Computers, Materials & Continua
DOI:10.32604/cmc.2022.019328
Article

Tech Science Press

A Position-Aware Transformer for Image Captioning

Zelin Deng[1,*], Bo Zhou[1], Pei He[2], Jianfeng Huang[3], Osama Alfarraj[4] and Amr Tolba[4,5]

[1]School of Computer and Communication Engineering, Changsha University of Science and Technology, Changsha, 410114, China
[2]School of Computer Science and Cyber Engineering, Guangzhou University, Guangzhou, 510006, China
[3]Advanced Forming Research Centre, University of Strathclyde, Renfrewshire, PA4 9LJ, Glasgow, United Kingdom
[4]Department of Computer Science, Community College, King Saud University, Riyadh, 11437, Saudi Arabia
[5]Department of Mathematics and Computer Science, Faculty of Science, Menoufia University, Egypt
[*]Corresponding Author: Zelin Deng. Email: zl_deng@sina.com

Received: 10 April 2021; Accepted: 16 June 2021

Abstract: Image captioning aims to generate a corresponding description of an image. In recent years, neural encoder-decoder models have been the dominant approaches, in which the Convolutional Neural Network (CNN) and Long Short Term Memory (LSTM) are used to translate an image into a natural language description. Among these approaches, the visual attention mechanisms are widely used to enable deeper image understanding through fine-grained analysis and even multiple steps of reasoning. However, most conventional visual attention mechanisms are based on high-level image features, ignoring the effects of other image features, and giving insufficient consideration to the relative positions between image features. In this work, we propose a Position-Aware Transformer model with image-feature attention and position-aware attention mechanisms for the above problems. The image-feature attention firstly extracts multi-level features by using Feature Pyramid Network (FPN), then utilizes the scaled-dot-product to fuse these features, which enables our model to detect objects of different scales in the image more effectively without increasing parameters. In the position-aware attention mechanism, the relative positions between image features are obtained at first, afterwards the relative positions are incorporated into the original image features to generate captions more accurately. Experiments are carried out on the MSCOCO dataset and our approach achieves competitive BLEU-4, METEOR, ROUGE-L, CIDEr scores compared with some state-of-the-art approaches, demonstrating the effectiveness of our approach.

Keywords: Deep learning; image captioning; transformer; attention; position-aware

1 Introduction

Image captioning [1] aims to describe the visual contents of an image in natural language, which is a sequence-to-sequence problem and can be viewed as translating an image into its corresponding descriptive sentence. With these characteristics, the model not only needs to be able to identify objects, actions, and scenes in the image, but also to be powerful enough to capture and express the relationships of these elements in a properly-formed sentence. This scheme analogically simulates the extraordinary abilities of humans to convert large amounts of visual information into descriptive semantic information.

Earlier captioning approaches [2,3] used some unsophisticated templates and two auxiliary modules object detector and attribute detector. The two detectors filled the blank items of the templates to generate a complete sentence. According to the great successes achieved by deep neural networks [4] in computer vision [5,6] and natural language processing [7,8], a broad collection of image captioning methods has been proposed [1,9,10]. Based on the neural encoder-decoder framework [1], these methods use the Convolutional Neural Network (CNN) [4] to encode the input image into image features. Subsequently, the Recurrent Neural Network (RNN) [11] is applied to decode these features word-by-word into a natural language description of the image.

However, there are two major drawbacks in the plain encoder-decoder based models as follows: (1) the image representation does not change during the caption generation process; (2) The decoder processes the image representation from a global view, rather than focusing on local aspects related to parts of the description. The visual attention mechanisms [12–15] can solve these problems by dynamically attending to different parts of image features relevant to the semantic context of the current partially-completed caption.

RNN-based caption models have become the dominant approaches in recent years, but the recurrent structure of RNN makes models suffer from gradient-vanishing or gradient-exploding with the growth of sentence and precludes parallelization within training examples. Recently, the work of Vaswani et al. [16] shows that the transformer has excellent performance on machine translation or other sequence-to-sequence problems. It is based on the self-attention mechanism and enables models to be trained in parallel by excluding recurrent structures.

Human-like and descriptive captions require the model to describe primary objects in the image and also present their relations in a fluent style. While image features obtained by CNN commonly correspond to a uniform grid of equally-sized image regions, each feature only contains information in its corresponding region, irrespective of the relative positions with any other features. Thus, it is hard to get an accurate expression. Furthermore, these image features are mainly visual features extracted from a global view of the image, and only contain a small amount of local visual features that are crucial for detecting small objects. Such limitations of image features keep the model from producing more human-like captions.

In order to obtain captions of superior quality, a Position-aware Transformer model for image captioning is proposed. The contributions of this model are as follows: (1) To enable the model to detect objects of different scales in the image without increasing the number of parameters, the image-feature attention is proposed, which uses the scaled-dot-product to fuse multi-level features within an image feature pyramid; (2) To generate more human-like captions, the position-aware attention is proposed to learn relative positions between image features, making features can be explained from the perspective of spatial relationship.

The rest of this paper is organized as follows. In Section 2, the previous critical works about image captioning and the transformer architecture are briefly introduced. In Section 3, the overall architecture and the details of our approach are introduced. In Section 4, the results of the experiment on the COCO dataset are reported and analyzed. In Section 5, the contributions of our work are concluded.

2 Related Works

2.1 *Image Captioning and Attention Mechanism*

Image captioning is the task of generating a descriptive sentence of an image. It requires an algorithm to understand and model the relations between visual and textual elements. With the development of deep learning, a variety of methods based on deep neural networks have been proposed. Vinyals et al. [1] firstly proposed an encoder-decoder framework, which used the CNN as the encoder and the RNN as the decoder. However, the input of RNN was a consistent representation of an image, and this representation was generally analyzed from an overall perspective, thus leading to a mismatch between the context of visual information and the context of semantic information.

To solve the above problems, Xu et al. [12] introduced the attention mechanism for image captioning, which guided the model to different salient regions of the image dynamically at each step, instead of feeding all image features to the decoder at the initial step. Based on Xu's work, more and more improvements in attention mechanisms have been developed. Chen et al. [13] proposed spatial and channel-wise attention, in which the attention mechanism calculated where (spatial locations at multiple layers) and what (channels) the visual attention was. Anderson et al. [14] proposed a combined bottom-up and top-down visual attention mechanism. The bottom-up mechanism chose a set of salient image regions through the object detection technology, the top-down mechanism used task-specific context to predict attention distribution of the chosen image regions. Lu et al. [15] proposed adaptive attention by adding a visual sentinel, determining when to attend to an image or the visual sentinel.

2.2 *Transformer and Self-Attention Mechanism*

Recurrent models have some limitations on parallel computation and have gradient-vanishing or gradient-exploding problems when trained with long sentences. Vaswani et al. [16] proposed the transformer architecture and achieved state-of-the-art results for machine translation. Experimental results showed that the transformer was superior in quality while being more parallelizable and requiring significantly less time to be trained. Recently, the work in [17,18] applied the transformer to the task of image captioning and improved the model performance. Without recurrence, the transformer uses the self-attention mechanism to compute the relation of two arbitrary elements of a single input, and outputs a contextualized representation of this input, avoiding the vanishing or exploding gradients and accelerating the training process.

2.3 *Relative Position Information*

Most attention mechanisms for image captioning attend to CNN features at each step [12,13], while CNN features do not contain relative position information. This makes relative position information unavailable during the caption generation process. However, not all the words have corresponding CNN features. Consider Fig. 1a and its ground truth caption “A brown toy horse stands on a red chair”. The words “stand” and “on” do not have corresponding CNN features, but can be determined by the relative position information between CNN features (see Fig. 1b). Therefore, we developed the position-aware attention to learn relative position information during training.

3 The Proposed Approach

To generate more reasonable captions, a Position-aware Transformer model is proposed to make full use of the relative position information. It contains two components: the image encoder, and the caption decoder. As shown in Fig. 2, the combination of the Feature Pyramid Network (FPN) [19], image-feature attention, and position-aware attention is regarded as the encoder to obtain visual features. The decoder is the original transformer decoder. Given an image, the FPN is first leveraged to obtain two kinds of image features, one is high-level visual features containing the global semantics of the image, the other is low-level visual features which are local details of the image [19]. These two kinds of features are fed into the image-feature attention and position-aware attention to get fused features containing relative position information. Finally, the transformer takes the fused features and the start token <BOS> or the partially-completed sentence as input, and then outputs probabilities of each word in the dictionary being the next word of the sentence.

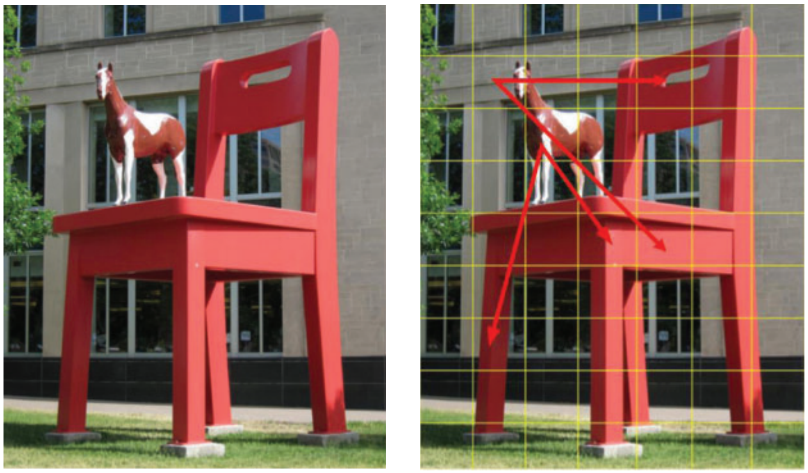
(a) (b)

Figure 1: Original image and relative position (a) Original image (b) Red arrows represent relative position information

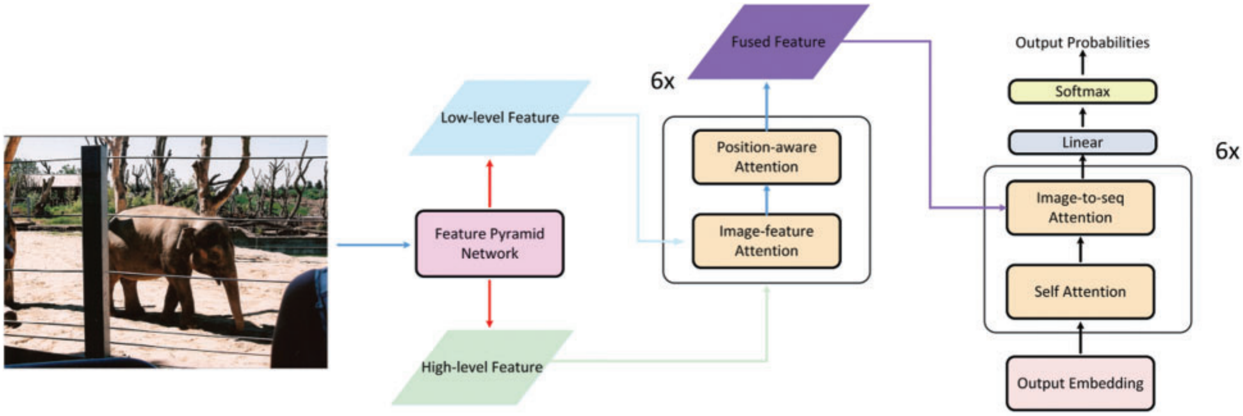

Figure 2: Overall structure of our proposed approach

3.1 Image-Feature Attention for Feature Fusion

The input of image captioning is an image. Traditional methods use a pre-trained CNN model on the image classification task as the feature extractor and mostly adopt the final conv-layer feature map as the image representation. However, not all objects in the image have corresponding features stored in this representation, particularly for those small-sized objects. As shown in Fig. 3.

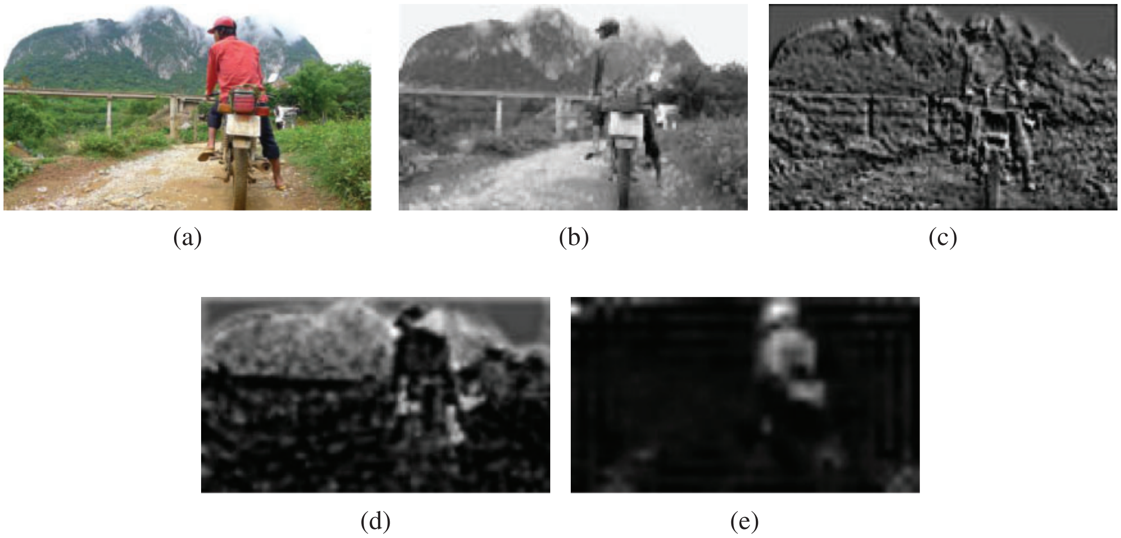

Figure 3: Original image and its features (a) Original image (b) The first-level feature (c) The second-level feature (d) The third-level feature (e) The fourth-level feature

Fig. 3a is the original image, and the others are image features having semantics from low-level to high-level. The lower the feature is, the more information it contains, and the weaker semantics it presents. Weaker semantics are harmful to the model to grasp the topic of the image; less information is negative for capturing the local details of the image. As a result, determining an optimal level of image features invariably leads to an unwinnable trade-off. To recognize image objects at different scales, we use the FPN model to construct a feature pyramid. Features in the pyramid combine low-resolution, semantically strong features with high resolution, semantically weak features via a top-down pathway and lateral connections. In this work, the feature pyramid has four feature maps in total. The first two are high-level features and the rest are low-level features.

Predicting on each level feature of a feature pyramid has many limitations, especially the inference time will increase considerably, making this approach impractical for real applications. Moreover, training deep networks end-to-end on all features is infeasible in terms of memory. To build an effective and lightweight model, we choose one feature from high-level features and low-level features respectively: $V^{low} = \{v_1^l, \ldots, v_m^l\}$, $v_i^l \in R^{d_{model}}$ and $V^{high} = \{v_1^h, \ldots, v_n^h\}$, $v_j^h \in R^{d_{model}}$, d_{model} is the hidden dimension of the model. Because low-level features are still too large to use (e.g., 4 times more than high-level features in spatial size), the image-feature attention is then used to fuse such two features according to Fig. 4.

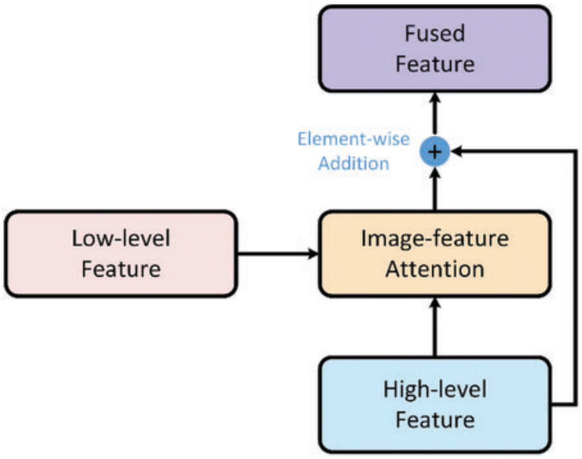

Figure 4: The structure of image-feature attention

As shown in Fig. 4, the image-feature attention takes V^{low} and V^{high} as input and firstly uses Eq. (1) to calculate the relevance-coefficients matrix C between elements in V^{low} and V^{high}.

$$C = \frac{\left(V^{high} W^{Q}\right)\left(V^{low} W^{K}\right)^{T}}{\sqrt{d_{model}}} \tag{1}$$

The relevance-coefficients matrix C is then used to compute attention weights $\mathcal{W}$ according to Eq. (2).

$$\mathcal{W} = softmax\left(C\right) \tag{2}$$

Finally, the attention weights $\mathcal{W}$ are applied to calculate a weighted sum of V^{low}, and the fused feature V^{fused} is computed by Eq. (3).

$$V^{fused} = V^{low}\mathcal{W}^{\mathcal{T}} + V^{high} \tag{3}$$

where d_{model} is the hidden dimension of our approach, W^{Q}, W^{K}, W^{V} are learnable parameters during the training process.

3.2 Position-Aware Attention

RNN networks capture relative positions between input elements directly through their recurrent structure. However, the recurrent structure is abandoned in the transformer to support the use of self-attention, and CNN features do not contain relative position information. As we mentioned earlier, relative position information is helpful for achieving an accurate expression, so introducing it explicitly is a considerably important step. When dealing with the machine translation task, the transformer manually introduces position information to the model using sinusoidal position coding. But sinusoidal position coding might not work for image captioning, because images and language sentences are two very different ways of describing things, images mainly contain visual information, while sentences mainly contain semantic information. In this work, rather than using an elaborated handwritten function as the transformer does, the position-aware attention is proposed to learn relative position information during training.

Because an image is split into a uniform grid of equally-sized regions from the perspective of image features, in this sense, we model the image features as a normative directed graph, see Fig. 5. Each vertex (the blue block in the image) stands for the feature of a certain image region, and each directed edge (the red arrows) denotes the relative position between two vertices. Note that in this graph all the edges are direct, because the relative positions from feature A to B are different from the relative positions from feature B to A.

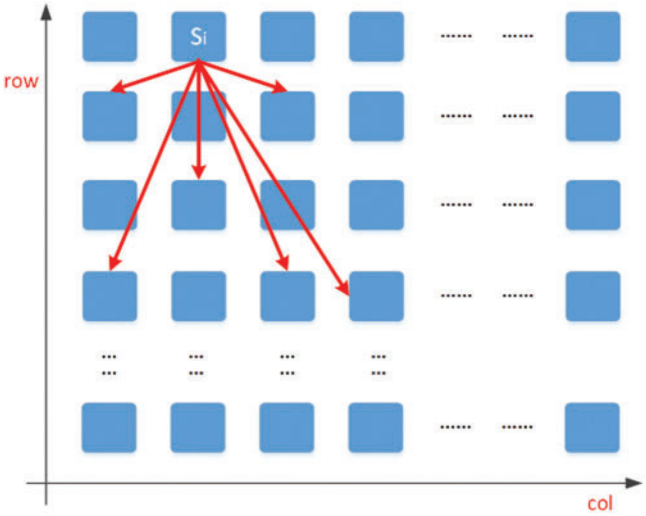

Figure 5: The directed graph model of image features

The position-aware attention takes two inputs, V^{fused}, and an edge matrix E in which each element E_{ij} represents the edge starts from vertex S_i to vertex S_j. In this case, we use Eq. (4) to calculate the relevance-coefficients within elements of V^{fused}.

$$C = \frac{\left(V^{fused} W^Q\right)\left(V^{fused} W^K + E\right)^T}{\sqrt{d_{model}}} \tag{4}$$

Then obtain a new representation of V^{fused} through incorporating relative position information according to Eq. (5).

$$V^{fused} = \left(V^{fused} + E\right)(softmax\,(C))^T + V^{fused} \tag{5}$$

Given a feature map of size $m \times n$, the directed graph model has mn vertices, and each vertex has edges that directly connect any other vertices, so the position-aware attention has to maintain $O\left(m^2 n^2\right)$ edges, which are redundant in most cases because objects are usually located sparsely in the image. Moreover, maintaining edges with space complexity $O\left(m^2 n^2\right)$ leads to parameters to be trained increasing significantly.

In order to reduce space complexity, the locations of two vertices in horizontal and vertical directions are leveraged to construct the relative positions between these two vertices. As shown in Fig. 6, the vertices are placed in a cartesian coordinate, and each vertex has an unique coordinate.

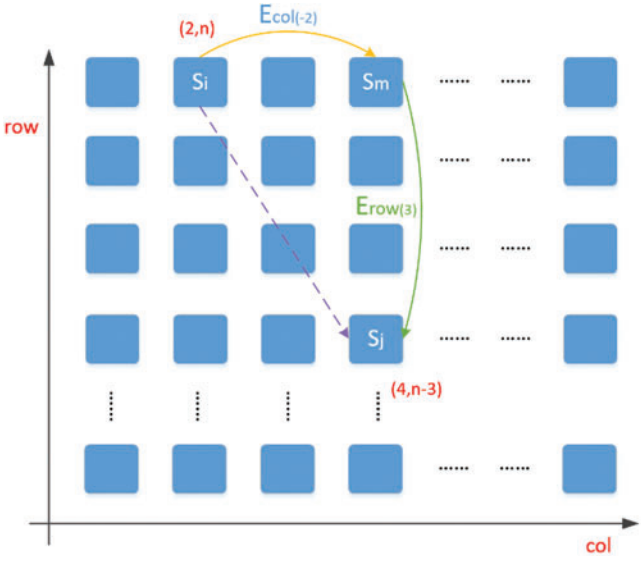

Figure 6: Using differences in horizontal and vertical directions to construct the relative positions

Algorithm 1: Calculate Edge Matrix $\mathbf{E}_{mn}$ for each element in m × n size feature map

```
Require: $m \geq 0$ & $n \geq 0$
  $E^{row} = \left[E^{row}_{-(m-1)}, \cdots, E^{row}_{0}, \cdots E^{row}_{(m-1)}\right]$, where $E^{row}_{i}$ is learning parameter.
  $E^{col} = \left[E^{col}_{-(n-1)}, \cdots, E^{col}_{0}, \cdots E^{col}_{(n-1)}\right]$, where $E^{col}_{j}$ is learning parameter.
  $row\_range = (0, \cdots, m-1), \quad col\_range = (0, \cdots, n-1)^T$
  $\mathbf{row\_matrix} = \underbrace{(row\_range, \cdots, row\_range)}_{n\times}$
  $\mathbf{col\_matrix} = \underbrace{(col\_range^T, \cdots, col\_range^T)^T}_{m\times}$
  $E_{ret} = [\quad]$
  for row in row_range do
    for col in col_range do
      $\mathbf{row\_index} = \begin{pmatrix} row & row & \cdots & row \\ row & row & \cdots & row \\ \vdots & \vdots & \ddots & \vdots \\ row & row & \cdots & row \end{pmatrix}_{m\times n}$, $\mathbf{col\_index} = \begin{pmatrix} col & col & \cdots & col \\ col & row & \cdots & col \\ \vdots & \vdots & \ddots & \vdots \\ col & col & \cdots & col \end{pmatrix}_{m\times n}$
      $\mathbf{row\_index} = \mathbf{row\_index} - \mathbf{row\_matrix}$
      $\mathbf{col\_index} = \mathbf{col\_index} - \mathbf{col\_matrix}$
      $\mathbf{E}_{mn} = E^{row}[\mathbf{row\_index} + (m-1)] + E^{col}[\mathbf{col\_index} + (n-1)]$
      $E_{ret} \quad append \quad \mathbf{E}_{mn}$
    end for
  end for
  return $E_{ret}$
```

Instead of using the edge that directly connects two vertices (the dashed line in Fig. 6), the coordinates of these two vertices are utilized to compute the edge. For example, S_i has coordinate $(2, n)$ and S_j has coordinate $(4, n-3)$, their distance (from S_i to S_j) in horizontal direction is -2, in vertical direction is 3, and their relative position (from S_i to S_j) E_{ij} is represented by $E_3^{row} + E_{-2}^{col}$. In practice, in order to get a compact computation process, we use **Algorithm 1** to get an edge matrix E for each element.

The model needs to store two kinds of edges in this way, one is $E^{row} = \left(E^{row}_{-(m-1)}, \ldots, E^{row}_{0}, \ldots, E^{row}_{(m-1)}\right)$, and the other is $E^{col} = \left(E^{col}_{-(n-1)}, \ldots, E^{col}_{0}, \ldots, E^{col}_{(n-1)}\right)$, there are $2 \cdot (m+n-1)$ edges in total. For a feature map of size $m \times n$, we reduce the space complexity of storing edges from $O\left(m^2n^2\right)$ to $O\left(max\left(m, n\right)\right)$ by using coordinates of two vertices to compute their edge.

4 Experimental Results and Analysis

4.1 Metrics

Our caption model was evaluated in several different evaluation metrics, including BLEU [20], CIDEr [21], METEOR [22], and SPICE [23], etc. These metrics focus on different aspects of generated captions and give a scalar evaluation value quantitatively. BLEU is a precision-based metric and is traditionally used in machine translation to measure the similarity between the generated captions and the ground truth captions. CIDEr measures consensus in generated captions by performing a Term Frequency-Inverse Document Frequency weighting for each n-gram. METEOR is based on the explicit word to word matches between the generated captions and the ground-truth captions. SPICE is a semantic-based method that measures how well caption models recover objects, attributes and relations shown in the ground truth captions.

4.2 Loss Functions

Given the ground truth sentence $S_{gt} = \{y_0, y_1, \ldots, y_t\}$ and its corresponding image I, the sentence S_{gt} was split into two parts $S_{target} = S_{gt}\left[0 : -1\right]$ and $S_{target\_y} = S_{gt}\left[1 : \right]$. The model was trained by minimizing the following cross-entropy loss:

$$L_{cross-entropy}\left(\theta\right) = -log\left(p_\theta\left(S_{target\_y} \mid S_{target}; \theta; I\right)\right) \tag{6}$$

where θ was the parameters of the model. At the training stage, the model was trained to generate the next ground-truth word given the previous ground-truth words, while during the testing phase, the model used the previously generated words from the model distribution to predict the next word. This mismatch resulted in error accumulation during generation at test time, because the model had never been exposed to its own predictions. To make a fair comparison with recent works [24]. At the beginning, the model was trained with standard cross-entropy loss for 15 epochs. After that, the pre-trained model continued to adjust its parameters under the proposed Reinforcement Learning (RL) method described in [24] for another 15 epochs.

This method can relieve the mismatch between training and testing by minimizing the negative expected reward:

$$L\left(\theta\right) = -E_{\omega^s \sim p\theta}\left[r\left(\omega^s\right)\right] \tag{7}$$

where $\omega^s = \left(\omega^s_1, \ldots, \omega^s_T\right)$ was the generated sentence and r was the CIDEr score of the generated sentence.

4.3 Dataset

The MSCOCO2014 dataset [25], one of the most popular datasets for image captioning, was used to evaluate the proposed model. This dataset contains 123,287 images in total (82783 training images and 40504 validation images respectively), each image has five different captions. To compare our experimental results with other methods precisely, the widely used "Karpathy" split [26] was adopted for MSCOCO2014 dataset. This split has 112,387 images for training, 5000 images for validation and 5000 images for testing. The performance of the model was measured on the testing set.

4.4 Data Preprocessing

The images were normalized to mean = [0.485, 0.456, 0.406], std = [0.229, 0.224, 0.225], and the captions with length larger than 16 got clipped. Subsequently, a vocabulary was built with three tokens <BOS>, <EOS>, <UNK> and the words that occurred at least 5 times in the preprocessed captions. The token <UNK> represented words appearing less than 5 times, the token <BOS> and <EOS> indicated the start and the end of a sentence. Finally, the captions were vectorized by the indices of words and tokens in the vocabulary. During the training process, for the convenience of transformation between words and indices, two maps *wtoi* and *itow* were maintained. *wtoi* maps a word or token to its corresponding index, and *itow* maps an index to the word or token.

4.5 Inference

The inference was similar to RNN-based models, and the word would be generated one by one at a time. Firstly, the model began with the sequence S_0 that only contained the start token <BOS>, and obtained the dictionary probability $y_i \sim p\,(y_i \mid S_0; \theta; I)$ through the first iteration. Afterwards, some sampling methods such as the greedy method or the beam search method were used to generate the first word y_1. Then, y_1 was fed back into the model to generate the next word y_2. This process would continue until the end token <EOS> or the max length L was reached.

4.6 Implementation Details

A FPN from a pretrained instance segmentation model [27] was used to produce features at five levels. Experiments were carried out based on the second and the fourth features. The spatial size of the second feature was set to 14 × 14 and the other was set to 28 × 28 via adaptive average pooling. We did not train the fine-tune model, thus, the parameters of the two features were fixed in the whole training process.

In Tab. 1, the hyperparameter settings of the position-aware transformer model trained with standard cross-entropy loss are presented.

For our model trained with standard cross-entropy loss, we used 6 attention layers, $d_{model} = 256$, 4 attention heads, $d_{head} = 64$, 1024 feed forward inner-layer dimensions, and $P_{dropout} = 0.1$. This model was trained for 15 epochs, each epoch had 12000 iterations and the batch size was 10. The initial learning rate of the model was 5×10^{-4}, the warmup strategy with $warmup_{steps} = 20000$ was used to speed up the training and the same weight decay strategy as in [16] was adopted for learning rate adjustment. The Adam optimizer [28] with $\beta_1 = 0.9$, $\beta_2 = 0.98$, and $\epsilon = 10^{-9}$ was used to update parameters of our model. During training, we employed label smoothing of value $label_{smoothing} = 0.1$ [29]. At the inference stage, the beam search method with a beam size of 3 was chosen for better caption generation. The Pytorch framework was adopted to implement our model for image captioning.

For our model optimized by CIDEr optimization (Initializing from the pretrained cross-entropy trained model), it was trained for another 15 epochs to adjust parameters. The initial learning rate was set to 1×10^{-5}, and both warmup and weight decay options were turned off. The rest of the settings were identical to the cross-entropy model.

Table 1: Hyperparameter settings of the model

Parameter	Value
epochs	15
learning_rate	0.0005
label_smoothing	0.1
warmup_steps	20000
$adam\beta_1$	0.9
$adam\beta_2$	0.98
sample_method	*beam_search*
beam_size	3
d_{model}	256
num_head	4
d_{head}	64
$P_{dropout}$	0.1

4.7 Ablation Studies

In this section, we conducted several ablative experiments for the position-aware transformer model on the MSCOCO datasets. In order to further verify the effectiveness of the sub-modules in our model, a Vanilla Transformer model for image captioning was implemented. It regarded the CNN and the transformer encoder as the image encoder and the transformer decoder as the caption decoder. Based on the vanilla transformer model, the other two models (FPN Transformer and Position-aware Transformer) were implemented as follows:

FPN Transformer: a model equipped with the image-feature attention sub-module and employed image features built by the FPN.

Position-aware Transformer: a model equipped with the image-feature attention and position-aware attention sub-modules. This model also used the image features built by the FPN.

In the experiments, Vanilla Transformer model used the ResNet to encode the given image I to the spatial image feature and the image feature was obtained from the 5th pool layer of the ResNet. The ResNet was pre-trained on the ImageNet dataset. We then apply adaptive average pooling to obtain an image spatial feature $V = \{v_1, \ldots, v_{14x14}\}$, $v_i \in R^{d_{model}}$, where 14 × 14 is the number of regions, and v_i represents a region of the image. FPN Transformer used the same FPN network as in [27] to encode the given image I and the image feature attention to fuse image features built by the FPN to size of 14 × 14 too. Position-aware Transformer was the proposed approach described in Fig. 2. All hyperparameters of the three models stayed the same if possible. In Tab. 2, the test results of the Vanilla Transformer, FPN Transformer and Position-Aware Transformer on BLEU-1, BLEU-2, BLEU-3, BLEU-4, METEOR, ROUGE-L, CIDEr metrics are presented, and the validation results of the three models are shown in Fig. 7.

As shown in Tab. 2, through image-feature attention and position-aware attention, the Vanilla Transformer model can achieve better performance in terms of BLEU-1, BLEU-2, BLEU-3, BLEU-4, METEOR, ROUGE-L and CIDEr.

Table 2: The performance of our models optimized by standard cross-entropy loss

Metric	BLEU-1	BLEU-2	BLEU-3	BLEU-4	METEOR	ROUGE-L	CIDEr
Vanilla transformer	74.9	58.2	44.6	34.1	27.1	55.4	107.6
FPN transformer	76.2	60.0	46.3	35.6	27.6	56.7	113.9
Position-aware transformer	76.7	60.8	46.9	36.0	27.8	56.5	114.9

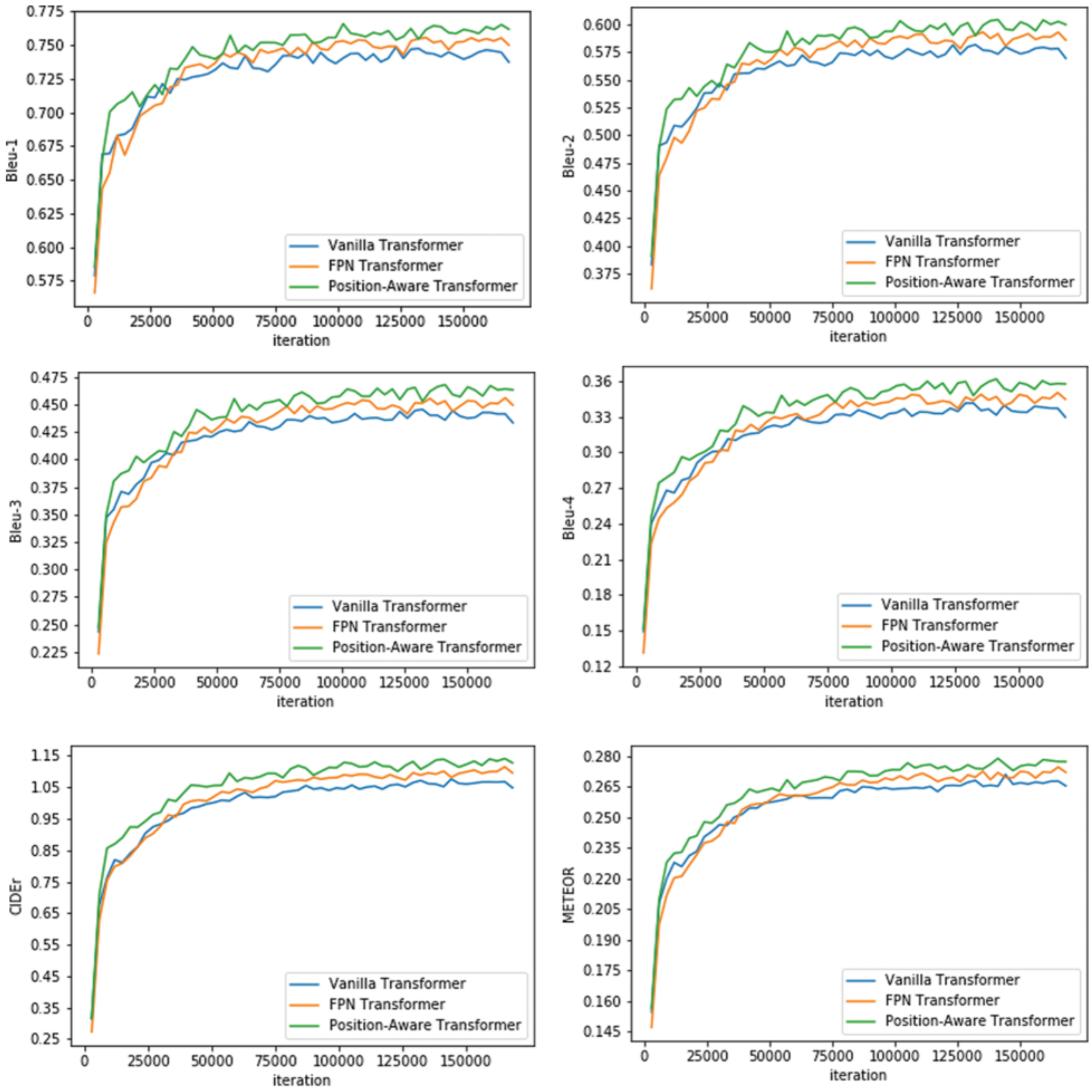

Figure 7: Validation results of several metrics

From Fig. 7, it turns out that FPN Transformer has better performance compared with Vanilla Transformer on all metrics, which is due to the fact that the FPN produces a multi-scale feature representation in which all levels are semantically strong, including the high-resolution levels. This enables a model to detect objects across a large range of scales by scanning the model over both positions and pyramid levels. Also, it can be noticed that the combination of image-feature attention and position-aware attention provides the best performance, mainly because that the position-aware attention makes features can be explained from the perspective of spatial relationship.

SPICE is a semantic-based method that measures how well caption models recover objects, attributes and relations. To investigate the performance improved by the proposed sub-modules, we report SPICE F-scores over various subcategories on the MSCOCO testing set in Tab. 3 and Fig. 8. When equipped with the image-feature attention, the FPN Transformer increases the SPICE-Objects metric by 2.2% compared with the Vanilla Transformer, exceeding the relative improvement of 1.85% on the SPICE-Relations metric and the relative improvement of 0.15% on the SPICE metric. It shows that the image-feature attention can improve the performance in terms of identifying objects. After incorporating the position-aware attention, the Position-aware Transformer shows more remarkable relative improvement of 9.0% on the SPICE-Relations metric than the relative improvements on the SPICE and the SPICE-Objects metrics, demonstrating that the position-aware attention improves the performance by identifying the relationships between objects.

Table 3: SPICE F-scores over various subcategories on the MSCOCO test set

Metric	SPICE	SPICE-objects	SPICE-relations
Vanilla transformer	20.1	36.2	5.4
FPN transformer	20.1	37.0	5.5
Position-aware transformer	20.9	37.8	6.0

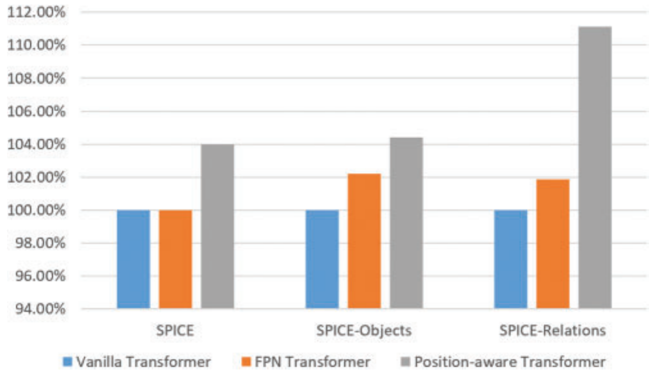

Figure 8: Performance comparison of different transformers

4.8 *Comparing with Other State-of-the-Art Methods*

The experimental results of the Position-aware Transformer and previous state-of-the-art models on the MSCOCO testing set are shown in Tab. 4. All results are produced by models trained with standard cross-entropy loss. The Soft-Attention model [12], which uses the ResNet-101 as the image encoder, is our baseline model.

Table 4: Experimental results of our approach compared with other methods (optimized by standard cross-entropy loss)

Metric	BLEU-1	BLEU-2	BLEU-3	BLEU-4	METEOR	ROUGE-L	CIDEr
Soft-attention [12]	70.7	49.2	34.4	24.3	23.9	–	–
Hard-attention [12]	71.8	50.4	35.7	25.0	23.0	–	–
Adaptive [15]	74.2	58.0	43.9	33.2	26.6	–	108.5
Bottom-up [14]	77.2	–	–	36.2	27.0	56.4	113.5
Position-aware transformer	76.7	60.8	46.9	36.0	27.8	56.5	114.9
Relative improvement	−0.07%	–	–	0.06%	3%	0.1%	1.2%

In contrast to recent state-of-the-art models, our model shows a better performance. When compared with the Bottom-Up model, the METEOR score, ROUGE-L score and CIDEr score increase from 27.0 to 27.8, 56.4 to 56.5, 113.5 to 114.9 respectively, the BLEU-1 score and BLEU-4 score obtain similar results. Among these metrics, METEOR, ROUGE-L and CIDEr are specialized for image captioning tasks, which validates the effectiveness of our model.

The experimental results of the Position-aware Transformer and Bottom-up model that trained with CIDEr optimization on the MSCOCO testing set are shown in Tab. 5.

As shown in Tab. 5, our model improves the BLEU4 score from 36.3 to 38.4, METEOR score from 27.7 to 28.3, ROUGE-L score from 56.9 to 58.4 and CIDEr score from 120.1 to 125.5 respectively. In addition, we can also see that all the metrics increase, specifically, the CIDEr metric gets 4.5% relative improvement. This shows that the proposed approach has better performance.

Table 5: Experimental results of our approach compared with the bottom-up (optimized by CIDEr optimization)

Metric	BLEU-1	BLEU-2	BLEU-3	BLEU-4	METEOR	ROUGE-L	CIDEr
Bottom-up [14]	79.8	–	–	36.3	27.7	56.9	120.1
Position-aware transformer	79.8	64.7	50.2	38.4	28.3	58.4	125.5
Relative improvement	0%	–	–	5.8%	2.1%	2.6%	4.5%

5 Conclusion and Future Work

A position-aware transformer with two attention mechanisms, i.e., the position-aware attention and image-feature attention, is proposed in this work. To generate more accurate and more fluent captions, the position-aware attention enables the model to make use of relative positions between image features. These relative positions are modeled as the directed edges in a directed graph in which vertices represent the elements of image features. In addition, to make the model be able to detect objects of different scales in the image without increasing the number of parameters, the image-feature attention brings multi-level features through the FPN and uses the scaled-dot-product to fuse multi-level features. With these innovations, we obtained a better performance than some state-of-the-art approaches on the MSCOCO benchmark.

At a high level, our work utilizes multi-level features and position information to increase performance. While this suggests several directions for future research: (1) The image-feature attention pick up features of particular levels for fusion. However, in some cases, determining these features depends on the specific image. For some images, all the objects may be large objects, so the fusion of low-level features may bring inevitable noises to the prediction process of the model due to the weak semantics of low-level features; (2) The position-aware attention uses the relative positions between features to infer the words with abstract concepts in descriptions, but not all such words are related to spatial relationships. Based on these issues, further research will be carried out subsequently, and we will apply this approach to the image retrieval based on text information.

Acknowledgement: The authors extend their appreciation to the Deanship of Scientific Research at King Saud University, Riyadh, Saudi Arabia for funding this work through research Group No. RG-1438-070. This work is supported by NSFC (61977018). This work is supported by Research Foundation of Education Bureau of Hunan Province of China (16B006).

Funding Statement: This work was supported in part by the National Natural Science Foundation of China under Grant No. 61977018, the Deanship of Scientific Research at King Saud University, Riyadh, Saudi Arabia for funding this work through research Group No. RG-1438-070 and in part by the Research Foundation of Education Bureau of Hunan Province of China under Grant 16B006.

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] R. Socher and F. Li, “Connecting modalities: Semi-supervised segmentation and annotation of images using unaligned text corpora,” in *Proc. the IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, San Francisco, CA, USA, pp. 966–973, 2010.

[3] B. Z. Yao, X. Yang, L. Lin, M. W. Lee and S. C. Zhu, “I2t: Image parsing to text description,” *Proceedings of the IEEE*, vol. 98, no. 8, pp. 1485–1508, 2010.

[4] A. Krizhevsky, I. Sutskever and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” *Proc. Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017.

[5] R. Chen, L. Pan, C. Li, Y. Zhou, A. Chen *et al.,* “An improved deep fusion CNN for image recognition,” *Computers, Materials & Continua*, vol. 65, no. 2, pp. 1691–1706, 2020.

[6] S. Lee, Y. Ahn and H. Y. Kim, “Predicting concrete compressive strength using deep convolutional neural network based on image characteristics,” *Computers, Materials & Continua*, vol. 65, no. 1, pp. 1–17, 2020.

[7] Z. Li, C. Chi and Y. Zhan, “Corpus augmentation for improving neural machine translation,” *Computers, Materials & Continua*, vol. 64, no. 1, pp. 637–650, 2020.

[8] J. Qiu, Y. Liu, Y. Chai, Y. Si, S. Su *et al.,* "Dependency-based local attention approach to neural machine translation," *Computers, Materials & Continua*, vol. 59, no. 2, pp. 547–562, 2019.

[9] O. Vinyals, A. Toshev, S. Bengio and D. Erhan, "Show and tell: Lessons learned from the 2015 mscoco image captioning challenge," *Proc. IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 4, pp. 652–663, 2016.

[10] J. Lu, J. Yang, D. Batra and D. Parikh, "Neural baby talk," in *Proc. the IEEE Conf. on Computer Vision and Pattern Recognition*, Salt Lake City, UT, USA, pp. 7219–7228, 2018.

[11] K. Cho, M. B. Van, C. Gulcehre, F. Bougares, H. Schwenk *et al.,* "Learning phrase representations using RNN encoder-decoder for statistical machine translation," in *Proc. the Conf. on Empirical Methods in Natural Language Processing*, Doha, Qatar, pp. 1724–1734, 2014.

[12] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville *et al.,* "Courville etal, Show, attend and tell: Neural image caption generation with visual attention," in *Proc. Int. Conf. on Machine Learning*, Miami, Florida, USA, pp. 2048–2057, 2015.

[13] L. Chen, H. Zhang, J. Xiao, L. Nie, J. Shao *et al.,* "SCA-CNN: Spatial and channel-wise attention in convolutional networks for image captioning," in *Proc. the IEEE Conf. on Computer Vision and Pattern Recognition*, Honolulu, HI, USA, pp. 5659–5667, 2017.

[14] 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.

[15] 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.

[16] 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*, Vancouver, Washington, USA, pp. 5998–6008, 2017.

[17] 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/CVF Conf. on Computer Vision and Pattern Recognition*, Washington, USA, pp. 10327–10336, 2020.

[18] G. Li, L. Zhu, P. Liu and Y. Yang, "Entangled transformer for image captioning," in *Proc. the IEEE/CVF Int. Conf. on Computer Vision*, Beach, CA, USA, pp. 8928–8937, 2019.

[19] T. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan *et al.,* "Feature pyramid networks for object detection," in *Proc. the IEEE Conf. on Computer Vision and Pattern Recognition*, Honolulu, HI, USA, pp. 2117–2125, 2017.

[20] K. Papineni, S. Roukos, T. Ward and W. Zhu, "Bleu: A method for automatic evaluation of machine translation," in *Proc. the 40th Annual Meeting of the Association for Computational Linguistics*, Morristown, NJ, USA, pp. 311–318, 2002.

[21] 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.

[22] M. Denkowski and A. Lavie, "Meteor universal: Language specific translation evaluation for any target language," in *Proc. the Ninth Workshop on Statistical Machine Translation*, Baltimore, Maryland, USA, pp. 376–380, 2014.

[23] 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.

[24] 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.

[25] 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.

[26] A. Karpathy and F. Li, "Deep visual-semantic alignments for generating image descriptions," in *Proc. the IEEE Conf. on Computer Vision and Pattern Recognition*, New York, NY, USA, pp. 3128–3137, 2015.

[27] X. Wang, T. Kong, C. Shen, Y. Jiang and L. Li, “Solo: Segmenting objects by locations,” in *Proc. European Conf. on Computer Vision*, Glasgow, UK, pp. 649–665, 2020.

[28] H. Zhong, Z. Chen, C. Qin, Z. Huang, V. W. Zheng *et al.,* “Adam revisited: A weighted past gradients perspective,” *Frontiers of Computer Science*, vol. 14, no. 5, pp. 1–16, 2020.

[29] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens and Z. Wojna, “Rethinking the inception architecture for computer vision,” in *Proc. the IEEE Conf. on Computer Vision and Pattern Recognition*, Las Vegas, NV, USA, pp. 2818–2826, 2016.