# Recurrent Fusion Network for Image Captioning

Wenhao Jiang[1], Lin Ma[1], Yu-Gang Jiang[2], Wei Liu[1], Tong Zhang[1]

[1]Tencent AI Lab, [2]Fudan University
{cswhjiang, forest.linma}@gmail.com, ygj@fudan.edu.cn,
wl2223@columbia.edu, tongzhang@tongzhang-ml.org

**Abstract.** Recently, much advance has been made in image captioning, and an encoder-decoder framework has been adopted by all the state-of-the-art models. Under this framework, an input image is encoded by a convolutional neural network (CNN) and then translated into natural language with a recurrent neural network (RNN). The existing models counting on this framework employ only one kind of CNNs, *e.g.*, ResNet or Inception-X, which describes the image contents from only one specific view point. Thus, the semantic meaning of the input image cannot be comprehensively understood, which restricts improving the performance. In this paper, to exploit the complementary information from multiple encoders, we propose a novel recurrent fusion network (RFNet) for the image captioning task. The fusion process in our model can exploit the interactions among the outputs of the image encoders and generate new compact and informative representations for the decoder. Experiments on the MSCOCO dataset demonstrate the effectiveness of our proposed RFNet, which sets a new state-of-the-art for image captioning.

**Keywords:** Image captioning, encoder-decoder framework, recurrent fusion network (RFNet).

## 1 Introduction

Captioning [1–7], a task to describe images/videos with natural sentences automatically, has been an active research topic in computer vision and machine learning. Generating natural descriptions of images is very useful in practice. For example, it can improve the quality of image retrieval by discovering salient contents and help visually impaired people understand image contents.

Even though a great success has been achieved in object recognition [8, 9], describing images with natural sentences is still a very challenging task. Image captioning models need to have a thorough understanding of the input image and capture the complicated relationships among objects. Moreover, they also need to capture the interactions between images and languages and thereby translate image representations into natural sentences.

The encoder-decoder framework, with its advance in machine translation [10, 11], has demonstrated promising performance for image captioning [7, 12, 13]. This framework consists of two parts, namely an encoder and a decoder. The encoder is usually a convolutional neural network (CNN), while decoder is

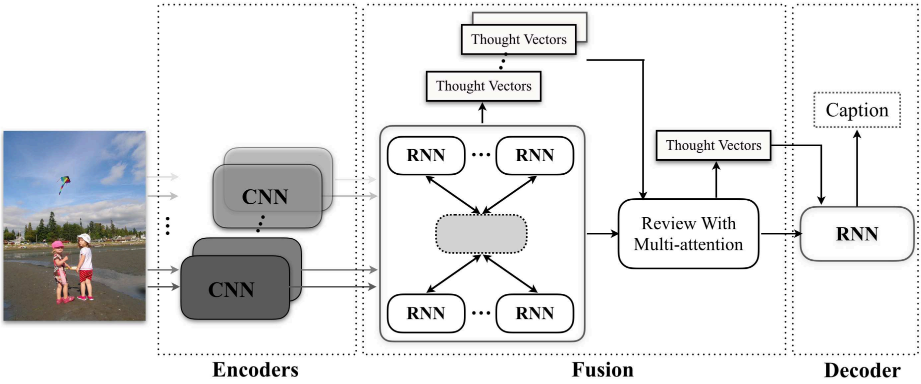

**Fig. 1.** The framework of our RFNet. Multiple CNNs are employed as encoders and a recurrent fusion procedure is inserted after the encoders to form better representations for the decoder. The fusion procedure consists of two stages. The first stage exploits interactions among the representations from multiple CNNs to generate multiple sets of thought vectors. The second stage performs multi-attention on the sets of thought vectors from the first stage and generates a new set of thought vectors for the decoder.

a recurrent neural network (RNN). The encoder is used to extract image representations, based on which the decoder is used to generate the corresponding captions. Usually, a pre-trained CNN for image classification is leveraged to extract image representations.

All existing models employ only one encoder, so the performance heavily depends on the expressive ability of the deployed CNN. Fortunately, there are quite a few well-established CNNs, *e.g.*, ResNet [14], Inception-X [15, 16], *etc.* It is natural to improve the image captioning models by extracting diverse representations with multiple encoders, which play a complementary role in fully depicting and characterizing the image semantic meaning. However, to the best of our knowledge, there are no models considering to exploit the complementary behaviors of multiple encoders for image captioning.

In this paper, to exploit complementary information from multiple encoders, we propose a recurrent fusion network (RFNet) for image captioning. Our framework, as illustrated in Fig. 1, introduces a fusion procedure between the encoders and decoder. The multiple CNNs, served as encoders, can provide diverse and more comprehensive descriptions of the input image. The fusion procedure performs a given number of RNN steps and outputs the hidden states as thought vectors. Our fusion procedure consists of two stages. The first stage contains multiple components and each component processes the information from one encoder. The interactions among the components are captured to generate thought vectors. Hence, each component can communicate with the other components and extract complementary information from them. The second stage compresses the outputs of the first stage into one set of thought vectors. During this procedure, the interactions among the sets of thought vectors are further exploited, thus, useful information is absorbed into the final thought vectors, which will be used as input of the attention model in the decoder. The intuition behind our proposed RFNet is to fuse all the information encoded by multiple encoders and produce thought vectors that are more comprehensive and representative than the original ones.

## 2 Related Works

### 2.1 Encoder-Decoder Methods for Image Captioning

Recently, inspired by advance in machine translation, the encoder-decoder framework [17, 10] has also been introduced to image captioning [18]. In this framework, a CNN pre-trained on an image classification task is used as the encoder, while a RNN is used as the decoder to translate the information from the encoder into natural sentences. This framework is simple and elegant, and several extensions have been proposed. In [12], an attention mechanism was introduced. The model with this attention mechanism could determine which subregions should be focused on automatically at each time step. In [19], ReviewNet was proposed. The review steps can learn the annotation vectors and initial states for the decoder, which are more representative than those generated by the encoder directly. In [20], a guiding network which models attribute properties of input images was introduced for the decoder. In [21], the authors observed that the decoder does not need visual attendance when predicting non-visual words, and hence proposed an adaptive attention model that attends to the image or to the visual sentinel automatically at each time step.

Besides, several approaches that introduce useful information into this framework have been proposed to improve the image captioning performance. In [22], the word occurrence prediction was treated as a multi-label classification problem. And a region-based multi-label classification framework was proposed to extract visually semantic information. This prediction is then used to initialize a memory cell of a long short-term memory (LSTM) model. Yao *et al.* improved this procedure and discussed different approaches to incorporate word occurrence predictions into the decoder [23].

Recently, in order to optimize the non-differentiable evaluation metrics directly, policy-gradient methods for reinforcement learning are employed to train the encoder-decoder framework. In [24], the cross entropy loss was replaced with CIDEr [25]. The system was then trained with the REINFORCE algorithm [26], which significantly improves the performance. Such a training strategy can be leveraged to improve the performance of all existing models under the encoder-decoder framework.

### 2.2 Encoder-Decoder Framework with Multiple Encoders or Decoders

In [27], multi-task learning (MTL) was combined with sequence-to-sequence learning with multiple encoders or decoders. In the multi-task sequence-to-sequence learning, the encoders or decoders are shared among different tasks. The goal of [27] is to transfer knowledge among tasks to improve the performance. For example, the tasks of translation and image captioning can be formulated together as a model with only one decoder. The decoder is shared between both basks and responsible for translating from both image and source language. Both tasks can benefit from each other. A similar structure was also exploited in [28] to perform multi-lingual translation. In this paper, we propose a model to combine representations from multiple encoders for the decoder. In [27, 28], the inputs of the encoders are different. But in our model, they are the same. Our goal is to leverage complementary information from different encoders to form better representations for the decoder.

### 2.3 Ensemble and Fusion Learning

Our RFNet also relates to information fusion, multi-view learning [29], and ensemble learning [30]. Each representation extracted from an individual image CNN can be regarded as an individual view depicting the input image. Combining different representations with diversity is a well-known technique to improve the performance. The combination process can occur at the input, intermediate, and output stage of the target model. For the input fusion, the most simple way is to concatenate all the representations and use the concatenation as input of the target model. This method usually leads to limited improvements. For the output fusion, the results of base learners for individual views are combined to form the final results. The common ensemble technique in image captioning is regarded as an output fusion technique, combining the output of the decoder at each time step [18, 19, 24]. For the intermediate fusion, the representations from different views are preprocessed by exploiting the relationships among them to form input for the target model. Our method can be regarded as a kind of intermediate fusion methods.

## 3 Background

To provide a clear description of our method, we present a short review of the encoder-decoder framework for image captioning in this section.

### 3.1 Encoder

Under the encoder-decoder framework for image captioning, a CNN pre-trained for an image classification task is usually employed as the encoder to extract the global representation and subregion representations of the input image. The global representation is usually the output of a fully connected layer and subregion representations are usually the outputs of a convolutional layer. The extracted global representation and subregion representations are denoted as $\mathbf{a}_0$ and $A = \{\mathbf{a}_1, \ldots, \mathbf{a}_k\}$, respectively, where $k$ denotes the subregion number.

### 3.2 Decoder

Given the image representations $\mathbf{a}_0$ and $A$, a decoder, which is usually a gated recurrent unit (GRU) [31] or long short-term memory (LSTM) [32], is employed to translate an input image into a natural sentence. In this paper, we use LSTM equipped with an attention mechanism as the basic unit of the decoder.

Recall that an LSTM with an attention mechanism is a function that outputs the results based on the current hidden state, current input, and context vector. The context vector is the weighted sum of elements from $A$, with the weights determined by an attention model. We adopt the same LSTM used in [12] and express the LSTM unit with the attention strategy as follows:

$$\begin{pmatrix} \mathbf{i}_t \\ \mathbf{f}_t \\ \mathbf{o}_t \\ \mathbf{g}_t \end{pmatrix} = \begin{pmatrix} \sigma \\ \sigma \\ \sigma \\ \tanh \end{pmatrix} \mathbf{T} \begin{pmatrix} \mathbf{H}_t \\ \mathbf{z}_{t-1} \end{pmatrix}, \tag{1}$$

$$\mathbf{c}_t = \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \mathbf{g}_t, \tag{2}$$

$$\mathbf{h}_t = \mathbf{o}_t \odot \tanh(\mathbf{c}_t), \tag{3}$$

where $\mathbf{i}_t$, $\mathbf{f}_t$, $\mathbf{c}_t$, $\mathbf{o}_t$, and $\mathbf{h}_t$ are input gate, forget gate, memory cell, output gate, and hidden state of the LSTM, respectively. Here,

$$\mathbf{H}_t = \begin{bmatrix} \mathbf{x}_t \\ \mathbf{h}_{t-1} \end{bmatrix} \tag{4}$$

is the concatenation of input $\mathbf{x}_t$ of the time step $t$ and hidden state $\mathbf{h}_{t-1}$, $\mathbf{T}$ is a linear transformation operator. $\mathbf{z}_t$ is the context vector, which is the output of attention model $f_{\text{att}}(A, \mathbf{h}_{t-1})$. Specifically,

$$e_{ti} = \text{sim}(\mathbf{a}_i, \mathbf{h}_{t-1}), \;\; \alpha_{ti} = \frac{\exp(e_{ti})}{\sum_{j=1}^{k} \exp(e_{tj})}, \;\text{ and } \mathbf{z}_t = \sum_{i=1}^{k} \alpha_{ti} \mathbf{a}_i,$$

where $\text{sim}(\mathbf{a}_i, \mathbf{h}_t)$ is a function to measure the similarity between $\mathbf{a}_i$ and $\mathbf{h}_t$, which is usually realized by a multilayer perceptron (MLP). In this paper, we use the shorthand notation

$$[\mathbf{h}_t, \mathbf{c}_t] = \text{LSTM}(\mathbf{H}_t, f_{\text{att}}(A, \mathbf{h}_{t-1})) \tag{5}$$

for the above equations.

The purpose of image captioning is to generate a caption $\mathcal{C} = (y_1, y_2, \cdots, y_N)$ for one given image $\mathcal{I}$. The objective adopted is usually to minimize a negative log-likelihood:

$$\mathcal{L} = -\log\ p(\mathcal{C}|\mathcal{I}) = -\sum_{t=0}^{N-1} \log p(y_{t+1}|y_t), \tag{6}$$

where $p(y_{t+1}|y_t) = \text{Softmax}(\mathbf{W}\mathbf{h}_t)$ and $\mathbf{h}_t$ is computed by setting $\mathbf{x}_t = \mathbf{E}\mathbf{y}_t$. Here, $\mathbf{W}$ is a matrix for linear transformation and $y_0$ is the sign for the start of sentences. $\mathbf{E}\mathbf{y}_t$ denotes the distributed representation of the word $\mathbf{y}_t$, in which $\mathbf{y}_t$ is the one-hot representation for word $y_t$ and $\mathbf{E}$ is the word embedding matrix.

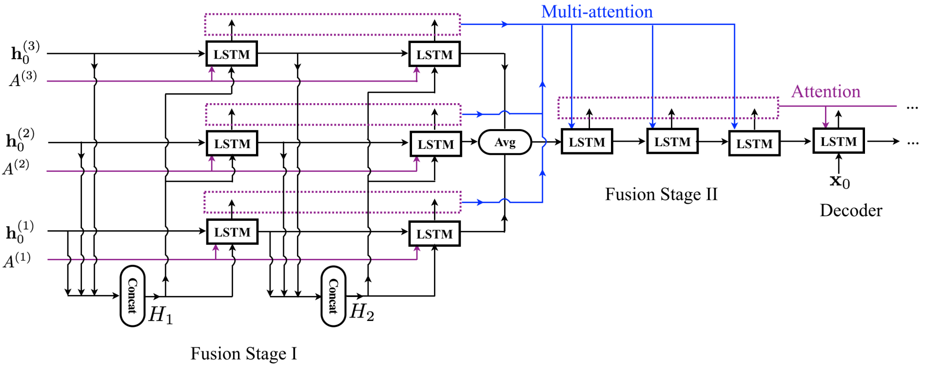

**Fig. 2.** An example with $M = 3$, $T_1 = 2$, and $T_2 = 3$ for illustrating our proposed RFNet (encoders are omitted). The fusion stage I contains $M$ review components. The input of each review component at each time step is the concatenation of hidden states from all components at the previous time step. Each review component outputs the hidden state as a thought vector. The fusion stage II is a review component that performs the multi-attention mechanism on the multiple sets of thought vectors from fusion stage I. The parameters of LSTM units in the fusion procedure are all different. The purple rectangles indicate the sets of thought vectors. Moreover, in order to make it clear, the discriminative supervision is not presented.

## 4 Our Method

In this section, we propose our recurrent fusion network (RFNet) for image captioning. The fusion process in RFNet consists of two stages. The first stage combines the representations from multiple encoders to form multiple sets of thought vectors, which will be compressed into one set of thought vectors in the second stage. The goal of our model is to generate more representative thought vectors for the decoder. Two special designs are adopted: 1) employing interactions among components in the first stage; 2) reviewing the thought vectors from the previous stage in the second stage. We will describe the details of our RFNet in this section and analyze our design in the experimental results section.

### 4.1 The Architecture

In our model, $M$ CNNs serve as the encoders. The global representation and subregion representations extracted from the $m$-th CNN are denoted as $\mathbf{a}_0^{(m)}$ and $A^{(m)} = \{\mathbf{a}_1^{(m)}, \ldots, \mathbf{a}_{k_m}^{(m)}\}$, respectively.

The framework of our proposed RFNet is illustrated in Fig. 2. The fusion procedure of RFNet consists of two stages, specifically fusion stage I and II. Both stages perform a number of RNN steps with attention mechanisms and output hidden states as the thought vectors. The numbers of steps in stage I and II are denoted as $T_1$ and $T_2$, respectively. The hidden states from stage I and II are regarded as the thought vectors. The thought vectors of stage I will be used as the input of the attention model of stage II. The hidden states and memory cells after the last step of fusion stage I are aggregated to form the initial hidden state and the memory cell for fusion stage II. The thought vectors generated by stage II will be used in the decoder. RFNet is designed to capture the interactions among the components in stage I, and extract useful information and compress the $M$ sets of thought vectors to generate more compact and informative ones in stage II. The details will be described in the following subsections.

### 4.2 Fusion Stage I

Fusion stage I takes $M$ sets of annotation vectors as inputs and generates $M$ sets of thought vectors, which will be aggregated into one set of thought vectors in fusion stage II. This stage contains $M$ review components. In order to capture the interactions among the review components, each review component need to know what has been generated by all the components at the previous time step.

We denote the hidden state and memory cell of the $m$-th review component after time step $t$ as $\mathbf{h}_t^{(m)}$ and $\mathbf{c}_t^{(m)}$. Their initial values $\mathbf{h}_0^{(m)}$ and $\mathbf{c}_0^{(m)}$ are initialized by

$$\mathbf{h}_0^{(m)} = \mathbf{c}_0^{(m)} = \mathbf{W}_0^{(m)} \mathbf{a}_0^{(m)}, \tag{7}$$

where $\mathbf{W}_0^{(m)} \in \mathbb{R}^{s \times d_m}$, $s$ is the size of LSTM hidden state and $d_m$ is the size of $\mathbf{a}_0^{(m)}$. At time step $t$, the hidden states of the $m$-th review component are computed as

$$\left[\mathbf{h}_t^{(m)}, \mathbf{c}_t^{(m)}\right] = \text{LSTM}_t^{(m)}\left(\mathbf{H}_t, f_{\text{att-fusion-I}}^{(m,t)}\left(A^{(m)}, \mathbf{h}_{t-1}^{(m)}\right)\right), \tag{8}$$

where

$$\mathbf{H}_t = \begin{bmatrix} \mathbf{h}_{t-1}^{(1)} \\ \vdots \\ \mathbf{h}_{t-1}^{(M)} \end{bmatrix} \tag{9}$$

is the concatenation of hidden states of all review components at the previous time step, $f_{\text{att-fusion-I}}^{(m,t)}(\cdot,\cdot)$ is the attention model for the $m$-th review component, and $\text{LSTM}_t^{(m)}(\cdot,\cdot)$ is the LSTM unit used by the $m$-th review component at time step $t$. Stage I can be regarded as a grid LSTM [33] with independent attention mechanisms. In our model, the LSTM unit $\text{LSTM}_t^{(m)}(\cdot,\cdot)$ can be different for different $t$ and $m$. Hence, $M \times T_1$ LSTMs are used in fusion stage I. The set of thought vectors generated from the $m$-th component is denoted as:

$$B^{(m)} = \left\{\mathbf{h}_1^{(m)}, \mathbf{h}_2^{(m)}, \cdots, \mathbf{h}_{T_1}^{(m)}\right\}. \tag{10}$$

In fusion stage I, the interactions among review components are realized via Eq. (9). The vector $\mathbf{H}_t$ contains the hidden states of all the components after time step $t-1$ and is shared as input among them. Hence, each component is aware of the states of the other components and thus can absorb complementary information from $\mathbf{H}_t$. This hidden state sharing mechanism provides a way for the review component to communicate with each other, which facilitates the generation of thought vectors.

### 4.3 Fusion Stage II

The hidden state and memory cell of fusion stage II are initialized with $\mathbf{h}_{T_1}^{(1)}, \cdots, \mathbf{h}_{T_1}^{(M)}$ and $\mathbf{c}_{T_1}^{(1)}, \cdots, \mathbf{c}_{T_1}^{(M)}$. We use averaging in our model:

$$\mathbf{h}_{T_1} = \frac{1}{M}\sum_{m=1}^{M} \mathbf{h}_{T_1}^{(m)}, \text{ and } \mathbf{c}_{T_1} = \frac{1}{M}\sum_{m=1}^{M} \mathbf{c}_{T_1}^{(m)}. \tag{11}$$

Fusion stage II combines $M$ sets of thought vectors to form a new one using the multi-attention mechanism. At each time step, the concatenation of context vectors is calculated as:

$$\tilde{\mathbf{z}}_t = \begin{bmatrix} f_{\text{att-fusion-II}}^{(1,t)}\left(B^{(1)}, \mathbf{h}_{t-1}\right) \\ f_{\text{att-fusion-II}}^{(2,t)}\left(B^{(2)}, \mathbf{h}_{t-1}\right) \\ \vdots \\ f_{\text{att-fusion-II}}^{(M,t)}\left(B^{(M)}, \mathbf{h}_{t-1}\right) \end{bmatrix}, \tag{12}$$

where $f_{\text{att-fusion-II}}^{(m,t)}(\cdot,\cdot)$ is an attention model. Hence, this stage contains $M$ independent attention models. They are all soft attention models, similar to [34]. With the context vector $\tilde{\mathbf{z}}_t$, the state transition is expressed as:

$$[\mathbf{h}_t, \mathbf{c}_t] = \text{LSTM}_t(\mathbf{h}_{t-1}, \tilde{\mathbf{z}}_t), \tag{13}$$

where $\text{LSTM}_t$ is the LSTM unit at time step $t$. Please note that all the LSTM units in this stage are also different.

Fusion stage II can be regarded as review steps [19] with $M$ independent attention models, which performs the attention mechanism on the thought vectors yielded in the first stage. It combines and compresses the outputs from stage I and generates only one set of thought vectors. Hence, the generated thought vectors can provide more information for the decoder.

The hidden states of fusion stage II are collected to form the thought vector set:

$$C = \{\mathbf{h}_{T_1+1}, \cdots, \mathbf{h}_{T_1+T_2}\}, \tag{14}$$

which will be used as the input of the attention model in the decoder.

### 4.4 Decoder

The decoder translates the information generated by the fusion procedure into natural sentences. The initial hidden state and memory cell are inherited from the last step of fusion stage II directly. The decoder step in our model is the same as other encoder-decoder models, which is expressed as:

$$[\mathbf{h}_t, \mathbf{c}_t] = \mathrm{LSTM}_{dec}(\mathbf{H}_t, f_{\text{att-dec}}\left(C, \mathbf{h}_{t-1}\right)), \tag{15}$$

where $\mathrm{LSTM}_{dec}(\cdot,\cdot)$ is the LSTM unit for all the decoder steps, $f_{\text{att-dec}}(\cdot,\cdot)$ is the corresponding attention model, $\mathbf{x}_t$ is the word embedding for the input word at the current time step, and

$$\mathbf{H}_t = \begin{bmatrix} \mathbf{x}_t \\ \mathbf{h}_{t-1} \end{bmatrix}. \tag{16}$$

### 4.5 Discriminative Supervision

We adopt the discriminative supervision in our model to further boost image captioning performance, which is similar to [35, 19]. Given a set of thought vectors $V$, a matrix $\mathbf{V}$ is formed by selecting elements from $V$ as column vectors. A score vector $\mathbf{s}$ of words is then calculated as

$$\mathbf{s} = \text{Row-Max-Pool}\left(\mathbf{WV}\right), \tag{17}$$

where $\mathbf{W}$ is a trainable linear transformation matrix and Row-Max-Pool$(\cdot)$ is a max pooling operator along the rows of the input matrix. The $i$-th element of $\mathbf{s}$ is denoted $s_i$, which represents the score for the $i$-th word. Adopting a multi-label margin loss, we obtain the loss function for discriminative supervision as:

$$\mathcal{L}_d\left(V\right) = \sum_{j\in\mathcal{W}} \sum_{i\notin\mathcal{W}} \max\left(0, 1 - \left(s_j - s_i\right)\right), \tag{18}$$

where $\mathcal{W}$ is the set of all frequent words in the current caption. In this paper, we only consider the 1,000 most frequent words in the captions of the training set.

By considering both the discriminative supervision loss in Eq. (18) and the captioning loss in Eq. (6), the complete loss function of our model is expressed as:

$$\mathcal{L}_{all} = \mathcal{L} + \frac{\lambda}{M+1}\left(\mathcal{L}_d(C) + \sum_{m=1}^{M} \mathcal{L}_d\left(B^{(m)}\right)\right), \tag{19}$$

where $\lambda$ is a trade-off parameter, and $B^{(m)}$ and $C$ are sets of thought vectors from fusion stages I and II.

# 5 Experimental Results

## 5.1 Dataset

The MSCOCO dataset[1] [36] is the largest benchmark dataset for the image captioning task, which contains 82,783, 40,504, and 40,775 images for training, validation, and test, respectively. This dataset is challenging, because most images contain multiple objects in the context of complex scenes. Each image in this dataset is associated with five captions annotated by human. For offline evaluation, we follow the conventional evaluation procedure [37, 23, 19], and employ the same data split as in [6], which contains 5,000 images for validation, 5,000 images for test, and 113,287 images for training. For online evaluation on the MSCOCO evaluation server, we add the testing set into the training set to form a larger training set.

For the captions, we discard all the non-alphabetic characters, transform all letters into lowercase, and tokenize the captions using white space. Moreover, all the words with the occurrences less than 5 times are replaced by the unknown token `<UNK>`. Thus a vocabulary consisting of 9,487 words is finally constructed.

## 5.2 Configurations and Settings

For the experiments, we use ResNet [14], DenseNet [38], Inception-V3 [15], Inception-V4, and Inception-ResNet-V2 [16] as encoders to extract 5 groups of representations. Each group of representations contains a global feature vector and a set of subregion feature vectors. The outputs of the last convolution layer (before pooling) are extracted as subregion features. For Inception-V3, the output of the last fully connected layer is used as the global feature vector. For the other CNNs, the means of subregion representations are regarded as the global feature vectors. The parameters for encoders are fixed during the training procedure. Since the reinforcement learning (RL) has become a common method to boost image captioning performance [39, 24, 40–42], we first train our model with cross entropy loss and fine-tune the trained model with CIDEr optimization using reinforcement learning [24]. The performance of models trained with both cross entropy loss and CIDEr optimization is reported and compared.

When training with cross entropy loss, the scheduled sampling [43], label-smoothing regularization (LSR) [15], dropout, and early stopping are adopted. For scheduled sampling, the probability of sampling a token from model is $\min(0.25, \frac{epoch}{100})$, where *epoch* is the number of passes sweeping over training data. For LSR, the prior distribution over labels is uniform distribution and the smoothing parameter is set to 0.1. Dropout is only applied on the hidden states and the probability is set to 0.3 for all LSTM units. We terminate the training procedure, if the evaluation measurement on validation set, specifically the CIDEr, reaches the maximum value. When training with RL [24], only dropout and early stopping are used.

[1] http://mscoco.org/

The hidden state size is set as 512 for all LSTM units in our model. The parameters of LSTM are initialized with uniform distribution in $[-0.1, 0.1]$. The Adam [44] is applied to optimize the network with the learning rate setting as $5 \times 10^{-4}$ and decaying every 3 epochs by a factor 0.8 when training with cross entropy loss. Each mini-batch contains 10 images. For RL training, the learning rate is fixed as $5 \times 10^{-5}$. For training with cross entropy, the weight of discriminative supervision $\lambda$ is set to 10. And discriminative supervision is not used for training with reinforcement learning. To improve the performance, data augmentation is adopted. Both flipping and cropping strategies are used. We crop 90% of width and height at the four corners. Hence, 10× images are used for training.

For sentence generation in testing stage, there are two common strategies. The first one is greedy search, which chooses the word with maximum probability at each time step and sets it as LSTM input for next time step until the end-of-sentence sign is emitted or the maximum length of sentence is reached. The second one is the beam search strategy which selects the top-$k$ best sentences at each time step and considers them as the candidates to generate new top-$k$ best sentences at the next time step. Usually beam search provides better performance for models trained with cross entropy loss. For model trained with RL, beam search and greedy search generate similar results. But greedy search is faster than beam search. Hence, for models trained with RL, we use greedy search to generate captions.

### 5.3 Performance and Analysis

We compare our proposed RFNet with the state-of-the-art approaches on image captioning, including Neural Image Caption (NIC) [18], Attribute LSTM [22], LSTM-A3 [23], Recurrent Image Captioner (RIC) [45], Recurrent Highway Network (RHN)[46], Soft Attention model [12], Attribute Attention model [48], Sentence Attention model [47], Review Net [19], Text Attention model [37], Att2in model [24], Adaptive model [21], and Up-Down model [42]. Please note that the encoder of Up-Down model is not a CNN pre-trained on ImageNet dataset [51]. It used Faster R-CNN trained on Visual Genome [52] to encode the input image.

*Evaluation metrics.* Following the standard evaluation process, five types of metrics are used for performance comparisons, specifically the BLEU [53], METEOR [54], ROUGE-L [55], CIDEr [25], and SPICE [56]. These metrics measure the similarity between generated sentences and the ground truth sentences from specific viewpoints, e.g n-gram occurrences, semantic contents. We use the official MSCOCO caption evaluation scripts[2] and the source code of SPICE[3] for the performance evaluation.

[2] https://github.com/tylin/coco-caption

[3] https://github.com/peteanderson80/coco-caption

**Table 1.** Performance comparisons on the test set of Karpathy's split [6]. All image captioning models are trained with the cross entropy loss. The results are obtained using beam search with beam size 3. $^{\Sigma}$ indicates an ensemble, $^{\dagger}$ indicates a different data split, and (−) indicates that the metric is not provided. All values are reported as percentage (%), with the highest value of each entry highlighted in boldface.

| Model | BLEU-1 | BLEU-2 | BLEU-3 | BLEU-4 | METEOR | ROUGE-L | CIDEr | SPICE |
|---|---|---|---|---|---|---|---|---|
| Soft Attention [12] | 70.7 | 49.2 | 34.4 | 24.3 | 23.9 | - | - | - |
| ReviewNet [19] | - | - | - | 29.0 | 23.7 | - | 88.6 | - |
| LSTM-A3 [23] | 73.5 | 56.6 | 42.9 | 32.4 | 25.5 | 53.9 | 99.8 | 18.5 |
| Text Attention [37] | 74.9 | 58.1 | 43.7 | 32.6 | 25.7 | - | 102.4 | - |
| Attribute LSTM [22] | 74.0 | 56.0 | 42.0 | 31.0 | 26.0 | - | 94.0 | - |
| RIC [45] | 73.4 | 53.5 | 38.5 | 29.9 | 25.4 | - | - | - |
| RHN [46] | 72.3 | 55.3 | 41.3 | 30.6 | 25.2 | - | 98.9 | - |
| Adaptive [21]$^{\dagger}$ | 74.2 | 58.0 | 43.9 | 33.2 | 26.6 | - | 108.5 | - |
| Att2in [24] | - | - | - | 31.3 | 26.0 | 54.3 | 101.3 | - |
| Up-Down [42] | 77.2 | - | - | 36.2 | 27.0 | 56.4 | 113.5 | 20.3 |
| Sentence Attention [47]$^{\Sigma}$ | 71.6 | 54.5 | 40.5 | 30.1 | 24.7 | - | 97.0 | - |
| Attribute Attention [48]$^{\Sigma}$ | 70.9 | 53.7 | 40.2 | 30.4 | 24.3 | - | - | - |
| NIC [18]$^{\dagger\Sigma}$ | - | - | - | 32.1 | 25.7 | - | 99.8 | - |
| Att2in [24]$^{\Sigma}$ | - | - | - | 32.8 | 26.7 | 55.1 | 106.5 | - |
| ReviewNet [19]$^{\Sigma}$ | 76.7 | 60.9 | 47.3 | 36.6 | 27.4 | 56.8 | 113.4 | 20.3 |
| RFNet | 76.4 | 60.4 | 46.6 | 35.8 | 27.4 | 56.5 | 112.5 | 20.5 |
| RFNet$^{\Sigma}$ | **77.4** | **61.6** | **47.9** | **37.0** | **27.9** | **57.3** | **116.3** | **20.8** |

*Performance comparisons.* The performance of models trained with cross entropy loss is shown in Table 1, including the performance of single model and ensemble of models. First, it can be observed that our single model RFNet significantly outperformed the existing image captioning models, except the Up-Down model [42]. Our RFNet performed inferiorly to Up-Down in BLEU-1, BLEU-4, and CIDEr, while superiorly to Up-down in METEOR, ROUGGE-L, and SPICE. However, the encoder of Up-Down model was pre-trained on ImageNet dataset and fine-tuned on Visual Genome [52] dataset. The Visual Genome dataset is heavily annotated with objects, attributes and region descriptions and 51K images are extracted from MSCOCO dataset. Hence, the encoder of Up-Down model is trained with far more information than the CNNs trained on ImageNet. With the recurrent fusion strategy, RFNet can extract useful information from different encoders to remedy the lacking of information about objects and attributes in the representations.

Moreover, we can observe that our *single RFNet model* performed significantly better than other ensemble models, such as NIC, Att2in, and behaved comparably with ReviewNet$^{\Sigma}$ which is an ensemble of 40 ReviewNets (8 models for each CNNs). But RFNet$^{\Sigma}$, an ensemble of 4 RFNets, significantly outperformed all the ensemble models.

The performance comparisons with RL training are presented in Table 2. We compared RFNet with Att2all [24], Up-Down model [42], and ensemble of ReviewNets. For our method, the performance of single model and ensemble of 4 models are provided. We can see that our RFNet outperformed other methods. For online evaluation, we used the ensemble of 7 models and the comparisons

**Table 2.** Performance comparisons on the test set of Karpathy's split [6]. All models are finetuned with RL. $^{\Sigma}$ indicates an ensemble, $^{\dagger}$ indicates a different data split, and (−) indicates that the metric is not provided. All values are reported as percentage (%), with the highest value of each entry highlighted in boldface. The results are obtained using greedy search.

| Models | BLEU-1 | BLEU-2 | BLEU-3 | BLEU-4 | METEOR | ROUGE-L | CIDEr | SPICE |
|---|---|---|---|---|---|---|---|---|
| Att2all [24] | - | - | - | 34.2 | 26.7 | 55.7 | 114.0 | - |
| Up-Down [42] | 79.8 | - | - | 36.3 | 27.7 | 56.9 | 120.1 | 21.4 |
| Att2all [24]$^{\Sigma}$ | - | - | - | 35.4 | 27.1 | 56.6 | 117.5 | - |
| ReviewNet [19]$^{\Sigma}$ | 79.6 | 63.6 | 48.7 | 36.4 | 27.7 | 57.5 | 121.6 | 21.0 |
| RFNet | 79.1 | 63.1 | 48.4 | 36.5 | 27.7 | 57.3 | 121.9 | 21.2 |
| RFNet$^{\Sigma}$ | **80.4** | **64.7** | **50.0** | **37.9** | **28.3** | **58.3** | **125.7** | **21.7** |

**Table 3.** Performance of different models on the MSCOCO evaluation server. All values are reported as percentage (%), with the highest value of each entry highlighted in boldface.

|  | BLEU-1 |  | BLEU-2 |  | BLEU-3 |  | BLEU-4 |  | METEOR |  | ROUGE-L |  | CIDEr |  | SPICE |  |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Methods | C5 | C40 | C5 | C40 | C5 | C40 | C5 | C40 | C5 | C40 | C5 | C40 | C5 | C40 | C5 | C40 |
| NIC [7] | 71.3 | 89.5 | 54.2 | 80.2 | 40.7 | 69.4 | 30.9 | 58.7 | 25.4 | 34.6 | 53.0 | 68.2 | 94.3 | 94.6 | 18.2 | 63.6 |
| Captivator [35] | 71.5 | 90.7 | 54.3 | 81.9 | 40.7 | 71.0 | 30.8 | 60.1 | 24.8 | 33.9 | 52.6 | 68.0 | 93.1 | 93.7 | 18.0 | 60.9 |
| M-RNN [49] | 71.6 | 89.0 | 54.5 | 79.8 | 40.4 | 68.7 | 29.9 | 57.5 | 24.2 | 32.5 | 52.1 | 66.6 | 91.7 | 93.5 | 17.4 | 60.0 |
| LRCN [50] | 71.8 | 89.5 | 54.8 | 80.4 | 40.9 | 69.5 | 30.6 | 58.5 | 24.7 | 33.5 | 52.8 | 67.8 | 92.1 | 93.4 | 17.7 | 59.9 |
| Hard-Attention [12] | 70.5 | 88.1 | 52.8 | 77.9 | 38.3 | 65.8 | 27.7 | 53.7 | 24.1 | 32.2 | 51.6 | 65.4 | 86.5 | 89.3 | 17.2 | 59.8 |
| ATT-FCN [48] | 73.1 | 90.0 | 56.5 | 81.5 | 42.4 | 70.9 | 31.6 | 59.9 | 25.0 | 33.5 | 53.5 | 68.2 | 94.3 | 95.8 | 18.2 | 63.1 |
| ReviewNet [19] | 72.0 | 90.0 | 55.0 | 81.2 | 41.4 | 70.5 | 31.3 | 59.7 | 25.6 | 34.7 | 53.3 | 68.6 | 96.5 | 96.9 | 18.5 | 64.9 |
| LSTM-A3 [23] | 78.7 | 93.7 | 62.7 | 86.7 | 47.6 | 76.5 | 35.6 | 65.2 | 27.0 | 35.4 | 56.4 | 70.5 | 116.0 | 118.0 | - | - |
| Adaptive [21] | 74.8 | 92.0 | 58.4 | 84.5 | 44.4 | 74.4 | 33.6 | 63.7 | 26.4 | 35.9 | 55.0 | 70.5 | 104.2 | 105.9 | 19.7 | 67.3 |
| PG-BCMR [41] | 75.4 | 91.8 | 59.1 | 84.1 | 44.5 | 73.8 | 33.2 | 62.4 | 25.7 | 34.0 | 55.0 | 69.5 | 101.3 | 103.2 | - | - |
| Att2all [24] | 78.1 | 93.7 | 61.9 | 86.0 | 47.0 | 75.9 | 35.2 | 64.5 | 27.0 | 35.5 | 56.3 | 70.7 | 114.7 | 116.7 | 20.7 | 68.9 |
| Up-Down [42] | 80.2 | **95.2** | 64.1 | 88.8 | 49.1 | 79.4 | 36.9 | 68.5 | 27.6 | 36.7 | 57.1 | 72.4 | 117.9 | 120.5 | 21.5 | 71.5 |
| RFNet | **80.4** | 95.0 | **64.9** | **89.3** | **50.1** | **80.1** | **38.0** | **69.2** | **28.2** | **37.2** | **58.2** | **73.1** | **122.9** | **125.1** | - | - |

are provided in Table 3. We can see that our RFNet still achieved the best performance. The C5 and C40 CIDEr scores were improved by 5.0 and 4.6, compared to the state-of-the-art Up-Down model [42].

*Ablation study of fusions stage I and II.* To study the effects of the two fusion stages, we present the performance of the following models:

- **RFNet$_{-\mathbf{I}}$** denotes RFNet without fusion stage I, with only the fusion stage II preserved. The global representations are concatenated to form one global representation and multi-attention mechanisms are performed on the subregion representation from the multiple encoders. The rest is the same with RFNet.
- **RFNet$_{-\mathbf{II}}$** denotes RFNet without fusion stage II. Multiple attention models are employed in the decoder and the rest is the same as RFNet.
- **RFNet$_{-\mathbf{inter}}$** denotes RFNet without the interactions in fusion stage I. Each component in the fusion stage I is independent. Specifically, at each time step, the input of the $m$-th component is just $\mathbf{h}_{t-1}^{(m)}$, and it is unaware of the hidden states of the other components.

The CIDEr scores on the test set of the Karpathy's split are presented in Table 4. We can see that both the two-stage structure and the interactions in the first stage are important for our model. With the interactions, the quality of thought vectors in the first stage can be improved. With the two-stage structure, the thought vectors in the first stage can be refined and compressed into more compact and informative set of thought vectors in the second stage. Therefore, with the specifically designed recurrent fusion strategy, our proposed RFNet provides the best performance.

**Table 4.** Ablation study of fusion stages I and II in our RFNet. The CIDEr scores are obtained using beam search with beam size 3 on the test set of the Karpathy's split.

| | $\text{RFNet}_{-\text{I}}$ | $\text{RFNet}_{-\text{II}}$ | $\text{RFNet}_{-\text{inter}}$ | RFNet |
|---|---|---|---|---|
| CIDEr | 108.9 | 111.5 | 112.1 | 112.5 |

*Effects of discriminative supervision.* We examined the effects of the discriminative supervision with different values of $\lambda$. First, it can be observed that introducing discriminative supervision properly can help improve the captioning performance, with 107.2 ($\lambda$=1) and 107.3 ($\lambda$=10) vs. 105.2 ($\lambda$=0) . However, if $\lambda$ is too large, *e.g.*, 100, the discriminative supervision will degrade the corresponding performance. Therefore, in this paper, $\lambda$ is set as 10, which provided the best performance.

**Table 5.** CIDEr scores with different $\lambda$ values on the test set. The captions are generated by greedy search.

| $\lambda$ | 0 | 1 | 10 | 100 |
|---|---|---|---|---|
| CIDEr | 105.2 | 107.2 | 107.3 | 104.7 |

## 6 Conclusions

In this paper, we proposed a novel recurrent fusion network (RFNet), to exploit complementary information of multiple image representations for image captioning. In the RFNet, a recurrent fusion procedure is inserted between the encoders and the decoder.This recurrent fusion procedure consists of two stages, and each stage can be regarded as a special RNN. In the first stage, each image representation is compressed into a set of thought vectors by absorbing complementary information from the other representations. The generated sets of thought vectors are then compressed into another set of thought vectors in the second stage, which will be used as the input to the attention module of the decoder. The RFNet achieved leading performance on the MSCOCO evaluation server, which corroborates the effectiveness of our proposed network architecture.

## References

1. Venugopalan, S., Rohrbach, M., Donahue, J., Mooney, R., Darrell, T., Saenko, K.: Sequence to sequence-video to text. In: Proceedings of the IEEE international conference on computer vision. (2015) 4534–4542
2. Krishna, R., Hata, K., Ren, F., Fei-Fei, L., Niebles, J.C.: Dense-captioning events in videos. In: ICCV. (2017) 706–715
3. Chen, Y., Wang, S., Zhang, W., Huang, Q.: Less is more: Picking informative frames for video captioning. arXiv preprint arXiv:1803.01457 (2018)
4. Wang, B., Ma, L., Zhang, W., Liu, W.: Reconstruction network for video captioning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2018) 7622–7631
5. Wang, J., Jiang, W., Ma, L., Liu, W., Xu, Y.: Bidirectional attentive fusion with context gating for dense video captioning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2018) 7190–7198
6. Karpathy, A., Fei-Fei, L.: Deep visual-semantic alignments for generating image descriptions. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2015) 3128–3137
7. Vinyals, O., Toshev, A., Bengio, S., Erhan, D.: Show and tell: Lessons learned from the 2015 mscoco image captioning challenge. IEEE transactions on pattern analysis and machine intelligence **39**(4) (2017) 652–663
8. He, K., Zhang, X., Ren, S., Sun, J.: Spatial pyramid pooling in deep convolutional networks for visual recognition. ECCV (2014)
9. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems. (2012) 1097–1105
10. Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y.: Learning phrase representations using rnn encoder-decoder for statistical machine translation. arXiv:1406.1078 (2014)
11. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. In: ICLR 2015. (2014)
12. Xu, K., Ba, J., Kiros, R., Cho, K., Courville, A.C., Salakhutdinov, R., Zemel, R.S., Bengio, Y.: Show, attend and tell: Neural image caption generation with visual attention. In: ICML. (2015)
13. Chen, X., Ma, L., Jiang, W., Yao, J., Liu, W.: Regularizing rnns for caption generation by reconstructing the past with the present. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (June 2018)
14. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2016) 770–778
15. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. In: CVPR. (2016)
16. Szegedy, C., Ioffe, S., Vanhoucke, V., Alemi, A.A.: Inception-v4, inception-resnet and the impact of residual connections on learning. In: AAAI. (2017) 4278–4284
17. Sutskever, I., Vinyals, O., Le, Q.V.: Sequence to sequence learning with neural networks. In: Advances in neural information processing systems. (2014) 3104–3112
18. Vinyals, O., Toshev, A., Bengio, S., Erhan, D.: Show and tell: A neural image caption generator. In: CVPR. (2015) 3156–3164

19. Yang, Z., Yuan, Y., Wu, Y., Cohen, W.W., Salakhutdinov, R.R.: Review networks for caption generation. In: NIPS. (2016)
20. Jiang, W., Ma, L., Chen, X., Zhang, H., Liu, W.: Learning to guide decoding for image captioning. The Thirty-Second AAAI Conference on Artificial Intelligence (2018)
21. Lu, J., Xiong, C., Parikh, D., Socher, R.: Knowing when to look: Adaptive attention via a visual sentinel for image captioning. arXiv preprint arXiv:1612.01887 (2016)
22. Wu, Q., Shen, C., Liu, L., Dick, A., van den Hengel, A.: What value do explicit high level concepts have in vision to language problems? In: CVPR. (2016)
23. Yao, T., Pan, Y., Li, Y., Qiu, Z., Mei, T.: Boosting image captioning with attributes. In: The IEEE International Conference on Computer Vision (ICCV). (Oct 2017)
24. Rennie, S.J., Marcheret, E., Mroueh, Y., Ross, J., Goel, V.: Self-critical sequence training for image captioning. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (July 2017)
25. Vedantam, R., Lawrence Zitnick, C., Parikh, D.: Cider: Consensus-based image description evaluation. In: CVPR. (2015)
26. Williams, R.J.: Simple statistical gradient-following algorithms for connectionist reinforcement learning. Machine learning **8**(3-4) (1992) 229–256
27. Luong, T., Le, Q.V., Sutskever, I., Vinyals, O., Kaiser, L.: Multi-task sequence to sequence learning. In: International Conference on Learning Representations. (2016)
28. Firat, O., Cho, K., Bengio, Y.: Multi-way, multilingual neural machine translation with a shared attention mechanism. In: Proceedings of NAACL-HLT. (2016) 866–875
29. Xu, C., Tao, D., Xu, C.: A survey on multi-view learning. arXiv preprint arXiv:1304.5634 (2013)
30. Zhou, Z.H.: Ensemble methods: foundations and algorithms. CRC press (2012)
31. Chung, J., Gulcehre, C., Cho, K., Bengio, Y.: Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv preprint arXiv:1412.3555 (2014)
32. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural computation **9**(8) (1997) 1735–1780
33. Kalchbrenner, N., Danihelka, I., Graves, A.: Grid long short-term memory. arXiv preprint arXiv:1507.01526 (2015)
34. Xu, H., Saenko, K.: Ask, Attend and Answer: Exploring Question-Guided Spatial Attention for Visual Question Answering. ECCV (2016)
35. Fang, H., Gupta, S., Iandola, F., Srivastava, R.K., Deng, L., Dollár, P., Gao, J., He, X., Mitchell, M., Platt, J.C., et al.: From captions to visual concepts and back. In: CVPR. (2015) 1473–1482
36. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: ECCV. (2014)
37. Mun, J., Cho, M., Han, B.: Text-guided attention model for image captioning. AAAI (2017)
38. Huang, G., Liu, Z., van der Maaten, L., Weinberger, K.Q.: Densely connected convolutional networks. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (July 2017)
39. Ranzato, M., Chopra, S., Auli, M., Zaremba, W.: Sequence level training with recurrent neural networks. ICLR-15 (2015)

40. Ren, Z., Wang, X., Zhang, N., Lv, X., Li, L.J.: Deep reinforcement learning-based image captioning with embedding reward. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (July 2017)
41. Liu, S., Zhu, Z., Ye, N., Guadarrama, S., Murphy, K.: Improved image captioning via policy gradient optimization of spider. In: The IEEE International Conference on Computer Vision (ICCV). (Oct 2017)
42. Anderson, P., He, X., Buehler, C., Teney, D., Johnson, M., Gould, S., Zhang, L.: Bottom-up and top-down attention for image captioning and vqa. arXiv preprint arXiv:1707.07998 (2017)
43. Bengio, S., Vinyals, O., Jaitly, N., Shazeer, N.: Scheduled sampling for sequence prediction with recurrent neural networks. In: Advances in Neural Information Processing Systems. (2015) 1171–1179
44. Kingma, D., Ba, J.: Adam: A method for stochastic optimization. The International Conference on Learning Representations (ICLR) (2015)
45. Liu, H., Yang, Y., Shen, F., Duan, L., Shen, H.T.: Recurrent image captioner: Describing images with spatial-invariant trnsformation and attention fieltering. arXiv:1612.04949 (2016)
46. Gu, J., Wang, G., Chen, T.: Recurrent highway networks with language cnn for image captioning. arXiv:1612.07086 (2016)
47. Zhou, L., Xu, C., Koch, P., Corso, J.J.: Watch what you just said: Image captioning with text-conditional attention. arXiv:1606.04621 (2016)
48. You, Q., Jin, H., Wang, Z., Fang, C., Luo, J.: Image captioning with semantic attention. In: CVPR. (2016) 4651–4659
49. Mao, J., Xu, W., Yang, Y., Wang, J., Huang, Z., Yuille, A.: Deep captioning with multimodal recurrent neural networks (m-rnn). ICLR (2015)
50. Donahue, J., Anne Hendricks, L., Guadarrama, S., Rohrbach, M., Venugopalan, S., Saenko, K., Darrell, T.: Long-term recurrent convolutional networks for visual recognition and description. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2015) 2625–2634
51. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on, IEEE (2009) 248–255
52. Krishna, R., Zhu, Y., Groth, O., Johnson, J., Hata, K., Kravitz, J., Chen, S., Kalantidis, Y., Li, L.J., Shamma, D.A., Bernstein, M., Fei-Fei, L.: Visual genome: Connecting language and vision using crowdsourced dense image annotations. (2016)
53. Papineni, K., Roukos, S., Ward, T., Zhu, W.J.: Bleu: a method for automatic evaluation of machine translation. In: ACL. (2002)
54. Banerjee, S., Lavie, A.: Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In: ACL workshop. (2005)
55. Lin, C.Y.: Rouge: A package for automatic evaluation of summaries. In: ACL-04 workshop. (2004)
56. Anderson, P., Fernando, B., Johnson, M., Gould, S.: Spice: Semantic propositional image caption evaluation. In: European Conference on Computer Vision, Springer (2016) 382–398