IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received March 30, 2021, accepted April 8, 2021, date of publication April 22, 2021, date of current version July 2, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3074937

Improving Visual Reasoning Through Semantic Representation

WENFENG ZHENG[1], (Member, IEEE), XIANGJUN LIU[1], XUBIN NI[1], LIRONG YIN[2], AND BO YANG[1], (Member, IEEE)

[1]School of Automation, University of Electronic Science and Technology of China, Chengdu 610054, China
[2]Department of Geography and Anthropology, Louisiana State University, Baton Rouge, LA 70803, USA

Corresponding author: Lirong Yin (yin.lyra@gmail.com)

This work was supported in part by the Sichuan Science and Technology Program under Grant 2021YFQ0003 and Grant 2019YJ0189.

ABSTRACT In visual reasoning, the achievement of deep learning significantly improved the accuracy of results. Image features are primarily used as input to get answers. However, the image features are too redundant to learn accurate characterizations within a limited complexity and time. While in the process of human reasoning, abstract description of an image is usually to avoid irrelevant details. Inspired by this, a higher-level representation named semantic representation is introduced. In this paper, a detailed visual reasoning model is proposed. This new model contains an image understanding model based on semantic representation, feature extraction and process model refined with watershed and u-distance method, a feature vector learning model using pyramidal pooling and residual network, and a question understanding model combining problem embedding coding method and machine translation decoding method. The feature vector could better represent the whole image instead of overly focused on specific characteristics. The model using semantic representation as input verifies that more accurate results can be obtained by introducing a high-level semantic representation. The result also shows that it is feasible and effective to introduce high-level and abstract forms of knowledge representation into deep learning tasks. This study lays a theoretical and experimental foundation for introducing different levels of knowledge representation into deep learning in the future.

INDEX TERMS VQA, the semantic net, visual reasoning, deep learning.

I. INTRODUCTION

Visual Question Answering (VQA) combines natural language processing with digital image processing. The general process for solving a VQA problem is to take the image and the corresponding question as input and finally get the answer [1]. The problems which are similar to VQA require more interdependent inference steps to solve.

The research is mainly divided into the non-deep learning model and deep learning model. Most non-deep learning models are based on Bayesian theory. Some researchers [2]–[13] proposed a Bayesian framework, predicting the type of answer to a question and generating an answer. Mateusz *et al.* proposed the multi-world question and answer model in 2014, proposed the DAQUAR data set, and modeled visual question and answer as SWQA model [14]. Kafle *et al.* proposed a Bayesian framework for solving visual Q&A in 2016. The framework generates the answer based on the prediction of the answer type [2]. As shown from the introduction above, those non-deep learning models performed poorly.

With the improvement of deep learning research, its research on the VQA field is becoming more mature. Aishwarya *et al.* proposed a production of the drestm Q + Norm I model in 2015 [15]. In the same year, Mengye *et al.* [16] proposed the VIS+LSTM model. Based on this, three variant models were constructed 2-VIS+BLSTM, IMG+BOW, and FULL. In 2015, Mateusz *et al.* proposed a neural-image-QA model, which is also known as the Neural query model [6], [17]. The feature of this model is that it can generate answers of variable length. Shih's work attempts to introduce attention mechanisms into VQA tasks [18].

The model's input is an image feature of a question, possible answers, and a series of automatically selected candidate areas. The work of Noh *et al.* USES the parameter

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

prediction network to generate dynamic parameters. Change VGG-16 [19] into A CNN network with three fully connected layers. This paper's main contribution is the combination of CNN and Dual Path Networks (DPN) to process ImageQA tasks. Jiasen Lu *et al.* proposed a multi-mode attention model of co-attention + Question Hierarchy to solve VQA tasks [5], [20]. The basic idea of co-attention is to use the image to get the problem's attention weight and use it to get the image area's weight. Question Hierarchy is a three-layer hierarchical structure of questions.

The first layer is the word level, which is used to represent each word as a word vector. The second layer is the phrase level, which adopts one-dimensional CNN to extract features. The third layer is question level, and RNN is used to encode the whole question. Andreas *et al.* proposed a modular neural network model in 2016 [21]. The model first USES a syntactic parser to split questions into corresponding linguistic substructures. The modularized neural network is selected automatically according to the structure. They validated the model on the VQA dataset [15], and the results reached a leading level at that time. On this basis, the inference and execution procedure for visual reasoning was proposed by William Li *et al.* [22].

DeepMind came up with relational networks for relational reasoning in 2017. The whole model has only two types of network, CNN and the full connection layer, with a straightforward structure. However, the experimental results are outstanding, reaching the leading level in the CLVER data set [23]. Although deep learning models have made significant progress, it still has a large gap. Current visual reasoning models are mainly to take pictures or image features as input. The difference is that humans use high-level, abstract information to describe the relationship. In the latest research, it was found that other researchers have also realized this problem and tried to use semantic information as input [24]. Therefore, in this research, we use the semantic representation of the image as input to explore whether introducing high-level semantic representation can be better. If better results can be gained, this idea can be introduced to other computer vision areas, even deep learning.

This research's primary goal is to replae image features and explore whether this replacement can lead to a better result. Semantic representation is readable and straightforward and can be further processed. Finally, the split semantic representation was combined with a particular rule to observe. The primary process of our research is as follows

A method of replacing the visual features of images with the semantic representation of images as a visual reasoning model input is proposed in this paper. This paper's method is based on the baseline model prosed in previous works [1]. We combine the parts and techniques involved in the method and construct weakened image processing and natural language processing. Based on the previous point [1], we improve the general method of extracting the general image's semantic representation. After extracting the image's semantic representation, two understanding modules are combined to form a high accuracy coding and decoding model of the representation vectors. Finally, we can test the final model and compare it with existing works by others.

II. DATASET

The Feifei Li team proposed the CLEVR data set used as the main data source of this study [25]. The data set contains three-dimensional images rendered with Blender and questions that require multiple steps of reasoning to get answers. The scene of CLEVR is completely generated by the program, and every detail is controllable, so there is a minimal bias. At the same time, the data set also provides a reasoning process, which is convenient for researchers to construct a reasoning system close to human logic [25]. This data set is used to analyze a variety of modern visual reasoning systems and is currently the mainstream data set in the field of visual reasoning.

CLEVR dataset contains 100,000 rendered 3D images and approximately 1 million auto-generated questions, of which 853,000 are different [10], [25].

III. METHOD

A. OBJECT DETECTION AND RECOGNITION

The main task of object detection and recognition is a computer vision task to distinguish the objects and irrelevant parts of the image, determine whether there are potential targets in the region, identify target types and determine the location of the target. There are some mainstream methods of target detection and recognition, such as R-CNN [26], Fast R-CNN [27], Faster R-CNN [27], Mask R-CNN [28], etc. which are all combined with the deep learning model with the region and high-performance classifier to complete the detection and recognition task. The advantage of this model is that it can obtain high detection and recognition accuracy, while the disadvantage is that the implementation of the whole model needs a lot of computation, which requires a high demand on hardware, and it is difficult to achieve real-time processing and has a long delay. After considering these, Liu *et al.* proposed a regression-based target detection and recognition method SSD similar to YOLO [29].SSDS are end-to-end models, so all identification and detection models can be trained and executed over a network. SSD made some improvements on the basis of YOLO. First, SSD introduced the anchor mechanism in Fast R-CNN [27], adding the idea of regional Suggestions on the basis of regression. Secondly, instead of using the global features of images, SSD uses the deep features around each target to detect and identify the target, extracts features from the feature maps of different depths of the deep neural network, and then uses these features to predict the target by regression. Therefore, SSD can make more judgments on a target by using multi-scale information and improve the accuracy without affecting the speed. The disadvantage of SSD is that it is sensitive to the size of the target object, and it is not as effective as the mainstream region-based recommendation method when making boundary box predictions for small objects.

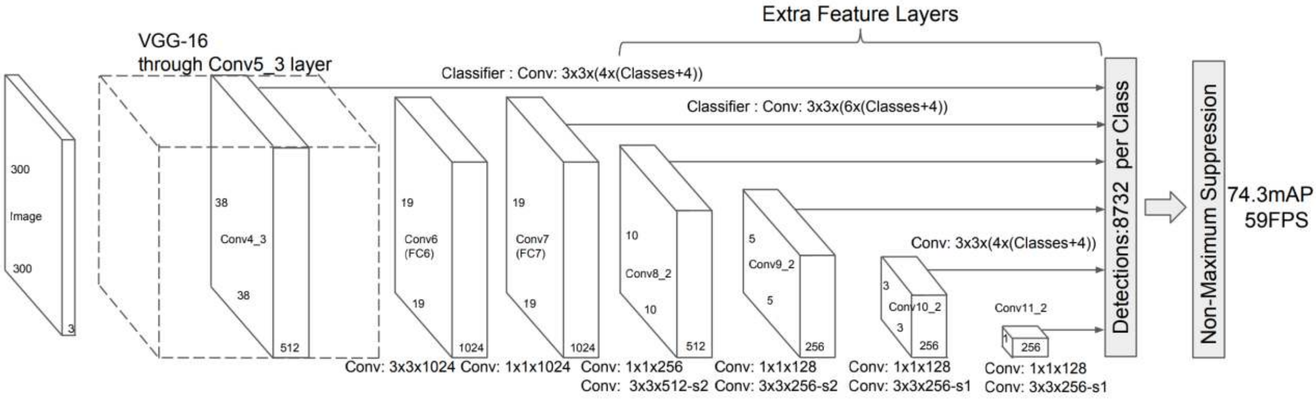

FIGURE 1. SSD Network [30], [31].

SSD algorithm can be divided into four parts: firstly, the depth features of the input images are obtained through the deep neural network; Then, according to the depth feature graph of different scales, different sizes of feature capture boxes are constructed to train with the real target frame as ground-truth. Then the features of the depth feature graph corresponding to the feature capture box are extracted to predict the target category and the real frame of the target in the capture box. At last, non-maximum Suppression (NMS) is used to filter the best prediction results. During training, SSD only receives images as input, and the categories and positions of objects in the images are used as training labels. No other information is needed. The structure of the model is shown in Fig. 1.

The input of the SSD model is an RGB image in the size of 300px × 300px, and then the feature of the whole image is extracted by vgg-16. In order to extract multi-scale features, multiple CNN layers with different scales were added after VGG-16. As shown in Fig. 1, the subsequent feature map used for identification and detection includes conv4_3, conv7, conv8_2, conv9_2, conv10_2, and conv11_2, which are used for multi-scale feature extraction and result prediction. The loss function of the model is:

$$L(x, c, l, g) = \frac{1}{N} \cdot (L_{conf}(x, c) + \alpha \cdot L_{loc}(x, l, g)) \qquad (1)$$

N represents the number of matching boxes. The function of x is to mark whether the corresponding feature fetching box contains the corresponding target, and $x_{ij}^{p} = \text{Error! Bookmark not defined.}$ indicates whether the i_{th} box matches the boundary box of the j_{th} target of the p type object. The x sets as 1 when matches, and as 0 when not match. So if the $\sum_i x_{ij}^{p} \geq 1$ shows the target bounding box have more than or equal to 1 box to match.

SSD requires the training set to have a label for each image and each object in the image. The tag includes the type of object and its mask. Usually, this part needs to be manually marked, which requires a lot of labor and time. In this study, we introduced the watershed and u-net method to reduce the labor and time cost.

1) WATERSHED

Considering the small number of training samples for target detection, most mainstream target detection and recognition models [26]–[30] need to learn a lot of relevant representations, including but not limited to classification representation and location regression representation. Therefore, these deep learning exercises need to rely on a large number of training samples. When the sample size is small, the training results are poor. In this case, deep learning combined with traditional image processing is used to detect the target in this study. Firstly, a deep learning model is used to obtain an intermediate result according to the original image, and then a traditional image processing algorithm is used to process the intermediate result to obtain the final target detection result. In addition, the work of object recognition is transferred to object attribute extraction, which reduces the representation of the object detection model.

In traditional computer vision, watershed segmentation is one of the standard methods to separate overlapping objects from images. The watershed algorithm is an image region segmentation algorithm whose essence is morphological segmentation method based on topology theory. The basic principle is to connect the points with similar positions and grayscale to form a closed interval. The basic steps of image segmentation using a watershed segmentation algorithm can be divided into three parts. Firstly, the color image needs to be converted into a grayscale image, and then the gradient of the grayscale image is calculated. Finally, the watershed algorithm is applied according to the gradient image.

2) U-NET

The more direct idea is to change the distribution of the Distance Map to improve the over-segmentation phenomenon in watershed segmentation. The traditional solution is to estimate the location of tags to guide the segmentation of subsets [32], [33], but it is not a good operation in practice [34].

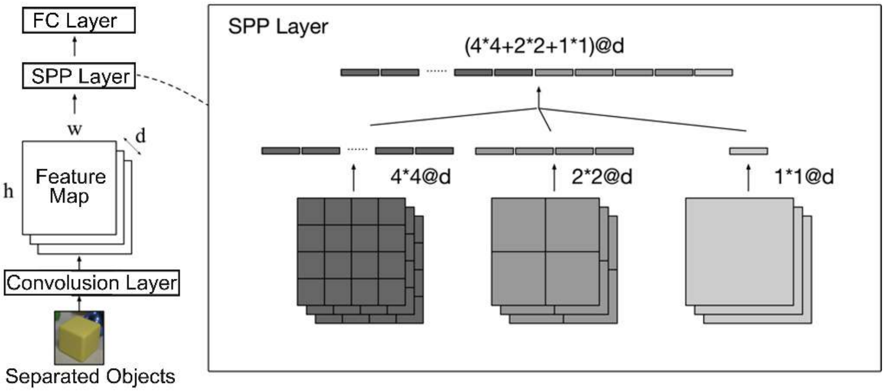

FIGURE 2. Maximum pooling of space pyramid.

The idea [34] is to train a neural network to learn the direction from the point inside each entity to the boundary, and then train a neural network to learn the energy level of the point inside the entity according to the direction diagram and finally apply the energy level to the watershed algorithm.

Based on this idea, in this experiment, there can be only one marker inside each entity, and the Distance Map value of the pixel inside each entity is the Distance from this point to the marker point. After that, the Distance Map value inside each entity is normalized to ensure that the center Distance of the entity is the largest and the edge is the smallest. In this way, we can ensure that the dividing watershed is the boundary of the entity. In the previous section, u-net was first used to obtain the mask of the original picture, and then the Distance Map of the mask was calculated. Finally, watershed segmentation was applied. The U-net will be directly used to learn the above construction method of Distance Map, and then the U-net results will be directly applied to watershed segmentation.

B. FEATURE VECTOR LEARNING

1) PYRAMIDAL POOLING OF SPACE

Since the picture of CLEVR is a 3D scene, the size of the bounding box, which is detected by the target, is not fixed because of the perspective; that is, the image size of each object that is cropped is different. In the study of this section, the models used to extract the properties of objects are mostly multi-layer CNNs plus multi-layer fully connected layer neural networks. For the convolutional layer, only one convolution kernel is slid on the image during operation. The parameters of the model are independent of the size of the input. For any size image, it can be treated as input, but the size of the output feature image will follow The size of the input image changes. The fully connected layer needs to connect each input pixel, so the parameters of the fully connected layer are related to the size of the input. Therefore, for a general classification model, it is necessary to scale or crop the object detected by the target to the same size in order to fix the number of parameters of the entire fully connected layer. However, scaling or cropping the image will result in loss and distortion of the image information to a certain extent, which limits the final recognition accuracy. Therefore, in this section, we will use the Spatial Pyramid Pooling (SPP) in SPP-Net of He *et al.* [35] to make the neural network accept images of different sizes as input.

As shown in Fig. 2, when inputs a picture, the method divides a picture with different scales. In the figure, the input feature map is divided into three different scales of 1×1, 2×2, and 4×4, and finally, a total of $4\times4+2\times2+1\times1 = 21$ blocks are obtained. A feature is then extracted from each block for a total of 21 dimensions. After the pooling operation, there are various pooling operations, including but not limited to average pooling and maximum pooling [36]. The maximum pooling of the space pyramid is to use the maximum pooling operation for these 21 feature blocks. SPP can convert an image of any size into a fixed-size feature block. Each of the divided scales is called a layer of a gold tower, and a feature block size is called a window size. For a layer of the pyramid, it is necessary to pool with a window size of size (w/n, h/n) to output a feature of n × n.

When the input of the multi-layer neural network is an image of any size, the conventional convolution and pooling can be performed until the network is down to several layers, and the SPP layer can be used when the connection layer is to be connected. Thereby, feature maps of any size can be converted into feature vectors of fixed dimensions.

2) RESIDUAL NETWORK

The residual network was proposed by He Kaiming *et al.* in "Deep Residual Learning for Image Recognition" in 2015 [37]. The residual network belongs to the deep convolutional network and won the championship in the three images of ImageNet's image classification, detection, and positioning. The advantage of the residual network is that it is easier to optimize than the traditional convolutional neural network, and the residual network solves the degradation problem caused by the increased depth of the neural network,

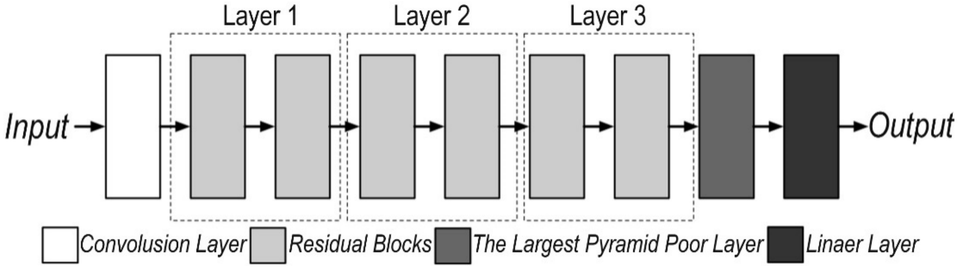

FIGURE 3. Generic property extraction model that can be used to extract the shape, color, and material of an object.

so the performance of the model can be further improved by simply increasing the depth of the network.

The first problem that may arise with increasing the depth of the neural network is the gradient disappearance or gradient explosion. This problem was solved smoothly by the Batch Normalization (BN) structure proposed by Ioffe and Szegedy [38] The reason why BN is useful is that the BN layer can normalize the output of each layer so that the gradient can remain dimensionally stable after backpropagation. However, when the number of network layers is increased to a certain extent, the training accuracy will reach saturation, which is called the problem of accuracy degradation. This decline is not due to the disappearance of the gradient or over-fitting, but because the network is too complex, it is difficult to achieve the ideal error rate with unconstrained training. Currently, widely used training methods such as SGD, AdaGrad, and RMSProp are challenging to achieve theoretical optimal convergence results after the network depth becomes more extensive than before. However, at the same time, it can be proved that in the case of an ideal training method, a deeper network will have a better effect on a shallower network. Assume that an additional layer of the network is added behind network A to form a new network B. If the additional network only performs an identity mapping on the output of A, then the error rates of network A and network B are equal, that is, the depth of the network will not be deepened. Make the results worse.

In order to achieve such an identity mapping, He Kaiming proposed a residual structure. The entire module has a branch that connects the input and output in addition to the normal network layer so that the final output is the sum of the output of the network layer and the input of the network layer, where $H(x) = F(x) + x$. Where x is the input, $F(x)$ is the output of the network layer, called the residual term, and $H(x)$ is the output of the entire structure. When $F(x) = 0, H(x) = x$ is an identity map. The reason why such a structure is designed is that if it is difficult to learn $H(x) = x$ directly from the network, the parameter initialization in each layer network is generally biased to 0 so that the redundant layer learns $F(x) = 0$. The updated parameters can converge faster, and learning $F(x) = 0$ is much simpler than learning $H(x) = x$. The residual structure converges the entire redundant network towards the direction of the identity map through an artificially constructed structure so that the final accuracy does not decrease due to the increase of the network depth.

3) THE ATTRIBUTE EXTRACTION

The attribute extractor needs to be able to extract from each object, segmented the attributes that each object is useful for answering the current question. In order to facilitate the addition or deletion of requirements, the final consideration is to design a discrete test attribute extractor, design a corresponding extraction module for each attribute, and use a unified API call. This allows hot-swapping without retraining the network when the data set changes or new attributes need to be added. For the CLEVR dataset, the attributes that are good for solving the problem are shape, color, size, position, and material. In this study, the training set of the attribute extractor is constructed by using the generated code of the CLEVR data set. Each sample of the training set contains a single object clipped from the graph and its corresponding information.

The same network detection can be used for shapes, colors, and materials. In general, a simple classification model can be used to achieve the goal, that is, a multi-layer CNN plus a multi-layer fully connected layer neural network to perform classification tasks. Here, in order to improve the final recognition accuracy and the training convergence speed, multiple residual blocks are used as the feature extraction layer, and the full connection layer and the LogSoftmax layer are connected later to obtain the classification result. The model structure is shown in Fig. 3. After the input object image is convolved through a layer, it passes through three Layers, and each Layer contains two residual blocks. Then, through the spatial pyramid pooling layer, the influence of different input image sizes on the input dimensions of the subsequent fully connected layer is avoided. The output is finally obtained through the fully connected layer and the LogSoftmax layer.

For the size of the target object, considering the perspective rule of the near and far, it is difficult to judge the size of the object from a single picture. It is necessary to consider the size of the object in the figure and its position. Therefore, the input to the model for judging the size of the object is a 4-dimensional vector, which includes the ratio of the row coordinates of the object to the height of the image, the ratio

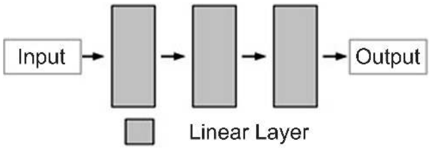

FIGURE 4. Structure of dimension extraction module.

of the column coordinates of the object to the width of the image, the ratio of the height of the object to the height of the image, and the ratio of the width to the width of the image. The structure of the model is straightforward and consists of three layers of fully connected layers. Except for the last fully connected layer followed by the LogSoftmax layer, the remaining fully connected layers are followed by the ReLU activation function. The structure is shown in Fig. 4.

For the position of the target object, since the outer bounding box of each object output by the target detection model already contains the coordinate information of the object, this part can be directly obtained without an additional training model.

C. QUESTION UNDERSTANDING

1) ENCODING MODEL BASED ON PROBLEM EMBEDDING

The basic design idea of the problem-solving module based on problem embedding is to embed a sequence of natural language questions into a vector space using an encoder. The semantic representation and problem of the scene can then be embedded and stitched to obtain the answer as the input of the multi-layer fully connected neural network. The advantage of this method is that it does not require additional reasoning annotation data and the neural network adaptive learning to effective representation through the joint training of semantic representation and problem representation.

This paper has introduced the basic idea of RNN and the LSTM network commonly used in the field of natural language processing. In the actual model construction of this paper, the LSTM variant Gated Recurrent Unit (GRU) is used because the structure of the GRU is more straightforward than the LSTM [39]–[41]. By combining the forgotten gate and the input gate into a single update gate, the number of gates is one less than the LSTM. A few matrix multiplications. GRU can save much time when the training data is large. GRU is different in implementation details from LSTM, but the basic idea and deployment process is similar. The main difference between GRU and LSTM lies in the decision of GRU to control both the forgetting gate and the update status unit. The update formula is as (2):

$$h_i^{(t)} = u_i^{(t-1)} h_i^{(t-1)} + \left(1 - u_i^{(t-1)}\right) \sigma\left(b_i + \sum_j U_{i,j} x_j^{(t)} + \sum_j W_{i,j} r_j^{(t-1)} h_j^{(t-1)}\right) \tag{2}$$

where u represents the update gate, which can be used to linearly control any dimension, changing the influence of the previous moment state and the current moment input on the current state is as (3):

$$u_i^{(t)} \sigma\left(b_i^u + \sum_j U_{i,j}^u x_j^{(t)} + \sum_j W_{i,j}^u h_j^{(t)}\right) \tag{3}$$

r represents the reset gate, which controls which parts of the current state are used to calculate the next target state, and introduces additional nonlinear effects between the past state and the future state as (4).

$$r_i^{(t)} \sigma\left(b_i^r + \sum_j U_{i,j}^u x_j^{(t)} + \sum_j W_{i,j}^u h_j^{(t)}\right) \tag{4}$$

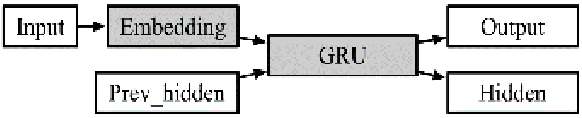

FIGURE 5. The basic units of the encoder.

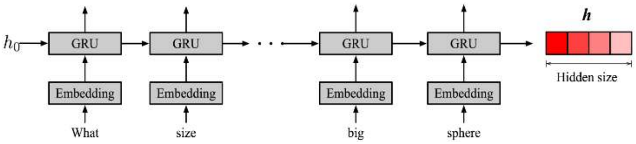

FIGURE 6. The process of encodin.

The basic unit of the Encoder constructed in this study is shown in Fig. 5. The number of hidden layers in the GRU is one, and the number of hidden layer units is 256. The word embedding layer has an input size of 93 and an output size of 256.

The encoding process of the Encoder is shown in Fig. 6. The initial hidden vector is h0. Each word in the input sequence is first embedded in a word embedding layer and then embedded in the word and GRU unit. A hidden layer output is used as input to the current GRU unit. The GRU gets output and an output of the hidden layer and then proceeds to the next round of input until all elements in the sequence have been entered. The hidden layer output h of the last time series is the embedding of the problem.

2) DECODING MODEL BASED ON MACHINE TRANSLATION

The work in this section is to construct a learnable reasoning model-based inference model [22]. The essential requirement is to make the reasoning process transparent and use the semantic representation of the image as the model's input. From a global perspective, the model receives the semantic representation s of an image and a question q as input to answer a. Unlike Johnson's model, the final result of the model of this study is based on the semantic network [22]. The language is directly obtained, similar to the operation of the database query. In the middle process, the model predicts a reasoning step z needed to solve the current problem according to the problem, and then takes the semantic representation

TABLE 1. The network structure of the Inference layer.

Order	Layer	Output Size
(1)	Linear(2|n| + |e_q|, 512)	m*512
(2)	ReLU	m*512
(3)	Linear(512, 512)	m*512
(4)	ReLU	m*512
(5)	Sum	512
(6)	Linear(512, 512)	512
(7)	ReLU	512
(8)	Linear(512, 512)	512
(9)	ReLU	512
(10)	Linear(512, 512)	512
(11)	ReLU	512
(12)	Dropout	512
(13)	Linear(512,|$\mathcal{A}$|)	|$\mathcal{A}$|
(14)	LogSoftmax	|$\mathcal{A}$|

of the image as the input of s and finally obtains the corresponding answer.

The whole system is divided into three parts: the image semantic representation extraction module extracting the image's semantic representations from the image x; a program generator for predicting the program z that may be involved according to the problem q; and an execution The engine, $\alpha = \varphi(s, z)$, executes the program z on the image semantic representation s to predict the answer a. The program generator is trained using an encoding-decoding model. In our research, the input to the execution engine is an abstract, interpretable semantic representation. We can manually design deterministic functions directly based on the semantic network to achieve specific functions.

After extracting the semantic representation of the image represented by the semantic network, since the programs included in the execution engine are deterministic programs, there is no need to convert the semantic network into a vector form, but directly use three the original form representation of the tuple, the content can be expressed in natural language. For example, the semantic representation of the scenario can be described as:

$$\begin{bmatrix} \begin{bmatrix} ['object1','shape','sphere'] \\ \vdots \\ ['object1','position', [0.49, 0.24]] \end{bmatrix} \\ \vdots \\ \begin{bmatrix} ['object3','shape','cylinder'] \\ \vdots \\ ['object3','position', [0.7, 0.57]] \end{bmatrix} \end{bmatrix} \tag{5}$$

In the CLEVR data set, each question is represented by natural language and functional programs. The functional program representation can accurately determine the basic reasoning skills required to answer each question and is ultimately stored in the text in the form of a pre-order traversal of the program tree.

The example of the problem-to-program mapping and the inference skills included in CLEVR are shown in Fig. 7.

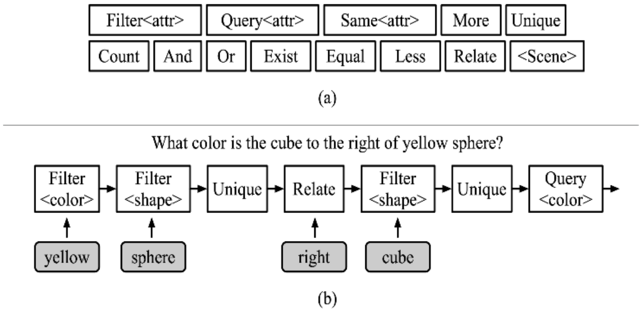

FIGURE 7. CLEVR data sets required basic reasoning skills. (a) basic reasoning skills. (b) examples of natural language mapping to program trees.

Fig. 7(a) is the basic reasoning skills involved in this study. The scene refers to the semantic representation of the scene, and only it can be used as the leaf node of the program tree. We can more accurately recover the final program structure by limiting each reasoning skill's input and output types. Fig. 7(b) shows an example of a natural language mapping to a program tree.

The program generator $z = \pi(q)$ function is to predict the function z involved in the problem from the natural language question q. Specifically, the natural language sequence is converted into a pre-order traversal sequence of the program. Such problems are very similar to machine translation, that is, translation from one language to another. This way, we can implement the program generator using the standard LSTM sequence pair sequence model [42]. Cyclic neural networks are characterized by memory so that they can predict the state after the previous state in the sequence. Memory is vital to the language model because different words have different meanings in different contexts. So cyclic neural networks are very suitable for use in language models.

The attention mechanism proposed by Bahdanau *et al.* utilizes the output information of each step in the encoding process [43]. The attention mechanism allows the network to have different input weights for each part of the input sequence during decoding, rather than relying solely on the content vector. During the decoding process, each output depends on each of the previous hidden state and each corresponding hidden state of the output sequence, that is as the following (6) and (7):

$$s_i = f(y_{i-1}, s_{i-1}, c_i) \tag{6}$$

$$p(y_i|y_1, y_2, \ldots, y_{i-1}) = g(y_{i-1}, s_{i-1}, c_i) \tag{7}$$

where ci is a context vector, which is a weighted sum of all hidden states h1, h2, $\cdots$, hT of the input sequence in (8)

$$c_i = \sum_{j=1}^{T} \alpha_{ij} h_j \tag{8}$$

The attention weight parameter αij in (8) is not a fixed value but is calculated by a neural network in (9) and (10).

$$\alpha_{ij} = \frac{exp(e_{ij})}{\sum_{k=1}^{T} exp(e_{ik})} \tag{9}$$

TABLE 2. Accuracy of different models on the CLEVR dataset.

	Accuracy of existing models on different question types (%)					
	Existence	Quantity	Quantity Comparing	Characteristic inquiry	Characteristic Comparing	Total
LSTM[22, 25]	61.8	42.5	69.3	36.525	51.125	47
CNN+LSTM[22, 25]	68.2	47.8	69.2	48.925	54.7	54.3
CNN+LSTM+SA[22, 25, 45]	68.4	57.5	66.633333	87.7	52.05	69.8
CNN+LSTM+SA+MLP[22, 25, 45]	77.9	59.7	73.566667	80.925	70.825	73.2
Human[22, 25]	96.6	86.7	85.666667	95	96	92.6
IEP[22, 25]	97.1	92.7	98.633333	98.15	98.9	96.9
RN[23]	97.8	90.1	93.6	97.9	97.1	95.5
Proposed method accuracy-1K (%)	82.348	80.713	82.313	84.047	84.927	82.943
Proposed method accuracy-5K (%)	92.173	88.815	92.703	93.21	92.891	92.001
Proposed method accuracy-10K (%)	96.963	90.665	94.424	98.176	99.673	96.139

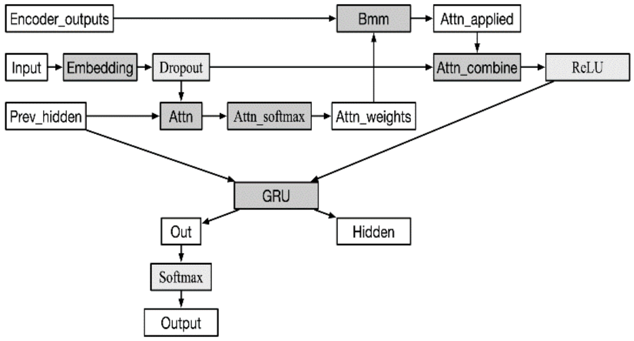

FIGURE 8. A decoder structure with an attention mechanis.

$$e_{ij} = a(s_{i-1}, h_j) \tag{10}$$

The neural network A receives the previous output hidden states (i-1), and the input is hidden state hj as an output to obtain e_{ij}, and then obtains the weight a_{ij} by normalization.

Focus mechanisms are added to build a more complex model based on the underlying decoder. The network input is first converted into a word vector. The word vector and the hidden state are stitched together. Then a fixed-length sequence is output through the linear layer plus the Softmax activation layer. This sequence is the attention sequence. Each number size indicates the importance of attention. Then the output of the encoding process and the attention weight are obtained by batch matrix multiplication. Finally, the result is spliced together with the network input through a linear. The layer transforms the dimension into a dimension accepted by the recurrent neural network, takes it as an input to the network, and finally gets the final output through the network. The model structure of the decoder with the attention mechanism added is shown in Fig. 8.

IV. RESULT

Experimental results of the strongly supervised reasoning model based on the reasoning process are shown in Table 2. In this experiment, the program generator was trained using all samples in CLEVR's official dataset. It was 99.7% accurate in the test set. Training samples that train image segmentation for Distance u-net were not provided in CLEVR's official dataset. This part of the dataset was generated by remodeling the source code of the CLEVR dataset. Due to the time-consuming rendering of the scene, only the dataset with a sample size of 1000 was built at the beginning of the experiment. It was found that the average accuracy rate of the model's final answer was only 82.94% when the program generator was 99.7% accurate. The bottleneck of the model lies in the accuracy of image segmentation.

To this end, the sample size of the image segmentation training set was further expanded in subsequent experiments. Finally, image segmentation data sets with a sample size of 1000 (1K), 5000 (5K), and 10000 (10K) were constructed, respectively. The Distance, the u-net segmentation model, was trained respectively. The final results are shown in Table 2. Notably, CLEVR's original dataset contains 70,000 images. In this study, the image interpretation section used at most 1/7 of the original dataset. In Fig. 9, when the semantic representation of the training scene of the 1K data set is used to extract the model, the final result is still about 14% different from the current leading results [23] and [22]. However, it is ahead of all existing models [22], [25], [45], and 9% more accurate than the results of the best existing model. When training the image understanding model with 5K data sets, the final accuracy is 92%, only 4.9% behind the current leading result [22], and close to the human test level of 92.6%. When 10K data sets are used to train the image understanding model, the final result is further improved, reaching 96.14%. This result exceeds the final result of [23] by 0.64%, which is only 0.76% different from that of [22] and exceeds the human test level by 3.54%. After that, we tried to increase the training amount of the image understanding model but found that the final result did not change significantly. It can be considered that the performance of the model has reached a bottleneck under the current design details.

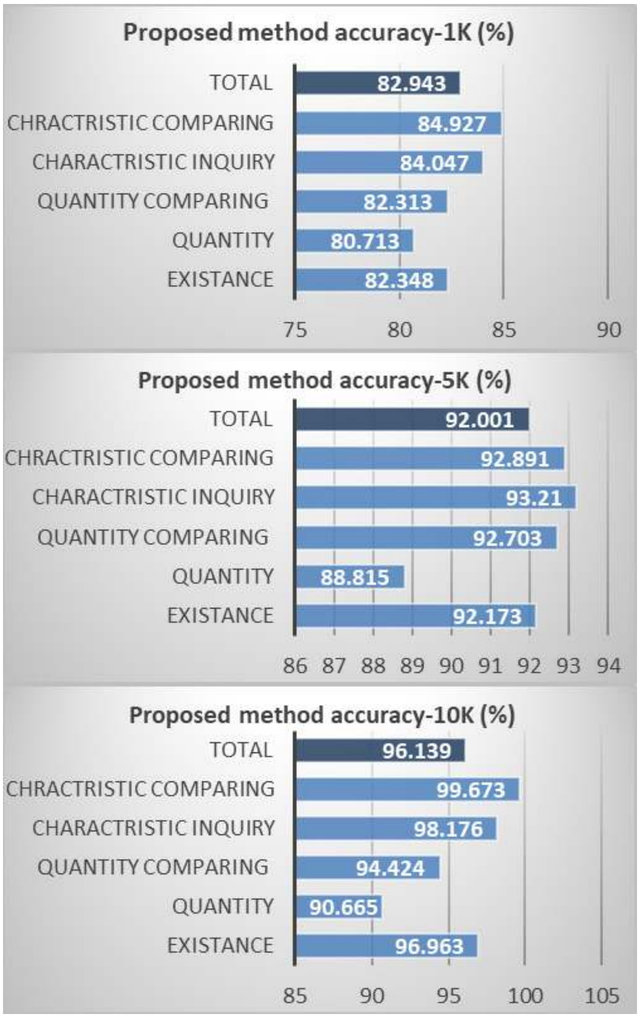

FIGURE 9. The proposed model's accuracy of different question types under 1k train set, 5k train set, and 10k train se.

V. DISCUSSION

The visual realization of the model is mainly aimed at the visual reasoning model based on the inference process. The model can visualize the visual reasoning from information extraction to representation formation to reasoning, which is convenient for humans to understand the mechanism and process of the whole system work. Troubleshoot the bottleneck of the model. Simultaneously, the visual reasoning model based on the inference process decouples the independent functional modules as much as possible. Only the corresponding components can be replaced without the entire network retraining when the visual reasoning task is changed. Therefore, the visual reasoning model based on the reasoning process is more suitable for the actual production environment.

The semantic representation of the image is used as the input of the visual inference model. In the experiment, deep learning combined with traditional image processing is used to achieve target detection, but the effect is limited. Therefore, in the future, we can improve the accuracy of this part by increasing the amount of data or adopt a better segmentation model such as Mask R-CNN. Fine-tune transfer learning can also be done using segmentation models trained on other datasets. Or data enhancements to the original data set.

Based on the former point [1], the general method of extracting semantic representation in the form of the semantic network from general images is perfected. The three elements of the semantic network are node, attribute class, and attribute value. Therefore, to build a semantic network, it is necessary to find the entities (nodes) in the image first, consider the target detection of the original image, and extract the objects contained therein. Then it is necessary to judge the useful feature types (attribute classes) according to the task types and then use a neural network to judge the coping attributes (attribute values). Finally, a semantic network describing the whole image is constructed. In the future, we can further optimize the organization of semantic representation and design more effective inference models for inference tasks.

After extracting the semantic representation of the image, the question and the image are taken as the input of the reasoning model to get the final answer. The construction of the inference model is mainly based on two different usage scenarios. One is to build a process-based reasoning model by using the supervision information about the reasoning process provided by the data set to increase the transparency and understandability of the model. The other is an end-to-end reasoning model that considers the more general case and simplifies the complexity of the model. Finally, the advantages and disadvantages of the model are analyzed and summarized. In the future, the combination of reinforcement learning technology and reinforcement learning training after a small amount of supervised information learning can be considered to reduce the dependence on data sets.

VI. CONCLUSION

The main contributions of this research to related fields are as follows:

1) Considering that the semantic network can express knowledge deeply, including the characteristics of entity structure, hierarchy, and causal relationship between entities, semantic representation instead of image visual-feature as a visual reasoning model is proposed. Make the reasoning process more transparent and increase the comprehensibility of the model. It is convenient to decouple the system from the bottleneck of the analysis model.

2) Improve U-Net so that the output of U-Net is not a mask of the scene object but a Distance Map for watershed segmentation. The output of U-Net can be directly used for watershed segmentation, equivalent to passing deep neural networks. The effect of watershed segmentation is optimized so that the method can obtain satisfactory image segmentation results under the condition of fewer samples.

3) Use the model based on the attention mechanism to transform the natural language into a potential logical representation, which can be used to map natural language into a program tree-like machine translation.

This paper demonstrates how the semantic representation can be used as an input and verifies that changing the representation of the image can further improve system performance. After replacing the visual feature, the accuracy of non-relational questions was significantly improved. Then the semantic vector was pre-processed by constructing a relation matrix. The semantic representation effect is competitive compared to visual representation, and the semantic representation is simple and easy to carry out other processes.

After analysis, it was summarized that introducing semantic information was equivalent to a feature selection and extraction before input. The selected features were useful for answering questions. Compared with the feature extraction of CNN, the semantic information is more accurate and less redundant. So, it is easier to find the precise relationship when handling relational reasoning.

REFERENCES

[1] X. Ni, L. Yin, X. Chen, S. Liu, B. Yang, and W. Zheng, "Semantic representation for visual reasoning," in *Proc. MATEC Web Conf.*, vol. 277. Les Ulis, France: EDP Sciences, 2019, Art. no. 02006.

[2] K. Kafle and C. Kanan, "Answer-type prediction for visual question answering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 4976–4984.

[3] W. Zheng, X. Liu, and L. Yin, "Sentence representation method based on multi-layer semantic network," *Appl. Sci.*, vol. 11, no. 3, p. 1316, Feb. 2021.

[4] S. Liu, Y. Gao, W. Zheng, and X. Li, "Performance of two neural network models in bathymetry," *Remote Sens. Lett.*, vol. 6, no. 4, pp. 321–330, Apr. 2015.

[5] Y. Ding, X. Tian, L. Yin, X. Chen, S. Liu, B. Yang, and W. Zheng, "Multi-scale relation network for few-shot learning based on meta-learning," in *Proc. Int. Conf. Comput. Vis. Syst.* Cham, Switzerland: Springer, 2019, pp. 343–352.

[6] Y. Tang, S. Liu, Y. Deng, Y. Zhang, L. Yin, and W. Zheng, "An improved method for soft tissue modeling," *Biomed. Signal Process. Control*, vol. 65, Mar. 2021, Art. no. 102367.

[7] S. Liu, L. Wang, H. Liu, H. Su, X. Li, and W. Zheng, "Deriving bathymetry from optical images with a localized neural network algorithm," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 9, pp. 5334–5342, Sep. 2018.

[8] Y. Tang, S. Liu, Y. Deng, Y. Zhang, L. Yin, and W. Zheng, "Construction of force haptic reappearance system based on geomagic touch haptic device," *Comput. Methods Programs Biomed.*, vol. 190, Jul. 2020, Art. no. 105344.

[9] X. Chen, L. Yin, Y. Fan, L. Song, T. Ji, Y. Liu, J. Tian, and W. Zheng, "Temporal evolution characteristics of PM2.5 concentration based on continuous wavelet transform," *Sci. Total Environ.*, vol. 699, Jan. 2020, Art. no. 134244.

[10] W. Zheng, X. Li, L. Yin, and Y. Wang, "The retrieved urban LST in Beijing based on TM, HJ-1B and MODIS," *Arabian J. Sci. Eng.*, vol. 41, no. 6, pp. 2325–2332, Jun. 2016.

[11] X. Li, W. Zheng, D. Wang, L. Yin, and Y. Wang, "Predicting seismicity trend in southwest of China based on wavelet analysis," *Int. J. Wavelets, Multiresolution Inf. Process.*, vol. 13, no. 2, Mar. 2015, Art. no. 1550011.

[12] X. Li, W. Zheng, L. Yin, Z. Yin, L. Song, and X. Tian, "Influence of social-economic activities on air pollutants in Beijing, China," *Open Geosci.*, vol. 9, no. 1, pp. 314–321, Aug. 2017.

[13] L. Yin, X. Li, W. Zheng, Z. Yin, L. Song, L. Ge, and Q. Zeng, "Fractal dimension analysis for seismicity spatial and temporal distribution in the circum-pacific seismic belt," *J. Earth Syst. Sci.*, vol. 128, no. 1, p. 22, Feb. 2019.

[14] M. Malinowski and M. Fritz, "A multi-world approach to question answering about real-world scenes based on uncertain input," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 27, 2014, pp. 1682–1690.

[15] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. L. Zitnick, and D. Parikh, "VQA: Visual question answering," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 2425–2433.

[16] M. Ren, R. Kiros, and R. Zemel, "Exploring models and data for image question answering," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 28, 2015, pp. 2953–2961.

[17] M. Malinowski, M. Rohrbach, and M. Fritz, "Ask your neurons: A neural-based approach to answering questions about images," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1–9.

[18] K. J. Shih, S. Singh, and D. Hoiem, "Where to look: Focus regions for visual question answering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 4613–4621.

[19] K. Simonyan, A. Vedaldi, and A. Zisserman, "Deep inside convolutional networks: Visualising image classification models and saliency maps," 2013, *arXiv:1312.6034*. [Online]. Available: https://arxiv.org/abs/1312.6034

[20] J. Lu, J. Yang, D. Batra, and D. Parikh, "Hierarchical question-image co-attention for visual question answering," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 29, 2016, pp. 289–297.

[21] J. Andreas, M. Rohrbach, T. Darrell, and D. Klein, "Neural module networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 39–48.

[22] J. Johnson, B. Hariharan, L. Van Der Maaten, J. Hoffman, L. Fei-Fei, C. L. Zitnick, and R. Girshick, "Inferring and executing programs for visual reasoning," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2989–2998.

[23] A. Santoro, D. Raposo, D. G. T. Barrett, M. Malinowski, R. Pascanu, P. Battaglia, and T. Lillicrap, "A simple neural network module for relational reasoning," in *Proc. NIPS*, 2017, pp. 4974–4983.

[24] K. Yi, J. Wu, C. Gan, A. Torralba, P. Kohli, and J. Tenenbaum, "Neural-symbolic VQA: Disentangling reasoning from vision and language understanding," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 31, 2018, pp. 1031–1042.

[25] J. Johnson, B. Hariharan, L. van der Maaten, L. Fei-Fei, C. L. Zitnick, and R. Girshick, "CLEVR: A diagnostic dataset for compositional language and elementary visual reasoning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2901–2910.

[26] 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.

[27] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1440–1448.

[28] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2017, pp. 2961–2969.

[29] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 779–788.

[30] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single shot multibox detector," in *Proc. Eur. Conf. Comput. Vis.* Springer, 2016, pp. 21–37.

[31] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, Dec. 2015.

[32] F. Meyer, "Topographic distance and watershed lines," *Signal Process.*, vol. 38, no. 1, pp. 113–125, Jul. 1994.

[33] V. Grau, A. U. J. Mewes, M. Alcaniz, R. Kikinis, and S. K. Warfield, "Improved watershed transform for medical image segmentation using prior information," *IEEE Trans. Med. Imag.*, vol. 23, no. 4, pp. 447–458, Apr. 2004.

[34] P. Dokladal, R. Urtasun, I. Bloch, and L. Garnero, "Segmentation of 3D head MR images using morphological reconstruction under constraints and automatic selection of markers," in *Proc. Int. Conf. Image Process.*, vol. 3, 2001, pp. 1075–1078.

[35] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 9, pp. 1904–1916, Sep. 2015.

[36] Y.-L. Boureau, F. Bach, Y. LeCun, and J. Ponce, "Learning mid-level features for recognition," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 2559–2566.

[37] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[38] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. Int. Conf. Mach. Learn. (PMLR)*, 2015, pp. 448–456.

[39] S. Lai, L. Xu, K. Liu, and J. Zhao, "Recurrent convolutional neural networks for text classification," in *Proc. AAAI Conf. Artif. Intell.*, vol. 29, no. 1, 2015, pp. 1–7.

[40] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Trans. Signal Process.*, vol. 45, no. 11, pp. 2673–2681, Nov. 1997.

[41] R. J. Williams and D. Zipser, "A learning algorithm for continually running fully recurrent neural networks," *Neural Comput.*, vol. 1, no. 2, pp. 270–280, Jun. 1989.

[42] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 27, 2014, pp. 3104–3112.

[43] J. K. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, "Attention-based models for speech recognition," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 28. Red Hook, NY, USA: Curran Associates, 2015, pp. 577–585.

[44] R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L.-J. Li, D. A. Shamma, M. S. Bernstein, and L. Fei-Fei, "Visual genome: Connecting language and vision using crowdsourced dense image annotations," *Int. J. Comput. Vis.*, vol. 123, no. 1, pp. 32–73, May 2017.

[45] Z. Yang, X. He, J. Gao, L. Deng, and A. Smola, "Stacked attention networks for image question answering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 21–29.

WENFENG ZHENG (Member, IEEE) received the Ph.D. degree in earth exploration and information technology from the Chengdu University of Technology, in 2008. Since 2008, he has been an Associate Professor with the School of Automation Engineering, University of Electronic Science and Technology of China. He has published more than 80 articles, five books, and authorized more than 30 Chinese national invention patents. His research interests include environmental science, information technology, and artificial intelligent. He is a member of the Association for Computing Machinery, the America Association Geographer, the American Geophysical Union, and the China Association of Inventions.

XIANGJUN LIU is currently pursuing the bachelor's degree in automation engineering with the University of Electronic Science and Technology of China. She is also a Research Assistant with the Research Center of Machine Perception and Intelligent Systems, University of Electronic Science and Technology of China. She is mainly responsible for machine learning method experiments and tests, and the sorting and analysis of experimental data, to publish two academic articles. Her research interests include machine learning and artificial intelligence.

XUBIN NI received the master's degree from the University of Electronic Science and Technology of China. He has published three articles. His main research interests include control science and engineering, automation, and intelligent perception, such as visual question answering, visual reasoning, and semantic representation.

LIRONG YIN received the Bachelor of Science degree in geography information science from the University of Iowa, and the Master of Science degree in geography from Louisiana State University, where she is currently pursuing the Ph.D. degree with the Department of Geography and Anthropology. She has study interest in remote sensing, server weather and climate change, coastal environment, natural hazard, and coupled human and natural dynamic system. She has experienced the artificial intelligence studies and machine learning techniques, geo-data processing, and information analysis skills. She is well experienced in programming and database design as a geo-analyst. She has published more than 20 articles.

BO YANG (Member, IEEE) received the master's degree in pattern recognition and intelligent systems from Shandong University, in 2003, and the Ph.D. degree in control theory and control engineering from Shanghai Jiao Tong University, in 2008. From 2010 to 2012, he was a Visiting Scholar with The Hong Kong Polytechnic University. Since 2014, he has been an Associate Professor with the School of Automation Engineering, University of Electronic Science and Technology of China.