Computers, Materials & Continua

Tech Science Press

DOI: 10.32604/cmc.2024.047822

ARTICLE

Enhancing Image Description Generation through Deep Reinforcement Learning: Fusing Multiple Visual Features and Reward Mechanisms

Yan Li, Qiyuan Wang* and Kaidi Jia

School of Cyber Security, Gansu University of Political Science and Law, Lanzhou, 730070, China

*Corresponding Author: Qiyuan Wang. Email: wangqiyuan@stu.gsupl.edu.cn

Received: 19 November 2023 Accepted: 11 January 2024 Published: 27 February 2024

ABSTRACT

Image description task is the intersection of computer vision and natural language processing, and it has important prospects, including helping computers understand images and obtaining information for the visually impaired. This study presents an innovative approach employing deep reinforcement learning to enhance the accuracy of natural language descriptions of images. Our method focuses on refining the reward function in deep reinforcement learning, facilitating the generation of precise descriptions by aligning visual and textual features more closely. Our approach comprises three key architectures. Firstly, it utilizes Residual Network 101 (ResNet-101) and Faster Region-based Convolutional Neural Network (Faster R-CNN) to extract average and local image features, respectively, followed by the implementation of a dual attention mechanism for intricate feature fusion. Secondly, the Transformer model is engaged to derive contextual semantic features from textual data. Finally, the generation of descriptive text is executed through a two-layer long short-term memory network (LSTM), directed by the value and reward functions. Compared with the image description method that relies on deep learning, the score of Bilingual Evaluation Understudy (BLEU-1) is 0.762, which is 1.6% higher, and the score of BLEU-4 is 0.299. Consensus-based Image Description Evaluation (CIDEr) scored 0.998, Recall-Oriented Understudy for Gisting Evaluation (ROUGE) scored 0.552, the latter improved by 0.36%. These results not only attest to the viability of our approach but also highlight its superiority in the realm of image description. Future research can explore the integration of our method with other artificial intelligence (AI) domains, such as emotional AI, to create more nuanced and context-aware systems.

KEYWORDS

Image description; deep reinforcement learning; attention mechanism

1 Introduction and Related Work

Image description generation has emerged as an important research area within the domains of computer vision and natural language processing, attracting extensive interest in recent years. In the realm of education, this technology assists in creating more interactive and inclusive learning materials, especially for young learners,non-native speakers and visually impaired person, by providing contextual visual explanations. In the rapidly evolving domain of artificial intelligence, image description tasks contribute significantly to the development of more advanced and intuitive

artificial intelligence (AI) systems, improving machine learning models' ability to interpret and describe complex visual data.

Based on the underlying technology, methods for generating image descriptions can be divided into three major categories: template-based, retrieval-based, and deep learning-based approaches [1].

Template-based image description methods utilize pre-defined templates designed by linguistic or domain experts that include fixed grammatical structures and vocabulary. Farhadi et al. [2,3] created a template for the image description task. When creating the description, they selected the appropriate template according to the features and content of the image, and filled the template with relevant words to form the description statement. The advantage of this approach is the clarity and grammatical correctness of the resulting sentences. However, due to the restrictions of the templates, these generated descriptions often lack individuality and diversity.

Meng et al. [4] used a search-based image description approach to create descriptions by exploiting the association relationship between images and text, retrieving the most similar existing descriptions of input images from a dataset. Commonly, this method begins with extracting features from the image using computer vision techniques, such as Convolutional Neural Networks (CNNs), to obtain image feature vectors [5]. These features are then matched with features from descriptions in a database, selecting the most fitting descriptions based on calculated similarities [6,7]. The strength of this method is that it can produce descriptions that are congruent with the image content and it demands extensive datasets [8]. However, the limitations of the description datasets may not cover all possible image descriptions, and the accuracy of the feature extraction process can be a limiting factor.

Deep learning-based image description often rely heavily on convolutional neural networks (CNNs) and recurrent neural networks (RNNs) for feature extraction and text generation, respectively. Vinyals et al. [9] have shown promising results using these methods, but they often struggle to accurately capture the subtle interactions between visual elements and their contextual language descriptions. Common issues include the generation of generic, repetitive, or contextually inaccurate captions, stemming from the inability of these models to fully grasp the complexity of real-world images and their diverse elements. Xu et al. [10] tried to capture the subtleties of cross-modal interactions in order to better generate descriptive statements. Moreover, existing methods often treat image description as a straightforward translation task, overlooking the subtleties of cross-modal interactions [11]. This simplification leads to a gap between the semantic richness of images and the descriptive power of the generated text [12]. The challenge is further compounded by the static nature of the training process in traditional supervised learning models, which limits their ability to adapt to new or varied data inputs dynamically [13]. Notably, recent research by Lu et al. [14,15] has introduced enhanced attention mechanisms and language models to better capture the subtleties of cross-modal interactions.

To sum up, in the field of image description generation, the intersection of computer vision and natural language processing presents unique challenges. Current methodologies, while having made significant strides, show clear limitations. Template-based methods suffer from a lack of diversity due to fixed structures. Retrieval-based methods are constrained by the limitations of their reference databases, often failing to generate novel or contextually apt captions. Deep learning approaches, though advanced, can struggle with the subtlety and accuracy of contextual interpretation, leading to generic or repetitive descriptions [16]. Our research is driven by the motivation to surmount these challenges. We aim to enhance the accuracy and contextual relevance of image captioning, offering a more nuanced understanding of the intricate relationship between visual data and its linguistic interpretation. This paper introduces a novel deep reinforcement learning approach to achieve this goal, marking a significant advancement in automated image captioning.

In view of the challenges discussed and the limitations of existing methods, this paper makes the following contributions:

1. New methods for extraction and alignment of visual and textual features are proposed. This method approach integrates state-of-the-art neural networks including ResNet-101, Faster Region-based Convolutional Neural Network (Faster R-CNN), Transformer and long short-term memory network (LSTM) to efficiently extract visual features of images and semantic features of text and perform semantic alignment.

2. Improvements are made to the reward function for reinforcement learning. This method uses the similarity between image and text as a criterion to penalize or reward to guide description generation.

Comprehensive Validation on Standard Dataset: The effectiveness of our approach is rigorously validated on the COCO dataset, demonstrating significant improvements in key performance metrics over traditional methods.

The rest of this article is organized as follows: Section 2 describes the key techniques we used. Section 3 describes the relevant settings of this experiment, including the overall model design, the data set used, the training process and the evaluation indicators. In section 4, our experimental results are presented and compared with the existing methods. Section 5 discusses the implications of our findings, potential applications, and future directions for research in this field.

2 Key Technology

2.1 Residual Networks

The convolutional neural network model proposed by the Microsoft Research team is a deep residual network (ResNet-101) known for its powerful performance and esoteric design, and has gained widespread attention and applications. This network is particularly suited for vision tasks; it is composed of multiple convolutional layers and pooling layers, capable of learning multi-level representations of an image. The convolutional layers employ kernels of various sizes to extract features from images, progressively capturing the semantic information from basic features such as edges and textures to more advanced features like shapes and parts of objects [17]. Furthermore, the network incorporates residual connections, which allow the skipping of certain layers to directly transmit information, effectively addressing the issues of vanishing gradients and model degradation during the training of deep networks. Fig. 1 illustrates a residual block, demonstrating how the structure of ResNet-101 can better capture details within an image while maintaining awareness of the overall semantic information.

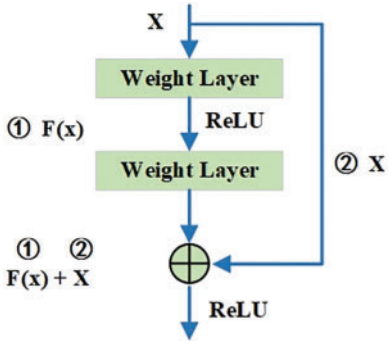


Figure 1: Schematic of the residual block

Fig. 1 illustrates a residual block, which consists of two parts: the main path and the residual connection. The main path includes two weight layers, which can be convolutional layers, typically followed by an activation function such as Rectified Linear Unit (ReLU). Within this structure, the input X first passes through the first weight layer, then through the ReLU activation function, before moving onto the second weight layer [18]. The output, F(x), represents the transformation of the input X as it passes through the main path of the network. The residual connection allows the input X to bypass these two weight layers and directly reach the end of the main path, effectively completing the calculation of F(x) + X, thus resolving issues of gradient disappearance or explosion during training.

2.2 Attention Mechanism

An "attention mechanism" in the context of neural networks is a computational framework that allows a model to dynamically focus on certain parts of its input as it processes data, much like how humans pay attention to particular aspects of what they see or hear. In deep learning, especially in models dealing with sequential data like natural language or time series, attention mechanisms can selectively weight the importance of different input features at different times, improving the model's ability to make predictions or generate outputs.

The core idea of the attention mechanism is to assign different weights to different input sequences, allowing the model to focus on the most relevant or important parts. The computation of these weights is typically based on the relationships among the inputs or the probabilistic distribution of attention. The attention mechanism consists of three key components: the query (Q), the key (K), and the value (V), as illustrated in Fig. 2. Essentially, the more important the information of the key is, the larger the corresponding value becomes. Specifically, each key is assessed in relation to the query using a compatibility function, yielding raw scores that are normalized through a softmax operation to form a probability distribution that signifies the relative importance of each key. These normalized scores are then multiplied with their associated values to allocate attention proportionally, with the resultant products summed to form the final attention value, encapsulating information most relevant to the query within the given context. This mechanism enables the model to dynamically focus and assign varying levels of importance to different parts of the input data.

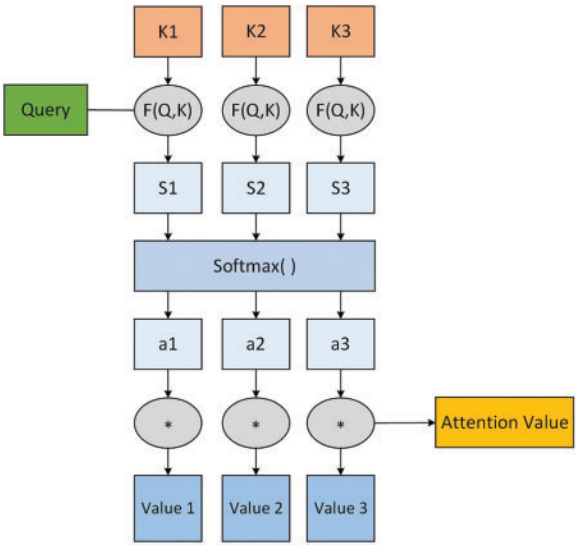


Figure 2: Schematic of the attention mechanism

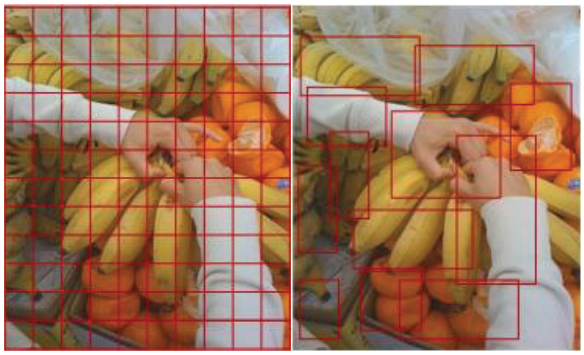
Fig. 2 presents the schematic of the attention mechanism. The computational process of the attention mechanism primarily includes:

- The process maps the input data to the query (Q), key (K), and value (V) spaces through linear transformations, and computing the similarity scores between the query (Q) and key (K), usually using the dot product;
- Applying a softmax function to the similarity scores to obtain attention weights;
- Taking a weighted average of the value (V) with the attention weights and performing a linear transformation to produce the final attention output. The overall calculation is expressed by the formula (1):

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

where Attention( ) is a function that computes the attention mechanism; Q represents the query vector; K and V represent the key-value pairs, with K^T indicating the transpose of the key vector; Softmax( ) is a normalization function; and d_k is the dimension of the key.

Anderson et al. [19] classified the attention mechanisms in computer vision into two types: top-down attention and bottom-up attention. In top-down attention, feature maps are extracted from the input image through operations like convolution and assesses weights for each element in the feature map [20], effectively performing a micro-weight analysis of various areas of the input image, as shown in Fig. 3a. Bottom-up attention utilizes object detection methods like Faster R-CNN [21] to extract target regions from the input image and assign weights to each region for the query vector, akin to the effect of hard attention, as depicted in Fig. 3b. Hence, this paper employs both top-down and bottom-up attention mechanisms to extract the average and local features of the image, respectively, and then uses attention weight calculations for the weighted fusion of features.



(a) top-down attention (b) bottom-up attention

Figure 3: Schematic of top-down and bottom-up attention

Fig. 3 serves to visually demonstrate the distinctions between top-down and bottom-up attentional processes in the context of a fruit selection task. In the top-down attention diagram, the grid overlay suggests a deliberate, strategic search pattern, likely guided by the subject's knowledge or expectation of finding certain types of fruit, such as bananas. This pattern reflects a cognitive bias towards areas of the scene where the target is thought to be, with attention systematically directed to grid segments according to the individual's task or prior knowledge.

In the bottom-up attention diagram, the irregularity and size variation in the grid boxes imply a reactive attentional shift driven by the perceptual features of the items, such as the bright color of an orange amidst the bananas. Attention is drawn involuntarily to these salient features, which stand out against the uniform background of the other fruits, illustrating how unexpected or conspicuous stimuli in the environment can dominate our sensory input and, thereby, command our attentional resources.

Thus, superimpositions on images are not random; they symbolize the distribution and focus of human attention, whether it is top-down, goal-directed attention to the intended location of a target, or bottom-up, unconscious attraction to visually salient features in a scene. This diagram can be used to elucidate the complex interplay between cognitive intentions and sensory stimuli that shape our attentional focus.

2.3 Reinforcement Learning

Reinforcement learning is a learning method that operates under the framework of Markov decision processes. Its core concept involves an agent interacting continuously with the environment, collecting information through these interactions, and optimizing decision strategies with the aim of obtaining the maximum cumulative reward. In other words, the agent attempts various actions and adjusts its behavior based on the rewards received from the environment. Currently, the main approaches to reinforcement learning can be divided into two categories: policy-based algorithms, which focus on directly optimizing action strategies to ensure the best action is chosen given a state; and value-based algorithms, which, unlike policy-based algorithms, primarily concentrate on evaluating the expected return of each action, thus establishing a mapping relationship between actions and their corresponding values.

To combine the advantages of both policy-based and value-based methods, this paper employs the Advantage Actor-Critic (A2C) algorithm [22] for training. The A2C algorithm accelerates the reinforcement learning process by leveraging a dual-structured approach that enhances policy performance and learning stability, often resulting in expedited convergence to optimal behaviors. A2C's efficiency can be quantified by its ability to converge to a desired policy quality with fewer environmental interactions a measure of sample efficiency compared to traditional methods. This algorithm is composed of two parts: the policy function (Actor), which directly learns the strategy of executing actions given a state; and the value function (Critic), which estimates the value of a state or state-action pair under the current policy. The A2C algorithm accelerates the learning process by using the value function to reduce the variance in the policy gradient estimation. In this way, the policy function and value function work in tandem, continuously adjusting and refining the strategy, enabling the model to learn more effectively. Fig. 4 illustrates the schematic diagram of the A2C algorithm.

As shown in Fig. 4, the Policy Network (Actor) decides on the action to take given the current state of the environment. The Value Network (Critic) estimates the value of the current policy by calculating the expected return (value q) of the current state-action pair. This is done by evaluating the rewards received from the environment after the actor takes an action. By utilizing the value function, the A2C algorithm reduces the variance in policy gradient estimation, which in turn accelerates the learning process. The interaction between the policy and value functions allows for continuous adjustment and refinement of strategies. This collaborative mechanism between the Actor and Critic leads to more effective learning and expedited convergence to optimal behaviors, enhancing the model's ability to generate accurate and contextually appropriate image descriptions.

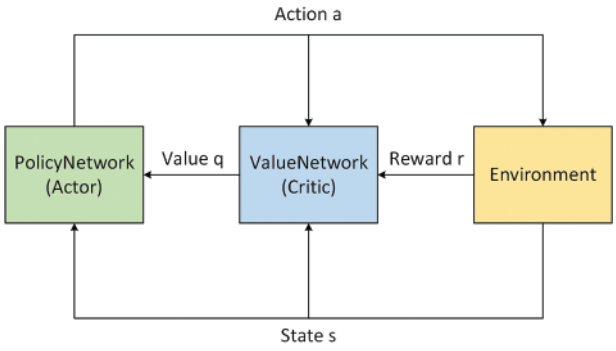


Figure 4: Schematic diagram of the A2C algorithm

Upon receiving the state information (state s) from the environment, the actor chooses an action (action a), which influences the environment. The environment then provides a new state and a reward (reward r) based on the action taken. The critic uses this reward to update the value function, which is used to evaluate the chosen action and guide the actor towards more rewarding actions in the future. The policy network is updated based on both the critic's evaluation and the received reward, aiming to improve the action selection policy over time.

This continuous loop of interaction between the actor and critic, with feedback from the environment, facilitates the learning process. The actor learns to select better actions, while the critic becomes better at evaluating the potential of different states and actions. This cooperative dynamic enables the model to effectively learn optimal strategies for decision-making tasks.

The application of the A2C algorithm in the task of image description generation within this paper necessitates the establishment of three core components: the policy function (Actor), the value function (Critic), and the advantage function (Advantage).

(1) Policy Function (Actor)

The policy function, commonly denoted as $\pi_\theta(a|s)$, is a parametric function, where θ represents the network parameters, a denotes actions, and s denotes the environmental states. The goal of the Actor is to generate a policy that defines the probability distribution for taking each possible action a in a given state s. This is achieved by maximizing the expected return a $J(\theta)$, where $J(\theta)$ is the performance measure of the policy π_θ, as defined in Eq. (2):

$$J(\theta) = E_{\pi_\theta}[R_t] \tag{2}$$

where R_t represents the cumulative reward starting from time step t.

(2) Value Function (Critic)

The value function, denoted as $V_\phi(s_t)$, where ϕ represents the network parameters, is used to estimate the value of a state s or a state-action pair (s, a) under the current policy π. The Critic's purpose is to reduce the variance of the reward estimates, thus aiding the Actor in making better decisions. The update of the value function is typically achieved through Temporal Difference (TD) learning, as shown in Eq. (3):

$$\delta_t = r_t + \gamma V_\phi(s_{t+1}) - V_\phi(s_t) \tag{3}$$

where δ_t is the TD error, r_t is the immediate reward, and γ is the discount factor used to calculate the present value of future rewards.

(3) Advantage Function (Advantage)

The advantage function is used to measure the difference between the current action and the average expected action, helping to determine whether an action has a relative advantage under the current state. In the A2C algorithm, the advantage function A(s,a) replaces the original return to guide policy updates. The advantage function assesses the value of executing a specific action a relative to the average expected action, which can be calculated using the value estimates provided by the Critic, as shown in Eq. (4):

$$A(s,a) = Q_{\phi}(s,a) - V_{\phi}(s) \tag{4}$$

where $Q_{\phi}(s,a)$ is the action-value function, representing the expected return of taking action a in state s.

As we have explored the key technologies and foundational concepts underlying our approach, including the attention mechanism and reinforcement learning, it becomes evident how these components synergize to enhance image description generation. This leads us to the core of our study-the model architecture. In the next section, we delve into the intricate design of our model, revealing how it integrates these technologies to optimize performance and achieve results in image description.

3 Model Architecture

In the realm of reinforcement learning, the primary objective is to enable the agent to identify the optimal behavioral strategy amidst complex interactions with the environment. To achieve this, the foundational behavioral patterns are modeled as policy functions. The policy function serves as the blueprint for the agent's decision-making, dictating the strategies the agent adopts in various environmental scenarios and guiding its actions at any given moment.

The reward function plays the role of a "compass" throughout this process. Each time the agent performs an action within the environment, the reward function evaluates the outcome of that action. Depending on the degree of alignment between the agent's actions and the desired outcomes, the reward function provides corresponding positive or negative feedback. This immediate feedback mechanism ensures that the agent can clearly discern which behaviors are advantageous and which may be harmful or ineffective.

The value function acts like a far-sighted observer, concerned not just with the agent's short-term actions but more with the long-term overall benefits. Specifically, the value function assesses not just the agent's single actions but the expected return of all possible actions in a given state. This provides the agent with a future perspective, helping it to anticipate long-term outcomes and choose strategies that are likely to yield long-term benefits.

Fig. 5 displays the model structure for reinforcement learning. From the diagram, it is evident that the policy function, reward function, and value function work in close coordination, collectively guiding the agent towards the optimal strategy throughout the learning and decision-making process.

The diagram presents a reinforcement learning framework for generating descriptions from images, where an encoder first processes the visual input into a feature-rich representation. This is followed by an attention mechanism that selectively focuses on different image segments, guiding the policy network embedded within the decoder to generate contextually relevant text. Simultaneously, a reward network evaluates the descriptive output, providing feedback that, along with the anticipated values from the value network, is used to iteratively refine the policy decisions. The value network's predictions are concatenated with the initial image features, creating a feedback loop that enhances the

attention mechanism's ability to focus on pertinent image parts across successive iterations, thereby optimizing the descriptive accuracy and relevance of the generated captions.

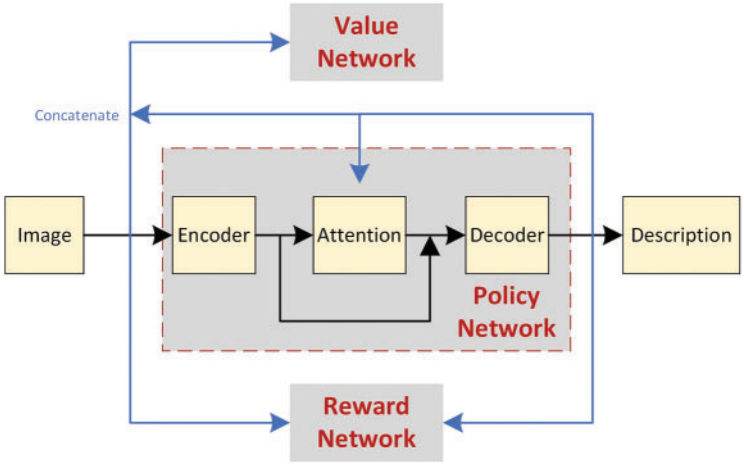


Figure 5: Schematic diagram of reinforcement learning model

3.1 Policy

The policy function in this paper employs an encoder-decoder architecture. The encoder's primary role is to extract feature representations from input images, capturing the main information within the images and providing the necessary input for the subsequent decoding process. The decoder, as the second component of the model, is tasked with accepting the image features provided by the encoder and generating descriptive sentences based on these features. The basic model structure is illustrated in Fig. 6.

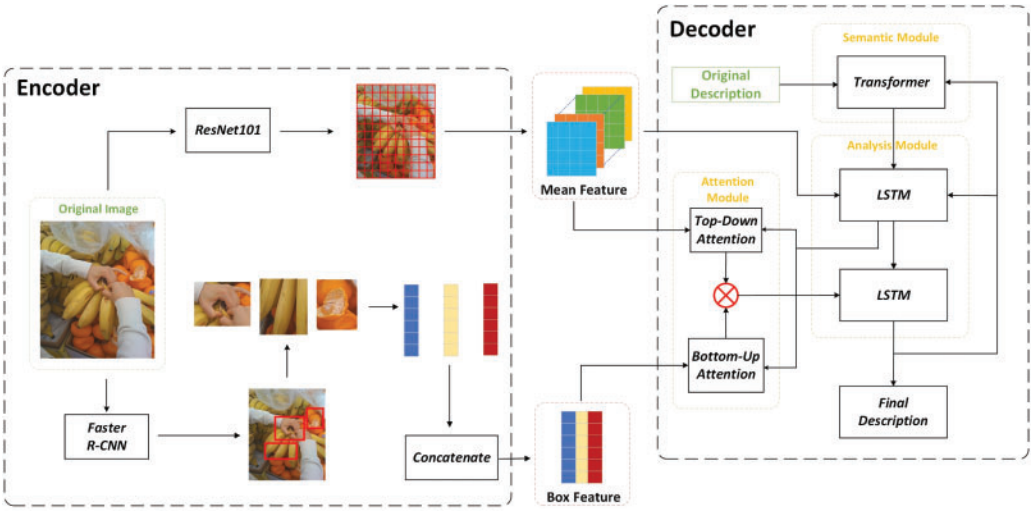


Figure 6: Schematic of the base model

The diagram outlines a sophisticated neural network architecture for generating image descriptions, integrating both convolutional and recurrent components with an attention mechanism. Initially, the encoder uses a ResNet-101 architecture to extract global features from the original image,

while a Faster R-CNN identifies and encodes specific objects within the image, producing localized box features. These two sets of features are concatenated to provide a comprehensive representation of the image content.

On the decoder side, an attention module utilizes both top-down and bottom-up attention mechanisms to focus on relevant parts of the image when generating a description. The top-down attention is a deliberate process guided by the current state of the language model, while the bottom-up attention is driven by the saliency of objects in the image features. These attention-guided features are then processed by an LSTM (Long Short-Term Memory) network, which is part of the analysis module, to handle the sequence prediction of the description.

Simultaneously, a transformer within the semantic module works to capture the complex dependencies between different parts of the image. The outputs from both the LSTM and transformer modules are integrated to produce the original description, which is further refined by additional passes through the LSTM to yield the final descriptive output. This architecture exemplifies the integration of deep learning techniques for complex tasks like image captioning, where both the visual features and their semantic relationships need to be understood and described in natural language.

3.1.1 Encoder

The encoder of our model synergistically employs two highly effective neural networks: ResNet-101 and Faster R-CNN, each renowned for their capabilities in image classification and object detection, respectively.

ResNet-101, at the core of our encoder, serves as the primary feature extractor for image classification. Its architectural advantage lies in its residual structure, which effectively circumvents the issue of vanishing gradients commonly encountered in deep networks. This feature ensures consistent training performance even as network depth increases, enabling ResNet-101 to capture intricate and multi-level features of complex images. In our scheme, ResNet-101 is utilized in a transfer learning setup, where it is pretrained on a large dataset and then adapted for our specific task. This results in a rich feature map of dimensions $7 \times 7 \times 2048$ from input images, forming a crucial component for the top-down attention mechanism as detailed in Eq. (5):

$$F_{mean} = ResNet{-}101(I_m) \tag{5}$$

where I_m represents the input image and F_{mean} denotes the average features extracted by ResNet-101.

On the other hand, Faster R-CNN excels in object detection, thanks to its innovative Region Proposal Network (RPN). The RPN efficiently generates high quality candidate regions for object detection, while the shared CNN feature extractor between the RPN and the main detection network optimizes computational resources and enhances the quality of the proposed regions. In our model, Faster R-CNN is tasked with detecting diverse objects within the image, contributing fine-grained scene information. The output from Faster R-CNN, forming the basis for the bottom-up attention mechanism, is a matrix of feature vectors with dimensions $n \times 512$ for n detected target regions, as shown in Eq. (6):

$$F_{box} = Faster\ R\text{-}CNN(I_m) \tag{6}$$

where I_m represents the input image and F_{box} refers to the local features extracted by Faster R-CNN.

By combining these two powerful networks, our encoder effectively captures both the holistic and detailed elements of the input images. This dual approach ensures a comprehensive understanding of

the visual content, essential for generating accurate and contextually relevant image descriptions in subsequent stages of the model.

3.1.2 Decoder

The decoder is the core component for generating descriptions, composed of the attention module, the semantic module, and the parsing module.

(1) The Attention Module

The attention module's main task is to use the output of the hidden state from the first layer LSTM as the query vector [23], combined with the image features extracted by the encoder, to calculate weights for both top-down and bottom-up attention mechanisms. After weighted fusion, the features are passed to the second-layer LSTM for further processing. In this study, the top-down attention mechanism primarily focuses on the entire image, assessing the contribution of each grid area to the generation of the current word, involving considerations of the global context. Conversely, the bottom-up attention mechanism concentrates on specific target areas, calculating the contributions of these n specific regions to the current output, representing a more fine-grained and local perspective of the image. By combining these two approaches, the decoder ensures that key information contributing to word generation is captured from both the global structure and local details of the image.

(2) The Semantic Module

This module employs the encoding part of a Transformer to deeply capture long-range dependencies within the input sequence, effectively address remote dependencies in text. Specifically, the encoding section of the Transformer consists of several layers stacked upon each other, each layer combining multi-head self-attention mechanisms, feed forward neural networks, layer normalization, and residual connections [24]. The network structure of the Transformer encoding is depicted in Fig. 7.

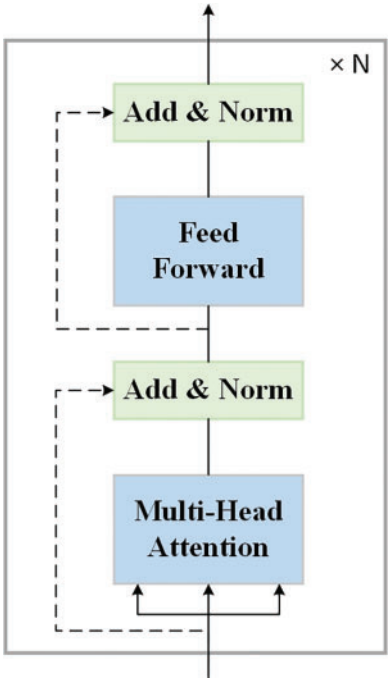


Figure 7: Schematic of the Transformer encoding architecture

As shown in Fig. 7, the diagram illustrates a single layer within the encoder of a Transformer model, which is repeated N times to form the full encoder stack. The process begins with a multi-head attention mechanism that allows the model to simultaneously process different parts of the input data, capturing a variety of contextual information. Following this attention phase, the outputs are then normalized and combined with the original inputs through a residual connection in the "Add & Norm" step, which helps to stabilize the learning process by allowing gradients to flow through the network more effectively.

Afterward, the data passes through a position-wise feed-forward network, which applies a set of learned transformations to the data. This is again followed by an "Add & Norm" step, similar to the one that follows the multi-head attention module, to further refine and stabilize the network's outputs. The output of this layer is then fed into the next identical layer, and this process is repeated N times, with each layer having the potential to learn increasingly complex representations of the input data.

The introduction of the multi-head self-attention mechanism allows the model to consider the various degrees of association between each element in the input sequence during processing. This not only enhances the interaction among the elements within the sequence but also enables the model to better capture complex contextual relationships. The feed forward neural networks further deepen and broaden this representation. Meanwhile, layer normalization and residual connections ensure that the model can learn and optimize in a more stable manner throughout the training process. In processing the input sequence, the semantic module not only ensures a full understanding of the contextual information but also provides a rich semantic foundation for the subsequent parsing module [25]. This makes the model more sensitive to details and relationships, thereby improving the accuracy and quality of the output. The main computational process of this module is shown in Eqs. (7) and (8):

$$S_i = Transformer\ (w_{t-1}, h_0) \tag{7}$$

$$H = LSTM_1\ (w_{t-1}, F_{mean}, S_i) \tag{8}$$

where $Transformer\ ()$ denotes the computational process of the Transformer encoding part; $LSTM_1()$ represents the computational process of the first layer LSTM; h_0 is the current semantic information; S_i is the extracted semantic information; w_{t-1} is the word generated at the previous time step; and H is the hidden state.

(3) Analysis Module

The Analysis Module, a key part of the decoder, is composed of two layers of Long Short-Term Memory (LSTM) networks. The first layer LSTM incorporates the word vector from the previous time step, average image features, and overall semantic information as inputs. This design not only takes into account the word context from the previous moment but also integrates the global features of the image and accumulated semantic information, providing a richer contextual background for the model. After processing by the first layer LSTM, the global image features and the semantics of the entire sentence are progressively refined. The hidden state of this LSTM layer is then used as the query vector for the attention module, enabling the weighted fusion of features. The attention-weighted regional features are further processed by the second layer LSTM, which translates them into corresponding word outputs. This approach ensures that each word in the image description closely aligns with the specific content of the image, enhancing the accuracy of the description. The main computational process of the Analysis Module is delineated in Eqs. (9) to (11):

$$Atten_{TD} = Top\text{-}Down\ \ (H, F_{mean}, F_{mean}) \tag{9}$$

$$Atten_{BU} = Bottom\text{-}Up\ \ (H, F_{box}, F_{box}) \tag{10}$$

$$w_t = LSTM_2\ \ (Atten_{TD}, Atten_{BU}, H) \tag{11}$$

where *Top-Down*() and *Bottom-Up*() represent the computational processes for top-down and bottom-up attention mechanisms, respectively; $LSTM_2()$ signifies the operation of the second layer LSTM; $Atten_{TD}$ and $Atten_{BU}$ are the attention weights from top-down and bottom-up approaches; H denotes the hidden state; and w_t is the word generated at the current moment.

For model training in this study, traditional supervised learning methods were employed. To evaluate model performance and provide gradients for backpropagation, a cross-entropy loss function was used [26], as shown in Eq. (12):

$$Loss = \frac{1}{N}\sum_i L_i = -\frac{1}{N}\sum_{i=0}^{N-1}\sum_{c=0}^{C-1} y_{ic}\log p_{ic} \tag{12}$$

where *Loss* represents the loss value; N is the number of sample categories; L_i is the entropy for each category; y_{ic} is the result computed by the model; p_{ic} is the probability corresponding to that result.

Cross-entropy is a commonly used method to assess the discrepancy between the outputs of a model and the actual labels, especially in classification problems and sequence generation tasks. To optimize the model more effectively, this paper opts for the Adam optimizer. Adam is an adaptive learning rate method that adjusts the learning rate according to the parameters of the model, allowing for more efficient updates of the model's parameters. Compared to the traditional stochastic gradient descent method, Adam performs better in many tasks because it converges more quickly and avoids getting stuck in local optima.

3.2 Reward

The reward function plays a crucial role in providing behavior feedback to the agent by generating a real-time scalar reward value for each action. This value indicates to the agent whether the action is appropriate in a specific environmental state.

Initially, the model receives descriptive text information, which must be converted into a vector form understandable by the computer. This conversion is accomplished through an embedding layer that maps each word to a high dimensional space. Subsequently, these word vectors are processed by a Transformer encoder. This step allows the model to capture the semantic relationships and contextual information within the text description, thereby creating a semantically rich embedding.

Simultaneously, the model processes the information from the image. Both the image features and the corresponding bounding box features are encoded into vector form. These two feature vectors are then fused to form a comprehensive visual embedding. This embedding provides the model with a complete and structured representation of the entire visual scene.

The core part of the model involves comparing these two embeddings: the semantic embedding of the text description and the visual embedding of the image. To measure their similarity, the cosine similarity between these two embeddings is calculated [27]. This similarity indicates the consistency between the text description and the image content. If the similarity value is high, indicating a high match between the text description and the image content, the reward value approaches 1; if the similarity value is low, indicating a significant mismatch, the reward value approaches −1. The computational process of the reward function is shown in Eqs. (13) to (15):

$$I_{map} = 0.5(W_{mean}F_{mean} + b_{mean} + W_{box}F_{box} + b_{box}) \tag{13}$$

$$S_{map} = W_{seg}S_i + b_{seg} \tag{14}$$

$$r = \cos\theta = \frac{I_{map} S_{map}}{\left\| I_{map} \right\| \left\| S_{map} \right\|} \tag{15}$$

where I_{map} represents the image features mapped to a high dimensional space; S_{map} represents the semantic features mapped to a high dimensional space; W_{mean}, W_{box}, W_{seg} are the mapping weights; b_{mean}, b_{box}, b_{seg} are the mapping biases; r is the reward value; and θ is the angle between the image features and semantic features.

3.3 Value

The value function is built upon a Multilayer Perceptron (MLP) structure, aiming to predict the future expected total return for a given state. In this model, the input information consists of the average image features, bounding box features, and descriptive text information.

Initially, descriptive sentences are converted into word embedding vectors. These vectors are then passed to an LSTM network, whose initial hidden state is derived from the average image features. The design of the LSTM network enables the model to capture the sequence and semantic information in the description, thereby generating a fixed size representation vector for the entire description.

Next, the average image features, bounding box features, and the descriptive vector obtained from the LSTM are integrated into a composite state vector. This vector is first processed through two linear layers and then outputs a scalar value. This value represents the expected cumulative return that can be obtained from the current state in the future.

This design structure allows the model to make decisions based on both visual and textual information, accurately estimating the expected return value in complex environments. During the training process of reinforcement learning, an iterative approach is generally used to train the model components effectively [28]. Initially, the reward function is fixed to provide stable and reliable feedback for the agent during the early stages of training. When training the value function, a supervised learning approach is usually adopted. Specifically, real images and descriptive sentences are used as inputs, and the model is expected to output a value function that approximates the real reward given by the reward function. To measure the gap between the model's predictions and the actual reward, this paper employs the mean squared error loss function $\mathrm{MSE}_{\mathrm{Loss}}$, as shown in Eq. (16), and adjusts the model's parameters using the Adam optimizer to minimize this gap:

$$MSE_{Loss} = \frac{1}{N} \sum_{i=1}^{N} \left(\widetilde{y}_i - y_i \right)^2 \tag{16}$$

where y_i is the predicted value; $\widetilde{y}_i$ is the actual value.

When training the policy function, a reinforcement learning method is adopted in this paper. In this case, only images are input, without descriptive sentences, allowing the model to autonomously generate descriptions based on the current policy. The reward function and the value function then provide feedback for the generated descriptions, adjusting the strategy so that it achieves the highest possible cumulative reward while minimizing the Loss, as shown in Eq. (17):

$$Loss = -\frac{1}{N} \sum_{n-1}^{N} \sum_{t-1}^{T} \left[r_t^n + V\left(s_{t+1}^n\right) - V\left(s_t^n\right) \right] \log\left[P_\pi\left(a_t^n | s_t^n\right) \right] \tag{17}$$

where n represents the sample number; t denotes the time step; s_t^n is the hidden state of the n-th sample at time t; a_t^n is the word generated by the n-th sample at time a_t^n; r_t^n is the feedback received by the n-th sample at time t; the advantage function is represented as $V\left(s_{t+1}^n\right) - V\left(s_t^n\right)$; $V\left(s_t^n\right)$ is a value function

assessing the value of the current state; P_π is the probability; $P_\pi\left(a_t^n|s_t^n\right)$ describes the probability of the policy function executing a certain action given the current state.

4 Experiments

4.1 Experimental Environment

The experimental environment for this study included a 10th Gen Intel(R) Core(TM) i7-10700K processor, using the deep learning framework Pytorch based on Python 3.10, with PyCharm as the development tool. The operating system was Windows, and training was conducted with an Nvidia 1050Ti GPU.

4.2 Data Preprocessing

This study employs the MSCOCO dataset, a large-scale image dataset encompassing a variety of scenes and objects. It contains 127,266 images, each accompanied by five unique descriptions, offering a wealth of visual and textual information. For training and evaluating the model, 117,266 images were used for training, 5,000 for validation, and an additional 5,000 for testing.

In order to build the model effectively, the following data preprocessing steps were undertaken. All description sentences were consolidated and text-cleaned, which included converting all words in the descriptions to lowercase and removing rare words that appeared less than five times. Moreover, to enhance the model's understanding and generation of descriptions, special markers were added to the descriptions, such as <start> to indicate the beginning of a description, <end> for its end, <unk> for unknown vocabulary, and <pad> as a padding symbol. These preprocessing steps ultimately resulted in a vocabulary of 10,601 words.

4.3 Experimental Details

Throughout the training process, the model initially trained the reward network through supervised learning to ensure it could provide accurate instantaneous reward assessments. Then, this network was frozen, and training alternated between the policy network and the value network. The selection of map dimensions in our model is based on specific requirements and the architecture of the model. The dimensions of the average image features were set to $7 \times 7 \times 2048$. These dimensions are chosen to match the output format of the ResNet-101 architecture used in the encoder, which is standard for processing images in deep learning models. The dimensions of the local features were set to $n \times 512$, where n represents the number of detected objects or regions in the image, and 512 is the feature vector length for each object or region. In the semantic module, the settings followed the hyperparameters of the Transformer, which had a dimension of 512. This is a common dimension size for Transformer models, balancing computational efficiency and capacity for complex tasks. The multi-head attention mechanism in this module had 8 heads, a standard choice for enabling the model to attend to different parts of the input data in parallel. The two-layer LSTM in the analysis module had a hidden layer dimension of 512, which is typical for LSTMs used in natural language processing tasks, providing a balance between model complexity and the ability to capture long-term dependencies in text. These dimensions are chosen to optimize the model's performance in processing and understanding both the visual and textual components of the data, as well as to ensure compatibility with standard architectures and practices in deep learning. During the 300 epochs of training, each batch size was set to 1, with effective batch size increased through the gradient accumulation technique. In the inference phase, a beam search strategy was employed, setting the beam size to 3.

4.4 Experimental Results and Analysis

4.4.1 Evaluation Metrics

In this study, BLEU, CIDEr, and ROUGE metrics are used to evaluate the performance of the model on the test dataset. Higher scores on these metrics indicate better quality of the generated descriptive sentences.

(1) BLEU

Bilingual Evaluation Understudy (BLEU) is a common evaluation method in the field of machine translation, used to measure the similarity between machine-generated translations and human reference translations. The core concept of BLEU is to compare the n-grams of the machine translation output and the reference translation [29]. The calculation process of BLEU is shown in Eq. (18):

$$BLEU = BP \cdot exp\left(\sum_{n=1}^{N} w_n \log P_n\right) \tag{18}$$

where BP represents the brevity penalty factor, P_n indicates the precision of n-grams, and w_n denotes the weights.

(2) CIDEr

Consensus-based Image Description Evaluation (CIDEr) is a metric specifically designed for evaluating image description generation. Its main goal is to measure the consensus between machine-generated descriptions and a set of reference descriptions. By assigning TF-IDF weights to each n-gram, this metric emphasizes n-grams that are unique in a particular description but infrequent in most descriptions. The calculation process involves comparing the TF-IDF vectors of the machine-generated description with those of each reference description. CIDEr's computation is centered around combining n-grams of different lengths and then averaging these similarities to form a composite score.

(3) ROUGE

Recall-Oriented Understudy for Gisting Evaluation (ROUGE) is an evaluation metric for assessing the quality and similarity between automatic summaries or machine translations and reference summaries. The core idea of this evaluation method is to compare the degree of n-gram matches between the generated text and the given reference text.

4.4.2 Result Analysis

After training the model, we employed a suite of evaluation metrics, including BLEU, CIDEr, and ROUGE scores, to rigorously assess its performance in generating image descriptions. The model iteration with the highest scores was selected as the optimal model. This model was then used to process a diverse set of test images, generating descriptive statements for each.

To quantify the results, we calculated the average scores across the test dataset for each evaluation metric. The BLEU score, for instance, provided insights into the linguistic accuracy of the descriptions, while the CIDEr score assessed the semantic relevance. Additionally, the ROUGE score offered a measure of the comprehensiveness of the descriptions.

These results were further analyzed to understand the model's performance in various scenarios, such as describing complex scenes or diverse objects. We also conducted a comparative analysis with baseline models to highlight the improvements our model offers. All these details are encapsulated in comprehensive tables and figures, with statistical analyses performed to validate the significance of the findings.

(1) Ablation Experiment

In this experiment, the baseline model is one that constructs the policy function in reinforcement learning. Therefore, the entire model can be divided into two categories according to the training steps: the baseline model and the deep reinforcement learning model. Table 1 displays the results of the ablation experiment. Theoretically, the deep reinforcement learning model, being an optimization of the baseline model, achieves higher scores on the evaluation metrics.

Table 1: Results of the ablation experiment

Model	BLEU-1	BLEU-4	CIDEr	ROUGE
Basic model	0.650	0.262	0.793	0.504
Ours	0.762	0.299	0.998	0.552

The baseline model's encoder extracts average image features using the pretrained ResNet-101 network, and local features using Faster R-CNN. The decoder, with the se-mantic features extracted by the Transformer and the average features from the encoder, feeds into the first-layer LSTM. Both average and local features extracted by the encoder enter the top-down and bottom-up attention for weighted fusion and, along with the output of the first layer LSTM, become the input for the second layer LSTM, which generates the descriptive sentences.

The deep reinforcement learning model in this study is an enhancement of the baseline model, incorporating reinforcement learning methods to improve the task of image description generation for more accurate descriptive sentences. Within the model, the reward function assigns rewards and it assigns penalties or rewards between $(-1, 1)$ depending on the similarity between the image and the text. The value function, using a Multilayer Perceptron (MLP), predicts the expected return for a given state. The policy function, which utilizes the aforementioned baseline model, is employed for the initial generation of descriptions. Thus, the deep reinforcement learning model, through a process of continuous trial and error learning, produces more accurate descriptive texts.

Theoretically, since the deep reinforcement learning model is an optimized version of the baseline model, it achieves improved scores across various evaluation metrics. As evidenced in Table 1, compared to the baseline model, the deep reinforcement learning model shows an increase of 0.112 in BLEU-1, 0.045 in BLEU-4, 0.205 in CIDEr, and 0.048 in ROUGE. These experimental results validate the feasibility and accuracy of using deep reinforcement learning methods for the task of image description generation.

(2) Comparative Experiment

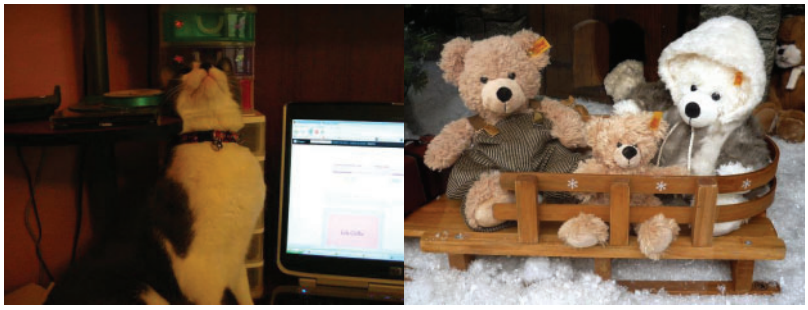
The comparative experiment results from various studies on image description tasks on the COCO dataset are compared with the method employed in this study, as shown in Table 2.

An examination of the results presented in Table 2 clearly demonstrates the superior performance of our experimental method in the BLEU-1 and CIDEr metrics when compared to other approaches. This enhanced performance can be largely attributed to the effective application of deep reinforcement learning in our model. Specifically, the reinforcement learning framework plays a pivotal role in the initial stages of description generation. By systematically rewarding or penalizing the production of individual words, it ensures that the generated descriptions not only adhere to local semantic accuracy but also maintain relevance to the visual context of the images.

Table 2: Comparative experiment results

Model	BLEU-1	BLEU-4	CIDEr	ROUGE
Basic model	0.650	0.262	0.793	0.504
Literature [7]	0.716	0.299	0.917	0.521
Literature [30]	0.731	0.316	0.943	0.535
Literature [9]	0.750	0.340	1.040	0.550
Literature [10]	0.705	0.277	0.865	0.516
Ours	0.762	0.299	0.998	0.552

For instance, Fig. 8a provides a compelling example of this accuracy. The model-generated description, 'There is a cat sitting beside a computer on a desk.' exhibits a high degree of similarity to the manually annotated reference sentence, 'A cat sitting beside a laptop on a desk.' This close alignment between the generated and reference descriptions underscores the efficacy of our method. It highlights how the deep reinforcement learning approach not only guides the model towards producing contextually appropriate descriptions but also enhances the overall precision of the language used. This result is indicative of the potential of deep reinforcement learning in bridging the gap between visual perception and natural language description, offering a promising direction for future advancements in automated image captioning.



(a) figure of better experiments (b) figure of mediocre experiments

Figure 8: Schematic of the experimental results

The experimental results reveal that while our model excels in BLEU-1 and CIDEr metrics, its performance in BLEU-4 and ROUGE metrics is comparatively average. This outcome highlights a significant challenge: our model appears to struggle with establishing long-distance word group relationships and crafting coherent overall sentence structures. Two potential factors might contribute to this issue. Firstly, the model may be achieving local optima during the training process, which hinders its ability to grasp more complex, global sentence structures. Secondly, the reliance on manually annotated reference texts for evaluation poses its own set of challenges. These reference texts might not always represent the full spectrum of correct descriptions, potentially leading to lower scores when the model's valid descriptions differ from these specific references.

For example, as illustrated in Fig. 8b, the model generates the description 'a brown teddy bear sitting next to a pile of teddy bears.' This description, while contextually accurate and valid upon manual assessment, markedly differs from the manually annotated reference sentence 'Three teddy

bears sit on a fake sled in fake snow.' The divergence between the generated description and the reference text, in this case, results in lower scores for the model in the BLEU-4 and ROUGE metrics. This scenario underscores the need for a more nuanced evaluation approach that can accommodate a wider range of correct descriptions and better capture the model's ability to generate contextually and semantically relevant sentences. Addressing these challenges will be crucial in refining the model's capabilities in generating more structurally complex and varied descriptions.

5 Conclusions

This paper introduces a novel approach for image description generation using deep reinforcement learning. By employing policy, value, and reward functions, our model effectively produces descriptive texts. Our experiments demonstrate the method's feasibility and accuracy. Looking ahead, we plan to investigate different neural network architectures, including various convolutional and recurrent networks, to deepen our understanding of the relationship between images and text. Additionally, we aim to incorporate advanced generative models like Generative Adversarial Networks (GANs) and diffusion models to enhance the naturalness and creativity of the image descriptions. This research marks a step towards more sophisticated and intuitive visual-to-textual interpretation by machines.

Acknowledgement: Not applicable.

Funding Statement: This research was funded by the Natural Science Foundation of Gansu Province with Approval Numbers 20JR10RA334 and 21JR7RA570. Funding is provided for the 2021 Longyuan Youth Innovation and Entrepreneurship Talent Project with Approval Number 2021LQGR20 and the University Level Innovation Project with Approval Numbers GZF2020XZD18 and jbzxyb2018-01 of Gansu University of Political Science and Law.

Author Contributions: The authors confirm contribution to the paper as follows: study conception and design: Li Yan, Wang Qiyuan; data collection: Jia Kaidi; analysis and interpretation of results: Wang Qiyuan, Li Yan, Jia Kaidi; draft manuscript preparation: Wang Qiyuan, Li Yan. All authors reviewed the results and approved the final version of the manuscript.

Availability of Data and Materials: Data openly available in a public repository. The data that support the findings of this study are openly available in dataset at http://cocodataset.org.

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

References

[1] H. Wang, Y. Zhang, and X. Yu, "An overview of image caption generation methods," *Comput. Intell. Neurosci.*, vol. 2020, no. 3, pp. 1–13, 2020. doi: 10.1155/2020/3062706.

[2] A. Farhadi *et al.*, "Every picture tells a story: Generating sentences from images," *Presented at the Computer Vision–ECCV 2010: 11th European Conference on Computer Vision*, Heraklion, Crete, Greece: Springer, Sep. 5-11, 2010, pp. 15–29.

[3] G. Kulkarni *et al.*, "Understanding and generating simple image descriptions," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 12, pp. 2891–2903, 2013. doi: 10.1109/TPAMI.2012.162.

[4] J. Meng, Y. Li, H. Liang, and Y. Ma, "Single-image dehazing based on two-stream convolutional neural network," *J. Artif. Intell. Technol.*, vol. 2, no. 3, pp. 100–110, 2022. doi: 10.37965/jait.2022.0110.

[5] M. Zheng, K. Zhi, J. Zeng, C. Tian, and L. You, "A hybrid CNN for image denoising," *J. Artif. Intell. Technol.*, vol. 2, no. 3, pp. 93–99, 2022. doi: 10.37965/jait.2022.0101.
[6] M. Hodosh, P. Young, and J. Hockenmaier, "Framing image description as a ranking task: Data, models and evaluation metrics," *J. Artif. Intell. Res.*, vol. 47, pp. 853–899, 2013. doi: 10.1613/jair.3994.
[7] J. Mao, W. Xu, Y. Yang, J. Wang, Z. Huang and A. Yuille, "Deep captioning with multimodal recurrent neural networks (m-RNN)," arXiv preprint arXiv:1412.6632, 2014.
[8] R. Chen, D. Pu, Y. Tong, and M. Wu, "Image-denoising algorithm based on improved K-singular value decomposition and atom optimization," *CAAI Trans. Intell. Technol.*, vol. 7, no. 1, pp. 117–127, 2022.
[9] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, "Show and tell: A neural image caption generator," in *Proc. of the IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 3156–3164.
[10] K. Xu *et al.*, "Show, attend and tell: Neural image caption generation with visual attention," in *Int. Conf. Mach. Learn.*, 2015, pp. 2048–2057.
[11] F. Ahmad, "Deep image retrieval using artificial neural network interpolation and indexing based on similarity measurement," *CAAI Trans. Intell. Technol.*, vol. 7, no. 2, pp. 200–218, 2022. doi: 10.1049/cit2.12083.
[12] M. Cornia, L. Baraldi, and R. Cucchiara, "Show, control and tell: A framework for generating controllable and grounded captions," in *Proc. of the IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 8307–8316.
[13] H. Chugh *et al.*, "An image retrieval framework design analysis using saliency structure and color difference histogram," *Sustainability*, vol. 14, no. 16, pp. 10357, 2022. doi: 10.3390/su141610357.
[14] J. Lu, C. Xiong, D. Parikh, and R. Socher, "Knowing when to look: Adaptive attention via a visual sentinel for image captioning," in *Proc. of the IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 375–383.
[15] K. H. Lee, X. Chen, G. Hua, H. Hu, and X. He, "Stacked cross attention for image-text matching," in *Proc. of the Euro. Conf. on Comput. Vis. (ECCV)*, 2018, pp. 201–216.
[16] P. Chun, T. Yamane, and Y. Maemura, "A deep learning-based image captioning method to automatically generate comprehensive explanations of bridge damage," *Comput. Civ. Infrastruct. Eng.*, vol. 37, no. 11, pp. 1387–1401, 2022. doi: 10.1111/mice.12793.
[17] M. Shafiq and Z. Gu, "Deep residual learning for image recognition: A survey," *Appl. Sci.*, vol. 12, no. 18, pp. 8972, 2022. doi: 10.3390/app12188972.
[18] Z. Zhao, Z. Luo, P. Wang, and J. Li, "Survey on image classification algorithms based on deep residual network," *Comput. Syst. Appl.*, vol. 29, no. 1, pp. 14–21, 2020. doi: 10.15888/j.cnki.csa.007243.
[19] P. Anderson *et al.*, "Bottom-up and top-down attention for image captioning and visual question answering," in *Proc. of the IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 6077–6086.
[20] R. C. Gonzalez, "Deep convolutional neural networks [Lecture Notes]," *IEEE Signal Process. Mag.*, vol. 35, no. 6, pp. 79–87, 2018. doi: 10.1109/MSP.2018.2842646.
[21] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *Adv. Neural. Inf. Process. Syst.*, vol. 28, pp. 91–99, 2015.
[22] Q. Cheng, G. Chen, L. Wang, and C. Guan, "Millimeter wave image object detection based on convolutional neural network," *Sci. Technol. Eng.*, vol. 20, pp. 5224–5229, 2020.
[23] X. Jia, E. Gavves, B. Fernando, and T. Tuytelaars, "Guiding the long-short term memory model for image caption generation," in *Proc. of the IEEE Int. Conf. Comput. Vis.*, 2015, pp. 2407–2415.
[24] Y. Pan, T. Yao, Y. Li, and T. Mei, "X-linear attention networks for image captioning," in *Proc. of the IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 10971–10980.
[25] D. P. Kingma, and J. Ba, "Adam: A method for stochastic optimization," arXiv preprint arXiv:1412.6980, 2014.
[26] S. Herdade, A. Kappeler, K. Boakye, and J. Soares, "Image captioning: Transforming objects into words," *Adv. Neural. Inf. Process. Syst.*, vol. 32, pp. 11137–11147, 2019.
[27] L. Ye, Z. Wang, X. Chen, J. Wang, K. Wu and K. Lu, "GSAN: Graph self-attention network for interaction measurement in autonomous driving," in *2020 IEEE 17th Int. Conf. on Mobile Ad Hoc and Sensor Systems (MASS)*, IEEE, 2020, pp. 274–282.
[28] A. Vaswani *et al.*, "Attention is all you need," *Adv. Neural Inf. Process. Syst*, vol. 30, pp. 5998–6008, 2017.

[29] Q. You, H. Jin, Z. Wang, C. Fang, and J. Luo, "Image captioning with semantic attention," in *Proc. of the IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 4651–4659.
[30] W. Liu, S. Chen, L. Guo, X. Zhu, and J. Liu, "CPTR: Full transformer network for image captioning," arXiv preprint arXiv:2101.10804, 2021.