Can a Machine Generate Humanlike Language Descriptions for a Remote Sensing Image?

Zhenwei Shi, *Member, IEEE* and Zhengxia Zou

***Abstract*— This paper investigates an intriguing question in the remote sensing field: "can a machine generate humanlike language descriptions for a remote sensing image?" The automatic description of a remote sensing image (namely, remote sensing image captioning) is an important but rarely studied task for artificial intelligence. It is more challenging as the description must not only capture the ground elements of different scales, but also express their attributes as well as how these elements interact with each other. Despite the difficulties, we have proposed a remote sensing image captioning framework by leveraging the techniques of the recent fast development of deep learning and fully convolutional networks. The experimental results on a set of high-resolution optical images including Google Earth images and GaoFen-2 satellite images demonstrate that the proposed method is able to generate robust and comprehensive sentence description with desirable speed performance.**

***Index Terms*— Fully convolutional networks (FCNs), high-resolution optical remote sensing image, image understanding, remote sensing image captioning.**

I. Introduction

WHEN does a machine "understand" an image? One definition might be when it can generate sentences that summarize the image content [1]. Nowadays, remote sensing imaging techniques have opened a door for people to observe the earth [2]. Many researchers are making efforts to let machines better understand the remote sensing image. Since language is one of the most common information carriers that is close to human cognition, a natural question then arises: "can a machine generate humanlike language description for remote sensing image?" In this paper, we try to find the answer.

Automatic remote sensing image description (also called remote sensing image captioning) aims to let machines describe the content of a remote sensing image in human languages and better present useful information to users. Although there have been many previous works such as remote sensing image labeling [3], remote sensing target detection [4]–[8], and scene classification [9], [10], remote sensing image captioning differs from all these tasks in that it aims to generate comprehensive sentences rather than predicting individual tags or words. To generate concise and meaningful sentence description, one must well recognize the ground elements under different levels, analyze their attributes, and exploit their class dependence and spatial relationships from the "view of God."

Despite the difficulties, there lie extensive potential applications of remote sensing image captioning for both civil and military use. Here we present two scenarios.

1) *Image retrieval:* The automation of this process can be very helpful for remote sensing image retrieval where users can go beyond the keyword search to describe their information needs and improve the accessibility of collecting useful images.
2) *Military intelligence generation:* At war time, battlefield images captured by spy drones or satellites can be automatically transformed to text or voice messages. These messages can be further sent to the frontline combat soldiers or the command center.

Automatically generating descriptions of image has long been a difficult and fundamental AI problem. Thanks to the fast development of computer vision and natural language processing technologies, it is now becoming a real possibility for intelligent systems to talk about natural images [1], [11]–[15]. Although an image may contain a vast amount of visually discernible information that is difficult to be completely characterized by limited natural languages, nonetheless, the recent progress on automatic generation of natural image captions has greatly disrupted the well-known adage that a picture is worth a thousand words [13]. During the past one or two years, many image captioning systems have shown that it is possible to describe the most salient information conveyed by images with accurate and conscious sentences.

For most natural image captioning methods, their algorithm flow can be divided into the following two stages: 1) image understanding and 2) language generation. In the first stage, the algorithm aims to recognize the objects in an image, analyze their attributes, and determine how the objects interact with each other. Some frequently used methods can be grouped into two categories: region feature-based methods [1], [11]–[13] and global feature-based methods [14], [15], where the former extract features individually and analyze the image content based on multiple visual regions, while the latter extract global features directly from the whole image. In the second stage, words are arranged

Manuscript received October 29, 2016; revised January 18, 2017; accepted February 27, 2017. Date of publication March 31, 2017; date of current version May 19, 2017. The work was supported in part by the National Natural Science Foundation of China under Grant 61671037 and Grant 61273245, in part by the Beijing Natural Science Foundation under Grant 4152031, and in part by the funding project of the State Key Laboratory of Virtual Reality Technology and Systems, Beihang University under Grant BUAA-VR-16ZZ-03. *(Corresponding author: Zhengxia Zou.)*

The authors are with the Image Processing Center, School of Astronautics, Beihang University, Beijing 100191, China, with the Beijing Key Laboratory of Digital Media, Beihang University, Beijing 100191, China, and also with the State Key Laboratory of Virtual Reality Technology and Systems, School of Astronautics, Beihang University, Beijing 100191, China (e-mail: shizhenwei@buaa.edu.cn; zhengxiazou@buaa.edu.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TGRS.2017.2677464

to form meaningful sentences based on the image content. A classical approach of this stage is to use predefined templates to generate sentences by filling detected visual elements [16]–[20]. Image retrieval-based methods [21], [22] are also frequently used. These methods are first to search for similar images and corresponding annotated sentences in the data set, and then create a new sentence based on the retrieval result. However, sentences generated by these methods are relatively fixed and limited. Recent works aim to automatically decode words learned from data into natural language sentences by training deep recurrent neural networks [12]–[15] to generate more creative and more flexible sentences.

Although noticeable progress has been made in natural image captioning, a similar problem has rarely been studied yet in the remote sensing field. Compared to natural image captioning, there are two main differences in the remote sensing image captioning task.

A. Multilevel Semantics

For remote sensing images, the same ground elements may present totally different semantics under different geographical scales. For example, consider a remote sensing image of an airport, where the pixel level semantics may correspond to the ground material features, such as metal, concrete, and soil, the target level semantics may correspond to the objects itself and its attributes, such as airplane, terminal building, and runway, while the environmental level semantics may correspond to a wider range of ground areas, such as airport, harbor, ocean, or city. A remote sensing image captioning task should be established on multiple levels of geographical scales.

B. Semantic Ambiguity

There may be some "gray zones" between different geographical semantic classes, especially for those large-scale regions with multiple semantical attributes. Sometimes, it is hard to characterize a particular area of remote sensing image by a single semantic label. For example, for those urban–rural area (an area with both urban and rural characteristics), or the junction area of the harbor and land structures, the image content may present ambiguous semantic characteristics.

In this paper, we propose an effective method for remote sensing the image captioning task in response to all the above characteristics. The motivation for our research lies in three aspects: 1) a fundamental AI problem; 2) an important but rarely studied task; and 3) potential remote sensing applications. In order to obtain a comprehensive and detailed description of the remote sensing images, some typical ground elements are decomposed as the following three levels in our framework (see Fig. 1).

1) *Key-Instance:* Airplane, oilpot, and ship.
2) *Envi-Element:* Airport, harbor, buildings, and farmland.
3) *Landscape:* City, suburbs, ocean, and mountain.

Arguably, the starting point for automatic remote sensing image description is to understand the image. In recent years, convolutional neural network (CNN) has played a very important role in tasks like image classification [23], [24], object recognition [25], [27], and natural image captioning [11], [13]. CNN constructs multiple layers of neural networks to learn high-level image features with better discrimination and robustness, as opposed to that in traditional methods, where features have to be handcrafted designed. The rapid development of CNN gave birth to a new technology: fully convolutional networks (FCNs) [28], [30]. FCN is specifically designed to predict a 2-D label map rather than a single label as CNN for an arbitrary-sized input image, which greatly increases the processing flexibility and computational efficiency. The powerful representation ability and structural flexibility of an FCN model make it suitable for many computer vision tasks, such as natural image object detection [26], [27], [29] and sematic segmentation [28], [30], which has shown greater potential than the traditional CNN-based methods.

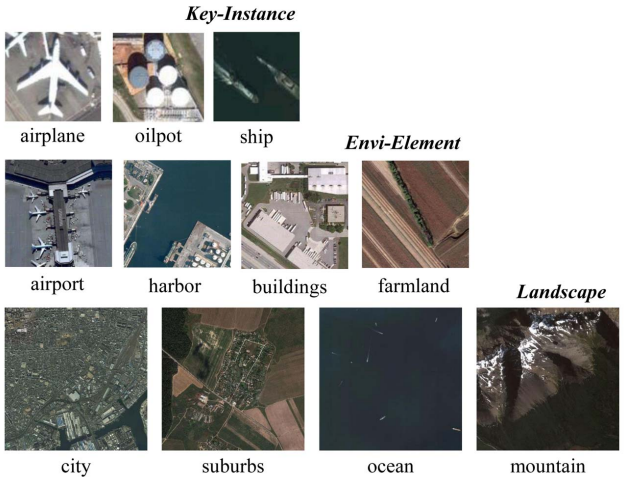


Fig. 1. Semantic decomposition of ground elements with different scales.

By leveraging the recent popular FCN technology, our method consists of two stages: 1) a *multilevel image understanding* stage, where ground elements of different levels are detected and identified by a single FCN model and 2) a *language generation* stage, where the language descriptions are generated by integrating the results of the previous stage. Fig. 3 shows the algorithmic flow of the proposed method, where the multilevel image understanding stage is further subdivided into three subtasks: key instance detection, envi-element analysis, and landscape analysis. The lower left corner of this figure shows the autogenerated description of the proposed method.

The rest of this paper is organized as follows. In Sections II and III, we introduce our FCN model for multilevel image understanding. In Section IV, we introduce our method for language generation. Some experimental results are given in Section V, and the conclusions are drawn in Section VI.

II. Multilevel Image Understanding

In this section, we give a detailed introduction to our FCN model and explain how it works in the multilevel image understanding stage.

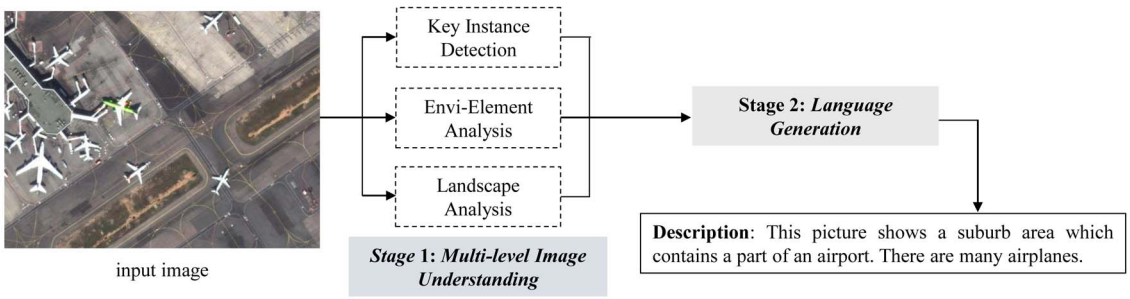


Fig. 2. Algorithmic flow of the proposed method. (Lower left corner) Autogenerated description of the proposed method.

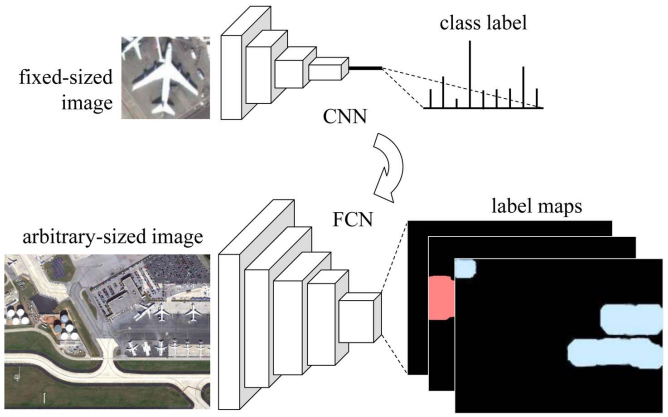


Fig. 3. Typical structures of a CNN and an FCN.

A. Fully Convolutional Networks

We deal with the three subtasks including key instance detection, envi-element analysis, and landscape analysis in a unified FCN model. In the previous remote sensing literature, these problems were often separated into divided tasks despite the high correlations between them.

FCN shares similar structural units with CNN. A typical FCN can be constructed by stacking with a series of convolutional layers, pooling layers, and activation layers.

Concretely, a convolution layer is designed to capture the basic local image pattern and is invariant to translation. The convolutional output is called a feature map as the output represents the corresponding feature of each image pixel. A pooling layer, acting as a downsampled filter, is designed to increase the scale invariance and also to decrease the computational cost of the subsequent layers. A typical form of pooling operation is "max-pooling," where each element of its output corresponds to the maximum value of the local area of the input. An activation layer is designed to add nonlinearity to the networks and to enhance its representation ability. An activation layer is often inserted behind a convolutional layer, with the operation of making pixelwise nonlinear transformation of the feature maps. The commonly used activation functions include sigmoid function, tanh function, and recent-popular ReLU function [23]. At the end of an FCN, a loss layer (also called a decision layer) is designed according to a specific task such as classification or regression, by replacing the traditional fully connected layer of CNN with 1×1 convolutional layers. There are no full-connected layers in an FCN and all the parameters are embedded in convolutional layers. Since convolution operation would not limit the input image size, FCN allows arbitrary-sized input image, while in classical CNN models [23], [24], the input image size must be fixed. FCN is designed in this way to predict a pixelwise output label map rather than a single class label for an arbitrary-sized input image. FCNs pixel-to-pixel output map naturally keeps the spatial information of a remote sensing image, which makes it suitable for our needs. Fig. 2, modified from [28], illustrates how a typical FCN differs from a typical CNN model in this problem. In the next two sections, we will give a detailed description of the proposed networks.

B. Network Structure

The input of our FCN model consists of three parts, corresponding to the three subtasks: key instance detection, envi-element analysis, and landscape analysis as is shown in Fig. 4.

For key instance detection, a whole test image $\mathcal{I}$ of $H \times W$ pixels and D channels is fed into the FCN model to produce a set of probability maps, where the probability value of each pixel indicates how likely a convolutional window covers a certain type of instance (oilpot, ship, or airplane). In the feedforward process, the data passes through a series of convolutional layers, activation layers, and pooling layers, finally forming a d-dimensional feature cube $\mathcal{C}_{\mathcal{I}}$, whose each "pixel" $\mathbf{x}_i \in \mathbb{R}^{d\times 1}$ refers to a feature representation of the corresponding convolutional window. At the end of the net, a linear decision layer is used to highlight the desired key instances, meanwhile, to suppress those undesired background regions. For instances of a certain class k, $h_k(\mathbf{x}_i)$ represents the output probability of $\mathbf{x}_i$

$$\begin{aligned} h_k(\mathbf{x}_i) &= P\big(+1|\mathbf{x}_i; \mathbf{w}_{KI}^{kT}\big)\big) \\ &= 1/\big(1+\exp\big(-\mathbf{w}_{KI}^{kT}\mathbf{x}_i\big)\big) \end{aligned} \tag{1}$$

namely, how likely the corresponding image window frames an instance, where $\mathbf{w}_{KI}^{k} \in \mathbb{R}^{d\times 1}$ is the discriminative coefficients for the key instances of class k, $k \in \{1, 2, 3\}$. Since the elementwise inner product operation in (1) at each location is identical to the 3-D convolution on the feature cube $\mathcal{C}_{\mathcal{I}}$ with a reshaped $1 \times 1 \times d$ filter, the decision layer can be reformed as a convolutional layer followed by an additional activation layer. The label of each output pixel depends on the class-id

TABLE I
DETAILED CONFIGURATIONS OF VGG-f AND THE PROPOSED FCN MODEL

Arch.	Conv1	Conv2	Conv3	Conv4	Conv5	Conv6	Layer7	Layer8
VGG-f	64×11×11 Step = 4 Pad = 0 ×2 Pool.	256×5×5 Step = 1 Pad = 2 ×2 Pool	256×3×3 Step = 1 Pad = 1 -	256×3×3 Step = 1 Pad = 1 -	256×3×3 Step = 1 Pad = 1 ×2 Pool	4096×6×6 Step = 1 Pad = 0 -	Fully-Connected 4096→4096	Fully-Connected 4096→1000
Ours	64×11×11 Step = 4 *Pad = 5* ×2 Pool	256×5×5 Step = 1 Pad = 2 *×1 Pool*	256×3×3 Step = 1 Pad = 1 -	256×3×3 Step = 1 Pad = 1 -	256×3×3 Step = 1 Pad = 1 *×1 Pool*	4096×6×6 Step = 1 *Pad = 3* -	*Convolution* *4096×1×1*	*Convolution* *4×1×1* *Fully-Connected* *2×(4096→4)*

with the maximum probability value. To detect the instances of different sizes, we downsample the original image several times while keeping all the FCN filters fixed.

For envi-element analysis, we should focus on regions of larger scales. Considering the semantic ambiguity that some large-scale regions may share multiple semantics at different locations, the input image are randomly cropped into a set of image patches $\mathcal{P}_1, \mathcal{P}_2, \ldots, \mathcal{P}_M$ at random locations. One can easily note that if the patch size is controlled within a suitable range (e.g., the size of the network's perceptive field, which we will talk about later), the output can just become a probability vector $\mathbf{h}(\mathcal{P}_i)$ with 1×1 spatial range rather than a probability map, which turns out to be the same output form as a traditional CNN. In this way, each patch is individually fed into the network and is treated as a multiclass classification process by a softmax layer at the end of the net

$$h_k(\mathcal{P}_i) = P(k|\mathcal{P}_i; \mathbf{W}_{EE}) = \frac{\exp(\mathbf{w}_{EE}^{kT}\mathbf{p}_i)}{\sum_{j=1}^{4}\exp(\mathbf{w}_{EE}^{jT}\mathbf{p}_i)} \tag{2}$$

where $\mathbf{p}_i \in \mathbb{R}^{d\times 1}$ is the feature representation of the patch $\mathcal{P}_i$, $h_k(\mathcal{P}_i)$ is the probability of the patch $\mathcal{P}_i$ being specified to the class k, and $\mathbf{W}_{EE} = [\mathbf{w}_{EE}^1, \ldots, \mathbf{w}_{EE}^4] \in \mathbb{R}^{d\times 4}$ are their discriminative projective coefficients. The final results of the test image can be obtained by an averaging of the results of all patches at different locations

$$\mathbf{h}(\mathcal{I}) = \frac{1}{M}\sum_{i=1}^{M}\mathbf{h}(\mathcal{P}_i). \tag{3}$$

For landscape analysis, the probability vector of each patch can be obtained in similar manner, while the only difference is to replace $\mathbf{W}_{EE}$ of (2) with the landscape coefficients $\mathbf{W}_{LS} \in \mathbb{R}^{d\times 4}$.

We build our FCN model based on VGG-f [24], [31], a classical CNN model that has shown state-of-the-art performance in natural image tasks such as image classification [24] and object detection [32]. In our FCN model, the first–seventh convolutional layers use shared weights for all the three subtasks, while the weights of the final decision layer are specifically trained for different subtasks. Table I shows the detailed configurations of our model and VGG-f. We have made some changes in VGG-f so that it can be better adapted to our task. Specific changes (emphasized in *italics* in Table I) lie in the following three aspects.

1) The fully connected layers are replaced by the 1×1 convolutional layers to accept arbitrary-sized input image and keep the spatial information of the output.
2) We choose smaller pooling step to adjust the size of the receptive field.
3) We use appropriate padding size for each convolutional layer and each pooling layer so that an instance window would not go out of the border of feature maps through the feedforward process.

C. Receptive Field Analysis

The receptive field is an important biologically inspired concept that derives from animals' visual cortex. In FCN, a receptive field refers to the spatial range of input pixels that contribute to the calculation of a single pixel of the output as is shown in Fig. 5. A network with a larger receptive field tends to catch the structural information of larger scale such as the envi-element and landscape, while that with a smaller one may concentrate more on the local details such as the appearance of key instances.

The radius of receptive field R of our networks is related to the convolutional filter radius r_i^c, convolutional step size s_i^c, pooling radius r_i^p, pooling step size s_i^p, and number of layers n, $i, j \in \{1, 2, \ldots, n\}$. It can be calculated by accumulating each layers' receptive radius r_i as[1]

$$R = \sum_{i=1}^{n} r_i = \sum_{i=1}^{n}\left(r_i^c r_i^p \prod_{j=1}^{i} s_j^c s_j^p\right). \tag{4}$$

The size of output map can be calculated as

$$[h, w] = [H, W]/\left(\prod_{i=1}^{n} s_j^c s_j^p\right) \tag{5}$$

where $[h, w]$ and $[H, W]$ are the sizes of input image and output maps. It is clear that using a larger radius or step size can increase the receptive field and decrease the resolution of the output maps. An appropriate choice of receptive field should be considered as a tradeoff between different target scales.

Let us take an image of 1 m/pixel for example, by adjusting the network structure as that shown in Table I, the receptive field can reduce to 171 m × 171 m, which well captures the appearance and scale of oilpot, ship and airplane. To further enlarge the receptive field for envi-element, we can downsample the input image while ignoring some local details. For example, by setting the downsampled factor as 3.0, the receptive field can be roughly scaled up into

[1]Equations (4) and (5) can only be applied to those networks with odd-sized units.

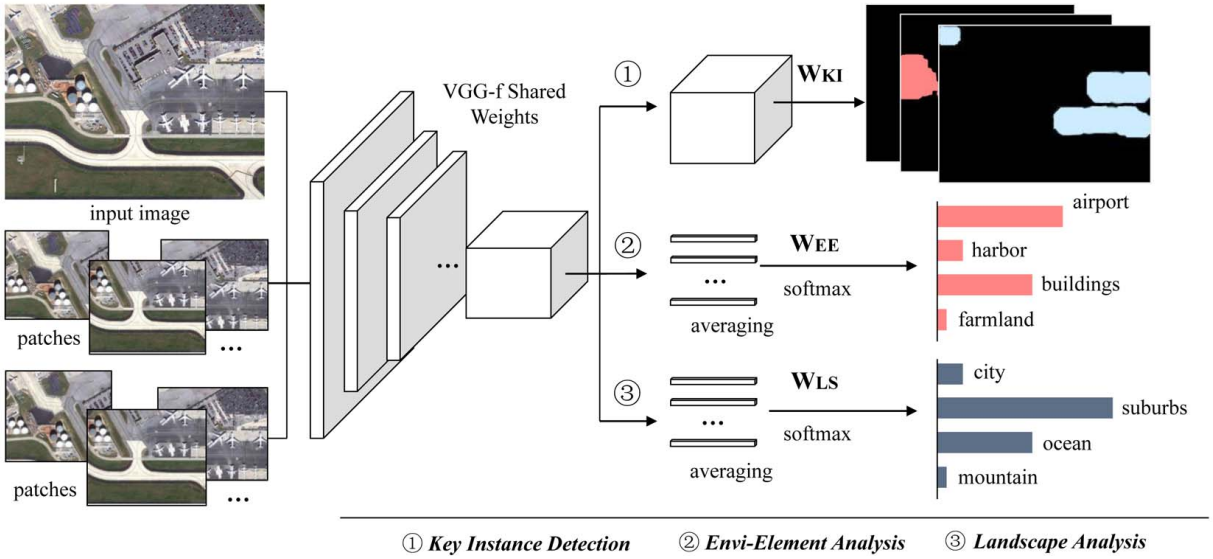


Fig. 4. Illustration of the proposed network for multilevel remote sensing image understanding.

500 m × 500 m, which can well cover the scale of an airport or a harbor. For landscape analysis, we do not use a larger receptive field since it may exceed the limit of the input image size.

III. Loss Layer Design

We simply follow the idea of "transfer learning" of some previous detection and classification literatures [33], [34]. During the training process, all the filters of the first–seventh layers are fixed as the feature extractor. The reason is that previous researches have shown the high-level features extracted from the activation of a deep convolutional network trained on a large fixed set of object recognition tasks, such as ImageNet [35], can be repurposed to novel generic tasks. The features even show a clear advantage for remote sensing image tasks without fine-tuning, such as oil tank detection [5] and remote sensing scene classification [34], despite their big differences from the originally trained tasks. In our model, the filters are transferred from VGG-f [31], which is well trained in a fully supervised fashion on ImageNet.

The loss of a network is usually task-related. A loss layer takes in data both from previous layers and ground-truth labels so that the filters of networks can be iteratively adjusted by comparing them with a loss function during the training process. Typical loss functions of FCN are designed based on pixel-by-pixel ground-truth maps [23], [24], where the ground-truth map has to be labeled manually. In an image captioning task, since it is not necessary to obtain accurate contour information of an instance, we redesign the loss of our model based on ground-truth bounding-boxes rather than pixel-by-pixel ground-truth maps.

Let Ω be a space of locations for each window within an image, and $\{\omega_1, \omega_2, \ldots, \omega_N\} \in \Omega$ could specify N training windows with corresponding positions and scales. Let Φ be a space of each ground-truth bounding-box within an image, and $\{\beta_1, \beta_2, \ldots, \beta_M\} \in \Phi$ could specify M ground-truth bounding-boxes of a certain class. In our training process,

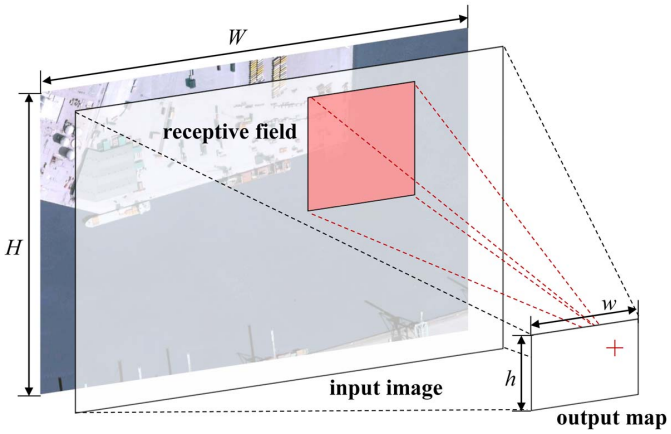


Fig. 5. Illustration of the reception field for an FCN.

the training data are randomly sampled at different positions and scales from a number of labeled images. Apart from the training data and the corresponding class label itself, the loss function is also designed to be related to a weight assigned to each training window, as is shown in Fig. 6. Let $m(\omega_i|\Phi) \in [0, 1]$ be the weight for a training window at location ω_i

$$m(\omega_i|\Phi) = \begin{cases} 0 & \text{if } \omega_i \bigcap \beta_j = \emptyset, \ \forall \beta_j \in \Phi \\ \dfrac{C(\omega_i \bigcap \beta_j)}{C(\omega_i \bigcup \beta_j)} & \text{if } \omega_i \bigcap \beta_j \neq \emptyset, \ \exists \beta_j \in \Phi \end{cases} \tag{6}$$

where $C(\omega_i \bigcap \beta_j)$ denotes the intersection area of two windows ω_i and β_j, $C(\omega_i \bigcup \beta_j)$ denotes the total area covered by the two windows. $m(\omega_i|\Phi)$ can be understood as the intersection-over-union (IOU) of two sets. It should be noted that although a window is not strictly equivalent to a "set" of pixels, here we still follow some concepts such as "intersection," "union," and "cardinality" to simplify notation.

If the weight between a training window and any ground-truth bounding-boxes is greater than 0.5, we see this window as a positive window, else, we see this window as a negative one. Suppose we have collected N training windows, including

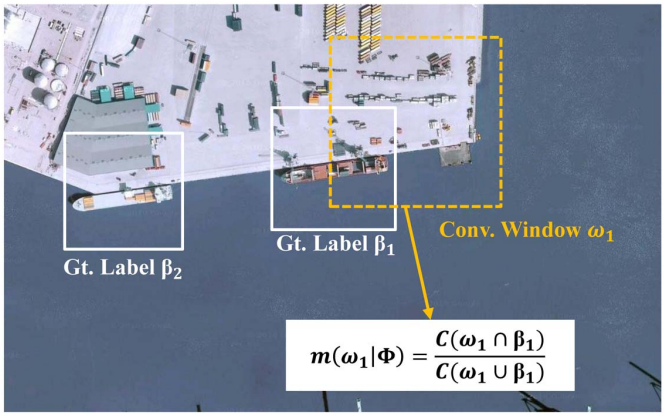


Fig. 6. We assign a score $m(\omega_i|\Phi)$ for each training window ω_i in our loss function indicating how much this window contributes to the loss.

N_t positive windows and N_t negative windows, then the log-likelihood of all windows can be represented as

$$\log(L(\mathbf{w}_{KI}^k) = \log\left(\prod_{i=1}^{N} P\left(y_i|\mathbf{x}_i; \mathbf{w}_{KI}^k\right)\right) \tag{7}$$

where $y_i = \{+1, -1\}$ represents the label of the training window for a certain instance class k. By dividing the above formulation into a "positive part" and a "negative part", substituting $P(y_i|\mathbf{x}_i; \mathbf{w}_{KI})$ by (1) and further introducing the weight $m(\omega_i|\Phi)$, we have

$$\begin{aligned}
&\log\left(L\left(\mathbf{w}_{KI}^k; m(\omega_i|\Phi)\right)\right) \\
&\quad = E_{p(\mathbf{x}+)}\left\{m(\omega_i|\Phi)\log\left(P\left(+1|\mathbf{x}_i; \mathbf{w}_{KI}^k\right)\right)\right\} \\
&\qquad E_{p(\mathbf{x}-)}\left\{\log\left(P\left(-1|\mathbf{x}_i; \mathbf{w}_{KI}^k\right)\right)\right\} \\
&\quad = -E_{p(\mathbf{x}+)}\left\{m(\omega_i|\Phi)\log\left(1+\exp\left(-\mathbf{w}_{KI}^{kT}\mathbf{x}_i\right)\right)\right\} \\
&\qquad - E_{p(\mathbf{x}-)}\left\{\log\left(1+\exp\left(\mathbf{w}_{KI}^{kT}\mathbf{x}_i\right)\right)\right\}.
\end{aligned} \tag{8}$$

The advantages of the above operation are as follows.

1) *Data Balance:* It is crucial to balance the samples during the training process. Since key instances only occupy a small group of pixels in a remote sensing image, training with likelihood of all samples randomly sampled from the image may suffer from unbalanced training data between key instances and backgrounds. By separating the log likelihood of positive and negative samples into the sum of their self-expectations, the networks can equally learn preferences toward different classes.
2) *Adaptive Weights:* While collecting training samples by traditional approaches, boundaries between positive and negative samples are usually not clear. By introducing the adaptive weight for each training sample, the appearance of positive samples that partly shift out of the window can also be well captured by the network. This can also be considered as an adaptive way of data augmentation.

The filters of the loss layer can be easily learned by maximum likelihood estimation. Since maximizing the log-likelihood $\log(L(\cdot))$ is just equivalent to minimizing the negative log-likelihood $-\log(L(\cdot))$, the learning process for key instance can be finally represented by solving the following unconstrained optimization problem:

$$\min_{\mathbf{w}_{KI}^k} J\left(\mathbf{w}_{KI}^k\right) = -\log(L(\mathbf{w}_{KI}^k; m(\omega_i|\Phi))) + \alpha\left\|\mathbf{w}_{KI}^k\right\|_2^2 \tag{9}$$

where α is a positive parameter for regularization term to increase the model's generalization ability, $k \in \{1, 2, 3\}$. The problem (9) can be efficiently solved by the stochastic gradient descent (SGD) algorithm [36], which has been commonly used for large-scale optimization problems. SGD is very similar to traditional batch gradient descent algorithm, except that the gradient is randomly estimated in each round of the update. The gradient of (9) with two small batches of n^+ positive samples from positive set S^+ and n^- negative samples from negative set S^- can be calculated as

$$\begin{aligned}
\nabla J\left(\mathbf{w}_{KI}^k\right) \doteq &-\frac{1}{n^+}\sum_{i\in \mathrm{S}^+}\frac{m(\omega_i|\Phi)\exp\left(-\mathbf{w}_{KI}^{kT}\mathbf{x}_i^+\right)}{1+\exp\left(-\mathbf{w}_{KI}^{kT}\mathbf{x}_i^+\right)}\mathbf{w}_{KI}^{kT} \\
&+\frac{1}{n^-}\sum_{i\in \mathrm{S}^-}\frac{\exp\left(\mathbf{w}_{KI}^{kT}\mathbf{x}_i^-\right)}{1+\exp\left(\mathbf{w}_{KI}^{kT}\mathbf{x}_i^-\right)}\mathbf{w}_{KI}^{kT} + 2\alpha\mathbf{w}.
\end{aligned} \tag{10}$$

In this way, $\mathbf{w}_{KI}^k$ can be updated with a learning rate μ

$$\mathbf{w}_{KI}^k \leftarrow \mathbf{w}_{KI}^k - \mu\nabla J\left(\mathbf{w}_{KI}^k\right) \tag{11}$$

until it converges to a constant.

For envi-element and landscape analysis, $\mathbf{W}_{EE}$ and $\mathbf{W}_{LS}$ can be solved by optimizing their softmax functions. Similar approaches [23], [36] have been widely used and introduced in other literatures and thus will not be discussed here.

IV. Language Generation

In the language generation stage, words are arranged into sentences by integrating information about previous results. Our model is designed based on a template-based approach with linguistic constraints, a technique that has been used for various practical applications such as summarization [37] and dialog systems [38]. Some recent works aim to generate sentences with language models automatically learned from image data, such as long short-term memory (LSTM) [12], [13], [15]. Although some of these learning-based methods have achieved the state-of-the-art results of generating sentences for natural images, in this paper, we still follow the classical paradigm. There are two main reasons for our choice. For the first reason, those learning-based models are trained based on a large number of annotated sentences (usually by millions of corpus), while we do not have such corpus for remote sensing images at present. For the second reason, using predefine templates, one can easily design new template and generate new rules according to the characteristic of remote sensing images and our specific needs.

In our model, the representation space of a remote sensing image can be arranged by triplets of

$$\{\mathrm{ELM},\ \mathrm{ATR},\ \mathrm{RLT}\} \tag{12}$$

where ELM refers to the ground elements of different levels. ATR refers to attributes such as quantity, size, and location.

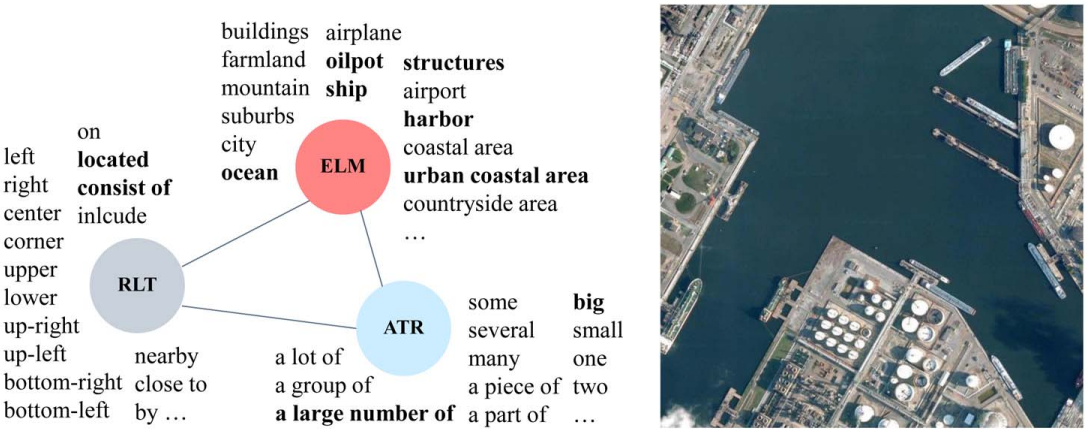


Fig. 7. We represent the representation space by triplets of {ELM, ATR, RLT}. The bolded text corresponds to the content of the example image.

RLT refers to the relationship of different elements. Fig. 7 shows an example image and the representation space of the triplets. Some commonly used corpus for remote sensing image captioning task are also shown in Fig. 7. Based on this paradigm, different sentence templates can be designed. For example, a possible sentence template can be defined as

$$\{\text{Prefix}, \langle \#E1, E1\rangle, \text{Prep}, \langle \#E2, E2\rangle\} \tag{13}$$

where Ei is the name of a ground element, e.g., "ship." $\#Ei$ is the number of instances of Ei, e.g., "one." $Prep$ defines the set of interactions {"by," "on the left side of," ...}. $Prefix$ defines the set of prefix of a sentence {"This is a picture of," "This image shows," ...}. Afterward, the generated sentences are checked by grammar rule and some language mistakes are corrected.

For prefix design, there has been two common viewpoints in natural image captioning filed. The first tends to neglect the prefix of a sentence [18], [19], since it does not provide extra information for machine learning applications such as content-based image retrieval. The second holds that adding a prefix is sometimes meaningful [17], [20] and should be added especially for some human-oriented applications such as voice military intelligence generation, since the prefix of a sentence provides complete and user-friendly language description for humans. As it is the first time we explore this problem in the remote sensing field, we follow a relatively conservative approach, to retain the prefix of the sentence.

There are several critical issues that should be noticed in language generation stage. First, a comprehensive analysis of the interaction and relationship between the ground elements is necessary to generate meaningful sentence. The main difference between the remote sensing image and the natural image lies in their projection relationship: one is taken from the perspective of the human eyes in daily life, while the other is taken above the head. In a natural image, people tend to focus on the interaction of 3-D space, such as "front and back," and "above and below." However, in the remote sensing image, we should pay more attention to the ground distance and orientation,[2] such as "near and far," and "left and right." Second, determining the attributes of key instances is equally important, such as the quantity and the size. This process highly depends on the correct location of each instance. To extract the bounding box location of each instance, we use the nonmaximal suppression [39] based on the detection map of each scale. It also should be noticed that the attribute of ground element such as "near and far," and "big and small" is closely related to image resolution and their specific class. For example, an airplane with the size of 80 pixels under 1 m/pixel resolution should be referred as "a big" airplane, while that a ship with that size under 2 m/pixel resolution could be referred as a "medium" one. Lastly, on the basis of the above mentioned, the description should be concise, because humans tend to describe the most significant objects or events of an image. In our method, we only selectively describe the top-one or the top-two elements with the highest scores of each semantic level, while other details are omitted.

V. Experimental Results and Analysis

In this section, some high-resolution optical images, including Google Earth images and GaoFen-2 (GF-2) satellite images are used to demonstrate the effectiveness and transfer ability of the proposed method. As there is no previous research on remote sensing image captioning task and no public data set, we do not directly compare with other image captioning methods. Our Code and trained models are available at http://levir.buaa.edu.cn/Code.htm.

A. Experiment Setup

Our experimental data set consists of a number of Google Earth images (RGB image) and GF-2 images (multispectral image). The resolution of the Google Earth image is 0.5 m/pixel. The resolutions of GF-2 spectral-fused images are 0.8 m/pixel. To verify the algorithm's performance under different scales, the images are cropped into some slices with different sizes and a fixed resolution. For large slices, their sizes range from 1000 to 8000 pixels. For small slices, their

[2]Here we only consider the most general cases, that the image does not have elevation information.

TABLE II
DETAILED INFORMATION OF THE EXPERIMENTAL IMAGE SLICES

Source	Size	Training	Testing	Resolution
Google Earth	1000∼8000 pxl	310	10	0.5m/pxl
	480×640 pxl	0	100	
GF-2 (fused)	1000∼3000 pxl	0	10	0.8m/pxl
	480×640 pxl	0	100	

sizes are fixed to 640 × 480 pixels. These image slices cover most types of ground features of the human living environment. Regions such as glacier, desert, and Gobi are not considered in our data set at this time. Detailed information about these image slices are listed in Table II.

Before the whole captioning process, the three visible bands of a GF-2 fused image slice are extracted and stacked into a pseudo RGB image. Since the raw pixel data of GF-2 is 10-bit depth, all images have been converted to 8-bit images by the ENVI software. It should be noted that although our experiments are made on 3-channel RGB images, our model itself does not restrict the data type. In fact, the change on data format (including the number of band and the number of quantization bits) only affects the configuration of the first layer, or more precisely, only affects the channel number of its convolutional filters. To work on different image types, we only need some slight changes in the channel number of the first layer's convolutional filter to match the image band number.

In our data set, we only use a part of the Google Earth image slices for training, and all the rest (including all GF-2 images) are used for testing (see Table II for details). In the training set, the large slices with ground-truth bounding-box labels are used to train the key instance detection model, while for training the envi-element and landscape analysis model, the slices are further cut and resized into a set of smaller slices with the same size as the network's perceptive field. For Google Earth images, all key instances' bounding-boxes and the class label of the ground elements such as airport, harbor, ocean, and city have been annotated manually. Some ambiguous instances are not labeled and are excluded from our data set. These refer to those ships and airplanes whose length is smaller than 20 pixels and those oilpots whose radius is smaller than 10 pixels. Those instances whose bounding-boxes are partially outside the image are also excluded. The key instances detected in these regions will not be taken into account either as a "false positive" or as a "true positive." Eliminating all these invalid instances, a total of 2772 oilpots, 2244 ships, and 1853 airplanes are labeled for experiments.

To detect instances with different directions, each positive training window is rotated several times. The negative training samples are a set of instance-free randomly sampled windows from the data set. In this way, more than 40k positive windows and 400k negative windows are finally used for training. To detect the instances of different sizes, each image slice is downsampled at 4 scales with the unified downsampled rate 1.2 at the very beginning of key instances detection process. For envi-element and landscape analysis, each image slice is downsampled at a ratio of 3.0 to obtain a larger perceptive field, and finally 180k and 167k patches are used for training their softmax weights. The weights of three subtasks are retrained several times using the augmented data set (initial data + hard examples) to produce the final training results. In (11), we set learning rate $\mu = 0.001$, regularization coefficient $\alpha = 0.01$. The batch number of (10) is set to $n^+ = n^- = 10$. Our learning algorithm shows high time efficiency. The objective function value (10) of the three corresponding instance classes only takes several minutes to converge to a constant.

B. Overall Results Statistics

Fig. 8 shows some typical image slices of the test set and the corresponding descriptions automatically generated by our method. All the slices have been resized to a suitable size for display. To quantitatively evaluate the descriptions generated by our method, we follow the subjective evaluation criterion introduced in [15], where the accuracy of the description is divided into four levels: "without errors," "with minor errors," "related to the image," and "unrelated to the image." An example of this criterion is shown in Fig. 9. For each test image slice, the evaluation process is made by 10 different persons by specifying the image to one of the four levels. The evaluation results' statistics are shown in Table III. The results suggest the strong knowledge-transfer ability of our method. Despite the fact that we do not use any GF-2 images for training, satisfactory results for GF-2 test images are still obtained.

It should be noted that we do not use any objective evaluation methods such as BLUE [40] or ROUGE [41], which have been commonly used in the current natural image captioning literature. This is because BLUE and ROUGE are all based on words' matched relevance between the generated sentence and the ground-truth sentence. Their score may strongly relate to the annotators' expression style, such as degree of simplicity, different tense, voice, and mood. A supervised method such as LSTM can learn a similar expression style of annotators by training a large number of annotated sentences. Although it is true that learning-based method may have more advantages in generating more flexible sentences with a higher score than the template-based method, especially for the description of natural images, nevertheless, in some particular application scenarios where the template-based language generation methods are more frequently used, the above criteria may have large bias.

Fig. 10 shows two typical failure examples of our method. Actually, the problem lies in the image understanding stage. There are no airplanes in Fig. 10(a), but our algorithm gives some related descriptions since the detector produces a number of false alarms. Fig. 10(b) is misidentified as a coastal area since the scene texture of some regions resembles ocean waves. One possible solution to this problem is to make use of the context information. For example, a ship is unlikely to appear on a piece of farmland, and an airplane is also unlikely to be standing on the sea. In this way, the output of the detector and classifier can be calibrated by integrating the information about different levels. This will be a part of our future work.

Since the quality of the final description can be highly affected by the performance of the first stage, to make deeper insights into the description failure, some objective evaluation

Fig. 8. Example image slices and sentences generated by our method for (a)–(c) Google Earth images and (d)–(f) GF-2 images. (a) This figure shows an urban coastal area. This figure consists of some buildings and structures. There are a lot of oilpots and many ships in this figure. (b) There is one large airplane in this picture. The airplane is located at the upper part of the picture. (c) This picture shows a countryside area which contains a piece of farmland. (d) This picture shows an urban area which contains some land structures. There are many oilpots and two large ships. The ship is located close to the oilpots. (e) This image shows an urban area. This image consists of some land structures. (f) This image shows a mountain area.

Fig. 9. Examples of the subjective evaluation criterion introduced in [15].

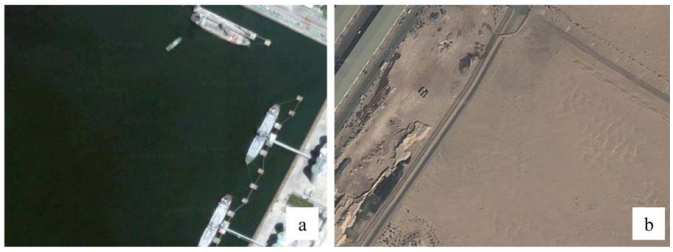


Fig. 10. Two typical failure examples of the generated sentences. The errors are highlighted in red. (a) Urban area with some ships and several airplanes. (b) Coastal area.

TABLE III
SUBJECTIVE EVALUATION RESULTS USING THE CRITERION INTRODUCED IN [15]

Source	Google Dataset	GF-2 Dataset
without errors	63%	48%
with minor errors	22%	23%
related to the image	10%	19%
unrelated to the image	5%	10%

criteria of the first stage are introduced, such as the precision–recall curves and confusion matrix. Fig. 11 shows some key instances of the detection results. Detection maps for oilpot, ship, and airplane are marked as red, yellow, and blue and are overlaid on the original image for display. All the slices have been resized to a suitable size for display. Although there are some hard instances such as ships adjacent to the land and airplane adjacent to the terminal building, we can see that the network has successfully highlighted most of these instances and suppressed the undesired backgrounds. In Fig. 11(b), a small piece of region between the airplanes is falsely detected as an oilpot. The precision and recall rates of all the test images

Fig. 11. (a)–(f) Some key instance detection results of Google Earth image slices. (g)–(j) GF-2 image slices.

are counted as follows:

$$\text{precision} = N_{\text{tp}}/(N_{\text{tp}} + N_{\text{fp}})$$
$$\text{recall} = N_{\text{tp}}/(N_{\text{tp}} + N_{fn}) \tag{14}$$

where N_{tp} represents the number of true-positives, N_{fp} represents the number of false-positives, and N_{fn} represents the number of false-negatives. The precision and recall rates of the Google Earth test images are shown in Fig. 12. When computing precision and recall, the exact target bounding-box should be generated to compare with the ground-truth. Since our work focuses on captioning, rather than detection, we did not draw the exact bounding-box for each detected target; instead, only original responses on the feature map are overlaid on the input image. There are two steps going from the score maps of a certain scale to the target bounding-boxes. In the first step, each pixel of score map whose response is larger than a certain threshold is compared with its neighbors. If the current pixel value equals the max value of its neighbors, it is then identified as the center of a candidate with the corresponding window size. To further reduce the number of overlapped bounding-boxes, only the bounding-box with the largest score is retained, while other overlapped bounding-boxes are deleted.

For envi-elements and landscape analysis, their confusion matrix of Google Earth images is shown in Fig. 13. The ground element of the highest accuracy is "ocean" (99%) while that of the lowest accuracy is "harbor" (79%). This is mainly because the training samples of "harbor" are slightly insufficient than others, which can be further improved by adding more samples to its training set. Fig. 12 and Fig. 13 suggest an overall high

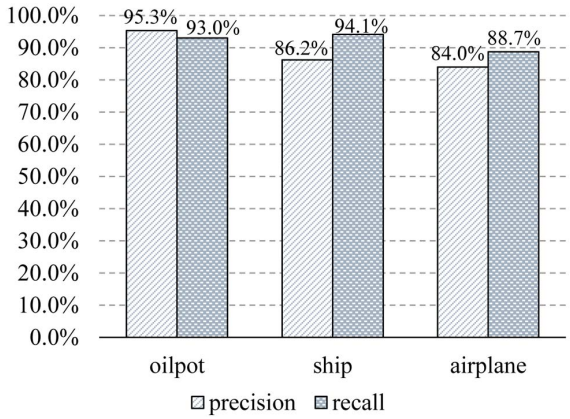


Fig. 12. Overall detection result statistics on Google Earth test images.

accuracy of our method. Although the analysis process cannot be simply seen as a classification process as some of these slices have multiple geographical semantics, as emphasized in Section I, here we still take the confusion matrix with the most salient semantic class as a reference of their performance.

C. Computational Efficiency

We test our method on an Intel i7 PC with 16G RAM and Nvidia Taitan Z graphics card. The programming platform is MATLAB 2015a + matconvnet-1.0 beta20 [42]. We use the GPU to accelerate the whole FCN model, which makes the program run 5∼20 times faster than in a single CPU thread.

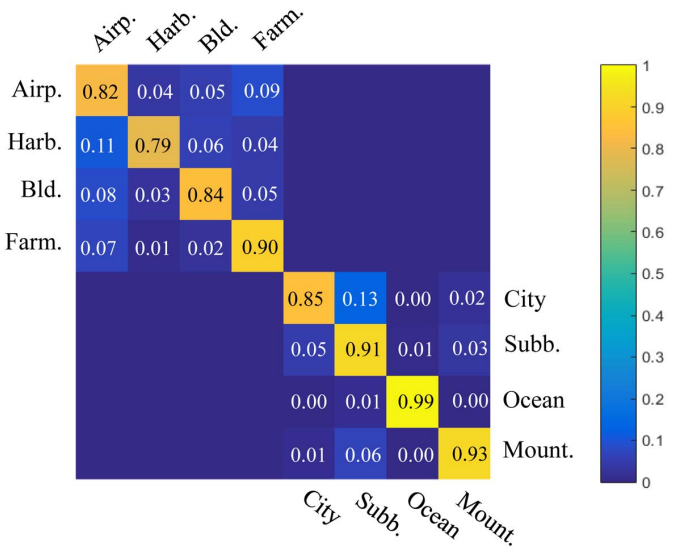


Fig. 13. Confusion matrix of envi-element and landscape for Google Earth image.

TABLE IV
DETAILED EXECUTION TIME OF THE PROPOSED METHOD

Image Size (pxl)	GPU Time (s)	CPU Time (s)
640 × 480	stage1 time = 1.25 stage2 time = 0.01 total time = 1.26	stage1 time = 8.12 stage2 time = 0.01 total time = 8.13
2400 × 1600	stage1 time = 5.71 stage2 time = 0.13 total time = 5.84	stage1 time = 139.10 stage2 time = 0.13 total time = 139.22

The average computational time of each stage of our method is counted.

For a 640 × 480 sized input image, our method only takes about only 1s to finish the whole captioning process. For a 2400 × 1600 image, the execution time is less than 6s. By building our model based on a unified FCN structure, our method shows high computational efficiency. This is mainly because FCN is able to reduce the computational redundancy at overlapping windows by computing a whole convolutional feature map for the entire input image and then deal with each feature vector extracted from the shared feature map. For larger sized remote sensing image, e.g., 10 000 × 10 000 pixels, we suggest dividing the image into discrete blocks due to the limited graphics memory. Table IV shows the detailed execution time of different stages.

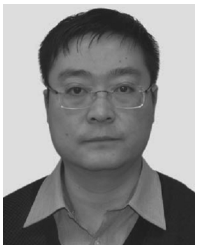
VI. CONCLUSION

We investigate an interesting question of can a machine automatically generate humanlike language description of remote sensing image. Our preliminary conclusion on this question is optimistic and we have proposed a remote sensing image captioning framework, where the experimental results on Google Earth and GF-2 images have demonstrated the superiority and transfer ability of the proposed method. Although there are still some spaces to improve our method, we consider it as a novel and promising framework that is fast, robust, and structurally compact. Our future work will focus on calibrating the semantics of different geographical levels and integrating more types of ground features in order to generate language descriptions with richer semantics.

REFERENCES

[1] H. Fang *et al.* (2015). "From captions to visual concepts and back." [Online]. Available: https://arxiv.org/abs/1411.4952
[2] L. Zhang, L. Zhang, and B. Du, "Deep learning for remote sensing data: A technical tutorial on the state of the art," *IEEE Geosci. Remote Sens. Mag.*, vol. 4, no. 2, pp. 22–40, Jun. 2016.
[3] S. Paisitkriangkrai, J. Sherrah, P. Janney, and A. van den Hengel, "Semantic labeling of aerial and satellite imagery," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 9, no. 7, pp. 2868–2881, Jul. 2016.
[4] Z. Zou and Z. Shi, "Ship detection in spaceborne optical image with SVD networks," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 10, pp. 5832–5845, Oct. 2016.
[5] L. Zhang, Z. Shi, and J. Wu, "A hierarchical oil tank detector with deep surrounding features for high-resolution optical satellite imagery," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 8, no. 10, pp. 4895–4909, Oct. 2015.
[6] Z. An, Z. Shi, X. Teng, X. Yu, and W. Tang, "An automated airplane detection system for large panchromatic image with high spatial resolution," *Optik-Int. J. Light Electron Opt.*, vol. 125, no. 12, pp. 2768–2775, Jun. 2014.
[7] J. Han, D. Zhang, G. Cheng, L. Guo, and J. Ren, "Object detection in optical remote sensing images based on weakly supervised learning and high-level feature learning," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 6, pp. 3325–3337, Jun. 2015.
[8] Z. Shi, X. Yu, Z. Jiang, and B. Li, "Ship detection in high-resolution optical imagery based on anomaly detector and local shape feature," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 8, pp. 4511–4523, Aug. 2014.
[9] V. Risojević and Z. Babić, "Unsupervised quaternion feature learning for remote sensing image classification," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 9, no. 4, pp. 1521–1531, Apr. 2016.
[10] B. Demir and L. Bruzzone, "Hashing-based scalable remote sensing image search and retrieval in large archives," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 2, pp. 892–904, Feb. 2016.
[11] A. Karpathy and L. Fei-Fei. (2014). "Deep visual-semantic alignments for generating image descriptions." [Online]. Available: https://arxiv.org/abs/1412.2306
[12] K. Xu *et al.* (2015). "Show, attend and tell: Neural image caption generation with visual attention." [Online]. Available: https://arxiv.org/abs/1502.03044
[13] J. Jin, K. Fu, R. Cui, F. Sha, and C. Zhang. (2015). "Aligning where to see and what to tell: Image caption with region-based attention and scene factorization." [Online]. Available: https://arxiv.org/abs/1506.06272
[14] J. Mao, W. Xu, Y. Yang, J. Wang, and A. L. Yuille. (2014). "Explain images with multimodal recurrent neural networks." [Online]. Available: https://arxiv.org/abs/1410.1090
[15] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan. (2014). "Show and tell: A neural image caption generator." [Online]. Available: https://arxiv.org/abs/1411.4555
[16] A. Farhadi *et al.*, "Every picture tells a story: Generating sentences from images," in *Proc. Eur. Conf. Comput. Vis.*, 2010, pp. 15–29.
[17] S. Li, G. Kulkarni, T. L. Berg, A. C. Berg, and Y. Choi, "Composing simple image descriptions using Web-scale N-grams," in *Proc. 15th Conf. Comput. Natural Lang. Learn.*, 2011, pp. 220–228.
[18] M. Mitchell *et al.*, "Midge: Generating image descriptions from computer vision detections," in *Proc. 13th Conf. Eur. Chapter Assoc. Comput. Linguistics*, 2012, pp. 747–756.
[19] Y. Yang, C. L. Teo, H. Daumé, III, and Y. Aloimonos, "Corpus-guided sentence generation of natural images," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2011, pp. 444–454.
[20] G. Kulkarni *et al.*, "BabyTalk: Understanding and generating simple image descriptions," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 12, pp. 2891–2903, Dec. 2013.
[21] P. Kuznetsova, V. Ordonezz, T. L. Bergz, and Y. Choi, "TREETALK: Composition and compression of trees for image descriptions," *Trans. Assoc. Comput. Linguistics*, vol. 2, no. 10, pp. 351–362, 2014.

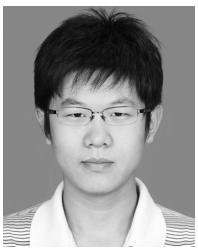
[22] P. Kuznetsova, V. Ordonez, A. C. Berg, T. L. Berg, and Y. Choi, "Collective generation of natural image descriptions," in *Proc. 50th Annu. Meeting Assoc. Comput. Linguistics, Long Papers*, vol. 1. 2012, pp. 359–368.
[23] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.
[24] K. Simonyan and A. Zisserman. (2014). "Very deep convolutional networks for large-scale image recognition." [Online]. Available: https://arxiv.org/abs/1409.1556
[25] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 580–587.
[26] J. Dai, Y. Li, K. He, and J. Sun. (2016). "R-FCN: Object detection via region-based fully convolutional networks." [Online]. Available: https://arxiv.org/abs/1605.06409
[27] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. (2015). "You only look once: Unified, real-time object detection." [Online]. Available: https://arxiv.org/abs/1506.02640
[28] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 3431–3440.
[29] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun. (2013). "OverFeat: Integrated recognition, localization and detection using convolutional networks." [Online]. Available: https://arxiv.org/abs/1312.6229
[30] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. (2014). "Semantic image segmentation with deep convolutional nets and fully connected CRFs." [Online]. Available: https://arxiv.org/abs/1412.7062
[31] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman. (2014). "Return of the devil in the details: Delving deep into convolutional nets." [Online]. Available: https://arxiv.org/abs/1405.3531
[32] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 91–99.
[33] J. Donahue *et al.*, "DeCAF: A deep convolutional activation feature for generic visual recognition," in *Proc. Int. Conf. Mach. Learn.*, 2014, pp. 647–655.
[34] O. A. B. Penatti, K. Nogueira, and J. A. dos Santos, "Do deep features generalize from everyday objects to remote sensing and aerial scenes domains?" in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, Jun. 2015, pp. 44–51.
[35] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, Jun. 2009, pp. 248–255.
[36] I. Goodfellow, A. Courville, and Y. Bengio, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016. [Online]. Available: http://www.deeplearningbook.org
[37] L. Zhou and E. Hovy, "Template-filtered headline summarization," in *Proc. Annu. Meeting Assoc. Comput. Linguistics Workshops*, 2004, pp. 56–60.
[38] S. Channarukul, S. W. McRoy, and S. S. Ali, "DOGHED: A template-based generator for multimodal dialog systems targeting heterogeneous devices," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics Human Lang. Technol. Demonstrations*, vol. 4. 2003, pp. 5–6.
[39] M.-M. Cheng, Z. Zhang, W.-Y. Lin, and P. Torr, "BING: Binarized normed gradients for objectness estimation at 300 fps," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 3286–3293.
[40] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "BLEU: A method for automatic evaluation of machine translation," in *Proc. Annu. Meeting Assoc. Comput. Linguistics*, 2002, pp. 311–318.
[41] C.-Y. Lin and E. Hovy, "Automatic evaluation of summaries using N-gram co-occurrence statistics," in *Proc. North Amer. Chapter Assoc. Comput. Linguistics Human Lang. Technol.*, vol. 1. 2003, pp. 71–78.
[42] *MatConvNet—Convolutional Neural Networks for MATLAB*, accessed on Oct. 30, 2016. [Online]. Available: http://www.vlfeat.org/matconvnet/



Zhenwei Shi (M'13) received the Ph.D. degree in mathematics from the Dalian University of Technology, Dalian, China, in 2005.

He was a Post-Doctoral Researcher with the Department of Automation, Tsinghua University, Beijing, China, from 2005 to 2007. He was a Visiting Scholar with the Department of Electrical Engineering and Computer Science, Northwestern University, Evanston, IL, USA, from 2013 to 2014. He is currently a Professor and the Chairman of the Image Processing Center, School of Astronautics, Beihang University, Beijing. He has authored or co-authored over 100 scientific papers in refereed journals and proceedings, including the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, the IEEE TRANSACTIONS ON NEURAL NETWORKS, the IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING, the IEEE GEOSCIENCE AND REMOTE SENSING LETTERS, and the IEEE CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION. His research interests include remote sensing image processing and analysis, computer vision, pattern recognition, and machine learning.

Dr. Shi serves as an Associate Editor of the *Infrared Physics and Technology*.



Zhengxia Zou received the B.S. degree from the School of Astronautics, Beihang University, Beijing, China, in 2013, where he is currently pursuing the Ph.D. degree with the Image Processing Center, School of Astronautics.

His research interests include machine learning and remote sensing image target detection.