the relationship graph when generating captions. Finally, an LSTM network with dual attention is trained to generate captions relying on the image representation and given captions. Experiments conducted on MS COCO and Visual Genome datasets demonstrate that the proposed model achieves higher accuracy compared to baseline methods and some recently published works.

Keywords. Image captioning, object detection, visual relationship, attention mechanism, deep neural network.

1. INTRODUCTION

Automatic image captioning automatically generates descriptions of image content in natural language, accurately depicting the main content of the image, such as objects and their relationships [1]. This task combines two popular fields in artificial intelligence: computer vision, which addresses image understanding, and natural language processing, which generates syntactically and semantically correct image descriptions [2]. This problem is gaining investigative interest due to its numerous practical applications.

Most recent successful image captioning techniques use deep neural network models with attention mechanisms and follow an encoder-decoder framework [3]. This framework con-

*Corresponding author.
E-mail addresses: thinhnv@hcmue.edu.vn (N.V.Thinh); langtv@huflit.edu.vn (T.V.Lang); thanhvt@hcmue.edu.vn (V.T.Thanh)

sists of two main parts: an encoder to learn representations that convert image content into feature vectors and a language model that acts as the decoder to generate captions for the image, where the input to the decoder is the feature vectors obtained from the encoder [4]. These models typically represent images as feature vectors using pre-trained CNNs. Therefore, These models typically represent images as feature vectors using pre-trained CNNs. Therefore, they need help understanding the semantics of individual objects and their relationships within the image, and they hardly align different parts of the input image with the words in the caption [5]. Thus, extracting objects in the image and determining their relationships is necessary. This allows the encoder to fully represent the image information to serve as input for the decoder, thereby improving caption accuracy.. Moreover, attention-based image captioning models help the decoder focus only on relevant parts when generating captions rather than using the entire image feature, which has proven effective [3]. However, most works use the visual features of parts of the image while ignoring the semantics of class labels and the relationship information of these objects. Therefore, a dual attention mechanism is necessary to combine attention to visual features and the semantics of object class labels.

To address the above mentioned challenges, this paper introduces a novel approach to enhancing image captioning accuracy. We do this by constructing a relationship graph that encapsulates the image content and leveraging a dual attention mechanism. The relationship graph is formed through object detection and relationship prediction, providing a comprehensive understanding of the image content and thereby improving caption accuracy. The dual attention mechanism, which exploits both visual features and object semantics, further enhances accuracy. The key contributions of this paper are:

- A novel model that significantly improves object detection accuracy is proposed. This model integrates graph convolutional networks, making it easily adaptable to other object detection techniques.

- A relationship prediction model for objects in the image is constructed based on object region features, relational context information, and relational knowledge between objects in the dataset. From there, A relationship graph is created to exhaustively depict the image's semantic content.

- A method for representing an image's relationship graph as feature vectors is introduced. This method leverages the semantics of the relationships based on graph convolutional networks.

- A dual attention mechanism that focuses on the features of object regions and nodes in the image's relationship graph is proposed. Then, an image captioning model based on the image's relationship graph and an LSTM network with dual attention is built.

The above discusses several issues related to the problem of image captioning based on objects and their relationships within the image. The remainder of the paper is organized as follows: Section 2 presents and discusses related works to identify the challenges posed by the problem; Section 3 details the proposed method of this paper. Several experiments and results are described in Section 4, and the final section provides some conclusions.

2. RELATED WORK

Recent image captioning studies based on deep learning predominantly follow two main approaches: CNN-LSTM-based methods and graph-based methods. Additionally, the attention mechanism is adopted in both approaches to enhance the decoder's effectiveness in generating captions.

2.1. CNN-LSTM-based methods

Hossain et al. [6] proposed an image captioning model based on an attention mechanism using DenseNet features. DenseNet extracts various feature samples as inputs for an LSTM with an attention mechanism to focus on relevant parts of the image when generating each word in the caption. The model was evaluated on the MS COCO dataset and achieved promising results on the BLEU-2, 3, and 4 metrics. Patwari et al. [7] introduced an image captioning method within an encoder-decoder framework. A pre-trained CNN (Inception-v3) is used as the encoder, and GRU (a simplified version of LSTM) as the decoder with an attention mechanism. Experimental results on the MS COCO dataset, evaluated by BLEU 1-4 scores, showed the effectiveness of this approach. However, these works have the disadvantage of only using pre-trained CNNs to extract image representation features. Therefore, it is difficult to fully recognize all objects in the image and their relationships to comprehensively represent the image's semantic content. Thinh et al. [8] proposed an image captioning model based on object detection and the attention mechanism. In this model, the input image undergoes object detection and feature extraction, where regional features and corresponding object class labels are used as input for the LSTM network with an improved attention mechanism to generate captions. The method was evaluated on the MS COCO dataset, demonstrating its effectiveness. Xie et al. [9] developed a model to enhance image captioning accuracy using Bidirectional LSTM and an attention mechanism. Input images are extracted for object region features with Faster R-CNN, and these features are fed into Bidirectional LSTM to generate captions. This model was evaluated on the Flickr 30K and MS COCO datasets, showing higher accuracy than the baseline and some recently published works. However, this work's limitation is that it only extracts object regions in images without exploiting the relationships between objects to represent the image semantics comprehensively, thereby improving caption accuracy.

2.2. Graph-based methods

Yao et al. [10] introduced an automatic image captioning model using R-CNN to extract objects in the image based on region features and LSTM with an attention mechanism to improve caption accuracy. The feature extraction process exploits semantic and spatial relationships between objects in the image using a Graph Convolutional Network (GCN). It uses spatial and semantic graphs created based on the objects and their relationships. However, this model has the limitation that the object detection process uses Faster R-CNN, which leads to struggles to accurately detect objects in images with many objects and complex details. Additionally, the spatial relationships are limited to four types, as proposed by the authors, which may only partially capture the complexity of real-world images. Chen et al. [11] proposed an Abstract Scene Graph structure from ground truth captions to control diverse and detailed image caption generation as the user desires. Yan et al. [12] improved the method [11] by combining transformer blocks with two LSTM modules to create smoother and more coherent captions. Specifically, the first LSTM layer integrates multimodal data, incorporating visual and textual features; the second LSTM layer generates captions, while the transformer block determines the contribution of different features to caption prediction. Both works exploit ground truth captions through abstract scene graphs to enhance the quality of the output captions. However, both also have the drawback of not fully exploiting the input image content, particularly the relationships between objects, leading to lower performance on some standard metrics.

From the survey and analysis of related works, it is evident that deep learning-based image captioning, particularly the approach of exploiting object information and their relationships to represent the image's semantic content fully, has been published by many research groups and shown to be effective. Moreover, the attention mechanism allows decoders to focus on essential image parts while ignoring redundant and noisy information to generate more accurate captions, proving feasible and practical. Building upon existing works and addressing the limitations of related published methods, this paper proposes a novel image captioning model. It is based on object detection, relationship prediction between objects in images, and LSTM with a dual attention mechanism, which overcomes the limitations of existing models and is expected to improve accuracy significantly.

3. PROPOSED METHOD

In this paper, the proposed image captioning method follows an encoder-decoder framework as shown in Figure 1, consisting of two main parts: an Image encoder, which functions to learn representations to fully capture the content of the image, and a Language decoder, which aims to generate captions based on the features obtained from the encoder and the ground truth captions.

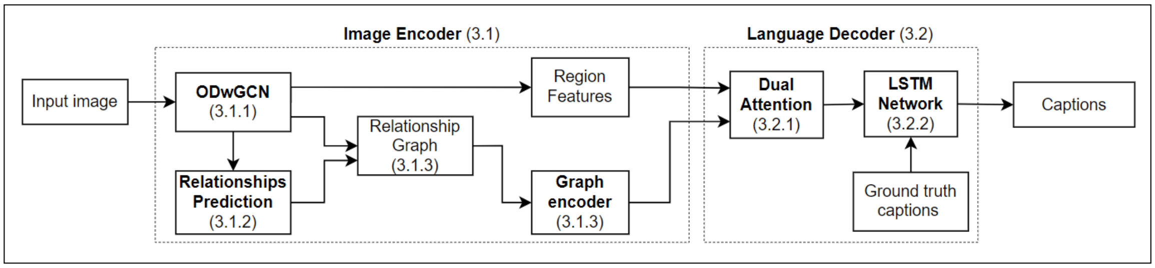

Figure 1: Overview diagram of the proposed image captioning method

To explicitly present the proposed method, the following symbols and corresponding definitions are used:

- The dataset for the image captioning task consists of N_T data samples, each data sample is a pair of (I, S). Here, I is the given image, and S is the caption of the image with $S = \{s_1, s_2, ..., s_{N_S}\}$, where s_i represents the i^{th} in the sentence, $\forall i = \overline{1, N_S}$.

- $\mathcal{B}$ and N_B are the set of object regions and the number of detected object regions (detected boxes) in the image, respectively.

- $\mathcal{C}$ and N_C are the set of class labels and the number of object class labels in the dataset, respectively.
- Y is the confidence matrix obtained from the results of the pre-trained object detection model, and $\hat{Y}$ is the confidence matrix after adjustment by the GCN network.
- $\mathcal{K}$ is the knowledge base containing N_E entities, including subjects, objects, and predicates, E is the set of entities.
- $\mathcal{R}$ and $N_{\mathcal{R}}$ are the set of relationships and the number of relationships in $\mathcal{K}$, respectively.
- $G^{(1)}, G^{(2)}, \mathcal{G}$ and $\mathcal{G}^*$ are the label correlation graph, entity graph, relationship graph, and extended relationship graph, respectively, $X^{(1)}$ and $X^{(2)}$ are the feature matrices of $G^{(1)}$ and $G^{(2)}$, X is the feature matrix of $\mathcal{G}^*$.
- $h_v^{(l)}$ is the hidden state of node v at layer l of the GCN, h_t is the hidden state of the LSTM cell at time step t.
- There are 3 loss functions, denoted as $L_1(\varphi))$ for the image captioning task, $L_2(\varphi)$ for the supervised training of the GraphSAGE network on the graph $G^{(1)}$ based on the labels of the object regions in the image, and $L_3(\varphi)$ for the unsupervised training of the GraphSAGE network based on the similarity of neighboring nodes in the graphs $G^{(2)}$ and $\mathcal{G}^*$. Where

$$L_1(\varphi) = -\frac{1}{N_T}\sum_{i=1}^{N}\sum_{t=1}^{N_S^{(i)}} \log P(s_t^{(i)}|s_1^{(i)}, s_2^{(i)}, ..., s_{(t-1)}^{(i)}, f_I^{(i)}; \varphi). \tag{1}$$

In (1), $N_S^{(i)}, s_t^{(i)}$ and $f_I^{(i)}$ are respectively the number of words in the caption, the correct word at time step t, and the image feature of the i^{th} data sample; $P(s_t^{(i)}|s_1^{(i)}, s_2^{(i)}, ..., s_{(t-1)}^{(i)}, f_I^{(i)})$ is the probability of predicting the word $y_t^{(i)}$ at time step t of the i^{th} sample based on the previously predicted words and the input image features.

$$L_2(\varphi) = -\frac{1}{N_B}\sum_{i=1}^{N_B}\sum_{j}^{N_C}(y_{ij}\log(\hat{y_{ij}}); \varphi). \tag{2}$$

In (2), $y_{ij}, \hat{y_{ij}} \in \{0, 1\}$ are respectively the ground-truth label and the predicted label of the object detection model combined with GCN for the bounding box i belonging to the object class j.

$$L_3(\varphi) = \sum_{v\in V^{(2)}}\left(\sum_{u\in\mathcal{N}(v)} -\log(\sigma(h_v^{(N_L)}h_u^{(N_L)}) + \sum_{n\in\mathcal{N}'(v)}\log(1-\sigma(h_v^{(N_L)}h_n^{(N_L)})); \varphi\right). \tag{3}$$

In (3), σ is the sigmoid function, $h_v^{(N_L)}$ is the embedding vector of node at the final layer (N_L), $\mathcal{N}(v)$ is the set of neighboring nodes of v, and $\mathcal{N}'(v)$ is the set of randomly selected non-neighboring nodes of v.

With the presented symbols and definitions, the following sections of the paper detail each part of the proposed method, including the image encoding process of the Image Encoder and the caption generation process of the Language Decoder.

3.1. Image encoder

The Image encoder includes: Subsection 3.1.1 an object detection model combined with GCN, referred to as ODwGCN; Subsection 3.1.2 a relationship prediction model for objects in the image, referred to as VRP+RK; Subsection 3.1.3 a relationship graph, including the method for creating the relationship graph and the representation of the nodes in the relationship graph.

3.1.1. Object detection model combined with graph convolutional networks

Pre-trained object detection models such as SSD, Faster R-CNN, and YOLO have achieved particular effectiveness in general computer vision tasks and image captioning. However, these models struggle to accurately detect objects in images with many objects and complex details. Therefore, this study proposes an improved object detection model called ODwGCN, as shown in Figure 2. It consists of two stages. (a) Stage 1 involves learning co-occurrence relationships between objects in the image using a Graph Convolutional Network (GCN). (b) Stage 2 adjusts the object detection results of pre-trained object detection models based on the relationships (adjustment coefficient) learned in Stage 1.

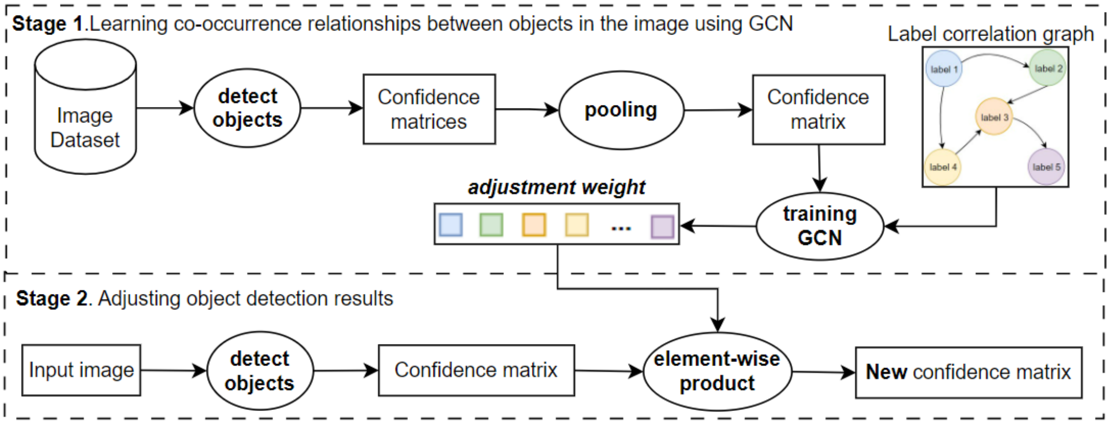

Figure 2: Improved object detection model based on Graph Convolutional Network (ODwGCN)

a. Stage 1: Learning co-occurrence relationships between objects in the image

In this stage, a label correlation graph is created, from which a GCN for object detection is built and trained to learn the co-occurrence relationships between objects in the image. The input is the object detection results from pre-trained object detection models.

a.1. Label correlation graph

The label correlation graph describes the co-occurrence relationships between object labels in the image dataset and is constructed using the ML-GCN method [13] to exploit the dependencies between objects to assist in adjusting object detection mistakes. In ML-GCN, each node in the graph represents a class label in the dataset, and the feature of each node is the word embedding of the object's class label. The graph's adjacency matrix is determined by counting the co-occurring class labels in the dataset's images. In this section, the features of the nodes are derived from the object detection results in the image instead of word embeddings to connect and exploit the image content.

Definition 1. The label correlation graph LC-Graph $G^{(1)} = (V^{(1)}, A^{(1)})$ is a directed graph consisting of:

- Vertex set $V^{(1)} = \{v_i^{(1)} \in C, \forall i = \overline{1, N_C}\}$, where each vertex $v_i^{(1)}$ represents a class label in the image dataset.
- Adjacency matrix $A^{(1)} = \{a_{ij}^{(1)} \in \{0,1\}, \forall i, j = \overline{1, N_C}\}$, where each element $a_{ij}^{(1)}$ in the adjacency matrix indicates the existence of a directed edge between two vertices $v_i^{(1)}$ and $v_j^{(1)}$.

a.2. Graph convolutional network for object detection

In this section, the method of learning to represent graph nodes based on the relationships between objects in the image is performed using GraphSAGE to overcome this limitation of GCN. GraphSAGE combines the feature vectors of neighbouring nodes into a vector representing the entire neighbourhood, then combines this information with the feature of the current node. With GraphSAGE network input as the LC-Graph ($G^{(1)}$) as defined in Definition 1, denoted as GraphSAGENet1, the implementation of Stage 1 is outlined in Algorithm 1.

After GraphSAGENet1 has been trained, the model has learned the set of weight matrices $W^{(1)}$. Then, the algorithm to generate embeddings for the graph nodes is described in Algorithm 2.

Algorithm 2 calculates the embedding vectors for the vertices in the graph across multiple layers. Each vertex starts with an initial feature vector $x_v^{(1)}$, and at each layer, the algorithm aggregates features from neighbouring nodes, then combines this information with the current feature of the node through a weight matrix and a non-linear activation function σ. This feature is then normalized before moving to the next layer. After passing through all the layers, the final feature of each vertex ($h_v^{(N_L)}$) is used as the vertex's representative embedding vector.

b. Stage 2: Adjusting object detection results

From the embedding vector (*or adjustment weight*) representing the relationships between object class labels learned in Stage 1, the adjustment of object detection results for each input image in this stage is performed through the following steps

Step 1 (*detect objects*): Use pre-trained object detection models to extract object regions in the input image, resulting in a confidence matrix Y.

Step 2: Execute an *element-wise product* between the matrix Y and the adjustment weight vector w, denoted as $\odot$, with the effect as follows

Algorithm 1 LearningWeightStage1($G^{(1)}, \hat{M}$)

```
Input: Image dataset $\mathcal{J}$, graph $G^{(1)} = (V^{(1)}, A^{(1)})$, pre-trained object detection model $\hat{M}$
Output: Weight vector $w$
begin
    $Y^{(i)} = \hat{M}(\mathcal{J}_i), \forall i = 1, \ldots, N_T$
    $X^{(1)} = []$
    for $i = 1$ to $N_T$ do
        for $j = 1$ to $N_C$ do
            $x^{(1)}_{ij} = \max_{k=1,\ldots,N_B}\{Y^{(i)}_{kj}\}$
        end
        # Add the row $x^{(1)}$ to the matrix
        $X^{(1)} = \begin{bmatrix} X^{(1)} \\ x^{(1)} \end{bmatrix}$
    end
    $N_{L_1}, W^{(1)} = $ TrainingGraphSAGENet1$(G^{(1)}, X^{(1)})$
    $w = $ GenerateEmbedding$(G^{(1)}, X^{(1)}, N_{L_1}, W^{(1)})$
    return $w$
end
```

$$\hat{Y} = Y \odot \alpha w = \begin{bmatrix} y_{11} & \ldots & y_{1N_C} \\ \vdots & \ddots & \vdots \\ y_{N_B 1} & \ldots & y_{N_B N_C} \end{bmatrix} \odot \alpha \begin{bmatrix} w_1 & \ldots & w_{N_C} \end{bmatrix} = \begin{bmatrix} y_{11}\alpha w_1 & \ldots & y_{1N_C}\alpha w_{N_C} \\ \vdots & \ddots & \vdots \\ y_{N_B 1}\alpha w_1 & \ldots & y_{N_B N_C}\alpha w_{N_C}. \end{bmatrix}.$$

Where $\hat{y}_{ij}$ indicates the probability of object region i belonging to class j and α is the adjustment coefficient. If $\alpha = 1$ and all elements of the vector w are 1, the confidence matrix does not need adjustment from the Graph Convolutional Network.

The matrix $\hat{Y}$ is the object detection result of the ODwGCN model.

3.1.2. Relationship prediction model for objects

The relationships between objects in an image play a crucial role in fully understanding the image content. However, recently published works often focus on specific relationships, such as positional relationships and actions (interactions between objects). Moreover, they typically only use object region features [14] or a combination of object region features and contextual relationship features (union of two object regions) [10] without exploiting the inherent relational knowledge in the dataset, leading to lower accuracy. Therefore, this study proposes a relationship prediction model that can recognize various types of relationships and exploit relational knowledge between entities in the dataset to improve accuracy.

The relationships between pairs of objects in an image are often represented as triplets $\langle subject, \textbf{predicate}, object\rangle$. In particular, the **predicate** is a word that links pairs of objects, such as $\langle woman, \textbf{riding}, motorcycle\rangle$, $\langle flower, \textbf{in}, vase\rangle$. In this paper, the relationship prediction model for objects is described in Figure 3, called VRP+RK, and consists of the main steps: (a) learning relational knowledge using GCN and (b) classifying relationships between objects using a Fully Connected (FC) network. Specifically, the input is two object

Algorithm 2 GenerateEmbedding($G^{(1)}, X^{(1)}, N_{L_1}, W^{(1)}$)

```
Input: Graph $G^{(1)} = (V^{(1)}, A^{(1)})$, feature vectors $X^{(1)} = \{x_v^{(1)}, \forall v \in V^{(1)}\}$ are number of
       layers $N_{L_1}$, weight matrices $W^{(l,1)}, \forall l = 1, \ldots, N_{L_1}$, non-linear activation function $\sigma$
Output: Embedding vectors $z_v, \forall v \in V^{(1)}$
begin
    $h_v^{(0)} \leftarrow x_v^{(1)}, \forall v \in V^{(1)}$
    for $l = 1$ to $N_{L_1}$ do
        for $v \in V^{(1)}$ do
            $h_{\mathcal{N}(v)}^{(l)} = f_{\text{agg}}\left(\{h_u^{(l-1)}, \forall u \in \mathcal{N}(v)\}\right)$
            $h_v^{(l)} = \sigma\left(W^{(l,1)} \cdot [h_v^{(l-1)}, h_{\mathcal{N}(v)}^{(l)}]\right)$
        end
        $h_v^{(l)} \leftarrow \frac{h_v^{(l)}}{\|h_v^{(l)}\|_2}, \forall v \in V^{(1)}$
    end
    $z_v \leftarrow h_v^{(N_{L_1})}, \forall v \in V^{(1)}$
    return $z_v$
end
```

regions and the contextual information of the relationship (region containing both objects). Besides that, the relational knowledge between the two objects is added to enhance accuracy. These features are concatenated (CONCAT) and then passed through fully connected layers to yield classification probabilities for $N_{\mathcal{R}+1}$ relationships ($N_{\mathcal{R}}$ relationships and the *none-relation* class).

a. Learning relational knowledge

The triplet dataset representing scene graphs in Visual Genome is used to exploit relational knowledge between objects to improve the accuracy of relationship prediction. The triplets in the training set are organized into a knowledge base $\mathcal{K}$ containing N_E entities, including *subjects*, *objects*, and *predicates*. In this study, the knowledge base $\mathcal{K}$ is represented as an entity graph from which the relationships between entities are learned using a GCN.

Definition 2. The entity graph E-Graph $G^{(2)} = (V^{(2)}, A^{(2)})$ is an undirected graph, including:

- Vertex set $V^{(2)} = \{v_i^{(2)} \in E, \forall i = \overline{1, N_E}\}$.
- Adjacency matrix $A^{(2)} = \{a_{ij}^{(2)} \in 0, 1, \forall i, j = \overline{1, N_E}\}$.

Here, E is the set of entities in $\mathcal{K}$. The binary adjacency matrix $A^{(2)}$ represents the relationships between entities, explicitly indicating the presence or absence of relationships between *subjects/objects* and *predicates*. For example, if $\mathcal{K}$ contains the triplet $\langle child, has, tie \rangle$, the elements $a_{child,has}^{(2)}$ and $a_{has,tie}^{(2)}$ in the adjacency matrix $A^{(2)}$ have a value of 1. The features of the entities (nodes of the graph) are the entities' word embedding vectors.

From the E-Graph defined in Definition 2 and the features of the vertex set, the GraphSAGE is used to learn the relational knowledge between entities, denoted as GraphSAGENet2. With $X^{(2)} = \{x_i^{(2)} \in E, \forall i = \overline{1, N_E}\}$ as the set of embedding vectors of the entities. In this section, the GraphSAGENet2 network is trained using an unsupervised learning method with the loss function L_3.

After GraphSAGENet2 has been trained and learned the weight matrices $W^{(l,2)}, \forall l = \overline{1, N_{L_2}}$. The embeddings of the graph nodes, also known as the relational knowledge of the entities, will be generated by applying Algorithm 2.

$$Z^{(\mathcal{K})} \leftarrow \text{GenerateEmbedding}\left(G^{(2)}, X^{(2)}, N_{L_2}, W^{(2)}\right). \qquad (4)$$

In (4), $Z^{(\mathcal{K})} \in \mathbb{R}^{N_E \times N_D}$ represents the relational knowledge of the entity set E (N_D is the dimensionality of the embedding vectors of the entities). Subsequently, the relational knowledge features of the object class labels from $Z^{(\mathcal{K})}$ can be extracted to supplement the input of the fully connected network to enhance relationship classification accuracy.

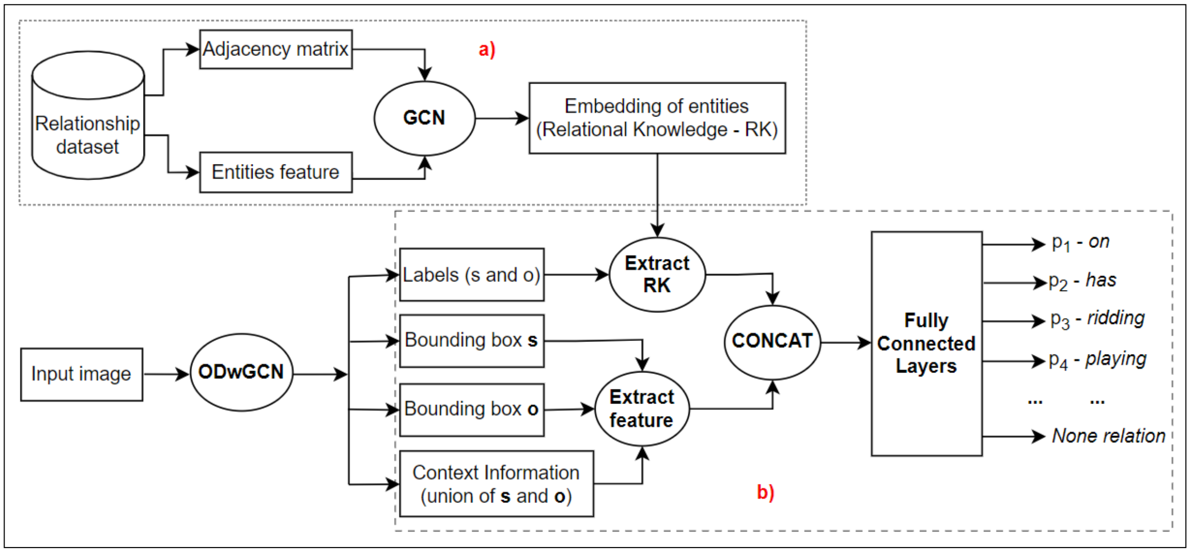

Figure 3: Model predicts relationships between objects in the image

b. Classifying relationships between objects

In this study, relationship prediction is performed as a classification task with the input of two object regions in the image (b_1 and b_2) and the relational knowledge of the object labels. The output is one of $N_{\mathcal{R}+1}$ relationships, including $N_{\mathcal{R}}$ relationships between two objects and `"none relation"`. The algorithm for predicting the relationship between two objects in the image is described as Algorithm 3.

3.1.3. Relationship graph

The relationship graph is a powerful tool for modeling and representing complex relationships between entities in the real world. It depicts entities as nodes and their relationships as edges. This section defines the relationship graph and then presents the method for creating it and how to represent it. In this study, the relationship graph of an image is defined as follows.

Algorithm 3 PredictRelationship($o_1, o_2, Z^{(\mathcal{K})}$)

```
Input: Bounding box $b_1$, bounding box $b_2$, $Z^{(\mathcal{K})}$
Output: $r$, the relationship between the two object regions
begin
    # Calculate the union of the two object regions
    $b_u \leftarrow UoI(b_1, b_2)$
    # Extract features of the object regions using a PreTrained CNN
    $f_1 \leftarrow \text{ResNet-101}(b_1),\ \ f_2 \leftarrow \text{ResNet-101}(b_2),\ \ f_u \leftarrow \text{ResNet-101}(b_u)$
    # Calculate the average value of the object feature vectors
    $f_{\text{avg}} \leftarrow \frac{1}{3}(f_1 + f_2 + f_u)$
    # Extract the relational knowledge of the object labels $b_1$ and $b_2$
    $k_{b1} \leftarrow z_1^{(\mathcal{K})} \in Z^{(\mathcal{K})},\ \ k_{b2} \leftarrow z_2^{(\mathcal{K})} \in Z^{(\mathcal{K})}$
    # Concatenate features
    $x \leftarrow f_{\text{avg}} \| k_{b1}, \| k_{b2}$
    # Predict the relationship
    $r \leftarrow \text{FCN}(x, W^{(3)})$
    return $r$
end
```

Definition 3. The relationship graph of an image, R-Graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, is a directed graph consisting of:

- Vertex set $\mathcal{V} = \{v_i \in \mathcal{B}, \forall i = 1, \ldots, N_{\mathcal{B}}\}$.
- Edge set $\mathcal{E} = \{e_{ij} = (v_i, v_j, r_{ij}) \in \mathcal{R}, \forall i, j = 1, \ldots, N_{\mathcal{B}}, i \neq j\}$.

a. Creating the relationship graph

After training the relationship classifier on the visual relationship dataset, this model and ODwGCN are used to create the relationship graph for the input image. First, N_B object regions in the input image are grouped into $N_B(N_B - 1)$ pairs of object regions. Then, the relationship prediction model is applied to yield the probability distribution over $(N_{\mathcal{R}} + 1)$ relationships. If the probability of the None-relation class is less than a given threshold θ (The experiment with θ= 0.5), an edge between the pair of vertices v_i and v_j will be established, and the label for this edge will be the relationship class with the highest probability. The algorithm for creating the relationship graph is described in Algorithm 4.

b. Representation of the relationship graph

Although the relationship graph accurately and efficiently represents all the information in the image, it is not suitable as input for most algorithms designed to use semantic information due to its heterogeneous nature [15]. Therefore, it is necessary to convert this graph representation into a linear form that preserves the graph information and can serve as input for learning architectures, specifically integrating it into a language model to generate image captions. Additionally, to exploit the semantics of object class labels and the relationships between objects, rather than only using object region features and graph structure as in previous works, this study proposes converting the relationship graph of the image into an extended relationship graph R-Graph*, $\mathcal{G}^* = (\mathcal{V}^*, \mathcal{E}^*)$:

Algorithm 4 CreateRelationGraph($I, Z^{(\mathcal{K})}$)

```
Input: Input image $I$, $Z^{(\mathcal{K})}$
Output: $R$-Graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$
begin
    # Object detection
    $\mathcal{B} \leftarrow$ ODwGCN($I$)
    $\mathcal{V} \leftarrow \{v_i \mid b_i \in \mathcal{B}\}$
    # Initialize the graph
    $\mathcal{G} \leftarrow (\mathcal{V}, \emptyset)$
    # Predict relationships and update edges in the graph
    foreach $(b_i, b_j) \in \mathcal{B} \times \mathcal{B}, i \neq j$ do
        $r_{ij} \leftarrow$ PredictRelationship($b_i, b_j, Z^{(\mathcal{K})}$)
        if $r_{ij} \neq$ "no relation" then
            $\mathcal{E} \leftarrow \mathcal{E} \cup \{(v_i, v_j, r_{ij})\}$
        end
    end
    return $\mathcal{G} = (\mathcal{V}, \mathcal{E})$
end
```

- Vertex set $\mathcal{V}^* = \{v_i^* \in \mathcal{L} \mid \forall i = (\overline{1, N_{\mathcal{L}}})\}$.
- Edge set $\mathcal{E}^* = \{e_{ij}^* \in \{0, 1\} \mid \forall i, j = (\overline{1, N_{\mathcal{L}}}),\ i \neq j\}$.

In which, the vertex set $\mathcal{L}$ consists of two types of vertices: those representing the labels of object regions and those representing the labels of relationships between objects (predicates). The edge set $\mathcal{E}^*$ is constructed according to the rule: If there is an edge $e_{ij} = (v_i, v_j, r_{ij}) \in \mathcal{E}$, then create two directed edges: one from vertex v_i to vertex r_{ij} and one from vertex r_{ij} to vertex v_j.

Learning to represent the vertices of the extended relationship graph R-Graph* uses GraphSAGE with an unsupervised learning method by optimizing the contrastive loss function L_3 based on the similarity between the vertices and their neighbouring vertices. The generation of the vertex embeddings is performed according to Algorithm 5.

From the results of the ODwGCN object detection model and the VRP+RK relationship prediction model between objects in Subsections 3.1.1 and 3.1.2, the object regions of the image are detected, and the relationship graph of the image is created. Next, a pre-trained convolutional neural network is used to extract features of the object regions, denoted as $\mathcal{F} = \{f_i \mid \forall i = (\overline{1, N_B})\}$. Finally, the relationship graph is represented as feature vectors using the method in Subsection 3.1.3, resulting in the feature vectors of the vertices z_v^*, $\forall v \in \mathcal{V}^*$. These features, combined with the attention weights in the dual attention mechanism, serve as input to the LSTM network to generate captions for the input image.

3.2. Language decoder

The language decoder includes Subsection 3.2.1 a dual attention mechanism to dynamically compute the weights of the relevant parts of the image as the decoder generates each word in the caption; Subsection 3.2.2 an LSTM network using the dual attention mechanism to generate captions for the image.

```
Algorithm 5 GenerateRGraphNodeEmbedding($\mathcal{G}^*, X, N_{L_3}, \mathcal{W}$)
Input: Graph $\mathcal{G}^* = (\mathcal{V}^*, \mathcal{E}^*)$, initial vector $X = \{x_v, \forall v \in \mathcal{V}^*\}$, number of layers $N_{L_3}$, weight
       matrices $W^{(l)}, \forall l = 1, \dots, N_{L_3}$
Output: Vertex embedding vector $z_v^*, \forall v \in \mathcal{V}^*$
begin
    $h_{v-}^{(0)} \leftarrow x_v, \forall v \in \mathcal{V}^*$
    $h_{v+}^{(0)} \leftarrow x_v, \forall v \in \mathcal{V}^*$
    for $l = 1$ to $N_{L_3}$ do
        for $v \in \mathcal{V}^*$ do
            $h_{\mathcal{N}-(v)}^{(l)} = f_{\text{agg}}\left(\{h_u^{(l-1)}, \forall u \in \mathcal{N}-(v)\}\right)$
            $h_{v-}^{(l)} = \sigma\left(W^{(l)} \cdot [h_v^{(l-1)}, h_{\mathcal{N}-(v)}^{(l)}]\right)$
            $h_{\mathcal{N}+(v)}^{(l)} = f_{\text{agg}}\left(\{h_u^{(l-1)}, \forall u \in \mathcal{N}+(v)\}\right)$
            $h_{v+}^{(l)} = \sigma\left(W_1^{(l)} \cdot [h_v^{(l-1)}, h_{\mathcal{N}+(v)}^{(l)}]\right)$
        end
    end
    $z_v^* \leftarrow \left[h_{v-}^{(N_{L_3})}, h_{v+}^{(N_{L_3})}\right], \forall v \in V^*$
end
```

3.2.1. Dual attention mechanism

The attention mechanism in the image captioning task allows the model to dynamically calculate attention weights for the relevant parts of the image at each step of the decoder's caption generation. The image features with these attention weights form a dynamic representation of the relevant parts of the image, called the context vector, denoted as c_t. This vector serves as input to the decoder during the caption generation process. This study introduces a dual attention mechanism consisting of two independently operating mechanisms: visual attention and graph attention. These attention mechanisms calculate the attention weights between the hidden state of the decoder with features of the object regions and the features of the vertices in the extended relationship graph to fully exploit the information of these regions and the graph. Specifically, the visual attention context vector $c_t^{(v)}$ and the graph attention context vector $c_t^{(g)}$ are generated according to Algorithm 6.

In Algorithm 6, the attention scores are first calculated through a linear transformation network f_{att} to determine the importance of the feature parts at the current time step for the decoder to decide the next word to generate. Next, the attention weights are obtained by using the softmax function to normalize the attention scores. Finally, the attention vector, also known as the context vector, is calculated as the weighted sum of the feature components of the object regions in the image and the features of the graph vertices.

3.2.2. LSTM network generates captions

The Long-Short Term Memory (LSTM) network addresses the shortcomings of the Recurrent Neural Network (RNN) in handling long-term dependencies due to the vanishing gradient problem when processing sequential data. In this paper, the LSTM network is used

Algorithm 6 CreateContextVector($\mathcal{F}, \mathcal{Z}^*, h_{t-1}$)

```
Input: Object region feature set $\mathcal{F}$, set of embedding vectors of the graph vertices $\mathcal{Z}^*$, the
       previous hidden state of the decoder $h_{t-1}$
Output: Context vector at time step $t$: $c_t^{(v)}, c_t^{(g)}$
begin
    # Calculate alignment score
    $e_{ti}^{(v)} = f_{\text{att}}(f_i, h_{t-1}), \forall i = 1, \dots, N_B$, $e_{ti}^{(g)} = f_{\text{att}}(z_i, s_{t-1}), \forall i = 1, \dots, N_{\mathcal{L}}$
    # Calculate attention weight
    $\rho_{ti}^{(v)} = \frac{\exp(e_{ti}^{(v)})}{\sum_{k=1}^{N_B} \exp(e_{tk}^{(v)})}, \sum_i \rho_{ti}^{(v)} = 1, 0 < \rho_{ti}^{(v)} < 1, \forall i = 1, \dots, N_B$
    $\rho_{ti}^{(g)} = \frac{\exp(e_{ti}^{(g)})}{\sum_{k=1}^{N_{\mathcal{L}}} \exp(e_{tk}^{(g)})}, \sum_i \rho_{ti}^{(g)} = 1, 0 < \rho_{ti}^{(g)} < 1, \forall i = 1, \dots, N_{\mathcal{L}}$
    # Calculate context vectors
    $c_t^{(v)} = \sum_{i=1}^{N_B} f_i \rho_{ti}^{(v)}$, $c_t^{(g)} = \sum_{i=1}^{N_L} z_i \rho_{ti}^{(g)}$
    return $c_t^{(v)}, c_t^{(g)}$
end
```

as a language model in conjunction with a dual attention mechanism to generate captions for images. The LSTM takes the word embedding x_t, the previously hidden state h_{t-1}, and the context vectors $c_t^{(v)}$ and $c_t^{(g)}$ generated from the dual attention mechanism as input. The input gate i_t, forget gate f_t, output gate o_t, and memory cell C_t of the LSTM at time step t are updated according to the following calculations

$$\begin{cases} i_t = \sigma\left(W_{xi} x_t + W_{hi} h_{t-1} + W_{ic}\left[c_t^{(v)}; c_t^{(g)}\right] + b_i\right) \\ f_t = \sigma\left(W_{xf} x_t + W_{hf} h_{t-1} + W_{fc}\left[c_t^{(v)}; c_t^{(g)}\right] + b_f\right) \\ o_t = \sigma\left(W_{xo} x_t + W_{ho} h_{t-1} + W_{oc}\left[c_t^{(v)}; c_t^{(g)}\right] + b_o\right) \\ \tilde{C}_t = \tanh\left(W_{xc} x_t + W_{hc} h_{t-1} + W_{cc}\left[c_t^{(v)}; c_t^{(g)}\right] + b_c\right) \\ C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t \\ h_t = o_t \odot \tanh(\tilde{C}_t). \end{cases} \tag{5}$$

Where, δ is the sigmoid function, W and b are the learned weights of the model, and $\odot$ denotes the element-wise multiplication. Finally, the hidden state h_t is used to predict the output word by generating the probability distribution p_t over the current word y_t using the *softmax* function as follows

$$y_t \sim p_t = \text{softmax}(W_p h_t + b_p). \tag{6}$$

In (6), W_b and b_p are the weights learned by the model. This LSTM network is trained using the BackPropagation Through Time algorithm with the loss function L_1.

4. EXPERIMENTS

Based on the theoretical foundation and the introduced model, this section implements experiments on datasets and evaluates the effectiveness of the proposed method using commonly used metrics in object detection, relationship prediction between objects, and image captioning.

4.1. Data and experimental setup

The Visual Genome dataset contains 108,077 images, each with an average of 35 objects and 21 pairwise relationships (triplets) between objects, used for the relationship prediction model. However, many annotations in this dataset are of low quality, and the object regions (bounding boxes) are overlapping, with ambiguous object names. These make it difficult for the model to learn the information effectively. Therefore, we performed preprocessing by filtering out low-quality annotations and overlapping object regions according to the method of Xu et al. [16]. After preprocessing, the 150 most common object classes (subjects/objects) and 50 most common relationships (predicates) are used for the relationship prediction model in this paper. The new dataset consists of 95,998 images, divided into two parts: 70% for training and 30% for testing. This preprocessed dataset is also widely used in other works [16, 17].

The improved object detection (ODwGCN) and image captioning model is experimented on the MS COCO image dataset. MS COCO is a large and benchmark dataset for object recognition, image segmentation, and image captioning tasks. This dataset includes 82,783 training images and 40,504 validation images. Each image has at least five human-generated captions, with some images having more than five. However, this experiment only uses the first five captions to ensure consistency with other images. To be consistent with other works in performance evaluation, the dataset is divided into three parts according to [15]: 82,783 images for training, 5,000 for validation, and 5,000 for testing. After preprocessing and removing words that appear less than five times, the caption vocabulary results in 10,010 words, with a maximum caption length of 16.

4.1.1. Implementation details

The experimental model of the proposed method is implemented using the Python programming language (3.9) and the deep learning framework PyTorch (2.0), torch-geometric, all executed on Google Colab Pro with the following specific configurations and parameters:

General setup: The Faster textR-CNN_wGCN model and ResNet101 are used to detect objects and extract features (2048 dimensions) of object regions in images. Object class labels, words in ground truth captions, entities, and relationships are embedded using the GloVe technique with an embedding size of 300. The number of layers in the GCN networks is set to 2.

Object detection model combined with GCN: In this section, only the GCN network is trained to learn the embeddings of the graph vertices. The GCN network consists of 2 graph convolutional layers with feature map sizes of $(1, c) \to (4, c) \to (1, c)$, where c is the number of objects in the dataset. For the MS COCO dataset, $c = 81$ (including 80 objects and background).

Relationship prediction model: The fully connected network is trained with the cross-entropy loss function, Adam optimizer, learning rate of 0.0003, maximum iterations of 20000, batch size set to 64, and dropout of 0.5.

Image captioning model: The LSTM's hidden state size is 512, and the generated caption's maximum length is 16. The LSTM network is trained with the cross-entropy loss function, applying the Adam optimizer, learning rate of 0.0001, and batch size of 128.

4.2. Evaluation metrics

In this experiment, the metrics used to evaluate the performance of the object detection model are mean Average Precision (mAP), mAP@0.5, and mAP@0.75; Recall@50 and Recall@100 are used to evaluate the relationship prediction model. For the image captioning task, metrics such as BLEU, METEOR, ROUGE, and CIDEr are used to assess the quality of the generated captions compared to the ground truth captions. Each metric calculates and evaluates the resulting captions from different perspectives. However, all these metrics share a common characteristic: higher values indicate better performance and the metrics are expressed as percentages (%).

4.3. Experimental results

The experimental results, discussion, and comparison with baseline works or recently published works are presented in this section to clarify the strengths and weaknesses of the proposed method. Specifically, it includes the experimental results of object detection, relationship prediction between objects, and image captioning.

4.3.1. Experimental results on object detection

Table 1: Object detection results of pre-trained object detection models compared to ODwGCN on the COCO dataset

Model	Backbone	mAP	mAP@0.5	mAP@0.75
SSD	VGG16	28.8	48.5	30.3
SSD_wGCN	VGG16	30.9	49.4	32.8
SSD	ResNet-101-FPN	31.2	50.4	33.3
SSD_wGCN	ResNet-101-FPN	33.4	53.6	35.4
Faster R-CNN	ResNet-101-FPN	36.2	59.1	39.0
Faster R-CNN_wGCN	ResNet-101-FPN	37.5	60.4	41.8
Faster R-CNN	Inception_ResNet_v2	34.7	55.1	36.7
Faster R-CNN_wGCN	Inception_ResNet_v2	36.2	56.9	38.6
YOLOX	DarkNet-53	47.4	67.3	52.1
YOLOX_wGCN	DarkNet-53	**48.2**	**68.2**	**53.4**

The object detection performance of the pre-trained models (SSD, Faster R-CNN, and YOLOX) and the ODwGCN model according to the metrics mAP, mAP@0.5, and mAP@0.75 is presented in Table 1. Models with the suffix wGCN indicate that object detection results are adjusted using the GCN network. The experimental results show that the ODwGCN model improved the mAP for all baseline object detection models in the experimental list, increasing by 0.9 to 3.2 units. Specifically, the SSD model with the ResNet101 network had the highest adjustment increase with 2.2 (mAP) and 3.2 (mAP@0.5), while the YOLOX model with DarkNet-53 had the lowest adjustment increase with 1.2 (mAP), 0.9 (mAP@0.5), and 1.3 (mAP@0.75). However, the object detection performance of YOLOX is higher than that of SSD and Faster R-CNN, so the YOLOX_wGCN model still achieved the highest performance in this experiment. These results demonstrate the effectiveness of adjusting object detection using the graph convolutional network based on the co-occurrence relationships of objects in the image. Notably, the ODwGCN model proposed in this paper can be combined with any object detection model to improve accuracy.

4.3.2. Experimental results on object relationship prediction

The relationship prediction results of the proposed method are Recall@50 = 54.7 and Recall@100=60.3. Compared with the results of other methods described in Table 2, our proposed method shows higher accuracy. Specifically, compared to the baseline method (VRD), it is 26.8% higher for the Recall@50 metric and 25.3% higher for the Recall@100 metric. The other two methods show an improvement ranging from 2.2 to 2.9 for Recall@50 and from 1.9 to 7.3 for Recall@100. These results indicate that contextual information about relationships, particularly the inherent relational knowledge between objects, significantly contributes to accurately detecting relationships between objects in images.

Table 2: Comparison of accuracy of relationship prediction methods on the experimental dataset

Method	Recall@50 (%)	Recall@100 (%)
VRD [18]	27.9	35.0
Message Passing [16]	44.8	53.0
MSTG [17]	52.5	58.4
VRP+RK	**54.7**	**60.3**

Table 3: Comparison of image captioning performance between methods on the experimental dataset

Method	BLUE-1	BLUE-2	BLUE-3	BLUE-4	METEOR	ROUGE	CIDEr
Show, attend and tell (Hard-ATT)-2015 [4]	71.8	50.4	35.7	25.0	23.0	-	-
Show, attend and tell (Soft-ATT)-2015 [4]	70.7	49.2	34.4	24.3	23.9	-	-
Dense_Soft-ATT-2019 [6]	68.3	47.4	32.5	22.9	22.6	53.0	74.3
En-De-Cap-2021 [7]	70.6	41.1	36.7	24.3	-	-	-
Caption TLSTMs-2022 [12]	-	-	-	22.9	**25.2**	50.9	**203.5**
Bi-LS-AttM-2023 [9]	68.8	51.0	35.9	25.2	21.5	-	41.2
OD-VR-Cap	**72.6**	**52.2**	**38.7**	**28.3**	24.8	**53.4**	85.1

4.3.3. Experimental results on image captioning

The experimental results of the proposed method are listed in Table 3 (the last row), with the BLEU1-4, METEOR, ROUGE, and CIDEr scores being 72.6, 52.2, 38.7, 28.3, 24.8, 53.4, and 85.1, respectively. To demonstrate the effectiveness of the proposed method (OD-VR-Cap), we compare these performance values with the performance of baseline methods [4] (the first work to apply the attention mechanism to the encoder-decoder framework for image captioning) and recently published methods on the MS COCO dataset (Table 4). In Table 3, bold values indicate the best results for the corresponding metric, and the symbol (-) indicates that the method did not evaluate this metric. Table 4 shows that the Caption TLSTMs2022 [12] method, which combines the LSTM network with a Transformer, outperforms all other methods in terms of METEOR and CIDEr scores, especially with a CIDEr score of 203.5 compared to BiLSAttM2023, Dense_Soft-ATT2019, and the method in this study, which have scores of 41.2, 74.3, and 85.1, respectively.

Caption TLSTMs-2022 has outstanding performance on the CIDEr metric because this paper uses abstract scene graphs to generate diverse captions, combined with Transformer blocks between two LSTM layers to produce fluent and coherent captions. However, the proposed method in this paper outperforms Caption TLSTMs-2022 on the BLEU1-4 and ROUGE metrics and surpasses all other methods in Table 3 on all mentioned metrics. The effectiveness of our proposed method stems from the relationship graph, which enables the model to capture complex dependencies between objects that are not easily addressed by baseline methods and are often overlooked by other approaches. This capability facilitates a more precise representation of object interactions, improving performance. Furthermore, the dual attention mechanism enhances the model's ability to concurrently focus on both the objects' visual features and the graph nodes' semantic attributes. This comprehensive focus enables the model to interpret better and utilize the available data, further contributing to performance gains. These results demonstrate that the proposed method in this paper is feasible and effective.

5. CONCLUSION

This paper approached the image captioning model based on relationship graphs and a dual attention mechanism to enhance the effectiveness of image captioning. Experiments on the MS COCO dataset have shown that the image captioning model in this study has higher accuracy than the baseline method and recently published works on the most common metrics. In other words, relationship graphs can fully represent the semantics of image content. Additionally, the LSTM network, which has a dual attention mechanism, helps improve accuracy during the caption generation process. Furthermore, the improved object detection model (ODwGCN) and relationship prediction introduced in this paper also show higher accuracy than baseline models and recently published works. Therefore, this image captioning method is feasible and practical, providing a foundation for developing image captioning systems applicable in various real-world fields.

ACKNOWLEDGMENT

We acknowledge the support of the Institute of Mechanics and Applied Informatics - VAST, the Faculty of Information Technology and Telecommunications - Graduate University of Science and Technology - VAST, and Ho Chi Minh City University of Education (HCMUE) for this study.

REFERENCES

[1] A. Verma, A. K. Yadav, M. Kumar, and D. Yadav, "Automatic image caption generation using deep learning," *Multimedia Tools and Applications*, vol. 83, no. 2, pp. 5309–5325, 2024. [Online]. Available: https://doi.org/10.1007/s11042-023-15555-y

[2] A. Jamil, K. Mahmood, M. G. Villar, T. Prola, I. D. L. T. Diez, M. A. Samad, and I. Ashraf, "Deep learning approaches for image captioning: Opportunities, challenges and future potential," *IEEE Access*, vol. 4, pp. 1–1, 2024. [Online]. Available: http://dx.doi.org/10.1109/ACCESS.2024.3365528

[3] Z. Zohourianshahzadi and J. K. Kalita, "Neural attention for image captioning: review of outstanding methods," *Artificial Intelligence Review*, vol. 55, no. 5, pp. 3833–3862, 2022. [Online]. Available: https://dx.doi.org/10.1007/s10462-021-10092-2

[4] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio, "Show, attend and tell: Neural image caption generation with visual attention," *Proceedings of the 32nd International Conference on Machine Learning*, vol. 37, pp. 2048–2057, 2015. [Online]. Available: https://proceedings.mlr.press/v37/xuc15.html

[5] J. Jia, X. Ding, S. Pang, X. Gao, X. Xin, R. Hu, and J. Nie, "Image captioning based on scene graphs: A survey," *Expert Systems with Applications*, vol. 231, p. 120698, 2023. [Online]. Available: https://dx.doi.org/10.1016/j.eswa.2023.120698

[6] M. Z. Hossain, F. Sohel, M. F. Shiratuddin, H. Laga, and M. Bennamoun, "Attention-based image captioning using densenet features," *Neural Information Processing: 26th International Conference, ICONIP 2019, Sydney, NSW, Australia, December 12–15, 2019, Proceedings, Part V 26*, pp. 109–117, 2019. [Online]. Available: https://dx.doi.org/10.1007/978-3-030-36802-9_13

[7] N. Patwari and D. Naik, "En-de-cap: An encoder decoder model for image captioning," *2021 5th International Conference on Computing Methodologies and Communication (ICCMC)*, pp. 1192–1196, May 2021. [Online]. Available: https://dx.doi.org/10.1109/ICCMC51019.2021.9418414

[8] N. V. Thinh, T. V. Lang, and V. T. Thanh, "Automatic image captioning based on object detection and attention mechanism," *The 16th National Conference on Fundamental and Applied IT Research, FAIR'2023*, pp. 395–404, 2023. [Online]. Available: https://dx.doi.org/10.15625/vap.2023.0063

[9] T. Xie, W. Ding, J. Zhang, X. Wan, and J. Wang, "Bi-ls-attm: A bidirectional lstm and attention mechanism model for improving image captioning," *Applied Sciences*, vol. 13, no. 13, p. 7916, 2023. [Online]. Available: https://dx.doi.org/10.3390/app13137916

[10] T. Yao, Y. Pan, Y. Li, and T. Mei, "Exploring visual relationship for image captioning," *Proceedings of the European conference on computer vision (ECCV)*, pp. 684–699, 2018. [Online]. Available: https://openaccess.thecvf.com/content_ECCV_2018/html/Ting_Yao_Exploring_Visual_Relationship_ECCV_2018_paper.html

[11] S. Chen, Q. Jin, P. Wang, and Q. Wu, "Say as you wish: Fine-grained control of image caption generation with abstract scene graphs," *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9962–9971, 2020. [Online]. Available: https://openaccess.thecvf.com/content_CVPR_2020/html/Chen_Say_As_You_Wish_Fine-Grained_Control_of_Image_Caption_Generation_CVPR_2020_paper.html

[12] J. Yan, Y. Xie, X. Luan, Y. Guo, Q. Gong, and S. Feng, "Caption tlstms: combining transformer with lstms for image captioning," *International Journal of Multimedia Information Retrieval*, vol. 11, no. 2, pp. 111–121, 2022. [Online]. Available: https://dx.doi.org/10.1007/s13735-022-00228-7

[13] Z.-M. Chen, X.-S. Wei, P. Wang, and Y. Guo, "Multi-label image recognition with graph convolutional networks," *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 5177–5186, 2019. [Online]. Available: https://openaccess.thecvf.com/content_CVPR_2019/html/Chen_Multi-Label_Image_Recognition_With_Graph_Convolutional_Networks_CVPR_2019_paper.html

[14] R. Zellers, M. Yatskar, S. Thomson, and Y. Choi, "Neural motifs: Scene graph parsing with global context," *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5831–5840, 2018. [Online]. Available: https://openaccess.thecvf.com/content_cvpr_2018/html/Zellers_Neural_Motifs_Scene_CVPR_2018_paper.html

[15] A. Karpathy and L. Fei-Fei, "Deep visual-semantic alignments for generating image descriptions," *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3128–3137, 2015. [Online]. Available: https://www.cv-foundation.org/openaccess/content_cvpr_2015/html/Karpathy_Deep_Visual-Semantic_Alignments_2015_CVPR_paper.html

[16] D. Xu, Y. Zhu, C. B. Choy, and L. Fei-Fei, "Scene graph generation by iterative message passing," *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5410–5419, 2017. [Online]. Available: https://doi.org/10.48550/arXiv.1701.02426

[17] P. Tian, H. Mo, and L. Jiang, "Scene graph generation by multi-level semantic tasks," *Applied Intelligence*, vol. 51, no. 11, pp. 7781–7793, 2021. [Online]. Available: https://dx.doi.org/10.1007/s10489-020-02115-2

[18] C. Lu, R. Krishna, M. Bernstein, and L. Fei-Fei, "Visual relationship detection with language priors," *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*, pp. 852–869, 2016. [Online]. Available: 10.48550/arXiv.1701.02426

Received on June 11, 2024
Accepted on September 18, 2024