
Image Descriptor with Visual Attention Mechanism using Long Short-term Memory

Lin-Ying Cheng
A53276414

Che-Ming Chia
A53273827

Shang-Wei Hung
A53267981

Tsun-Hsu Lee
A53271531

Team Name: Jimmy SD
Department of Electrical and Computer Engineering
University of California San Diego
{lycheng, chchia, shung, thlee}@eng.ucsd.edu
Github: <https://github.com/lychengr3x/Image-Descriptor>

Abstract

Automatically learning to describe the content and information of images becomes important in the field of artificial intelligence. In this project, we follow most state-of-the-art networks based on Convolutional neural networks and Recurrent neural networks (CNN-RNN). We utilize CNN to extract features over the image, and then adopt RNN to generate captions from these features. To address the problem of object missing in the predicted text, we append a attention network to force the visual features to be considered at each time step. We present various configurations of CNN models and using merely LSTM for our RNN model. Throughout this project, we evaluate our networks on MS COCO dataset.

1 Introduction

Generating captions of images automatically is a challenging problem that has received a large amount of interest from the computer vision and natural language processing communities. It has a breakthrough in computer vision and has been a fundamental problem in artificial intelligence. The impact of image captioning can be seen in many different application tasks. One of the common application is the usage of search engines that matches the search query and the images. Not only for the models be powerful enough to solve the computer vision challenges of detecting and determining the objects in an image, but they are also capable of capturing and interpreting their relationship in a natural language. Some other applications after more mature in image captioning could be images descriptor for people who are blind or low vision and rely on sounds or text. Even more, real-time video description is also a fascinating application.

However, this task has been proven to be very hard for artificial systems before the advent of deep learning models. The majority of previous work in visual recognition before deep learning has focused on labeling images with a fixed set of visual categories is more like classification. The description with closed vocabulary assumption is very restrictive in previous work compared to how rich descriptions that a human can compose. In deep learning approach, we widely used encoder-decoder framework to tackle the problem of automatic image captioning.

The image encoder is a Convolutional Neural Network (CNN). CNNs belong to a class of neural networks which are used to extract features from an image. With the combination of very dense and deep CNN, we could extract high level features of the image and take it as an input of the decoder. The decoder for generating captions is using Recurrent Neural Networks. RNNs are widely used in

problems related to Natural Languages Processing (NLP) because of the output dependency on any previous states (outputs).

Besides, even if we can extract high level features from CNNs, we may lose information from the propagation of the RNN. So, in order to having more thorough description, we add the attention mechanism(17) directly from the output of CNNs features to each node of RNN for generating captions. This allows for the salient features to dynamically come to the forefront as needed. This is especially important to distill information in image down to the most salient objects as one effective solution that has been widely adopted.

In this project, we use MS COCO Dataset(11) for training and evaluation. We experiment with different configurations of CNNs, such as ResNet152, ResNet101, ResNet50(5), and VGG19(14). After extracting the features, we use Long Short-term Memory (LSTM)(6) to generate the captions. We also incorporate the attention mechanism such that the neural network will focus on a particular region in the image while interpreting the related region of the image. Finally, we present the results and evaluate the results with BLEU(13) score for above mentioned configurations.

2 Related Works

2.1 Model Evolution

There have been several attempts at solving the problem of automatic image captioning before deep learning models were popularized. Take Baby Talk(9) as an example, in this research they first smooth the output of computer vision-based detection and recognition algorithms with statistics mined from large pools of visually descriptive text to determine the best content words to use to describe an image. Then choose words to construct natural language sentences based on the predicted content and general statistics from natural language. This years, many of recent researches use neural networks for caption generation. Such models typically extract a visual feature vector via a CNN, and then send that vector to a language model for caption generation. Representative works include (3), (4), and (8). The differences of these various methods mainly lie in variation of the types of CNN architectures and language models. For example, the feature vectors are fed into LSTM only in the beginning of RNN in (12), while it was used at each time step of the RNN in (16). In our CNN architecture, we compare the performance of VGG(14) and ResNet(5) with several number of layers and fed the feature vectors into one specific RNN model-LSTM at each time step which is similar to (16).

Most recently, (12) utilized visual attention mechanism to learn which part to focus on during image captioning. Inspired by the presence of attention in the human visual system, (7) firstly proposed soft attention and hard attention to make the decoder exposed to different aspects of image information at each time step. (1) pre-trained an object detection model on a large dense captioning data set, and used it to obtain image features at conceptual level for attention. In our works, we do not need to pre-trained object detection model and only trained on image captioning data set without assistant data sets. Typically, these models can be characterized as top-down approaches, with context provided by a representation of a partially-completed caption in the case of image captioning.

2.2 Related Evaluation Methods

Image captioning is notoriously difficult to make evaluations due to the inherent ambiguity. Human evaluation scores are reliable but costly to obtain. Therefore, there are several proposed metrics to automatically evaluate the image caption results. Commonly used evaluation metrics such as BLEU (13) , ROUGE (10) and CIDEr (15) are mostly based on n-gram overlap and tend to be insensitive to semantic information. Another improved metric called SPICE proposed by Anderson et al. get higher correlation with human judgement but encounters difficulties with repetitive sentence. All of the above metrics worth nothing since they rely solely on similarity between candidate and reference captions instead of taking the image into consideration. In our works, however, we evaluate our model using BLEU since it was most popular one. Unfortunately, the scores are not very high even when the captions specifically point out every major parts of the images.

3 Models

Our models generally draws inspiration from the neural encoder-decoder framework for the task of image captioning. The details of encoder and decoder will be illustrated in the below subsections. Besides, we also append attention mechanism (17) into our network, which will be introduced in the last subsection.

3.1 Encoder

In a modified manner of recent successful model for this task, Convolutional Neural Network (CNN) is commonly used in the state-of-the-art networks for image tasks. CNN is capable to extract the features and encode the image into a compact feature representation, and then fed this feature vector into the decoder. In this project, we experiment with different advanced configurations of CNN such as VGG(14) and ResNet(5).

The input to VGG network is a fixed-sized 224×224 RGB image. In preprocessing, we substrate the mean RGB value among each pixel in the training set. The image is passed through a stack of convolutional layers and maxpooling layers. All hidden layers are equipped with Rectified Linear Units (ReLU). We use the built-in VGG19 model pretrained on ImageNet and fine-tune for image captioning task. We replace the final fully-connected (FC) layer with a one specific to our task with output dimension equal to the embedded size. Besides, with attention mechanism, we need to extract the feature map before the last FC layer. Thus, to meet the dimension for the attention network, we change the output dimension for the second to the last FC layer.

In addition to VGG model, we utilize ResNet for our CNN part. ResNet adds on skip connection to ease the training of deeper networks. To make pretrained ResNet suitable for our task, we change the output dimension of the final FC layer to the embedded size. In this project, we have tried pretrained ResNet152, ResNet101 and ResNet50, and fine tune for image captioning task.

3.2 Decoder

(16) have proposed to directly maximize the probability of the correct description given the image by using the below equation:

$$\theta^* = \arg \max_{\theta} \sum_{(I,S)} \log p(S|I, \theta) \quad (1)$$

,where θ are the parameters of our model, I is an image and S is correct transcription. After applying the chain rule to model the joint probability, we decide to model the joint probability with Recurrent Neural Network. In this project, we only adopt Long short-term memory (LSTM)(6). The decision is due to the vanishing and exploding gradients problem for the general RNN tasks.

$$\log p(S|I, \theta) = \log p(S_t|I, \theta, S_0, S_1, S_2) \quad (2)$$

The basic cell of standard LSTM is named as LSTM cell, which is shown in Figure 1. Unlike RNNs, the LSTM cells also take the cell output state C_t and the previous cell input state C_{t-1} into account during training. Owing to the gated structure designed in the LSTM cells, LSTM is able to deal with long-term dependencies issue. There are three types of gates in a LSTM cell: Input gate, Forget gate and Output gate. These gated structure help LSTM to be much flexible and scalable model for sequential input data. We denote the input gate, the forget gate and the output gate at time t to be i_t , f_t , o_t , respectively. These gates structure can be calculated by using the below equations:

$$f_t = \sigma_g(W_f x_t + U_f h_{t-1} + b_f) \quad (3)$$

$$i_t = \sigma_g(W_i x_t + U_i h_{t-1} + b_i) \quad (4)$$

$$o_t = \sigma_g(W_o x_t + U_o h_{t-1} + b_o) \quad (5)$$

$$\tilde{C}_t = \tanh(W_c X_t + U_c h_{t-1} + b_c) \quad (6)$$

where W_f , W_i , W_o , W_C are the weights connecting between the hidden layer input to the three gates and the input cell state, while U_f , U_i , U_o and U_C are the weights mapping the previous cell output state to the three gates and the input cell state. b_f , b_i , b_o and b_c stands for biases. Operation σ represents the gate activation function, which usually uses the sigmoid function.

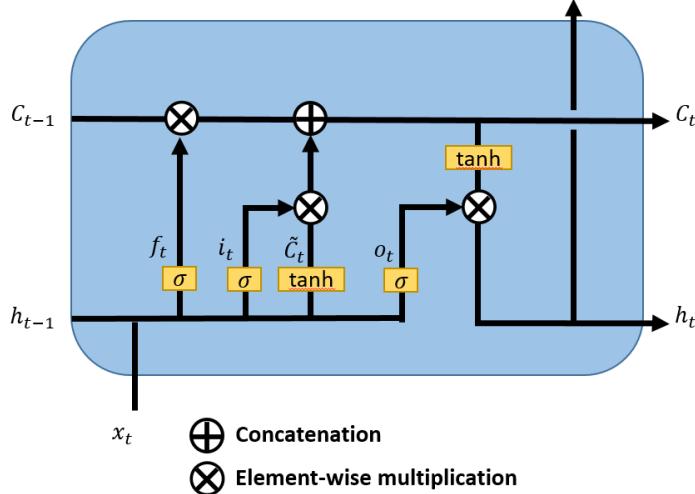


Figure 1: Overview of Long short-term memory. σ usually represents the sigmoid activation function. Operation symbols are the same for the remaining figures.

Following the above four formulas, at each time iteration t , the cell output state C_t and the state output h_t can be calculated as below:

$$C_t = f_t \otimes C_{t-1} + i_t \otimes \tilde{C}_t \quad (7)$$

$$h_t = o_t \otimes \tanh(C_t) \quad (8)$$

In our LSTM model, we use only one hidden layer. After passing the image through the convolutional layers, we can get a feature map vector of size 2048. We then use one Linear Layer to adjust it to output of size 256. This output vector will be served as an input to the LSTM. LSTM will maintain its hidden state and cell state at each time step, and the output token is provided as an input of LSTM at the next time step. The token predicted at each time step can be translated into a word. The combination of the predicted words is exactly our predicted image caption sequence.

3.3 Without Attention Mechanism

We demonstrate the architecture of the non-attention based basic model in Figure 2. Before training stage, we have to gather words from the COCO training dataset. We will remove the words whose count is under a threshold. In one hot representation, each vocabulary is viewed as independent word. However, some words are not exactly independent. For example, girl and woman, word and words. Instead of directly using one hot encoding to represent the word in the word space, we use learnable embedding layers to transform the raw word index into word embeddings to solve the problems brought by one hot representation.

In this method, the visual feature only be used for the very first time step in the decoder, whose information is not ample enough for whole LSTM network to generate a summarized sentence of the entire image. Therefore, a modification to the use of visual features is considered.

3.4 Attention Mechanism

To address the above issue, attention mechanism is needed. Attention-based LSTM can generate the description of that image while focus over different certain region of the input image by adjusting the weights of visual feature map from the output of CNN. To implement attention-based LSTM (see Figure 3), we append a small network called attention network (Figure 4) before each input to LSTM at each time step. Moreover, the visual feature map after CNN will be used for each time step in LSTM. Just like general LSTM, the cell state and the hidden state will be maintain in LSTM model

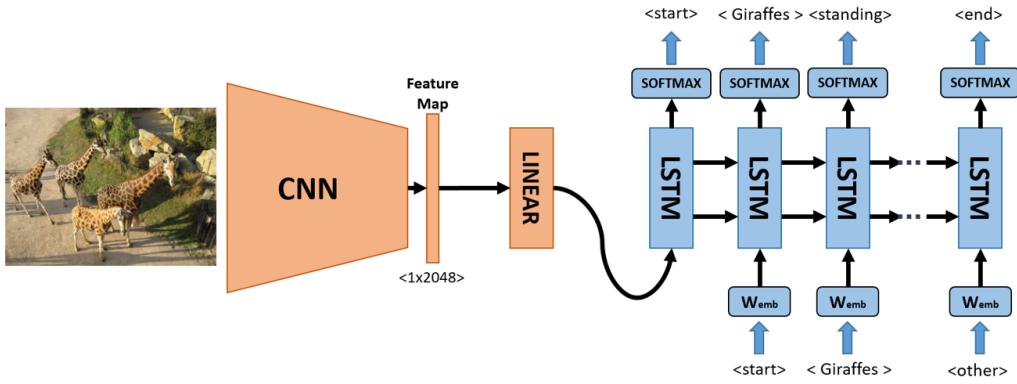


Figure 2: The architecture overview of general image captioning model. It is a encoder-decoder structure consisting of CNN and LSTM. W_{emb} indicates the word embeddings.

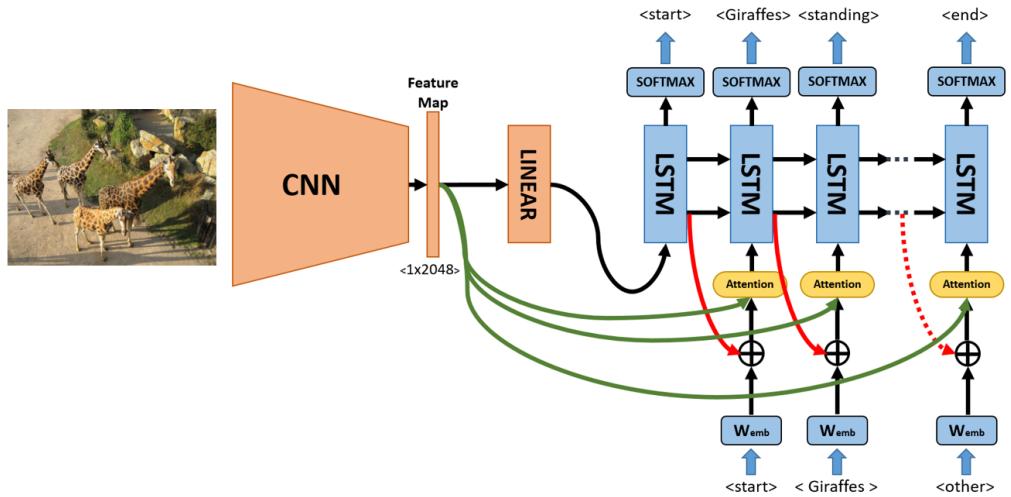


Figure 3: The architecture overview of our image captioning model with Attention networks. Green arrow indicates feature maps flow while red arrow shows the flow of hidden state. Yellow box represents the Attention networks. W_{emb} indicates the word embeddings.

for each time step. Furthermore, the hidden state will also be utilized to produce the input for the next LSTM input.

This attention networks not only take the output of LSTM from the previous time step as the input but also the current hidden state and visual feature map. In Figure 4, to get attention weights, we concatenate the word embedding of current input token and the current hidden state of the LSTM. After passing the stacked vector through linear layer and softmax function, the attention weights is formulated. Then, performing element-wise multiplication between attention wights and CNN feature map to get attending weights. These attending weights represents the certain features this time step the LSTM model pay attention to. These attending weights are then concatenated with the word embeddings and adjust the size to the same as word embeddings by passing through a linear layer. The output will exactly be the input to the LSTM model for this time step.

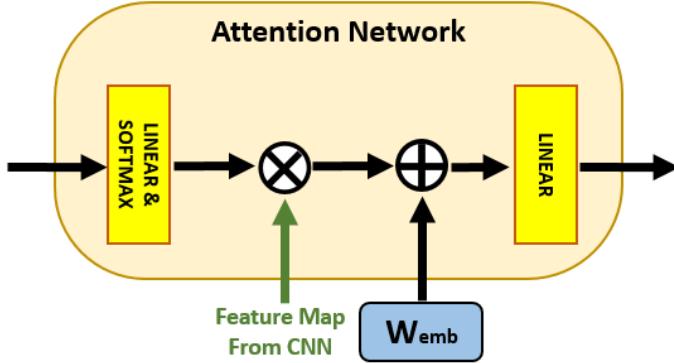


Figure 4: Overview of Attention network. It also take the feature map from CNN as an input. W_{emb} indicates the word embeddings. Weights calculated after the first yellow layer are called attention weights while the weights after the element-wise multiplication is named as attending weights in our model.

4 Dataset

MS COCO and other large datasets has enabled the training of more complex models such as neural networks. Since MS COCO offers a much larger amount of training data than Pascal1K, in this project, we use dataset collected by Microsoft COCO(2). These images are split into training, validation and testing sets. The images were gathered by searching for pairs of 80 object categories and various scene types on Flickr. The number of captions gathered is 413,915 captions for 82,783 images in training, 202,520 captions for 40,504 images in validation and 379,249 captions for 40,775 images in testing. We first have our training data trained on several different configurations of encoder-decoder and then evaluate the performance on validation data.

5 Results

Figure 5 is our training loss and evaluation loss. According to the figure, you can see that the model begin overfitting after seven epochs. However, since we have initially refer to other recent papers that most of them evaluate their model on 5 epochs, we take the checkpoint at epoch 5 to evaluate our model instead of using the best epochs theoretically.

Given that the field of image captioning has seen significant advances in the last few years, we do think it is more meaningful and persuasive to report BLEU metric since it has better correlation to human judgement and it is commonly used in state-of-the-art networks. According to the results in Table 1, the BLEU score has been improved by using attention mechanism except in Resnet101. We find the captions generated by attention model in Resnet101 have clearly covered more details than the non-attention one via human observation. Thus, this inconsistent outcome might be caused by the algorithm utilized by BLEU calculation since this score is calculated based on the occurrence of key words instead of the semantic of the text.

Some examples evaluated on the MS COCO dataset can been seen in Table 2. It is interesting to see that in the first image the model with attention mechanism can illustrate the image in more details compared to the caption that obtained by network without attention mechanism. Furthermore, we testify our model on a private image as in Table 3. Based on the limitation of dictionary, some objects cannot be represented in our predicted text since there is no such word in the vocabulary dictionary. Here, we show an image with commonly used object – laptop. According to the captions we obtain from each model, the attention mechanism will only sometimes make the caption illustrate more detail but never mess up the original captions.

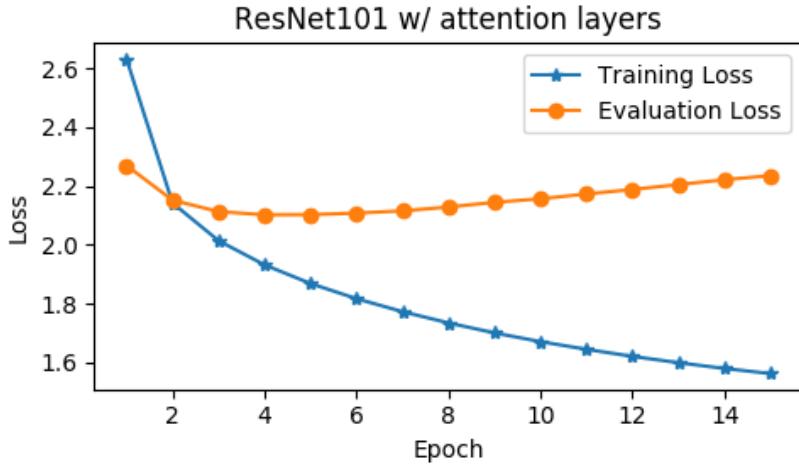


Figure 5: Loss

	Without Attention Mechanism	With Attention Mechanism
Resnet152	0.195719	0.195783
Resnet101	0.196287	0.195804
Resnet50	0.195392	0.196017
VGG19	0.196555	0.196830

Table 1: Result of BLEU score over four different CNN models over MS COCO dataset. The left column is without attention mechanism, while the right column is with attention mechanism.

6 Conclusion

We have present an improved method for neural image caption generator, an end-to-end neural network system that can automatically view an image and generate a reasonable description in English. Neural image caption is based on CNN that encodes image into several features, followed by a RNN that generates a corresponding sentence. The model is trained to obtain the maximum likelihood of the captions given the image. Experiments on COCO dataset shows the robustness of our model in terms of qualitative results which generates reasonable captions of the image. The quantitative evaluation results, using BLEU also show reliable performance. It is clear that from these experiments, as the dataset of images and the vocabulary dictionary increases, so will the performance of approach such as neural image caption generator.

	<p>Resnet152</p> <ul style="list-style-type: none"> · a pizza with toppings on a table. · a pizza with pepperonis , olives , peppers and peppers. <p>Resnet101</p> <ul style="list-style-type: none"> · a pizza with a lot of toppings on it. · a pizza sitting on top of a wooden table. <p>Resnet50</p> <ul style="list-style-type: none"> · a pizza sitting on top of a wooden table. · two pizzas on a wooden table with a pizza cutter. <p>VGG19</p> <ul style="list-style-type: none"> · a pizza with cheese and spinach on a table. · a pizza sitting on top of a white plate.
	<p>Resnet152</p> <ul style="list-style-type: none"> · a person on a snowboard in the snow. · a person riding a snowboard down a snow covered slope. <p>Resnet101</p> <ul style="list-style-type: none"> · a person on skis is in the snow. · a person skiing down a snowy mountain side. <p>Resnet50</p> <ul style="list-style-type: none"> · a person skiing down a snowy slope with trees in the background. · a skier is skiing down a snowy hill. <p>VGG19</p> <ul style="list-style-type: none"> · a person skiing down a mountain with a mountain in the background. · a man riding a snowboard down a snow covered slope.
	<p>Resnet152</p> <ul style="list-style-type: none"> · a group of young men playing a game of soccer. · a group of young children playing a game of frisbee. <p>Resnet101</p> <ul style="list-style-type: none"> · a group of people playing with a frisbee in a field. · a young boy holding a bat while standing on a baseball field. <p>Resnet50</p> <ul style="list-style-type: none"> · a man and a child are playing with a frisbee. · a man and a little girl playing frisbee in a park. <p>VGG19</p> <ul style="list-style-type: none"> · a group of people playing frisbee in a park. · a group of people playing a game of frisbee.

Table 2: Result of image caption on MS COCO dataset. The left column shows the evaluated image while the right column displays the predicted result. From up to bottom, we demonstrate the predicted text of the image using Resnet152, Resnet101, Resnet50 and VGG 19. In each model, we provide the network without attention mechanism and then followed by the one with attention mechanism.

**Resnet152**

- a man sitting at a desk with a laptop and a laptop.
- a man sitting at a table with a laptop computer.

Resnet101

- a man sitting at a desk with a laptop computer.
- a man sitting at a desk with a laptop computer.

Resnet50

- a man sitting at a desk with a laptop computer.
- a man sitting at a table using a laptop computer.

VGG19

- a man sitting at a table with a laptop.
- a man sitting at a table using a laptop computer.

Table 3: Result of image caption on our own photo. The left column shows the evaluated image while the right column displays the predicted result. From up to bottom, we demonstrate the predicted text of the image using Resnet152, Resnet101, Resnet50 and VGG 19. In each model, we provide the network without attention mechanism and then followed by the one with attention mechanism.

References

- [1] P. Anderson, X. He, C. Buehler, D. Teney, M. Johnson, S. Gould, and L. Zhang. Bottom-up and top-down attention for image captioning and vqa. 2017.
- [2] X. Chen, H. Fang, T. Lin, R. Vedantam, S. Gupta, P. Dollár, and C. L. Zitnick. Microsoft COCO captions: Data collection and evaluation server. *CoRR*, abs/1504.00325, 2015.
- [3] X. Chen and C. L. Zitnick. Mind’s eye: A recurrent visual representation for image caption generation. *CVPR*, 2015.
- [4] J. Devlin, H. Cheng, H. Fang, S. Gupta, L. Deng, X. He, G. Zweig, and M. Mitchell. Language models for image captioning: The quirks and what works. *ACL*, 2015.
- [5] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *CVPR*, 2016.
- [6] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 1997.
- [7] X. Jia, E. Gavves, B. Fernando, and T. Tuytelaars. Guiding the long-short term memory model for image caption generation. *IEEE International Conference on Computer Vision*, page 2407–2415, 2015.
- [8] A. Karpathy and L. Fei-Fei. Deep visual-semantic alignments for generating image descriptions. *CVPR*, 2015.
- [9] G. Kulkarni, V. Premraj, S. Dhar, S. Li, Y. Choi, A. C. Berg, and T. L. Berg. Baby talk: Understanding and generating image descriptions. *CVPR*, 2011.
- [10] C.-Y. Lin. Rouge: A package for automatic evaluation of summaries. *ACL Workshop*, 2004.
- [11] T. Lin, M. Maire, S. J. Belongie, L. D. Bourdev, R. B. Girshick, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft COCO: common objects in context. *CoRR*, abs/1405.0312, 2014.
- [12] J. Mao, W. Xu, Y. Yang, J. Wang, Z. Huang, and A. Yuille. Deep captioning with multimodal recurrent neural networks. *ICLR*, 2015.
- [13] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu. Bleu: a method for automatic evaluation of machine translation. *ACL*, 2002.
- [14] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *ICLR*, 2015.
- [15] R. Vedantam, C. L. Zitnick, and D. Parikh. Cider: Consensus-based image description evaluation. *CVPR*, 2015.
- [16] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan. Show and tell: A neural image caption generator. *CVPR*, 2015.
- [17] K. Xu, J. Ba, R. Kiros, K. Cho, A. C. Courville, R. Salakhutdinov, R. S. Zemel, and Y. Bengio. Show, attend and tell: Neural image caption generation with visual attention. *CoRR*, abs/1502.03044, 2015.