

Article

Boosting Memory with a Persistent Memory Mechanism for Remote Sensing Image Captioning

Kun Fu ^{1,2,3,4,5,†} , Yang Li ^{2,3,4,*}, Wenkai Zhang ^{1,4}, Hongfeng Yu ^{1,4} and Xian Sun ^{1,3,4}

¹ Aerospace Information Research Institute, Chinese Academy of Sciences, Beijing 100190, China; fukun@mail.ie.ac.cn (K.F.); zhangwk@aircas.ac.cn (W.Z.); hfyu@mail.ie.ac.cn (H.Y.); sunxian@mail.ie.ac.cn (X.S.)

² School of Microelectronics, University of Chinese Academy of Sciences, Beijing 100190, China

³ School of Electronic, Electrical and Communication Engineering, University of Chinese Academy of Sciences, Beijing 100190, China

⁴ Key Laboratory of Network Information System Technology, Institute of Electronics, Chinese Academy of Sciences, Beijing 100190, China

⁵ Institute of Electronics, Chinese Academy of Sciences, Suzhou 215000, China

* Correspondence: liyang182@mails.ucas.ac.cn; Tel.: +86-10-5888-7208

† These authors contributed equally to this work.

Received: 9 April 2020; Accepted: 6 June 2020; Published: 9 June 2020



Abstract: The encoder–decoder framework has been widely used in the remote sensing image captioning task. When we need to extract remote sensing images containing specific characteristics from the described sentences for research, rich sentences can improve the final extraction results. However, the Long Short-Term Memory (LSTM) network used in decoders still loses some information in the picture over time when the generated caption is long. In this paper, we present a new model component named the Persistent Memory Mechanism (PMM), which can expand the information storage capacity of LSTM with an external memory. The external memory is a memory matrix with a predetermined size. It can store all the hidden layer vectors of LSTM before the current time step. Thus, our method can effectively solve the above problem. At each time step, the PMM searches previous information related to the input information at the current time from the external memory. Then the PMM will process the captured long-term information and predict the next word with the current information. In addition, it updates its memory with the input information. This method can pick up the long-term information missed from the LSTM but useful to the caption generation. By applying this method to image captioning, our CIDEr scores on datasets UCM-Captions, Sydney-Captions, and RSICD increased by 3%, 5%, and 7%, respectively.

Keywords: image caption, remote sensing, long short-term memory, persistent memory mechanism

1. Introduction

Different from other remote sensing tasks in the vision field, such as object detection [1] or semantic segmentation [2], the remote sensing image caption task [3] involves generating a sentence that describes the image accurately and comprehensively. In addition, remote sensing images increase the difficulty of image captioning due to their wide coverage and long distance from shooting sites.

Many models are based on an end-to-end encoder–decoder framework [4,5] where a convolutional neural network (CNN) [6] extracts the image features and a Recurrent Neural Network (RNN) generates a caption with the features. However, in some cases, the RNN will have the problem of long-term dependence on information and the bad memory effect of early information. For the vanishing gradient problem of the RNN, the earliest solution method [7] is to replace the RNN with Long Short-Term Memory (LSTM). The LSTM network can add and remove information from the cell

through the gated unit. However, it is hard for LSTM to store specific facts accurately in the image caption task. Because of the forget gate layer, the LSTM will overwrite information that is irrelevant to the current time. If the model needs the overwritten information to predict the next word, the model cannot obtain it. In this case, the information unrelated to the current time can help to predict the next word. Therefore, it is necessary to pick up and utilize this useful memory information when necessary to produce a comprehensive and accurate sentence.

In order to overcome this limitation, adding an external storage structure to the LSTM network is a new perspective. Graves et al. [8] firstly propose a Neural Turing Machine and apply it to a copy task and associative task. They extend the capabilities of neural networks by coupling them to external memory resources, which they can interact with attentional processes. Preliminary results demonstrate that Neural Turing Machines can infer simple algorithms such as copying, sorting, and associative recall from input and output examples. Another work [9] implements differentiable function. It takes this idea and applies it to a complex text reasoning task. Chunseong et al. [10] use the memory as a context repository of prior knowledge for personalized image captioning. Taking inspiration from [8] and [11], we can generate more comprehensive sentences by giving the network more information with external storage in a remote sensing image caption task.

In this paper, we mainly propose a new model component named the Persistent Memory Mechanism (PMM) as shown in Figure 1. The baseline model architecture is illustrated on the left of Figure 1. The improved model framework we propose is shown on the right of Figure 1. Our new method is based on the encoder–decoder framework. The entire module component can be trained end-to-end. The advantage of the PMM is that the model can extract useful information related to the predictions at the current time from the external memory of stored information. In addition, the external memory is a memory matrix that uses a combined addressing system of content and location-based addressing. It can store all of the hidden layer vectors of LSTM before the current time step. In conclusion, our main contributions of this paper are as follows:

1. We firstly present a novel model component named the Persistent Memory Mechanism (PMM) for the remote sensing image captioning task. This new method gives a new perspective to better solve the image caption task in the remote sensing domain.
2. We apply the PMM to the image caption model. The PMM takes the form of external memory storage so that it can extend the LSTM naturally with vector states and capture the information lost from the LSTM but useful for caption generation.
3. We verify the new model on three publicly available remote sensing datasets including UCM-Captions, Sydney-Captions, and RSICD. In particular, the results show that the CIDEr scores increase by 3%, 5%, and 7%, respectively.

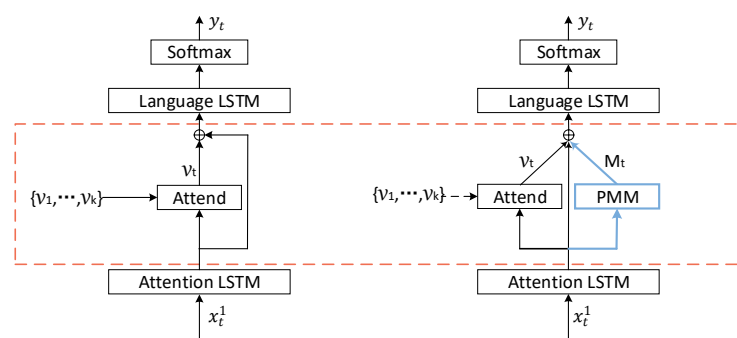


Figure 1. On the left is the framework of the baseline model [11] and on the right is the framework with the Persistent Memory Mechanism (PMM). LSTM stands for Long Short-Term Memory.

2. Related Works

2.1. Development of Natural Image Captioning

Many different models have been proposed for image captioning with the rapid development of computer vision and natural language processing. They can be divided into two categories in general: template-based methods [12–16] and neural-based methods [17–19].

The template-based method is the earliest method for image caption. This method [12–14] needs to generate different caption templates for images. Then it fills in the blanks with the outputs of objection, attribute classification, and scene recognition. However, these methods use fixed templates to generate captions. These monotonous captions cannot meet people's needs. Some works [15,16] have improved the templates with techniques such as adopting more powerful language templates, and so on. However, these methods have limitations. The types of sentence templates are limited, and the lengths of sentence templates are not variable. What is more, template-based methods cannot be trained end-to-end.

The neural-based method adopts encoder–decoder frameworks, which are widely used in image captioning tasks. This framework [17–19] is introduced from machine translation. Generally, the encoder uses a CNN such as VGG [6] or ResNet [20] to extract image features. The decoder uses a network such as RNN and LSTM to generate captions. The earliest image caption model [21,22] uses a feed-forward neural network in the decoder. Then, some other methods [23,24] firstly use a recurrent neural network (RNN) instead of the feed-forward neural network. However, this brings the problem of gradient dissipation. To avoid this problem, Vinyals et al. [7] try to use an LSTM instead of the RNN. They propose a neural image captioning (NIC) model to generate sentences for describing natural images. This great work can solve this problem by learning to control the input and hidden state. Recently, more works [25,26] have introduced attention mechanisms. These methods can adaptively choose image features or word features to help LSTM predict the next word. At each time step, these attention models will focus on different regions in images and give different weights to image features. Attention mechanisms are widely used in encoders to extract image features or in decoders to generate sentences.

2.2. Development of Remote Sensing Image Captioning

Remote sensing image captioning has been gradually studied by people with the development of natural image captioning. However, there are few studies on remote sensing image captioning due to the lack of sufficient datasets and the special characteristics of remote-sensing images.

Qu et al. [27] propose a multimodal neural network model with an encoder–decoder framework for semantic understanding of high resolution remote-sensing images. This method adopts a CNN to extract image features, and the image features are then fused with the hidden state to predict the word step by step. Finally, the model connects all of the predicted words to obtain the final sentence. In addition, this is the earliest work in which the neural network model is applied to a remote sensing image caption task. [28] find that the image-caption methods for natural images can be transferred to remote-sensing images. They adopt an encoder–decoder model based on the attention mechanism for remote sensing image captioning. What is more, they propose a new, large remote-sensing dataset named RSICD to fully advance the task of remote-sensing captioning. More people have begun to study other methods that can be applied to remote sensing image captioning. For example, Zhang et al. [29] propose a new label-attention mechanism in the encoder part. Then the image features can be added by attention masks to improve the salience of key regions in images. Finally, sentences are generated by decoders with the help of context vectors. These works all improve the image feature extraction part, but they ignore the importance of the decoder, which is used for sentence generation. However, as far as we know, no one has improved the remote sensing image-captioning task in the decoder section by adding an external memory database.

3. Method

The overall framework still adopts the mainstream encoder–decoder architecture as shown in Figure 2. The RNN firstly completes some processing based on the global image features and the word predicted at the previous moment. Then the processed result is taken as the input vector of the PMM as the serial number 1 in Figure 2 shows. The PMM will search the relevant storage information from its own storage memory database based on this input information, as the serial number 2 in Figure 2 shows. The searching result and the input information at the current time are sent to a softmax layer as the output vector of the PMM, as serial numbers 3 and 4 in Figure 2 show. At the same time, the PMM will update its own storage memory with the input information, as serial number 5 in Figure 2 shows. In addition, the storage memory of the PMM is a vector matrix that is updated at each time step, as the orange rectangle in Figure 2 shows. Finally, the RNN will generate the next word according to the local image features and the output of the PMM. Model applying the PMM can be trained end-to-end. More details will be elaborated on in the following sections.

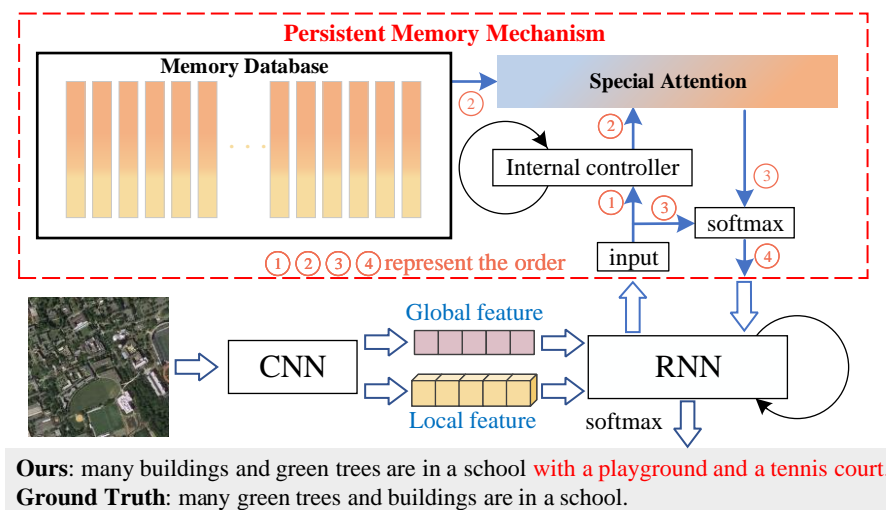


Figure 2. Diagram of the proposed framework. The data processed by the recurrent neural network (RNN) is fed into the PMM. Then the PMM outputs the corresponding information found in its external memory in a numerical order. Meanwhile, the memory database is updated to facilitate the search at the next time step.

In Section 3.1, we firstly describe the Neural Image Captioning model proposed in [11], which uses an encoder–decoder architecture with region-based attention. Then we apply the Persistent Memory Mechanism (PMM) to it in Section 3.2.

3.1. Encoder–Decoder Model for Image Captioning

Given an image I and the ground truth sequence $y = \{y_1, \dots, y_t\}$, the purpose of the encoder–decoder model is to maximize the following objectives:

$$\hat{\theta} = \arg \max_{\theta} \sum_{(I,y)} \log P(y|I; \theta), \tag{1}$$

where θ are model parameters. Using the chain rule, we can expand the log likelihood of the joint probability distribution. According to the attended image features $i_{1:T}$ and each sequence $y_{1:T}$, we can obtain the following formula:

$$\log p(\mathbf{y}) = \sum_{t=1}^T \log p(y_t | y_{1:t-1}, i_t, \mathbf{I}). \quad (2)$$

In this framework, each conditional probability with the recurrent neural network is modeled as:

$$\log p(y_t | y_{1:t-1}, i_t, \mathbf{I}) = f(\mathbf{h}_t, \mathbf{v}_t), \quad (3)$$

where f is a nonlinear function that outputs the probability of y_t , and \mathbf{v}_t are the visual context vectors at time t extracted from image \mathbf{I} . \mathbf{h}_t are the output vectors of the RNN at time t . In this paper, we adopt the captioning model, which is composed of two LSTM layers using a standard implementation. Here, we have neglected the propagation of memory cells for notational convenience. \mathbf{h}_t is modeled as the following notation at every single time step:

$$\mathbf{h}_t = LSTM(\mathbf{x}_t, \mathbf{h}_{t-1}), \quad (4)$$

where \mathbf{x}_t are the input vectors of LSTM. In our model, the attention LSTM understands the general content of the image. The language LSTM uses the image features and information from the PMM to generate the final description. We use a convolutional neural network to extract global image features and attended image features, respectively, in our experiments first. Then these features are fed into the decoder for training. In order to persistently store the memory information of the previous moment to guide the LSTM to predict some words, we introduce the persistent memory mechanism.

3.2. Model with Persistent Memory Mechanism

After we extract the local image features $\mathbf{v}_t = (\mathbf{v}_1, \dots, \mathbf{v}_k)$, $\mathbf{v}_i \in R^{2048}$ from the ResNet as the attended image features, we can naturally get the global image features

$$\mathbf{v}_g = \frac{1}{k} \sum_{i=1}^k \mathbf{v}_i. \quad (5)$$

In order to make full use of context information, the input vector for the first LSTM consists of three parts. It contains the output of the second LSTM at the previous moment, the global image features of the input image, and the word encoding generated at the previous moment. It can be expressed as follows:

$$\mathbf{x}_t^1 = \text{concatenate}(\mathbf{h}_{t-1}^2, \mathbf{v}_g, y_{t-1}). \quad (6)$$

In order to generate a detailed and accurate description of the current moment, the input vector of the second LSTM also includes three parts. It contains the output of the first LSTM, the attention features of the input image, and the extracted vector that best matches the stored memory information at the current moment. It can be expressed as follows:

$$\mathbf{x}_t^2 = \text{concatenate}(\mathbf{h}_t^1, \delta_v, M_t). \quad (7)$$

3.2.1. Generation of δ_v

Given the hidden state of the first LSTM, we can generate the attention distribution ε_t for each of the attended image features \mathbf{v}_t at each time step t as follows:

$$\varepsilon_t = \mathbf{w}_d^T \tanh(W_v \mathbf{v}_t + W_h \mathbf{h}_t^1), \quad (8)$$

$$\alpha_t = \text{softmax}(\varepsilon_t), \quad (9)$$

where $W_v, W_h \in R^{k \times d}$ and $w_d \in R^k$ are learnable model parameters. The attended image feature used as input to the language LSTM is finally represented as a convex combination of all input features:

$$\delta_v = \sum_{i=1}^k \alpha_{ti} \mathbf{v}_{ti}. \tag{10}$$

3.2.2. Generation of M_t

In this section, we will describe the PMM module in detail. The component block diagram is shown in Figure 3. The part in the blue box is a separate component. The controller interacts with the external information via an input and output vector, which can be a recurrent controller or a feed-forward controller. Here, we choose LSTM as the controller. LSTM has a better ability to deal with the long-term dependencies in the sequence, thus learning better about how to interact with external memory.

Firstly, the external input vector, namely the output \mathbf{h}_t^1 of the attention LSTM, passes through another LSTM that serves as the control information interaction. Then the information related to the current moment content is obtained from the external memory under the direction of the controller. The searching result and the input information at the current time are sent into a softmax layer as the output vector of the PMM. Finally, the PMM will update its memory through the input information of the current moment. The process of searching the memory database can be understood as using a soft-attention mechanism to obtain the stored memory information according to the learned searching parameters. The process of updating the memory database can be understood as adding or deleting some information from the memory database according to the information at the current moment. We will look at this component in more detail next.

The external input vector \mathbf{h}_t^1 and the previous searching state S_{t-1} of the component are taken as the input vector \mathbf{x}_t^3 of the current controller:

$$\mathbf{x}_t^3 = \text{concatenate}(\mathbf{h}_t^1, S_{t-1}). \tag{11}$$

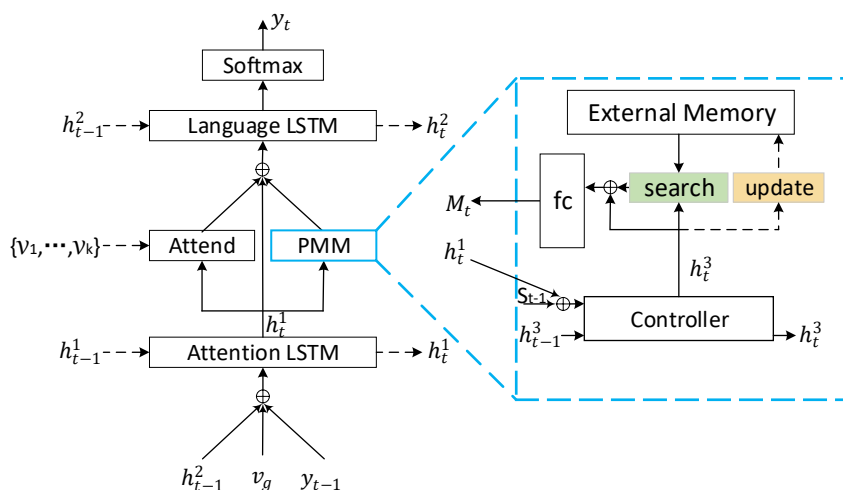


Figure 3. Detailed structure of the model PMM. The green box represents searching the corresponding memory information. The yellow box represents updating the database.

Taking inspiration from [8], we use the combined addressing system of content and location-based addressing to get the important searching parameter ω_t . Then we use this parameter to search the best information from the memory database and update the memory database. At this moment, the output vector \mathbf{h}_t^3 of the controller contains the relationship between the current moment information and the memory database. Therefore, the process of getting the parameter ω_t is very important.

Now we will introduce how to obtain the parameter ω_t . We take part of \mathbf{h}_t^3 (denoted as K_t) as the reference vector of the current moment. Then we use it to search the memory information that matches K_t in the memory database (denoted as D_t) to obtain the weight θ_t as follows:

$$\theta_t = \text{softmax}(\mu \cdot \varphi[K_t, D_t]), \quad (12)$$

where μ is a model parameter that can amplify or attenuate the precision of φ . While φ is a function of similarity measured by cosine similarity. The function φ scores how well the input vector \mathbf{h}_t^3 matches the memory database D_t . We can retrieve roughly the desired stored information at this point in the process. It can help the PMM to know the vector to be stored later. Then, to make full use of the relationship between the current time information and the previous time information, we introduce a parameter g_t to guide the next update of parameter θ_t . The advantage of this formula is that the new weight parameter θ'_t can be generated according to the degree of correlation between the information at the current time and the parameters ω_{t-1} (the important parameter at the previous time step), θ_t . Then, we obtain the final weight parameter ω_t :

$$\theta'_t = g_t \theta_t + (1 - g_t) \omega_{t-1}, \quad (13)$$

$$\omega_t = \text{softmax}(\theta'_t). \quad (14)$$

With this parameter ω_t , we can obtain the memory vector S_t that matches from the memory database. Then we take this memory vector S_t together with the input vector \mathbf{h}_t^1 as the output M_t of the memory component:

$$S_t = \sum_{i=1}^{i=n} \omega_t(i) D_t(i), \quad (15)$$

$$M_t = \text{concatenate}(S_t, \mathbf{h}_t^1). \quad (16)$$

At the same time, with this parameter ω_t and the input vector \mathbf{h}_t^1 at the current time, we can update the data D_t in the memory database as follows:

$$D_t = D_{t-1} - \omega_t D^{del} + \omega_t D^{add}, \quad (17)$$

where D^{del} and D^{add} are two different parts of the input vector \mathbf{h}_t^1 . In addition, all of the above operations are differentiable, so our method can be trained end-to-end.

4. Experiments

Based on the baseline model [11], our experiment is implemented by adding the PMM component and adopting a specific feature extraction method for remote-sensing images. In this section, we will briefly introduce the datasets and evaluation metrics. Then we will show our experimental details. Finally, we will give the experimental comparison results and analysis.

4.1. Dataset

In our experiment, we used three public remote-sensing datasets, namely the UCM-Captions dataset [30], the Sydney-Captions dataset [31], and the RSICD dataset [28]. Each dataset is manually tagged with five descriptive sentences.

There are 2100 images in the UCM-Captions (<https://pan.baidu.com/s/1mjPTToHq#list/path=%2F>) dataset, including 21 classes: agricultural, airplane, baseball diamond, beach, buildings, chaparral, dense residential, forest, freeway, golf course, harbor, intersection, medium residential, mobile home

park, overpass, parking lot, river, runway, sparse residential, storage tanks, and tennis court. Each image has a size of 256×256 pixels.

There are 613 images in the Sydney-Captions (<https://pan.baidu.com/s/1hujEmcG#list/path=%2F>) dataset, including 7 classes: residential, airport, meadow, river, ocean, industrial, and runway. Each image has a size of 500×500 pixels.

There are 10,921 images in the RSICD (<https://pan.baidu.com/s/1bp71tE3#list/path=%2F>) dataset, including 30 types of scenes: airport, bridge, beach, baseball field, open land, commercial, center, church, desert, dense residential, forest, farmland, industrial, mountain, medium residential, meadow, port, pond, parking, park, playground, river, railway station, resort, storage tanks, stadium, sparse residential, square, school and viaduct. Each image has a size of 224×224 pixels.

4.2. Evaluation Metrics

In this paper, we report all of the results of our experiments using Microsoft COCO caption evaluation tools, including BLEU-n [32], Meteor [33], Rouge-L [34], CIDEr [35], and SPICE [36]. All metrics are computed with the publicly released code (<https://github.com/ruotianluo/coco-caption/tree/ea20010419a955fed9882f9dcc53f2dc1ac65092/pycocoevalcap>). BLEU and Meteor are commonly used in machine translation of short sentences. In this paper, we take the value of n to be 4, as usual. Rouge-L is a measure based on the accuracy of co-occurrence and recall of the longest common clause. CIDEr and SPICE are important indicators of image description. CIDEr is used to measure the consistency between the description generated by the model and the truth value. SPICE is used to calculate the F-score of matching tuples between the predicted and reference scene graphs generated by captions, and this new metric is found to better correlate with human judgments.

4.3. Implementation Details

4.3.1. Encoder

In the encoder, we directly adopt ResNet-101, which is pre-trained on the ImageNet dataset [37] for image feature extraction. In addition, in order to get as close to the original image information as possible, we directly use the features after ResNet-101 as the input image features. As a result, we use the feature extracted from the last convolutional layer as the attended feature $\mathbf{v}_i = \{\mathbf{v}_1, \dots, \mathbf{v}_k\}$, $\mathbf{v}_i \in R^{2048}$. For the UCM-Captions, Sydney-Captions, and RSICD remote-sensing datasets, the dimension of the attended feature are $2048 \times 8 \times 8$, $2048 \times 16 \times 16$, and $2048 \times 7 \times 7$, respectively. Then we reshape them to 64×2048 , 256×2048 , and 49×2048 (k is 64 or 256 or 49 in the formula above), respectively.

For text descriptions, we remove words that occur less than five times in the text vocabulary of each dataset. The three text terms are then mapped to the dimension of 512, respectively, as the text input of the decoder.

4.3.2. Decoder

In the decoder, we set the size of the RNN to 512 to represent the number of hidden nodes per layer, and we use an RNN with a single layer of LSTM units. During the model training, both the vocabulary and the image feature fed in the decoder have an encoding size of 512. In addition, we use the Adam stochastic gradient descent algorithm with alpha 0.9 and beta 0.999. The initial learning rate is $5e-4$ for the decoder model.

4.4. Results of Experiments

We divide the data set into three parts: 80% training set, 10% verification set, and 10% testing set. In our experiments, we processed the remote-sensing images with ResNet-101. Then we get the features after the final convolutional layer, and then use an adaptive pooling method to extract attended features as in [11] as our baseline model (hereinafter referred to as UpDown). We also directly

use the features after the ResNet-101 as the attended local features (hereinafter referred to as DF) for comparison. As for our proposed model (Persistent Memory Mechanism), it is listed separately as a model component (hereinafter referred to as PMM). The symbol “+” represents the model used. To make a fair comparison, we adopt the results for models trained with standard cross-entropy loss. In addition, we also use several models [26,38,39] (hereinafter referred to as SAT, Att2in, SM-Att) for comparison. In Tables 1–3, we present our experimental results on the above three remote-sensing datasets. The bold numbers stand for the best scores.

Table 1. Comparison of experimental results for UCM-Captions.

Methods	BLEU-1	BLEU-4	METEOR	ROUGE-L	CIDEr	SPICE
SAT [26]	83.0	65.2	43.9	78.5	335.1	50.0
Att2in [38]	84.0	68.1	46.5	80.2	354.0	52.3
SM-Att [39]	81.1	63.0	43.5	77.9	338.6	48.8
UpDown [11]	84.7	69.0	46.9	81.3	355.3	53.3
UpDown+PMM	85.2	68.0	45.8	81.1	352.9	50.9
UpDown+DF	85.7	70.0	46.9	81.4	357.7	52.8
UpDown+DF+PMM	86.2	71.2	48.2	82.5	365.4	53.6

Table 2. Comparison of experimental results for Sydney-Captions.

Methods	BLEU-1	BLEU-4	METEOR	ROUGE-L	CIDEr	SPICE
SAT [26]	78.8	52.4	40.7	71.9	217.4	41.0
Att2in [38]	80.1	54.6	40.1	70.5	225.2	41.8
SM-Att [39]	74.3	51.8	36.4	67.7	234.0	39.8
UpDown [11]	81.5	55.3	40.3	71.9	225.2	40.2
UpDown+PMM	81.2	55.4	40.8	72.3	229.3	41.3
UpDown+DF	81.1	55.1	39.9	71.6	226.2	40.3
UpDown+DF+PMM	81.6	55.6	41.5	73.6	236.5	43.0

Table 3. Comparison of experimental results for RSICD.

Methods	BLEU-1	BLEU-4	METEOR	ROUGE-L	CIDEr	SPICE
SAT [26]	73.2	44.3	36.2	65.2	244.2	46.1
Att2in [38]	72.6	43.8	36.0	64.4	245.6	45.9
SM-Att [39]	67.0	40.7	32.6	58.0	257.4	46.9
UpDown [11]	72.9	43.3	36.2	64.8	245.0	45.7
UpDown+PMM	73.9	45.2	36.5	65.4	253.7	46.8
UpDown+DF	73.2	44.7	37.2	65.3	257.9	47.4
UpDown+DF+PMM	73.6	45.4	37.3	66.0	263.4	47.7

4.5. Results Analysis

4.5.1. Quantitative Analysis

The experimental results from different methods are shown in Tables 1–3. Obviously, the scores of CIDEr and SPICE are all improved. The reason is that our method can store the information of the previous moment more persistently. Other methods will gradually overwrite the previous information over time. Although most of the overwritten information is useless, it is not available when we need it sometimes. In this situation, our method can describe the picture more accurately. Therefore, the scores of other evaluation metrics are also improved slightly. What is more, either the “UpDown+PMM” model or the “UpDown+DF+PMM” model can get a higher score than baseline model “UpDown”, even though the baseline model itself has a much higher score than most of the existing methods in

each evaluation metric. This strongly shows that the PMM indeed generates more comprehensive sentences to improve the descriptive effect, and it has successfully solved the problem about incomplete information memory in the decoder. The reason why the “UpDown+DF” model can also obtain a higher score than the baseline model is that we send them into the model for training in two parts while minimizing the loss of feature extraction, so as to make full use of the image information when the CNN extracts features. There is an interesting phenomenon in Table 1: the model “UpDown+PMM” is slightly inferior to the baseline model “UpDown”. The reason is that the average length of all of the sentences in the UCM-Captions dataset is short, and the number of images in each category is small. Our PMM can better improve the description effect of the dataset with long sentences or a large number of pictures in each category.

The reasons for the difference in scores for the three datasets are the size and classes of the datasets. The larger the datasets are, the more diverse the sentences are, and the more difficult it is to generate good descriptive sentences, meaning the lower the scores are. Therefore, it’s not difficult to understand that the dataset RSICD has the lowest scores. However, the UCM-Captions dataset has more pictures than the Sydney-Captions dataset, and the average length of the average sentence length of the Sydney-Captions dataset is longer than that of the UCM-Captions dataset, so the scores of the Sydney-Captions dataset are lower than those of the UCM-Captions dataset. In particular, the scores on CIDEr and SPICE evaluation metrics have been well improved.

4.5.2. Qualitative Analysis

Figure 4 is part of the captions generated by Updown, our model, and the ground-truth on the three remote-sensing datasets. From the comparison of the captions at the bottom of each set of pictures, we can see that the captions generated by our model can obtain more comprehensive captions than the baseline model. Even some objects that are not in the ground-truth captions but appear in the image (such as the red parts) can be described. This shows that our model can translate image features into text more effectively and present more details. This also suggests that our model can capture useful memories that are accidentally forgotten. In addition, sometimes our captions are the same as the ground-truth captions, but the expression is different. This is because the same types of pictures may have different expressions of artificial description. However, although the results described by our model are comprehensive, there are still some problems (such as the green part of the first picture on the third row). Object “basketball” gives an error number, which indicates that our model still needs to work hard on feature extraction of small objects in remote-sensing images. This is also a problem to be solved for remote-sensing images that are shot from outer space.

In summary, our model can produce a caption that is closer to the ground-truth captions. Sometimes our captions are more comprehensive than the ground-truth captions. Moreover, the PMM component can also be applied to the existing model to achieve a better promotion effect.



Figure 4. Qualitative results of our model “UpDown+DF+PMM” and baseline model “UpDown”. The first line of pictures is from UCM-Captions. The second line of pictures is from Sydney-Captions. The third line of pictures is from RSICD. The blue part and the red part are special captions.

4.5.3. Parameter of Memory Database Analysis

In this section, we analyze some parameters used in the PMM. It is mainly the size setting of the memory in model PMM. The matrix inside the memory stores the previously memorized information. The size of matrix affects the degree of information loss when the current information is updated to the database. We set the size of D_t to 20×512 and 20×256 and 20×128 respectively for comparison. 20 is the length of each sentence. 512, 256 and 128 represent the dimensions of the database. Because RSICD datasets has more and richer contents, we choose to carry out this part of the experiment on this dataset. This section of the experiment is conducted based on model “UpDown + DF + PMM”, and the experimental results are shown in the following Table 4. The bold numbers stand for the best scores:

Table 4. Results with different D_t on RSICD.

D_t Size	BLEU-1	BLEU-4	METEOR	ROUGE-L	CIDEr	SPICE
128	73.5	44.9	36.6	65.4	251.3	46.5
256	73.0	44.8	37.8	66.1	257.7	46.9
512	73.6	45.4	37.3	66.0	263.4	47.7

Table 4 shows that the PMM works better when the dimension is 512. Since the output dimension of the controller is 512, the transformation loses the least information when the dimension of the database is also set to 512, so the effect is the best.

5. Conclusions

In this paper, we propose a novel component named the Persistent Memory Mechanism and combine it with an advanced model for a remote sensing image-captioning task. This new model can retain and output the memory information for a longer time without being affected by the

vanishing gradient problem. In addition, we use a simple but effective feature extraction method for remote-sensing images in the encoder. We generate a more comprehensive and accurate sentence. More importantly, the baseline model using our method can obtain higher scores on all evaluation metrics. However, the search for information by our model will slightly increase the learning time. We will use a multiple attention mechanism to shorten the search time for information in the next step. In conclusion, the PMM can provide a new way to improve the semantic captioning effect of remote-sensing images.

Author Contributions: Conceptualization, Y.L.; methodology, Y.L.; software, Y.L.; validation, Y.L., W.Z., and X.S.; formal analysis, Y.L.; investigation, Y.L.; resources, X.S. and W.Z.; data curation, Y.L.; writing—original draft preparation, Y.L.; writing—review and editing, X.S., W.Z., and H.Y.; visualization, Y.L.; supervision, W.Z.; project administration, X.S. and W.Z.; funding acquisition, K.F. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Natural Science Foundation of China under Grant 41701508.

Acknowledgments: The authors would like to thank all colleagues in the lab, who are willing to give useful opinions. The authors would like to express their sincere appreciation for the reviewers for their helpful comments and suggestions.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Kanjir, U.; Greidanus, H.; Oštir, K. Vessel detection and classification from spaceborne optical images: A literature survey. *Remote Sens. Environ.* **2018**, *207*, 1–26. [[CrossRef](#)]
2. Guan, X.; Qi, W.; He, J.; Wen, Q.; Wang, Z. PURIFICATION OF TRAINING SAMPLES BASED ON SPECTRAL FEATURE AND SUPERPIXEL SEGMENTATION. *ISPRS - Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci.* **2018**, *XLII-3*, 425–430. [[CrossRef](#)]
3. Wang, B.; Lu, X.; Zheng, X.; Li, X. Semantic descriptions of high-resolution remote sensing images. *IEEE Geosci. Remote Sens. Lett.* **2019**, *16*, 1274–1278. [[CrossRef](#)]
4. Liu, D.; Zha, Z.J.; Zhang, H.; Zhang, Y.; Wu, F. Context-aware visual policy network for sequence-level image captioning. *arXiv* **2018**, arXiv:1808.05864.
5. Luo, R.; Price, B.; Cohen, S.; Shakhnarovich, G. Discriminability objective for training descriptive captions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 6964–6974.
6. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
7. Vinyals, O.; Toshev, A.; Bengio, S.; Erhan, D. Show and tell: A neural image caption generator. In Proceedings of the IEEE conference on computer vision and pattern recognition, Boston, MA, USA, 7–12 June 2015; pp. 3156–3164.
8. Graves, A.; Wayne, G.; Danihelka, I. Neural Turing machines. *arXiv* **2014**, arXiv:1410.5401.
9. Graves, A.; Wayne, G.; Reynolds, M.; Harley, T.; Hassabis, D. Hybrid computing using a neural network with dynamic external memory. *Nature* **2016**, *538*, 471–476. [[CrossRef](#)] [[PubMed](#)]
10. Chunseong Park, C.; Kim, B.; Kim, G. Attend to you: Personalized image captioning with context sequence memory networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 895–903.
11. McQueen, D.J.; Johannes, M.R.S.; Post, J.R.; Stewart, T.J.; Lean, D.R.S. Bottom-Up and Top-Down Impacts on Freshwater Pelagic Community Structure. *Ecol. Monogr.* **1989**, *59*, 289–309. [[CrossRef](#)]
12. Farhadi, A.; Hejrati, M.; Sadeghi, M.A.; Young, P.; Rashtchian, C.; Hockenmaier, J.; Forsyth, D. Every picture tells a story: Generating sentences from images. *European Conference on Computer Vision*; Springer: Berlin/Heidelberg, Germany, 2010; pp. 15–29.
13. Sun, C.; Gan, C.; Nevatia, R. Automatic concept discovery from parallel text and visual corpora. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 2596–2604.

14. Kulkarni, G.; Premraj, V.; Ordonez, V.; Dhar, S.; Li, S.; Choi, Y.; Berg, A.C.; Berg, T.L. Babytalk: Understanding and generating simple image descriptions. *IEEE Trans. Pattern Anal. Mach. Intell.* **2013**, *35*, 2891–2903. [[CrossRef](#)] [[PubMed](#)]
15. Kuznetsova, P.; Ordonez, V.; Berg, A.C.; Berg, T.L.; Choi, Y. Collective generation of natural image descriptions. In Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1, Association for Computational Linguistics, Jeju Island, Korea, 8 July 2012; pp. 359–368.
16. Mitchell, M.; Han, X.; Dodge, J.; Mensch, A.; Goyal, A.; Berg, A.; Yamaguchi, K.; Berg, T.; Stratos, K.; Daumé III, H. Midge: Generating image descriptions from computer vision detections. In Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics. Association for Computational Linguistics, Avignon, France, 23 April 2012; pp. 747–756.
17. Bahdanau, D.; Cho, K.; Bengio, Y. Neural machine translation by jointly learning to align and translate. *arXiv* **2014**, arXiv:1409.0473.
18. Nakayama, H.; Nishida, N. Zero-resource Machine Translation by Multimodal Encoder-decoder Network with Multimedia Pivot. *Mach. Transl.* **2016**, *31*, 49–64. [[CrossRef](#)]
19. Sutskever, I.; Vinyals, O.; Le, Q. Sequence to sequence learning with neural networks. *Adv. NIPS* **2014**, *2*, 3104–3112.
20. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Madison, WI, USA, 26 June–1 July 2016; pp. 770–778.
21. Kiros, R.; Salakhutdinov, R.; Zemel, R. Multimodal neural language models. In Proceedings of the International Conference on Machine Learning, Beijing, China, 10 November 2014; pp. 595–603.
22. Mao, J.; Xu, W.; Yang, Y.; Wang, J.; Yuille, A.L. Explain images with multimodal recurrent neural networks. *arXiv* **2014**, arXiv:1410.1090.
23. Chen, X.; Zitnick, C.L. Learning a recurrent visual representation for image caption generation. *arXiv* **2014**, arXiv:1411.5654.
24. Fang, H.; Gupta, S.; Iandola, F.; Srivastava, R.K.; Deng, L.; Dollár, P.; Gao, J.; He, X.; Mitchell, M.; Platt, J.C.; et al. From captions to visual concepts and back. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 1473–1482.
25. Lu, J.; Xiong, C.; Parikh, D.; Socher, R. Knowing when to look: Adaptive attention via a visual sentinel for image captioning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 375–383.
26. Xu, K.; Ba, J.; Kiros, R.; Cho, K.; Courville, A.; Salakhutdinov, R.; Zemel, R.; Bengio, Y. Show, attend and tell: Neural image caption generation with visual attention. In Proceedings of the International Conference on Machine Learning, Lille, France, 6–11 July 2015; pp. 2048–2057.
27. Qu, B.; Li, X.; Tao, D.; Lu, X. Deep semantic understanding of high resolution remote sensing image. In Proceedings of the 2016 International Conference on Computer, Information and Telecommunication Systems, Kunming, China, 6–8 July 2016; pp. 1–5.
28. Lu, X.; Wang, B.; Zheng, X.; Li, X. Exploring models and data for remote sensing image caption generation. *IEEE Trans. Geosci. Remote Sens.* **2017**, *56*, 2183–2195. [[CrossRef](#)]
29. Zhang, Z.; Diao, W.; Zhang, W.; Yan, M.; Gao, X.; Sun, X. LAM: Remote Sensing Image Captioning with Label-Attention Mechanism. *Remote Sens.* **2019**, *11*, 2349. [[CrossRef](#)]
30. Yang, Y.; Newsam, S. Bag-of-visual-words and spatial extensions for land-use classification. In Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems, San Jose, CA, USA, 2–5 November 2010; pp. 270–279.
31. Zhang, F.; Du, B.; Zhang, L. Saliency-guided unsupervised feature learning for scene classification. *IEEE Trans. Geosci. Remote Sens.* **2014**, *53*, 2175–2184. [[CrossRef](#)]
32. Papineni, K.; Roukos, S.; Ward, T.; Zhu, W.J. BLEU: A method for automatic evaluation of machine translation. In Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, Philadelphia, PA, USA, 7–12 July 2002; pp. 311–318.
33. Lavie, A.; Agarwal, A. METEOR: An automatic metric for MT evaluation with high levels of correlation with human judgments. In Proceedings of the Second Workshop on Statistical Machine Translation. Association for Computational Linguistics, Ann Arbor, MI, USA, 11 June 2005; pp. 228–231.

34. Lin, C.Y. Rouge: A package for automatic evaluation of summaries. In Proceedings of the Text Summarization Branches out, Barcelona, Spain, 25–26 July 2004; pp. 74–81.
35. Vedantam, R.; Lawrence Zitnick, C.; Parikh, D. Cider: Consensus-based image description evaluation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 4566–4575.
36. Anderson, P.; Fernando, B.; Johnson, M.; Gould, S. SPICE: Semantic Propositional Image Caption Evaluation. *Adapt. Behav.* **2016**, *11*, 382–398.
37. Deng, J.; Dong, W.; Socher, R.; Li, L.J.; Li, K.; Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009; pp. 248–255.
38. Rennie, S.J.; Marcheret, E.; Mroueh, Y.; Ross, J.; Goel, V. Self-critical sequence training for image captioning. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 20 July 2017; pp. 7008–7024.
39. Zhang, X.; Wang, X.; Tang, X.; Zhou, H.; Li, C. Description generation for remote sensing images using attribute attention mechanism. *Remote Sens.* **2019**, *11*, 612. [[CrossRef](#)]



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).