# A Survey of Three State-of-the-art VQA Models

Zhaoyi Zhang
zzhang825@wisc.edu

Richard Tang
jtang79@wisc.edu

Yiwu Zhong
yzhong52@wisc.edu

## Abstract

*Visual Question Answering (VQA) has gained increasing attention in recent years. Despite of major progress made in the VQA methods, existing methods are still far from the actual human intelligence. In this survey, we investigate the recent development of VQA methods, visualize the qualitative outputs from the milestone VQA models to analyze the model capacity, summarize the bottlenecks for future research including improving attention mechanisms and answering counting, common sense, and text reading questions, and summarize the common techniques used in VQA methods such as using better visual features and attention mechanisms.*

## 1. Introduction

As human, we can effortlessly to answer questions about a given image. However, it's challenging for AI models to mimic this behavior, since, to provide the correct answer, these models have to understand both the image content and the question. Thus, the VQA (Visual Question Answering) challenge [2] has been proposed to seek potential solutions from researchers. This challenge has received great attention from two major deep learning communities: Computer Vision and Natural Language Processing.


Figure 1: Examples of VQA. Image source: [5].

Consider if we have a successful VQA model which can understand the image content correctly and communicate the visual information efficiently to humans, there will be numerous practical applications. One application is to help visually impaired and blind people to obtain information from their surrounding environments and images they encounter online. Another application is to summarize and help to make decisions based on a large amount of surveillance data.

In recent years, various models were developed to improve the VQA performance and major progress has been made. To better understand how and why current state-of-the-art VQA models work well and identify the typical model design, we look closely into three representative models: Bottom-Up and Top-Down (BUTD) [1], Bilinear Attention Networks (BAN) [8], and Modular Co-Attention Networks (MCAN) [20].

Specifically, our project consists of the following steps. First, we survey on the VQA papers and dive into the model details. By reading papers that proposed these models, we find that (1) all three models use image features extracted by the bottom-up attention model, (2) use different attention mechanisms and different model architectures to focus on particular image regions and question words. Second, we evaluate the model performance on VQA-v2 [5]. Lastly, we investigate how the model work in details. By visualizing model attention weights on input images and inspecting the answer predictions for certain types of questions, we identify some common bottlenecks across all three models. For example, they are not good at answering questions related to counting objects, common sense, and reading texts in the images.

## 2. Related Work

Benefited from the large-scale human-annotated dataset VQA [2] and its balanced version VQA-v2 [5], the task of visual question answering has attracted great attention in last few years. Given an input image and a question in natural language, the model learns to select the correct answer among the answer candidates. It is thus a classification problem based on the understanding on both vision domain and the language domain.

A typical VQA model consists of four steps: 1) extracting visual features from input image, 2) extracting semantic

features from the input question sentence, 3) fusing the features of visual modality and natural language modality, 4) predicting the answer based on the fused gigantic feature.

At the early stage [21, 2, 9], the image and question are first represented as global features (*e.g.*, CNN and LSTM) followed by a multimodal fusion module to predict the answer. The main limitation of using global feature representation is that the model might lose the critical information to correctly answer the question (*e.g.*, "Who is wearing the glasses"). To this end, Anderson et al. [1] (BUTD) proposed to use the detected regions from an object detector to represent the image content and to use attention mechanism to focus on the important regions conditioned on the given question. BUTD brought a major improvement on VQA, attributed to better visual representation (region features vs. global image features) and attention mechanism (*e.g.*, focus on the image regions relevant to the input question). Beyond the attention on image regions, BAN [8] and MCAN [20] developed the co-attention mechanism to learn the attention on both the image regions and the question words. Most recently, the visual-language pre-training models [12, 17, 10, 3] further boosted the VQA performance by pre-training Transformer [18, 4] on the large-scale image-text pairs.

In this project, we focus on evaluating the representative models BUTD, BAN and MCAN on VQA-v2 datasets. These models were only trained on VQA-v2, unlike the vision-language models pre-trained on image-text pairs.

## 3. Proposed Method

### 3.1. BUTD

We choose the Bottom-Up and Top-Down Attention (BUTD) method to serve as a baseline for the VQA models to mimic the human visual system when trying to complete a certain task. The BUTD method is split into two steps: the attention mechanisms driven by non-visual or task-specific context as "top-down" and purely visual feed forward attention mechanism as "bottom-up".

The VQA model for BUTD used a multimodal embedding with a mixture of the question and image features. First, they use tanh as activation, as a hidden status of the gated recurrent unit (GRU). Then the output is generated by a multi-label classifier operating over a fixed set of candidate answers. It generates answer and non-standardized weigh attention according to GRU, then calculate the standardized weigh attention and attention features.

### 3.2. BAN

BAN introduces the idea of bilinear attention, which considers every pairs of input channels (e.g. image regions and question words), to improve the existing co-attention mechanism. The co-attention networks can predict visual and textual attention distributions, and attend to particular image regions and questions words at the same time. By doing so, co-attention networks save computation time but ignore the potential interaction between images regions and words.

Low-rank bilinear pooling is used to computes the joint feature representations of every pairs of input channels, and bilinear attention is used to focus on particular input pairs.

**Low-rank bilinear model.** A scalar output $\mathbf{f_i}$ is computed as follows:

$$\mathbf{f}_i = \mathbf{x}^T \mathbf{W}_i \mathbf{y} \approx \mathbf{x}^T \mathbf{U}_i \mathbf{V}_i^T \mathbf{y} = \mathbb{1}^T \left( \mathbf{U}_i^T \mathbf{x} \circ \mathbf{V}_i^T \mathbf{y} \right) \quad (1)$$

where $\mathbb{1} \in \mathbf{R}^d$ is a vector of ones, and $\circ$ is Hadamard product. The bilienar weight matrix $\mathbf{W}$ is replaced with two smaller matrices $\mathbf{U}_i \mathbf{V}_i^T$.

**Low-rank bilinear pooling.** A vector output $\mathbf{f}$ is computed as follows:

$$\mathbf{f} = \mathbf{P}^T \left( \mathbf{U}^T \mathbf{x} \circ \mathbf{V}^T \mathbf{y} \right) \quad (2)$$

where $\mathbf{f} \in \mathbb{R}^d, \mathbf{P} \in \mathbb{R}^{d \times c}, \mathbf{U} \in \mathbb{R}^{N \times d}, \mathbf{V} \in \mathbb{R}^{M \times d}$. The pooling matrix $\mathbf{P}$ is introduced to allow $\mathbf{U}$ and $\mathbf{V}$ to only be 2-dimensional matrices, and thus greatly reduce the number of parameters.

**Bilinear attention map.** The attention map $\mathcal{A}$ is defined as follows:

$$\mathcal{A} := \text{softmax} \left( \left( \left( \mathbb{1} \cdot \mathbf{p}^T \right) \circ \mathbf{X}^T \mathbf{U} \right) \mathbf{V}^T \mathbf{Y} \right) \quad (3)$$

where $\mathcal{A} \in \mathbf{R}^{\rho \times \phi}, \mathbb{1} \in \mathbf{R}^{\rho}, \mathbf{p} \in \mathbf{R}^K$, and the softmax is element-wise softmax.

**Bilinear attention networks.** To generalize the low-rank bilinear model to multi-channel inputs, $\mathbf{X} \in \mathbb{R}^{N \times \rho}$ and $\mathbf{Y} \in \mathbb{R}^{M \times \phi}$, a vector output $\mathbf{f}_k$ is computed as follows:

$$\mathbf{f}'_k = \left( \mathbf{X}^T \mathbf{U}' \right)_k^T \mathcal{A} \left( \mathbf{Y}^T \mathbf{V}' \right)_k \quad (4)$$

where $\mathbf{U}' \in \mathbb{R}^{N \times K}, \mathbf{V}' \in \mathbb{R}^{M \times K}, (\mathbf{X}^T \mathbf{U}')_k \in \mathbb{R}^{\rho}$, $(\mathbf{Y}^T \mathbf{V}')_k \in \mathbb{R}^{\phi}$, and $\mathbf{f}'_k$ is the $k$-th column in $\mathbf{f}'$. Then, the bilinear joint representation $\mathbf{f}$ is computed as follows:

$$\mathbf{f} = \mathbf{P}^T \mathbf{f}' \quad (5)$$

where $\mathbf{f} \in \mathbb{R}^C$ and $\mathbf{P} \in \mathbb{R}^{K \times C}$.

**Classifier.** Finally, a two-layer multi-layer perceptron is used to classify the joint feature representation $\mathbf{f}$.

### 3.3. MCAN

As previous methods, MCAN use the detected region features to represent the image content and use word embedding and LSTM to represent each word in the question. On top of that, MCAN proposed co-attention mechanism with Transformer architecture to fuse the information from visual and language domain. Finally, based on the fused features, MCAN predict the final answer with a regular classification layer.

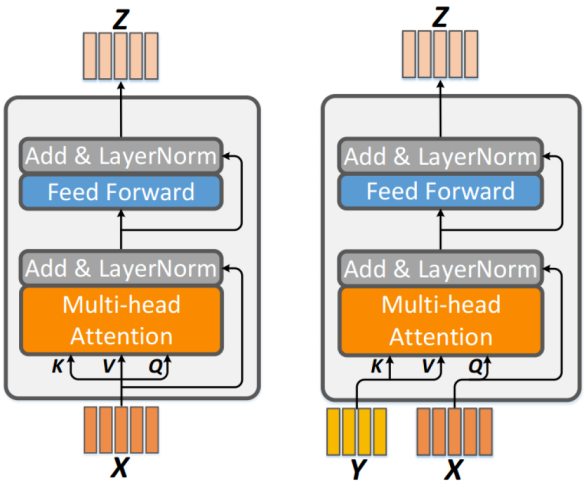


Figure 2: The co-attention mechanism proposed in MCAN. Image source: [20] Left: the self-attention mechanism in Transformer. Right: the proposed co-attention mechanism for the interaction between visual domain and language domain.

The key idea of MCAN is to embed the interaction between the visual inputs and language inputs into each layer of the model. Unlike previous methods that fuse the visual information and language information at the last few layers of the model, MCAN adopts the Transformer architecture and integrate the co-attention mechanism in each intermediate layer of Transformer. The details of co-attention mechanism is shown in Figure 2. Specifically, the visual information, denoted as $Y$, serve for the key vector and the value vector, and the language information, denoted as $X$, serve for the query vector. This special design establishes the cross-modality interaction and thus potentially help to fuse the information.

# 4. Experiments

## 4.1. Dataset

We use the most commonly used dataset in VQA task: VQA-v2 [5]. It is publicly available at `https://visualqa.org/download.html`. In particular, we only used Balanced Real Images, which contains 204,721 images from COCO dataset [11] and 1,105,904 open-ended questions on those images. See the details of the VQA-v2 Balanced Real Images dataset in Table 1.

## 4.2. Preprocessing

Images features of COCO images are extracted by the bottom-up attention model, which contains Faster R-CNN [15] and ResNet-101 [6]. Faster R-CNN is used to detect objects of certain classes and localize them with bounding boxes in the input images. All regions where any object detection probability is higher than a threshold are selected. For each selected region, the image feature is the 2048-dimensional mean-pooled convolutional feature extracted from this region using ResNet-101.

| | Images | Questions |
|---|---|---|
| Training | 82,783 | 443,757 |
| Validation | 40,504 | 214,354 |
| Testing | 81,434 | 447,793 |

Table 1: The number of images and questions contained in training, validation, testing set of the Balanced Real Images in VQA-v2.

Questions embeddings are obtained using pre-trained GloVe [14] word embeddings. The language features are extracted from the questions embeddings using either GRU or LSTM depending on the model at run time.

## 4.3. Software

We use PyTorch [13] and OpenVQA [19] as our frameworks to run experiments. PyTorch is an open source machine learning framework. OpenVQA is a framework for visual question answering built on top of PyTprch. It implements some of state-of-the-art VQA models and contains three benchmark datasets.

## 4.4. Hardware

To accelerate the attention weight and prediction computation on the dataset, we use the free GPU resources (Nvidia K80s, T4s, P4s and P100s) provided by Google Colaboratory.

## 4.5. Implementation Details

To save time, we download the bottom-up attention features and pre-trained models provided by OpenVQA and GloVe word embeddings from spacy [7]. Since all the pre-trained models are pre-trained on training and validation set, to evaluate the performance of models on unseen data, we compute the attention weights for each bounding box in each input image and the answer predictions on each image and question pair in the test set. Since BAN has 8 attention maps that have attention weights for each pair of bounding boxes and question words, for the visualization purposes, we sum the attention weights for each bounding box, and take the average of 8 maps. The same post-processing is also applied to the attention weights predicted by MCAN.

The code for the computation mainly comes from OpenVQA source code. The attention weights are saved in Python numpy npz format, and the predictions are saved in json format for later visualization and analysis. After doing the computation and saving the results, we obtain three npz

files and three json files. The test set does not have publicly available ground truth labels, so we submit the prediction files to the evaluation server to obtain the accuracies.

## 5. Results and Discussion

In this section, we compare 3 representative VQA models: BUTD, BAN and MCAN. (1) We quantitatively evaluate the models on the VQA benchmark and report the accuracy results. (2) We investigate the model behavior qualitatively on 3 aspects, including: the attention quality, the questions that require model to count the image objects, the questions that require model to infer common sense knowledge. We now present the details in the following sections.

### 5.1. Benchmark Evaluation

Table 2 presents the model accuracy on VQA-v2 dataset. BAN and MCAN have very close overall accuracy while BUTD has slightly lower performance.

On visual side, all models use the same region features from the bottom-up attention model. But BAN and MCAN exploit the co-attention mechanism which connects each image region and each word in input question. We conjecture that the performance difference comes from various attention mechanisms (*e.g.*, the attention over image regions vs. the attention over image regions and question words) and unique model design (*e.g.*, BAN used bilinear pooling vs. MCAN used self-attention of Transformer). However, by only looking at the accuracy can not bring us much insight. To this end, we mainly focus on the qualitative results of models on different aspects, as shown in the next few sections.

| | Yes/No | Number | Other | Overall |
|---|---|---|---|---|
| BUTD | 83.28 | 46.21 | 58.11 | 67.13 |
| BAN | 85.2 | 49.67 | 60.01 | 69.21 |
| MCAN | 86.9 | 52.14 | 60.97 | 70.65 |

Table 2: Model performance on VQA benchmark

### 5.2. Attention visualization

By visualizing attention weights, we find that even though complex and different attention mechanisms are used in all three models, the models do not always attend to the correct part of the image.

For the question "What are people sitting next to" in Figure 3, all three models predict the answers that are relevant to the image content, but the only answer that is intuitively correct is "tree", predicted by BAN. However, all three models predicted high attention weights for the bounding boxes of the bench and the couple. In particular, BAN focuses on the bench instead of the tree, but gives the correct answer, which could be a result of BAN associates brown and green colors in the image with trees. For this question, we think the models capture the meaning of "sitting" but fail to understand the phrase "next to", and as a result, focus on the wrong objects.

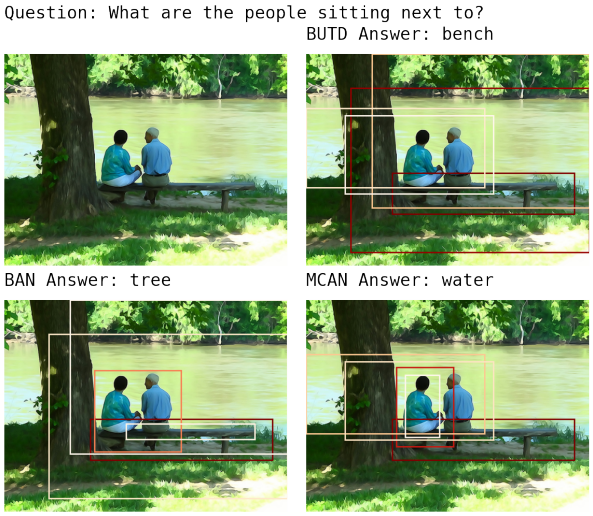


Figure 3: An example of attending to the wrong image regions. The correct answer should be "tree". Only top 5 highest attention weights are visualized. The darker the color of the bounding box, the higher the attention weight it has.

For the question "What figures are on the front of the bus" in the Figure 4, both BUTD and MCAN focus on the same, but totally wrong object, which is the dog, and predict answers of "dog" and "animal". BAN gives higher weight for the figures on the front of the bus than the dog, but still fails to give a more reasonable answer.

We find that there are cases where the model attends to the wrong object and gives an answer that are related the wrong object. In this case, the model can fail to give the correct answer.

### 5.3. Answer prediction

Now we investigate how the models perform on certain types of questions. Here we consider 3 most common question types: counting questions, common sense questions, and text reading questions.

**Counting questions.** Figure 5 and Figure 6 show the prediction results for counting questions. Most predictions are the wrong answers (*e.g.*, "How many smokestacks are in the background", "How many munchkins are on the plate"). According to the observation of hundreds examples of counting questions, we found that the existing VQA models lack the ability to explicitly count the objects in the images. We conjecture that the models just predict a random number when asked a counting question. These ran-

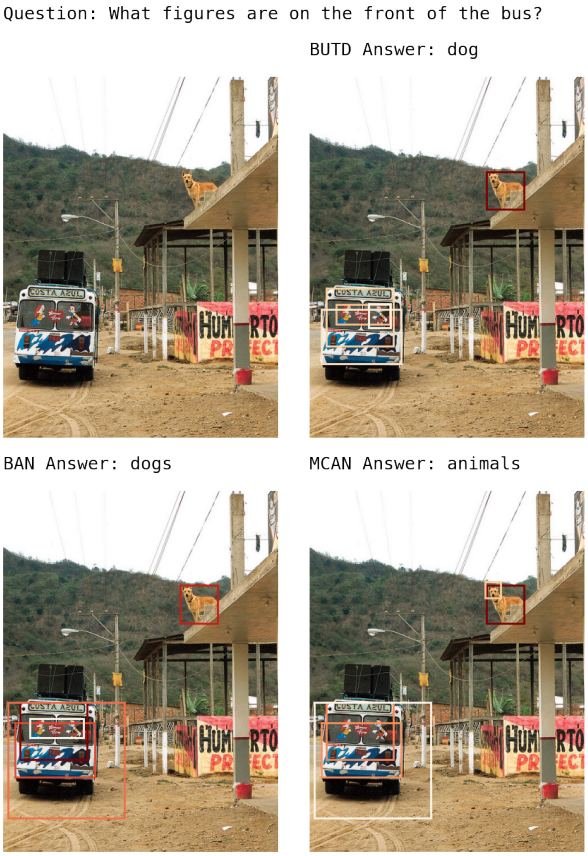


Figure 4: An example of attending to the wrong image regions and thus predicting the wrong answer. The correct answer could be "cartoon figures".

dom numbers highly rely on the bias in the training dataset.

**Common sense questions.** Then we use these VQA models to answer common sense questions in Figure 7 and Figure 8. It is not hard to say that all three VQA models do not perform well on answering common sense questions. For example, in Figure 7, it is obvious that the man is looking at himself in the mirror and tying a tie. However, the model predictions such as "taking picture", "working", and "talking on phone" are totally wrong but somehow related to the scene. In Figure 8, it is also obvious that the blue fruit that the question is asking for is blueberry. BUTD and MCAN answer with a red fruit which strawberry is not in the picture. We think that all three models associate certain elements in the image with particular phrases during the train phase.

**Text reading questions.**Another problem we encounter is that all the models have a hard time extracting certain text or figures in the image. Questions like "What is the number on the man's back" or "What is the title of this book" cannot be answered correctly by most of the models, as shown in Figure 9.

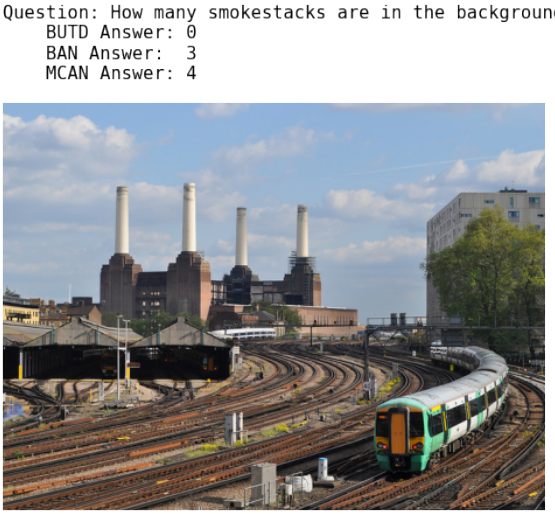


Figure 5: An example of model predictions for the counting question. The correct answer should be 4.

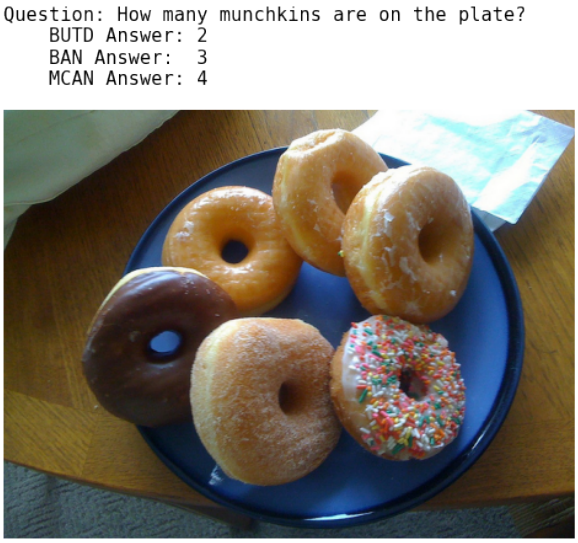


Figure 6: An example of model predictions for the counting question. The correct answer should be 6.

### 5.4. Bottleneck Summary

According to the visualization in the previous sections, there are 4 identified bottlenecks in current VQA models. (1) Most recent VQA models apply the attention mechanism yet don't necessarily look at the correct regions when answering the question. This type of deep model is not interpretable to human. A potential way to improve the interpretability is to force the models to attend to the correct regions during training. (2) The existing models always fail to answer the counting questions. This suggests that the cur-

Question: What is the man doing?
BUTD Answer: taking picture
BAN Answer: working
MCAN Answer: talking on phone

Figure 7: An example of model predictions for the common sense question. The correct answer should be "tying a tie".

Question: What is the blue fruit?
BUTD Answer: strawberry
BAN Answer: kiwi
MCAN Answer: raspberry

Figure 8: An example of model predictions for the common sense question. The correct answer should be "blueberry".

rent models don't have the actual reasoning ability. They just randomly guess a number as the answer. The answer is likely the bias in the training data. Developing a counting module to explicitly count the number of detected regions may be helpful. (3) The models are not equipped with the common sense knowledge. This is mostly limited by the dataset itself. Because during the training, no explicit common sense knowledge is annotated in the dataset. Hence, additional annotation of necessary common sense knowledge for each question can help to address this problem. (4) The models can not even read the simple text in the images. The cause could be the design of models that they infer the

Question: What number is the batter?
BUTD Answer: 25
BAN Answer: 27
MCAN Answer: 11


Figure 9: An example of model predictions for problem required reading texts in the image. The correct answer should be "12".

answer based on the visual features. However, these visual features don't provide exact shape of the text and thus the models fail to read the texts. A simple approach to enable text reading can be designing an additional module to extract the text inside the detected bounding boxes, such as the mature OCR (Optical Character Recognition) techniques.

### 5.5. Common Techniques in VQA

According to our survey on VQA models and the experiments on the representative models, we found 2 common techniques for improving the performance of VQA models: better visual representation, and better interaction between visual modality and language modality.

Most recent VQA models make use of the region features proposed by BUTD to represent the image content. These region features are extracted from a object detector [15]. The region features from this particular object detector benefits the VQA models better than the other object detectors and the global image features from CNN. By a closer inspection, we found that the region features from BUTD were trained by not only object annotation but also the attribute annotation. Hence, we conjecture that these region features contain more detailed information about the image objects, such as the attributes of the objects (*e.g.*, "yellow banana"). These attributes of objects can be helpful for answering the questions.

Another common technique in the recent VQA models is the attention mechanism. BUTD learned paying more attention to the objects that are important to the given question. This attention mechanism can be considered as the interaction between each image region and the whole question sentence. Beyond that, BAN and MCAN proposed

co-attention, that is, the interaction between each image region and each word in the question. Co-attention requires more computation yet provides larger capacity to the models. That might be the reason why BAN and MCAN can both outperform BUTD, as shown in Table 2. Also, MCAN used Transformer architecture which already contains the self-attention mechanism in each layer. The additional attention weights might give MCAN an advantage over BAN.

## 6. Conclusions

In this project, we aim to identify common characteristics of the current state-of-the-art VQA models, as well as their shortcomings. By selecting and analyzing three representative VQA models, we successfully achieve our goal.

We summarize 2 common techniques used in VQA models. 1) Better visual features. The BUTD model proposed to use the region features from object detector to represent the image content. This provides much more details about the images and can help boost performance. 2) The attention mechanism. The attention can help the model to focus on different parts of the images and the questions. This provides more flexibility to the models.

By inspecting the attention weights visualization, we find that models sometimes cannot predict the correct attention weights based on visual features and language features. The reason can be that the VQA models do not fully understand the meaning of the question or know the classes of objects present in different image regions.

With the doubt of whether VQA models can answer questions that are easy for human, we examine the answer predictions on counting and common sense questions. We find that VQA models generally predict random but wrong answers that are related to the question. Thus, VQA models are terrible at answering those two type of questions, which means that the current model don't have the actual reasoning ability, and random answers reflect the bias in the training set.

Therefore, we think if we want to do further researches, one direction could be comparing three models we selected with VQA models that perform well on TextVQA [16], which is a dataset that involves visual reasoning based on texts in images. Another direction could be investigating VQA models that can answer counting questions correctly.

## 7. Acknowledgements

We would like to thank the authors of OpenVQA [19] for their great open source package and providing pre-trained models and VQA-v2 dataset.

## 8. Contributions

Before the midterm, everyone was responsible for reading papers about VQA and understanding a different VQA model. Zhaoyi picked BAN, Richard picked BUTD, and Yiwu picked MCAN. After the midterm, we started to run the experiments and met weekly (more frequent towards the deadline) to discuss our progress and plan next steps. Everyone ran the computation of attention weights and predictions for their own model, and Zhaoyi wrote the code for result visualization. Lastly, everyone wrote about their own models and other different parts but equal amount of the final report.

## References

[1] 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 *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6077–6086, 2018.

[2] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. L. Zitnick, and D. Parikh. Vqa: Visual question answering. In *Proceedings of the IEEE international conference on computer vision*, pages 2425–2433, 2015.

[3] Y.-C. Chen, L. Li, L. Yu, A. El Kholy, F. Ahmed, Z. Gan, Y. Cheng, and J. Liu. Uniter: Universal image-text representation learning. In *European Conference on Computer Vision (ECCV)*, pages 104–120. Springer, 2020.

[4] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[5] 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. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[6] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 770–778. IEEE Computer Society, 2016.

[7] M. Honnibal, I. Montani, S. Van Landeghem, and A. Boyd. spaCy: Industrial-strength Natural Language Processing in Python, 2020.

[8] J. Kim, J. Jun, and B. Zhang. Bilinear attention networks. *CoRR*, abs/1805.07932, 2018.

[9] J.-H. Kim, S.-W. Lee, D.-H. Kwak, M.-O. Heo, J. Kim, J.-W. Ha, and B.-T. Zhang. Multimodal residual learning for visual qa. *arXiv preprint arXiv:1606.01455*, 2016.

[10] X. Li, X. Yin, C. Li, P. Zhang, X. Hu, L. Zhang, L. Wang, H. Hu, L. Dong, F. Wei, et al. Oscar: Object-semantics aligned pre-training for vision-language tasks. In *European Conference on Computer Vision (ECCV)*, pages 121–137. Springer, 2020.

[11] 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.

[12] J. Lu, V. Goswami, M. Rohrbach, D. Parikh, and S. Lee. 12-in-1: Multi-task vision and language representation learning. In *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[13] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

[14] J. Pennington, R. Socher, and C. Manning. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar, Oct. 2014. Association for Computational Linguistics.

[15] 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.

[16] A. Singh, V. Natarjan, M. Shah, Y. Jiang, X. Chen, D. Parikh, and M. Rohrbach. Towards vqa models that can read. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8317–8326, 2019.

[17] H. Tan and M. Bansal. Lxmert: Learning cross-modality encoder representations from transformers. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2019.

[18] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 30, 2017.

[19] Z. Yu, Y. Cui, Z. Shao, P. Gao, and J. Yu. Openvqa. `https://github.com/MILVLG/openvqa`, 2019.

[20] Z. Yu, J. Yu, Y. Cui, D. Tao, and Q. Tian. Deep modular co-attention networks for visual question answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6281–6290, 2019.

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