YOLO-ing the Visual Question Answering Baseline

Luca Lanzendörfer, Sandro Marcon, Lea Auf der Maur
Team Pendulum, Department of Computer Science, ETH Zürich

***Abstract*—We propose a model for Visual Question Answering (VQA) based on iBOWIMG. Our model uses image features with attention extracted from InceptionV3 as well as object features extracted from the VQA dataset using YOLO object detection. Our model is able to achieve a competitive score on the VQA v1 test-dev scoreboard. Furthermore, we analyse shortcomings of the dataset and explore the current state of VQA.**

I. Introduction

Visual Question Answering is a novel field with countless real-world applications. VQA is the task of having a machine study an image and a natural language question related to that image and subsequently answering it (see Fig. 1). This interdisciplinary field combines Computer Vision and Natural Language Processing, having its roots in image captioning [1] where a machine generates a natural language sentence describing a given image. These fields have received increased interest through breakthroughs from state-of-the-art models using various methods from Deep Learning.

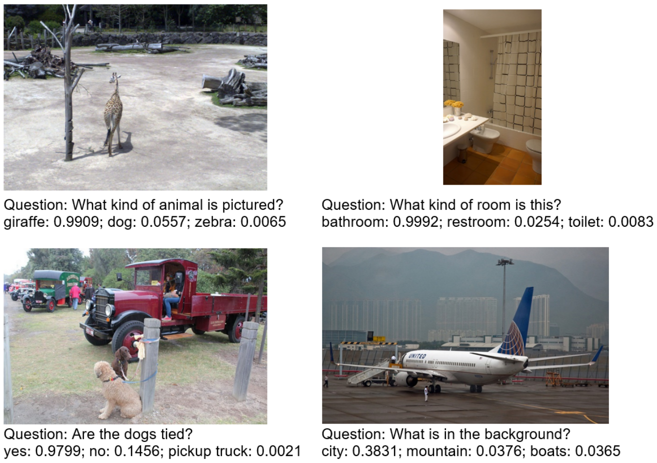
Much of this interest stems from the view that insights into image captioning and VQA will help move towards solving general AI.

However, unlike image captioning where the state-of-the-art algorithms are already able to generate plausible captions, Visual Question Answering still has difficulties obtaining human level performance even though the rate of recent advances looks to be fruitful.

The main focus of this paper was to analyse and beat iBOWIMG, a simple but competitive model suggested by Zhou *et al* [2] for Visual Question Answering on the VQA v1 dataset [3]. To this end we propose a model which utilizes image features extracted from InceptionV3 [4] and leverages object features from YOLO [5], a state-of-the-art object detector to obtain a competitive score on the VQA v1 test-dev scoreboard.

II. Related Work

There have been several attempts at Visual Question Answering before the introduction of the VQA dataset. These were however limited in scope [6] and were often of synthetic nature ([7], [8]). This changed with the introduction of the first VQA dataset containing roughly 250'000 images, 760'000 questions and 10 million answers.



Question: What kind of animal is pictured?
giraffe: 0.9909; dog: 0.0557; zebra: 0.0065

Question: What kind of room is this?
bathroom: 0.9992; restroom: 0.0254; toilet: 0.0083

Question: Are the dogs tied?
yes: 0.9799; no: 0.1456; pickup truck: 0.0021

Question: What is in the background?
city: 0.3831; mountain: 0.0376; boats: 0.0365

Figure 1: Images and questions from the VQA test2015 dataset. Every image is shown with a question and the three highest predicted classes for that question.

To extract our image features from this dataset we make extensive use of the InceptionV3 model, a deep convolutional neural network pretrained on ImageNet. It can classify images into 1000 different categories as well as extract the representations of the categories as image features. This was used to obtain a multi-dimensional feature representation of the image which we then used for attention.

In addition to image features our proposed model uses object features extracted from YOLO. YOLO is a real-time object detector which consists of a neural network that takes a single look at the image. It defines weighted regions of interest where the weights correspond to the confidence that an object lies in that region and then classifies each region into a category.

We will now briefly discuss some of the previous approaches to VQA. The majority of models proposed for VQA have been based on different types of neural networks. This can be attributed to state-of-the-art breakthroughs in computer vision [9] and natural language processing [10] which have been achieved by leveraging neural networks.

The neural network models proposed for VQA can be split into two categories: Attention based and non-attention based. For models without an attention mechanism, Ma *et al* [11] propose a model based solely on CNNs. They encode both the image as well as the question through two separate CNNs and then feed the combined output into a third CNN which uses a softmax output layer to predict the answer.

Noh *et al* propose DPPnet [12], a model based on VGGnet [13]. By removing the last softmax layer of VGGnet and adding more fully connected layers, where the third layer has its parameters tuned by a GRU network [14] they reason that a model needs a non-fixed amount of parameters to be strong enough for VQA.

iBOWIMG, the baseline model proposed by Zhou *et al* [2] uses image features extracted from GoogLeNet [15] and a simple bag-of-words model concatenated together and connected to a softmax output layer to predict the answers. This surprisingly simple model is able to achieve comparable results to the methods described above while being substantially less complex.

We will now briefly discuss some of the proposed attention-based models. One of the first attention-based VQA models was proposed by Shih *et al* [16] which uses VGGnet to encode the image. The questions are obtained through a simple averaging model, an attention vector is then computed to emphasise a region of the image based on the question. The question embedding and attention weighted image vector are then used to predict the answer.

Hierarchical Co-attention (CoAtt) proposed by Lu *et al* [17] models visual as well as question attention. Additionally, this model differs in that it encodes questions at different levels of abstraction and uses two different forms of attention. Alternating co-attention where the image and question are attended iteratively one after the other and parallel co-attention where the image and question are attended simultaneously.

The current state-of-the-art model and winner of the 2017 VQA Challenge proposed by Teney *et al* [18] uses GRUs on the question embedding and also uses attention to focus on important parts of the image. Among others they use *gated tanh* activation functions inspired by LSTMs and ensembling to obtain state-of-the-art performance.

III. Proposed Model

This section presents our proposed model (see Fig. 2) and will give an overview of the underlying methods. In summary, we use iBOWIMG as a foundation, on top of it, we use image features from InceptionV3 which are weighted according to the attention generated from the question and image. Additionally, incorporating object features from YOLO allows us to obtain a competitive score.

A. Question features

Every image in the VQA dataset contains three questions. We tokenize each question and threshold each word to a value of 6, i.e. whenever a word occurs less than six times in the dataset it is ignored. The same procedure is repeated for the answers which are needed for training, except that we threshold them at 3 instead of 6. These threshold values were taken from the iBOWIMG model and left unchanged. The answer encoding is slightly more involved and explained in more detail in Section IV-C.

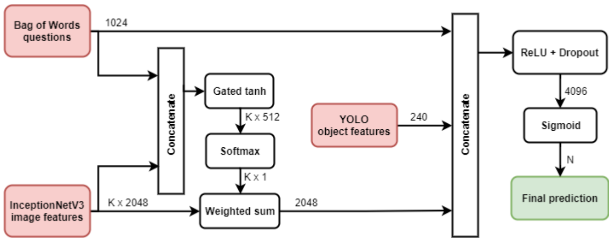


Figure 2: Overview of our proposed model.

Each question is then transformed into a one-hot encoding which is then passed through a fully-connected layer with 2048 units. This layer can be interpreted as a word embedding which embeds the questions into a 2048 dimensional space. These embedded questions are then used for attention and for the final prediction. To be able to generate an attention vector for every image we replicate the question embedding vector K times and concatenate it with the image feature matrix.

B. Image features

To extract the image features from the VQA dataset we use InceptionV3 which was pretrained on ImageNet. The features returned by InceptionV3 are of dimension $8 \times 8 \times 2048$ which are then downsampled to $4 \times 4 \times 2048$ through bilinear interpolation. Finally, we reshape the features to 16×2048, where $K = 16$ then represents the number of image locations on which we compute attention.

Since every image in the VQA dataset is related to three questions we replicate every image feature matrix three times. The image features concatenated together with the corresponding question form a $K \times (2048 + 1024)$ matrix which is then passed through the attention generator.

C. Object features

We extracted feature vectors from the YOLO object detector model [5], which is pretrained on the COCO dataset [19]. The library based on YOLO can identify 80 different objects in a given input image and outputs the position of the objects and the confidence of the detection. In order to give more informative features to the iBOWIMG model, the outputs of YOLO are encoded as vectors of size 80×1, where each column contains the number of detected objects of the given type. Three of these object vectors are produced for a detection confidence threshold of 25%, 50% and 75% and then concatenated with the image features and question features.

D. Classifier

The concatenated questions and image features are passed through a gated hyperbolic tangent activation function. This was suggested by Teney *et al* [18] as it achieves better results than the simpler ReLU or tanh activation functions when used to compute attention. Gated tanh functions are closely related to LSTMs and GRUs and have been used successfully in NLP [20]. The formulas for the gated tanh are given as follows:

$$\hat{y} = tanh(Wx + b)$$

$$\gamma = \sigma(Ux + c)$$

$$y = \gamma * \hat{y}$$

where W, U, b and c are learned weights and biases respectively. The $*$ represents an element-wise product, where γ can be interpreted as an element-wise gate on the information passed through $\hat{y}$.

The output of the gated tanh is then reduced to a K-dimensional vector by passing it through a softmax layer. This vector is called *attention vector*, where the k-th entry contains the importance of the k-th image location with relation to the current question. The attention vector is then used to compute a weighted sum of the image features as follows:

$$\hat{x} = \sum_{k=1}^{K} \alpha_k X_k$$

where α_k is the k-th entry of the attention vector, X_k is the k-th object location and $\hat{x}$ is the resulting 2048×1 dimensional attention weighted image feature.

The attention weighted image features are concatenated with the embedded questions and object features and passed through a fully connected layer with a ReLU activation function. This configuration was inspired by iBOWIMG and works nicely. However, to avoid overfitting we use a dropout layer [21] with $p = 0.5$. Finally, the soft labels are predicted from a fully connected layer with a sigmoid activation function. We found that a sigmoid activation yielded better results than the commonly used softmax activation when using soft labels. The use of soft labels is explained in more detail in Section IV-C.

IV. Analysis

This section presents our findings. We will discuss the advantage of soft labels, object features and give an explanation for the low scores across the different models for the 'number' category as well as information on how we trained the model.

A. Data

We used the first version of the VQA Challenge dataset to train, test and validate our models. The dataset is a collection of images and questions and is split into training, validation and testing sets, which contain 82737, 40504 and 81434 images respectively and three times as many questions. For every question ten answers are annotated by different workers of Amazon Mechanical Turk (AMT). The evaluation metric used by the VQA Challenge is the following:

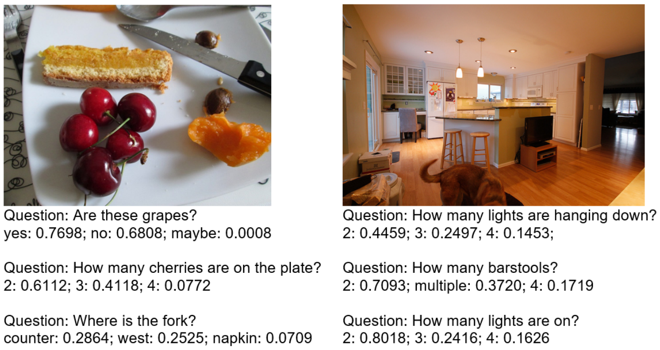

Question: Are these grapes?
yes: 0.7698; no: 0.6808; maybe: 0.0008
Question: How many cherries are on the plate?
2: 0.6112; 3: 0.4118; 4: 0.0772
Question: Where is the fork?
counter: 0.2864; west: 0.2525; napkin: 0.0709
Question: How many lights are hanging down?
2: 0.4459; 3: 0.2497; 4: 0.1453;
Question: How many barstools?
2: 0.7093; multiple: 0.3720; 4: 0.1719
Question: How many lights are on?
2: 0.8018; 3: 0.2416; 4: 0.1626

Figure 3: Failure cases. Each image is shown with the three associated questions and three highest predicted classes respectively.

$$\text{acc(answer)} = \min\left(\frac{\text{\#humans said that answer}}{3}, 1\right) \quad (1)$$

B. Poor accuracy on numbers

By looking at the results of the iBOWIMG model and other neural networks on the VQA leaderboard, one can notice that almost all the models perform poorly in the 'number' category, i.e. on the questions whose answers are supposed to be a number (see Fig. 3). This is not surprising for several reasons. On one hand, the evaluation used for the VQA Challenge performs a string comparison in order to verify if an answer is correct. Thus if the answer is very specific, it is difficult for the model to predict it. For instance if the correct answer is ”three cows” and the prediction is ”three”, it counts as an error even though the prediction was correct. The same happens with synonyms, but this is a general issue for every category of the VQA Challenge, although it is especially common for numbers. Words like 'many', 'lots', 'hundreds', 'dozens', etc. are treated differently even though they often express the same concept. On the other hand, questions whose answers require counting to large numbers also cause problems. Imagine a picture of a stadium crowded with people and someone asking ”How many people are there?”. Different humans would probably estimate a different value or answer using words like ”many” or ”a lot”. If the stadium contained 5431 people for instance, all the numbers in a range close to the actual number of people could be considered correct (or at least partially correct), as nobody will expect a human to perfectly guess or count the number of people. The VQA evaluation metric, however, does not assign any partial points for very close answers. Because of this we propose a different metric, e.g. the absolute relative error, which is more suited for this type of question as it penalizes close predictions less than very different ones.

C. *Soft labels*

As already mentioned in Section IV-A, for every question ten possible answers are annotated by AMT. Furthermore, the evaluation metric considers answers to be correct if at least three humans gave this specific answer, otherwise only partial points or no points are given (see Section IV-A for more details). Thus it is possible that several different answers are correct for the same image-question pair. The simplest and most commonly used approach is to convert answers to actual labels and to perform majority voting, i.e. to consider the most certain answer only. A softmax activation function is then used in the output layer and the model is trained using the categorical cross-entropy loss function.

We believe a better approach is to consider every possible answer as a probability [18]. In order to be coherent with the evaluation metric, the soft labels are produced using formula 1. In this setting a sigmoid activation function is used in the last layer, followed by a binary cross-entropy loss. Note that although the outputs are normalized to $(0, 1)$, they do not represent a probability distribution anymore, as they do not sum up to one. The outputs can therefore be seen as a multinomial logistic regression, where the scores represent the probabilities of a given answer to a given image-question pair. The answer with the highest probability is then selected as the final prediction.

The above formulation presents some advantages compared to the majority voting scheme. Since it is possible to have multiple correct answers for the same question the network can learn more as no information is lost.

For computational reasons the output vocabulary, i.e. the set of candidate answers, was determined from all the correct answers which appear more than three times in the dataset. This yielded a total of 5163 possible answers for the whole dataset (training and validation sets).

D. *Features from object detection*

A weakness of the iBOWIMG model are the visual features. These are vectors of dimension 1024 extracted from the second last layer of GoogLeNet. Although this neural network performs very well on the ImageNet dataset, it does not provide all the necessary information in order to answer certain questions on the VQA dataset. The features are in fact closely related to the output of ImageNet, which is a string describing the image. Most of the information regarding small parts of the image are therefore lost. To compensate this problem there are several solutions. An early approach we followed was to attach a CNN to the iBOWIMG architecture and train it either from scratch end-to-end or using pretrained weights. We experimented with this architecture but due to the high computational requirements we were unable to obtain any noteworthy results. Another possible solution is to use image features with attention or features extracted from a model trained on the same underlying image dataset as the VQA dataset. We were able to obtain a significant improvement by following the latter approach.

	Open-Ended			
	Overall	yes/no	number	others
IMG[3]	**28.13**	64.01	00.42	03.77
BOW[3]	**48.09**	75.66	36.70	27.14
BOWIMG[3]	**52.64**	75.55	33.67	37.37
CompMem[22]	**52.62**	78.33	35.93	34.46
NMN + LSTM[23]	**54.80**	77.70	37.20	39.30
ACK[24]	**55.72**	79.23	36.13	40.08
iBOWIMG[2]	**55.72**	76.55	35.03	41.69
DPPnet[12]	**57.22**	80.71	37.24	41.69
iBOWIMG + YOLO	**57.06**	79.16	33.97	43.42
iBOWIMG + YOLO + SL	**57.47**	78.85	34.42	44.43
iBOWIMG + YOLO + AT	**58.56**	80.54	34.87	45.16
Final Proposed Model	**59.98**	81.30	37.20	46.92

Table I: Performance comparison on test-dev (Open-Ended). SL stands for soft labels, AT for attention and our final proposed model combines all three: YOLO, attention and soft labels.

	Multiple-Choice			
	Overall	yes/no	number	others
IMG[3]	**30.53**	69.87	00.45	03.76
BOW[3]	**53.68**	75.71	37.05	38.64
BOWIMG[3]	**58.97**	75.59	34.35	43.41
iBOWIMG[2]	**61.68**	76.68	37.05	54.44
DPPnet[12]	**62.48**	80.79	38.94	52.16
iBOWIMG + YOLO	**62.90**	79.25	35.95	55.02
iBOWIMG + YOLO + SL	**62.68**	78.91	36.55	54.72
iBOWIMG + YOLO + AT	**64.34**	80.64	37.81	56.40
Final Proposed Model	**65.05**	81.38	39.24	56.94

Table II: Performance comparison on test-dev (Multiple-Choice)

E. *Training*

We trained our model using SGD and Adam optimizer with a learning rate of 0.0001 for 280 epochs and a batch size of 1024. Hyper-parameters were chosen by looking at the evolution of the accuracy and the loss on the original training and validation set. However, in order to be as efficient as possible and not to discard any useful information, the network was then trained again on both the training and validation sets.

The model was run on a single Nividia GeForce GTX 1080 GPU for 12 hours on the ETH Leonhard GPU cluster. Learning takes around 200 GB of RAM. This is mostly due to the high dimensionality of both the bag of words and the output space. Note that by substituting the bag of words with some other sequence embedding layer we could reduce the memory consumption significantly.

V. Results

In this section we will discuss the obtained results and demonstrate the usefulness of attention and object features.

For the questions we found that using a dense layer instead of a word embedding layer in Keras gave significantly better results. Using this trainable layer instead of a pretrained GloVe [25] embedding also resulted in better performance. The tradeoff here being that we have a significantly larger memory footprint and training is more time-consuming.

Another observation is that by only using the number of objects and not their positions from YOLO, the spatial information about the object positions in the image is lost which would be necessary for attention with YOLO. We were not able to embed the positions in a meaningful way. However, this method might be worth pursuing in future work.

One of the main problems of the first version of the VQA dataset is that there is an implicit bias towards questions. Therefore, a simple model consisting only of a bag of words (BOW model) can already achieve a decent performance, as can be seen in Table I and II. The newer VQA dataset [26] mitigates this problem by having an image pair for every question to reduce the prior on the questions.

Since we developed a model on the first VQA dataset it was important to have more emphasis on the images to obtain a competitive score.

We conducted a simple experiment in order to demonstrate that the image features used in our model and described in Section III-B are more effective than the ones used in iBOWIMG, which are obtained from GoogLeNet. We substituted the image features extracted by InceptionV3 with random noise and then predicted using our proposed model. An overall score of 53.25 was obtained for the Multiple-Choice case, which as one can see in Table II is very close to the score obtained by BOW. This shows that the attention mechanism implemented in our model plays a significant role in generating a good prediction.

A similar experiment was conducted for the YOLO features. We replaced the features with a zero vector and observed an overall score of 56.19. An interesting insight is that the individual score for the 'number' category was 31.22, which shows that the YOLO features, even though the overall improvement was only minor, are still useful. This result was in line with our reasoning as we specifically targeted this category as it is the one with the largest margin of possible improvements compared to human performance.

We used Python for our data processing and all of our neural network models were programmed in Keras [27].

VI. Conclusion

We presented a model for VQA based on a neural network with attention and object features. Incorporating object features showed that giving the network a notion of objects can be useful, especially when the question is related to numbers. Furthermore, we believe that the trend of attention enhanced networks is a good path to follow as the current state-of-the-art models have shown. However, unlike most state-of-the-art approaches we were able to obtain competitive results without the use of recurrent neural networks. We believe that this is either because of the simplicity of the questions or because the model is not able to leverage more complex approaches to extract more details from questions. This reasoning is in line with other reports [18]. Additionally, even though the VQA dataset was constructed to remove any biases towards either questions or images, we have found this not to be the case. On the contrary, we found that the models are able to learn some hidden priors on the questions as discussed in Section V. An example of such a case is shown in Figure 3, where there is a strong bias towards the numbers 2, 3 and 4. This phenomenon has also been reported by others ([26], [18]). This flaw has been addressed in the newer VQA dataset, however, as others have pointed out ([28], [29]), most of the questions can still be answered without being able to reason about them or the corresponding images.

As mentioned in the introduction, the current state of VQA has not yet managed to approach human level performance, which exceeds an overall accuracy of 80% [3]. We believe this disparity is mainly caused by the current models not having any additional information other than the dataset on which they have been trained and by not being able to reason about the provided images and questions. We believe that even if the current approaches look to be successful up to a certain degree a more general approach is needed. One such approach might be the use of compositional models ([30], [28]). We believe that this area as well as the search for other, broader approaches could be the basis for future work.

References

[1] R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L.-J. Li, D. A. Shamma, M. Bernstein, and L. Fei-Fei, "Visual genome: Connecting language and vision using crowdsourced dense image annotations," 2016. [Online]. Available: https://arxiv.org/abs/1602.07332

[2] B. Zhou, Y. Tian, S. Sukhbaatar, A. Szlam, and R. Fergus, "Simple baseline for visual question answering," *arXiv preprint arXiv:1512.02167*, 2015.

[3] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. L. Zitnick, and D. Parikh, "VQA: visual question answering," *CoRR*, vol. abs/1505.00468, 2015. [Online]. Available: http://arxiv.org/abs/1505.00468

[4] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2818–2826.

[5] J. Redmon and A. Farhadi, “Yolo9000: Better, faster, stronger,” *arXiv preprint arXiv:1612.08242*, 2016.

[6] M. Malinowski and M. Fritz, “A multi-world approach to question answering about real-world scenes based on uncertain input,” in *Advances in Neural Information Processing Systems*, 2014, pp. 1682–1690.

[7] K. Tu, M. Meng, M. W. Lee, T. E. Choe, and S.-C. Zhu, “Joint video and text parsing for understanding events and answering queries,” *IEEE MultiMedia*, vol. 21, no. 2, pp. 42–70, 2014.

[8] D. Geman, S. Geman, N. Hallonquist, and L. Younes, “Visual turing test for computer vision systems,” *Proceedings of the National Academy of Sciences*, vol. 112, no. 12, pp. 3618–3623, 2015.

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

[10] D. Bahdanau, K. Cho, and Y. Bengio, “Neural machine translation by jointly learning to align and translate,” *arXiv preprint arXiv:1409.0473*, 2014.

[11] L. Ma, Z. Lu, and H. Li, “Learning to answer questions from image using convolutional neural network.” in *AAAI*, vol. 3, no. 7, 2016, p. 16.

[12] H. Noh, P. Hongsuck Seo, and B. Han, “Image question answering using convolutional neural network with dynamic parameter prediction,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 30–38.

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

[14] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, “Learning phrase representations using rnn encoder-decoder for statistical machine translation,” *arXiv preprint arXiv:1406.1078*, 2014.

[15] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, “Going deeper with convolutions,” in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[16] K. J. Shih, S. Singh, and D. Hoiem, “Where to look: Focus regions for visual question answering,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 4613–4621.

[17] J. Lu, J. Yang, D. Batra, and D. Parikh, “Hierarchical question-image co-attention for visual question answering,” in *Advances In Neural Information Processing Systems*, 2016, pp. 289–297.

[18] D. Teney, P. Anderson, X. He, and A. v. d. Hengel, “Tips and tricks for visual question answering: Learnings from the 2017 challenge,” *arXiv preprint arXiv:1708.02711*, 2017.

[19] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, “Microsoft coco: Common objects in context,” in *European conference on computer vision*. Springer, 2014, pp. 740–755.

[20] Y. N. Dauphin, A. Fan, M. Auli, and D. Grangier, “Language modeling with gated convolutional networks,” *arXiv preprint arXiv:1612.08083*, 2016.

[21] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: a simple way to prevent neural networks from overfitting.” *Journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[22] A. Jiang, F. Wang, F. Porikli, and Y. Li, “Compositional memory for visual question answering,” *arXiv preprint arXiv:1511.05676*, 2015.

[23] J. Andreas, M. Rohrbach, T. Darrell, and D. Klein, “Deep compositional question answering with neural module networks. arxiv preprint,” *arXiv preprint arXiv:1511.02799*, vol. 2, 2015.

[24] Q. Wu, P. Wang, C. Shen, A. Dick, and A. van den Hengel, “Ask me anything: Free-form visual question answering based on knowledge from external sources,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 4622–4630.

[25] J. Pennington, R. Socher, and C. D. Manning, “Glove: Global vectors for word representation,” in *Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 1532–1543. [Online]. Available: http://www.aclweb.org/anthology/D14-1162

[26] 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*, vol. abs/1612.00837, 2016. [Online]. Available: http://arxiv.org/abs/1612.00837

[27] F. Chollet *et al.*, “Keras,” https://github.com/keras-team/keras, 2015.

[28] J. Johnson, B. Hariharan, L. van der Maaten, L. Fei-Fei, C. L. Zitnick, and R. B. Girshick, “CLEVR: A diagnostic dataset for compositional language and elementary visual reasoning,” *CoRR*, vol. abs/1612.06890, 2016. [Online]. Available: http://arxiv.org/abs/1612.06890

[29] R. Hu, J. Andreas, M. Rohrbach, T. Darrell, and K. Saenko, “Learning to reason: End-to-end module networks for visual question answering,” *CoRR*, vol. abs/1704.05526, 2017. [Online]. Available: http://arxiv.org/abs/1704.05526

[30] J. Andreas, M. Rohrbach, T. Darrell, and D. Klein, “Learning to compose neural networks for question answering,” *CoRR*, vol. abs/1601.01705, 2016. [Online]. Available: http://arxiv.org/abs/1601.01705