Visual Question answering: Implementation of Robust Baseline and Cutting-Edge Models

Huamin Zhang*
University of Illinois at Urbana-Champaign
Urbana, IL
huaminz2@illinois.edu

Yunan Zhang*
University of Illinois at Urbana-Champaign
Urbana, IL
yunanz2i@illinois.edu

Svetlana Lazebnik
University of Illinois at Urbana-Champaign
Urbana, IL
slazebni@illinois.edu

Abstract

We investigate the problem of visual question answering(VQA) via implementing two cutting edge methods as well as explore some of the related models to see if there's possible improvement opportunities. We mainly implement Question-Image Co-Attention model (HieCoAtt)[34] *and Deeper LSTM Question + norm Image (d-LSTM+nI)*[31] *in PyTorch, following Peter Anderson's designing philosophy*[30]*. Models are trained on MSCOCO full datasets on Blue Water. We evaluate the model performances on the challenging datasets, though it does not achieve better performance compared to their original version, our models still get comparable scores. We make our implementation open source on Github for others to share with.*

** refers to equal contribution.*

1. Introduction

With the progress in both natural language and computer vision, multimodal learning becomes recent trend, combing natural language processing and computer vision for higher level understanding of scene interpretation. Early effort in this trend is iamge caption task, which requires model to generate text descriptions of the given image that covers the objects in the image and the relationship between them. In our project, we try to do a task that steps on step further, namely Visual Question Answering. The Visual Question Answering problem is defined as given an image and a natural language question about the image, the task is to provide an accurate natural language answer. Visual questions selectively target different areas of an image, including background details and underlying context. Visual question answering not only need model to generate descriptive texts, but also requires such texts answer the given question, which needs higher level inception than image caption task. More formally, this task involves feeding an image and a text question about the image and returning a correct natural language answer. Compared to image caption task, VQA is a lot easier to evaluate, due to the fact that there's no unique description for an image, which somehow make generic descriptions more or less hold for a large collection of datasets. However, for VQA, it's a lot easier to come up with a narrow question for an image to force the model choose a specific answer. This feature convinces us that VQA is a better evaluation of model's real perception of computer vision and natural language.

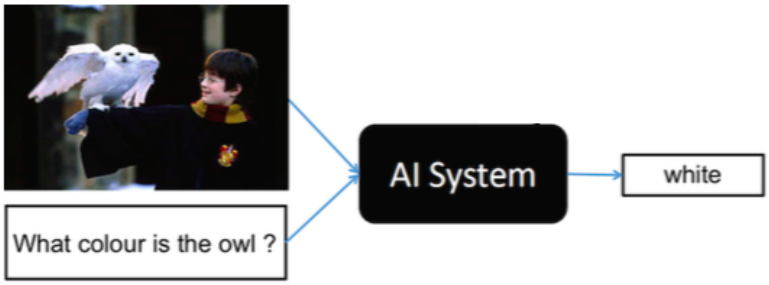

Figure 1. Illustrative example of VQA

For humans, it might be easy to answer questions like 'how many players are there on the stage?','what's the brand of the wine?' or 'Is the candle lightening or

not?', etc. However, for machines, such task is far from trivial. Though both computer vision and natural language processing have achieve significant progress during hte past years, they're mostly developed separately. But VQA requires excellence in multi-discipline by nature. Relevant models needs to decide what is the information wanted based text modality and then retrieve from the image, and used that to answer the question. Also, the question itself has various complexity. For example, it should be easy to answer question like 'is there a tiger in the image' since advances in object detection makes this simple. But the question 'What is the color of the tiger' and 'What is the color of the tiger standing behind the bigger tiger' is really confusing for machines since they involve some reasoning and knowledge out of the image. may requires the model have some abstract reasoning skills.

Figure 2. Sample results from previous methods[33]

Our project implements two cutting edge model proposed in 2017, which employs LSTM for question encoding and deep CNN for image feature encoding. Fusion of LSTM and CNN has been a mainstream solution to VQA task, while the model we select to implement has some advantages.

The first model trains a deeper LSTM and CNN to do the task. To be exact, the model is a fusion of two-channel vision(image) and language model, culminating with a softmax over K possible ouputs. This model set the general philosophy for this task, which is followed by nearly all new models proposed for this task. Most of new papers still tries to encode image via some extensions of CNN and encode questions via extensions of RNN and project them into the same space.

The second model tries to incorporate attention mechanism into their model to enhance model performance. Different from previous attempts in applying attention mechanism to VQA, which mainly focus on visual attention, namely where to look for, this model fuse word level attention, namely decide which word in the question is more important. In this way, it gains a natural symmetry on both sub-task. For example, given the same image of a classroom filled with students, the question 'how many students are there in the classroom' should have the same answer as the question 'how many students can you see in the picture'. In this example, the only thing matters is the first three words since they decide what entity we are looking for, namely student, and what information about this entity are we interested in, namely their number. With the help of attention mechanism, the model will focus on the words matters instead of meaningless linguistic variation that irrelevant to the essence of the question.

Specifically, the second model we implement has two novel features: one is co-attention, which jointly model visual and question attention. Then other is attention hierarchy that guide co-attention at word level, phrase level

and sentence level. Namely we do embedding for single word, bigram or trigram for phrase, or RNN encoding for question. Then we use these embedding to guided co-attention.

Besides these two models, we also try to improve performance of the original model via incorporate some new cutting-edge structures like Resnet and Relation Net into the image with the intention to improve model performance.

To conclude, our contributions can be summarized into two parts:

1. Reimplement two influential models that is popular in the VQA task. Train them on large datasets and get comparable performance to their original models.

2. Incorporate new structures into the original models and does ablation study for potential improvement.

2. Related Work

In this part, we will briefly go through progress made in VQA during the past few years and introduce some basic background knowledge.

Vinyals *et al.*[2] adopts a similar model structure to the models we implement, but it limits its dataset to a size that contains only 16 basic colors. Geman *et al.* [3] limits its answer generator to certain word choices via a answer template. In Malinowski's word [4], question is embedded by LSTM, which is conditioned on the feature extracted from image via CNN. The final answer is decoded by the last hidden state of the LSTM. Lin and Parikh [5] deals with fill-in-blank questions via generating abstract scenes to capture visual common sense. Sadeghi [6] and Vendantam [7] also takes a similar strategy, and they both do blank assertions with the help of visual information. Santoro's work [8] is another direction of VQA which focus more on reasoning relationship between different objects. Santoro [8] takes advantage of MLP to perform pairwise comparison over each location of extracted convolutional features over an image. The MLP takes not only visual features, but also LSTM embedding of the questions as input. Then RNs will sum over the output of MLP, and put the sumed vector into a classifier to get the final answer. However, pairwise comparison somehow limits its efficiency, and make it suffer from strong prior assumption. Perez *et al.* [9]alleviate this problem via selectively learn conditioned network features. Other related work includes image captioning and image tagging $[12-14]$.

Attention Mechanism: in this part, we'll briefly introduce attention mechanism used in our implemented model.Basically, attention mechanism assign weights to important pixels or word instead of treating them equally to their counterpart. The assigned weights help the classifier better make decision. Sometimes, it can also help improve efficiency via avoiding unnecessary calculations. We'll mention attention mechanism's application in CV and NLP tasks in the following two paragraphs.

Image attention: Zhu *et al.* [18] apply spatial attention module to LSTM to point and ground QA. Andreas [19] combine NMN(neural module networks) and language parser. The language parser is responsible for predicting which NN is to answer the questions. Yang *et al.* [20] follows similar philosophy, but apply attention in a stacked way. Xiong *et al.* [22] fetches an answer via querying an attention based gated recurent unit.

Language attention: In NLP setting, attention mechanism is usually applied to machine translation task, where long sequence become obstacles for improving translation quality. Attention mechanism helps alleviate this via learn an alignment vector over the input sentences[23]. Yin *et al.* [24] apply attention to two CNN to capture bigrams. In Rocktschel's work [25], attention is used to reason about the entailment in two sentences.

3. Methods

In this section, we'll briefly introduce the model we implement, including their structures and mechanism.

The main model (deeper LSTM + CNN) we implement can be divided into three components: the Image Channel, the Question Channel, the combination of the output of the two channels. The image channel embeds the image, the question channel embeds the question while the MLP or MCB combines the two channel and make the final prediction. While training the model, the loss function we use is Cross Entropy Loss, the optimizer we use is Adam optimizer with learning rate decay method.

3.1. Image Channel:

We implement two image embedding methods.
1)VGGNet[26]: VGG model secured the first and the second places in the localisation and classification tracks respectively in ImageNet Challenge 2014. The main contribution of VGG network is 'a thorough evaluation of networks of increasing depth using an architecture with very small (33) convolution filters, which shows that a significant improvement on the prior-art configurations can be achieved by pushing the depth to 1619 weight

layers'[26]. This work shows the importance of depth network in visual representations. Here we implement VGGNet16 to extract iamge features, which is also similar with previous work.

ResNet[35]: Here we implement a more cutting-edge

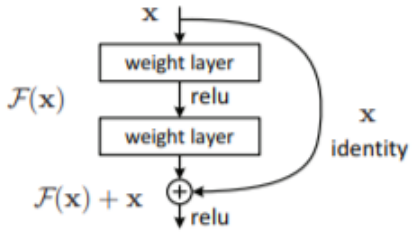

Figure 3. Residual learning module: $y = F(x, \{W_i\}) + x$ [35]

model, Resnet152, which the original model doesn't uses. ResNet is 'a residual learning framework to ease the training of networks that are substantially deeper than those used previously. 'It explicitly reformulate the layers as learning residual functions with reference to the layer inputs, instead of learning unreferenced functions.'[35] ResNet won the 1st place on the ILSVRC 2015 classification task and shows that the residual networks are easier to optimize, and can gain accuracy from considerably increased depth. Compared to VGGNet16, Resent is available for deep structures while guarantee the training efficiency through their residual learning framework.

```
DeeperLSTM(
  (image_embed): Sequential(
    (0): Linear(in_features=4096, out_features=1024, bias=True)
    (1): Tanh()
  )
  (text_embedding): Sequential(
    (0): Linear(in_features=10581, out_features=300, bias=True)
    (1): Dropout(p=0.5)
    (2): Tanh()
  )
  (rnn): LSTM(300, 512, num_layers=2, batch_first=True)
  (fc): Linear(in_features=2048, out_features=1024, bias=True)
  (tanh): Tanh()
  (mlp): Sequential(
    (0): Dropout(p=0.5)
    (1): Linear(in_features=1024, out_features=1000, bias=True)
    (2): Tanh()
    (3): Dropout(p=0.5)
    (4): Linear(in_features=1000, out_features=1000, bias=True)
  )
)
```

Figure 5. Architecture of Deeper LSTM (Image preprocessed)

deeper LSTM Q model we use here has two hidden layers, which is used to obtain 2048 dimension embedding for the question. The final embedding is a concatenation of last cell state and last hidden state representations, where both are a 512 dimension vector. The output will be fed into a fully connected layer which is followed by a tanh non-linearity to transform 2048 dimension to 1024 dimension. In this way, the question are embedded as a 1024 dimension vector.

Moreover, we also try a 'deeper' deeper LSTM Q model to deal with question embedding. In previous works, people always using LSTM model with two hidden layers to do question embedding. Here we use a deeper LSTM model with three hidden layers and try to figure out whether a deeper LSTM network will be better.

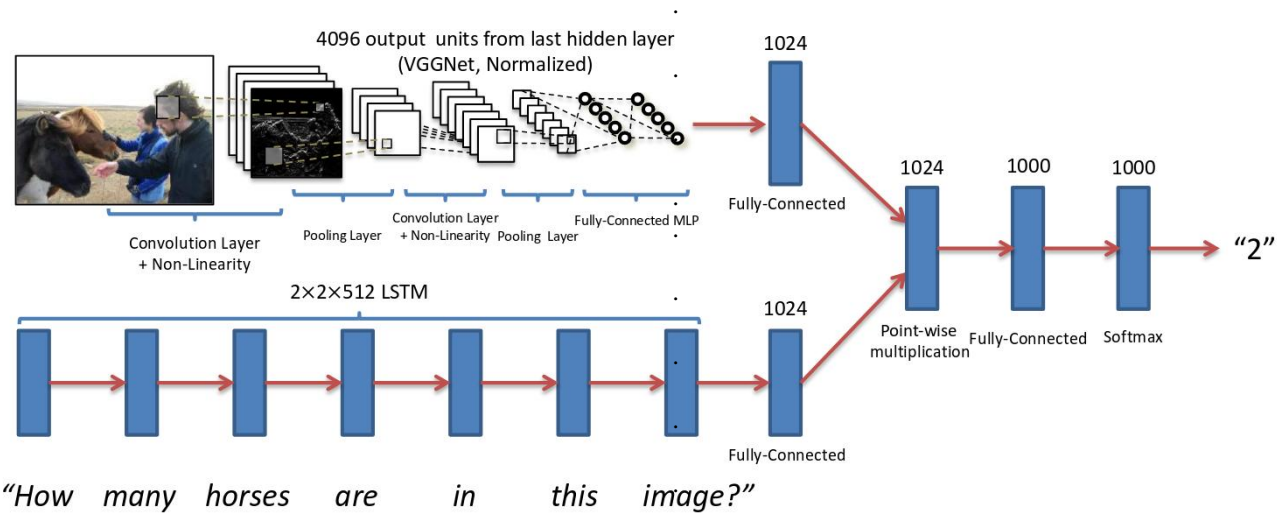

Figure 4. Basic Architecture: MLP(CNN + Deeper LSTM) [35]

3.2. Question Channel:

For the question embedding part, we implements the deeper LSTM Q mentioned in the original paper[35]. The

3.3. Joining image channel and question channel

In this part, the image and question embeddings are combined to obtain a single embedding. This is also the key problem VQA needs to be solve.In this paper, we

implement two different fusion methods in this part. One is the method proposed in [28], namely MLP to do the concatenation. The other is MCB[27].

MLP (MultiLayer Perceptron):
An MLP is a simple feedforward neural net that maps a feature vector (of fixed length) to an appropriate output. In our problem, this output will be a probability distribution over the set of possible answers. First we need to transform the image embedding to fit it to the question embedding. The output of the image channel is fed to a fully connected layer, which transform image embedding to 1024 dimension. This outcome is then pass to a tanh non linearity to match the LSTM embedding of the question. In this way, image embedding and question embedding are available for further manipulation. In previous work, Lu *et al.* [28] does so via element-wise multiplication, which fuse two signals together.

MCB (Multimodal Compact Bilinear Pooling)[27]:
Multimodal Compact Bilinear Pooling is firstly introduced in Tenenbaum's work [29], which allows all elements of both vectors to interact with each other in a multiplicative way. To be exact, the model take the outer product of two vectors and learn a model about the product. However, in the original setting of [30], the output may have a too high dimension, making the number of parameters infeasible for training. In VQA setting, the output can have billions of parameters for training, rendering high memory consumption and training time. So [27] is applied to project the output(element-wise product) to a lower dimensional space (Fast Fourier Transform (FFT) space). These ideas are summarized in Figure 6.

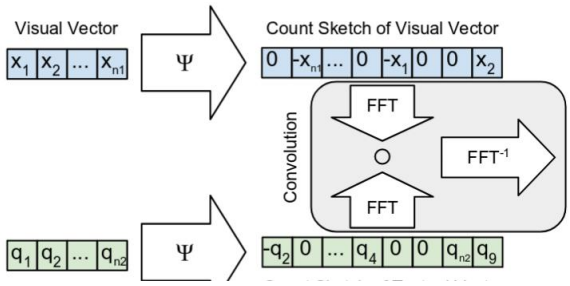

Figure 6. Multimodal Compact Bilinear Pooling (MCB)[27]

3.4. Co-Attention

Co-attention[34] is used to capture the important part both in images and questions. Two modals are connected via performing pairwise similarity calculation and each pair of image locations and question locations. Mathematically, this is done via $C = tanh(Q^T W_b V)$, where the $W_b$contains weights. With this weigh matrix, we can then learn to predict image and question attention maps via $H^v = tanh(W_V + (W_q Q)C)$. The image attention will be $a^v = softmax(w_{hv}^T H^v)$. The question attention map follows a similar form. Finally, with $a^v and a^q$, we can calculate the image attention vector and the question attention vector via $\hat{v} = \sum_{n=1}^{N} a_n^v V_n, \hat{q} = \sum_{t=1}^{T} a_t^q q_t$.

3.5. Baselines

To put the accuracy of these models in perspective, we compare our implemented models to the following baselines. Prior: This one predicts the most common anser in the training set, for all test questions. The most answer is 'yes' in both the unbalanced and balanced sets.

4. Experiments

4.1. Datasets

We use Balanced Real Image data in VQA v2 datasets(json files for text and raw image JPEG data), which includes three part. One is annotations, which consists of 4,437,570 answers in training annotations, and 2,143,540 answers in validation annotations. The second part is input questions, which consists of 443,757 training questions , 214,354 validation questions, 447,793 test questions. All questions are annotated with 10 concise, open-ended answers each. There are three types of question: 'Yes/No', 'Number', 'Other'. The third is MSCOCO dataset, which consists of 82,783 training images, 40,504 validation images, 81,434 test images. (http://www.visualqa.org/download.html)

4.2. Results and Analysis

Table 1 shows the results of our 4 implemented models on VQA v2 dataset. As we can see, the best model is $LSTM + Resnet + MCB$, outperforming the other three models we implement. First, this is probably due to we incorporate Resnet rather than VGG16 into the network architectures, which gets the same result as Lu's work[34] that "ResNet features outperform or match VGG features in all cases due to the use of a better CNN.'. Also, comparing the third and fourth model to the first two model, we also see that MCB can better fuse the image channel and question channel information than MLP. The result we think is MCB introduces an additional 2048-D convolution using FFT after the element-wise product and transform to a lower dimensional space before going to the fully connected layer.

Method	Y/N	Num	Other	All
Prior Yes	70.81	0.39	1.15	29.6
LSTM+VGG+MLP	74.90	32.43	36.24	51.7
LSTM+Resnet+MLP	75.94	33.39	37.24	52.7
LSTM+VGG+MCB	77.20	31.12	42.11	55.3
LSTM+Resnet+MCB	77.90	32.69	43.80	56.6

Table 1

Table 2 shows the performance of our experiment on 3 layers LSTM and 2 layers LSTM with VGGNet16 + MLP. It seems that two models have very similar performance despite their layers. In our experiment, 3 layer LSTM does not perform better than its 2 layer LSTM and sometimes even perform worse on some tasks. We think 2 layesr LSTM may already be able to capture sufficient information, but extra parameters added by another layer might make the model more complex and harder to converge, and also need more space and time resources.

Method	Y/N	Num	Other	All
Two Layers LSTM+VGG	74.90	32.43	36.24	51.7
Three Layers LSTM+VGG	75.11	33.20	35.72	51.6

Table 2

In Tables 3, we compare our best performance model with [34].implemented HieCoAttenVQA. We can see that co-attention brings about 4 percent improvement overall and outperform old models in all kinds of questions. This is consistent with the observation made in [34]. We assume that the question level might has some strong indication of the answer that the model should focus on. Question attention helps the model improve this aspect, while image attention might also help attach more significance to the object we might be interested in.

Method	Y/N	Num	Other	All
Our Best Model	77.90	32.69	43.80	56.6
HieCoAttenVQA	78.60	36.97	50.21	60.6

Table 3

4.3. Qualitative Results

5. Conclusion and Future Work

Conclusion Future Work Currently, we haven't reach the state of the art performance as the original model we implement achieves. We think it is because we may not run enough epoch due to our machine and time limitation, which leads to underfitting. For the VQA problem, we think there may be some factors that people can improve the performance. The first factor is that the image embedding

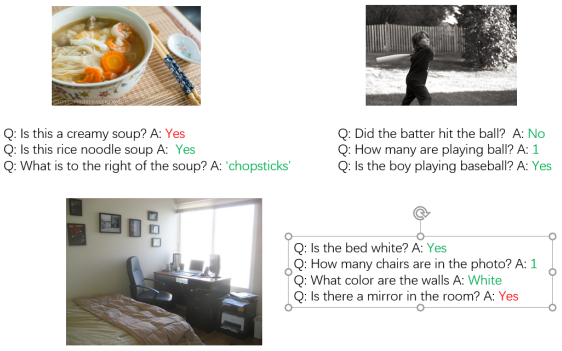

Figure 7. Sample Result From LSTM + ResNet + MCB

method. Now there are more new CNN architectures that perform better on image processing, like Mask R-CNN, GoogLeNet, ResNetInception-ResNet-v2, FractalNet or DenseNet. We can replace the image channel model with the new ones and to figure out whether it will have a better performance. The second factor is the model to fuse image and word embedding. We think the factor is the most important in VQA problem and people could find algorithms better than MLP and MCB which are the popular architectures. These two factors are the directions we believe that are most likely to break through the performance of existing models.

We implemented several models of VQA and conduct sufficient experiments on them. In our experiments, we find that for image feature extraction, ResNet outperforms VGG in most of the cases. While MCB plays a better role than MLP in fusing image information and question information. Lastly, without the help of co-attention, the model performance can be further improved.

6. Contributions

Huamin Zhang implements the original LSTM-CNN mdoels using pytorch and train them on his GPU. He also tune the model with variations of network structures including VGG, Resnet and MCB. Yunan Zhang conduct the survey for the project, incorporate attention mechanism into his model as well as preprocess the data for training. He also trains HieCoAtten model using the official implementation. They analysis experiment results and write the report together.

7. Reference

[1]M. Malinowski and M. Fritz. A Multi-World Approach to Question Answering about Real-World Scenes based on Uncertain Input. In NIPS, 2014.
[2]O. Vinyals, A. Toshev, S. Bengio, and D. Erhan. Show

and Tell: A Neural Image Caption Generator. In CVPR , 2015.
[3]D. Geman, S. Geman, N. Hallonquist, and L. Younes. A Visual Turing Test for Computer Vision Systems. In PNAS , 2014.
[4] M. Malinowski, M. Rohrbach, and M. Fritz. Ask your neurons: A neural-based approach to answering questions about images. In ICCV , 2015.
[5]X. Lin and D. Parikh. Dont just listen, use your imagination: Leveraging visual common sense for non-visual tasks. In CVPR , 2015.
[6]F. Sadeghi, S. K. Kumar Divvala, and A. Farhadi. Viske: Visual knowledge extraction and question answering by visual verification of relation phrases. In CVPR , 2015.
[7]R. Vendantam, X. Lin, T. Batra, C. L. Zitnick, and D. Parikh. Learning common sense through visual abstraction. In ICCV, 2015.
[8]Santoro, A.; Raposo, D.; Barrett, D. G.; Malinowski, M.; Pascanu, R.; Battaglia, P.; and Lillicrap, T. 2017. A simple neural network module for relational reasoning. CoRR abs/1706.01427.
[9]Perez, E.; de Vries, H.; Strub, F.; Dumoulin, V.; and Courville, A. C. 2017. Learning visual reasoning without strong priors. In MLSLP Workshop at ICML.
[10]E. Perez, F. Strub, H. de Vries, V. Dumoulin, and A. Courville. FiLM: Visual reasoning with a general conditioning layer. arXiv preprint arXiv:1709.07871, 2017.
[11]G. Kulkarni, V. Premraj, S. L. Sagnik Dhar and, Y. Choi, A. C. Berg, and T. L. Berg. Baby Talk: Understanding and Generating Simple Image Descriptions. In CVPR , 2011.
[12]A. Farhadi, M. Hejrati, A. Sadeghi, P. Young, C. Rashtchian, J. Hockenmaier, and D. Forsyth. Every Picture Tells a Story: Generating Sentences for Images. In ECCV , 2010. 2.
[13]M. Mitchell, J. Dodge, A. Goyal, K. Yamaguchi, K. Stratos, X. Han, A. Mensch, A. Berg, T. L. Berg, and H. Daume III. Midge: Generating Image Descriptions From Computer Vision Detections. In ACL, 2012.
[14] H. Fang, S. Gupta, F. N. Iandola, R. Srivastava, L. Deng, P. Dollar, J. Gao, X. He, M. Mitchell, J. C. Platt, C. L. Zitnick, and G. Zweig. From Captions to Visual Concepts and Back. In CVPR , 2015.
[15]Lin Ma, Zhengdong Lu, and Hang Li. Learning to answer questions from image using convolutional neural network. In AAAI, 2016.
[16]Mateusz Malinowski, Marcus Rohrbach, and Mario Fritz. Ask your neurons: A neural-based approach to answering questions about images. In ICCV , 2015.
[17]Mengye Ren, Ryan Kiros, and Richard Zemel. Exploring models and data for image question answering. In NIPS, 2015.
[18]uke Zhu, Oliver Groth, Michael Bernstein, and Li Fei-Fei. Visual7w: Grounded question answering in images. In CVPR, 2016.
[19]Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Dan Klein. Deep compositional question answering with neural module networks. In CVPR , 2016.
[20]Zichao Yang, Xiaodong He, Jianfeng Gao, Li Deng, and Alex Smola. Stacked attention networks for image question answering. In CVPR , 2016.$https://github.com/TingAnChien/san-vqa-tensorflow$
[21]Huijuan Xu and Kate Saenko. Ask, attend and answer: Exploring question-guided spatial attention for visual question answering. arXiv preprint arXiv:1511.05234, 2015.
[22]Caiming Xiong, Stephen Merity, and Richard Socher. Dynamic memory networks for visual and textual question answering. In ICML , 2016.
[23]Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In ICLR, 2015.
[24]Wenpeng Yin, Hinrich Schtze, Bing Xiang, and Bowen Zhou. Abcnn: Attention-based convolutional neural network for modeling sentence pairs. In ACL, 2016.
[25]Tim Rocktschel, Edward Grefenstette, Karl Moritz Hermann, Tom Ko cisky, and Phil Blunsom. Reasoing about entailment with neural attention. In ICLR , 2016.
[26]K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. CoRR , abs/1409.1556, 2014.
[27]Akira Fukui, Dong Huk Park, Daylen Yang, Anna Rohrbach, Trevor Darrell, Marcus Rohrbach. Multimodal Compact Bilinear Pooling for Visual Question Answering and Visual Grounding. EMNLP16.
[28]Jiasen Lu, Xiao Lin, Dhruv Batra, and Devi Parikh. Deeper LSTM and normalized CNN Visual Question Answering model.$https://github.com/VT-vision-lab/VQA_LSTM_CNN$
[29]oshua B Tenenbaum and William T Freeman. 2000. Separating style and content with bilinear models. Neural computation , 12(6):12471283.
[30]Anderson, P., He, X., Buehler, C., Teney, D., Johnson, M., Gould, S., Zhang, L. (2017). Bottom-up and top-down attention for image captioning and VQA. arXiv preprint arXiv:1707.07998.
[31]Goyal, Y., Khot, T., Summers-Stay, D., Batra, D., Parikh, D. (2017, July). Making the V in VQA matter: Elevating the role of image understanding in Visual Question Answering. In CVPR (Vol. 1, No. 6, p. 9). [32]Moses Charikar, Kevin Chen, and Martin Farach-Colton. 2002. Finding frequent items in data streams. In Automata, languages and program-ming , pages 693703. Springer.
[33]https://avisingh599.github.io/deeplearning/visual-qa/
[34]Lu, J., Yang, J., Batra, D., Parikh, D. (2016). Hierarchical question-image co-attention for visual question

answering. In Advances In Neural Information Processing Systems (pp. 289-297). [35]He, K., Zhang, X., Ren, S., Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778)