Factors Influencing The Performance of Image Captioning Model: An Evaluation

Duc-Cuong Dao *
SoICT, HUST†
duccuong.hust@gmail.com

Thi-Oanh Nguyen *
SoICT, HUST†
oanhnt@soict.hust.edu.vn

Stéphane Bressan
SoC, NUS‡
steph@nus.edu.sg

ABSTRACT

Recently, neural network-based methods have shown impressive performances in captioning task. There have been numerous attempts with many proposed architectures to solve this captioning problem. In this paper, we present the evaluation of different alternatives in architecture and optimization algorithms for a neural image captioning model. First, we present the study of a image captioning model that is comprised of two modules – a convolutional neural network which encodes the input image into a fixed-dimensional feature vector and a recurrent neural network to decode that representation into a sequence of words describing the input image. After that, we consider different alternatives regarding architecture and optimization algorithm to train the model. We conduct a set of experiments on standard benchmark datasets to evaluate different aspects of the captioning system using standard evaluation methods that are utilized in image captioning literatures. Based on the results of those experiments, we propose several suggestions on architecture and optimization algorithm of the image captioning model that is balanced in terms of the performance and the feasibility to be deployed on real-world problems with commodity hardware.

CCS Concepts

•**Computing methodologies** → **Image representations;** *Natural language generation; Neural networks;*

Keywords

Image Captioning; Long Short-Term Memory; Convolutional Neural Networks; Neural Image Captioning

1. INTRODUCTION

A system that can automatically describe the content of natural images in natural sentences (e.g., English) has many practical applications. For instance, such a system can be instrumental in helping visually-impaired users to navigate in different environments such as in supermarket or in a crowded road with many transportation as well as pedestrians. Moreover, textual descriptions can provide a rich source of information from underlying image, which can be an excellent mean of visual and semantic information for image search engines. Being able to generate descriptions for images, a search engine can analyse and process query mainly based on generated descriptions and return the results in much performant speed and more relevant to users query's expectations.

However, automated image captioning, which is quite trivial to human, is a challenging problem for computer in which the captioning model needs to not only understand the visual representations of input image, but it also has to "translate" such representation into a sequence of words that is syntactically and semantically correct to describe the image.

Image captioning problem is defined as taking an image, analyzing its visual content and generating a description in natural language that reflects most important factors in the image. There have been numerous attempts to solve this problem with three main approaches, viz., template-based methods, transfer-based methods and neural network-based methods. In template-based methods, image attributes are first obtained by an object detector over the image, then they are filled into predefined sentence templates [8, 33, 19]. One major disadvantage of this method is that some descriptions cannot fit into handcrafted templates since their structure is relatively rigid. Transfer-based methods are relied on image retrieval. The captions are attained by transferring the captions of images that are visually similar to query image [21, 20, 22]. As these methods directly rely on images in training dataset, they expose a poor performance on unseen images. In contrast, neural network-based methods [32, 24, 14] neither require predefined sentence templates nor rely on examples to transfer. They are able to automatically learn the visual representation of the image, the grammatical syntax of the sentences and coordinate them together to obtain captions of the given image. Practically, neural image caption models have shown outstanding performances compared to other methods and most of recent researches on image captioning utilize this class of methods [32, 24, 14, 7, 6]. Therefore, we choose to focus on neural network-based methods in this paper.

Neural image captioning is inspired by recent advances in Neural Machine Translation [3, 28], which formulates captioning task as a translation problem. Instead of starting with a source sentence, an image is provided to initiate the generation in a word-by-word manner. There exist two different approaches for this class of method. The first approach uses a pipelined process in which a convolu-

*Corresponding authors
†School of Information and Communication Technology, Hanoi University of Science and Technology, Hanoi, Vietnam
‡School of Computing, National University of Singapore, Singapore

MoMM '16, November 28 - 30, 2016, Singapore, Singapore

ISBN 978-1-4503-4806-5/16/11...$15.00

DOI: http://dx.doi.org/10.1145/3007120.3007136

tional neural network is exploited to generate bag of words conditioned on the input image. Then a maximum entropy language model is used to rearrange these words to form a coherent sentence [7]. The second approach explores models that are trained in an "end-to-end" fashion. Specifically, the activations at the final hidden layer of an object detection CNN are used as input to a recurrent neural network language model [14, 24, 2]. Devlin *et al.* [5] has shown the later approach achieves better performance on standard evaluation metrics. Kiros *et al.* [16, 17] tackles the problem by co-embedding image features with previously generated word in a multimodal co-embedding space. Karpathy *et al.* [14], Mao *et al.* [24] use a multimodal approach which embeds image features extracted by a CNN and word vectors learned by a RNN together to generate captions. Karpathy's model exploits a structured objective to align words into a proper sentence while Mao's model learns to generate word by maximizing the probability distribution of generating a word given previous words and image representation.

In this work, we study an approach by Vinyals *et al.* [32], namely *Show and Tell*, which proposes a neural probabilistic image captioning model consisted of two modules - visual module and language module. The visual module is a convolutional neural network (CNN) which encodes input image into a fixed-dimensional feature vector. At the core of the language module is a variation of recurrent neural network (RNN), dubbed *Long Short-term Memory* (LSTM) [12] decoding that vector into a sequence of words describing the image (as illustrated in Figure 1) . The model is trained in an "end-to-end" manner. We then propose several alternatives to the visual module of the model. Another important factor when building a neural network-based model is the choice of optimization algorithms to train the model. For this, we empirically evaluate three different optimization methods to train the model on MSCOCO dataset [23]. Based on those observations, we propose several suggestions regarding the architecture and optimization algorithms for neural image captioning model in favor of the balance between the performance and the feasibility to be deployed on real-world applications with commodity hardware.

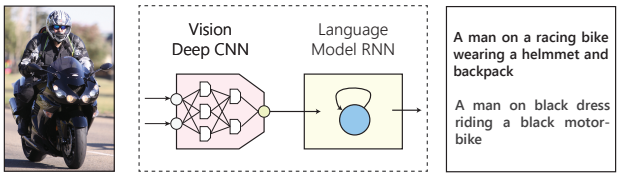


Figure 1: Show and Tell's general architecture: a CNN encoding input image into a fixed-dimensional feature vector and a RNN decoding that vector into a sequence of words describing the image (adapted from [32])

The contribution of our work are two-fold. First, we provide a detailed explanation of the architecture and implementation of an automated image captioning model with convolutional and recurrent neural networks. The model is trained in an "end-to-end" fashion and can generate sensible captions for various types of images. Second, we conduct an extensive set of experiments to evaluate different configurations of the model on standard benchmark datasets and evaluation metrics. Based on such experiments, we comparatively suggest the optimal settings for certain application.

The rest of this paper is organized as follows: the details of our methodology, including model's architecture and optimization algorithms, are described in Section 2. Next, Section 3 gives the details of experiments and evaluations on standard benchmark datasets including MSCOCO and Flickr30k [34] using standard metrics BLEU [25], METOER [4] and CIDEr [31]. Finally, some conclusions and further development directions are drawn in Section 4.

2. METHODOLOGY

This section presents the details of our methodology. We start by investigating two CNNs as the vision module and an LSTM-based language module. After that, we present briefly several algorithms which can be used to train the captioning model.

2.1 Model architectures

The model used in this paper is originally introduced by Vinyals *et al.* [32]. Inspired by recent advances in neural machine translation, the author proposes a neural and probabilistic model which is trained to directly maximize the probability of obtaining the correct caption sentence given the input image.

$$\theta^* = \text{argmax}_\theta \sum_{(I,S)} \log p\left(S|I;\theta\right) \tag{1}$$

where θ is the parameters of the model, I is the input image, S is the output sequence of words that make up the description for image I.

2.1.1 Vision module

Originally, [32] exploits a complex convolutional neural network as their vision module, namely GoogleNet [29]. In a nutshell, the network is made up of 9 units called *Inception module* stacked together with occasional pooling layers and convolutional layers. Albeit their model achieves state-of-the-art results on full-frame image captioning, there is no clear explanation from their work on why they choose such CNN and only results with GoogleNet are shown.

We are motivated to investigate the effects of different convolutional neural networks on the overall performance of captioning model. Therefore, we consider two other well-known convolutional neural networks, AlexNet [18] and VGGNet [27], as the vision module of the model.

AlexNet [18] is the first convolutional neural network to be applied and make a breakthrough in large-scale image classification. The network is comprised of 7 layers: 5 convolutional layers (CONV1 to CONV5) optionally followed by max-pooling layer and 2 fully connected layers (FC6 and FC7). The input to the network is a square image of size $224 \times 224 \times 3$ and the output of the last layer of the network is fed into a 1000-way softmax function in ImageNet ILSVRC classification task [26]. The receptive field size [1] is gradually reduced from 11×11 in CONV1 to 3×3 in CONV5. In total, this network has approximately 60 million parameters.

VGGNet [27] is a deep convolutional neural network with 16 layers. Analogous to AlexNet, this network is also formed by stacking together many convolutional layers followed by nonlinear and pooling layers. The input to VGGNet is also an image of size $224 \times 224 \times 3$ and the output of the network is also fed into a 1000-way softmax layer for classification task. The differences between VGGNet and AlexNet is not only in the depth of network but also in the size of receptive field. VGGNet uses a fixed receptive field size of 3×3. This permits the addition of more convolutional

[1] size of the filter in each layer

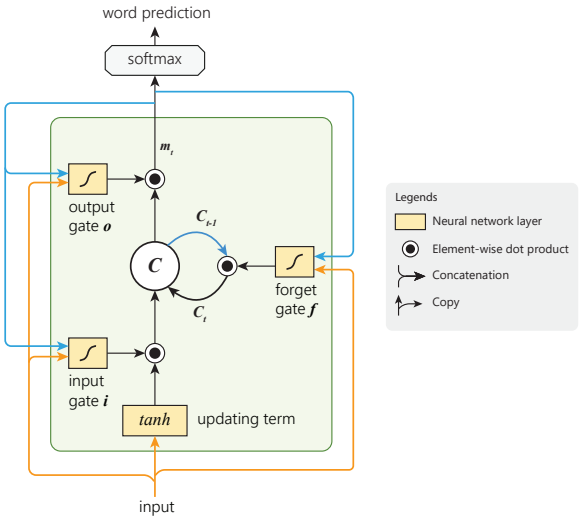


Figure 2: Long short-term memory block with one cell for word prediction (adapted from [32]).

layers to the network, which results in a network of 16 layers [2]. The network has roughly 138 million parameters.

2.1.2 *Language module*

To compose natural image descriptions, a language module has to not only capture the representation of individual words but also integrate the inter-relationship between words in a sentence. Recurrent neural network has been shown to be very good at sequence modeling. However, vanilla RNN suffers from vanishing and exploding gradients problem [12] due to long-term dependencies, which makes them very difficult to train. Following [32], we use Long short-term memory (LSTM) [12], a special RNN architecture designed to cope with those problems, as the language model for captioning model.

The LSTM contains special units called *memory blocks* in the recurrent hidden layer. At the core of each memory block is a *memory cell* C which encodes knowledge at every timestep of what inputs have been observed up to this step. Figure 2 shows the architecture of a LSTM with one memory block that is used for word prediction. The behavior of C is governed by three gates - the input gate (i), the output gate (o) and the forget gate (f). The three gates are nonlinear summation units that collect activations from inside and outside the block, and control the activation of the cell via multiplications (element-wise matrix multiplications).

At each timestep t, the equations for the gates, cell update and output of LSTM are as follows:

$$i_t = \sigma\left(W_{ix}x_t + W_{im}m_{t-1} + b_i\right) \tag{2a}$$
$$f_t = \sigma\left(W_{fx}x_t + W_{fm}m_{t-1} + b_f\right) \tag{2b}$$
$$o_t = \sigma\left(W_{ox}x_t + W_{om}hm_{t-1} + b_o\right) \tag{2c}$$
$$C_t = f_t \odot C_{t-1} + i_t \odot g\left(W_{cx}x_t + W_{cm}m_{t-1} + b_g\right) \tag{2d}$$
$$m_t = C_t \odot o_t \tag{2e}$$
$$z_t = f\left(m_t\right); \quad z = \sum_{t=1}^{T} z_t \tag{2f}$$

where

- $\sigma\left(x\right) = \dfrac{1}{1+e^{-x}}$
- $g\left(x\right) = \tanh x = \dfrac{e^x - e^{-x}}{e^x + e^{-x}} = 2\sigma\left(2x\right) - 1$
- $\odot$ denotes element-wise multiplication
- m_t is the hidden state at timestep t
- $b_{\{i,f,o,g\}}$ are the biases of different gates
- f is a function to be applied to the output of the LSTM. It is problem dependent.
- T the maximum length of generated caption

The activation function of three gates is sigmoid (σ) which squashes the input into range $[0, 1]$. Thus, the gates can optionally decide whether to let the information from previous timesteps go through (i.e. when gate's value is 1) or not (i.e., when gate's value is 0).

2.2 Optimization algorithms

One of the important design choices of a neural network-based system is the optimization algorithm used to train that system. Stochastic gradient descents (SGD) has been extensively employed to train deep neural networks due to their ease in implementation and robustness for problems with a lot of training samples. Vanilla SGD, however, does not guarantee a good convergence, but introduces several challenges that need to be addressed :

- *Choosing a proper learning rate* η is difficult. A learning rate that is too large can lead to the loss function eventually diverges. A too small learning rate makes the convergence much slowly.
- *Learning rate schedules* try to adjust the learning rate η by reducing it according to a pre-defined schedule or change in the loss function between epochs falls below a threshold. These schedules and thresholds have to be defined in advance, thus are unable to adapt to the characteristics of new datasets.
- *Saddle points* - the points where one dimension slopes up and the other slopes down - are often surrounded by a plateau of the same error in the error surface, which makes it considerably hard for SGD to escape, as the gradient is close to zero in all dimensions.

There have been many attempts to improve the efficiency of SGD, i.e., to overcome the above challenges. The following sub-sections briefly present two of popular optimization algorithms, dubbed RMSProp and Adam that we are interested to investigate in experiment section.

2.2.1 *RMSProp*

Basically, SGD perform updates on parameters on each sample (or mini batch of sample) of the current iteration. Hence, the loss tends to be noisy and unstable. To solve this problem, Momentum method can be used. Intuitively, momentum keeps track of history of gradient updates and combine them with current update to make

[2] In their paper, Simonyan *et al.* also proposes a 19-layer version of VGGNet. However, they show that its performance is not as good as that of the 16-layer network. Therefore, in this work, we stick with the 16-layer VGGNet

the update more stable. The Momentum method can be formulated as:

$$v_t = \gamma v_{t-1} - \eta \cdot \frac{\partial \mathcal{L}(\theta)}{\partial \theta} \quad (3)$$

where v_t is the momentum of gradients at timestep t; γ denotes the momentum coefficient which is usually set to 0.9 in practice; $\mathcal{L}$ is the objective function (loss function) and θ is the parameter of the model.

RMSProp [30] is based on this momentum idea with further enhancement is that it keeps running average of its recent gradient magnitudes and divides the next gradient by this average so that loosely gradient values are normalized. The description of RMPS is outline in Algorithm 1.

Algorithm 1 The RMSProp algorithm

Require: Learning rate η, decay rate α
Require: Initial parameters θ
Require: Constant ϵ used to stabilize division by small numbers
1: Initialize accumulation variables $\mathbf{r} = 0$
2: **while** stopping criterion is not satisfied **do**
3: Take a minibatch of m samples $\left(x^{(i)}; y^{(i)}\right) \in \mathcal{D}$
4: Compute gradient: $\mathbf{g} \leftarrow \frac{1}{m} \nabla_\theta \sum_i \mathcal{L}\left(f\left(x^{(i)}; y^{(i)}\right)\right)$
5: Accumulate squared gradient: $\mathbf{r} \leftarrow \alpha \mathbf{r} + (1-\alpha)\, \mathbf{g} \odot \mathbf{g}$
6: Compute parameter update: $\Delta\theta = -\frac{\eta}{\sqrt{\epsilon + \mathbf{r}}} \odot \mathbf{g}$
7: Update the parameter: $\theta \leftarrow \theta + \Delta\theta$
8: **end while**

The associated parameters for RMSProp are learning rate η, decay rate α and constant ϵ.

2.2.2 Adam

Adam [15] is an improved version of RMSProp. The description of this optimization algorithm is as in Algorithm 2.

Algorithm 2 *Adam*: an algorithm for stochastic optimization

Require: η: learning rate
Require: $\beta_1, \beta_2 \in [0, 1)$: Exponential decay rates for the moment estimates
Require: $f(\theta)$: Objective function with parameter θ
Require: θ_0: Initial parameter vector
1: $m_0 \leftarrow 0$ (Initialize 1^{st} moment vector)
2: **while** stopping criterion is not satisfied **do**
3: $\mathbf{g_t} \leftarrow \nabla_\theta f_t(\theta_{t-1})$ (Get gradients w.r.t object at timestep t)
4: $m_t \leftarrow \beta_1 \cdot m_{t-1} + (1-\beta_1) \cdot \mathbf{g_t}$ (Update biased first moment estimate)
5: $v_t \leftarrow \beta_2 \cdot v_{t-1} + (1-\beta_2) \cdot \mathbf{g_t}^2$ (Update biased second raw moment estimate)
6: $\hat{m_t} \leftarrow m_t / \left(1 - \beta_1^t\right)$ (Compute bias-corrected first moment estimate)
7: $\hat{v_t} \leftarrow v_t / \left(1 - \beta_2^t\right)$ (Compute bias-corrected second raw moment estimate)
8: $\Delta\theta = -\eta \frac{\hat{m_t}}{\left(\sqrt{\hat{v_t}} + \epsilon\right)}$ (Compute parameter updates)
9: $\theta_t \leftarrow \theta_{t-1} + \Delta\theta$ (Update parameters)
10: **end while**

There are a few important differences between RMSProp and Adam. While RMSProp generates its parameter updates using momentum on a rescaled gradient, Adam updates are directly estimated using a running average of the first and second moment of the gradient. RMSProp also lacks the bias-correction term; this matter most in the case of small value of β_2 since in that case not correcting the bias leads to very large learning rate and often divergence.

3. EXPERIMENTS AND RESULTS

This section presents the details of experiments we conduct to evaluate our alternatives for vision module and optimization algorithms for image captioning model. We start by giving information about the data sets as well as the evaluation metrics for all experiments in Section 3.1. The details of experiments and results on different optimization algorithms; LSTM network size and vision modules are described in Section 3.2, Section 3.3 and Section 3.4 respectively.

3.1 Dataset and Evaluation metrics

To assert whether a generated caption is successful or not is a nontrivial task due to the fact that natural language (e.g., English) is inherently ambiguous. However, since most of the literatures on image captioning relies on metrics such as BLEU [25], METOER [4], CIDEr [31] to evaluate their models, we report our evaluation on those metrics. In addition, we visually inspect all generated captions on the test data set and classify them into four groups: (1) describe without errors, (2) describe with minor errors, (3) somewhat related to the image and (4) not related to the image.

Table 1: Dataset statistics

Dataset	Image			Caption
	train	val	test	vocabulary size
Flickr30K	29,738	1,000	1,000	6,987
MSCOCO	82,783	20,504	5,000	9,613

For evaluation we use standard datasets, namely MSCOCO [23] [3] and Flickr30k [34] [4]. Each image in those datasets comes with 5 groundtruth captions. The statistics of each dataset are shown in Table 1.

Since MSCOCO and Flickrs30k are not accompanied with a test set in their public distributions, we follow [14, 32] whom extract $1,000$ and $5,000$ images from the whole Flickrs30k and MSCOCO, respectively, to form the test set. The remains are reserved for training and validation sets.

The preprocessing step includes tokenizing sentences and removing words whose occurrence is less than five. All images are squashes into a square size of 224×224 to fit the input size of vision module.

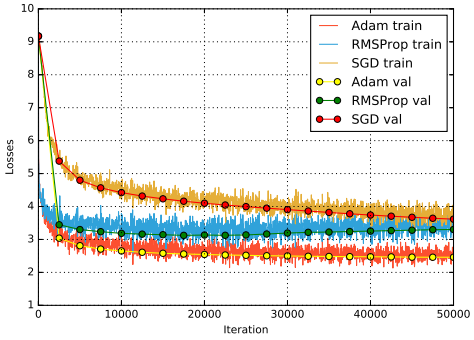
We conduct our experiments on a NVIDIA Tesla K80 [GK210B] board with 12GB VRAM, CPU Intel Xeon E5-2620v3. Albeit the Tesla board comes with two NVIDIA cards internally, we only train all models on a single graphic card. In addition, the implementation is based on NeuralTalk [5] public repository.

3.2 Optimization algorithms

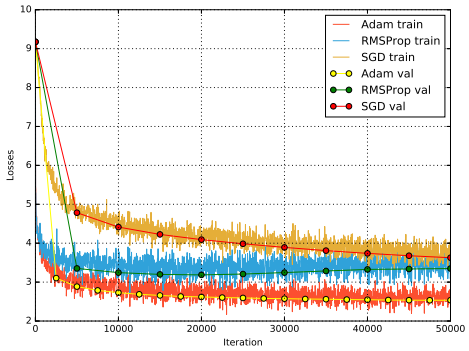
Empirically, in order to get good performance with SGD, one needs to manually adjust the initial value of learning rate for each model and each problem, as well as design a suitable schedule to

[3] http://mscoco.org
[4] http://shannon.cs.illinois.edu/DenotationGraph/
[5] https://github.com/karpathy/neuraltalk

update that learning rate if necessary. This experiment examines SGD, RMSProp and Adam optimizers for stochastic optimization with respect to the convergence of objective function as well as overall performance of the captioning model. The settings for those optimization algorithms are as follow: SGD ($\eta = 0.01$, no momentum and weight decay [6]), RMSProp ($\eta = 0.0005, \alpha = 0.8, \epsilon = 10^{-8}$) and Adam ($\eta = 0.0005, \alpha = 0.8, \beta = 0.999, \epsilon = 10^{-8}$). Hyperparameters $(\alpha, \beta, \epsilon)$ for RMSProp and Adam are chosen as recommended by their papers [30] and [15] respectively. The learning rate η is empirically selected using strategy proposed by [1]. We start out with several η values on small subset of training set and train the models within $50,000$ iterations. η is manually adjusted so that the loss function moves towards convergence.



(a) VGGNet as CNN module



(b) AlexNet as CNN module

Figure 3: Comparison of 3 optimization algorithms: SGD, RMSProp and Adam with respect to training/validation losses.

Figure 3 visualizes the loss function on training and validation sets within the first $50,000$ iterations. In both cases, Adam algorithm gives the fastest convergence rate; the loss function quickly reaches a "plateau" around 2.5 after approximately $25,000$ iterations. AlexNet model converges slightly slower than the VGGNet one. Pure SGD (with no momentum and weight decay) has the lowest convergence rate [7]. Generally, for all algorithms and models, the loss function steadily descends toward converged direction, which indicates the chosen learning rates and other hyperparameters are properly suitable for this problem.

Table 2 shows the language evaluation scores on MSCOCO test data set. As can be seen from the table, regardless of the vision module, models trained with Adam still achieves the best results on most of the scores, except for the case of VGGNet model trained with RMSProp which gets 21.2 point on BLEU-4. When trained with a same optimization algorithms, VGGNet model outperforms AlexNet model on all scores. VGGNet model trained with Adam is comparably on par with *Show and Tell*, especially on BLEU-2 (45.5 vs 46.1) and BLEU-3 (31.1 vs 32.9) scores. It is also better than other recent methods LRCN [6] and BRNN [14]. In addition, while not being as good as VGGNet model, AlexNet model are still more performant than nearest neighbor method.

Basically, BLEU score measures the correlation between generated captions and referential groundtruth sentences based on n-gram comparison. We hypothesize that BLEU-2 and BLEU-3 are more reflective than BLEU-1 and BLEU-4 for evaluation of automatically generated captions for given images. These scores balance between the strong correlation of generated captions with groundtruths and the flexibility for novel captions that might not be presented in the training set. That is to say, with aforementioned results, VGGNet model trained with Adam algorithm is a credible alternative for original GoogLeNet with pure SGD. Moreover, since Adam ($\eta = 0.0005, \alpha = 0.8, \beta = 0.999, \epsilon = 10^{-8}$) yields fastest convergence speed, it will used as the optimization algorithm in subsequent experiments.

3.3 LSTM size

Beside the learning rate η and optimization algorithm, the hidden layer size h is an important hyperparameter that affects the performance of LSTM network [9]. This experiment is conducted to evaluate the effects of different hidden sizes h to the captioning model so as to figure out a suitable one as a trade-off between the performance of the model and the required training time as well as the computational capability of machine used in inference phase. To this end, we consider three different values of h that are largely used in researches of LSTM. The learning rate is set to $\eta = 0.0005$, and the optimization method is Adam.

The plots in Figure 4 show that all three LSTM hidden sizes result in the remarkably similar convergence rates, both training and validation losses converges quickly. Furthermore, as expected, larger LSTM network performs better, however the required training time and the number of network's parameters are also considerably increased. Therefore, the model might not be usable when it is deployed to different machines at the test time even though prediction is made by forwarding the inputs to the model. It can also be inferred from the figure that with the same number of hidden units h, VGGNet model yields a better result than AlexNet model in both BLEU and METEOR scores. A model formed by AlexNet as CNN and a LSTM of $1,000$ hidden units only gets scores comparably good as a model whose VGGNet is used as CNN and LSTM with 256 hidden units. As a trade-off, we recommend to use LSTM with $h = 512$ hidden units.

3.4 Vision module

A good CNN, which can extract salient information from the raw image, is essentially an important component of the captioning model. Recent researches have proposed several deep architectures with impressive performances [18, 27, 11, 10]. The general trend is to increase the depth of the network - the number of layers within the network. However, deeper networks have significantly more parameters, which makes them harder to train and more prone to overfitting.

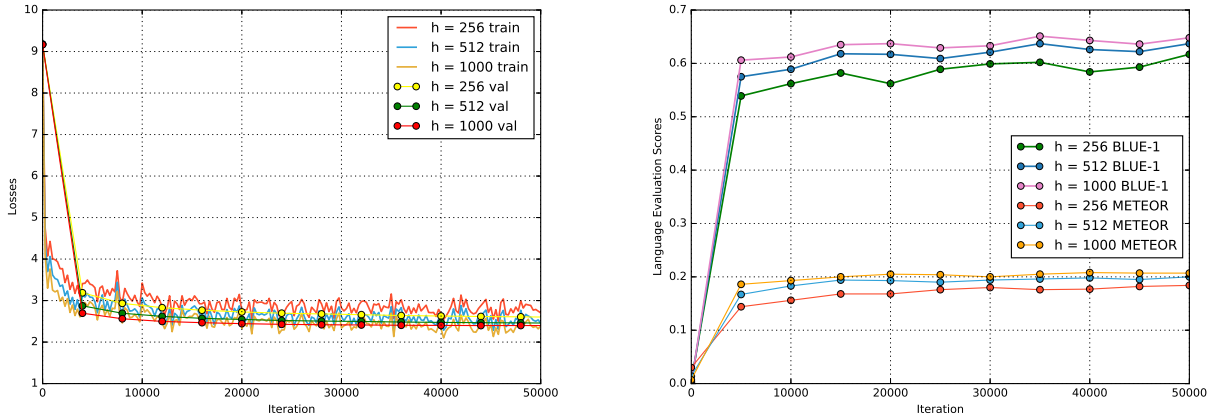
This experiment examines two CNNs as the vision module, namely AlexNet and VGGNet. We settle other settings by using Adam

[6] As in original Show and Tell paper

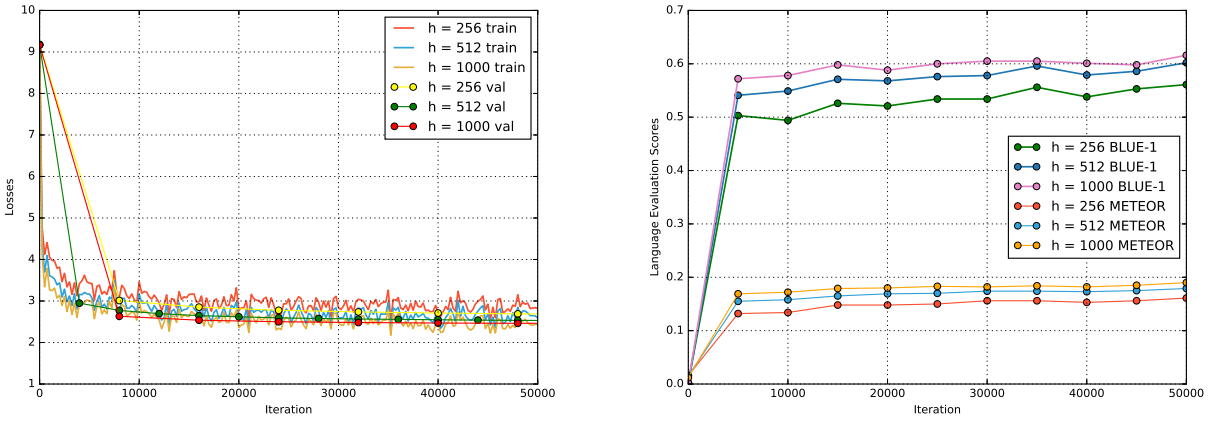
[7] SGD with learning rate $\eta = 0.0005$ converges significantly slower than the other methods with same η

Table 2: Evaluation of image captions generated on MSCOCO test set. B-n is BLEU score that uses up to n-grams.

Model	Optimizer	B-1	B-2	B-3	B-4	METEOR	CIDEr
VGGNet	Adam	**64.1**	**45.5**	**31.1**	19.8	**20.9**	**69.1**
	RMSProp	63.2	44.4	30.3	**21.2**	20.3	66.8
	SGD	57.7	37.7	23.3	14.7	16.7	41.5
AlexNet	Adam	**60.8**	**41.5**	**27.9**	**19.2**	**18.4**	**54.8**
	RMSProp	59.6	40.1	26.3	17.6	18.3	52.2
	SGD	56.3	36.1	22.2	14.0	16.0	37.1
LRCN[6]	–	62.8	44.2	30.4	–	–	–
BRNN[14]	–	62.5	45.0	32.1	23.0	19.5	66.0
Show and Tell [32]	SGD	***66.6***	***46.1***	***32.9***	***24.6***	–	–
Nearest Neighbor [14]	–	48.0	28.1	16.6	10.0	15.7	38.3



(a) VGGNet model



(b) AlexNet model

Figure 4: Comparison of LSTM hidden sizes: $h \in \{256, 512, 1000\}$ with respect to training/validation losses and language evaluation score. Due to limited space, only BLEU-1 and METEOR scores are shown in language evaluation score plots

($\eta = 0.0005, \alpha = 0.8, \beta = 0.999, \epsilon = 10^{-8}$) as the optimizer and LSTM with 512 hidden units as the language module. The CNN weights are initialized with pre-trained models on ImageNet dataset [26]. Following [32], we let the LSTM weights uninitialized for randomly initializing them does not result in significant improvement.

Table 3 shows comparison on different criteria for AlexNet and VGGNet as the vision module. As expected, AlexNet model ex-

Table 3: Comparison between AlexNet and VGGNet as the vision module of captioning models

Criterion	AlexNet	VGGNet
#parameters of the CNN	$56,723,488$	$136,358,208$
#parameters of captioning model	$68,631,936$	$148,266,656$
Size of model	207 MB	566 MB
Single forward on GPU	2.0479 ms	3.6705 ms
Single forward on CPU	1894.63 ms	3667.99 ms
BLUE-1	60.8	64.1
METOER	18.4	20.9
CIDEr	54.8	69.1

hibits an inferior performance over VGGNet model for scores computed on all generated captions over images in test dataset. However, it is also clear that AlexNet model is twice smaller and attains faster inference speed than VGGNet model. In addition, AlexNet model is still far better than the traditional nearest neighbor method. Nonetheless, one might come up with an obvious question: *"Can AlexNet model generate sensible and reliable captions?"*

Figure 5 illustrates the generated captions for randomly-sampled images in MSCOCO test data set. The results are shown for both AlexNet and VGGNet along with the groundtruth captions. As can be seen from the figure that both model are able to generate sensible captions that describe the important visual information of given images. These models tend to generate sentences which are semantically analogous to those in the training set. Moreover, the generated captions are grammatically correct though there is no explicit grammatical rules in the training set. The AlexNet model generates captions that are relatively consensus to the one produced by VGGNet model though they are less detailed than the VGGNet generated captions. For example, in the first image of Group 2 in Figure 5, both models can recognize the *"living room"* scene; but AlexNet model was not able to generate captions for two people while VGGNet model can easily do this. There is the case where AlexNet model fails to generate correct caption while VGGNet can generate captions without any errors as shown in the third image of Group 2 in Figure 5. VGGNet generated caption *"A young girl riding a skateboard on a ramp"* captures salient information of the image while AlexNet model's caption *"A man is holding a tennis racket in his hand"* is completely not related to the image.

From those observations, regarding the choice of vision module, VGGNet is a good alternatives to GoogLeNet without much performance hurt. AlexNet, though does not achieves as good results as VGGNet on standard evaluation metrics, is lighter and attains faster speed in inference phase. Despite being less detailed when compared to captions generated by VGGNet model, captions generated by AlexNet can generally capture semantic visual information of the input image. Therefore, for applications that speed and compactness are more preferred than detailed captions such as those in mobile applications, AlexNet can be a sufficient choice.

4. CONCLUSIONS

In this paper, we present the study of a neural image captioning model which is composed of a convolutional neural networks for visual feature extractions and a long short-term memory network for language modeling. Additionally, we propose the alternatives to different module of the model as well as consider different algorithms to train the model. We conduct an extensive set of experiments to validate our proposals on standard benchmark datasets (MSCOCO, Flickr30K) using standard evaluation metrics (BLUE, METEOR, CIDEr scores).

It is clear to see that the model trained with proposed alternatives and optimization can generally generate syntactically and semantically correct English sentences to describe the content of given images. The proposed enhancements result in models that are not as complex as the original one while being on par on standard evaluation metrics. Specifically, we suggest to train the model with Adam optimization algorithm for its robustness in terms of convergence. VGGNet can be used as a substitution for GoogLeNet as the vision module of the model without serious degradation to performance of the model. In addition, AlexNet can also be used for those systems whose speed and compactness of the model are more preferred than detailed captions of intricate information of input image.

Though presented models are able to generate sensible captions for given image, they only take a single look at the input image, which limits their ability to recognize and generate detailed captions. Therefore, in the future we want to incorporate object detection mechanisms into the captioning model to generate richer descriptions of the image. In addition, we are interested in researching methods to deploy neural captioning models into mobile applications or autonomous driving cars. At the moment, most of the heavy work when employing the model lies on the CNN component which often requires powerful discrete-graphical devices for running. Hence, neural captioning models cannot be fully embedded into mobile applications. One possible direction is to compress the model as in [13]. Nevertheless, more thorough investigations need to be conducted.

5. ACKNOWLEDGMENTS

This research is funded by the Hanoi University of Science and Technology (HUST) under project number T2016-PC-054 and is partially supported by the National Research Foundation, Prime Minister's Office, Singapore under its Campus for Research Excellence and Technological Enterprise (CREATE) programme. NUS Ref: R-702-005-101-281.

6. REFERENCES

[1] L. Bottou. *Stochastic Gradient Tricks*, volume 7700, pages 430–445. Springer, January 2012.

[2] X. Chen and C. Lawrence Zitnick. Mind's eye: A recurrent visual representation for image caption generation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2422–2431, 2015.

[3] K. Cho, B. Van Merriënboer, Ç. Gülçehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar, Oct. 2014. Association for Computational Linguistics.

[4] M. Denkowski and A. Lavie. Meteor universal: Language specific translation evaluation for any target language. In *Proceedings of the EACL 2014 Workshop on Statistical Machine Translation*, 2014.

[5] 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. *CoRR*, abs/1505.01809, 2015.

[6] J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell.

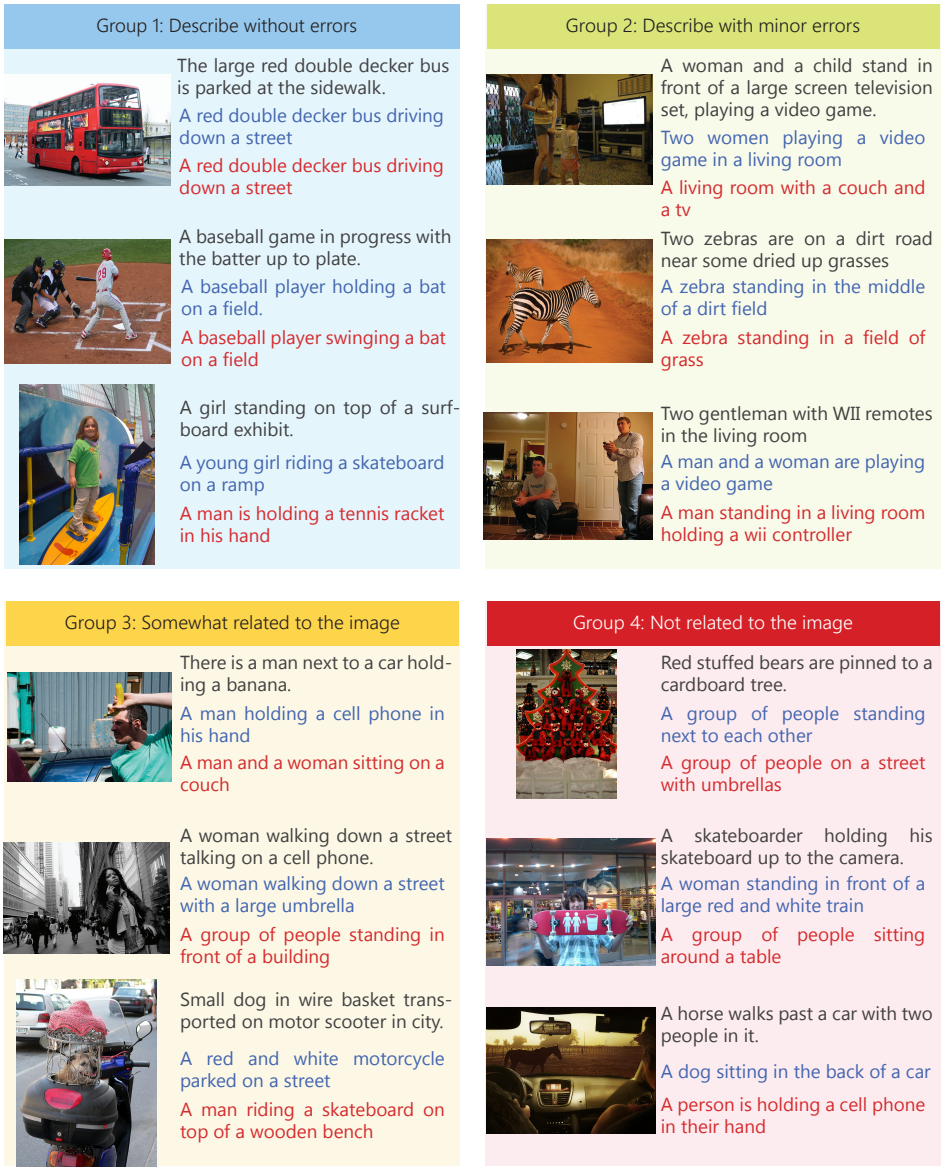


Figure 5: Samples of generated captions on MSCOCO test set. The resultant captions are grouped into 4 groups by human rating. For each image, first sentence is the groundtruth; the second sentence is caption generated by VGGNet model and the last one is generated by AlexNet model. Captioning results are visually classified into 4 groups: (1) describe without errors, (2) describe with minor errors, (3) somewhat related to the image and (4) not related to the image. The classification is based on captions generated by VGGNet model. AlexNet model's captions are included for comparison.

Long-term recurrent convolutional networks for visual recognition and description. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2625–2634, 2015.

[7] H. Fang, S. Gupta, F. N. Iandola, R. K. Srivastava, L. Deng, P. Dollár, J. Gao, X. He, M. Mitchell, J. C. Platt, C. L. Zitnick, and G. Zweig. From captions to visual concepts and back. *CoRR*, abs/1411.4952, 2014.

[8] A. Farhadi, M. Hejrati, M. A. Sadeghi, P. Young, C. Rashtchian, J. Hockenmaier, and D. Forsyth. Every picture tells a story: Generating sentences from images. In *European Conference on Computer Vision*, pages 15–29. Springer, 2010.

[9] K. Greff, R. K. Srivastava, J. Koutník, B. R. Steunebrink, and J. Schmidhuber. Lstm: A search space odyssey. *arXiv preprint arXiv:1503.04069*, 2015.

[10] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.

[11] K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1026–1034, 2015.

[12] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, Nov. 1997.

[13] F. N. Iandola, M. W. Moskewicz, K. Ashraf, S. Han, W. J. Dally, and K. Keutzer. Squeezenet: Alexnet-level accuracy

with 50x fewer parameters and <1mb model size. *CoRR*, abs/1602.07360, 2016.

[14] A. Karpathy and L. Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3128–3137, 2015.

[15] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014.

[16] R. Kiros, R. Salakhutdinov, and R. S. Zemel. Multimodal neural language models.

[17] R. Kiros, R. Salakhutdinov, and R. S. Zemel. Unifying visual-semantic embeddings with multimodal neural language models. *arXiv preprint arXiv:1411.2539*, 2014.

[18] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.

[19] G. Kulkarni, V. Premraj, V. Ordonez, S. Dhar, S. Li, Y. Choi, A. C. Berg, and T. L. Berg. Babytalk: Understanding and generating simple image descriptions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(12):2891–2903, 2013.

[20] P. Kuznetsova, V. Ordonez, A. C. Berg, T. L. Berg, and Y. Choi. Collective generation of natural image descriptions. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 359–368. Association for Computational Linguistics, 2012.

[21] P. Kuznetsova, V. Ordonez, A. C. Berg, T. L. Berg, and Y. Choi. Generalizing image captions for image-text parallel corpus. Citeseer, 2013.

[22] P. Kuznetsova, V. Ordonez, T. L. Berg, and Y. Choi. Treetalk: Composition and compression of trees for image descriptions.

[23] 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*, pages 740–755. Springer, 2014.

[24] J. Mao, W. Xu, Y. Yang, J. Wang, Z. Huang, and A. Yuille. Deep captioning with multimodal recurrent neural networks (m-rnn). *arXiv preprint arXiv:1412.6632*, 2014.

[25] 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*, ACL '02, pages 311–318, Stroudsburg, PA, USA, 2002. Association for Computational Linguistics.

[26] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.

[27] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 201.

[28] I. Sutskever, O. Vinyals, and Q. V. Le. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112, 2014.

[29] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–9, 2015.

[30] T. Tieleman and G. Hinton. Lecture 6.5—rmsprop: Divide the gradient by a running average of its recent magnitude. COURSERA: Neural Networks for Machine Learning, 2012.

[31] R. Vedantam, C. L. Zitnick, and D. Parikh. Cider: Consensus-based image description evaluation. *CoRR*, abs/1411.5726, 2014.

[32] 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*, pages 3156–3164, 2015.

[33] Y. Yang, C. L. Teo, H. Daumé III, and Y. Aloimonos. Corpus-guided sentence generation of natural images. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 444–454. Association for Computational Linguistics, 2011.

[34] P. Young, A. Lai, M. Hodosh, and J. Hockenmaier. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *Transactions of the Association for Computational Linguistics*, 2:67–78, 2014.