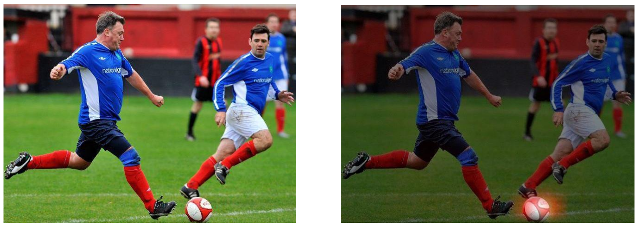
Show, Ask, Attend, and Answer:
A Strong Baseline For Visual Question Answering

Vahid Kazemi Ali Elqursh
Google Research
1600 Amphitheater Parkway
{vahid, elqursh}@google.com

Abstract

This paper presents a new baseline for visual question answering task. Given an image and a question in natural language, our model produces accurate answers according to the content of the image. Our model, while being architecturally simple and relatively small in terms of trainable parameters, sets a new state of the art on both unbalanced and balanced VQA benchmark. On VQA 1.0 [2] open ended challenge, our model achieves 64.6% accuracy on the test-standard set without using additional data, an improvement of 0.4% over state of the art, and on newly released VQA 2.0 [8], our model scores 59.7% on validation set outperforming best previously reported results by 0.5%. The results presented in this paper are especially interesting because very similar models have been tried before [32] but significantly lower performance were reported. In light of the new results we hope to see more meaningful research on visual question answering in the future.

What color is the ball?



- **red and white: 0.33**
- red: 0.22
- white and red: 0.19
- white: 0.17
- red, white: 0.01

Figure 1. Top 5 predictions from our model and their probabilities for an example image/question pair. On the right we visualize the corresponding attention distribution produced by the model.

caption难评估；难辨别模型是否真的学到了，因为一个通用话可以作为很多img的caption

1. Introduction

Deep neural networks in the last few years have made dramatic impact in computer vision and natural language processing fields. We are now able to build models that recognize objects in the images with high accuracy [15, 26, 9]. But we are still far from human level understanding of images. When we as humans look at images we don't just see objects but we also understand how objects interact and we can tell their state and properties. Visual question answering (VQA) [2] is particularly interesting because it allows us to understand what our models truly see. We present the model with an image and a question in the form of natural language and the model generates an answer again in the form of natural language.

A related and more throughly researched task to VQA is image caption generation [31, 28], where the task is to generate a representative description of an image in natural language. A clear advantage of VQA over caption generation is that evaluating a VQA model is much easier. There is not a unique caption that can describe an image. Moreover, it is rather easy to come up with a single caption that more or less holds for a large collection of images. There is no way to tell what the model actually understands from the image based on a generic caption. Some previous work have been published that tried to mitigate this problem by providing dense [12] or unambiguous captions [19], but this problem is inherently less severe with VQA task. It is always possible to ask very narrow questions forcing the model to give a specific answer. For these reasons we believe VQA is a good proxy task for creating rich representations for modeling language and vision.

Some novel and interesting approaches [6, 22] have been published in the last few years on visual question answering that showed promising results. However, in this work, we show that a relatively simple architecture (compared to the recent works) when trained carefully bests state the art.

Figure 2 provides a high level overview of our model. To summarize, our proposed model uses long short-term memory units (LSTM) [11] to encode the question, and a deep residual network [9] to compute the image features. A soft attention mechanism similar to [31] is utilized to compute multiple glimpses of image features based on the state of the LSTM. A classifier than takes the image feature glimpses and the final state of the LSTM as input to produce probabilities over a fixed set of most frequent answers. On VQA 1.0 [2] open ended challenge, our model achieves 64.6% accuracy on the test-standard set without using additional data, an improvement of 0.4% over state of the art, and on newly released VQA 2.0 [8], our model scores 59.7% on validation set outperforming best reported results by 0.5%.

This paper proves once again that when it comes to training neural networks the devil is in the details [4].

2. Related work

In this section we provide an overview of related work.

Convolutional neural networks (CNNs) [16] have revolutionized the field of computer vision in the recent years. Landmark paper by Krizhevsky et al. [15] for the first time showed great success on applying a deep CNN on large scale ImageNet [5] dataset achieving a dramatic improvement over state of the art methods that used hand designed features. In the recent years researchers have been hard at work training deeper [26], very deep [27], and even deeper [9] neural networks. While success of neural networks are commonly attributed to larger datasets and more compute power, there are a lot of details that we know and consider now that were not known just a few years ago. These include choice of activation function [21], initialization [7], optimizer [14], and regularization [10]. As we show in this paper at times getting the details right is more important than the actual architecture.

When it comes to design of deep neural networks, very few ideas have been consistently found advantageous across different domains. One of these ideas is notion of attention [20, 28], which enables deep neural networks to extract localized features from input data.

Another neural network model that we take advantage of in this work is Long Short-Term Memory (LSTM) [11]. LSTMs have been widely adopted by machine learning researchers in the recent years and have shown oustanding results on a wide range of problems from machine translation [3] to speech recognition [24].

All of these ideas have already been applied to visual question answering task. In fact the model that we describe in this work is very similar to stacked attention networks [32], nevertheless we show significant improvement over their result (5.8% on VQA 1.0 dataset). While more recently much more complex and expensive attention models have been explored [6, 22, 18] their advantage is unclear in the light of the results reported in this paper.

3. Method

Figure 2 shows an overview of our model. In this section we formalize the problem and explain our approach in more detail. 就是直接做分类的

We treat visual question answering task as a classification problem. Given an image I and a question q in the form of natural language we want to estimate the most likely answer $\hat{a}$ from a fixed set of answers based on the content of the image.

$$\hat{a} = \arg\max_{a} P(a|I, q) \tag{1}$$

where $a \in \{a_1, a_2, ..., a_M\}$. The answers are chosen to be the most frequent answers from the training set.

3.1. Image embedding

We use a pretrained convolutional neural network (CNN) model based on residual network architecture [15] to compute a high level representation ϕ of the input image I.

$$\phi = \text{CNN}(I) \tag{2}$$

ϕ is a three dimensional tensor from the last layer of the residual network [9] before the final pooling layer with $14 \times 14 \times 2048$ dimensions. We furthermore perform l_2 normalization on the depth (last) dimension of image features which enhances learning dynamics.

3.2. Question embedding

We tokenize and encode a given question q into word embeddings $E_q = \{\mathbf{e}_1, \mathbf{e}_2, ..., \mathbf{e}_P\}$ where $\mathbf{e}_i \in \mathcal{R}^D$, D is the length of the distributed word representation, and P is the number of words in the question. The embeddings are then fed to a long short-term memory (LSTM) [11].

$$\mathbf{s} = \text{LSTM}(E_q) \tag{3}$$

We use the final state of the LSTM to represent the question.

3.3. Stacked attention

SAN

Similar to [32], we compute multiple attention distributions over the spatial dimensions of the image features.

$$\alpha_{c,l} \propto \exp F_c(\mathbf{s}, \phi_l) \quad \ni \quad \sum_{l=1}^{L} \alpha_{c,l} = 1 \tag{4}$$

$$\mathbf{x}_c = \sum_{l} \alpha_{c,l} \phi_l \tag{5}$$

Each image feature glimpse $\mathbf{x}_c$ is the weighted average of image features ϕ over all the spatial locations $l = \{1, 2, ..., L\}$. The attention weights $\alpha_{c,l}$ are normalized separately for each glimpse $c = 1, 2, ..., C$.

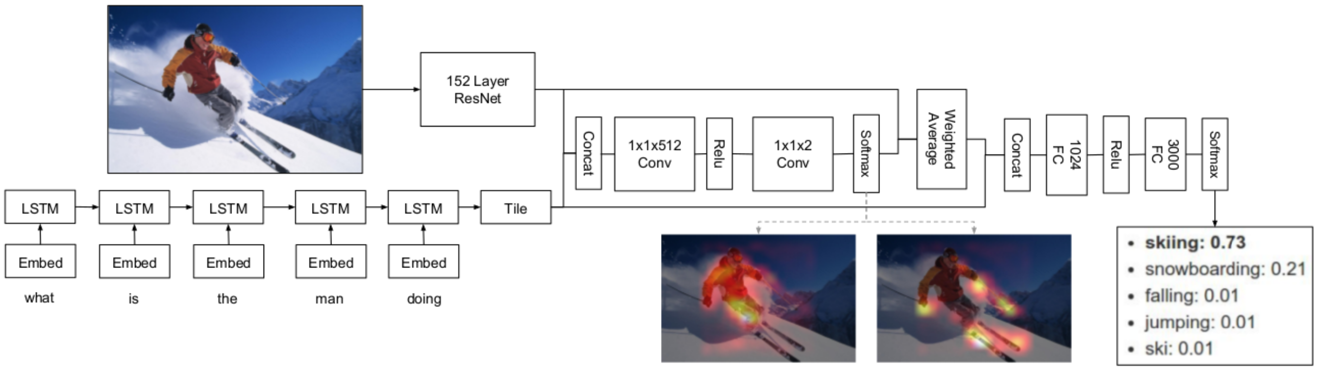


Figure 2. An overview of our model. We use a convolutional neural network based on ResNet [9] to embed the image. The input question is tokenized and embedded and fed to a multi-layer LSTM. The concatenated image features and the final state of LSTMs are then used to compute multiple attention distributions over image features. The concatenated image feature glimpses and the state of the LSTM is fed to two fully connected layers two produce probabilities over answer classes.

In practice $F = [F_1, F_2, ..., F_C]$ is modeled with two layers of convolution. Consequently F_i's share parameters in the first layer. We solely rely on different initializations to produce diverse attention distributions.

3.4. Classifier

Finally we concatenate the image glimpses along with the LSTM state and apply nonlinearities to produce probabilities over answer classes.

$$P(a_i|I, q) \propto \exp G_i(\mathbf{x}, \mathbf{s}) \quad (6)$$

where

$$\mathbf{x} = [\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_C]. \quad (7)$$

$G = [G_1, G_2, ..., G_M]$ in practice is modeled with two fully connected layers.

Our final loss is defined as follows.

$$\mathcal{L} = \frac{1}{K} \sum_{k=1}^{K} -\log P(a_k|I, q) \quad (8)$$

Note that we average the log-likelihoods over all the correct answers $a_1, a_2, ..., a_K$.

4. Experiments

4.1. Dataset

4.1.1 VQA 1.0

We evaluate our model on both balanced and unbalanced versions of VQA dataset. VQA 1.0 [2] is consisted of 204,721 images form the MS COCO dataset [17]. We evaluate our models on the real open ended challenge which consists of 614,163 questions and 6,141,630 answers. The dataset comes with predefined train, validation, and test splits. There is also a 25% subset of the the test set which is referred to as test-dev split. For most of experiments we used the train set as training data and reported the results on the validation set. To be comparable to prior work we additionally train our default model on train and val set and report the results on test set.

4.1.2 VQA 2.0

We also evaluate our model on the more recent VQA 2.0 [8] which is consisted of 658,111 questions and 6,581,110 answers. This version of the dataset is more balanced in comparison to VQA 1.0. Specifically for every question there are two images in the dataset that result in two different answers to the question. At this point only the train and validation sets are available. We report the results on validation set after training on train set.

4.2. Evaluation metric

We evaluate our models on the open ended task of VQA challenge with the provided accuracy metric.

$$Acc(a) = \frac{1}{K} \sum_{k=1}^{K} \min(\frac{\sum_{1 \leq j \leq K, j \neq k} \mathbb{1}(a = a_j)}{3}, 1) \quad (9)$$

where $a_1, a_2, ..., a_K$ are the correct answers provided by the user and $K = 10$. Intuitively, we consider an answer correct if at least three annotators agree on the answer. To get some level of robustness we compute the accuracy over all 10 choose 9 subsets of ground truth answers and average.

5. Results

5.1. Baselines

In this section we describe the details of our default baseline as well as its mutations.

Steps	1K	3K	6K	12K	25K	50K	100K	200K
Default	37.16	46.96	55.07	58.12	59.76	60.65	60.94	60.95
No l_2 normalization	42.65	44.87	49.07	51.12	51.75	52.15	53.56	54.69
No dropout on FC/Conv layers	32.78	38.19	49.02	58.63	57.90	57.42	57.30	56.98
No dropout on LSTM layers	45.85	51.55	55.75	57.63	59.60	59.79	59.95	59.80
No attention	38.09	48.36	51.42	54.43	56.02	57.13	57.65	57.72
Sampling loss	47.24	47.67	51.80	54.85	56.69	57.62	58.85	59.44
With positional features	33.26	41.37	55.36	57.95	59.75	60.44	61.02	61.09
Bidirectional LSTM	42.33	52.38	55.93	58.32	59.99	60.63	60.69	60.63
Word embedding size: 100	39.53	50.24	53.94	56.74	58.92	59.96	60.75	60.90
Word embedding size: 300 (default)	37.16	46.96	55.07	58.12	59.76	60.65	60.94	60.95
Word embedding size: 500	37.21	47.15	55.44	58.43	59.98	60.60	61.01	61.04
LSTM state size: 512	46.59	51.20	55.33	57.96	59.46	60.31	60.79	61.09
LSTM state size: 1024 (default)	37.16	46.96	55.07	58.12	59.76	60.65	60.94	60.95
LSTM state size: 2048	33.24	39.11	50.86	57.48	59.75	60.65	60.93	60.80
LSTM state size: 1024 1024	37.78	48.19	54.28	57.20	59.34	60.22	60.62	60.75
Attention size: 512 1	36.54	45.74	54.23	57.42	59.46	60.22	60.85	60.96
Attention size: 512 2 (default)	37.16	46.96	55.07	58.12	59.76	60.65	60.94	60.95
Attention size: 512 3	36.26	45.16	55.22	57.96	59.77	60.60	60.87	61.12
Attention size: 1024 1	45.60	50.72	54.61	57.57	59.52	60.46	60.92	60.92
Attention size: 1024 2	35.04	42.72	55.56	58.03	59.66	60.54	61.14	61.10
Classifier size: 3000	30.19	43.12	53.38	56.18	57.82	58.25	58.24	58.12
Classifier size: 1024 3000 (default)	37.16	46.96	55.07	58.12	59.76	60.65	60.94	60.95
Classifier size: 2048 3000	48.28	52.57	56.02	58.51	59.96	60.46	60.84	60.95
Classifier size: 1024 1024 3000	44.51	49.53	53.25	55.95	57.59	58.83	60.05	60.66

Table 1. This table shows the result of different mutations of our default model. All models are trained on training set of VQA 1.0 [2] and the accuracy is reported on validation set according to equation 9. Applying l_2 normalization, dropout, and using soft-attention significantly improves the accuracy of the model. Some of the previous works such as [6] had used the sampling loss, which we found to be leading to significantly worse results and longer training time. Different word embedding sizes and LSTM configurations were explored but we found it to be not a major factor. Contrary to results reported by [32] we found using stacked attentions to only marginally improve the result. We found a two layer deep classifier to be significantly better than a single layer, adding more layers or increasing the width did not seem to improve the results.

In all of the baselines input images are scaled while preserving aspect ratio and center cropped to 299×299 dimensions. We found stretching the image to harm the performance of the model. Image features are extracted from pre-trained 152 layer ResNet [9] model. We take the last layer before the average pooling layer (of size $14 \times 14 \times 2048$) and perform l_2 normalization in the depth dimension.

The input question is tokenized and embedded to a $D = 300$ dimensional vector. The embeddings are passed through $\tanh$ nonlinearity before feeding to the LSTM. The state size of LSTM layer is set to 1024. Per example dynamic unrolling is used to allow for questions of different length, although we cap maximum length of the questions at 15 words.

To compute attention over image features, we concatenate tiled LSTM state with image features over the depth dimension and pass through a 1×1 dimensional convolution layer of depth 512 followed by ReLU [21] nonlinearity. The output feature is passed through another 1×1 convolution of depth $C = 2$ followed by softmax over spatial dimensions to compute attention distributions. We use these distributions to compute two image glimpses by computing the weighted average of image features.

We further concatenate the image glimpses with the state of the LSTM and pass through a fully connected layer of size 1024 with ReLU nonlinearity. The output is fed to a linear layer of size $M = 3000$ followed by softmax to produce probabilities over most frequent classes.

We only consider top $M = 3000$ most frequent answers in our classifier. Other answers are ignored and do not contribute to the loss during training. This covers 92% of the answers in the validation set in VQA dataset [2].

We use dropout of 0.5 on input features of all layers including the LSTM, convolutions, and fully connected layers.

We optimize this model with Adam optimizer [14] for $100K$ steps with batch size of 128. We use exponential decay to gradually decrease the learning rate according to the following equation.

$$l_{step} = 0.5^{\frac{\text{step}}{\text{decay steps}}} l_0$$

The initial learning rate is set to $l_0 = 0.001$, and the decay steps is set to $50K$. We set $\beta_1 = 0.9$ and $\beta_2 = 0.999$.

During training CNN parameters are kept fixed. The rest of the parameters are initialized as suggested by Glorot et al. [7].

Table 1 shows the performance of different baselines on validation set of VQA 1.0 [2] when trained on the training set only. We have reported results for the following mutations of our default model:

- **No l_2 norm**: ResNet features are not l_2 normalized.
- **No dropout on FC/Conv**: Dropout is not applied to the inputs of fully connected and convolution layers.
- **No dropout on LSTM**: Dropout is not applied to the inputs of LSTM layers.
- **No attention**: Instead of using soft-attention we perform average spatial pooling before feeding image features to the classifier.
- **Sampled loss**: Instead of averaging the log-likelihood of correct answers we sample one answer at a time.
- **With positional features**: Image features ϕ are augmented with x and y coordinates of each cell along the depth dimension producing a tensor of size $14 \times 14 \times 2050$.
- **Bidirectional LSTM**: We use a bidirectional LSTM to encode the question.
- **Word embedding size**: We try word embeddings of different sizes including 100, 300 (default), and 500.
- **LSTM state size**: We explore different configurations of LSTM state sizes, this include a one layer LSTM of size 512, 1024 (default), and 2048 or a stacked two layer LSTM of size 1024.
- **Attention size**: Different attention configurations are explored. First number indicates the size of first convolution layer and the second number indicates the number of attention glimpses.
- **Classifier size**: By default classifier G is consisted of a fully connected layer of size 1024 with ReLU non-linearity followed by a $M = 3000$ dimensional linear layer followed by softmax. We explore shallower, deeper, and wider alternatives.

l_2 normalization of image features improved learning dynamics leading to significantly better accuracy while reducing the training time.

We observed that applying dropout on multiple layers (including fully connected layers, convolutions, and LSTMs) is crucial to avoid over-fitting on this dataset.

As widely reported we confirm that using soft-attention significantly improves the accuracy of the model.

Different word embedding sizes and LSTM configurations were explored but we found it to be not a major factor. A larger embedding size with a smaller LSTM seemed to work best.

Some of the previous works such as [6] had used the sampling loss, which we found to be leading to significantly worse results and longer training time.

Contrary to results reported by [32] we found using stacked attentions to only marginally improve the result.

We found a two layer deep classifier to be significantly better than a single layer, adding more layers or increasing the width did not seem to improve the results.

5.2. Comparison to state of the art

Table 2 shows the performance of our model on VQA 1.0 dataset. We trained our model on train and validation set and tested the performance on test-standard set. Our model achieves an overall accuracy of 64.6% on the test-standard set, outperforming best previously reported results by 0.4%. All the parameters here are the same as the default model.

While architecturally our default model is almost identical to [32], some details are different. For example they use the VGG [25] model, while we use ResNet [9] to compute image features. They do not mention l_2 normalization of image features which found to be crucial to reducing training time. They use SGD optimizer with momentum $\mu = 0.9$, while we found that Adam [14] generally leads to faster convergence.

We also reported our results on VQA 2.0 dataset 3. At this point we only have access to train and validation splits for this dataset. So we trained the same model on the training set and evaluated the model on the validation set. Overall our model achieves 59.67% accuracy on the validation set which is about 0.5% higher than best previously reported results.

6. Conclusion

In this paper we presented a new baseline for visual question answering task that outperforms previously reported results on VQA 1.0 and VQA 2.0 datasets. Our model is architecturally very simple and in essence very similar to the models that were tried before, nevertheless we show once the details are done right this model outperforms all the previously reported results.

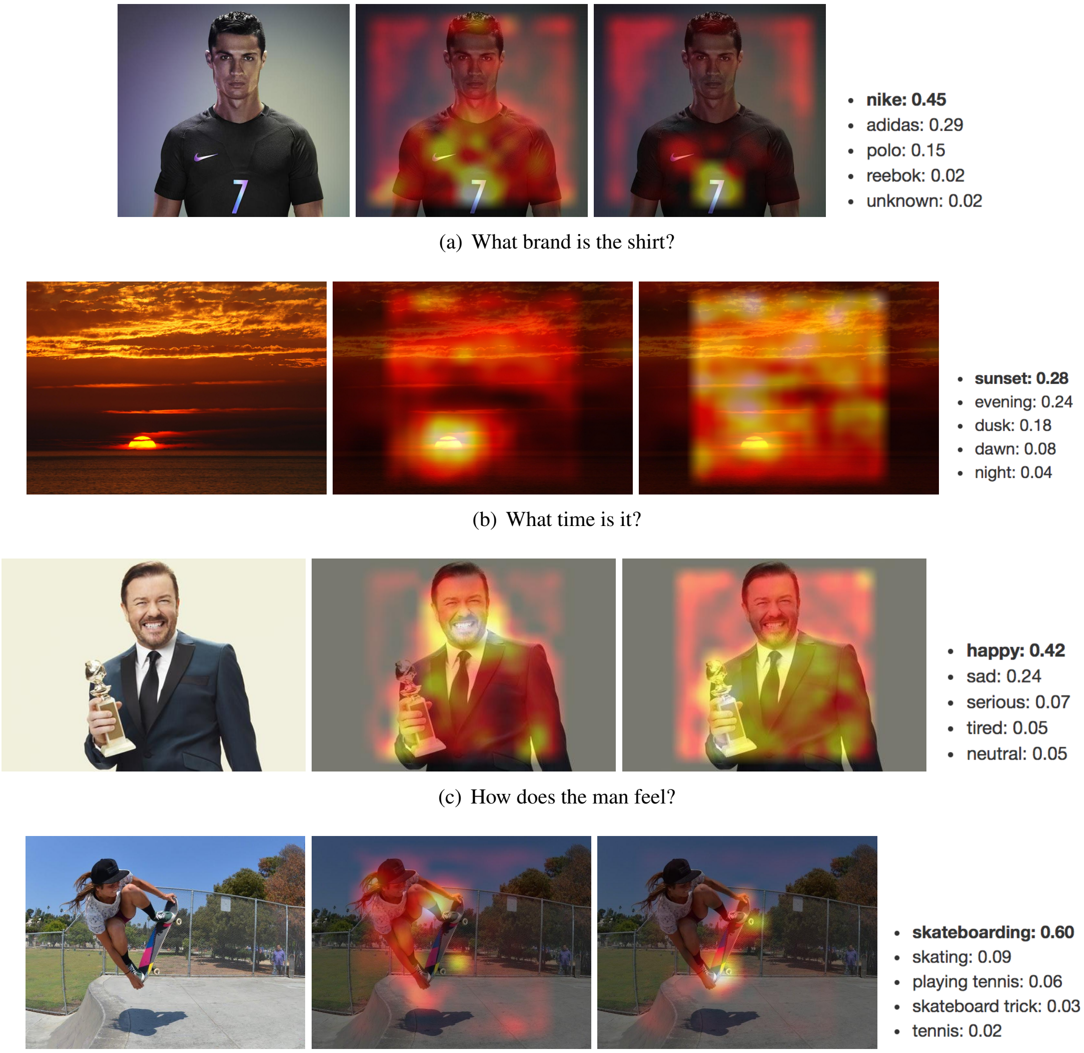
References

[1] J. Andreas, M. Rohrbach, T. Darrell, and D. Klein. Neural module networks. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 6

[2] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. L. Zitnick, and D. Parikh. Vqa: Visual question answering. In *International Journal of Computer Vision*, 2015. 1, 2, 3, 4, 5, 6

	Test-Dev				Test-Standard			
Method	Y/N	Num	Other	All	Y/N	Num	Other	All
VQA team [2]	80.5	36.8	43.1	57.8	80.6	36.5	43.7	58.2
SAN (VGG) [32]	79.3	36.6	46.1	58.7	-	-	-	58.9
NMN (VGG) [1]	81.2	38.0	44.0	58.6	-	-	-	58.7
ACK (VGG) [29]	81.0	38.4	45.2	59.2	81.1	37.1	45.8	59.4
DMN+ (VGG) [30]	80.5	36.8	48.3	60.3	-	-	-	60.4
MRN (ResNet) [13]	82.3	38.8	49.3	61.7	82.4	38.2	49.4	61.8
HieCoAtt (ResNet) [18]	79.7	38.7	51.7	61.8	-	-	-	62.1
DAN (VGG) [22]	82.1	38.2	50.2	62.0	-	-	-	-
RAU (ResNet) [23]	81.9	39.0	53.0	63.3	81.7	38.2	52.8	63.2
MCB (ResNet) [6]	82.2	37.7	54.8	64.2	-	-	-	-
DAN (ResNet) [22]	**83.0**	**39.1**	53.9	64.3	**82.8**	38.1	54.0	64.2
Ours (ResNet)	82.2	**39.1**	**55.2**	**64.5**	82.0	**39.1**	**55.2**	**64.6**

Table 2. This table shows a comparison of our model with state of the art on VQA 1.0 dataset. While our model is architecturally simpler and smaller in terms of trainable parameters than most existing work, nevertheless it outperforms all the previous work.



(a) What brand is the shirt?

(b) What time is it?

(c) How does the man feel?

(d) What is the girl doing?

Figure 3. Qualitative results on sample images shows that our model can produce reasonable answers to a range of questions.

Method	Y/N	Num	Other	All
HieCoAtt [18]	71.80	36.53	46.25	54.57
MCB [6]	77.37	36.66	51.23	59.14
Ours	**77.45**	**38.46**	**51.76**	**59.67**

Table 3. Our results on VQA 2.0 [8] validation set when trained on the training set only. Our model achieves an overall accuracy of 59.67% which marginally outperforms state of the art on this dataset.

[3] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473, 2014. 2

[4] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman. Return of the devil in the details: Delving deep into convolutional nets. In *BMVC*, 2014. 2

[5] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009. 2

[6] 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. In *EMNLP*, 2016. 1, 2, 4, 5, 6, 7

[7] X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. In *AISTATS*, 2010. 2, 5

[8] 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. 1, 2, 3, 7

[9] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016. 1, 2, 3, 4, 5

[10] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *CoRR*, abs/1207.0580, 2012. 2

[11] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 9:1735–1780, 1997. 2

[12] J. Johnson, A. Karpathy, and L. Fei-Fei. Densecap: Fully convolutional localization networks for dense captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4565–4574, 2016. 1

[13] J.-H. Kim, S.-W. Lee, D. Kwak, M.-O. Heo, J. Kim, J.-W. Ha, and B.-T. Zhang. Multimodal residual learning for visual qa. In *Advances in Neural Information Processing Systems*, pages 361–369, 2016. 6

[14] D. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 2, 4, 5

[15] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012. 1, 2

[16] Y. LeCun, K. Kavukcuoglu, and C. Farabet. Convolutional networks and applications in vision. In *ISCAS*, 2010. 2

[17] T.-Y. Lin, M. Maire, S. J. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014. 3

[18] 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*, pages 289–297, 2016. 2, 6, 7

[19] J. Mao, J. Huang, A. Toshev, O. Camburu, A. L. Yuille, and K. Murphy. Generation and comprehension of unambiguous object descriptions. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 1

[20] V. Mnih, N. Heess, A. Graves, and K. Kavukcuoglu. Recurrent models of visual attention. In *NIPS*, 2014. 2

[21] V. Nair and G. E. Hinton. Rectified linear units improve restricted boltzmann machines. In *ICML*, 2010. 2, 4

[22] H. Nam, J.-W. Ha, and J. Kim. Dual attention networks for multimodal reasoning and matching. *CoRR*, abs/1611.00471, 2016. 1, 2, 6

[23] H. Noh and B. Han. Training recurrent answering units with joint loss minimization for vqa. *CoRR*, abs/1606.03647, 2016. 6

[24] H. Sak, A. W. Senior, and F. Beaufays. Long short-term memory recurrent neural network architectures for large scale acoustic modeling. In *INTERSPEECH*, 2014. 2

[25] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 5

[26] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–9, 2015. 1, 2

[27] 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*, pages 2818–2826, 2016. 2

[28] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan. Show and tell: A neural image caption generator. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 1, 2

[29] Q. Wu, P. Wang, C. Shen, A. R. Dick, and A. van den Hengel. Ask me anything: Free-form visual question answering based on knowledge from external sources. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 6

[30] C. Xiong, S. Merity, and R. Socher. Dynamic memory networks for visual and textual question answering. In *ICML*, 2016. 6

[31] K. Xu, J. Ba, J. 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. In *ICML*, 2015. 1, 2

[32] Z. Yang, X. He, J. Gao, L. Deng, and A. Smola. Stacked attention networks for image question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 21–29, 2016. 1, 2, 4, 5, 6