Bilinear Pooling and Co-Attention Inspired Models for Visual Question Answering

Fei "Frank" Yang
Stanford University
Department of Electrical Engineering
feiyang@stanford.edu

Ayooluwakunmi Jeje
Stanford University
Department of Computer Science
jejekunmi@stanford.edu

Abstract

In recent years, open-ended visual question answering has been an area of active research. In this work, we present our exploration of two state-of-art architectures including the Multi-modal Compact Bi-linear Pooling (MCB) and Dynamic Memory Network (DMN) and analysis of the result and performance of the models. We found both models to perform comparably on the VQA v2.0 dataset based on predicted answer accuracy. We also qualitatively analyzed how the models capture the interaction between images and questions by visualizing the attention maps and saliency maps of our models.

1. Introduction

The simple task of answering a question about a picture is immediately relatable to anyone who has ever had to identify a giraffe in a vacation photo or describe the color of a dress in a family portrait. Visual Question and Answering (VQA) defined in this natural form of answering an open-ended question about a given image has been a notoriously difficult problem to tackle because it's at the intersection of two "AI-complete" tasks - computer vision and natural language processing. However, it recently emerged as an exciting area of research in the deep learning community, as neural networks have made tremendous strides in image and text-based tasks, specifically due to the rise of recurrent neural networks (RNNs) and convolution neural networks (CNNs).

The heart of VQA's technical challenge lies in comprehension of information from two different modalities: text and image. Research efforts have been channeled into the development of datasets that are effective in testing this aspect and models that are effective in combining visual and language information. For example, the recent release of the VQA v2.0 balances the dataset by doubling the number of image-question pairs. And, recent advances in performance have been driven by methods that use bi-linear pooling and co-attention to better capture the complex associations between the two different modalities.

Figure 1: An example of the VQA task from `visualqa.org`

The visual question answering task means providing an accurate natural language answer to a natural language question about an image. While the VQA challenge includes two other varieties: multiple choice questions and visual grounding, this project will focus on open-ended question answering, mimicking real-world situations [1]. An example of the visual questioning and answer task is shown in fig. 1.

In our project, we explore using CNN and RNN architectures involving attention. The remainder of the report will discuss briefly the current literature, describe the implementation of two architectures using bi-linear pooling and co-attention, and finally analyze the performance and results of these models.

2. Related Work

To inform our approach to visual question answering, we did a survey of architectures that have been proposed. Here, we present a summary of two of the best performing approaches to tackle the challenge of incorporating the bi-modal nature of VQA. The first approach is the use of

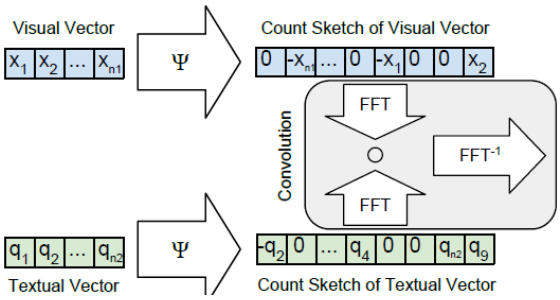


Figure 2: Multi-modal Compact Bi-linear Pooling (MCB) [3]

bilinear pooling [6, 3], and the second approach is the use of memory and attention [11, 7].

Bi-linear Pooling While bilinear pooling has been proposed as a technique for joining the representation of the two modalities that is more expressive than current approaches that rely on concatenating vectors or applying element-wise sum or product, it has been recently demonstrated to be computationally efficient and out-perform state-of-art. Fukui et. al [3] proposed Multimodal Compact Bilinear Pooling (MCB) to compress bilinear pooling that computes the outer product between the text and image vector representations for a single modality. MCB approximates bilinear pooling by randomly projecting the image and text representation to a higher dimensional space and then convolving both vectors efficiently by using element-wise product in Fast Fourier Transform space. What's exciting to us about MCB is that any attention-based model potentially stand to benefit by incorporating MCB into its architecture. An illustration of the MCB algorithm from the Fukui paper [3] is shown in fig. 2. Their model, which is the state-of-the-art, was able to achieve 66.9% accuracy in the open-ended 2016 VQA challenge.

Memory and Attention Memory and attention-based models relies on the hypothesis that in order to answer a question correctly the model has to understand where in the image to "look" or which words of the questions to "listen". A number of the attention-based models have been proposed that have perform close to state-of-art, including the Dynamic Memory Networks (DMN) model proposed by Xiong et. al. and Hierachical Co-attention (HiCoAtt) model proposed by Lu et. al. DMN proposes an episodic memory module that allows the models to focus attention on a subset of facts from the image input module. HiCoAtt model proposed an hierarchical architecture that co-attends to the image and question at the word, phrase, and question level.

3. Method

Inspired by the works in machine translation including Cho et. al [2] and Sutskever et. al. [10] that used a sequence-to-sequence encoder and decoder architecture to great success, we also aim to tackle the visual question and answering task using an encoding and decoding framework, as detailed in the following sections.

3.1. Encoding Architectures

We experimented with two different state-of-art architectures for encoding the multi-modal relationships between visual and textual information, including Multi-Modal Compact Bi-linear Pooling (MCB) proposed by Fukui et. al. [3] and Dynamic Memory Network proposed by Xiong et. al. [11].

3.1.1 Multi-Modal Compact Bi-linear Pooling (MCB)

This model extracts representations of the image and the questions, pools the vectors using MCB. The overall model architecture is shown in the figure 3.

We extracted image features with dimensions $14 \times 14 \times 512$ using the last layer of the VGG-16 network pre-trained on the ImageNet dataset [9].

Input questions are tokenized into words, and the word vector representation is obtained using the Glove6B embeddings pre-trained on Wikipedia 2014 and Gigaword 5 dataset [8]. The embeddings are passed through a tanh layer and then feed into a GRU to obtain a 512-D vector representation of the question. The question vector is tiled into dimensions $14 \times 14 \times 512$.

The image features and tiled question vectors are then passed into a MCB layer to obtain a $14 \times 14 \times 1024$ tensor which is used to compute attention values. Soft-attention has shown to be an effective mechanism to incorporate salient features of the visual representation into image-captioning models [12], and soft-attention is easily integrated into our architecture using MCB.

The $14 \times 14 \times 1024$ tensor after MCB pooling is passed through a convolutional layer with relu activation to obtain an intermediate $14 \times 14 \times 512$ tensor which is then passed through another convolutional layer to produce 14×14 scores. A softmax layer is then used to output the 14×14 attention weights for different regions of the visual representation. A weighted sum of the image feature vectors across the 14×14 regions is taken using the attention map to produce an attended visual representation that is then fused with the textual representation on a image and sentence level using another MCB pooling layer to produce a 1024-D vector which is the combined encoding of our image and question. Predicting attention map using MCB is demonstrated to be very effective in capturing saliency of interactions between visual and textual information as shown in

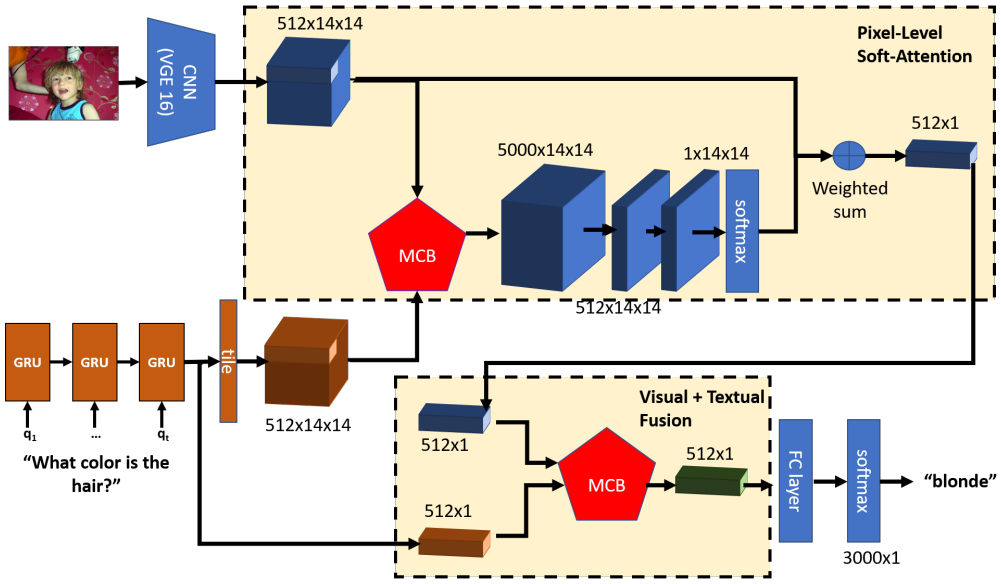


Figure 3: Multi-modal Compact Bi-linear Pooling Architecture (MCB)

the paper by Fukui et al. [3].

3.1.2 Dynamic Memory Network (DMN)

The dynamic memory network was introduced by Xiong et. al [11] as a general architecture consisting of a memory and input module for visual question and answering. The overall architecture is shown in figure 4.

The input module as shown in fig. 4 extracts image features with dimensions $14 \times 14 \times 512$ in the same way as described in the MCB section. These features are reshaped into a sequence of 196 vectors for each of the image regions. A fully connected layer with tanh activation is used to obtain sequences inputs which are passed into a bidirectional GRU to add global information. The outputs of the forward and backward GRUs are summed to obtain a sequence of facts, $F = [f_1, \ldots, f_n]$, which contain global information pertaining to different image regions. A vector representaion of our question, q, is obtained by passing the question embedding sequence into a GRU.

The episodic memory module as shown in fig. 4 learns to focus attention on a subset of the global-aware input "facts" $F = [f_1, \ldots, f_n]$ by associating a scalar value the attention gate g_i^t with each fact f_i during pass t, as shown in the formula below from Xiong et. al. [11]:

$$z_i^t = [f_i \cdot q; f_i \cdot m^{t-1}; |f_i - q|; |f_i - m^{t-1}]$$
$$Z_i^t = W^{(2)} \tanh(W^{(1)} z_i^t + b^{(1)}) + b(2)$$
$$g_i^t = \frac{exp(Z_i^t)}{\sum_{k=1}^{M_i} \exp(Z_k^t)}$$

where ; represents vector concatenation, $|\cdot|$ represents element-wise absolute value, and $\cdot$ represents element-wise multiplication.

Attention Mechanism To incorporate this attention, we experimented with both soft attention and attention based GRU. For soft attention, the context vector for pass t over the image facts is calculated as $c^t = \sum_{i=1}^{N} g_i^t f_i$. The attention based GRU method obtains context vector by passing the sequence F into a modified GRU unit where the update gate is replaced with the attention. That is, $h_i = AttnGRU(x_i, h_{i-1})$ is updated by:

$$r_i = \sigma(W^{(r)} x_i + U^{(r)} h_{i-1} + b^{(r)})$$
$$\tilde{h}_i = \tanh(W x_i + r_i \circ U h_{i-1} + b^{(h)})$$
$$h_i = g_i^t \circ \tilde{h}_i + (1 - g_i^t) \circ h_{i-1}$$

The memory of our system is initialized with q. That is, $m^0 = q$. The memory is updated as follows:

$$m^t = ReLU(W^t [m^{t-1}; c^t; q] + b)$$

The final memory vector(m^2) is taken as the combined encoding of the image and question.

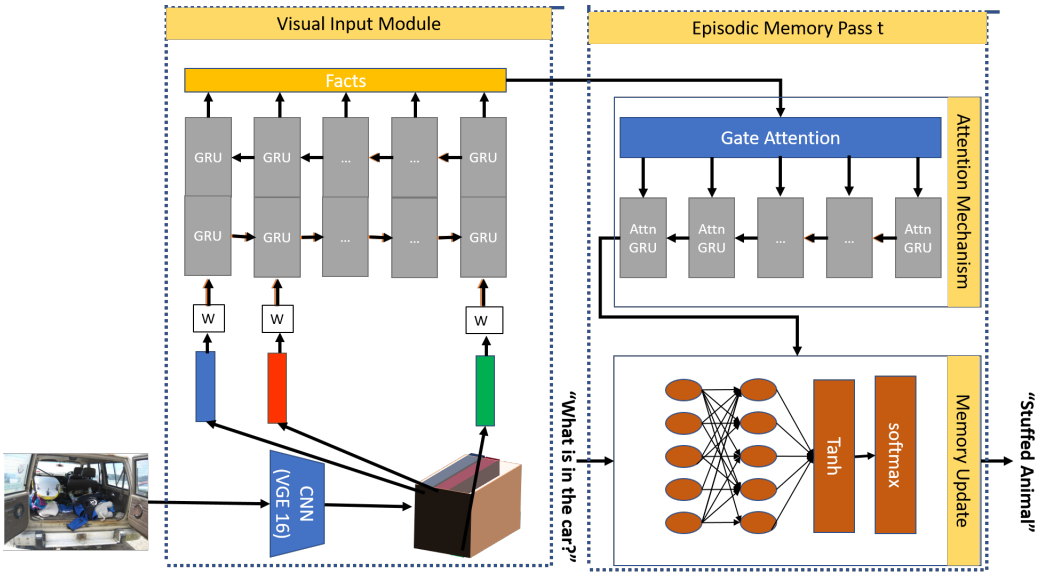


Figure 4: Dynamic Memory Network Architecture (DMN)

3.2. Decoding Architecture

The models we implemented are designed to predict an answer from a set of possible answers. In order to get the candidate answers, we counted the number of times each answer occurred in our dataset and limited our candidates to the answers that occurred at least 99 times. Therefore, to decode, we take in the combined vector representation of an image and its corresponding question to compute class scores using a fully-connect layer and softmax function. Finally, the cross entropy loss with logits was used for training to perform a 3003 way classification over our candidate answers.

4. Dataset

Researchers have proposed several rich datasets, the most notable of which is `visualqa.org` including more than 200,000 COCO images, 600,000 questions, and 6,000,000 answers. The latest release of the dataset in April 2017 v2.0 balances the dataset to counteract against models exploiting the inherent structure in language rather than learning visual modalities. For example, v2.0 collected approximately 195K complementary images for trains [4]. In total, v2.0 doubled the number of image-question pairs, so that every question is associated with not just a single image but rather a pair of similar images that results in two different answers to the same question [13].

4.1. Data Preprocessing

A summary of the specific COCO dataset we are using is shown in Table 1

	Training	Validation	Testing
Number of images	82,783	40,504	81,434
Number of questions	443,757	214,354	447,793
Number of answers	4,437,570	2,143,540	-

Table 1: Dataset summary

The testing answers have not been made public. Therefore, we split the validation set into validation and testing tests. There are about 5.4 questions per image, 10 ground truth answers per question. Question types include binary yes/no answers, numerical ones, and open-ended ones.

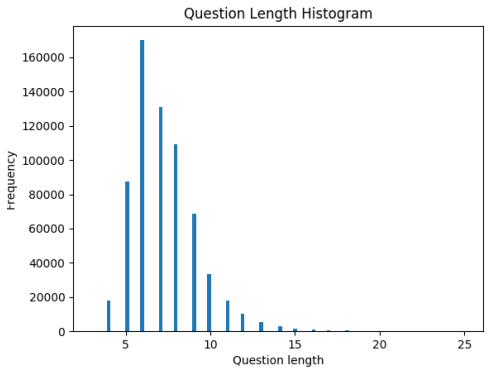
The maximum question token length is 25 and the maximum answer token length is 30. Figures 5a and 5b show the distributions of question and answer token lengths.

We preprocessed the images of size $m \times n$ by either centrally cropping the image or padding it evenly with zeros to size $(\min(m,n), \min(m,n))$. The resulting images were then resized to 224 by 224 pixels using bilinear interpolation. Additionally, we normalized the images by subtracting the VGG16 mean from all images.

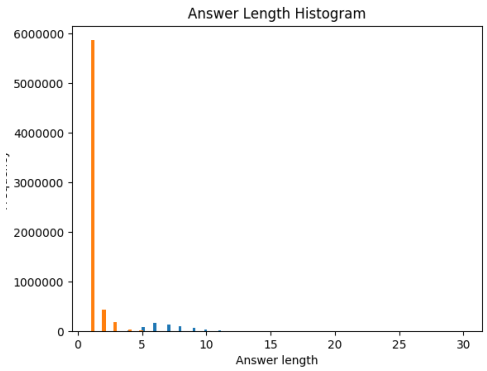
5. Experiments and Results

5.1. Experiment Setup

The training of each model presents its own set of challenges, but hyper-parameter searching of both architecture-specific and optimization-related hyper-parameters were the most important factors during our training process.



(a) Histogram of question lengths



(b) Histogram of answer lengths

Figure 5: Histogram of question and answer length

Architecture-specific Hyper-Parameters In addition to the optimization hyper-parameters such as learning rate, both the MCB and DMN have hyper-parameters associated with the architecture. For example, the output dimensionality of the MCB feature determines how well it approximates the bilinear feature. Based on the paper from Fukui et. al. [3], while a 16,000-D vector yields the highest accuracy, a smaller vector such as a 1,024-D vector that performs only 1.5% worse is used to boost training time during experiments. Additionally, the performance of the DMN model depends on the size of the input global facts, the size of the memory vector and the number of passes in the episodic memory module. Because the DMN paper [11] does not provide guidance on these parameters and our limited computing resources and time, we decided to use a 200-D vector for input "facts", 200-D vector for memory module, and two passes of episodic memory module to increase training speed.

Optimization Hyper-Parameters We experimented with different optimization solvers, batch size, dropout probability, learning rate, learning rate decay rate, and gradient clipping. We found the most success using the Adam optimizer, which was demonstrated to be an efficient and adaptive learning algorithm by Kingma [5]. Due to time-constraint, we were unable to conduct a thorough search of the remaining hyper-parameters. Instead relying on best-practices, we used a batch size of 64 samples, an initial learning rate about $1e-3$ to $5e-4$, and a learning rate decay rate of 0.99 to balance training speed with training accuracy. Finally, gradient was clipped at 10 to prevent exploding gradient problem for RNN-intensive architectures such as DMN.

5.2. Evaluation Metrics

We used the official evaluation benchmark as follows:

$$\text{Accuracy} = \min\{\frac{\text{\# of humans providing the answer}}{3}, 1\}$$

In other words, an answer is considered 100% accurate only if at least 3 people provide that exact answer. In order to be consistent with human accuracies, machine accuracies are averaged over all 10 choose 9 sets of human annotators, to be robust to variability in the phrasing of human answers.

5.3. Analysis and Evaluation

To get a sense of the difficulty of the VQA challenge and a better handle on the dataset, we implemented two baseline models, as described in the following.

1. **Prior Baseline**: We implemented a baseline which always predicts the most common answer in the training data. The test accuracy is about 24.7%, which is in-line with what is discussed in the literature.

2. **Language-based Baseline**: We also developed a language-only baseline that only uses the question but not the image as input to predict the answer. The model uses a Glove6B pre-trained embedding that encodes each question word which is then fed into a GRU. The final output hidden state representation of the question is fed through a fully-connected layer and classified using a softmax loss function. The baseline test accuracy is around 25.9%. We did not spend much time optimizing this model.

Comparison to State-of-the-Art Table 2 shows the results of the state-of-the-art on the VQA v2.0 dataset along with our single best models of the MCB and DMN architecture. The baseline performances are also included for illustration.

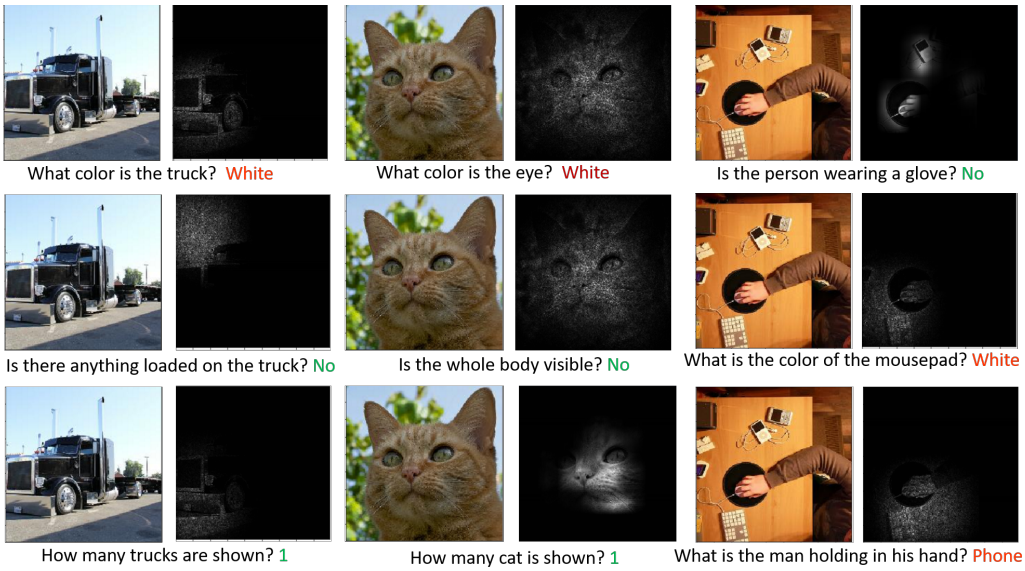


Figure 6: Examples of qualitative results of attention for VQA using the MCB model. The original images are shown on the left, and the attention map or saliency map is shown on the right. The white regions are the most active areas. The predicted answer of a question asked is shown below every image, and the answer is colored green if right and red if wrong.

Models/Question Types	Y/N	Num	Other	All
Prior Baseline (All "No")	65.1%	0.003%	0.006%	24.7%
Language Only Model	-	-	-	25.9%
MCB + GloveB + Attention	50.7%	20.7%	16.8%	32.2%
DMN$^+$	53.9%	25.1%	14.1%	33.1%
2016 VQA Winner Benchmark	78.8%	38.2%	53.4%	62.3%

Table 2: Open-ended Question results on the VQA dataset compared with baseline and state-of-the-art

As results table shows, DMN$^+$ model performs overall comparably to the MCB model with attention, but both models perform much better than the baselines. While MCB performs slightly better on the "other" question types, DMN$^+$ performs slightly better in yes/no and number types of questions. We hypothesize that MCB out-performs DMN$^+$ at "other" questions because as the original authors demonstrated answering "other" questions requires more understanding of the salient interactions between visual and textual information than yes/no and number questions. On the other hand, relying more on the inherent structure in language, DMN$^+$ uses the memory and attention architecture to out-perform the MCB model in the "yes/no" and "number" questions. Finally, the performance of the two models is only about half of the state-of-the-art accuracy which is the MCB implementation from Fukui et. al. in 2016 [3], because we need more time to fully train the models.

5.4. Attention Map and Answer Visualization

We visualized the attention maps and the predicted answers to qualitatively understand how the MCB model predicts answers and how it places attention on an image depending on the question asked. The results are shown in fig 6.

As shown, the model is able to place attention on different parts of an image depending on the question. This is encouraging, as it strongly suggests that model is capturing the interaction between visual and textual information. For example, the model places attention on the front of the truck when asked "What color is the truck?" and places attention on the back of the truck when asked "Is there anything loaded on the truck?". Furthermore, When the model predicts the wrong answer, the attention is focused on the wrong part of the image. For example, when the question is "What is the man holding in his hand?". The attention is more focused on the keyboard. This suggests the improving attention placing is key to improving the overall performance of the model, and given more time to train this model, the attention map would improve significantly.

In addition to capturing the interaction between visual and textual information via attention, the model also seems to capture the semantic information embedded in a question as the model seem to be able to predict the right type of answer depending on the type of question asked. For example, the model predicts a number when the question is "how many ...?" and yes or no when the question is "Is there ..".

However, at this incomplete stage of training, it's hard to judge qualitatively whether the model captured more information from the inherent structure in the language or the interaction between image and question.

6. Conclusion

In this project, we implemented and trained two different state-of-the-art architectures for visual question and answering. While the models were not able to reach state-of-the-art accuracy due to time-constraints during training, they already significantly out-performed the baselines and demonstrated uncanny comprehension of visual and textual information, as shown in our result visualizations. The code to replicate our experiment and implementation is available at `https://github.com/jejekunmi/knf_vqa/tree/master`.

This projects shows there is still plenty of rooms for improvement for developing novel architecture to better capture the ineraction between query phrasing representation and visual representations. For example, we can combine MCB with DMN by substituting the attention mechanism of the DMN architecture with a MCB layer. Additonally, we may also explore models that generate answers using a decoding layer for example with an additional RNN layer similar to what is done in machine translation or image captioning applications.

References

[1] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. L. Zitnick, and D. Parikh. VQA: visual question answering. *CoRR*, abs/1505.00468, 2015.

[2] K. Cho, B. van Merrienboer, Ç. Gülçehre, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *CoRR*, abs/1406.1078, 2014.

[3] A. Fukui, D. H. Park, D. Yang, A. Rohrbach, T. Darrell, and M. Rohrbach. Multimodal compact bilinear pooling for visual question answering and visual grounding. *CoRR*, abs/1606.01847, 2016.

[4] Y. Goyal, T. Khot, D. Summers-Stay, D. Batra, and D. Parikh. Making the V in VQA matter: Elevating the role of image understanding in visual question answering. *CoRR*, abs/1612.00837, 2016.

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

[6] T. Lin, A. Roy Chowdhury, and S. Maji. Bilinear CNN models for fine-grained visual recognition. *CoRR*, abs/1504.07889, 2015.

[7] J. Lu, J. Yang, D. Batra, and D. Parikh. Hierarchical question-image co-attention for visual question answering. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 289–297. Curran Associates, Inc., 2016.

[8] J. Pennington, R. Socher, and C. D. Manning. Glove: Global vectors for word representation. In *EMNLP*, volume 14, pages 1532–1543, 2014.

[9] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

[10] I. Sutskever, O. Vinyals, and Q. V. Le. Sequence to sequence learning with neural networks. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 3104–3112. Curran Associates, Inc., 2014.

[11] C. Xiong, S. Merity, and R. Socher. Dynamic memory networks for visual and textual question answering. *CoRR*, abs/1603.01417, 2016.

[12] 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.

[13] P. Zhang, Y. Goyal, D. Summers-Stay, D. Batra, and D. Parikh. Yin and yang: Balancing and answering binary visual questions. *CoRR*, abs/1511.05099, 2015.

7. Code References

- `https://github.com/ronghanghu/tensorflow_compact_bilinear_pooling`
- `https://github.com/shmsw25/mcb-model-for-vqa`
- `https://github.com/therne/dmn-tensorflow`