IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received April 22, 2019, accepted May 16, 2019, date of publication May 20, 2019, date of current version June 4, 2019.

*Digital Object Identifier 10.1109/ACCESS.2019.2917771*

# Multilayer Dense Attention Model for Image Caption

**ERIC KE WANG[1], XUN ZHANG[1], FAN WANG[1], TSU-YANG WU[2], AND CHIEN-MING CHEN[2]**
[1]Harbin Institute of Technology, Shenzhen 518055, China
[2]College of Computer Science and Engineering, Shandong University of Science and Technology, Shandong 266590, China

Corresponding author: Chien-Ming Chen (chienmingchen@ieee.org)

This work was supported in part by the National Natural Science Foundation of China under Grant 61572157, Grant 2016A030313660, and Grant 2017A030313365, in part by the Guangdong Province Natural Science Foundation under Grant JCYJ20160608161351559, Grant KQJSCX70726103044992, Grant KQJSCX20170327161755, Grant JCYJ20170811155158682, and Grant JCYJ20160428092427867, and in part by the Shenzhen Municipal Science and Technology Innovation Project.

**ABSTRACT** The image caption is a technology that enables us to understand the contents and generate descriptive text, of images using machines. With the development of deep learning, means of using it to understand image content and generate descriptive text has become a hot research topic. This paper proposes a multilayer dense attention model for image caption. A faster recurrent convolutional neural networks (Faster R-CNN) is employed to extract image features as the coding layer, the long short-term memory (LSTM)-attend is used to decode the multilayer dense attention model, and the description text is generated. The model parameters are optimized using strategy gradient optimization in reinforcement learning. Use of dense attention mechanisms in the coding layer can effectively avoid the interference of non-salient information and selectively output the corresponding description text for the decoding process. The experimental results in the field of general images validate the model's good ability to understand images and generating text.

**INDEX TERMS** Attention, image caption, LSTM, RCNN.

## I. INTRODUCTION

Image caption is a cross-disciplinary research problem involving computer vision, natural language processing (NLP), and machine learning. The input of the image caption model is an image and the output is a text describing the image. This task requires the model to recognize the objects in an image, understand the relation between objects, and express them in a natural language sentence. Actually, many practical applications need the image caption technology. For example, after taking a picture, users can use this technology to match the appropriate text, which can replace the user's manual filling with text. Besides, it can help the visually impaired to understand the image content. Similar tasks include video caption, where the input is a video and the output is its description. These tasks require the system to be able to understand the relation between objects to capture the semantic information of the image and generate human-readable sentences, they are still a challenge for machines.

The essence of image caption is to study the problem of how to realize mapping from the visual-to-language framework. With the development of artificial intelligence and big data, the algorithms are expected to generate natural language sentences that can describe the image content. However, this basic behavior in humans' daily lives is a great challenge for machines: image caption needs to be ''translated'' between two forms of information (image to text), which involves multimode fusion, and derives many research difficulties in the field of pattern recognition.

Among the difficulties, how to represent and measure the similarity between image and text accurately in a robust way is the key problem [1], [2]. The existing solution can be divided into two main categories according to the ways of modeling the relation between image and text: one-to-one matching and many-to-many matching. One-to-one matching usually extracts the global feature representation of the image and text, and then uses the objective function of the structured or canonical correlation analysis to project their features into a common space to make similar pairs. Image text is close in space, that is, it has high similarity. However, this matching method only roughly measures the global similarity of the image text, without specifically considering

The associate editor coordinating the review of this manuscript and approving it for publication was Sudhakar Radhakrishnan.

which local content of the image text is semantically similar. Therefore, in some tasks requiring precise similarity measurement, such as fine-grained cross-modal retrieval, its experimental accuracy is often low. In the many-to-many matching method, multiple local instances from the image text are extracted separately, and then the local similarity of multiple paired instances are measured and fused to obtain global similarity [3], [4]. However, not all examples extracted by these methods depict semantic concepts. In fact, most of them are meaningless and independent of the matching tasks. Only a few significant semantic instances determine the matching degree. These redundant instances can also be considered noises that interfere with the matching process of a few semantic instances and increase the computational complexity of the model. In addition, existing methods usually need to explicitly use additional target detection algorithms or expensive manual labeling in case extraction.

This pper proposes a multilayer dense attention model. A faster Recurrent Convolutional Neural Networks (Faster R-CNN) is employed to extract image features as the coding layer, Long Short-Term Memory(LSTM)-attend is used to decode the multilayer dense attention model, and the description text is generated. The parameters of the model are optimized using strategy gradient optimization in reinforcement learning. Using multiple layers of dense attention mechanisms in the coding layer can effectively avoid the interference of non-salient information, and selectively output the corresponding description text for the decoding process. The experimental results in the field of general images show that our model has good image understanding and text generation abilities, and its generation effect is better than state-of-the-art models.

## II. RELATED WORK

Google introduced a neural image caption generator [5] in 2014. It is not like the previous methods based on rules and classification, which are influenced by the successful cases of the machine translation model. CNN-based InceptionNet is employed to extract image features. In contrast, RNN is used as a decoder to accept CNN-extracted images, where a vanilla RNN can be replaced with a LSTM or GRU to obtain better long-term memory. At the same time, neural talk [6] has been proposed at Stanford University, whose architecture is almost the same as that of Google's model. The only difference is that it uses VGGnet as its image feature extractor.

The above two models are the initial models based on the encoder-decoder framework, which has a significant impact on the field of image understanding. Subsequently, many related studies begin with this framework.

Microsoft proposed an improvement method in the encoder side [10]. It used multiinstance training to train a word explorer to generate a series of words that may appear in the caption for each image. Then, the obtained words are used as input to generate a series of descriptive sentences about the picture, using the language model. Finally, the results of the sentences were selected from them. This method of generating sentences by extracting keywords as the input undoubtedly provides a reference for the coding method, which combines the image and semantics.

Li and Chen [9] proposed a new feature extraction method. When extracting image features, a series of target detection frames are obtained by the target detection algorithm as image features and an attribute detector is trained with image features as the input. Attributes as high-level semantic features, together with the extracted image features, are used as the input of the specially designed visual-semantic LSTM, and then decoded. This use of the target detection makes the input features more ''dense,'' rather than directly entering the whole image as before, to achieve a visual attention-like effect. Bottom-up and top-down attention [8] also employs a similar encoder structure.

Wang *et al.* proposed a new decoder structure, called skeleton-attribute decoder, which consists of Skel-LSTM and Attr-LSTM. Skel-LSTM uses image features extracted by CNN to obtain a trunk sentence, while Attr-LSTM uses the latter to obtain a series of attributes for each word in the trunk sentence; then, the two parts of the words are merged into a final caption. A similar work is neural baby talk. Inspired by baby talk [11], based on sentence template filling, Lu *et al.* [12] proposed an image caption method based on template generation and slotting. The main idea is to divide the words of the generated sentences into substantive and non-substantive vocabularies. The sentence template is obtained from a language model, and the words come from the non-substantive vocabulary. Entity words are obtained directly from the image by the target detection method, and then used to fill the empty slots in the sentence template to form a sentence. This method pioneered the use of neural networks to extract sentence templates, which successfully solved the problem of lack of diversity input in the second part of the traditional template-filling-based methods.

Similar to the idea of separating decoders, to obtain stylized image caption results, Mathews *et al.* [15] used two decoders: a term generator, using CNN image features as input, and a series of basic semantic pairs obtained through GRU, which are composed of words and attributes. Then, the basic semantics acquired by the term generator were input into the language generator to produce the final output. The language generator used bidirectional GRU coding to sequence the basic semantics, and then the new GRU to decode. To further improve the decoder, Gu *et al.* [17] proposed the idea of stack caption with progressive refinement. Its main innovation lies in the use of a coarse-grained decoder and several fine-grained decoders, in which the coarse-grained decoder accepts image features as input and obtains coarse-grained description results. Next, there is a fine-grained decoder in each stage for more fine-grained decoding, whose input comes from the output results and image features of the decoder in the previous stage, and the attention mechanism is used to expand the fine-grained decoder in different aspects of coarse-grained results in each stage, and ultimately, more detailed results are obtained.

The above research fully demonstrated that using hierarchical or segmented decoding ideas at the decoder can significantly improve the effect of image caption. Such decoding ideas are more similar to the human thinking mode and can be interpreted. It can be foreseen that such a hierarchical structure has the potential to become mainstream in the future.

In addition, there is a completely different study from the traditional RNN-based decoding, which uses a convolutional neural network as the decoder for the caption. Its representative is the convolutional image caption [16] published in CVPR in 2018. This study radically uses masked CNN instead of traditional RNN as a decoder to decode. At the coding stage, words and extracted image features are input into the convolutional encoder at each step, and convolutional decoding is used; finally, the word probability is obtained using the soft max function. This method avoids the time-series limitation of RNN and achieves faster training speed under the same parameters.

It has always been a research hotspot to automatically detect and describe unmentioned objects in pictures beyond Ground Truth's limitation of describing objects. For example, Yao *et al.* [18] introduced the copy mechanism into an image caption decoder in 2017. The basic idea is to introduce the traditional encoder-decoder model into a picture and train a target detector for the image features, then calculate the similarity between the output layer of the decoder LSTM and the detected entity to determine whether the entity should be copied at this step. Because the vocabularies of the caption model and the target detection model are different, the introduction of the copy mechanism can enrich the semantics of the image caption model by introducing entities that are not present in the original caption into the results.

When describing a picture, it may not be enough to use only the knowledge on the picture. The establishment and application of a knowledge atlas is a rapidly developing field, and thus, it is worth introducing external knowledge through a knowledge atlas in image caption. Attempts in this regard include the entity-aware image caption [18] published on EMNLP 2018. In this study, a method similar to neural baby talk is adopted. First, a language template with the entity empty slot is obtained using the encoder-decoder model. Then the entity filling slot is used. In this study, the description of pictures with similar labels of training data is used as the context, from which named entities are extracted and input into the knowledge atlas, and a combination of entities with the highest probability in the atlas is selected as the slot input. This method of introducing external knowledge significantly improves the semantic richness.

With the development of image caption technology, the use of single-image features does not seem to be enough to continue improving the effect of the caption. The joint training of a multimodel in a decoder has been considered. A representative study is that by Gers and Schmidhuber [23], who proposed an image caption model: Groupcap model. This study is inspired from the expectation of encoding multiple images in the caption process so as to obtain similarity and diversity simultaneously. The first part of the model is a visual grammar analysis tree, which is used to model image features by using the tree mechanism. The monitoring signal for this process comes from the parsing tree for the ground-truth statement. The second part of the model is structured correlation and diversity restriction module. For the input picture triplet, the similarity and diversity between them are determined by the leaf node entity of their parsing tree. Besides the target image, each training picture triplet has a positive or negative label to indicate whether it is close to the target image. The training aims to maximize the similarity of the same group of pictures and minimize the similarity of the non-same group. For diversity, the goal is the opposite. The third part is the caption generation link. The three parts are trained jointly to obtain the best output using all extracted features. The multi-model joint training method introduces a new graphics model to acquire the ability to distinguish images, which improves the model's ability to understand image features.

These are some representative image captions developed on the basis of the coder-decoder model. At the coding end, it mainly embodies the introduction of target detection and keyword extraction. At the decoder end, innovative methods such as hierarchical decoding process, convolutional network decoding, and external knowledge introduction are embodied. It can be predicted that the work of image understanding will continue to be extended in such a basic structure for quite a long time [30].

### A. ATTENTION

The success of the attention mechanism in machine translation has aroused interest in image caption. Xu *et al.* [7] proposed an attend-and-tell attention mechanism for image caption. The basic idea of this mechanism is to use a convolution layer to obtain image features, weigh the attention of the image features, and then send them to RNN for decoding. This paper proposes two attention mechanisms: soft-attention mechanism and hard-attention mechanism. Soft-attention mechanism learns an attention weight between 0 and 1 for each image region, whose sum is 1, and then weights the sum of each image region. In the hard-attention mechanism, the maximum weight is set to 1, while the other regions are set to 0, so as to achieve the goal of paying attention to only one region. The attention mechanism has been widely used in practical applications. Because of its good effect and expandability, it has become a mainstream model component.

A representative attention improvement mechanism comes from the scheme "knowing when to look" [21]. Considering that the traditional spatial attention mechanism does not have the flexibility to decide when to acquire new features from image features, the concept of "visual sentry" is proposed. The sentry vector represents the knowledge acquired in the decoder's memory, and is a gate mechanism that controls the weights of the image features after the sentry vector and attend, and adds them as the decoding vector of the time step. Such a method enables the model to make a decision

regarding whether to pay more attention to the semantics acquired in the language model or to the new image features at each step. The experimental results show that this method has better effect and interpretability than the traditional attention mechanism.

Another innovative mechanism for improving image feature attentions is the bottom-up and top-down attention proposed by Anderson *et al.* [8]. Its main innovation lies in using a faster R-CNN for target detection, obtaining corresponding detection targets and labels, and achieving the effect of bottom-up attention. In addition, the LSTM layer of attention is used in the decoder to adjust the real-time attention of the input image features according to the output. This attention mode enables the model to focus on the more obvious and important objects in the image, by marking the description most important.

Besides the attention mechanism of image features, that of linguistic features is also a natural research direction. The language attention mechanism focuses on a series of semantic concepts, which are usually extracted from words with high frequency in training focus. By paying attention to these concepts in the acquisition of sentences, the effect of enriching semantics can be achieved. A representative work in this context is Wu *et al.* [20].

Recently, the attention mechanism has focused on how to combine semantic and visual attentions. Liu *et al.* [21] presented an image caption model, called Sim Net, at the EMNLP conference. Its innovation lies in the use of a merging network to combine visual attentions and semantic attention, in which nouns are used as candidate topics and multi-instance learning is used to extract subject terms from image features as objects of semantic attention. After input attentions, the input image features are coded into the input vectors together with the input text vectors at that time, and then the results of the attend and input codes are sent to the M-Gate through MLP, together with the input image features, through output attentions. The M-Gate weights the subject and input vectors again according to the output attentions to determine which aspects of the semantics and images should be considered in the next step. This dual attention to semantic and image features makes the process of sentence generation pay more attention to the theme and obtain more caption around the theme.

## III. METHOD

Our scheme improves the encoder-decoder framework by using an attention-based model. First, to extract the salient region of an image, we adopt the commonly used method of target detection, which is called the bottom-up attention model, and extract the region that needs special attention in the image. Then, it is put into the decoder to decode. The decoder uses a multilayer top-down attention LSTM model. Then, the design of the parameter model is deeply analyzed, including the selection of the optimizer, the method of training a small batch, and how to prevent over-fitting. Finally, the flowchart of the whole image description is given from the macroscopic viewpoint.

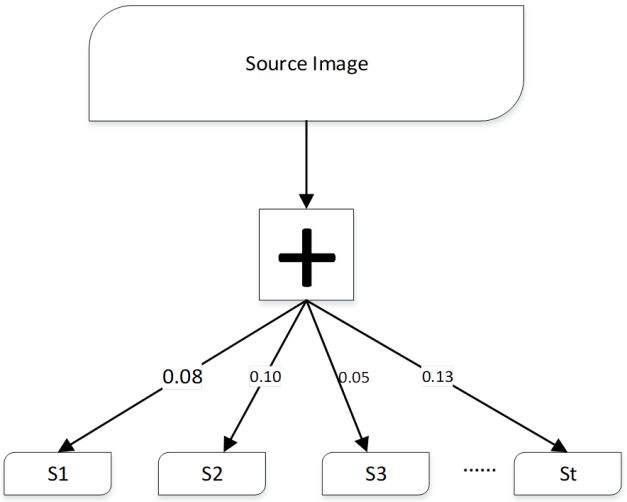


**FIGURE 1.** Attention mechanism.

### A. ATTENTION MECHANISM

Image caption is used to input an image and output its corresponding description. It is commonly carried out using the ''encoder-decoder'' model. The encoder is a convolution network that extracts the high-level features of the image and represents it as a coding vector. The decoder is a language model of the cyclic neural network, and the initial input is a coding vector for generating the description text of the image. In the task of image caption, there are two main problems: encoding capacity bottleneck and long-distance dependence. Therefore, we employ the attention mechanism to select the information. When generating each word in the description, the input of the cyclic neural network is not only the information of the previous word but also the attention mechanism to select some relevant information from the image. The attention mechanism helps the model get rid of the constraints of the fixed vectors, and the decoder fuses different parts of the source image with each output word. It is especially important for the model to decide what to participate based on the input image and what has been generated. As shown in the following figure 1, The words output by each decoder depend on a weight combination of the feature map of an image, and not just the last state. Weight determines the contribution of each feature face to the output state. Therefore, if the weight is large, the decoder will pay more attention to the corresponding parts of the original image when generating the current words of the caption. Weights are usually calculated based on the feature surface of the source image and the hidden layer information at the last moment of the target language. However, the weights are relative, so their calculation should not only be affected by the above two factors but also be related to their historical information. The relevant calculation methods are shown from (1) to (4).

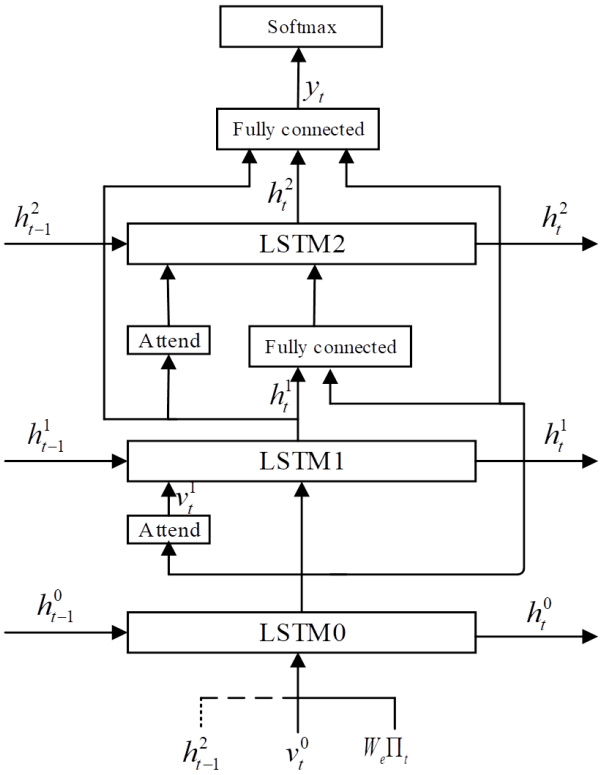


**FIGURE 2.** Dense attention model.

$$e_{ij} = V_a^T tanh(W_a s_{i-1}^y + U_a s_j^x) \quad (1)$$

$$b_{ij}^h = \sum_{i=1}^{t-1} exp(e_{ij}) \quad (2)$$

$$a_{ij} = \frac{b_{ij}}{\sum_{k=1}^{T_x} b_{ik}} \quad (3)$$

$$c_i = \sum_{i=1}^{T_x} \partial_{ij} s_j \quad (4)$$

$W_a$, $U_a$, $V_a$ indicate the weight matrix; $s_{i-1}^y$ is the output of the image side at the former time; and $s_j^x$ is the output of the language model from the source image.

### B. MULTILAYER DENSE ATTENTION MODEL

As shown in figure 2, for a given image, our multilayer dense attention model aims to extract multiple image features. Each image feature represents a significant region of the image. Spatial features can be generated using the bottom-up attention mechanism model; in other words, they are the spatial output layers of CNN. The multiple attention mechanism model is based on the artificial neural network (ANN). Specifically, it relies on various recurrent neural networks, long short-term memory, and attention mechanism. The overall architecture of the model is still based on the encoder-decoder structure, which is divided into two layers: bottom-up attention and top-down attention.

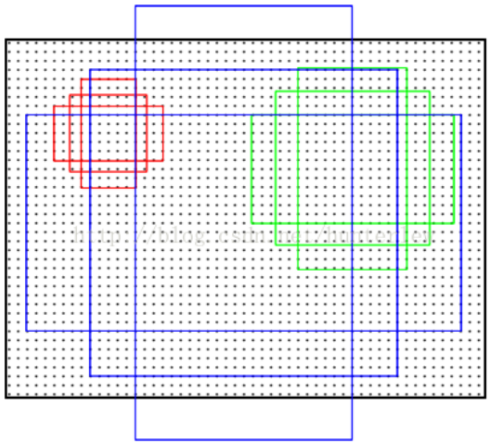

**FIGURE 3.** Instance of extracting anchor.

#### 1) BOTTOM-UP FEATURE EXTRACTION

The bottom-up mechanism is proposed to extract salient image regions. Each proposed candidate region is a pooled convolution feature vector. We use the Faster R-CNN to implement bottom-up attention.

Faster R-CNN detects objects through two processes. The first process, called region proposal network (RPN), predicts candidate regions of objects. In terms of convolution itself, it is equivalent to the sliding window. First, we use Resnet's last full convolution output to obtain a series of feature maps. Then, we need to determine whether the target exists in the corresponding receptive field in each sliding window center. It is necessary to determine whether the target exists in the corresponding receptive field in each sliding window center. Therefore, an anchor is proposed as the core of the RPN network. Because the target size and ratio of length to width are different, windows with multiple scales are needed. The anchor provides a benchmark window size, which is different in size according to multiple scales and the length-width ratio. In this way, anchors of several scales can be obtained, as shown in Figure3.

To pre-train the bottom-up attention mechanism to extract image feature models, we first use the pre-trained ResNet-101 [22] model on Imagenet. At the same time, to learn the representation of different attribute features, we add an additional training to predict the attribute classes, i.e., the pre-training of attribute relations on the visual genome dataset. To obtain the attribute of a region, we link the average pooling convolution feature with the embedding representation of the real object class, and then use it as an input of an additional layer. Additional layers are defined to determine the attribute and non-attribute categories. The loss function of the faster R-CNN consists of (5). We preserve these parts, and then add the additional loss of the multiclassification; that is, the loss of training attribute prediction.

$$L(\{p_i\}, \{u_i\}) = \frac{1}{N_{cls}} \sum_i L_{cls}(p_i, p_i^*) + \lambda \frac{1}{N_{reg}} \sum_i p_i^* L_{reg}(t_i, t_i^*) \quad (5)$$

where $p_i$ is the probability of anchor prediction being the target and $p_i^*$ is real class label. $t_i = \{t_x, t_y, t_w, t_h\}$ is a vector that denotes four parameterized coordinates of the predicted bounding box, $t_i^*$ is the coordinate vector of the ground-truth bounding box corresponding to a positive anchor, and $L_{cls}(p_i, p_i^*)$ indicates two classes, that is, logarithmic loss of target and non-target:

$$L_{cls}(p_i, p_i^*) = -log[p_i^* p_i + (1 - p_i^*)(1 - p_i)] \tag{6}$$

$L_{reg}(t_i, t_i^*)$ is the regression loss, used to compute $L_{reg}(t_i, t_i^*) = R(t_i - t_i^*)$, $R$ is the function of smooth $L_1$, and $p_i^* L_{reg}$ means that only the prospect anchor ($p_i^* = 1$) has regression losses. The output of the *cls* and *reg* layers comprises $\{p_i\}$ and $\{u_i\}$, which are normalized by $N_{cls}$, $N_{reg}$, and a balanced weight $\lambda$, respectively.

### 2) TOP-DOWN TEXT GENERATION

For a given set of image features $V$, we propose to derive the importance of each image feature according to the top-down attention mechanism, so as to generate the corresponding text in each time series. In the process of text generation, LSTM [23] decodes and generates the output sequence text as the generated content. This paper proposes a multilayer dense attention model, combining the attention LSTM with linguistic LSTM, which have achieved the best results in some evaluation indicators. The text generation model comprises three LSTM layers. In the following introduction, LSTM can be denoted as a single time step, such as (7)

$$h_t = LSTM(x_t, h_{t-1}) \tag{7}$$

where $x_t$ is the input vector of LSTM and $h_t$ is the output of LSTM. We build a three-layer LSTM, as shown in figure 2.

1) The input of the first LSTM consists of $h_{t-1}^2$, which is the output of the LSTM at the previous time, image feature $\bar{v} = \frac{1}{k}\sum_i v_i$ after average pooling and word embedding generated at the previous time, as shown in (8):

$$x_t^0 = [h_{t-1}^2, \bar{v}, W_e \prod_t] \tag{8}$$

where $W_e \in R^{E\times|\sum|}$ is the matrix for embedding dictionary words $\sum$ and $\prod_t$ is the one-hot codes of input words at time $t$. These inputs provide the LSTM at the first level with semantic information that needs to be addressed at the moment. Note that the word embedding matrix is learned when it is randomly initialized. The output $h_t^0$ of the first LSTM can be considered as the input of the next two layers of LSTM and the first attention layer. For K image features, each feature can be denoted as $v_i$, and a normalized attention weight $a_{i,t}$ can be obtained.

$$a_{i,t} = w_a^T tanh(W_{va} v_i + W_{ha} h_t^1) \tag{9}$$

$$a_t = softmax(a_t) \tag{10}$$

where $W_{va} \in R^{H\times V}$, $W_{ha} \in R^{H\times M}$, and $W_a \in R^H$ are the vector expressions obtained by learning. The attention image layer is a combination of all input featuresąêo

$$\hat{v}_t = \sum_{i=1}^{K} a_{i,t} vi \tag{11}$$

As can be seen from the above formula, its output can be regarded as an input of the next LSTM layer.

2) The second layer LSTM input comprises the output of the attention image layer and that of the first layer LSTM.

$$x_t^1 = [\hat{v}_t, h_t^0] \tag{12}$$

Similar to the first layer, its output serves as the input of the next layer of image attention and that of the next LSTM. The input of the third layer LSTM is composed of the output of the upper layer LSTM and that of the former two layers LSTM.

$$x_t^2 = [\hat{v}_t, f(h_t^0, h_t^1)] \tag{13}$$

where $f$ is a full connection network. The output of LSTM in this layer is regarded as the input of the next fully connected network.

The final output is obtained from the output of the first three layers of LSTM through a fully connected network, where $y_{1:T}$ represents the word sequence $(y_1, \ldots, y_T)$. At each time $t$, the conditional distribution of each possible output word is shown as follows:

$$p(y_t|y_{1:t-1}) = softmax(X_p h_t^2 + b_p) \tag{14}$$

$$Wp \in R^{|\sum|\times M} \tag{15}$$

where $b_p \in R^{|\sum|}$ is the weight learned. The complete distribution of the output sequence can be calculated using the conditional distribution of each output worąêo

$$p(y_{1:T}) = \prod_{t=1}^{T} p(y_t|y_{1:t-1}) \tag{16}$$

### 3) GRADIENT OPTIMIZATION METHOD

In terms of parameter optimization, most of the previous models used the method of optimizing the cross-entropy loss function to make the loss function descend to the gradient. However, there are two main problems in image caption. One is ''exposure bias,'' which is the problem of using different text decodings in training and prediction stages (Training uses real text decoding, while prediction uses predicted text decoding to predict the next word. Each word generated is based on the previous word. If the word produced at a certain time has errors, then the word produced at the next time will also be affected, resulting in the accumulation of errors). Another problem is to use the cross-entropy loss function training instead of using the language scoring criteria for the evaluation. The decreasing direction of the loss function of the degree descent method does not necessarily lead to a better direction of the evaluation index. In view of this, our

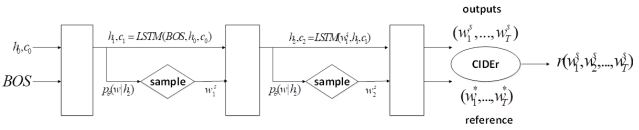


**FIGURE 4.** Optimization process.

improvement lies in making the training process as similar to the testing process as possible, that is, to evaluate the quality of the generated text means to use the text evaluation indicators such as bleu, CIDEr, and meteor. When training, the input at each moment uses the words generated at the previous moment, but because the interpretative text operation generated is not differentiable, it is impossible to perform reverse propagation. In recent years, reinforcement learning has been greatly developed, which can deal with these problems well and optimize the gradient non-differentiable problems in sequential learning. As shown in the figure 4 above, LSTM can be seen as an ''agent'' that interacts with external environments (such as text information and image features). The parameter $\theta$ of the network can be regarded as a policy parameter $p_\theta$, which can lead to generate the next word, also known as ''action.'' After each ''action'' occurs, this ''agent'' (LSTM) updates its network state (LSTM neurons, hidden layer states, and attention parameters). After the last tagged word (EOS) of the sequence is generated, the agent observes a ''reward,'' such as the score of CIDER of the whole sequence. We take this reward as $r$, and calculate the evaluation index from the real and generated sequences.

In the language model (LM), given the first several words of a sentence, it is expected that it can predict the next words. That is, the probability distribution of the possible occurrence of the first K + 1 words is given. *Perplexity* is used in the field of NLP to measure the difference between the predicted text and the real text. In information theory, perplexity is used to measure the degree to which a probability distribution or probability model predicts a sample. It is related to the maximum likelihood loss function in probability theory. *Perplexity* can also be used to compare two probability distributions or probability models. The probability distribution model or probability model with a low degree of confusion can better predict samples. It estimates the probability of a sentence based on each word and regularizes the length of the sentence; its formula is shown as follows:

$$\begin{aligned} PP(S) &= P(w_1 w_2 \dots w_N)^{-\frac{1}{N}} \\ &= \sqrt[N]{\frac{1}{P(w_1 w_2 \dots w_N)}} \\ &= \sqrt[N]{\prod_{i=1}^{N} \frac{1}{P(w_1 w_2 \dots w_N}} \end{aligned} \tag{17}$$

where S indicates sentence, N is the length of the sentence, and the larger $p(w_i)$ is, the higher is the probability we expect the sentence to have.

*Perplexity* can be regarded as an average branch factor, that is, how many choices can be made when predicting the next word, and the fewer the optional words, the more accurate is the model. The loss function is defined as the average puzzlement degree and regularization term of each word in a sentence.

$$\nabla_\theta L(\theta) \approx -(r(w^s) - b)\nabla_\theta log p_\theta(w^s) \tag{18}$$

where $b(\theta)$ is a valid baseline. The baseline $b$ is an arbitrary function as long as it does not depend on *action* $w^s$. Baseline does not change the expected gradient, but can change the variance of the gradient estimation to some extent. The expression of the final gradient, using the chain rule, can be given as

$$\nabla_\theta L(\theta) = \sum_{t=1}^{T} \frac{\partial L(\theta)}{\partial s_t} \frac{\partial s_t}{\partial \theta} \tag{19}$$

where $s_t$ is the input of *softmax*, use reinforcement learning and baseline $b$, and an approximate value of gradient $\frac{\partial L(\theta)}{\partial s_t}$ can be given as follows:

$$\frac{\partial L(\theta)}{\partial s_t} \approx (r(w^s) - b)(p_\theta(w_t|h_t) - l_{w_t^s}) \tag{20}$$

The captioning model is represented by a given real text sequence $y^*_{1:T}$ and a parameter $\theta$. We minimize the cross-entropy loss function as shown in the following formulas:

$$p(y_{1:T}) = \prod_{t=1}^{T} p(y_t|y_{1:t-1}) \tag{21}$$

$$L_{XE}(\theta) = -\sum_{t=1}^{T} log(p_\theta(y_t^*|y_{1:t-1}^*)) \tag{22}$$

We record the results of optimizing CIDEr and initialize it from the cross-entropy training model. Our goal is to minimize the negative expected score of the original equation:

$$L_R(\theta) = -E_{y_{1:T} \sim p_\theta}[r(y_{1:T})] \tag{23}$$

Here, $r$ is a fractional equation (such as CIDEr), and according to the method described in self-critical sequence training (SCST) [29], the gradient of the loss equation can be approximated by the following formula:

$$\nabla L_R(\theta) \approx -(r(y^s_{1:T}) - r(\hat{y}_{1:T}))\nabla_\theta log p_\theta(y^S_{1:T}) \tag{24}$$

where $y^S_{1:T}$ is a sampled caption and $r(y^s_{1:T})$ is defined as the baseline score by greedy decoding algorithms on current model. Optimizing the loss function is the goal of training, that is, maximizing the probability of generating sentences. To optimize the parameters in the model, the back-propagation method is generally used for training.

**TABLE 1. Configuration.**

| Devic | Resource | Configuration |
|---|---|---|
| Hardware | CPU | Intel Xeon E5-2620 2.40GHz |
| | Graphic Card | NVIDIA Tesla K80 |
| | OS | Ubuntu 16.04 LTS |
| Software | Language | Python |
| | Deep Learning Framework | Pytorch |
| | Machine Learning tools | scikit-learn (sklearn) |
| | NLP tools | NLTK |

## IV. EXPERIMENT

### A. CONFIGURATION AND DATASETS

The experiment environment configuration is shown in following table.

1) Microsoft COCO Caption Data Set [5]
The MSCOCO caption dataset includes 330,000 images and their corresponding 1.5 million text descriptions. On average, each picture corresponds to five text descriptions. The MSCOCO datasets are designed to collect data including those of multiple objects with scenarios.
2) Flickr dataset [24]
The Flickr dataset includes Flickr 8K and Flickr 30K. The image data source of the Flickr 30K dataset is Yahoo's photo album website. The numbers of images in the dataset are 8,000 and 31,783, respectively. Most of the images in the two databases show human participation in an activity. The corresponding manual labels of each image are five sentences. The two databases are collected and annotated in the same way, so the grammar of their annotations is similar. The database is also partitioned according to the standard training set and validation test set. The Flickr8K and Flickr30K datasets are the earliest datasets that have been used in image caption.
3) AI challenge Chinese dataset (Chinese_AI)
In AI challenge competition, an image description database is built, which is convenient for the participants to construct an image caption model. It includes the training dataset, which contains 210,000 images and their corresponding Chinese descriptions. The validation dataset includes 30,000 images and their corresponding Chinese descriptions. Each image corresponds to five Chinese descriptions with similar semantics. One sentence is used to describe the main information in a given image, which challenges image understanding in the Chinese context.

### B. EVALUATION INDICATOR

#### 1) BLEU

To evaluate the quality of the generated text, we adopt Bleu [25], a common criterion in machine translation. Bleu is reasonable for evaluating the quality of the generated text. Because the model proposed in this paper is image description, it is similar to machine translation, comparing the similarity between the generated text and the reference text and calculating a comprehensive score. The higher the score, the better will be the machine translation. We first calculate the accuracy of the modified n-gram followed by the geometric average length of N, and then consider it as the number of times; we finally multiply it by a penalty factor BP. The formula is as follows:

$$BP = min(1, e^{1-\frac{r}{c}}) \tag{25}$$

$$BLEU_N = BP \cdot e^{\frac{1}{N}\sum_{n=1}^{N} logp_n} \tag{26}$$

where $r$ is the length of the standard reference sentence and $c$ is the length of the generated sentence. The parameters $r$ and $c$ are calculated from the whole test dataset. If there are more than one reference sentences, we choose the length closest to the candidate sentence.

#### 2) METEOR

METEOR [26] measures are based on single-precision weighted harmonic mean and word recall rate. Its purpose is to solve some inherent defects in BLEU standards. METEOR also includes other indicators that do not find some other functions, such as synonym matching. Calculating METEOR requires a pre-defined set of alignment M. This calibration is based on WordNet's synonym library and is achieved by minimizing chunks in the corresponding statements. METEOR calculates the reconciliation average of accuracy and recall between the best candidate translation and the reference translation.

$$Pen = \gamma(\frac{ch}{m})^{\theta} \tag{27}$$

$$F_{mean} = \frac{P_m R_m}{\partial P_m + (1-\partial)R_m} \tag{28}$$

$$P_m = \frac{|m|}{\sum_k h_k(c_i)} \tag{29}$$

$$R_m = \frac{|m|}{\sum_k h_k(S_{ij})} \tag{30}$$

$$METEOR = (1 - Pen)F_{mean} \tag{31}$$

Among them, $\alpha$,$\gamma$ and $\theta$ are default parameters for evaluation. Therefore, the final evaluation of METEOR is a harmonic average based on block (chunk) decomposition matching and the representative decomposition matching quality, and contains a penalty coefficient Pen. Unlike BLEU, METEOR considers both accuracy and recall based on the whole corpus, and finally, obtains the measure.

#### 3) CIDER

The CIDEr [27] index regards each sentence as a ''document'' and expresses it in the form of a TF-IDF vector; it then calculates the cosine similarity between the reference caption and the caption generated by the model as a score. In other words, it is a vector space model. An image is $I_i \in I$ ($I$ denotes all sets of test sets). For each n-gram $W_k$ and reference caption $s_{ij}$, tf-idf can be calculated as

$$g_k(s_{ij}) = \frac{h_k(s_i j)}{\sum_{w_l \in \Omega} h_l(s_{ij})} log(\frac{|I|}{\sum_{I_p \in I} min\{1, \sum_q h_k(s_{pq})\}}) \tag{32}$$

**TABLE 2. Modes of ROUGE.**

| | |
|---|---|
| ROUGE-N | Co-occurrence Statistics Based on N-gram |
| ROUGE-L | Collinearity accuracy and recall statistics based on longest common clause |
| ROUGE-W | Cooccurrence accuracy and recall rate statistics of weighted longest public clauses |
| ROUGE-S | Cooccurrence accuracy and recall statistics of discontinuous binary groups |

The $\Omega$ in the formula is the vocabulary of all n-grams. The denominator part of IDF represents the number of pictures $W_k$ appearing in the reference caption.

$$F_{mean} = P_m R_m \partial P_m + (1 - \partial) R_m \tag{33}$$

Then, the value of CIDEr can be calculated using cosine similarity:

$$CIDEr_n(c_i, S_i) = \frac{1}{m} \sum_j \frac{g^n(c_i)^T g^n(s_{ij})}{\|g^n(c_i)\| \cdot \|g^n(s_{ij})\|} \tag{34}$$

Similar to BLEU,

$$CIDEr_n(c_i, S_i) = \sum_{n=1}^{N} \omega_n CIDEr_n(c_i, S_i) \tag{35}$$

#### 4) ROUGE

ROUGE [28](Recall-Oriented Understudy for Gisting Evaluation) is a similarity measurement method based on the recall rate, similar to BLEU. There is no evaluation function that mainly examines the adequacy and faithfulness of the translation and cannot evaluate the fluency of reference translation. It calculates the co-occurrence probability of N-gram in the reference translation and the translation to be evaluated. ROUGE includes the following four types:

ROUGN-N:

$$ROUGE - N = \frac{\sum_{S \in \{ReferencesSummaries\}} \sum_{gram_n \in S} Count_{match}(gram_n)}{\sum_{S \in \{ReferencesSummaries\}} \sum_{gram_n \in S} Count(gram_n)} \tag{36}$$

As shown in (42), n denotes n tuples and $Count_{match}(gram_n)$ is the maximum number of matched n-grams in the sentences to be evaluated. From the molecule, we can see that $ROUGE-N$ is a measure based on Recall. Definition: Longest Common Subsequence (LSC), suppose sequences $x = \{x_1, x_2, \ldots, x_m\}$ and $Y = \{y_1, y_2, y_3, \ldots, y_n\}$, if there is a strictly incremental sequence $\{i_1, i_2, \ldots, i_n\}$, which is the index of X, then Y is a subsequence of $X$ for each $j = 1, 2, \ldots, k$ and $x_{ij} = y_j$, and the common subsequence of the maximum length of sequences $X$ and $Y$ is called LCS. The F-measure based on $LCS(X, Y)$ evaluates the similarity between two sentences $X$ and $Y$. Suppose $X$ is the reference text and $Y$ is test text. Then, $F_{lcs}$ can be calculated asąêo

$$R_{lcs} = \frac{LCS(X, Y)}{m} \tag{37}$$

**TABLE 3. Division of standard dataset.**

| Dataset name | Training set | Validation set | test set |
|---|---|---|---|
| MSCOCO | 82783 | 40504 | 40775 |
| Chinese_AI | 220000 | 10000 | 10000 |
| Flickr8K | 6000 | 1000 | 1000 |
| Flickr30K | 28000 | 1000 | 1000 |

$$P_{lcs} = \frac{LCS(X, Y)}{N} \tag{38}$$

$$F_{lcs} = \frac{(1 + \beta^2) R_{lcs} P_{lcs}}{R_{lcs} + \beta^2 P_{lcs}} \tag{39}$$

where $m$, $n$ are the sequence lengths of $X$ and $Y$, respectively,

$$\beta = \frac{P_{lcs}}{R_{lcs}} \tag{40}$$

If $\beta$ is over high, then $R_{lcs}$ needs to be considered.

### C. EXPERIMENTAL RESULTS

The multilayer dense attention model is suitable for general datasets. This topic uses the Pytorch deep learning framework to implement the model, faster RCNN to extract the features, and the multiple-attention LSTM model to decode. Pytorch is the most popular deep learning framework at present. Using Pytorch can greatly improve the efficiency of building a network model. We have carried out validation experiments on Flickr dataset, MSCOCO. and Chinese dataset. The division of the training set, verification set, and test set is shown in Table 3. First, we preprocess the training text data and replace the low-frequency words with a special symbol UNK. Combining all words to create a large dictionary aims to facilitate the calculation of the word frequency of all words that appear, so as to facilitate the calculation of CIDEr after the occurrence. At the same time, we preprocess the text data for word segmentation, Jieba for Chinese dataset, Standford NLP for English data set, faster RCNN model for pre-training to extract salient regions of images, and fine-tune from the first round. All text data are stored in H5 format, which makes it easy to read the program quickly. For image preprocessing, the basic convolution neural network is Resnet101. We set the dimension of each word to 512. Because it takes considerable memory to load multiple graphs simultaneously, we set the batchsize to 16. We use the clip gradient to prevent gradient explosion. Our default optimization method is SGD, and the learning rate is set to $4e^{-4}$. We choose the CIDEr index to optimize the strategy gradient. We design several comparative experiments to compare with the latest papers. To select appropriate hyperparameters as the neurons of the circulating neural network, and to prove the effect of intensive learning training of CIDEr on the results, we design the following comparative experiments. From Table 4, it can be concluded that GRU, as a neuron of the circulating neural network, performs basically the same as LSTM on all evaluation indicators, and the performance of various indicators on the Chinese dataset is better than that on the MSCOCO dataset. On the Chinese dataset, we use the dictionary of the Chinese

**TABLE 4.** Effect on experiments with various RNN neurons.

| Dataset | RNN Type | Bleu | | Meter | Cider | Rouge |
|---|---|---|---|---|---|---|
| | | B-3 | B-4 | | | |
| MSCOCO | GRU | **52.6** | **35.2** | 23.2 | 118.4 | 55.2 |
| | LSTM | 51.0 | 34.0 | **24.8** | 118.3 | **56.5** |
| Chinese_AI | GRU | 66.4 | **62.5** | **43.2** | **211.6** | **71.9** |
| | LSTM | **68.3** | 60.3 | 42.6 | 211.5 | 71.7 |

**TABLE 5.** Comparison of optimizers.

| Dataset | Optimizer | Bleu | | Meter | CIDEr | Rouge |
|---|---|---|---|---|---|---|
| | | B-3 | B-4 | | | |
| MSCOCO | CIDEr Optimizer | **51.0** | **34.0** | **24.8** | **118.3** | **56.5** |
| | Cross Entropy Loss(CEL) | 50.80 | 32.7 | 23.2 | 107.5 | 55.8 |
| Chinese_AI | CIDEr Optimizer | **68.3** | **60.3** | **42.6** | **211.5** | **71.7** |
| | Cross Entropy Loss(CEL) | 66.2 | 58.9 | 40.8 | 200.5 | 69.2 |

training corpus as the lexicon of CIDEr calculation, which can help in improving the accuracy to some extent. GRU and LSTM can better solve the problem of long-term dependence. GRU, as its variant, has a simple structure and fast training speed. However, in this experiment, the qualities of the text produced by GRU and LSTM are similar.

As can be seen from table 5, the intensive learning training with CIDEr exhibits significant improvement on all indicators. It can be concluded that the obvious idea is to make the training and evaluation results as similar as possible. In training, instead of optimizing the maximum likelihood loss, direct maximization of CIDER (or BLEU, METEOR, and ROUGE) can effectively optimize the target evaluation indicators directly, thus avoiding the inconsistency between the evaluation indicators in training and test sets. The test process is as consistent as possible with the training process, even if the input of the previous moment is used instead of the known text input, which can avoid the accumulation of errors. At the same time, we adopt the text obtained by greedy decoding as the baseline model, so as to ensure that the reward in reinforcement learning proceeds in the normal direction of optimization.

Then, we use the hyperparameters adjusted to the best effect as our current model; that is, recurrent neurons adopt the LSTM basic structure, add CIDEr for reinforcement learning training, and compare with other state-of-art models with the same configuration. Our multilayer attention model is abbreviated as DenseAtt, as shown in table 6 and figure 5 below.

### D. ANALYSIS

The proposed multilayer dense attention mechanism shows good results on four datasets in the general domain; basically, all the indicators are better than the best image description model at present. Specifically, on the evaluation index of CIDEr, the multilayer dense attention model of this subject achieves state-of-art on each dataset. After the analysis, we use reinforcement learning to optimize the CIDEr index, which is proved to be effective experimentally.

In the process of image extraction, we adopt the bottom-up image extraction strategy, which can achieve better results.

**TABLE 6.** Performance comparison with state-of-the-art models.

| Dataset | Models | Bleu | | Meter | CIDEr | Rouge |
|---|---|---|---|---|---|---|
| | | B-3 | B-4 | | | |
| MSCOCO | SCST:Att2in[29] | - | 33.3 | 26.3 | 111.4 | 55.3 |
| | SCST:Att2all[29] | - | 34.2 | 26.7 | 114.0 | 55.7 |
| | Google NIC | 32.9 | 24.6 | 23.7 | 107.8 | 55.1 |
| | Attention NIC | 35.7 | 25.0 | 23.0 | 106.0 | 54.8 |
| | m-rnn | 35.0 | 25.0 | 23.0 | 105.9 | 52.8 |
| | DenseAtt(ours) | **51.0** | **34.0** | **24.8** | **118.3** | **56.5** |
| Chinese_AI | SCST:Att2in[29] | 58.9 | 55.3 | 33.7 | 109.2 | 67.2 |
| | SCST:Att2all[29] | 59.2 | 53.2 | 35.8 | 110.4 | 68.4 |
| | DenseAtt(ours) | **66.0** | **58.1** | **41.6** | **118.9** | **69.5** |
| Flickr8k | SCST:Att2in[29] | 34.6 | 30.2 | 22.9 | 159.2 | 45.2 |
| | SCST:Att2all[29] | 33.2 | 32.8 | 23.7 | 156.3 | 44.6 |
| | Google NIC | 27.0 | - | 20.4 | 152.6 | 45.8 |
| | Attention NIC | 31.4 | 21.3 | 19.8 | 155.9 | 47.2 |
| | DenseAtt(ours) | **47.7** | **33.4** | **23.1** | **167.5** | **46.9** |
| Flickr30k | SCST:Att2in[29] | 48.4 | 40.2 | 20.4 | 200.3 | 52.8 |
| | SCST:Att2all[29] | 47.6 | 41.5 | 21.8 | 201.4 | 52.9 |
| | M-RNN | 28.0 | 19.0 | 14.6 | 160.2 | 32.8 |
| | DenseAtt(ours) | **52.0** | **39.1** | **26.0** | **209.1** | **51.0** |

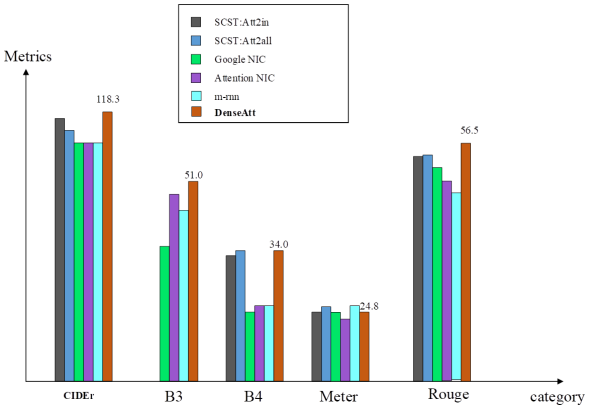


**FIGURE 5.** Comparison of MSCOCO models.

Because the early stage of image feature extraction directly affects the generation of the corresponding text, we should avoid the influence of background information as much as possible. At the same time, in the process of image extraction, we use pre-trained convolution neural network parameters to speed up the whole training process.

In the process of generating the corresponding text, that is, in the decoding process, we adopt multiple top-down attention mechanisms to help us better combine the extracted image information, decode the image according to the salient image information, and eliminate the influence of irrelevant information on the experiment.

At the same time, we use the strategy gradient optimization in reinforcement learning to directly optimize CIDEr, and effectively solve the problem of asymmetric information between the training and test sets. We use greedy decoding to obtain the baseline model, and then obtain the reward to optimize the parameters.

## V. CONCLUSION

We propose a multilayer dense attention model for image captioning task. A faster RCNN is employed to extract the image features as the coding layer, LSTM-Attend is used to decode the multilayer dense attention model, and the description text is generated. The experimental results show that our model has a good ability of image understanding and text generation.

## ACKNOWLEDGMENT

The authors thank the reviewers for their comments.

## REFERENCES

[1] W. Gan, J. C.-W. Lin, P. Fournier-Viger, H.-C. Chao, and P. S. Yu, "HUOPM: High-utility occupancy pattern mining," *IEEE Trans. Cybern.*, to be published.
[2] J. C.-W. Lin, Y. Zhang, B. Zhang, P. Fournier-Viger, and Y. Djenouri, "Hiding sensitive itemsets with multiple objective optimization," *Soft Comput.*, pp. 1–19, Feb. 2019.
[3] J.-S. Pan, L. Kong, T.-W. Sung, P.-W. Tsai, and V. Snášel, "$\alpha$-fraction first strategy for hierarchical model in wireless sensor networks," *J. Internet Technol.*, vol. 19, no. 6, pp. 1717–1726, 2018.
[4] J. Guan and E. Wang, "Repeated review based image captioning for image evidence review," *Signal Process., Image Commun.*, vol. 63, pp. 141–148, Apr. 2018.
[5] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, "Show and tell: Lessons learned from the 2015 MSCOCO image captioning challenge," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 4, pp. 652–663, Apr. 2017.
[6] A. Karpathy and L. Fei-Fei, "Deep visual-semantic alignments for generating image descriptions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 3128–3137.
[7] K. Xu *et al.*, "Show, attend and tell: Neural image caption generation with visual attention," *Comput. Sci.*, vol. 2015, pp. 2048–2057, Feb. 2015.
[8] P. Anderson, "Bottom-up and top-down attention for image captioning and visual question answering," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 6077–6086.
[9] N. Li and Z. Chen, "Image captioning with visual-semantic LSTM," in *Proc. IJCAI*, 2018, pp. 793–799.
[10] H. Fang *et al.*, "From captions to visual concepts and back," in *Proc. CVPR*, Jun. 2015, pp. 1473–1482.
[11] G. Kulkarni *et al.*, "Baby talk: Understanding and generating simple image descriptions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2011, pp. 1601–1608.
[12] J. Lu, J. Yang, D. Batra, and D. Parikh, "Neural baby talk," in *Proc. CVPR*, Jun. 2018, pp. 7219–7228.
[13] R. Kiros, R. Salakhutdinov, and R. Zemel, "Multimodal neural language models," in *Proc. 31st Int. Conf. Mach. Learn. (ICML)*, 2014, pp. 595–603.
[14] Y. Wang, Z. Lin, X. Shen, S. Cohen, and G. W. Cottrell, "Skeleton key: Image captioning by skeleton-attribute decomposition," in *Proc. CVPR*, Jun. 2017, pp. 7272–7281.
[15] A. Mathews, L. Xie, and X. He, "SemStyle: Learning to generate stylised image captions using unaligned text," in *Proc. CVPR*, Jun. 2018, pp. 8591–8600.
[16] J. Aneja, A. Deshpande, and A. G. Schwing, "Convolutional image captioning," in *Proc. CVPR*, Jun. 2018, pp. 5561–5570.
[17] J. Gu, J. Cai, G. Wang, and T. Chen, "Stack-captioning: Coarse-to-fine learning for image captioning," in *Proc. 32nd AAAI Conf. Artif. Intell.*, 2018, pp. 1–9.
[18] T. Yao, Y. Pan, Y. Li, and T. Mei, "Incorporating copying mechanism in image captioning for learning novel objects," in *Proc. CVPR*, Jul. 2017, pp. 5263–5271.
[19] D. Lu, S. Whitehead, L. Huang, H. Ji, and S.-F. Chang, "Entity-aware image caption generation," in *Proc. EMNLP*, 2018, pp. 4013–4023.
[20] Q. Wu, C. Shen, L. Liu, A. Dick, and A. van den Hengel, "What value do explicit high level concepts have in vision to language problems?" in *Proc. CVPR*, Las Vegas, NV, USA, Jun. 2016, pp. 203–212.
[21] F. Liu, X. Ren, Y. Liu, H. Wang, and X. Sun, "simNet: Stepwise image-topic merging network for generating detailed and comprehensive image captions," in *Proc. EMNLP*, 2018, pp. 137–149.
[22] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. CVPR*, Jun. 2016, pp. 770–778.
[23] F. A. Gers and J. Schmidhuber, "LSTM recurrent networks learn simple context-free and context-sensitive languages," *IEEE Trans. Neural Netw.*, vol. 12, no. 6, pp. 1333–1340, Nov. 2001.
[24] N. H. Phan, V. D. T. Hoang, and H. Shin, "Adaptive combination of tag and link-based user similarity in flickr," in *Proc. 18th ACM Int. Conf. Multimedia*, 2010, pp. 675–678.
[25] K. Papineni *et al.*, "BLEU: A method for automatic evaluation of machine translation," in *Proc. Annu. Meeting Assoc. Comput. Linguistics*, 2002, pp. 311–318.
[26] M. Denkowski and A. Lavie, "Meteor 1.3: Automatic metric for reliable optimization and evaluation of machine translation systems," in *Proc. 6th Workshop Stat. Mach. Transl.*, 2011, pp. 85–91.
[27] R. Vedantam, C. L. Zitnick, and D. Parikh, "CIDEr: Consensus-based image description evaluation," in *Proc. CVPR*, Jun. 2015, pp. 4566–4575.
[28] C.-Y. Lin, "ROUGE: A package for automatic evaluation of summaries," in *Proc. ACL-Workshop, Text Summarization Branches Out*, 2004, pp. 1–8.
[29] S. J. Rennie, E. Marcheret, Y. Mroueh, J. Ross, and V. Goel, "Self-critical sequence training for image captioning," in *Proc. CVPR*, Jul. 2017, pp. 1179–1195.
[30] C.-M. Chen, B. Xiang, Y. Liu, and K. H. Wang, "A secure authentication protocol for Internet of vehicles," *IEEE Access*, vol. 7, pp. 12047–12057, 2019.

**ERIC KE WANG** received the Ph.D. degree from the Department of Computer Science, The University of Hong Kong, in 2009. He is currently an Associate Professor with the Harbin Institute of Technology (HIT), China. He is also a Senior Researcher with the Key Laboratory of Shenzhen Internet Information Collaborative Technology and Application, HIT. His main research interests include machine learning and data security. He has received two granted projects from the National Science Funding (NSFC) of China. Besides, he has developed two software platforms for deep learning tools and received two authorized related patents.

**XUN ZHANG** received the bachelor's degree from Northeastern University, China. She is currently pursuing the master's degree with the Harbin Institute of Technology (HIT), China. Her research interests involve machine learning and signal processing by deep learning.

**FAN WANG** received the bachelor's degree from the Harbin Institute of Technology, China, where she is currently pursuing the master's degree. Her research interests include deep learning and information hiding.

**TSU-YANG WU** received the Ph.D. degree from the Department of Mathematics, National Changhua University of Education, Taiwan, in 2010. He was an Assistant Professor with the Innovative Information Industry Research Center, Shenzhen Graduate School, Harbin Institute of Technology. He is currently an Associate Professor with the College of Computer Science and Engineering, Shandong University of Science and Technology, China. His research interests include cryptography and network security, and machine learning. He serves as an Executive Editor of the *Journal of Network Intelligence* and as Associate Editor of *Data Science* and *Pattern Recognition*.

**CHIEN-MING CHEN** received the Ph.D. degree from the National Tsing Hua University, Taiwan. He is currently an Associate Professor with the College of Computer Science and Technology, Shandong University of Science and Technology, Shandong, China. His current research interests include network security, privacy, and machine learning.