Image Captioning based on Recurrent Neural Network Model

Heng Qiao
Department of Electrical
and Computer Engineering
University of Florida
Email: hengqiao@ufl.edu

Tong Shao
Department of Electrical
and Computer Engineering
University of Florida
Email: stlm1991@ufl.edu

Yichen Liang
Department of Electrical
and Computer Engineering
University of Florida
Email: yichenliang@ufl.edu

***Abstract*—Image captioning is the task to automatically generate the description of the content of an image, which is a fundamental problem in artificial intelligence that connects computer vision and natural language processing. In this course project, we plan to develop an image captioning system based on the recurrent neural network(RNN) Model. As the most widely used scheme, two neural networks are introduced. The first one is a pre-trained convolutional neural network (CNN) that converts the image into feature vectors, such as the VggNet, ResNet and Inception. Serving as the core of the scheme, the second one adopts the recurrent neural network (RNN) model. It takes the image feature as the input and generate the word vectors of a sentence (caption). The model is trained to maximize the likelihood of the target description sentence given the training image. Based on this, we aim at implementing some assistant techniques to further improve the performance, such as the semantic structures and more optimized neural network structures. This scheme have been implemented in Tensorflow. Experiments have been conducted on public dataset MS COCO and related evaluation scores such as the BLEU-4 score are provided. Preliminary results show that the basic model could achieve a BLEU-4 score of over 0.27, which is a pretty good performance. Further improvement in training as well as the model will be conducted in the future.**

I. Introduction and Motivation

There has been a recent surge of interest in the area of image captioning thanks to the development of machine learning technology [1]. Image captioning is the procedure to automatically generate natural language to describe the content of an image, as shown in Fig. 1. It will produce a sentence to describe the content of the image, including the activities, items, people, natural scenes and etc. This is a very crucial part of artificial intelligence, which serves as a link between visual and language. And it's important in the areas such as autopilot, Q&A system and etc.

Various methods have been developed to generate image captions and results are improving. But most of them are still far away from human performance. Those methods include the traditional retrieval (ranking) method, which considering it as finding the most matching words sequentially. And another one is the generative approach, which uses statistical model or neural network and train it to generate more natural captions.

Motivated by the fast development of this research topic, we plan to design a scheme to generate natural and accurate captions given an input image. We adopt the machine learning methods and two neural networks are introduced. The first one is a pre-trained convolutional neural network (CNN) that converts the image into feature vectors, such as the VggNet [2], ResNet [3], Inception [4] and Region based CNN [5]. Serving as the core of the scheme, the second one adopts the recurrent neural network (RNN) model [6]. It takes the image feature as the input and generate the word vectors of a sentence (caption). The target (output) of the RNN is in the form of word embeddings, which is often a 300-dimension vectors of dense representation of a word, while the total vocabulary words are around 9000. This scheme will be implemented in Tensorflow and trained and tested on public dataset MS COCO.

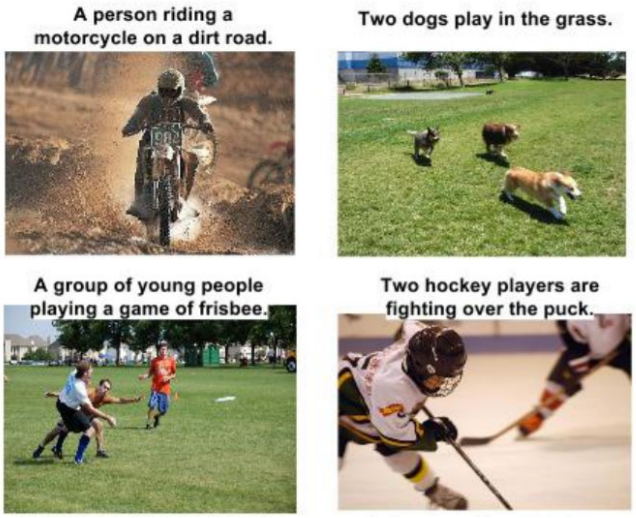

A person riding a motorcycle on a dirt road.
Two dogs play in the grass.
A group of young people playing a game of frisbee.
Two hockey players are fighting over the puck.

Fig. 1. Examples of image captioning.

II. Related Work

Related work include the traditional retrieval (ranking) method. It considers it as finding the most matching words sequentially. It usually has no or little grammar and structure constraints. Thus, the relations between words in a sentence are not modeled and the generated captions don't look very natural.

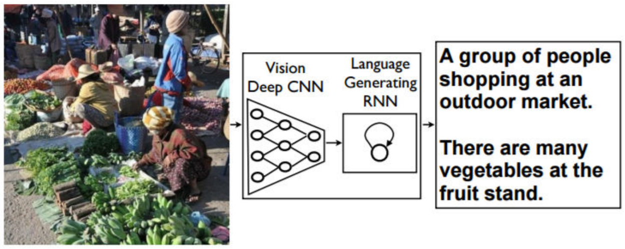

Vision
Deep CNN
Language
Generating
RNN
A group of people shopping at an outdoor market.
There are many vegetables at the fruit stand.

Fig. 2. Overview architecture of image captioning scheme.

With the help of new computing hardware and access to big amount of data, deep learning has been widely used in many areas [1], such as pattern recognition, autonomous car, and etc. remarkable learning capability of deep learning has shown in many areas. Among them, the most common form of deep learning, is supervised learning [7]. Supervised learning requires labeled training data and is often used to outputs a label when given a new input [8]. To deal with inputs with correlations in time such as an sentence, recurrent neural network (RNN) [6], typically in the form of long short-term memory (LSTM) units, is employed. All these contribute to the development of another class, which is called the generative approach. It uses statistical model or neural network and train it to generate more natural captions. One of the most common methods in image captioning is to use convolutionary neural networks to first extracts features from an image, obtaining a fixed-length vector representation of a given image or video, which is then fed into a language model, usually presented to be a recurrent neural network (RNN), typically implemented with long short-term memory (LSTM) units [9], to generate a natural language caption [10].

Recent work also consider explicit high-level semantic concepts of the input image/video. For instance, in [11], a model of semantic attention is proposed which selectively attends to semantic concepts through a soft attention mechanism.

III. Overall Architecture

We aim at designing a typical image captioning system while adding some techniques to further enhance the performance.

As shown in Fig. 2, the basic architecture has two parts. The first one is a pre-trained convolutional neural network (CNN) that converts the image into feature vectors, such as the VggNet, ResNet and Inception. And the second one adopts the recurrent neural network (RNN) model. It takes the image feature as the input and generate the word vectors of a sentence (caption).

Based on this, we will also implement some assistant techniques to further improve the performance, such as the semantic structures and more optimized neural network structures. The detailed scheme will be explained in more detail in the next section.

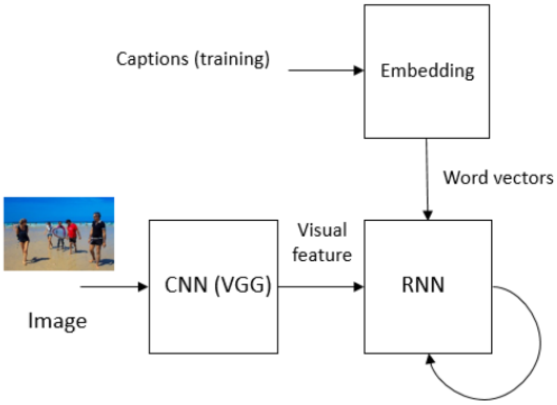


Fig. 3. Proposed the image captioning scheme.

IV. Recurrent Neural Network (RNN) Model

As shown in Fig. 3, the basic architecture has two parts. The first one is a pre-trained convolutional neural network (CNN) that converts the image into feature vector. The second one adopts the recurrent neural network (RNN) model, which is implemented with long short-term memory (LSTM) units. It takes the image feature as the input and the word embedding vectors of sentences as the output in training.

A. Image Feature Extraction

Image feature extraction could be used to identify what objects are in a certain image and where they are. This computer vision task can be implemented by some deep learning networks and we use those pre-trained convolutional neural network (CNN) to converts the image into feature vectors. The most popular ones include the VggNet [2], ResNet [3], Inception [4] and Region based CNN (R-CNN) [5].

1) Alexnet: AlexNet is the first deep learning architecture and it is simple and powerful. AlexNet includes 5 convolutional layers and 3 fully connected layers. AlexNet uses ReLU for the nonlinearity functions, so AlexNet trains much faster than other networks which use Tanh or Sigmoid function. AlexNet also solves a problem of over-fitting. To reducing over-fitting, AlexNet implements a dropout layer after every fully-connected layer. In dropout layers, each neuron has a random probability to switch off. In different training, different neurons are switched off, and only part of neurons in network are trained. In this way, the network does not rely on all neurons in this network, thus over-fitting is avoided.

2) GoogleNet: GoogleNet (Inception Network) is an architecture created by researchers in Google. The novel part of GoogleNet is inception module. In an inception module, it allows performing pooling operations and convolution operations together, then all results are concatenated. In this way, multiple features from multiple filters could help performance better, since network has many options to choose when training itself to solve the task. GoogleNet contains multiple inception modules stacked one by one. This architecture enables model to converge faster, because in one layer, there is a joint training and parallel training.

3) VGGNet: VGGNet is a 19-layers CNN with multiple 33 kernel-sized filters and 22 max pooling layers. Multiple smaller sized filters are better to learn more complex features and cost less as they could decrease the number of parameters and increase the depth of the network. Max pooling layers reduces the volume size. VGGNet has good performance in image classification and localization, and it can work as a benchmark in some tasks. Moreover, it is easy to find pre-trained networks of VGG on the Internet and apply it in different applications.

4) R-CNN: Region Based CNN (R-CNN) is the most influence advanced deep learning architectures. R-CNN is aimed to identify objects correctly in a given image with a bounding box and a label of objects. In order to implement this, R-CNN first creates a lot of bounding boxes in image, and then find if there are some boxes which include an object. Given the proposals of objects, R-CNN sends the image of objects in the box into a pre-trained AlexNet. Then, R-CNN applies a Support Vector Machine (SVM) to find out and classify the object in the box. After determining what object is inside the box, R-CNN continues to improve the result. R-CNN runs the box of image into a linear regression model. In this way, R-CNN could generate a better and tighter fitted bounding box of object as a final result.

We will try different features, while some papers show they have similar performance in computer vision tasks.

B. Sentence and Word representation

We also need to input words into LSTM as the desired output in training, which can only process numerical data. So finding a way to represent word using number is necessary. A straight forward method is to use one-hot vector, in which each word is represented by a vector with length of the total number of words in vocabulary, and each word is assigned with a unique index. The value of every elements in the vector is zero except the one with same index with corresponding word, which is set to one. For example, if the word image has index 2, then the one-hot vector for this word will be [0, 0, 1, 0, 0, , 0]. The index given to each word is random.

This method is simple and straightforward but there are some problems with this representation. First of all, since the length of each vector is equal to the size of vocabulary, which is very large most of the times. This will cause the model to have a very large amount of parameters which in turn makes it much harder to train and much easier to over-fit. Second, the representation of each word is perpendicular to each other, so that there are no similarity in vectors, even if two words could be very similar.

In our project, we use a dense instead of discrete representation of words. We hope that the length of vector could stay the same while the size of vocabulary grows, and that similar words could have similar representation on the vector space. One way to do this is using word2vec representation [13], where each word in the vocabulary is represented with a fixed length vector. Instead of having all 0s except one 1 in the vector, we have non-integer number in every elements. Thus the similarity of two vector could be calculated by measuring the distance of them, and because we have more than just one 1 in the vector, the vector with any length would have the capability of holding a vocabulary with any size. There are some more desirable features of this representation. The vector dose not just captured the similarity of words, it also captured the relations of them as shown in Fig. 4. The general idea of training for getting these vector representation is similar to auto-encoder/decoder, but in this project we use the pre-trained vector representation instead of training everything from beginning.

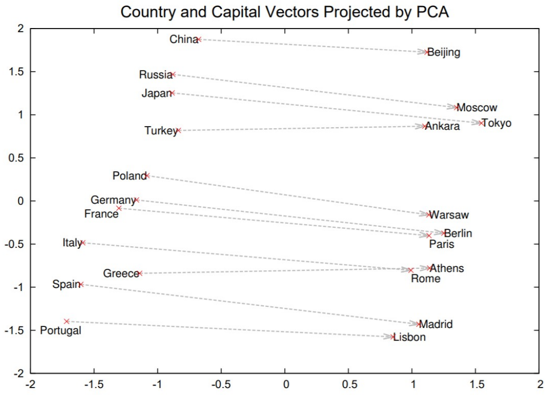


Fig. 4. The vectors of some words are projected to 2 dimension using PCA so that they could be better visualized. We can see that the vector automatically learned the concept of Country and Capital and even the relationships between them. With this feature in mind we can even do vector arithmetic on words, for example with Vector(China) C Vector(Spain)+ Vector(Beijing) we can get a vector very close to Vector(Madrid) [12].

C. Sentence Generation via RNN

As mentioned before, we adopt a generative model using recurrent neural network (RNN), which is implemented with long short-term memory (LSTM) units. In this scheme, we use LSTM to train and generate captions. This LSTM units have a recurrent layer of 512 dimensions. As shown in Fig. 5, the will be sent to LSTM via a dense connection, while each word vector in the caption has a time relation each other.

The LSTM contains the input word layer W, the LSTM units R and the output layer U. For each input word w_t, the unit r_t will get the information from w_t, r_{t-1} and the image feature. In this way, the model will learn the probability distribution of the next word given previous one as well as the image feature. Therefore, it will finally predict the next word given the first one and generate a sentence. Because it takes the relationship between each word in a sentence into account, the generated captions look more natural. The model is trained to maximize the likelihood of the target description sentence given the training image.

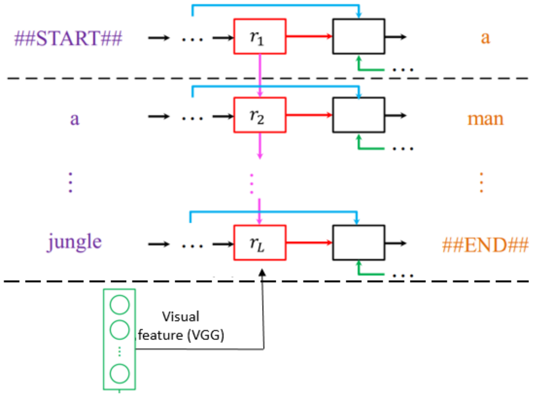


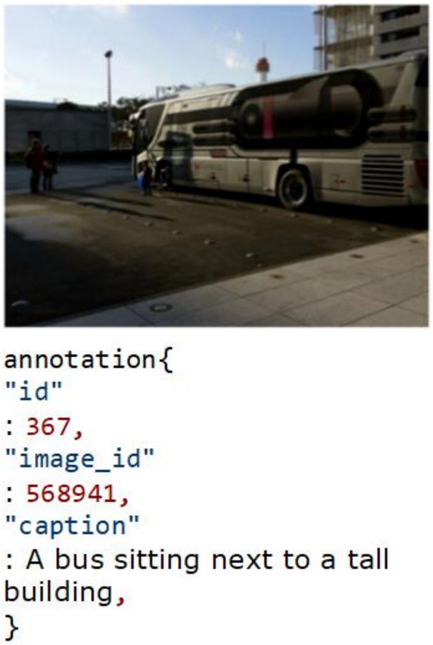
Fig. 5. The detailed structure of recurrent neural network (RNN) scheme to deal with word sequence.

V. Performance Evaluation

Our main goal is to implement an image captioning system which could automatically describe the content of an image as described before. The scheme will be implemented in Tensorflow and experiments will be conducted on MS COCO dataset. We will show some typical results as well as some score results on open dataset.

A. Experimental Setting

MS COCO dataset [14] contains roughly 80000 training images and 40000 images for testing. Each image contains up to 5 captions as groundtruth for training and testing, as shown in Fig. 6. We use MS COCO as the training and evaluation of several scores.



```
annotation{
"id"
: 367,
"image_id"
: 568941,
"caption"
: A bus sitting next to a tall
building,
}
```

Fig. 6. The MS COCO dataset's format.

TABLE I
BLEU-4 and Meteor scores of proposed schemes on MS COCO.

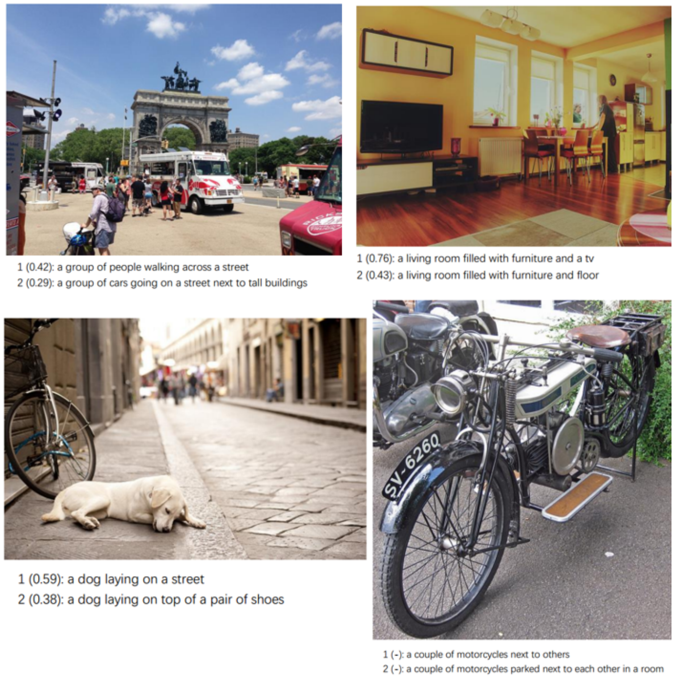
Method	Meteor Score	BLEU-4 Score
Show and Tell	0.237	0.277
Proposed	0.236	0.273

B. Performance Metrics

We evaluate the performance based on the MS COCO dataset via some popular metrics. They are BLEU, METEOR and CIDER. Basically, they tried to compared the similarity between the generated captions and the groundtruth via the euclidean distance between word vectors. The higher score usually means better captions. The detailed explanation can be found in [15], [16] and [17].

C. Experimental Results

As described before, this scheme should produce reasonable captions for input images. We present some results of images with typical scenes such as sports activities, natural scenes, indoor scenes and etc., as shown in Fig .7.



1 (0.42): a group of people walking across a street
2 (0.29): a group of cars going on a street next to tall buildings

1 (0.76): a living room filled with furniture and a tv
2 (0.43): a living room filled with furniture and floor

1 (0.59): a dog laying on a street
2 (0.38): a dog laying on top of a pair of shoes

1 (-): a couple of motorcycles next to others
2 (-): a couple of motorcycles parked next to each other in a room

Fig. 7. Typical results of our image captioning scheme.

And the current score on MS COCO is also provided in Table I, while some state-of-art methods' results are listed as well.

Some state-of-art methods' results are usually higher than 30. Thus, we still need to improve the model and fine-tune the training. And perhaps some techniques should be added as well. We hope it should be close to and even better, i.e. of the same level as the state-of-the-art methods such as the Show and Tell [10].

VI. Current Progress and Project Management

A. Current Status

Weekly plan status:
2/5-2/18: Implement basic scheme in Tensorflow. (100% finished)
2/19-2/25: Paper review and decide the assistant technique we will use to improve the scheme. (80% finished, not fully decided)
2/26-3/4: Get initial experiments results. (100% finished)
3/5-3/12: Prepare for the Midterm Presentation. (100% finished)

B. Team Coordination

Tasks for each member:
Heng Qiao: paper review, coding
Tong Shao: system design, coding
Yichen Liang: coding, reports and slides

C. Milestones, Weekly Plans, and Deliverables

Schedules:
3/13-3/19: Implement the assistant technique in the scheme.
3/20-3/26: Get full experiments results.
3/27-4/23: Write final papers and related materials for the final presentation.
4/24-4/30: Prepare for the UF Big Data Day.

Deliverables:
code
demo
paper and slides

References

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[2] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[3] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[4] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning." in *AAAI*, vol. 4, 2017, p. 12.

[5] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, 2015, pp. 91–99.

[6] T. Mikolov, M. Karafiát, L. Burget, J. Cernockỳ, and S. Khudanpur, "Recurrent neural network based language model." in *Interspeech*, vol. 2, 2010, p. 3.

[7] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[8] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[9] A. Graves, A.-r. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *Acoustics, speech and signal processing (icassp), 2013 ieee international conference on*. IEEE, 2013, pp. 6645–6649.

[10] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan, "Show and tell: A neural image caption generator," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3156–3164.

[11] Q. You, H. Jin, Z. Wang, C. Fang, and J. Luo, "Image captioning with semantic attention," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 4651–4659.

[12] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in neural information processing systems*, 2013, pp. 3111–3119.

[13] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[14] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European Conference on Computer Vision*. Springer, 2014, pp. 740–755.

[15] R. Vedantam, C. Lawrence Zitnick, and D. Parikh, "Cider: Consensus-based image description evaluation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 4566–4575.

[16] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: a method for automatic evaluation of machine translation," in *Proceedings of the 40th annual meeting on association for computational linguistics*. Association for Computational Linguistics, 2002, pp. 311–318.

[17] M. D. A. Lavie, "Meteor universal: Language specific translation evaluation for any target language," *ACL 2014*, p. 376, 2014.