Computers, Materials & Continua

Tech Science Press

DOI: 10.32604/cmc.2023.036564
*Article*

# Fine-Grained Features for Image Captioning

**Mengyue Shao[1], Jie Feng[1,*], Jie Wu[1], Haixiang Zhang[1] and Yayu Zheng[2]**

[1]Zhejiang Sci-Tech University, Hangzhou, 310020, China
[2]Zhejiang University of Technology, Hangzhou, 310020, China

*Corresponding Author: Jie Feng. Email: arlose@zstu.edu.cn
Received: 04 October 2022; Accepted: 13 February 2023

**Abstract:** Image captioning involves two different major modalities (image and sentence) that convert a given image into a language that adheres to visual semantics. Almost all methods first extract image features to reduce the difficulty of visual semantic embedding and then use the caption model to generate fluent sentences. The Convolutional Neural Network (CNN) is often used to extract image features in image captioning, and the use of object detection networks to extract region features has achieved great success. However, the region features retrieved by this method are object-level and do not pay attention to fine-grained details because of the detection model's limitation. We offer an approach to address this issue that more properly generates captions by fusing fine-grained features and region features. First, we extract fine-grained features using a panoramic segmentation algorithm. Second, we suggest two fusion methods and contrast their fusion outcomes. An X-linear Attention Network (X-LAN) serves as the foundation for both fusion methods. According to experimental findings on the COCO dataset, the two-branch fusion approach is superior. It is important to note that on the COCO Karpathy test split, CIDEr is increased up to 134.3% in comparison to the baseline, highlighting the potency and viability of our method.

**Keywords:** Image captioning; region features; fine-grained features; fusion

## 1 Introduction

Image captioning is the task of producing a sentence that conforms to visual semantics for an input image. This makes us not only need to identify the objects in the image but also need to explore the semantic information between the objects. The research on image captioning is mainly divided into two parts: one is to extract more effective image features and the other is to improve the caption model expression ability. According to the great success of deep learning methods in Computer Vision (CV) and Natural Language Processing (NLP), a Convolutional Neural Network-Recurrent Neural Network (CNN-RNN) has been proposed [1]. CNN network was used to extract image features, and then the RNN network was used as the caption model to extract semantic information between features and transform it into sentences consistent with visual semantics.

At present, the commonly used image features are the region features [2], which are extracted by Faster R-CNN network rather than the basic CNN network. This feature has rich information, which is conducive to generating richer expressions than the simple CNN network used before. However, this feature still has two defects: 1. Due to the nature of Faster R-CNN, features in the rectangular area are extracted, and a rectangular area represents the information of an object. For actual objects, their contours are irregular, so the target of selecting a rectangular area inevitably incorporates some background information into the object 2. The detection model is object-level and lacks fine-grained details.

Therefore, we want to segment the contour of the target to avoid other information mixing into the target and extract fine-grained features to make up for the deficiency of region features so that the generated sentence is more consistent with visual semantics. We choose the panoramic segmentation algorithm to segment the objects in the image and obtain fine-grained and cleaner features. Different from traditional image segmentation [3], the panoramic segmentation algorithm can obtain features of different dimensions and obtain more information. In addition, it is a collection of semantic segmentation and instance segmentation, and each pixel can be assigned a semantic label and an instance label, thus describing an object more comprehensively.

The RNN-based caption model was very popular at first. There are many innovative types of research based on the RNN model, but the RNN model has the problem of gradient disappearing or gradient eruption in long sentence training. With the outstanding performance of the transformer [4] in NLP, the transformer-based caption model has been used in image captioning. We chose X-LAN [5] model as the baseline. This model updates the self-attention module of the traditional transformer, proposes X-linear attention, makes full use of the bilinear pool to capture the middle second-order features, and measures the distribution of spatial and channel bilinear attention. To fuse region features with our extracted fine-grained features, we experiment with two fusion methods. As shown in Fig. 1, a) is the direct fusion method and b) is the two-branch fusion method. b) is inspired by [6]. We have compared the performance of these two methods in subsequent experiments. In addition, we have performed many experiments on the COCO [7] benchmark dataset and compared it with some existing methods. Our method has achieved competitive BLEU [8], METEOR [9], ROUGE-L [10], CIDEr [11] and SPICE [12] scores on the COCO Karpathy test split.

We summarize the contributions of this paper as follows: (1) We propose extracting features using Panoptic Feature Pyramid Network (FPN), a panoramic segmentation algorithm. Panoptic FPN [13] is a pixel-level algorithm that can effectively extract fine-grained features. Using the characteristics of the segmentation algorithm, we obtain an accurate target region and extract cleaner target features. (2) We study the problem of the fusion of different kinds of features and propose two fusion methods. (3) We conducted a large number of experiments on the COCO dataset to compare the performance of these two methods.

The rest of this paper is organized as follows. Section 2 introduces the related work of image captioning. Section 3 presents the details of our method, including feature extraction and two fusion methods. Section 4 is the experimental part, which performs various ablation experiments and compares the performance with the baseline model. In addition to the comparison of metrics, we also compared the generated sentences with the baseline to show the excellence of our method more intuitively. Section 5 is the conclusion.

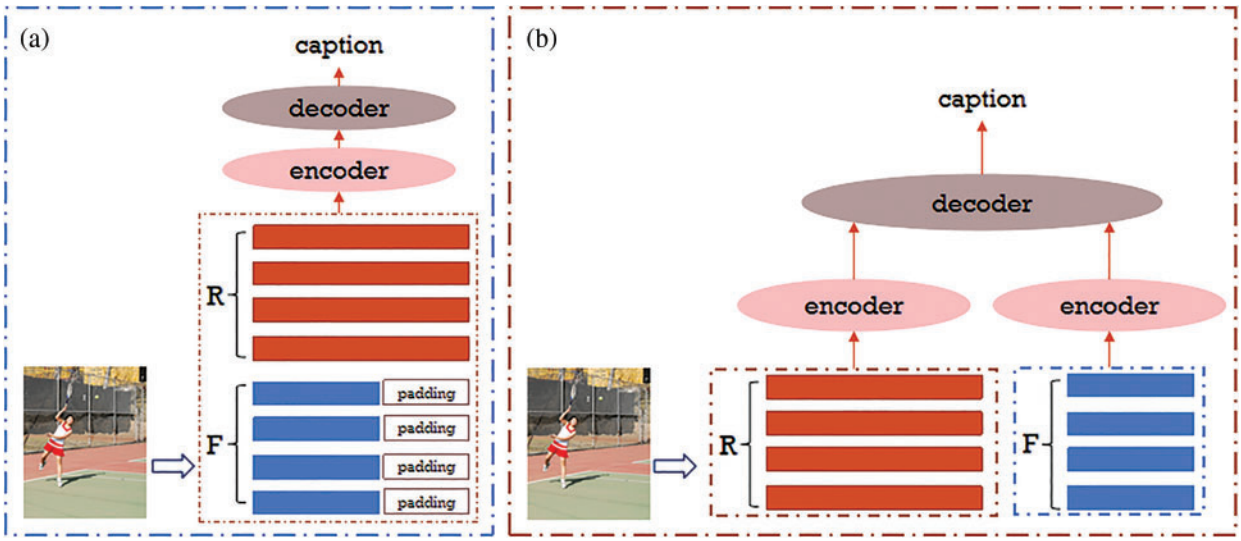


**Figure 1:** Schematic diagram of two fusion methods. R means region features and F means fine-grained features. (a) is the direct fusion method. The fine-grained features are filled to 2048 dimensions and concatenated with region features. (b) is the two-branch fusion method

## 2 Related Works

### *2.1 Image Captioning*

Encoder-decoder frameworks have been used extensively in image captioning. Inspired by the field of machine translation, Vinyals et al. [1] adopt the encoder-decoder framework for the first time and combined it with the actual demand for multimode image captioning. The encoder used a pre-trained CNN to extract image features. The decoder uses an RNN structure to transform image information into text information. Later, many researchers improved on encoder-decoder frameworks CNN-RNN, such as [14–16]. CNN networks used for image feature extraction are constantly changing. At first, basic networks like Visual Geometry Group (VGG) and Residual Neural Network (ResNet) are used, then target detection networks like Faster R-CNN are used and now multi-modal networks like Contrastive Language-image Pretraining (CLIP) [17] are used. At present, the region features extracted by Faster R-CNN are still playing an important role in image captioning. With the amazing performance of the transformer in NLP, the transformer-based caption model gradually replaces the RNN network to occupy the dominant position.

### *2.2 Features Extraction*

Anderson et al. first adopted the object detection network Faster R-CNN to extract image features. They used this network for pretraining on the Visual Genome (VG) [18] dataset and added training output for predicting attribute classes. The obtained pre-trained model is then used to extract the image features of the COCO dataset. The feature obtained by such processing not only contains the feature information of the target region but also contains the attribute features of the target so that the obtained region features can be better described. Since then, many researchers have conducted studies based on this feature.

Jiang et al. [19] used experiments to prove that the accuracy of region features did not come from regional frames but from 1. Large-scale annotation: large-scale target and attribute annotation

collected from the VG dataset for pretraining 2. High resolution: High spatial resolution of the input image used to compute features. In addition, the reasoning speed of grid features is much faster than that of region features, so he extracts the grid features of Faster R-CNN to replace region features. The RSTNet [20] method proposed by Zhang et al. and the Dual-level Collaborative Transformer (DLCT) [21] method proposed by Luo et al. both use grid features.

In recent years, with the proposal of CLIP, researchers have also begun to use CLIP to extract image features. CLIP is trained on large-scale text data, so the model portability is good, and the development of image captioning is not limited to the COCO dataset. For example, ClipCap [22], proposed by Mokady et al., uses the CLIP model to encode the input image, obtains an image feature clip_embed, maps it to text space, and finally uses a text decoder to generate sentences. Deng et al. [23] tried to use the FPN model to extract multilevel features so that the model could detect objects of different scales in images more effectively without increasing parameters. Nejatishahidin et al. [24] tried to use an image segmentation network to extract the object mask and mid-level representation feature maps so that they can achieve competitive performance under a limited training data regime.

This paper attempts to use a segmentation method to extract fine-grained image features. As shown in Fig. 2, the object detection algorithm locates the general region of the object and boxes this region to obtain a vector feature, which is the object level. Our new method is to give each pixel an instance label, so the segmented area is consistent with the shape of the object, which is a pixel-level feature.

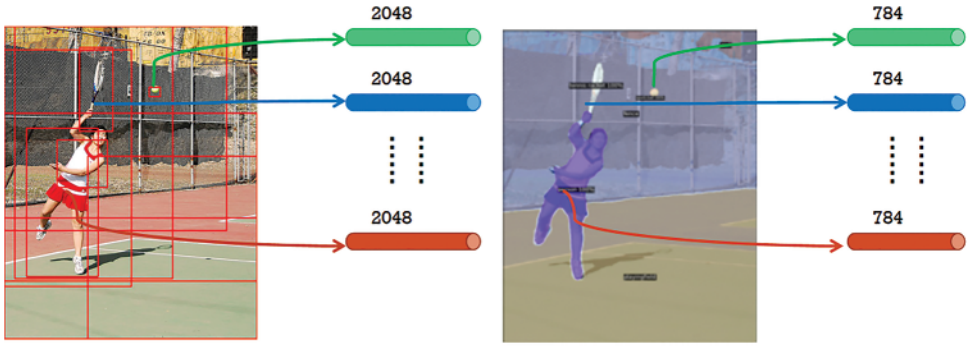


**Figure 2:** The contrast of the two features. (a) Region features extraction method. Region boxes are obtained from the faster R-CNN network, and each region box forms a vector of 2048 dimensions. (b) Fine-grained features extraction method. Accurate segmentation regions are obtained by the Panoptic FPN network, and each segmentation region forms a vector of 784 dimensions

### *2.3 Feature Fusion*

Region features are pre-trained on VG datasets so that they contain information about many attributes. It is difficult to completely replace region features with other features so people study how to integrate region features with other features.

Luo et al. [21] propose to integrate grid features [19] with the region features reasonably to supplement the fine-grained details that are missing from the region features. Then, the grid features have more than 300 G, which is more than 10 times the size of the region features. The information in it is too complex, and a complex model needs to be designed to extract the features in it. Although its metrics are greatly improved, such feature fusion cost is high, and the applicability of features is

low. Wang et al. [25] propose multi-dimensional features, which include text features, region features and patch features. The patch features are extracted by dividing the original image into blocks and the text feature is word-embedding vectors. They compile these three features into the attention model and then combined them. This method has the same problem as above. The added features are too complex to consume a lot of computing resources.

Therefore, we hope to extract small-scale fine-grained features and fusion with region features in a simple way that can also achieve competitive results. In this paper, we try two methods to fuse fine-grained features and region features and compare the two methods experimentally.

### *2.4 Transformer-Based Method*

With the transformer network becoming more prominent in the field of NLP and the emergence of many related models such as [26,27], researchers are inspired by this and integrate the transformer into the RNN encoder.

Huang et al. [28] proposed the attention on attention (AoA) module. AoA expands the attention of the caption model through two linear lines, which is conducive to modeling and fusing information of different modes (such as text and image). Guo et al. [29] proposed geometry-aware self-attention (SA), which improves the SA module by considering the pairwise geometric relationship and content information of objects, thus helping to reason about visual information. Cornia et al. [30] proposed the memory-augmented attention module to establish prior knowledge of the relationship between image regions. Pan et al. [5] proposed the X-linear attention module. The integration of X-linear attention blocks into the caption model can capture information about higher-order internal modes and multimodal interactions, aiming to enhance visual information and perform complex multimodal reasoning for image captioning. There are many related studies [31,32].

In summary, in the development of image captioning, the frequently used models are modified based on the transformer, so we use the X-LAN network based on transformers as the baseline network.

## 3 Method Details

The method in this paper is mainly divided into two parts: feature extraction and feature fusion. Image features are obtained through a feature extraction process. Image features are divided into fine-grained features obtained by Panoptic FPN and region features obtained by Faster R-CNN. We will detail the process of obtaining fine-grained features using Panoptic FPN in Section 3.1. Then, we will explore how to fuse features to better produce sentences that conform to visual semantics. Section 3.2 mainly introduces how to fuse features.

### *3.1 Feature Extraction*

#### *3.1.1 Region Features*

We use the features extracted by Anderson et al. as region features [2]. They trained Faster R-CNN using the VG dataset and added an output layer to define softmax distributions on each attribute class and one “attribute-free” class so that the extracted features could contain not only their category information but also their attribute information. Finally, the trained network is used to extract the region features we need on the COCO dataset. Because of its rich information, it enhances the expression ability of subsequent models.

### *3.1.2 Fine-Grained Feature*

Since the feature obtained by the network is in a rectangular area and many targets are irregular in shape, most extracted features are mixed with background features in the target features, and fine-grained information is missing. Therefore, we propose a panoptic segmentation algorithm, Panoptic FPN, to extract fine-grained features and complete the deficiency of region features.

There are two main directions for image segmentation. One is semantic segmentation: each pixel is divided into a category, and the image is segmented according to the category. The other is instance segmentation: each pixel is divided into an instance, and the area where the instance is located is segmented. Panoptic segmentation [33] is a collection of two directions. Panoptic segmentation requires that each pixel in an image be assigned a semantic label and an instance id. The semantic label refers to the category of the object, while the instance id corresponds to the different number of the same object. For image captioning, it is not enough that we only know what objects are in a certain area box. We also need to know how many objects are the same and what are the characteristics of these objects. The Panoptic FPN network can accurately identify which class each pixel belongs to and distinguish different individuals of the same class. Individuals differ because of their colors, materials and other attributes. The Panoptic FPN network can distinguish different individuals, so the fine-grained features extracted by Panoptic FPN contain attribute information, which is more conducive to the visual semantic description. The following describes the process of extracting fine-grained features.

Given image I, feature map X is obtained after ResNet. X obtains the required fine-grained features through the instance segmentation branch.

Instance segmentation branch: As shown in Fig. 3, this method combines the Faster R-CNN network and the Mask R-CNN [34] network. X is a set of pyramid features, which are passed through the Region Proposal Network (RPN) to obtain 1000 candidate boxes. According to the candidate boxes obtained, $7 \times 7$ region of interest (RoI) pooling is performed on different pyramid layers, and a refined box and class label for each region are predicted through multiple fully connected layers. However, such a process cannot obtain the required segmentation features. To output instance segmentation features, this method refers to Mask R-CNN, which extends Faster R-CNN by adding a Fully Convolutional Network (FCN) [35] branch. RoI pooling ($14 \times 14$) is performed on different pyramid layers according to the previous refinement boxes, and then multiple convolutions are used to predict the binary segmentation mask and its category for each candidate region. Multilayer convolution includes 4 groups of $3 \times 3$ convolution, ReLU, and a group of $2 \times 2$ transpose convolution, ReLU, and $1 \times 1$ convolution. The segmentation feature size is obtained as $N \times 80 \times 28 \times 28$, where N means that the image has N regions, 80 means 80 classes and $28 \times 28$ is the feature size. According to the obtained categories, the features of this class are selected from 80 features, and the features of $N \times 1 \times 28 \times 28$ are finally obtained, which are reshaped into the size of $N \times 784$ to obtain the required fine-grained features.

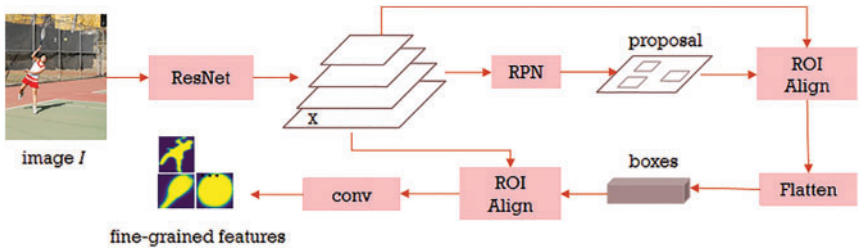


**Figure 3:** Detailed extraction methods for fine-grained features

### *3.2 Feature Fusion*

Region features and fine-grained features are two similar but different features. How to fuse these two features is a problem. In this paper, we propose two ways to fuse this feature and verify the effect of these two methods in the subsequent experiment. These two methods are shown in Fig. 1. Section 3.2.1 describes the direct fusion method and Section 3.2.2 describes the two-branch fusion method.

#### *3.2.1 Direct Fusion*

The direct fusion method is the first fusion method we employ. We concatenate two features directly into the baseline network. This method can visually see the impact of the features we extract on the results and verify the effectiveness of the features. The baseline is the X-LAN proposed by Yingwei Pan et al., and its key innovation point is the X-linear attention block.

X-linear attention block: The attention module of the traditional attention mechanism can be described as calculating the similarity scores of some queries (Q) and keys (K) and generating the weighted output of value (V) based on the similarity scores, where Q, K and V are all vectors and can be expressed as:

$$Attention\,(Q, K, V) = softmax\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{1}$$

The biggest difference between X-linear attention and traditional attention is that they use element multiplication instead of matrix multiplication to compute the similarity vectors between queries and keys. It can be expressed as:

$$X - Attention\,(Q, K, V) = softmax\,(FFN\,(K \odot V)) \odot V \odot R\,(K \odot V) \tag{2}$$

$$FFN\,(x) = max\left(0,\ x\ W^{F1} + b^{F1}\right)\ W^{F2} + b^{F2} \tag{3}$$

$$R\,(x) = \sigma\ \left(mean\left(\ max\left(0,\ x\ W^{R1} + b^{R1}\right)\right)\ W^{R2} +\ b^{R2}\right) \tag{4}$$

where $W^{F1}\,W^{F2}\,W^{R1}\,W^{R2}$ are embedding matrices, $\sigma$ denotes the sigmoid unit, and $\odot$ represents elementwise multiplication.

Pan et al. integrated the X-linear attention block into the encoder-decoder module. The encoder module is based on the Basic Transformer encoder, and the decoder module is based on the Basic Transformer decoder combined with Long Short Term Memory (LSTM). The X-LAN encoder contains $M + 1$ X-linear attention blocks. Through the feature extraction module, we can obtain features R and F, which represent the region feature and the fine-grained feature, respectively. The direct fusion method is shown in Fig. 4a. The input of the X-LAN encoder is $K = V = concat\,(R,\ F)$, $Q = \frac{1}{N}\sum_{i=1}^{N} V_i$. Assume that the output of each block is $g^i\ when\ i \geq 1$, and the final output of the X-LAN encoder is $A$ and $G = LayerNorm\left(W_G\,[Q,\ g^1,\ g^2,\ \ldots,\ g^M]\right)$. $W_G$ is the embedding matrix.

#### *3.2.2 Two-Branch Fusion*

The region feature and fine-grained feature focus on different details, but both contain information about the target and background, so direct splicing will cause redundant information. Moreover, different extraction methods make the number and type of objects and the size of features extracted from the same image different. Therefore, using the two-branch method can make two similar but different features learn independently and influence each other to better express the features.

The two-branch method is shown in Fig. 4b. After embedding, R and F are entered into the region encoder and fine-grained encoder, respectively. Embedding enables R and F to be projected to the same dimension of 1024 from 2048 and 784 dimensions, respectively. At the same time, the similar coding structure enables the two features to be mo closely expressed. Embedding includes a linear layer, ReLU, and group normalization. The internal structure of both encoders is based on the X-LAN encoder. The input of the region encoder is $K = V = R$, $Q = \frac{1}{N}\sum_{i=1}^{N} R_i$ and the output is $A_r$, $G_r$. The fine-grained encoder indicates that the input is different from that of the region encoder. The input is $K = V = F$, $Q = G_r$ and the output is $A_f$, $G_f$. For the transformer-based caption model, the encoder part is the further extraction of image features, while the decoder part is the image-to-text conversion, so we choose to fuse the two features before entering the decoder. We will directly concatenate $A_r$ and $A_f$ send them to the decoder of the X-LAN together with $G_f$, and finally obtain a sentence that conforms to visual semantics.

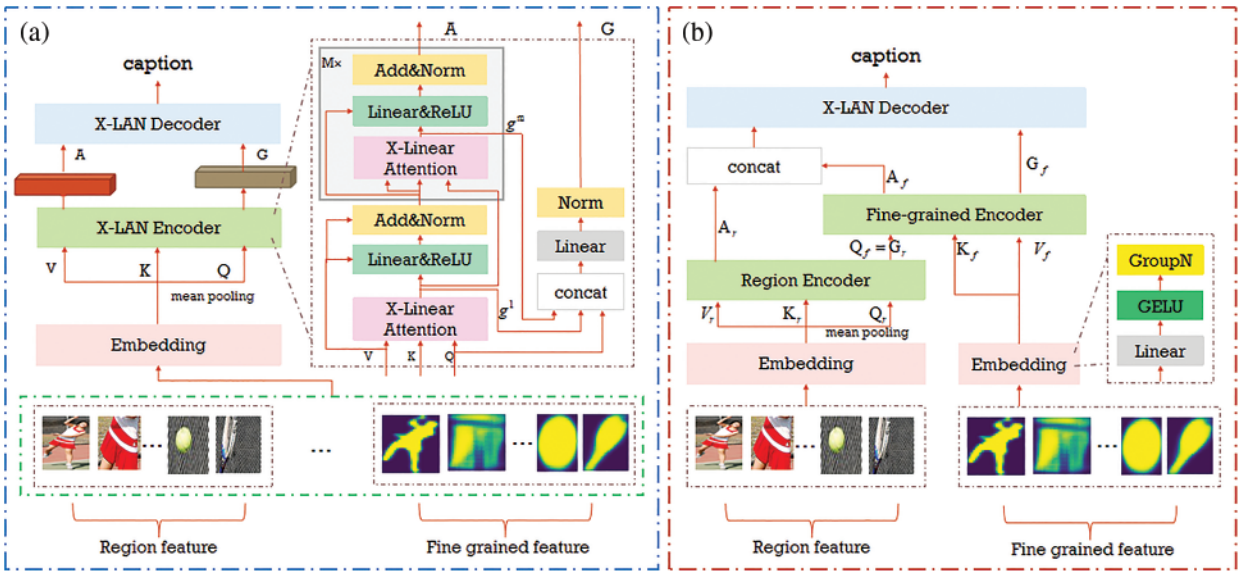


**Figure 4:** Details of the two fusion methods. (a) is the direct fusion method. (b) is the two-branch fusion method that feeds the region features into the region encoder and the fine-grained feature into the fine-grained encoder (both of which are based on the X-LAN encoder) and shares some information to obtain the image compiling result

## 4 Experiments

### *4.1 Dataset*

Our experiments are conducted on the most popular image captioning benchmark, COCO. The COCO dataset contains 123,287 images, including 82,783 training images, 40,504 validation images, and 40,775 test images. Each image has five reference captions. We follow the widely adopted Karpathy et al. [36] split to repartition COCO 2014 with 113,287 images for training, 5000 images for validation, and 5000 images for offline evaluation. We preprocessed the sentences in the training set, converted them to lowercase, and deleted the words that appeared less than six times, leading to the final vocabulary with 9,488 unique words. To evaluate our proposed model, we used five

widely accepted standard automatic evaluation metrics to evaluate the quality of generated sentences, including BLEU, ROUGE, ROUGE-L, CIDEr and SPICE.

### *4.2 Implementation Details*

We use Faster R-CNN to extract image region features and use Panoptic FPN to generate fine-grained features. The region feature is a 2048-dimensional vector, and the fine-grained feature is a 784-dimensional vector. The specific training method is as follows. We follow the training plan in [5] and optimize the whole architecture under the condition of cross-entropy loss. The batch size is set as 10, the iteration is 100, and the warm-up step is set as 10000. In addition, we adopt a planned sampling strategy [37], where the probability increases linearly by 0.05 every six periods from 0 to 0.5. After that, we used a self-critical training strategy, followed the training plan in [38], set the learning rate as 0.00001, and used the CIDEr reward to further optimize the whole model when the maximum number of iterations was 60.

### *4.3 Performance Comparison*

#### *4.3.1 Offline Evaluation*

Tables 1–4 summarizes the performance of the state-of-the-art models and our approach on the offline COCO Karpathy test split. We report the results optimized with both cross entropy loss and CIDEr score. Meanwhile, we separately show the performances for single and ensemble/fused models. Our baseline is X-LAN, so our main comparison object is X-LAN.

**Table 1:** Performance comparisons of standard cross-entropy loss for a single model, where B@N, M, R, C and S are short for BLEU @N, METEOR, ROUGE-L, CIDEr and SPICE scores

| Metric | B@1 | B@2 | B@3 | B@4 | M | R | C | S |
|---|---|---|---|---|---|---|---|---|
| LSTM [1] | - | - | - | 29.6 | 25.2 | 52.6 | 94.0 | - |
| SCST [38] | - | - | - | 30.0 | 25.9 | 53.4 | 99.4 | - |
| LSTM-A [15] | 75.4 | - | - | 35.2 | 26.9 | 55.8 | 108.8 | 20.0 |
| RFNet [39] | 76.4 | 60.4 | 46.6 | 35.8 | 27.4 | 56.8 | 112.5 | 20.5 |
| Up-Down [2] | 77.2 | - | - | 36.2 | 27.0 | 56.4 | 113.5 | 20.3 |
| GCN-LSTM [40] | 77.3 | - | - | 36.8 | 27.9 | 57.0 | 116.3 | 20.9 |
| LBPF [41] | 77.8 | - | - | 37.4 | 28.1 | 57.5 | 116.4 | 21.2 |
| SGAE [42] | 77.6 | - | - | 36.9 | 27.7 | 57.2 | 116.7 | 20.9 |
| AoANet [28] | 77.4 | - | - | 37.2 | 28.4 | 57.5 | 119.8 | 21.3 |
| X-LAN [5] | 78.0 | 62.3 | 48.9 | 38.2 | 28.8 | 58.0 | **122.0** | 21.9 |
| Our | **78.5** | **63.1** | **49.4** | **38.5** | **28.9** | **58.2** | 121.1 | **22.0** |

Tables 1–4 shows our single and ensemble/fused models consistently perform better than other models, which include the RNN baselines (LSTM, LSTM-A), transformer-based methods (AoANet, X-LAN) and others. Through CIDEr score optimization, our single model is competitive with X-LAN on CIDEr while our ensemble/fused model improves X-LAN by 0.6%. The performance improvements demonstrate the key advantage of the fine-grained features and the proposed two-branch fusion method. LSTM proposes to use LSTM to generate natural sentences and LSTM-A improves LSTM by focusing on semantic attributes. Hierarchy Parsing (HIP) [14] used a tree structure to segment the

image and obtain multi-level features. Self-critical Sequence Training (SCST) [38] proposed optimizing image captioning systems using reinforcement learning. Recurrent Fusion Network (RFNet) [39] introduces multiple encoders and fusion LSTM to improve sentence quality. Up-Down extracts the region features from the image to enrich the description. Graph Convolutional Networks (GCN)-LSTM [40] and Look Back Predict Forward (LBPF) [41] are improvements over LSTM. Auto-encoding Scene Graphs (SGAE) [42] incorporates the language inductive bias into the caption model for more human-like captions. AoANet expands the attention of the caption model through two linear lines, which is conducive to modeling and fusing information of different modes (such as text and image). X-LAN proposed X-linear attention blocks to optimize the caption model, which can capture information about higher-order internal modes and multimodal interactions. Our approach is to optimize image features. The success of our method shows that enriching the details of image features is an effective way to improve sentence quality.

**Table 2:** Performance comparisons of standard cross-entropy loss for ensemble/fused models

| Metric | B@1 | B@2 | B@3 | B@4 | M | R | C | S |
|---|---|---|---|---|---|---|---|---|
| SCST [38] | - | - | - | 32.8 | 26.7 | 55.1 | 106.5 | - |
| RFNet [39] | 77.4 | 61.6 | 47.9 | 37.0 | 27.9 | 57.3 | 116.3 | 20.8 |
| GCN-LSTM [40] | 77.4 | - | - | 37.1 | 28.1 | 57.2 | 117.1 | 21.1 |
| SGAE [42] | - | - | - | - | - | - | - | - |
| HIP [14] | - | - | - | 38.0 | 28.6 | 57.8 | 120.3 | 21.4 |
| AoANet [28] | 78.7 | - | - | 38.1 | 28.5 | 58.2 | 122.7 | 21.7 |
| X-LAN [5] | 78.8 | 63.4 | 49.9 | 39.1 | 29.1 | 58.5 | **124.5** | 22.2 |
| Our | **79.1** | **63.8** | **50.2** | **39.2** | **29.1** | **58.6** | 124.1 | **22.2** |

**Table 3:** Performance comparisons of CIDEr score optimization for a single model

| Metric | B@1 | B@2 | B@3 | B@4 | M | R | C | S |
|---|---|---|---|---|---|---|---|---|
| LSTM [1] | - | - | - | 31.9 | 25.5 | 54.3 | 106.3 | - |
| SCST [38] | - | - | - | 34.2 | 26.7 | 55.7 | 114.0 | - |
| LSTM-A [15] | 78.6 | - | - | 35.5 | 27.3 | 56.8 | 118.3 | 20.8 |
| RFNet [39] | 79.1 | 63.1 | 48.4 | 36.5 | 27.7 | 57.3 | 121.9 | 21.2 |
| Up-Down [2] | 79.8 | - | - | 36.3 | 27.7 | 56.9 | 120.1 | 21.4 |
| GCN-LSTM [40] | 80.5 | - | - | 38.2 | 28.5 | 58.3 | 127.6 | 22.0 |
| LBPF [41] | 80.5 | - | - | 38.3 | 28.5 | 58.4 | 127.6 | 22.0 |
| SGAE [42] | 80.8 | - | - | 38.4 | 28.4 | 58.6 | 127.8 | 22.1 |
| AoANet [28] | 80.2 | - | - | 38.9 | 29.2 | 58.8 | 129.8 | 22.4 |
| X-LAN [5] | 80.8 | 65.6 | 51.4 | 39.5 | 29.5 | **59.2** | 132.0 | 23.4 |
| Our | **80.9** | **65.8** | **51.6** | **39.7** | **29.6** | 59.1 | **132.0** | **23.4** |

**Table 4:** Performance comparisons of CIDEr score optimization for ensemble/fused models

| Metric | B@1 | B@2 | B@3 | B@4 | M | R | C | S |
|---|---|---|---|---|---|---|---|---|
| SCST [38] | - | - | - | 35.4 | 27.1 | 56.6 | 117.5 | - |
| RFNet [39] | 80.4 | 64.7 | 50.0 | 37.9 | 28.3 | 58.3 | 125.7 | 21.7 |
| GCN-LSTM [40] | 80.9 | - | - | 38.3 | 28.6 | 58.5 | 128.7 | 22.1 |
| SGAE [42] | 81.0 | - | - | 39.0 | 28.4 | 58.9 | 129.1 | 22.2 |
| HIP [14] | - | - | - | 39.1 | 28.9 | 59.2 | 130.6 | 22.3 |
| AoANet [28] | 81.6 | - | - | 40.2 | 29.3 | 59.4 | 132.0 | 22.8 |
| X-LAN [5] | 81.6 | 66.6 | 52.3 | 40.3 | 29.8 | 59.6 | 133.7 | 23.6 |
| Our | **81.8** | **66.8** | **52.6** | **40.7** | **29.8** | **59.7** | **134.3** | **23.8** |

### *4.3.2 Online Evaluation*

We use the ensemble versions to generate captions on the official testing set to the online testing server. Table 5 details the performances over official testing images with 5 reference captions (c5) and 40 reference captions (c40). For online evaluation, we ensemble 4 models and adopt the backbone ResNet-101. The results clearly show that compared to the baseline X-LAN (ResNet-101), our method exhibits better performance across most metrics.

**Table 5:** Comparison with other image captioning models published on COCO online test server

| Model | B@1 | | B@2 | | B@3 | | B@4 | | M | | R | | C | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Metric | c5 | c40 | c5 | c40 | c5 | c40 | c5 | c40 | c5 | c40 | c5 | c40 | c5 | c40 |
| LSTM-A [15] | 78.7 | 93.7 | 62.7 | 86.7 | 47.6 | 76.5 | 35.6 | 65.2 | 27.0 | 35.4 | 56.4 | 70.5 | 116.0 | 118.0 |
| Up-Down [2] | 80.2 | 95.2 | 64.1 | 88.8 | 49.1 | 79.4 | 35.2 | 68.5 | 27.6 | 36.7 | 56.3 | 72.4 | 117.9 | 120.5 |
| RFNet [39] | 80.4 | 95.0 | 64.9 | 89.3 | 50.1 | 80.1 | 38.0 | 69.2 | 28.2 | 35.4 | 58.2 | 73.1 | 122.9 | 125.1 |
| SGAE [42] | 81.0 | 95.3 | 65.6 | 89.5 | 50.7 | 80.4 | 38.5 | 69.7 | 28.2 | 37.2 | 58.6 | 73.6 | 123.8 | 126.5 |
| GCN-LSTM [40] | 80.8 | 95.2 | 65.5 | 89.3 | 50.8 | 80.3 | 38.7 | 69.7 | 28.5 | 37.6 | 58.5 | 73.4 | 125.3 | 126.5 |
| AoANet [28] | 81.0 | 95.0 | 65.8 | 89.6 | 51.4 | 81.3 | 39.4 | 71.2 | 29.1 | 38.5 | 58.9 | 74.5 | 126.9 | 129.6 |
| HIP [14] | 81.6 | 95.9 | 66.2 | 90.4 | 51.5 | 81.6 | 39.3 | 71.0 | 28.8 | 38.1 | 59.0 | 74.1 | 127.9 | 130.2 |
| X-LAN [5] | 81.1 | 95.3 | 66.0 | 89.8 | 51.5 | 81.5 | 39.5 | 71.4 | **29.4** | 38.9 | 59.2 | 74.7 | 128.0 | 130.3 |
| Ours | **81.4** | **95.6** | **66.2** | **90.2** | **51.7** | **81.9** | **39.7** | **71.8** | 29.3 | **38.9** | **59.2** | **74.9** | **128.2** | **130.9** |

### *4.3.3 Qualitative Analysis and Visualization*

Fig. 5 shows a few examples of image captioning results by baseline and our proposed method with ground truth (GT) sentences. As these examples show, our extracted features provide richer information for the region features, which can help the network better capture the key information in the image. According to examples 1–4, we can see that all the keywords in GT can be reflected in our sentences, while X-LAN can only capture part of them. For example, "boat" in 1, "television" in 2, "grassy" in 3, and "kitchen" in 4 cannot be captured by baseline. In addition, due to the fine-grained features we extract, we can capture more details and generate more representational expressions. For example, in 5–6, although the baseline captures the key vocabulary, the description of the target is not specific enough, while our method uses a rich vocabulary to describe the attributes and states of the target as much as possible.

Furthermore, a comparison with GT demonstrates that neither our method nor the baseline accurately describes the background. For instance, in the first image, numerous boats are depicted as background since they are far away; therefore, in our description, we will refer to several boats as one boat. Additionally, our method is unable to capture "temples" in 3 and "wall" in 6. It is challenging to concentrate on the background information because both fine-grained and region features are more object-focused. In the following trials, we attempted to extract background features, however, the sample fusion methods won't be effective for this feature, necessitating additional research.

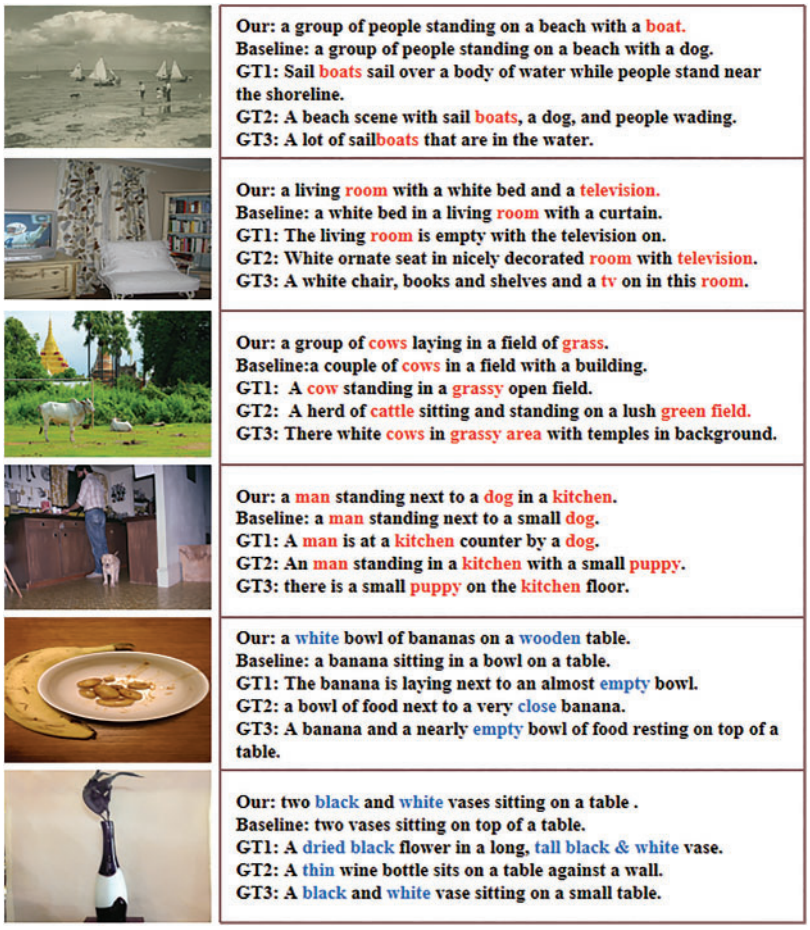


**Figure 5:** Examples of image captioning results by X-LAN and our method, coupled with the corresponding ground truth sentences

### *4.3.4 Ablation Study*

We performed several ablation studies to compare the two fusion methods. In addition, we record some attempts in the process of extracting features.

**Method** We propose two simple fusion methods: the direct fusion method and the two-branch fusion method. Table 6 shows the impact of these two approaches on the results. The two-branch method will turn out to be superior to the other. Since both region features and fine-grained features are object-focused, they have a lot of information in common. While the two-branch fusion method

can more clearly distinguish between the two aspects and complement each other, the direct fusion method will render this similar information redundant. In the table, R is the region feature, and F is the fine-grained feature.

**Table 6:** Ablation study on the use of two fusion methods (Cross-entropy optimization)

| Metric | B@1 | B@2 | B@3 | B@4 | M | R | C | S |
|---|---|---|---|---|---|---|---|---|
| X-LAN (baseline) | 78.0 | 62.3 | 48.9 | 38.2 | 28.8 | 58.0 | **122.0** | 21.9 |
| Concat (R, F) | 78.3 | 63.0 | 49.3 | 38.3 | 28.8 | 58.1 | 121.5 | 21.9 |
| Two-branch (R, F) | **78.5** | **63.1** | **49.4** | **38.5** | **28.9** | **58.2** | 121.1 | **22.0** |

**Feature** In 4.3.3, we analyze the gap between our method and GT. To bridge this gap, we extracted not only fine-grained features but also background features during the experiment. Pyramid features enter the instance segmentation branch to obtain fine-grained features, while pyramid features enter the semantic segmentation branch to obtain background features.

Semantic segmentation branch: This branch upsamples the pyramid features to the same scale. Fig. 6 illustrates this in detail. Starting at the deepest FPN level (1/32 scale), we perform three upsampling stages to generate feature maps at the 1/4 scale, where each upsampling stage includes $3 \times 3$ convolution, group norm, ReLU, and $2\times$ bilinear upsampling. This strategy is repeated for FPN scales 1/16, 1/8, and 1/4 (gradually reduced in the incremental sampling phase). The result is a set of feature maps with the same ratio of 1/4, summed by the elements, and then passed through a $1 \times 1$ convolution to obtain the semantic feature. Finally, $1 \times 1$ convolution, $4\times$ bilinear upsampling, and softmax were used to generate per-pixel category labels at the original image resolution. Then, the pixels belonging to the background category are selected to synthesize the background segmentation maps. Each pixel in the semantic feature takes the maximum value in 54 channels to obtain the feature with channel 1. According to the background segmentation maps, the corresponding region is segmented on the semantic feature to obtain the background feature.

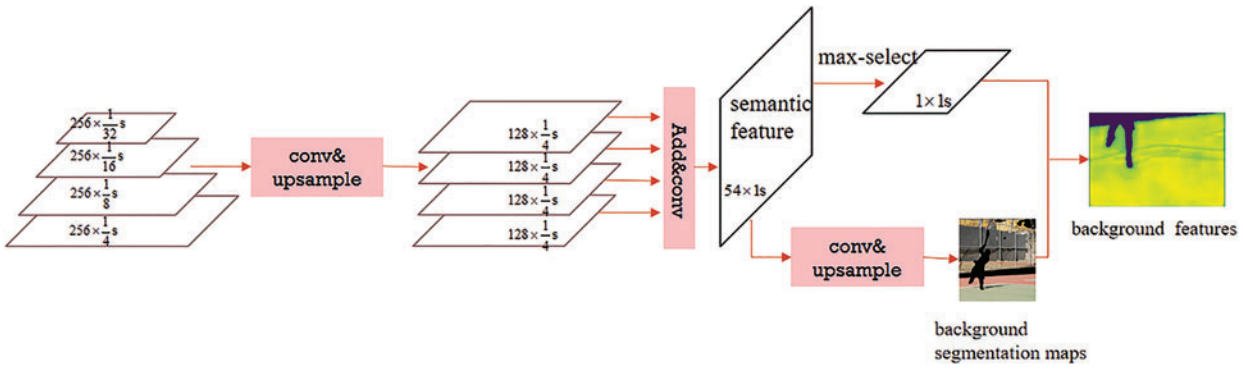


**Figure 6:** Detailed extraction methods for background features. Figure adapted from [13]

We experimented with both features. As shown in Table 7, we find that fine-grained features have a large influence on the indicators Blue but will reduce CIDEr's scores. However, background features have an impact on each metric, but the impact is small. Overall, fine-grained features are better. At the same time, we also tested the fusion of these two features with region features and found that the results were not ideal. Therefore, we choose to use fine-grained features and adopt the two-branch

approach as the final method. In Table 7, R is the region feature, F is the fine-grained feature, and B is the background feature.

**Table 7:** Ablation study on the use of two features (Cross-entropy optimization)

| Metric | B@1 | B@2 | B@3 | B@4 | M | R | C | S |
|---|---|---|---|---|---|---|---|---|
| Concat (R, F) | 78.3 | 63.0 | 49.3 | 38.3 | 28.8 | 58.1 | 121.5 | 21.9 |
| Concat (R, B) | 78.1 | 62.6 | 49.1 | 38.3 | 28.9 | 58.1 | **122.1** | 22.0 |
| Two-branch (R, F) | **78.5** | **63.1** | **49.4** | **38.5** | **28.9** | **58.2** | 121.1 | **22.0** |
| Two-branch (R, B) | 78.4 | 62.9 | 49.1 | 38.1 | 28.5 | 57.9 | 120.8 | 21.7 |
| Concat (R, F + B) | 78.4 | 62.8 | 49.0 | 38.1 | 28.7 | 57.8 | 121.5 | 22.0 |
| Two-branch (R, F + B) | 78.3 | 62.7 | 49.0 | 38.0 | 28.8 | 58.0 | 121.4 | 21.8 |

## 5 Conclusion

In this paper, we propose a method that uses the fusion of fine-grained features and region features to generate captions more accurately. We extract fine-grained features using the Panoptic FPN algorithm, which offers a fresh perspective on feature extraction. Panoptic FPN can capture multi-scale information so that more details can be focused. To make the two features better fusion, we adopted two common fusion methods and compared the two methods. We have performed extensive experiments on the COCO dataset, and the results show that the effectiveness of our extracted features and the use of the two-branch method to merge region features and fine-grained can exceed the baseline on most metrics, and CIDEr reaches 134.3% on the COCO Karpathy test split.

In terms of features, we discover that region features and fine-grained features focus more on the target and less on the background. In the experimental section, we extracted background features, however, the fusion result was poor. This gives us two directions for future research: (1) We plan to use other methods to extract background features; (2) We plan to completely exploit background features with the aid of a novel fusion method in order to further improve the performance of the model.

**Acknowledgement:** The authors extend their appreciation to the anonymous reviewers for their constructive comments and suggestions.

**Funding Statement:** This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grant 6150140, and in part by the Youth Innovation Project (21032158-Y) of Zhejiang Sci-Tech University.

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

## References

[1] O. Vinyals, A. Toshev, S. Bengio and D. Erhan, "Show and tell: A neural image caption generator," in *Proc. the IEEE Conf. on Computer Vision and Pattern Recognition*, Boston, MA, USA, pp. 3156–3164, 2015.

[2] P. Anderson, X. He, C. Buehler, D. Teney, M. Johnson *et al.,* "Bottom-up and top-down attention for image captioning and visual question answering," in *Proc. the IEEE Conf. on Computer Vision and Pattern Recognition*, Salt Lake City, UT, USA, pp. 6077–6086, 2018.

[3] S. Mahajan and A. K. Pandit, "Image segmentation and optimization techniques: A short overview," *Medicon Engineering Themes*, vol. 2, no. 2, pp. 47–49, 2022.
[4] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones *et al.,* "Attention is all you need," in *Proc. Advances in Neural Information Processing Systems*, Long Beach, CA, USA, pp. 5998–6008, 2017.
[5] Y. Pan, T. Yao, Y. Li and T. Mei, "X-linear attention networks for image captioning," in *Proc. the IEEE Conf. on Computer Vision and Pattern Recognition*, Seattle, WA, USA, pp. 10971–10980, 2020.
[6] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," in *Proc. Advances in Neural Information Processing Systems*, Montreal, Quebec, Canada, pp. 568–576, 2014.
[7] T. Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona *et al.,* "Microsoft coco: Common objects in context," in *Proc. European Conf. on Computer Vision*, Zurich, Switzerland, pp. 740–755, 2014.
[8] K. Papineni, S. Roukos, T. Ward and W. J. Zhu, "Bleu: A method for automatic evaluation of machine translation," in *Proc. the 40th Annual Meeting of the Association for Computational Linguistics*, Philadelphia, PA, USA, pp. 311–318, 2002.
[9] M. Denkowski and A. Lavie, "Meteor universal: Language specific translation evaluation for any target language," in *Proc. of the Ninth Workshop on Statistical Machine Translation*, Baltimore, Maryland, USA, pp. 376–380, 2014.
[10] C. Y. Lin, "Rouge: A package for automatic evaluation of summaries," in *Text Summarization Branches Out: Proc. of the ACL-04 Workshop*, Barcelona, Spain, pp. 74–81, 2004.
[11] R. Vedantam, Z. C. Lawrence and D. Parikh, "Cider: Consensus-based image description evaluation," in *Proc. the IEEE Conf. on Computer Vision and Pattern Recognition*, Boston, MA, USA, pp. 4566–4575, 2015.
[12] P. Anderson, B. Fernando, M. Johnson and S. Gould, "Spice: Semantic propositional image caption evaluation," in *Proc. European Conf. on Computer Vision*, Amsterdam, Netherlands, pp. 382–398, 2016.
[13] A. Kirillov, R. Girshick, K. He and P. Dollár, "Panoptic feature pyramid networks," in *Proc. the IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, Long Beach, CA, USA, pp. 6399–6408, 2019.
[14] T. Yao, Y. Pan, Y. Li and T. Mai, "Hierarchy parsing for image captioning," in *Proc. the IEEE/CVF Int. Conf. on Computer Vision*, Seoul, South Korea, pp. 2621–2629, 2019.
[15] T. Yao, Y. Pan, Y. Li, Z. Qiu and T. Mai, "Boosting image captioning with attributes," in *Proc. the IEEE Conf. on Computer Vision*, Venice, Italy, pp. 4894–4902, 2017.
[16] J. Lu, C. Xiong, D. Parikh and R. Socher, "Knowing when to look: Adaptive attention via a visual sentinel for image captioning," in *Proc. the IEEE Conf. on Computer Vision and Pattern Recognition*, Honolulu, HI, USA, pp. 375–383, 2017.
[17] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh *et al.,* "Learning transferable visual models from natural language supervision," in *Proc. Int. Con. on Machine Learning*, Virtual Event, pp. 8748–8763, 2021.
[18] R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata *et al.,* "Visual genome: Connecting language and vision using crowdsourced dense image annotations," *International Journal of Computer Vision*, vol. 123, no. 1, pp. 32–73, 2017.
[19] H. Jiang, I. Misra, M. Rohrbach, E. Learned-Miller and X. Chen, "In defense of grid features for visual question answering," in *Proc. the IEEE Conf. on Computer Vision and Pattern Recognition*, Seattle, WA, USA, pp. 10267–10276, 2020.
[20] X. Zhang, X. Sun, Y. Luo, J. Ji, Y. Zhou *et al.,* "RSTNet: Captioning with adaptive attention on visual and non-visual words," in *Proc. the IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, Nashville, TN, pp. 15465–15474, 2021.
[21] Y. Luo, J. Ji, X. Sun, L. Cao, Y. Wu *et al.,* "Dual-level collaborative transformer for image captioning," in *Proc. the AAAI Conf. on Artificial Intelligence*, Arlington, Virginia, vol. 35, no.3, pp. 2286–2293, 2021.
[22] R. Mokady, A. Hertz and A. Bermano, "Clipcap: Clip prefix for image captioning," 2021. [Online]. Available: https://arxiv.org/abs/2111.09734
[23] Z. Deng, B. Zhou, P. He, J. Huang, O. Alfarraj *et al.,* "A position-aware transformer for image captioning," *Computers, Materials and Continua*, vol. 70, no. 1, pp. 2005–2021, 2021.
[24] N. Nejatishahidin, P. Fayyazsanavi and J. Kosecka, "Object pose estimation using mid-level visual representations," 2022. [Online]. Available: https://arxiv.org/abs/2203.01449

[25] W. Wang, Z. Chen and H. Hu, "Hierarchical attention network for image captioning," in *Proc. the AAAI Conf. on Artificial Intelligence*, Honolulu, Hawaii, USA, pp. 8957–8964, 2019.
[26] J. Devlin, M. W. Chang, K. Lee and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Human Language Technologies-Proc. of the Conf.*, Minneapolis, MN, USA, vol. 1, pp. 4171–4186, 2019.
[27] Z. Dai, Z. Yang, Y. Yang, J. G. Carbonell, Q. V. Le *et al.,* "Transformer-XL: Attentive language models beyond a fixed-length context," in *Proc. the 57th Annual Meeting of the Association for Computational Linguistics*, Florence, Italy, pp. 2978–2988, 2019.
[28] L. Huang, W. Wang, J. Chen and X. Wei, "Attention on attention for image captioning," in *Proc. the IEEE Conf. on Computer Vision*, Seoul, South Korea, pp. 4634–4643, 2019.
[29] L. Guo, J. Liu, X. Zhu, P. Yao, S. Lu *et al.,* "Normalized and geometry-aware self-attention network for image captioning," in *Proc. the IEEE Conf. on Computer Vision and Pattern Recognition*, Seattle, WA, USA, pp. 10327–10336, 2020.
[30] M. Cornia, M. Stefanini, L. Baraldi and R. Cucchiara, "Meshed-memory transformer for image captioning," in *Proc. the IEEE Conf. on Computer Vision and Pattern Recognition*, Seattle, WA, USA, pp. 10578–10587, 2020.
[31] L. Zhou, H. Palangi, L. Zhang, H. Hu, J. J. Corso *et al.,* "Unified vision-language pre-training for image captioning and vqa," in *Proc. the AAAI Conf. on Artificial Intelligence*, New York, USA, vol. 34, no. 7, pp. 13041–13049, 2020.
[32] S. Elbedwehy, T. Medhat, T. Hamza and M. F. Alrahmawy, "Efficient image captioning based on vision transformer models," *Computers, Materials and Continua*, vol. 73, no. 1, pp. 1483–1500, 2022.
[33] A. Kirillov, K. He, R. B. Girshick, C. Rother and P. Dollár, "Panoptic segmentation," in *Proc. the IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, Long Beach, CA, USA, pp. 9404–9413, 2019.
[34] K. He, G. Gkioxari, P. Dollár and R. B. Girshick, "Mask r-cnn," in *Proc. the IEEE Conf. on Computer Vision*, Venice, Italy, pp. 2961–2969, 2017.
[35] J. Long, E. Shelhamer and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. the IEEE Conf. on Computer Vision and Pattern Recognition*, Boston, MA, USA, pp. 3431–3440, 2015.
[36] A. Karpathy and F. Li, "Deep visual-semantic alignments for generating image descriptions," in *Proc. the IEEE Conf. on Computer Vision and Pattern Recognition*, Boston, MA, USA, pp. 3128–3137, 2015.
[37] S. Bengio, O. Vinyals, N. Jaitly and N. Shazeer, "Scheduled sampling for sequence prediction with recurrent neural networks," in *Proc. Advances in Neural Information Processing Systems*, Montreal, Quebec, Canada, vol. 28, pp. 1171–1179, 2015.
[38] S. J. Rennie, E. Marcheret, Y. Mroueh, J. Ross and V. Goel, "Self-critical sequence training for image captioning," in *Proc. the IEEE Conf. on Computer Vision and Pattern Recognition*, Honolulu, HI, USA, pp. 7008–7024, 2017.
[39] W. Jiang, L. Ma, Y. G. Jiang, W. Liu and T. Zhang, "Recurrent fusion network for image captioning," in *Proc. the European Conf. on Computer Vision*, Munich, Germany, vol. 2, pp. 499–515, 2018.
[40] T. Yao, Y. Pan, Y. Li and T. Mai, "Exploring visual relationship for image captioning," in *Proc. the European Conf. on Computer Vision*, Munich, Germany, pp. 711–727, 2018.
[41] Y. Qin, J. Du, Y. Zhang and H. Lu, "Look back and predict forward in image captioning," in *Proc. the IEEE Conf. on Computer Vision and Pattern Recognition*, Long Beach, CA, USA, pp. 8367–8375, 2019.
[42] X. Yang, K. Tang, H. Zhang and J. Cai, "Auto-encoding scene graphs for image captioning," in *Proc. the IEEE Conf. on Computer Vision and Pattern Recognition*, Long Beach, CA, USA, pp. 10685–10694, 2019.