Journal of Artificial Intelligence Research 64 (2019) 181-196 Submitted 01/18; published 01/19

Multi-scale Hierarchical Residual Network for Dense Captioning

Yan Tian TIANYAN@ZJGSU.EDU.CN
Xun Wang WX@ZJGSU.EDU.CN
Jiachen Wu 616113970@QQ.COM
Ruili Wang PROF.RUILI.WANG@GMAIL.COM
Bailin Yang YBL@ZJGSU.EDU.CN
School of Computer Science and Information Engineering,
Zhejiang Gongshang University, Hangzhou 310014, P.R.China

Abstract

Recent research on dense captioning based on the recurrent neural network and the convolutional neural network has made a great progress. However, mapping from an image feature space to a description space is a nonlinear and multimodel task, which makes it difficult for the current methods to get accurate results. In this paper, we put forward a novel approach for dense captioning based on hourglass-structured residual learning. Discriminant feature maps are obtained by incorporating dense connected networks and residual learning in our model. Finally, the performance of the approach on the Visual Genome V1.0 dataset and the region labelled MS-COCO (Microsoft Common Objects in Context) dataset are demonstrated. The experimental results have shown that our approach outperforms most current methods.

1. Introduction

Image captioning is a task of automatically generating a sentence to describe an image, while dense captioning replaces a constant number of object categories with a broader set of visual concepts by using phrases to describe object(s) in an image, which is foundational to many important applications, like semantic image search, visual intelligence in chatting robots, photo and video sharing on social media, and aid for people perceiving the world around them. Recently, dense captioning has received much interests from the research community.

Previous approaches predict region captions from image feature maps by combining the Convolution Neural Network (CNN) (LeCun, Bottou, Bengio, & Haffner, 1998) and the Recurrent Neural Network (RNN) (Werbos, 1988). However, the performance is difficult to improve because of some bottlenecks including: 1) object detection is still an open issue in computer vision; 2) mapping from an image feature space to a description space is nonlinear and multimodel. Deep networks have the potential to learn the nonlinear mapping well, but they are hindered by the vanishing/exploding gradients problems (He, Zhang, Ren, & Sun, 2016a).

Recently, the residual learning network (He et al., 2016a) and its extensions have shown competitive capability in nonlinear and multimodel classification because they alleviate the vanishing gradients by adding shortcut layers (residual layers), which strengthen the gradient to propagate through a network with considerable depth.

However, despite the exploration of residual learning in classification, there have been few works to introduce residual learning into sequential prediction tasks such as dense captioning. In this paper, we focus on finding the optimal way to combine residual learning and recurrent neural networks in the sequential prediction task.

We propose a new method based upon hourglass-structured (Newell, Yang, & Deng, 2016) residual learning for dense captioning. First, multi-scale feature maps are obtained by an hourglass network, then extra fully connected layers are employed to obtain a fixed dimension feature, and lastly time-series word prediction is made by residual long-short-term memory (residual LSTM). The novelty of this paper is illustrated by the following.

- A new dense captioning approach is proposed that captures multi-scale object information by using an hourglass network. Dense connected networks are introduced into residual learning to increase network capacity, and discriminant feature maps are obtained by using residual learning.
- A new residual LSTM is proposed to decrease the vanishing/exploding gradients.
- The experimental results indicate that our approach is competitive when compared to some state-of-the-art approaches regarding dense captioning.

2. Related Work

In this section, we briefly review the work on image captioning and analysis the advantage and drawback of each approach, and then introduce the development of the deep residual learning.

2.1 Image Captioning

At present, the image captioning approaches are divided into the following categories.

Search-based Approaches. Search-based approaches refer to extracting the image feature and comparing it with all the image features in the dataset to select the most semantically similar sentences. The similarities between the sentence and the image are evaluated by utilizing an intermediate space in terms of the interaction between the scene and the object (Farhadi et al., 2010), where the sentences corresponding to the image features with high visual similarities were chosen as the generated sentences for the test image. An approach searches images in a dataset by utilizing the combination of the object, human, and background information, and associating the related sentences to the query image (Ordonez, Kulkarni, & Berg, 2011). An alignment model that aligned two modalities (i.e., the feature space and word space) through a multimodal embedding, and then, each word was independently predicted by a matching method (Karpathy & Li, 2015). Many annotation sentences are required in these approaches, which costs abundant human resources and makes it hard to scale up the sentence set. Besides, these approaches cannot create novel descriptions.

Sequence Learning-based Approaches. Sequence learning-based approaches are inspired by the success of sequence-to-sequence encoder-decoder frameworks in machine translation. Motivated by the human visual system, an attention model (Xu et al., 2015) was introduced into image captioning, which allowed for salient features to dynamically come to the forefront whenever needed. Later, this work was extended and an adaptive attention model with a visual sentinel (Lu, Xiong, Parikh, & Socher, 2017) was proposed that could decide whether to attend to the image or to the visual sentinel at each time step. While the attention model was becoming popular, it was found that attributes play a key role in image captioning. Therefore, the top-down and bottom-up approaches were combined with a semantic attention model (Xu et al., 2015). Later, high-level semantic information was extended and an image captioning framework LSTM-A (long-short-term memory with attributes) was presented by training it under an end-to-end manner (Yao, Pan, Li, Qiu, & Mei, 2017).

Template-based Approaches. Although sequence learning-based approaches can achieve encouraging accuracy in image captioning, they often omit details in the image, which can be solved in template-based approaches. Template-based approaches involve objects in the image first being detected, and then a language model is combined to provide a proper image description. Multiple instance learning based detectors and the maximum-entropy language model were used (Fang et al., 2015), and global semantics were captured by re-ranking caption candidates. To address the description and the detection issues together, fully convolutional localization network (FCLN) (Johnson, Karpathy, & Li, 2016) was developed to deal with the dense captioning task. Then, joint inference and context fusion were corporated to realize the accurate detection of each visual concept (Yang, Tang, Yang, & Li, 2017). All these approaches have considered the development in object detection, but the hard samples still cannot be detected. As a result, the multimodel mapping from the image space to the captioning space was not well modeled due to the hard sample scarcity.

2.2 Deep Residual Learning

When deeper networks started to converge, a degradation problem would arise: accuracy became saturated (which is unsurprising) and then degraded rapidly as the depth of network increasing.

Deep residual learning (He et al., 2016a) was introduced to address the degradation problem. They explicitly let each few stacked layers fit a residual map rather than a desired underlying map. Experiments showed that it is easier to optimize residual networks, and the accuracy could be gained due to a considerably increased depth.

Later, this work was extended and identity mappings was used as the after-addition activation and the skip connection so that one unit is able to propagate signal to any other unit directly in both the forward and the backward path (He, Zhang, Ren, & Sun, 2016b).

However, it was argued that training a very deep residual network had a problem of diminishing feature reuse (Zagoruyko & Komodakis, 2016), which made it very slow to train these networks that have either only a few blocks are able to learn valuable representations or most blocks sharing very little information with a limited contribution to the result. Hence, they increased the width and decreased the depth of residual networks, thus making network structures with wide residual networks (WRNs).

The optimization ability of residual networks was mined and RoR (residual networks of residual networks) was proposed (Zhang et al., 2017). RoR promoted learning capability of the residual network by adding level-wise shortcut paths to the original residual networks.

Recently, an hourglass-structured residual learning network was employed in object detection (Fu, Liu, Ranga, Tyagi, & Berg, 2017) and human pose estimation (Newell et al., 2016), but there was little evidence indicating that residual learning could be utilized on sequence prediction issues, such as dense captioning. Therefore, we proposed a new approach based on hourglass-structured residual learning for dense captioning during which both the CNN and RNN were explored to obtain accurate object descriptions.

3. Model Architecture

The design of our architecture is inspired by the need to capture multi-scale object information and then to describe each object by using natural language. Our architecture draws on the the developments in recent work on residual learning networks, object detection, and image captioning.

The framework is shown in Fig. 1. It is based on the hourglass model and starts with a 7×7 convolution layer Conv1 with stride 2 before a residual block and a max pooling layer brings the resolution down to 4×4 smaller than the input image. Subsequent residual blocks are utilized to acquire the discriminant feature, and an hourglass module is employed to analyze proposals in different receptive fields.

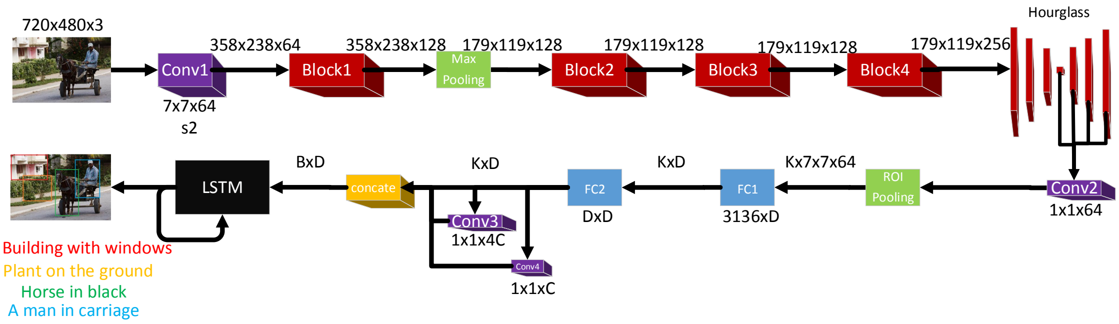


Figure 1: Illustration of the Framework. The numbers between volumes are the size of the tensor. K is the number of proposals that are obtained from an hourglass network, D is the hidden unit dimension, C is the object class number (including the background), and B objects with the highest confidence scores are selected and fed into LSTMs.

A convolution layer Conv2 is added before the region-of-interest (ROI) pooling which brings three advantages: 1) the number of feature channels is considerably reduced (from 512 to 64); 2) the sliding window classifier becomes simpler; 3) the modification of the kernel size in conv2 is reduced from 3×3 to 1×1 in order to restrict the receptive field of the convolution layer.

Information integration and cross-channel interaction are realized by adding two additional fully connected layers FC1 and FC2. Then, the region feature is used to generate detection scores and bounding box offsets by using 1×1 convolution layers Conv3 and Conv4.

The objects with the highest confidence scores are selected, and their region feature maps are concatenated and fed into LSTMs to generate region descriptions. Each LSTM unit predicts a word and this prediction is used as the input to the next LSTM unit. The first LSTM unit receives the object region feature maps as the input and produces the first word prediction.

3.1 Detection Model

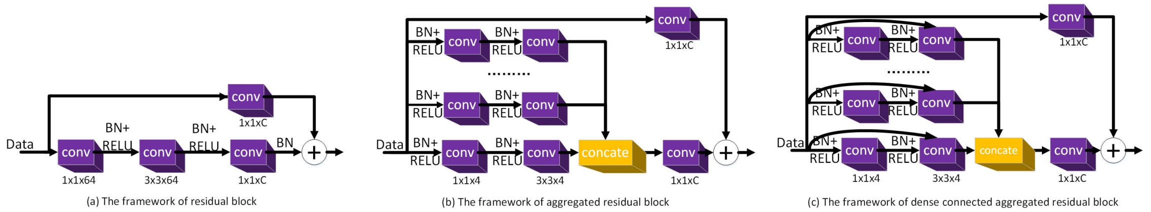
In this part, we introduce the detection model in the block level and the hourglass structure.

3.1.1 Block Structure

The original structure of a residual block is shown in Fig. 2(a). The basic convolution layer has 3×3 filters because a sequence of small convolution layers can replace a large convolution layer, and the order of residual blocks is Conv-BN-ReLU (Convolution - Batch Normalization - Rectified Linear Units). Another 1×1 convolution layer is added as a projection short-cut to match dimensions. C is an output dimension.

The original residual block has too many parameters (i.e. weights and biases) to be tuned. We propose a carefully crafted architecture that allows the increasing of the depth and width of the network while keeping the computational budget constant.

Firstly, inspired by ResNeXt (Xie, Girshick, Dollr, Tu, & He, 2017), we revise the original residual block by combining an inception model (Szegedy et al., 2015) which achieves compelling accuracy while keeping low theoretical complexity. The input is split into several lower-dimensional space (by convolutions), processed by a set of specialized filters, and merged by concatenation instead of summing up all the branches which is employed in ResNeXt. The split-transform-merge structure is expected to be close to the representational power of large and dense block while keeping a lower computational complexity. This new block is named as the aggregated residual block (AR-B), which is shown in Fig. 2(b). The order of the convolutional layer is BN-ReLU-Conv (Batch Normalization - Rectified Linear Units - Convolution). There are two contributions of pre-activation. First, the optimization is further eased (compared with the baseline ResNeXt) since the activation function after the concatenation is an identity mapping, and the signal can propagate directly from one unit to another. Second, using the Batch Normalization layer as pre-activation contributes to the regularization of the models. Although the Batch Normalization layer normalizes the signal in the original residual unit, it is soon merged to the shortcut; therefore the combined signal is not normalized. Then the unnormalized signal becomes the input of the next weight layer. In contrast, in our pre-activation version, we normalize the inputs to all weight layers. The final 1×1 convolution layer is also used as a projection short-cut.



(a) The framework of residual block

(b) The framework of aggregated residual block

(c) The framework of dense connected aggregated residual block

Figure 2: Architecture of the residual block and its extensions. (a) refers to the original residual block, and (b) the aggregated residual block, and (c) the dense connected aggregated residual block.)

Now that a shortcut in the network makes sense in dealing with the vanishing gradients, shortcuts are added between unconnected layers, which is similar to DenseNets (Huang, Liu, van der Maaten, & Weinberger, 2017). This revision alleviates the vanishing gradients, strengthens feature propagation, encourages reusage of features, and considerably reduces the number of parameters, to name a few. This new block is named as the dense connected aggregated residual block (DCARB), which is shown in Fig. 2(c). The feature maps of all preceding layers are used as inputs for the last convolutional layer in each branch, and the outputs of that are concatenated as the input into the subsequent layer. However, this block is different from DenseNets, where a fixed dimension output is necessary for the hourglass structure being constructed. Therefore, another 1×1 convolution layer is also employed as a projection shortcut.

3.1.2 Hourglass Structure

The hourglass structure in our framework is illustrated in Fig. 3 which is set up as below. Convolution layers with stride 2 are employed to transform features down to a lower resolution. At each down-sampling step, the network is divided into two parts and applies another DCARB at the original scale. After reaching the lowest scale, the network begins the processes of up-sampling and the combination of features across different resolutions. The up-sampling of the lower resolution is performed before an element-wise addition of the two sets of features, and the feature maps are outputted to obtain multi-scale proposals. The structure of the hourglass network is symmetric. Therefore, for each layer of decreasing resolution, there is a corresponding layer of increasing resolution.

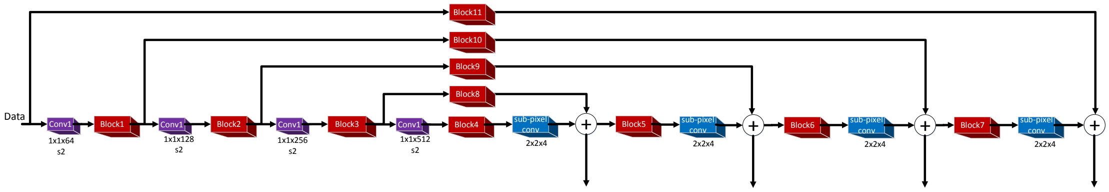


Figure 3: Architecture of the hourglass structure.

We make r the up-scaling ratio. In general, both the input feature maps and the output feature maps can have C channels; thus, they are represented as real-valued tensors of size $H \times W \times C$ and $rH \times rW \times C$, respectively.

The deconvolution layer (Zeiler, Taylor, & Fergus, 2011) multiplies each input pixel with a stride r filter, and accumulates the output windows. Nevertheless, reduction (summing) after convolution is computationally expensive. Hence, we substitute the deconvolution layer in the origin hourglass with the sub-pixel convolution layer (Shi et al., 2016) to improve efficiency. The shuffle operation is used in both ShuffleNet (Zhang, Zhou, Lin, & Sun, 2018) and our approach. The channel shuffle is employed to solve the side effects in ShuffleNet, while sub-pixeled convolution with 4 upscaling filters for each feature map is used to implement a fast deconvolution in our approach. The kernels are convolved with the input directly, and then low resolution feature maps $\mathbf{X}$ with r^2 channels are obtained with periodic shuffling to recreate a high one $\mathbf{Y}$.

$$\mathbf{Y} = PS(\mathbf{W}_L * \mathbf{X} + \mathbf{b}_L), \tag{1}$$

where the convolution weights are $\mathbf{W}_L$, and thus, it has the shape $n_{L-1} \times r^2C \times k_L \times k_L$. PS is the simple periodic shuffling that rearranges the elements of a tensor of shape $H \times W \times Cr^2$ to shape $rH \times rW \times C$. Periodic shuffling, as a type of a bit operation, can be implemented with extremely high efficiency, which makes this approach faster than up-pooling and deconvolution.

3.2 RNN Prediction Model

In this section, an approach to generate descriptions from region feature maps by directly maximizing the probability of the correct translation in an end-to-end fashion is proposed. Given that region feature maps are extracted from an image, the RNN transforms the variable length input into a fixed dimensional vector and utilizes this representation to decode it to a desired output sentence.

The probability of the correct description is maximized as below:

$$\theta = \arg\max_{\theta} \sum_{I,S} \log p(S|I;\theta), \tag{2}$$

where θ is a parameter in our model, I is the region feature maps extracted from an frame, and S is the correct transcription. The length of S is unbounded for the reason that it can represent any sentence. Thus, usually, the chain rule is applied in order to model the joint probability over $S_0, ..., S_N$, where N is the length of the examples as

$$\log p(S|I;\theta) = \sum_{t=0}^{N} \log p(S_t|I, S_0, ..., S_{t-1}, \theta), \tag{3}$$

where (S, I) represents a pair of training example during training. We optimize the sum of the log probabilities mentioned in Eq. (3) over the whole training set by using stochastic gradient descent (SGD).

It is natural to use RNN to model $p(S_t|I, S_0, ..., S_{t-1}, \theta)$, where the variable number of words dependent on up to $t-1$ is expressed by a fixed length hidden state h_t.

After receiving a new input x_t, this memory is updated through a non-linear function f:

$$h_{t+1} = f(h_t, x_t), \tag{4}$$

For f, we use an LSTM network incorporating memory units that allow for the network to learn when to update hidden states and when to forget previous hidden states, which has shown considerable effectiveness on temporal prediction tasks.

The LSTM memory is created and copied for the image region feature maps and each sentence word, so that all LSTMs share a same set of parameters and the hidden variable h_{t-1} of the LSTM at time $t-1$ is transferred to the LSTM at time t:

$$\begin{aligned} x_{-1} &= I, && (5)\\ x_t &= W_e S_t, \quad t \in \{0, ..., N-1\} && (6)\\ p_{t+1} &= LSTM(x_t), \quad t \in \{0, ..., N-1\} && (7) \end{aligned}$$

where each word is represent by a one-hot vector S_t with its dimension equal to the size of the dictionary. Notice that we use S_0 for a the start word and S_N for the stop word, which represents the beginning and the end of the sentence, respectively. The words are mapped to the space of the region feature maps by utilizing the word embedding matrix W_e. The region feature maps I are used at $t = -1$ only in order to tell the LSTM about the image information.

In the inference stage, we just sample the first word according to p_1, then provide the corresponding embedding as the input and sample p_2. We continue this approach until we reach some maximum length or the special end-of-sentence token.

3.2.1 RESIDUAL LSTM

Training a deep recurrent neural network is difficult because of gradients vanishing gradients, but the residual network is successfully in training more than 100 convolution layers for image classification and detection. The principle insight of the residual network is to offer an extra path of gradient by adding a shortcut path between layers.

The initial design of the residual LSTM, called ResLSTM block (Zhang, Chan, & Jaitly, 2017), is to simply insert an input path to the LSTM output without scaling. However, as the number of layers increases, highway paths keep accumulating, which results in a significant performance loss. Without a proper scaling method, the variance of the residual LSTM output would continue to increase.

In this section, we put forward a novel approach for a residual LSTM. Our residual LSTM is inspired by the fact that the separation of the temporal-domain cell update with the spatial-domain shortcut path could provide more flexibility to address the vanishing/exploding gradients. Different from the highway LSTM (Zhang et al., 2016), our residual LSTM does not gather a highway path on the internal memory cell c_t. Instead, we add a shortcut path to the LSTM output layer h_t to make more shortcut gradients address the vanishing/exploding gradients.

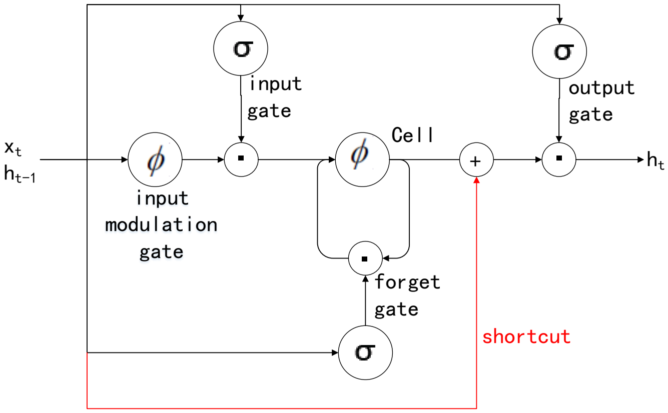


Figure 4: Architecture of LSTM and Residual LSTM. LSTM has no shortcut, while Residual LST-M has a shortcut.

Fig. 4 describes a structure of a residual LSTM layer. Our residual LSTM has a shortcut path from input x_t (or hidden in the last time step h_{t-1}) to a projection output h_t. In this paper, we use a preceding output layer as the shortcut path, although it can be any lower output layer. Equation for residual LSTM is updated as follows:

$$h_t = o_t \odot [\phi(c_t) + W_x x_t], \tag{8}$$

where W_x is a projection matrix to scale the LSTM output.

Instead of an internal memory cell, we use an output layer for the spatial shortcut path of our residual LSTM network, which can be less interfered with the temporal gradient flow. Each output layer at the residual LSTM network learns residual mapping that is not learnable from a highway path. As a result, there is no need for each new layer to waste time or resources to generate outputs which is similar to the output of prior layers. The LSTM projection matrix is reused by our residual

LSTM as a gate network. For a normal LSTM network size, we can save more than 10% of the learnable parameters from a residual LSTM over a highway LSTM.

3.3 Loss Function

Our dense captioning network can be trained by minimizing the belowing loss function:

$$L = L_{det} + \alpha L_{bbox} + \beta L_{cap}, \tag{9}$$

where L_{det}, L_{bbox}, and L_{cap} represent the detection loss, bounding box localization loss, and caption prediction loss, respectively; α and β are the influence factors chosen by parameter tuning; L_{det} is the cross-entropy loss for object classification; L_{bbox} is a regularized loss (Tian, Wang, & Wang, 2017); and L_{cap} is a cross-entropy term for the word prediction at each time step of the sequential model.

There is little knowledge about how to find the optimal results of these influence factors. Therefore, we tune these parameters using training data with engineering experience. We only fix other parameters and modify the influence factors. Each time, we change one of the influencing factors and evaluate whether the loss in Eq. (9) is decreased. If the loss is decreased, we will fix this new parameter value, and vice versa. Although this method is biased, we find its performance is satisfying in our approach, and we set α = 0.1 and β = 0.05 during experiments.

4. Results

In this section, we compare the efficiency and the performance of the proposed approach with others.

4.1 Hardware and Software Environment

We conduct experiments on a workstation with an Intel i7-4790 3.6 GHz CPU, 32GB memory, and an NVIDIA GTX Titan X graphics. We build our algorithm upon Torch 7 (Collobert, Kavukcuoglu, & Farabet, 2011) to test the performance and computational efficiency.

4.2 Implementation Details

In the training stage, we adjust the hyperparameters according to the cross-validation on the Visual Genome dataset. The min-batch size is 1, and each input image is first resized to a longer side of 720 pixels. We initialize Conv1 and Blocks 1-4 with weights that are pretrained on ImageNet (Deng et al., 2009) and all other weights from a Gaussian with a standard deviation of 0.01. Stochastic gradient descent is used. We set the momentum to 0.9, and the initial rate to 0.001 which is halved every 100k iterations. Weight decay is not employed in training. Fully connected layers (FC1 and FC2) have rectified linear units and are regularized with Dropout. This produces a code of dimension 4096 that encode its visual appearance compactly for each region. We only utilize descriptions with less than 10 words for efficiency and 10000 words with the highest frequency as the vocabulary. We replace other words with a default tag. An LSTM with 256 hidden nodes is employed for sequential modeling.

In the validation stage, we get $B = 300$ region proposals with the highest predicted confidences, and the NMS (non-maximum suppression) is employed.

4.3 Datasets

We verified our proposed approach on the Visual Genome dataset(Krishna et al., 2017) and partial Microsoft Common Objects in Context (MS-COCO) dataset (Lin et al., 2014). Visual Genome has three versions: V1.0, V1.2 and V1.4. For the purpose of comparison, our experiments are mainly based on the Visual Genome V1.0 dataset. We use 77398 images for training and 5000 images for validation and testing which is same to the train/val/test splits in (Johnson et al., 2016). MS-COCO is the largest dataset regarding image captioning, with 82,783 images for training, 40,504 images for validation and 40,775 images for testing. MS-COCO is a challenging dataset because most of its images contain multi-objects under complex scenes. To evaluate the dense captioning task, part of the MS-COCO dataset is labeled by the third party (Krishna et al., 2017) for obtaining the rich local region annotations.

4.4 Evaluation Criterion

The mAP (mean average precision) is employed as the evaluation criterion to measure the description and detection accuracy jointly. We compute the average precision for different intersection over union (IoU) thresholds (.3, .4, .5, .6, and .7) for detection accuracy, and different Meteor (Yao et al., 2017) score thresholds (0, .05, .1, .15, .2, and .25) for language similarity. Meteor is adopted because this metric is thought to be the most highly closed to human judgements in settings with a small amount of references.

4.5 Ablation Study

We conduct extensive ablation experiments and demonstrate the effects of several important components in our framework. All experiments in this subsection are performed on the Visual Genome V1.0 dataset.

Detection plays a key role in dense captioning. We compare our approach to residual block and ResNeXt to assess the effectiveness of the image feature extraction model, which is illustrated in Table 1. For a fair comparison, all the approaches use a detection confidence threshold of 0.6. According to the statistical results, the ARB has a weak performance improvement when compared to the ResNeXt because the only difference between them is the manner of merger, and the DCARB improves the detection accuracy in terms of mAP by approximately 1.0, and the detected hard sample further improve the description by a margin of approximately 0.5.

Table 1: Experimental results of the feature extraction on the Visual Genome V1.0 dataset.

Approach	Language (Meteor)	Detection (mAP)
Residual Block (He et al., 2016a)	29.81 ± 0.03	7.25 ± 0.02
ResNeXt (Xie et al., 2017)	30.24 ± 0.04	8.03 ± 0.04
ARB	30.23 ± 0.03	8.17 ± 0.02
DCARB	**30.77** ± 0.05	**9.13** ± 0.04

Hourglass-structured network can be stacked, and we then consider the question that how many hourglass-structured network should be used to find the optimal result in the dense captioning? The stack number should balance the effectiveness and the efficiency. The detection accuracy can be improved as the stack number increases, which can be seen in Table 2. However, the performance improvement is limited when the stack number is greater than 1, while the computational complexity continues to increase according to the stack number. Therefore, in the next experiments, the stack number is fixed to 1 to obtain the satisfactory performance without unnecessary computational load.

Table 2: Detection performance of the stacked hourglass on the Visual Genome V1.0 dataset. The results are measured by mAP.

Stack Number	0	1	2	3
Stacked Hourglass	6.81	9.13	9.15	9.16

We then evaluate the effectiveness of the captioning prediction model. The accuracy comparison can be seen in Table 3. The detection accuracy is not affected because the detection part is a preceding module, and Residual LSTM improves the description accuracy by approximately 0.7 when compared to the corresponding ResLSTM approach.

Table 3: Experimental results of the captioning prediction on the Visual Genome V1.0 dataset.

Approach	Language (Meteor)	Detection (mAP)
LSTM	29.35 ± 0.05	9.13 ± 0.04
Highway LSTM(Zhang, Chen, Yu, Yaco, Khudanpur, & Glass, 2016)	29.83 ± 0.06	9.13 ± 0.04
ResLSTM (Zhang et al., 2017)	30.04 ± 0.05	9.13 ± 0.04
Residual LSTM	**30.77** ± 0.06	9.13 ± 0.04

4.6 Experimental Results on the Visual Genome Dataset

Fig. 5 shows some example predictions of bounding boxes and captions on the Visual Genome V1.0 dataset. We represent the top few most possible predictions. Our model gives abundant snippet descriptions of regions and precisely grounds the captions in the images. In addition, our model can not only detect and localize the whole body but also provide partial information, such as the nose of a zebra, the window of a car, the leg of a human, and so on. The partial information may provide attributes or geometric knowledge for details in image captioning.

Table 4 depicts the evaluation results in the Visual Genome V1.0 dataset. We compare our approach with other state-of-the-art dense captioning methods, such as the full-image RNN, fully convolutional localization network (FCLN), and T-LSTM. The full-image RNN model is trained using full images and captions. For comparison purposes, in the FCLN approach, different region proposal methods, such as the EdgeBoxes (EB) (Zitnick & Dollar, 2014), and the Region Proposal Network (RPN) (Ren, He, Girshick, & Sun, 2015), are employed to extract 300 boxes for each image during test. T-LSTM incorporates the joint inference and context fusion in order to realize the accurate detection of each visual concept.

The template-based dense captioning approaches trained on the same images can obtain similar results. Our residual LSTM can improve the language accuracy by almost 1.0% because it is less interfered with a temporal gradient flow.

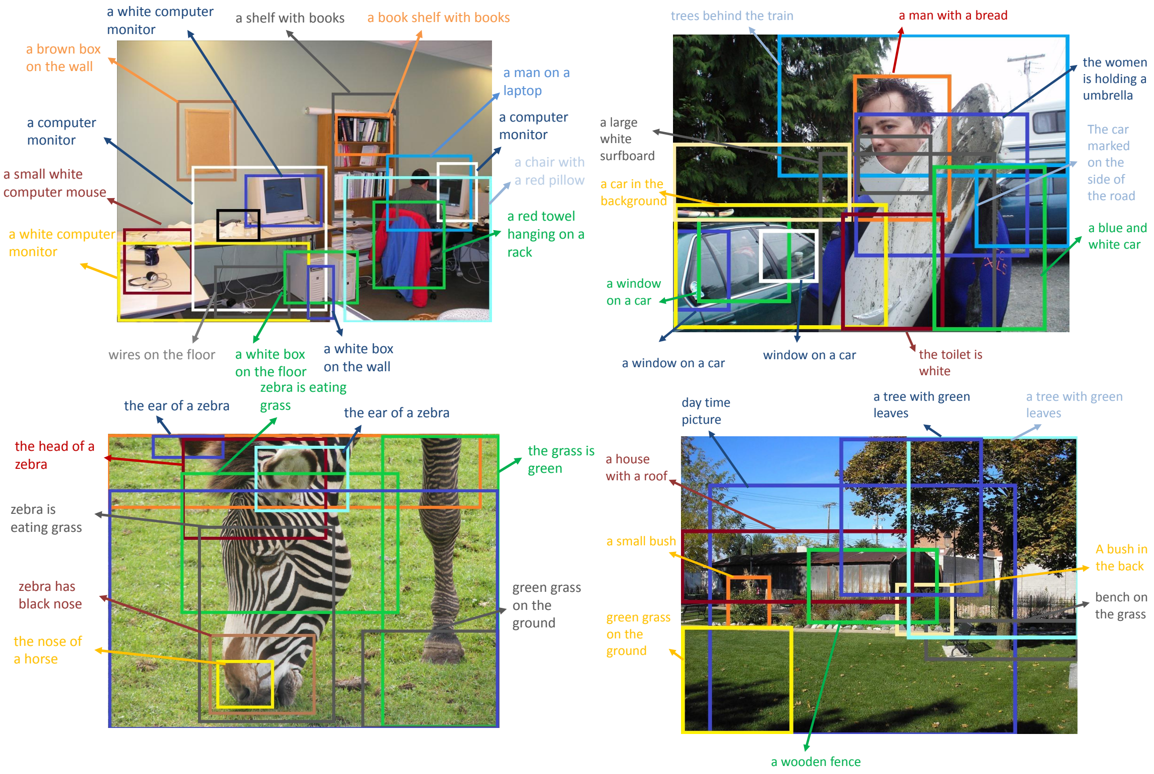


Figure 5: Example captions generated and localized by our model on the test images in the Visual Genome V1.0 dataset. We render the top few most confident predictions.

From Table 4, we can see that the proposed approach influences the final performance. For example, the performance is improved by 0.15% when using the RPN network than EB regions. The T-LSTM joint infers the object detection and the description, which greatly improves the object detection accuracy, and a ROI pooling is utilized to obtain the global information to help the dense captioning. With these two modifications, the T-LSTM obtains a mAP of 9.31 which is the best performance in the Visual Genome V1.0 dataset. We introduce the residual learning and dense connected networks into the object detection and the dense captioning. The experiments show that when compared with the FCLN approach, our approach can improve the performance with a margin of 4.55% if the dense connected network (DCARB) is employed in detection.

In efficiency comparisons, FCLN is the fastest which uses about 240 milli-seconds to process a frame and utilizes the VGG16 (Simonyan & Zisserman, 2014) network with only 16 layers to detect and localize objects. The T-LSTM is the most effective, which costs approximately 450 milli-seconds to process a frame because it has to rectify object location using caption information. Our approach updates the basic block and costs approximately 334 milli-seconds to process a frame, which has a comparable efficiency compared with the FCLN.

Table 4: Dense captioning evaluation on the Visual Genome V1.0 dataset. The results of all approaches are obtained from the original papers.

Approach	Language (Meteor)	Performance (AP)	Runtime (ms)
Full Image RNN (Karpathy & Li, 2015)	19.70	4.27	3170
FCLN on EB (Johnson et al., 2016)	26.40	5.24	360
FCLN on RPN (Johnson et al., 2016)	27.30	5.39	**240**
T-LSTM (Yang et al., 2017)	29.80	9.31	450
Our	**30.77**	**9.94**	334

4.7 Experimental Results on the MS-COCO Region Captions Dataset

The origin MS-COCO dataset does not contain region captions. Luckily, the Visual Genome dataset has images taken from the intersection of the MS-COCO and YFCC100M datasets (Thomee et al., 2016). Therefore, we make a comparison of the language and object detection results with those of other state-of-the-art dense captioning methods in these MS-COCO frames with region annotations. We apply the model trained in the Visual Genome dataset to test frames in the MS-COCO Region Captions dataset. To compare with other approaches, parameters in our approach are set according to descriptions in other approach (Karpathy & Li, 2015). In the evaluation, $K = 300$ boxes are generated after RoI pooling , and B boxes are generated with the highest predicted confidence after non-maximum suppression (NMS). Then, the corresponding region features are fed into the residual LSTM network. We apply a beam-1 search to generate region descriptions efficiently, where the word with the highest probability is chose at each time step.

Table 5 depicts the evaluation results on the MS-COCO Region Captions dataset. We compare our approach with other state-of-the-art dense captioning approaches, like the full-image RNN, Region RNN, and FCLN.

Table 5: Language and Object detection Results in the MS-COCO dataset.

Approach	Language	Object	Detection	(mAP)
	Meteor	IoU@0.1	IoU@0.3	IoU@0.5
EB+ImgRNN(Karpathy & Li, 2015)	23.3 ± 0.03	38.4 ± 0.04	15.6 ± 0.03	5.3 ± 0.02
Region RNN (Karpathy & Li, 2015)	23.4 ± 0.04	46.0 ± 0.05	27.3 ± 0.03	10.8 ± 0.02
FCLN (Johnson et al., 2016)	23.6 ± 0.03	56.0 ± 0.04	34.5 ± 0.04	15.3 ± 0.03
Our	**24.9** ± 0.02	**60.6** ± 0.05	**38.4** ± 0.04	**18.0** ± 0.03

’ImgRNN’ obtains language accuracy with a mAP of 23.3, and ’region RNN’ and ’FCLN’ have tiny effects on the language accuracy improvement, only increasing the mAP by 0.1% and 0.3%, respectively. Our residual LSTM can improve the language accuracy by 1.3% on account of the discriminant representation in the LSTM block.

Region RNN uses local feature maps as objects’ contextual information to infer the objects’ locations, while full-image RNN uses global information. Therefore, region RNN obtains a better result than full-image RNN. The FCLN develops a feature extraction network (VGG16 network), removes the ROI pooing layer and localizes the object with a fully convolution network. Hence, it obtains even better accuracy. Assuming that the transformation from image feature spaces to object location parameters is nonlinear and multimodel, we use a deeper network to model this complex mapping, and develop a new residual learning on the hourglass CNN network. Experiments show that our approach can increase the mean average precision by 4.6% when the IoU threshold is 0.1 and dense connected network is introduced. Similar results can be obtained when the IoU threshold is changed to 0.3 and 0.5, respectively.

5. Conclusion

In this paper, we proposed a new dense captioning approach to capture multi-scale object information with an improved hourglass network. First, a dense connected aggregated residual block is presented to construct an hourglass-structured network, and then, a new residual LSTM is presented to decrease the vanishing gradients further. Experiments on the Visual Genome V1.0 database and the MS-COCO Region Captions dataset have shown that our approach can effectively and efficiently improve dense captioning accuracy.

Acknowledgment

This work was supported in part by the National Natural Science Foundation of China (No. U1609215, 61602407, 61472363, 61672460, and 61702453). Natural Science Foundation of Zhejiang Province (No. LY19F030005 and LY18F020008). Open Project Program of the State Key Lab of CAD&CG (No. A1811), Zhejiang University. Xun Wang is the corresponding author. We thank Guofeng Zhang for supervision and help.

References

Banerjee, S., & Lavie, A. (2005). Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pp. 65–72.

Collobert, R., Kavukcuoglu, K., & Farabet, C. (2011). Torch7: A matlab-like environment for machine learning. In *NIPS Workshop*, pp. 192–206.

Deng, J., Dong, W., Socher, R., Li, L., Li, K., & Li, F. (2009). Imagenet: A large-scale hierarchical image database. In *CVPR*, pp. 248–255.

Fang, H., Gupta, S., Iandola, F., Srivastava, R., Deng, L., Dollár, P., ..., & Zweig, G. (2015). From captions to visual concepts and back. In *CVPR*, pp. 1473–1482.

Farhadi, A., Hejrati, M., Sadeghi, M., Young, P., Rashtchian, C., Hockenmaier, J., & Forsyth, D. (2010). Every picture tells a story: Generating sentences from images. In *ECCV*, pp. 83–92.

Fu, C., Liu, W., Ranga, A., Tyagi, A., & Berg, A. (2017). Dssd: Deconvolutional single shot detector. *ArXiv preprint. [Online]. Available: https://arxiv.org/pdf/1701.06659*.

Girshick, R. (2015). Fast r-cnn. In *ICCV*, pp. 1440–1448.

He, K., Zhang, X., Ren, S., & Sun, J. (2016a). Deep residual learning for image recognition. In *CVPR*, pp. 770–778.

He, K., Zhang, X., Ren, S., & Sun, J. (2016b). Identity mappings in deep residual networks. In *ECCV*, pp. 630–645.

Huang, G., Liu, Z., van der Maaten, L., & Weinberger, K. (2017). Densely connected convolutional networks. In *CVPR*, pp. 573–580.

Johnson, J., Karpathy, A., & Li, F. (2016). Densecap: Fully convolutional localization networks for dense captioning. In *CVPR*, pp. 4565–4574.

Karpathy, A., & Li, F. (2015). Deep visual-semantic alignments for generating image descriptions. In *CVPR*, pp. 3128–3137.

Krishna, R., Zhu, Y., Groth, O., Johnson, J., Hata, K., Kravitz, J., ..., & Li, F. (2017). Visual genome: Connecting language and vision using crowdsourced dense image annotations. *IJCV*, *123*(1), 32–73.

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. In *Proceedings of the IEEE*, pp. 2278–2324.

Lin, T., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., ..., & Zitnick, C. (2014). Microsoft coco: Common objects in context. In *ECCV*, pp. 740–755.

Lu, J., Xiong, C., Parikh, D., & Socher, R. (2017). Knowing when to look: Adaptive attention via a visual sentinel for image captioning. In *CVPR*, pp. 375–383.

Newell, A., Yang, K., & Deng, J. (2016). Stacked hourglass networks for human pose estimation. In *ECCV*, pp. 483–499.

Ordonez, V., Kulkarni, G., & Berg, T. (2011). Im2text: Describing images using 1 million captioned photographs. In *NIPS*, pp. 77–85.

Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. In *NIPS*, pp. 91–99.

Shi, W., Caballero, J., Huszár, F., Totz, J., Aitken, A., Bishop, R., ..., & Wang, Z. (2016). Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *CVPR*, pp. 1874–1883.

Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. In *ICLR*, pp. 45–52.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., ..., & Rabinovich, A. (2015). Going deeper with convolutions. In *CVPR*, pp. 1–9.

Thomee, B., Shamma, D., Friedland, G., Elizalde, B., Ni, K., Poland, D., ..., & Li, L. (2016). Yfcc100m: The new data in multimedia research. *Communications of the ACM*, *59*(2), 64–73.

Tian, Y., Wang, H., & Wang, X. (2017). Object localization via evaluation multi-task learning. *Neurocomputing*, *253*(1), 34–41.

Werbos, P. (1988). Generalization of backpropagation with application to a recurrent gas market model. *Neural networks*, *1*(4), 339–356.

Xie, S., Girshick, R., Dollr, P., Tu, Z., & He, K. (2017). Aggregated residual transformations for deep neural networks. In *CVPR*, pp. 5987–5995.

Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A., Salakhudinov, R., ..., & Bengio, Y. (2015). Show, attend and tell: Neural image caption generation with visual attention. In *ICML*, pp. 2048–2057.

Yang, L., Tang, K., Yang, J., & Li, L. (2017). Dense captioning with joint inference and visual contex. In *CVPR*, pp. 2193–2202.

Yao, T., Pan, Y., Li, Y., Qiu, Z., & Mei, T. (2017). Boosting image captioning with attributes. In *ICCV*, pp. 22–29.

You, Q., Jin, H., Wang, Z., Fang, C., & Luo, J. (2016). Image captioning with semantic attention. In *CVPR*, pp. 4651–4659.

Zagoruyko, S., & Komodakis, N. (2016). Wide residual networks. In *BMCV*, pp. 871–878.

Zeiler, M., Taylor, G., & Fergus, R. (2011). Adaptive deconvolutional networks for mid and high level feature learning. In *ICCV*, pp. 2018–2025.

Zhang, K., Sun, M., Han, T., Yuan, X., Guo, L., & Liu, T. (2017). Residual networks of residual networks: Multilevel residual networks. *TCSVT*, *pp*(99), 62–81.

Zhang, X., Zhou, X., Lin, M., & Sun, J. (2018). Shufflenet: An extremely efficient convolutional neural network for mobile devices. In *CVPR*, pp. 6848–6856.

Zhang, Y., Chan, W., & Jaitly, N. (2017). Very deep convolutional networks for end-to-end speech recognition. In *ICASSP*, pp. 4845–4849.

Zhang, Y., Chen, G., Yu, D., Yaco, K., Khudanpur, S., & Glass, J. (2016). Highway long short-term memory rnns for distant speech recognition. In *ICASSP*, pp. 5755–5759.

Zitnick, C., & Dollar, P. (2014). Edge boxes: Locating object proposals from edges. In *ECCV*, pp. 391–405.