Visual Question Answering in Distributed Setting

Souvik Karmakar(14700) , Dinesh Kumar (14428)
Advisor:- Prof. Partha Pratim Talukdar

***Abstract*—Given a picture and Natural Language question about the image, the errand is to give a precise Natural Language answer. Reflecting genuine situations, for example, helping the visually challenged, question and answer both are open-ended.Visual questions specifically target diverse regions of a picture, that contains the details of the background and hidden settings.**

I. Introduction

THE undertaking of Visual Question Answering (VQA) includes a picture and a related natural language question, to that question the machine must give the correct answer. This Visual Question Answering is the task that lies at the intersection of the many fields such as Natural Language processing, Computer Vision and artificial intelligence. This task can be pictorially seen in the figure shown below that was taken from the paper [1].

What is on the coffee table ?

candles

What color is the hydrant ?

black and yellow

Fig. 1: The VQA relates the picture and the natural language question.Examples of training questions and their correct answer from the VQA v2 dataset [2]

So the basic task is that an image and a question is given and we have to answer the question based on the contents of the image. But sometimes it happens that the some of the images in the dataset may not be associated with the question related to it, then in that case the processing time is invested in those images that are not even relevant to the questions asked.

Hence the problem is addressed as the two sub-problems that are as follows, firstly we will check the relevance of the question to the image then, secondly will answer the questions for the relevant images.

As we can see in the below figure taken from paper [3],the question is **What color is the backpack ??** and the first image is the relevant one as the $premise$ of the question is $backpack$ which is absent in the second image, hence is irrelevant to the question.

Fig. 2: Examples of QRPE dataset showing the relevance of image on the basis of premise.

The model is based on the principle of a joint embedding of the image and the question,which is passed over a multi-label classifier.This is the general approach followed by many modern VQA Methods [4].

II. Background

The errand of VQA has accumulated expanding enthusiasm for the recent years since the original paper of Antol et al.[5].Eventhough this work lies at the intersection of the many fields stated in the above section but still it has primarily focused on the field of the Computer Vision, as for answering the question it should have deep understanding of the visuals. There are other tasks that are related to the vision and language such as image captioning [6,9] and visual dialog [8] are getting attention now-a-days.

Datasets:- From this survey we can see that the large number of dataset had been created for the task of VQA [7]. Each dataset contains different pictures, ordinarily from Flickr or potentially from the COCO dataset [10], together with human-proposed questions and ground truth answers. The VQA dataset that was proposed in 2015 by Antol et al.[5] was benchmark.

Reality that hinders the compelling assessment and examination of contending techniques. The perception prompted the presentation of another rendition of the dataset, alluded to as VQA v2 [11]. Another dataset utilized is the Visual Genome [12]. This multipurpose dataset contains comments of pictures as scene charts. Those comprise fine-grained portrayals of the picture contents. They give an arrangement of visual components showing up in the scene (e.g.objects, people), together with their characteristics (e.g. color,appearance) and the relations between them. Annotations were not used directly but for training the Faster R-CNN model[14] they were served in [13] used for obtaining object-centric features. We are simply gonna use Visual Genome dataset as in this questions are directly related to the images.

Methods:- VQA depends on three segments. (1) proposing question anwering as an classification problem, unraveled with (2) a profound deep neural system that actualizes a joint embedding model, (3) prepared end-to-end with supervision of precedent inquiries/answers. To begin with, question answering is posed as classification over a set of applicant answers, questions are generally visual in nature and the right answers subsequently just range a little arrangement of words and expressions.

Second, most VQA models depend on a profound neural system that actualizes a joint inserting of the picture and of the inquiry.The two inputs image and the question is mapped into the fixed-size vector representations and this is done with CNN and RNN respectively.Further non-linear mappings of those portrayals are typically translated as projections into a joint "semantic" space.Then further its been concatenated using the element-wise multiplication.

Third, we train the whole neural network as end-to-end from questions,images and its ground truth answers. Most of the VQA methods are based on the complex attention mechanism [4], and recent studies shown that careful implementation and selection of hyperparameters leads to the better performance [15,16].

III. Baseline Model

The baseline model can be seen in the figure taken from paper [1]. This baseline model can be summarized as the implementing the joint CNN for image and RNN embeddings of the question ,with the attention for the images those are taken by considering the questions. Now gonna explain the each part of the model in the below subsections.

A. Question Embeddings

During training and testing each instance of the input is an image and a text question.Questions are then split into words i.e. tokenised and for computational efficiency questions are taken upto 14 words rest all are discarded (very few questions contains words more than 14). each word is represented in a vector form with 300-d and these vectors are initialised with the GLoVe word embeddings pretrained on the Wikipedia corpus.Padding is used for the questions with words less than 14, finally GRU is fed with embeddings of size $14X300$. Internal state of the GRU is of dimension 512 which is used as the final state of the question embedding q.

B. Image Features

On passing the image as an input to the Faster R-CNN framework, the output which we get is a vector of size $KX2048$, where K is the number of image locations and each of these locations are represented by 2048-d vector which encodes the representation of the image for that particular region.

C. Image Attention Multimodal fusion

Traditional question-guided attention mechanism is implemented which is usually used in most of the VQA models.For every location i= 1 ... K, the question embedding q and image feature vector v_{i} are concatenated and passed over a non-linear layer f and a linear layer to get a scalar attention weight $\alpha_{\mathrm{i,t}}$ associated with the location specified. mathematically it is shown below.

$$a_i = \boldsymbol{w}_a f_a([\boldsymbol{v}_i, \boldsymbol{q}]) \quad (1)$$

$$\boldsymbol{\alpha} = \mathrm{softmax}\,(\boldsymbol{a}) \quad (2)$$

$$\hat{\boldsymbol{v}} = \Sigma_{i=1}^{K} \alpha_i \boldsymbol{v}_i \quad (3)$$

where w_{a} is learned parameter vector.This features taken from all the locations is normalized and summed to obtain a vector of size 2048 that vector $\hat{v}$ represents the image that had been attended. Now the simple Hadamard product is used for combining the outputs of the non-linear layers when fed with the question (q) and image ($\hat{v}$) representations. mathematically its shown below the joint representation of the image and question which is given as an input to the output classifier.

$$\boldsymbol{h} = f_q(\boldsymbol{q}) \circ f_v(\hat{\boldsymbol{v}}) \quad (4)$$

D. Output Classifier

The vector h is passed through non-linear layer and then linear mapping to get the score of for every element of N candidates.

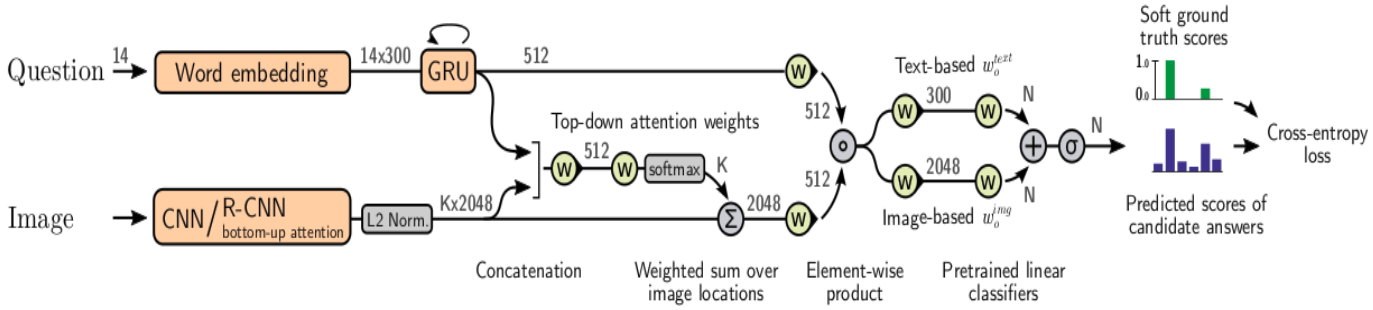

Fig. 3: Baseline Model architecture[1]

$$\hat{s} \;=\; \sigma\big(\, \boldsymbol{w}_o \;\; f_o(\boldsymbol{h})\,\big) \tag{5}$$

the last stage predicts the correctness of every candidate answer and can be viewed as a logistic regression, mathematically is shown below,

$$L \;=\; -\sum_{i}^{M}\sum_{j}^{N} s_{ij}\log(\hat{s}_{ij}) \;-\; (1-s_{ij})\log(1-\hat{s}_{ij}) \tag{6}$$

where s is the soft accuracy of ground truth answers. The pros of using this formulation is that for every question , multiple correct answers optimization is allowed and another advantage is compared to the binary signals the soft scores provides better training signals.

1) Scoring Method: We consider only the answers that occur more than 8 times in the combined set of validation and training answers. Next consider the training and validation answers separately, assigning soft scores according to the following criteria:

- If an answer occurs once in the training set(or validation), a score of 0.3 is assigned.
- If an answer occurs twice in the training set(or validation), a score of 0.6 is assigned.
- If an answer occurs thrice in the training set(or validation), a score of 0.9 is assigned.
- If an answer occurs more than 3 times, a score of 1.0 is assigned.

2) Non-linear Layers:: The Baseline model uses gated hyperbolic tangent activation in each non-linear layer. Mathematically equations are shown below,

$$\tilde{\boldsymbol{y}} \;=\; \tanh\left(W\boldsymbol{x}+\boldsymbol{b}\right) \tag{8}$$

$$\boldsymbol{g} \;=\; \sigma(W'\boldsymbol{x}+\boldsymbol{b}') \tag{9}$$

$$\boldsymbol{y} \;=\; \tilde{\boldsymbol{y}} \circ \boldsymbol{g} \tag{10}$$

where σ is sigmoid activation function, $W, W' \;\epsilon\; \Re^{n\times m}$ are learned weights and $\circ$ represents Hadamard product i.e. element wise product.

IV. Our Model

Initially we tried to implement the, baseline model in tensorflow, however our model failed to converge. Next, we replaced the gated tanh with gated linear unit. This improved the convergence to some extent. Next we removed the gated units completely with ReLU. Finally, we settled on LeakyReLU, with layer normalization[18]. We use Adam Optimizer with a fixed learning rate of 0.0001.

Later we have trained the model in Tensorflow distributed framework using various strategies such as Asynchronous , Drop Stale Synchronous and Horovod API (modified Ring AllReduce, with modifications on learning rate)[17]. We give an account of the time taken by various experiments in Table II. The accuracy of each experiment is reported in Table I.

V. Results and Analysis

We run all the experiments on 4 node 4 gpu cluster. Hence in all the experiments we use a single parameter server, with 2, 3 worker nodes. We only compute the validation score and loss at our chief worker.We had faced some latency issues while running our experiments on our lab cluster, due to overload. We make the following analysis of our results:

- **Asynchronous:** While running the experiment 3 workers, our worker node finished the experiment faster, than the other 2 workers. In case of the experiments with 2 workers, our chief worker finished the experiment much slower than the other worker node. Hence, in case of 3 workers we see that the loss converges, after more number of epochs. The chief worker in case of 2 workers, had the gradients accumulated from the other worker. Hence the loss converged faster. Ideally Asynchronous should have converged fastest given all the workers worked in the same time.
- **Drop Stale Asynchronous:** We have only run experiments on a 3 worker 1 parameter server setup. We make a similar observation in case when the staleness was 60, where the chief worker ran experiments faster than the other two workers. Hence we see convergence of loss after more number of epochs. In case of staleness 40 and 80, both the chief workers had taken a similar amount of time. When staleness is set to 80, less number of gradients are dropped and hence we see convergence in less number of epochs.
- **Horovod**: Horovod uses Ring-All-Reduce implementation[17], without any parameter servers. It also multiplies the learning rate with the number of workers. The model converges faster than both Asynchronous and Drop Stale optimizers. Ideally it should have converged at the same time as the baseline,

however maybe due to higher learning rate, the model converged in more number of epochs. However the overall performance gain in terms of time and accuracy is quite significant. We have also implemented Ring AllReduce without modification inn learning rate.

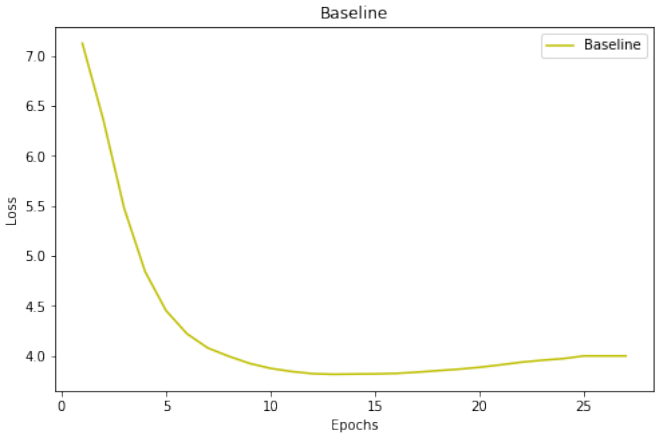

Fig. 4: Baseline Loss plot

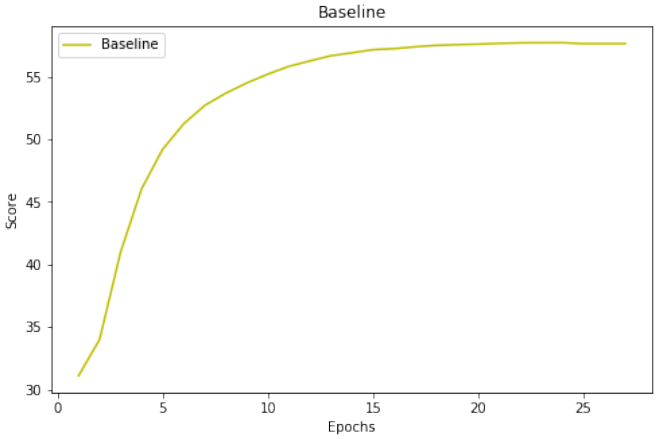

Fig. 5: Baseline Score plot

TABLE I: Scores of different distributed framework

Experiment	Score
Paper	63.15
Local(single-gpu)	57.55
Asynchronous (3 worker)	56.65
Asynchronous (2 workers)	57.12
Drop Stale 40 (3 worker)	57.26
Drop Stale 60 (3 worker)	56.67
Drop Stale 80 (3 worker)	56.41
Horovod (3 worker)	57.91
Ring All Reduce (3 worker)	56.45

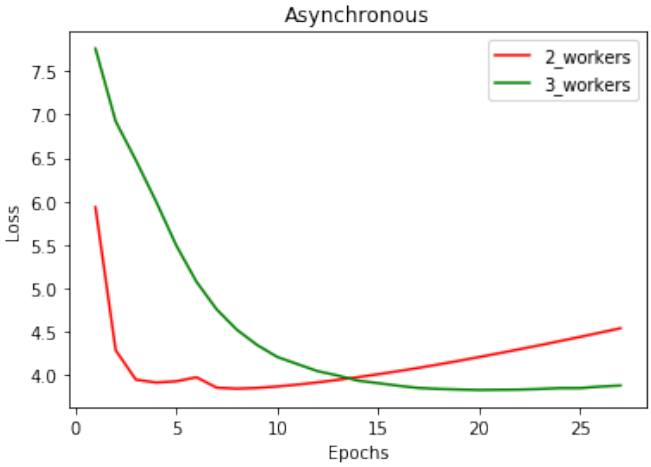

Fig. 6: Asynchronous Loss plot

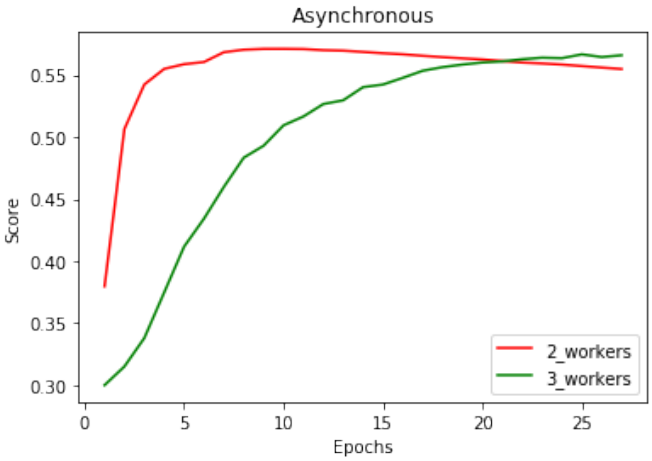

Fig. 7: Asynchronous Score plot

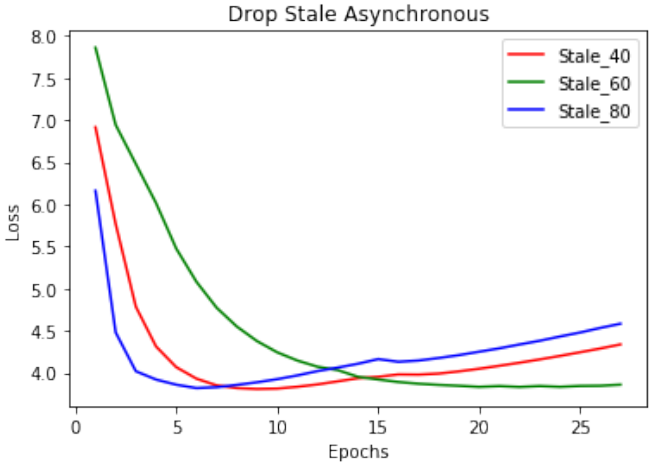

Fig. 8: Drop Stale Asynchronous Loss plot

VI. Conclusion

We have implemented the baseline model and then compared the results with the models implemented in the Distributed Framework which resulted in slighter lower score but fast implementation of the VQA in the distributed setting. The latency of the worker nodes resulted in the degradation in performance. For better performance the model could be fine tuned and that will the future work for this.

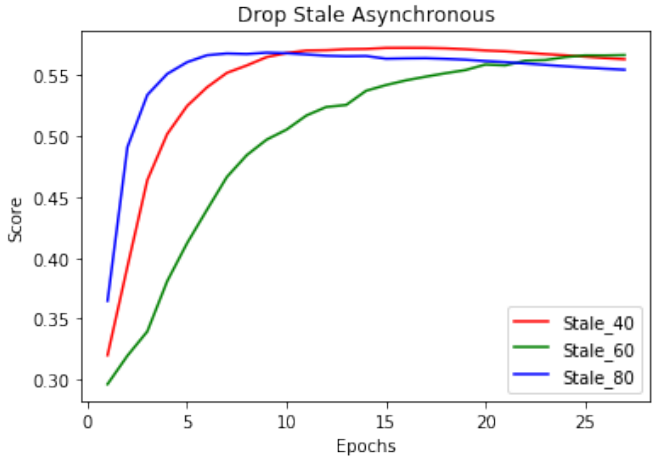

Fig. 9: Drop Stale Asynchronous Score plot

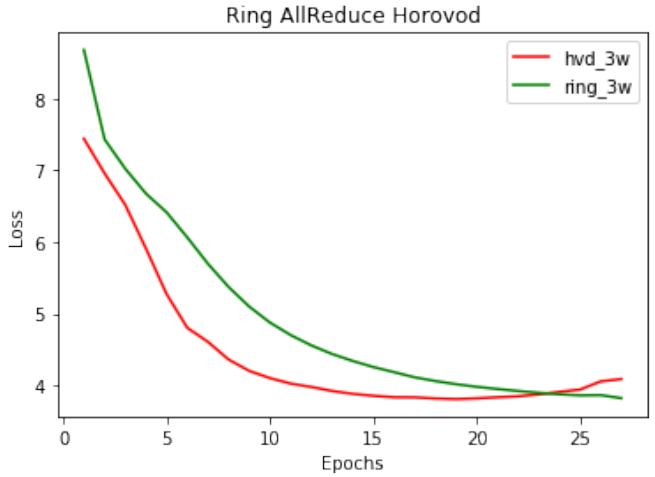

Fig. 10: Horovod Loss plot

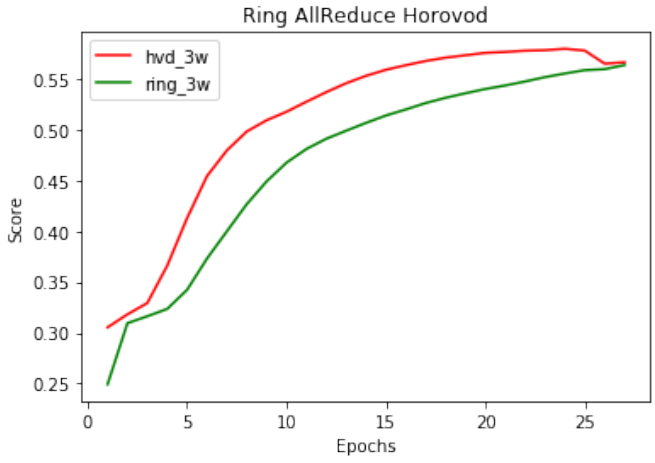

Fig. 11: Horovod Score plot

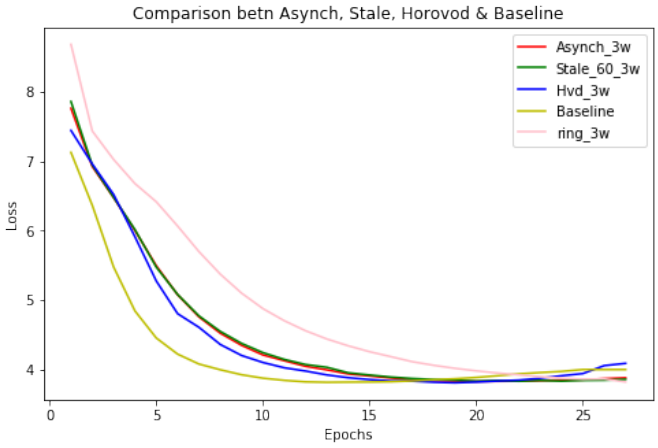

Fig. 12: All models Loss plot

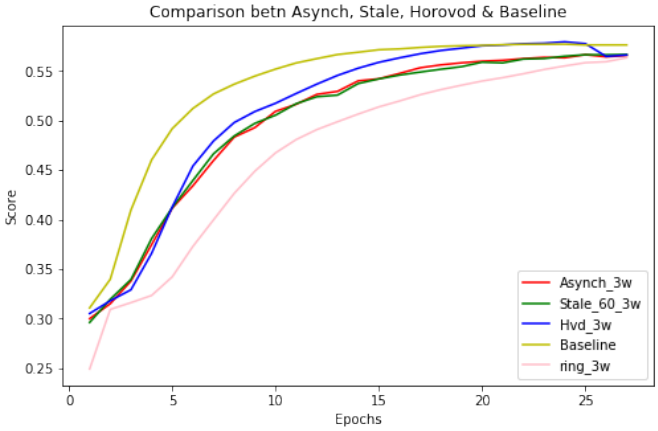

Fig. 13: All models Score plot

TABLE II: Runtime of different workers nodes in different distributed framework

Experiment	Worker 1	Worker 2	Worker 3
Local(single-gpu)	17680	—	—
Asynchronous(3 worker)	7164	13757	13423
Asynchronous(2 workers)	15850	10455	—
Drop Stale 40(3 worker)	14438	7459	7750
Drop Stale 60(3 worker)	7331	15142	16341
Drop Stale 80(3 worker)	15356	7248	7392
Horovod (3 worker)	12285	—	—
Ring All-Reduce (3 worker)	12377	—	—

REFERENCES

[1] Teney, Damien et al. "Tips and Tricks for Visual Question Answering: Learnings from the 2017 Challenge." CoRR abs/1708.02711 (2017): n. pag.

[2] 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. arXiv preprint arXiv:1612.00837, 2016.

[3] Mahendru, Aroma et al. âĂIJThe Promise of Premise: Harnessing Question Premises in Visual Question Answering.âĂİ EMNLP (2017).

[4] 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,2017

[5] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. L. Zitnick, and D. Parikh. VQA: Visual Question Answering.In Proc. IEEE Int. Conf. Comp. Vis., 2015.

[6] H. Fang, S. Gupta, F. Iandola, R. Srivastava, L. Deng,P. DollÃąr, J. Gao, X. He, M. Mitchell, J. Platt, et al. From captions to visual concepts and back. In Proc. IEEE Conf. Comp. Vis. Patt. Recogn., 2015.

[7] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan. Show and tell: A neural image caption generator. In Proc. IEEE Conf.Comp. Vis. Patt. Recogn., 2014.

[8] A. Das, S. Kottur, K. Gupta, A. Singh, D. Yadav, J. M. Moura, D. Parikh,

and D. Batra. Visual Dialog. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017.

[9] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan. Show and tell: A neural image caption generator. In Proc. IEEE Conf.Comp. Vis. Patt. Recogn., 2014.

[10] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ra- manan, P. DollÃąr, and C. L. Zitnick. Microsoft COCO: Common objects in context. In Proc. Eur. Conf. Comp. Vis., 2014.

[11] 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. arXiv preprint arXiv:1612.00837, 2016.

[12] 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.arXiv preprint arXiv:1602.07332, 2016.

[13] 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 vqa. arXiv preprint arXiv:1707.07998, 2017.

[14] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In Proc. Advances in Neural Inf. Process. Syst. 2015.

[15] A. Jabri, A. Joulin, and L. van der Maaten. Revisiting visual question answering baselines. 2016.

[16] V. Kazemi and A. Elqursh. Show, ask, attend, and answer: A strong baseline for visual question answering. arXiv preprint arXiv:1704.03162, 2017.

[17] Goyal, Priya, et al. "Accurate, large minibatch SGD: training imagenet in 1 hour." arXiv preprint arXiv:1706.02677,2017.

[18] Ba, Jimmy Lei, Jamie Ryan Kiros, and Geoffrey E. Hinton. "Layer normalization." arXiv preprint arXiv:1607.06450, 2016.