Attention Features Based on Question Types in Visual Question Answering

Belen Saldias-Fuentes
Massachusetts Institute of Technology
belen@mit.edu

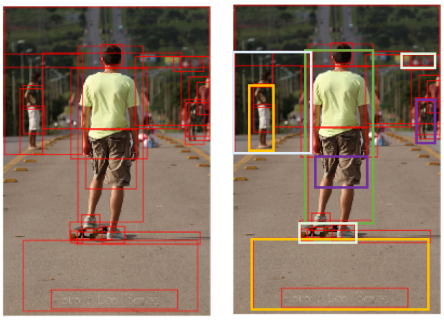
Abstract

When humans count the number of objects in a scene, we may not remember the color specific objects, because we tend to focus our attention on the task we aim to solve. However, we usually have a panoramic view that allows us to understand the context of the scene. Similarly, Visual Question Answering (VQA) machines can be trained to focus on different general and specific image descriptors to become more efficient or accurate. Previous works have mainly focused on features extraction as well as combining different attention types, but have not paid much attention to evaluate which attention features type, bottom-up or top-down, responds better to which question types. In this work, we present a model that concatenates bottom-up and top-down features for VQA. Applying a relevance analysis to abstract and real-world images shows that bottom-up features strongly influence responses to general and ambiguous questions, while top-down attention focuses mainly on object-detection tasks classes.

1. Introduction

Humans can quickly respond to visual questions even when we ignore several scene details because answering questions does not require our full attention. In addition, we use a different attention type to count objects that the type that we use to search for specific image details. In general, we use different strategies to respond to different question types. Some question types are [1]: "what is in," "what color is the," "how many," "are they," "why is the," "will the girl," among others. Also, using images' contexts may allow us to respond to specific question types better.

Several previous works (see [11] for a survey) have focused on combining different feature types and attention models. However, they fail to provide an analysis of the effect of these attention types on the question types. This work focuses on understanding the relevance of bottom-up and top-down attention features [2] (see figure 1) in the answers to different question types. In addition, we propose a novel VQA model for that purpose.



(a) (b)

Figure 1. Attention types. Bottom-up attention (a) give the same importance to all salient regions of an image, providing a general view of the scene. Top-down attention (b) weights salient regions according to the question asked.

Our model outperforms (63.61%) the reference model of the 2017 VQA challenge winner (63.15%), as well as an efficient implementation of that reference model (63.37%), for the MSCOCO dataset (section 3.3). However, we are far behind the 2018 VQA future winner, which reports an overall accuracy score of 70%.

The presented analysis based on question types shows that machines, as well as humans, can focus on different image descriptors to respond to different questions. In particular, we find that bottom-up attention is used to respond to color-related, general and ambiguous questions, while top-down attention focuses on objects names.

Our main contributions are:

- A model that combines bottom-up and top-down attention in two late fusions (instead of one, as typical models) to predict answers using both attention types.
- A quantitative and qualitative analysis of the influence of each attention type on different question types.

The rest of this paper is organized as follows: Section 2 presents some important related works. Section 3 explains our approach. Section 4 reports quantitative and qualitative results. Finally, in section 5 we state our main conclusions.

2. Related work

The Visual Question Answering (VQA) task was proposed by Agrawal *et al*. [1] in 2015. They also proposed a model to solve this task, released two datasets with images and corwdsourced questions and responses, and conducted a general analysis of these datasets. Since then, several deep neural networks have been used to solve the VQA task.

Wu *et al*.'s 2017 survey [11] shows that current attention models mainly use top-down approaches. However, this tendency shifted during the last year. Anderson *et al*. [2], through a very well-defined extensive exploration of architectures, prove that combining bottom-up and top-down attention creates a strong basis for any VQA model. Anderson *et al*. authored the winning entry of the 2017 VQA Challenge (with an overall accuracy score near 63% for the reference model and almost 70% for an ensemble model). using their tips and tricks kindly shared in [10].

For the 2018 VQA challenge, [5] and [9] show that different attention types relate to different question types. In particular, Shi *et al*. [9] propose a question-type-guided attention model that dynamically balances between bottom-up and top-down visual features. Their method sets a state-of-the-art technique and accuracy score (above 80%). They also further develop and outperform the idea we propose in this paper, and show that we are in the right direction.

Unlike all these previous works, our model is designed mainly to deepen our understanding of the attention features effect on question types. We did not find similar analyses during our bibliography review process.

3. Approach

This section describes our approach to understand the effect of attention types on the performance of responses to different question types. We propose a new deep learning model for VQA. We use as base code an open-source efficient PyTorch implementation [3] of the 2017 VQA Challenge's winner[2, 10]. We refer to this efficient model implementation as **base model** (details in section 3.4.1).

3.1. Question embedding

We embed the questions using a publicly available[1] version of the 300-dimensional GloVe word embeddings [7] and the output of a trained Gated Recurrent Unit (GRU) with one hidden layer for those words not present in the embedding, as suggested by [2]. We create a dictionary starting with all words from the training and validation set. Questions and answers are preprocessed using the following procedure. First, sentences are converted to lower case and split into words lists using single space. We then replace all contractions and punctuation symbols and convert number names to digits. Finally, we filter the possible answers to all those words which appear at least nine times across responses. Note that this response reduction sets an upper boundary to the achievable classification accuracy score.

3.2. Image features

3.2.1 Bottom-up attention features

We extract bottom-up features from a pre-trained model [2]. It uses a Faster R-CNN on top of a ResNet-101 CNN [8] to output a set of attention regions (bounding-boxes) along with associated feature vectors. These boxes are filtered out through non-max suppression to get the boxes that intersect the least but cover the highest number of objects in the image (using intersection-over-union and a confidence threshold). The output descriptors per region are mean pooled convolutional features, concatenated with an embedding of the ground-truth object class within that region and then passed through a softmax function. In this work, we extract 36 boxes, along with corresponding 1024-dimensional vectors, to have a uniformly-sized input per image.

3.2.2 Top-down attention features

Top-down attention is one of the most common approaches in current VQA models [11, 2]. It looks for something specific and determines feature weights to solve that specific task. Top-down attention is different from bottom-up attention, since the latter looks for all salient regions instead of driving the search by a specific task or question. In this work, we use top-down attention over bottom-up regions so that we can solve specific tasks and general tasks.

The 2017 VQA winner uses a concatenation-based attention module, where each image descriptor is concatenated with its question embedding and passed together through a softmax function to obtain top-down attention weights. In this work, we use an attention module proposed by Hu *et al*. [3] in the base model. It was inspired by compositional modular networks [4]. The base model proposed is to have a module that inputs the image descriptor and the question embedding in different channels, and then have one fully-connected layer per channel, use fusion through element-wise product, and use a softmax to output the top-down attention weights. This module showed to have a great positive impact on the VQA model performance without requiring extra data or incurring in extra computational costs (find the base model details in section 3.4.1).

Formally, we define Hu *et al*.'s proposed attention model [3] as follows[2]. Let v_i be the feature vector corresponding to the $i = 1 \dots K$ region in one image; let the associated question embedding be q. We define:

$$\hat{v}_i = fc\_v(v_i) \qquad \hat{q} = fc\_q(q)$$

$$r_i = \hat{v}_i \circ \hat{q}$$

[1] http://nlp.stanford.edu/projects/glove/

[2] Based on the publicly available code and its documentation.

$$a_i = w\, r_i$$
$$\alpha = \mathrm{softmax}(a)$$
$$\hat{v} = \sum_{i=1}^{K} \alpha_i v_i$$

where fc_v and fc_c are non-linear fully-connected layers, r_i is the fc_v and fc_c element-wise product, w is a learned parameter vector, and α are the attention weights that are normalized over all locations with a softmax function. Finally, $\hat{v}$ represents the attended image as one 1024-dimensional vector.

3.3. Datasets

3.3.1 Abstract scenes

For the evaluation of high-level reasoning required for VQA, we use the official VQA dataset of abstract scenes [1], which contains 30,000 images and 90,000 questions for training and validation. This set contains indoor and outdoor scenes that are created using more than 100 objects and 31 animals in different poses. After question embedding, we count 426 possible output classes for this dataset.

3.3.2 Real-world images

For the evaluation of low-level vision tasks, we use the MSCOCO 2014 dataset [6], which contains more than 123,000 images and 650,000 questions for training and validation. MSCOCO images are real-world scenes captured in diverse places and with much higher visual complexity than the abstract scenes dataset. After question embedding, we count 3,129 possible output classes for this dataset.

3.4. Proposed architecture

3.4.1 Base Model

As base code, we use an open-source efficient PyTorch implementation of the winning entry of the 2017 VQA Challenge, implemented by Hu *et al*. [3][3]. We would like to acknowledge their work. Even though it has not been officially published, it has made a significant contribution to the development of this and other works [5][4].

The base model is called efficient because it takes around one hour to converge in a Titan XP GPU, while the 2017 VQA winner takes between 12 and 18 hours to converge in a Tesla K40 GPU. This efficiency does not come with any performance reduction; the base model reports a validation accuracy score of 63.58% (we obtained 63.37% for the base model), while the winning entry reaches 63.15%.

The main modifications that the base model proposes, and our decisions about them, are as follows:

- *Number of detected objects*: The base model uses a fixed number of objects per image, while the 2017 winner uses non-maximum suppression to find the optimum number of objects (form 10 to 100) per image. We extracted 36 bounding-boxes per image for both real-world and abstract scenarios.

- *Activation functions*: The base model uses simple ReLU activations, while the winning entry model recommends using gated tanh. Since we want to analyze the influence of each feature attention type in the final classification, we use ReLU activations to easily understand the activation function shape.

- *Top-down attention*: The base model's and 2017 winner's attention methods are explained in section 3.2.2. We use the base model proposal because it was crucial to high performance.

3.4.2 Our Model

We propose an architecture, shown in figure 2, that uses both feature attention types: bottom-up and top-down.

On the one hand, top-down attended images are represented with a 1024-dimensional vector, as explained in section 3.2.2. On the other hand, to reduce the 36 bottom-up attention regions to a single representative vector per image, we followed an ablation study. For both real-world images and abstract scenes, we compared configurations that reduced the boxes to the max-norm vector, the min-norm vector, the median-norm vector, and the average vector. Finally, we use the average over-image locations to represent each image's bottom-up attention in a 1024-dimensional vector.

Most of the recent methods for VQA have used a single late fusion of the language and visual channels (question embeddings and image representations, respectively) previous to the classifier. In this work, we propose the use of two late fusions through element-wise products, one fusion per feature attention type. We then concatenate these fusions to feed the classifier with a 2048-dimensional input vector. Note that by concatenating these features, we have half the concatenated vector (dimensions 1 to 1024) with one type of attention and the other half (dimensions 1025 to 2048) with the other type of attention. This conceptual separation allows us to analyze which half of each vector is activated the most depending on the question types.

Note that our use of bottom-up attention can also be seen as a residual connection that helps the model to avoid vanishing gradients and not to forget about the original data (which are bottom-up features).

We use the same classifier structure proposed in [10]. We use a ReLU activation function instead of a gated tanh (as explained in 3.4.1). Also, our input size is 2048-

[3] Available at https://github.com/hengyuan-hu/bottom-up-attention-vqa
[4] Available at https://github.com/jnhwkim/ban-vqa

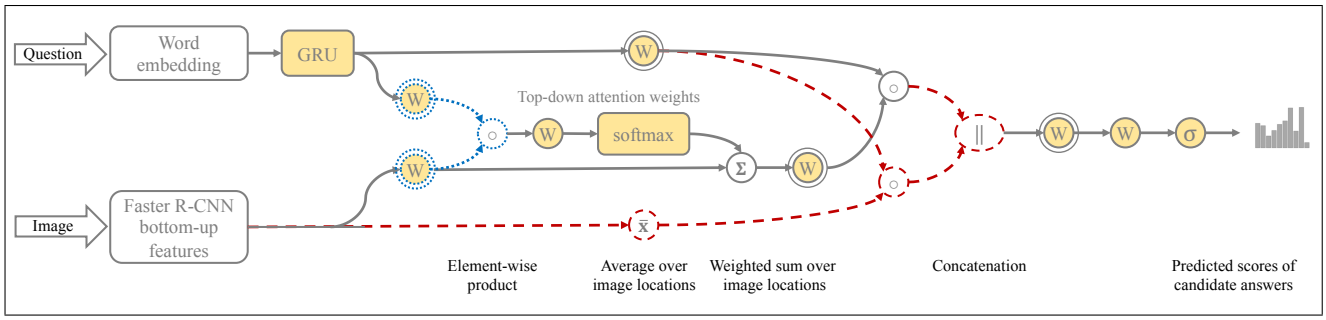


Figure 2. Overview of the proposed architecture. A deep neural network that uses two late fusions of the input question and image, followed by a multi-class classifier. Yellow elements are learned for this task. We use a pre-trained word embedding and a neural net that outputs bottom-up attention features. The elements Ⓦ represent linear layers, and Ⓦ non-linear layers (ReLU). All continuous grey elements and connections are proposed in the winning entry of the 2017 VQA Challenge. Dotted blue elements and connections are proposed in the base model implementation. Dashed red elements and connections are our contribution to this VQA architecture. Note that dotted blue lines also refer to the start of top-down attention while dashed red lines (before the concatenation) represent the bottom-up attention track.

rather than 1024-dimensional. Both classifiers return 1024-dimensional vectors that are dimensionally reduced by a linear transformation to match the number of output classes.

3.5. Training

We found that a p2.8xlarge AWS instance works best for the process of feature extraction, training and evaluation. In this machine, our model, as well as the base model, takes 3 hours to converge using real-world images and 1 hour for abstract scenes. Using this machine, we have enough GPUs (NVIDIA K80) and RAM to run different models and analyses simultaneously. On the other hand, for training, validation, and all analyses, we use a batch size of 512.

As proposed in [10], we use soft scores as ground truth targets. This representation casts the traditional classification task as a regression problem. We use the cross-entropy loss function and the Adamax optimizer with the PyTorch default parameters (lr=2e-3, betas=(0.9, 0.999), eps=1e-8, weight_decay=0), as proposed in [3].

4. Evaluation

4.1. Classification accuracy score

Table 1 shows results for the validation sets in both abstract and real-world scenarios. To make our model comparable to the base model, we also trained the base model until it reached peak performance as defined by its authors. Note that for real-world images, the proposed model reaches its best performance faster than the base model.

Table 2 shows results for training sets. Note that the accuracy score of the proposed model for real-world images is lower than that of the base model because our model uses fewer epochs to reach its optimum validation score.

We do not report tests results for two reasons. First, to obtain test results we need to submit our prediction to an official API of the VQA Challenge and wait until results are released. Second, the extraction of the bottom-up features is time-consuming and would have taken about 15 hours in total for both datasets using expensive AWS machines.

Model	Abstract scenes	Real-world images
Base model	70.72% (33th)	63.37% (19th)
Proposed model	71.15% (33th)	63.61% (12th)
Upper-bound	95.73%	92.66%

Table 1. Validation set results. We report the best validation accuracy scores per scenario. We found the best results running 40 epochs for abstract scenes, and 30 epochs for real-world images. Numbers in parentheses indicate the epoch in which these models reached their best performance.

Model	Abstract scenes	Real-world images
Base model	93.27%	79.62%
Proposed model	93.22%	75.06%

Table 2. Training set accuracy scores. We report the training scores to understand which model tends to overfit more. We see that both models behave similarly with abstract scenes. However, for real-world images, we can see that in comparison with the base model, our model reaches higher accuracy in fewer epochs (details in table 1) and memorizes less training data.

4.2. Question type analysis

4.2.1 Obtaining question types

Question types are defined from the initial words of a question. In the datasets that we use (section 3.3), most of the instances already have a type label. We tag the unlabeled questions without as follows. First, we try to match these questions with existing tags. If, after this matching process, we still have unlabeled questions, we create new types using the first two words of the untagged items and assign them

as labels. More details are provided in section 4.2.4.

4.2.2 Attention types relevance

For the analyses in this section, we use the outputs of the ReLU function in the output classifier (section 3.4.2). Note that this ReLU inputs the normalized 2048-dimensional concatenated vectors (containing 1024 dimensions per attention type). Recall that a ReLU returns the input value unless it is a negative number, in which case the output is 0.

To define whether one attention type is more relevant than the other, we proceed as follows. For every question type, we extract the 2048-dimensional activated vector for all image-question pairs belonging to that question type. We then add the first 1024 and the last 1024 dimensions of each activated vector to end up with 2-d representations for each image-question pair, where the first element is the sum of the activation weights of top-down features, and the second element is that of bottom-up features. Next, for each question type, we add the 2-d vectors of its image-question pairs. We end up with two values per question type: the sum of top-down and the sum of bottom-up weights. For further analysis, we normalize these attention weights to determine the percentage weight of each attention type.

We cannot simply assume from the above-defined calculations that the attention type with the highest percentage is responsible for the question responses. Hence, we set a confidence interval (CI) for the average difference between the percentages of attention types. All question types with differences out of the CI are considered as affected by our model and used for further analysis. We repeat this procedure separately for training and validation sets.

4.2.3 Qualitative analysis

Figure 4 shows word-clouds for the question types whose responses improve their performance with our model. We set a 60% CI. Note that for questions about abstract scenes, we improve the performance of responses of fewer question types than for real-world images, maybe because these scenes are easy to represent with both attention types, and because that set has fewer question types (see section 4.2.4). Extra relevant conclusions are noted in figure 4's caption.

4.2.4 Quantitative analysis

In table 3 and figure 3 we show an overview of the question types for validation sets. In tables 4 and 4.2.4 we show the influence of the attention types on our results, considering a 75% CI. We make the attention types and determinations regarding their relevance (see section 4.2.2) publicly available in order to encourage further exploration[5].

[5] https://github.com/bcsaldias/VQA-analysis/tree/master/analysis/public-data

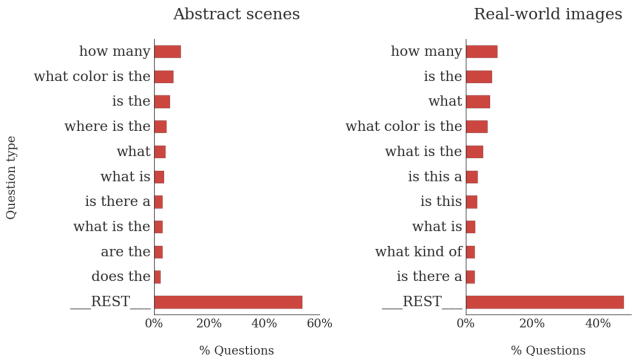


Figure 3. Top 10 questions types per scenario.

	Abstract scenes	Real-world images
Total number of question types	265 (train+val: 438)	776 (train+val: 1650)
Max samples	69	66
Median samples	2	2
Min samples	1	1
Avg samples	113	276

Table 3. Statistics for the number of samples per question type. We can see that in both scenarios, half of the question types have fewer than (or exactly) 2 samples. These long-tail distributions inspire us to focus in general (bottom-up) structures to predict answers. More details of question type distributions are available in [1].

Attention features	Abstract scenes	Real-world images
Bottom-up	26%	46%
Top-down	74%	54%

Table 4. Feature types' contributions to our accuracy scores. Note that for abstract scenes, our model mainly considers top-down features. The reason might be that, in this world with few details, top-down attention already covers most of the image information.

Attention features	Abstract scenes	Real-world images
Bottom-up	17%	28%
Top-down	83%	72%

Table 5. Features types' contributions to the improvement of accuracy scores achieved by our model. Note that in figure 4 we have fewer question types for bottom-up attention, which is explained by this table. Also, as we suggest in figure 4, "how many" is the question type most affected by our model, showing that abstract-scenes are strongly biased toward top-down attention.

5. Conclusion

We present a novel architecture for VQA that concatenates bottom-up and top-down features. This concatenation allows the model to respond to questions using contextual as well as question-specific information, while weighting them differently for each question type. We outperform

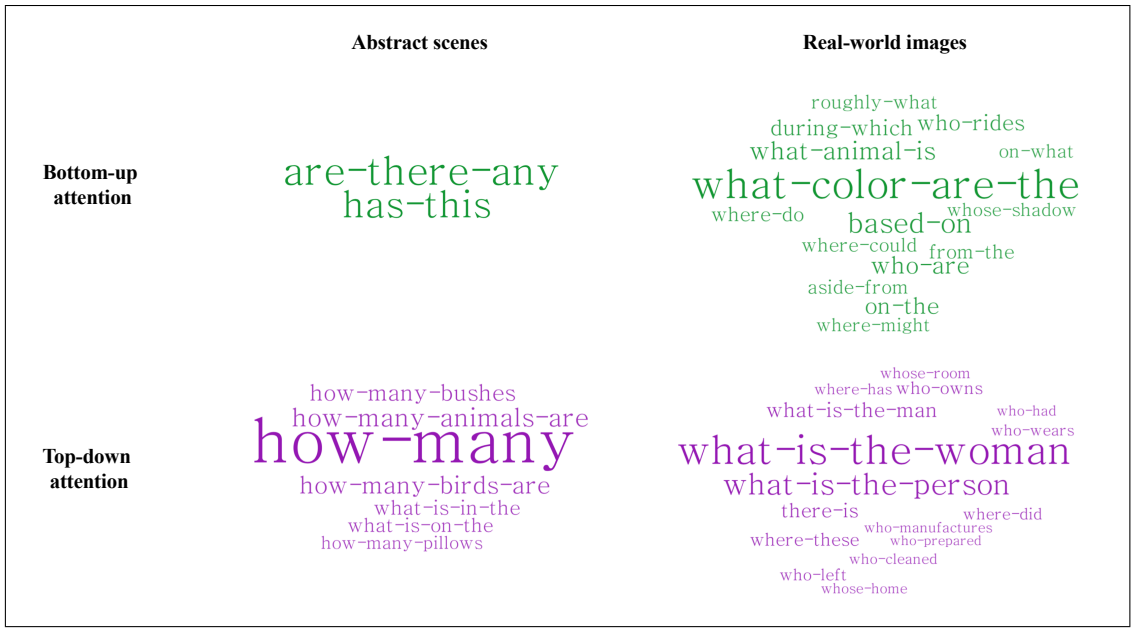


Figure 4. Question types with improved accuracy scores in the proposed model. Question types are clustered based on what features were the most relevant for them. Word sizes allow us to rank how many new objects the proposed model classifies correctly for each question type, but they do not represent actual magnitudes or proportions. Note that the question types in each world-cloud are very different from those of the others. The counting task ("how many") was improved the most; one reason could be that the proposed model can learn very specific top-down attention features while still having generalization skills from bottom-up attention. We can also see that bottom-up attention features for complex (real-world) images allow us to respond better to ambiguous question types such us "roughly what," "where could," and "where might." Overall, we can see that bottom-up attention tends to respond to more general questions, whose responses are not necessarily based on object names (object-detection tasks classes) but on other features like colors or future/past/present actions, while top-down attention strongly focuses on the objects class names to produce responses for counting objects or detecting locations. Note that each question type in this visualization has more than two samples; altogether they represent nearly the 4% of all question types and cover about the 5% of the total amount of question samples for validation sets. These percentages could explain the small improvement in accuracy scores presented in the quantitative analyses section.

the 2017 VQA Challenge's, but not this year's, winner. Furthermore, we conduct analyses of the relevance of these feature types in relation to question types. We show that using bottom-up attention improves the performance of responses to general and ambiguous question types, while top-down attention heavily focuses on object-detection tasks classes.

Acknowledgements. We would like to acknowledge Hu *et al.*'s [3] work, that allowed us to succeed in ours. This work was supported by the MIT 6.869 class and the Lab for Social Machines at MIT via AWS credits.

References

[1] A. Agrawal, J. Lu, S. Antol, M. Mitchell, C. L. Zitnick, D. Batra, and D. Parikh. Vqa: Visual question answering. *arXiv preprint arXiv:1505.00468*, 2015.

[2] P. Anderson, X. He, C. Buehler, D. Teney, M. Johnson, S. Gould, and L. Zhang. Bottom-up and top-down attention for image captioning and visual question answering. In *CVPR*, 2018.

[3] H. Hu, A. Xiao, and H. Huang. An efficient pytorch implementation of the winning entry of the 2017 vqa challenge. `https://github.com/hengyuan-hu/bottom-up-attention-vqa`, 2018.

[4] R. Hu, M. Rohrbach, J. Andreas, T. Darrell, and K. Saenko. Modeling relationships in referential expressions with compositional modular networks. In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, pages 4418–4427. IEEE, 2017.

[5] J.-H. Kim, J. Jun, and B.-T. Zhang. Bilinear Attention Networks. *arXiv preprint arXiv:1805.07932*, 2018.

[6] T. Lin, M. Maire, S. J. Belongie, L. D. Bourdev, R. B. Girshick, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft COCO: common objects in context. *CoRR*, abs/1405.0312, 2014.

[7] J. Pennington, R. Socher, and C. D. Manning. Glove: Global vectors for word representation. 2014.

[8] S. Ren, K. He, R. B. Girshick, and J. Sun. Faster R-CNN: towards real-time object detection with region proposal networks. *CoRR*, abs/1506.01497, 2015.

[9] Y. Shi, T. Furlanello, S. Zha, and A. Anandkumar. Question type guided attention in visual question answering. *arXiv preprint arXiv:1804.02088*, 2018.

[10] D. Teney, P. Anderson, X. He, and A. van den Hengel. Tips and tricks for visual question answering: Learnings from the 2017 challenge. *arXiv preprint arXiv:1708.02711*, 2017.

[11] Q. Wu, D. Teney, P. Wang, C. Shen, A. Dick, and A. van den Hengel. Visual question answering: A survey of methods and datasets. *Computer Vision and Image Understanding*, 163:21–40, 2017.