VisualQA – Deep Learning Methods For Making Images Talk

Bagchi, Deblin*, Kumari, Sangeeta†, Mahajan, Pravar‡ and Ghai, Piyush§
Department of Computer Science & Engineering, The Ohio State University
Columbus, OH 43202
Email: *bagchi.16@osu.edu, †kumari.14@osu.edu, ‡mahajan.89@osu.edu, §ghai.8@osu.edu

***Abstract*—VisualQA is the task of providing a natural language answer, when supplied with an image and a supporting question about that image. A good VisualQA system is an open ended one, where the questions can be about different target regions in an image which includes the background and the underlying context. This report provides an in-depth analysis on the results obtained on the MSCOCO Dataset [1] for the VisualQA task.**

I. Introduction

VisualQA is an interesting and challenging real world problem with wide ranging applications. It can aid the visually impaired by helping them understand images from the web and in the real world. VisualQA can also help in surveillance systems, by querying, rather than manually looking at each surveillance image. A lot of researchers also believe that VisualQA can also serve as a Visual Turing Test [2].

Training a VisualQA model is challenging as there can be multiple ways to ask the same question. Some of the questions can be semantically similar as well (*eg : Is there a woman next to the car ?* and *Is there a car next to a woman?*) are semantically similar questions. The questions can be arbitrary and open ended as well. They can encompass several sub-problems like object detection (*What is in the image?*), attribute classification (*What color is the cat?*), scene classification (*Is there sunshine outside?*), questions related to counting (*How many horses in the image?*) etc.

In this report, we experiment with the MSCOCO Dataset obtained from [3] and report the results on the models we tried. The models will be explained in detail in V.

II. Related Work

Multimodal Representations : Many approaches to visualQA have aimed at learning representations for the visual and language modality in a joint multimodal space. [1] proposed combining an image embedding model with a question embedding model. The image embedding model was a CNN Model, while the question embeddings were generated an LSTM. They also proposed MSCOCO Dataset [3], which is a large dataset, consisting of over *204,721* images, over *760K* questions and *10M answers*.

Attention: Instead of using the entire image embedding from the fully connected layer of deep CNN, many others have explored image attention models and question attention models. [4] propose a novel co-attention model for VQA, that combines the attention for image and the question. They also build a hierarchical architecture that combines the attention for question at the word level, phrase level and the question level.

Fusion Methods : The image + Bag of Words model in [5] are among the first ones to use concatenation to merge image embeddings with question embeddings. [6] propose a multimodal tucker fusion method, based on bilinear interactions between modalities. They claim to have achieved the current state of the art results on the MSCOCO Dataset.

III. Dataset

The VQA Dataset [3] consists of two versions, **v1.0** & **v2.0**. The dataset contains raw images for the questions, questions annotated with image ids and answers. There are also other versions of the dataset which contain balanced binary abstract scenes. For the images there are also captions provided for some part of the dataset.

A. Questions

For this work, we use the VQA2 dataset, which consisted of *443,757* questions for the training dataset and *214,354* questions for the validation dataset. The questions are also divided into several types of questions, *What is ... , Is there ...., How many .... , Does the....*, essentially, counting based questions, attribute classification, scene classification, open ended questions etc. Table III.1 lists out some sampled questions from the dataset.

Table III.1: Question examples

Is this a civilian aircraft ?
What fruit is shown ?
Is he dressed to play basketball ?
What are the people in the background doing ?
What is to the right of the soup ?

B. Answers

Most of the answers for the questions are single word answers, while most of the answers are less than five words in the dataset. This is not surprising, since most of the questions require eliciting specific information from the image. Table III.2 lists out the top 10 common answers and their count in the training dataset.

Table III.2: Top answers in the training dataset

Answer Text	Count
Yes	84978
No	82516
1	12540
2	12215
white	8916
3	6536
blue	5455
red	5201
black	5066
0	4977

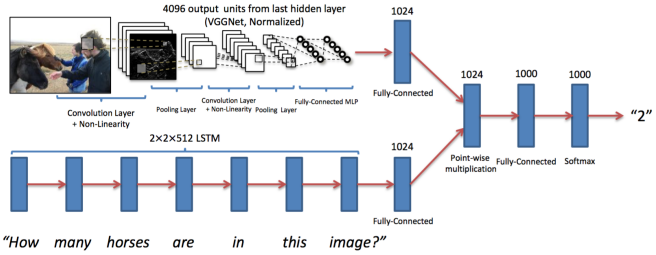

Figure IV.1: LSTM + Image CNN Model

IV. Baseline Model

For the baseline model for VQA, we chose the LSTMQ + I model proposed in [1]. The model is pictorially depicted in IV.1. The image is taken from [1].

The model consists of two parts :

- **Image Model** : The image embeddings are generated using the pretrained VGG19 model. More details on the VGG model and its use in generating embeddings have been discussed in Section VI-B
- **Language Model**: The language model consists of a two layer LSTM of 512 hidden units, the output of which is passed onto a fully connected layer to get a 1024 dimension output. The input to the LSTM layer is a 300 dimensional embeddings of the questions, where the embeddings are used from *GloVe* vectors [7].

The outputs from both the image and language layer are fused by using the Concatenation Fusion Model (section VI-D) which are then passed through fully connected layer with dropouts. Finally, an output layer consisting of softmax activation function is used for generating output probabilities for each of the top 1000 answers.

Several open source implementations of this baseline model are available off the shelf on github. [1] [2] [3].

V. Proposed Approaches

For this work, we modify the Baseline Model by experimenting with the its three different components - The Image Model, the Language Model and the Fusion Model. Various alternatives of each of the components have been discussed in VI. A compendium of results have been present in Table VII.1

[1]https://github.com/anantzoid/VQA-Keras-Visual-Question-Answering
[2]https://github.com/avisingh599/visual-qa
[3]https://github.com/VT-vision-lab/VQA_LSTM_CNN

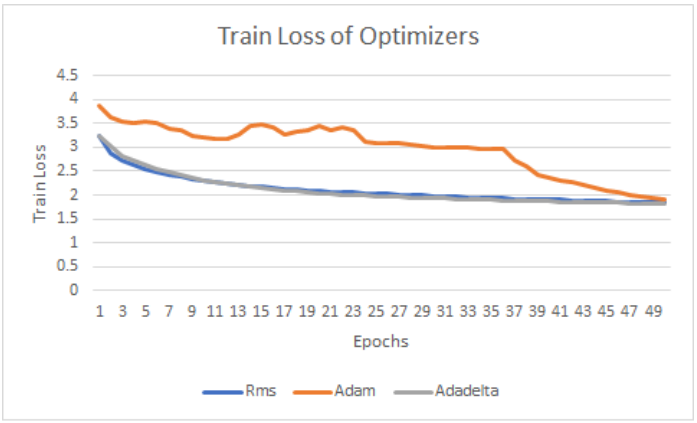

Figure VI.1: Train Loss of Optimizers on 50 epochs

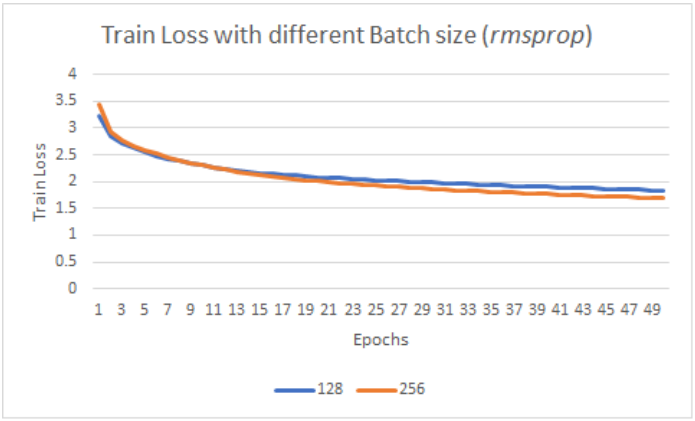

Figure VI.2: Train Loss of Rmsprop on batch size 128 and 256

VI. Experimental Setup

A. Hyperparameters

The hyperparameter tuning was done on the baseline model as described in the section IV. The following were the hyperparameters adjusted :

1) Optimizers and Learning Rate: We try studying the behavior of learning curve with different optimizers like *adam* [8], *rmsprop*, and *adadelta* [9]. The existing model uses *rmsprop* [10] with no learning-rate decay. When learning-rate is kept constant throughout the epochs, the train loss observed was high so we decrease the learning-rate by 0.1 after every 5 epochs. Since *adadelta* gets rid of learning-rate completely from its formulation , this decay was not needed. Keeping the batch-size constant (at 256), we find *rmsprop* giving much better learning curve than others.

2) Batch-Size: Since *rmsprop* gave the best learning curve, we experimented with different batch sizes, 128 and 256.

3) Batch Normalization: Batch normalization [11] has been effectively used to reduce internal covariate shift so we try adding this after each of our LSTM layers and surprisingly, it worsened the training loss as well as validation accuracy. We did not further take into account the batch normalization, as it consistently lead to poorer results on the validation dataset.

4) Activation Function: We experimented with two activation functions : *tanh* & *relu*. Apart from *Adam* optimizer, where the difference in validation accuracy was over 6%, the activation function did not seem to affect the results much with the other experiments.

The results for the hyper parameter tuning are summarized in TableVI.1.

Table VI.1: Validation accuracy on various hyper parameters, on the **Open Ended MSCOCO Dataset**

Optimizer	Activation Function	Batch Size	Val Accuracy
Adam	relu	256	0.3787
Adam	tanh	128	0.4221
Adam	tanh	256	0.4177
RmsProp	relu	128	0.4678
RmsProp	tanh	128	0.4693
RmsProp	tanh	256	0.4803
AdaDelta	tanh	128	0.4558

We did not expect such a big difference just by choosing different set of optimizer and activation functions. Therefore all our experiments use Adam optimizer with relu activation function and batch size of 128.

B. Image Modeling

In order to generate image embeddings, we used two state-of-the-art architectures which have been pre-trained for the Imagenet Large Scale Visual Recognition Challenge (ILSVRC) [12]. Embedded representations were obtained by collecting the output of the penultimate layers of these models after feeding the images as inputs to the models.

- **VGG19** The VGG19 model [13] is a 19-layer deep network which was used by the VGG team in ILSVRC-2014 competition. Each image embedding is a vector of size 4096.
- **RESNET** The RESNET model [14] is a 152-layer deep neural network architecture by the Microsoft Research team. This model was the winner of ILSVRC-2015 competition. Each image embedding is a vector of size 2048.

While training our models, the weights of these embedding models were frozen (kept constant).

C. Language Modeling

We explore several techniques to extract features from questions, on the word, phrase as well as sentence level. On the word level, we compare the effectiveness of one-hot word vectors vs embedding words onto a learned vector space(*GloVe*)[7]. These vectors are then fed to a recurrent neural network model(LSTM or BiLSTM) to learn a fixed-size embedding for the entire sentence. We found that freezing the GloVe embedding layer improved accuracy.

On the phrase level, we have explored simple compositional n-gram features.

The compositional n-gram features are calculated as follows. If Q = $(q_1, q_2, ...q_n)$ is a sentence with q_i being the individual word representations, then we use a max pooling over normalized sums of word representations to extract compositional features from windows of size 1,2 and 3.

$max^\star \left(q_i, \left(\frac{q_{i-1}+q_i}{\sqrt{2}}\right), \left(\frac{q_{i-2}+q_{i-1}+q_i}{\sqrt{3}}\right)\right)$

where, $max^\star$ refers to element-wise max operation on vectors

Normalization was necessary to neutralize the effect of summation so that the elementwise maximum operation is unbiased and chooses the proper features.

To represent the entire question, we use a GRU initialized with the parameters of a pretrained Skip-thoughts model [15] which predicts the surrounding sentences of an encoded passage. Sentences that share semantic and syntactic properties are thus mapped to similar vector representations.

D. Fusion Models

The issue of merging visual and linguistic information is crucial in VQA. Complex and high-level interactions between textual meaning in the question and visual concepts in the image have to be extracted to provide a meaningful answer.

We consider three different types of fusion models to combine the image embeddings and question embeddings. The inputs to the fusion models are image embedding and question embedding vectors, the outputs are vectors whose sizes depend on the fusion models used.

In the model descriptions given below, we will be representing image embeddings as $V = (v_1, v_2 \dots v_m)$ and question embeddings as $Q = (q_1, q_2, \dots q_n)$ with m not necessarily equal to n unless stated otherwise.

- **Concatenation** In this model, image embedding vectors and question embedding vectors are concatenated. The image embedding vector and question embedding vector shapes may be same or different. Formally, the output of this fusion model may be represented as:

$$X = (v_1, v_2 \dots v_m, q_1, q_2, \dots q_n)_{m+n}$$

- **Element-wise Product (MLB)** In this model, the fusion vector is calculated as element-wise product of the question and image embeddings. This model necessitates the vector length of question and image embeddings to be the same. Formally, the output of this fusion model may be represented as:

$$\begin{aligned} X &= V \frown Q \\ &= (v_1 q_1, v_2 q_2, \dots v_n q_n) \end{aligned}$$

- **Tucker Fusion (MUTAN)** In [16], Ben-younes et. al suggests a novel multimodal fusion scheme based on bilinear interactions between modalities. To control the

Fusion Method	Image Features	Question Features	Val-Accuracy
Mutan	ResNet152	Skipthought	58.2
Mutan	ResNet152	One-hot	33.1
Mutan	VGG19	Skipthought	45.4
MLB	VGG19	BiLSTM	47.4
MLB	VGG19	Onehot	40.2

Table VII.1: Validation Accuracy of Different Models on MSCOCO Open Ended Dataset

number of model parameters, MUTAN reduces the size of the mono-modal embeddings, while modeling their interaction as accurately as possible with a full bilinear fusion scheme.

VII. Results & Discussions

From the results in VII.1, we see that the best results were obtained for the combination of Resnet152 as image embedding model, Skip Thoughts as language embedding model, and Tucker Fusion as the fusion model. Since Resnet152 model has shown to outperform VGG19 in image recognition tasks [14] and Skipthoughts have been shown to be better at representing sentences than word embeddings followed by LSTM [15], we hypothesize that the results are largely due to improved embedding. We have tested our hypothesis by changing one of the embedding models at a time while keeping the others constant, and we see that the accuracy diminishes, reinforcing our stated hypothesis. Further we see that the Tucker Fusion model proposed in [6] works better at gluing the information obtained from image embeddings and question embeddings, since changing the fusion model only also diminishes the results.

VIII. Conclusion

Through a comparative analysis of different combinations of language, image and fusion embedding models, we were able to show that the best results were obtained with the best embedding models (rather unsurprising!). However, we also observed that fusing of information from image and question embedding models is also an important determinant in the performance models, therefore more sophisticated models like Tucker Fusion should be explored. More research in this direction will also help in other tasks which involve fusing information from two different sources - like image retrieval via captions.

IX. Future Work

To the best of our knowledge, all the current work in this area pose the problem of Visual Question Answering as an answer selection task. Most frequent answers in the training corpus are selected, labeled as different classes, and the model is trained to identify the answer as one of these classes. As a further work, we propose to modify the framing of the task as that of an answer generation. This will improve the accuracy since top 1000 most frequent answers only cover about 80% of all the answers.

Image captioning is another challenge by the VQA organizing team. The training and validation dataset are therefore annotated with captions as well. Another area of investigation could be leveraging the caption dataset by posing the model as a multi-task learning problem - generate the caption and use the caption as additional information for getting answers. Since caption may contain words or phrases which form the answer, this can help in increasing the accuracy of the question answering task.

References

[1] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. L. Zitnick, and D. Parikh, "VQA: Visual Question Answering," in *International Conference on Computer Vision (ICCV)*, 2015.

[2] M. Malinowski, M. Rohrbach, and M. Fritz, "Ask your neurons: A deep learning approach to visual question answering," *CoRR*, vol. abs/1605.02697, 2016.

[3] "Visual question answering, http://visualqa.org/index.html,"

[4] J. Lu, J. Yang, D. Batra, and D. Parikh, "Hierarchical question-image co-attention for visual question answering," *CoRR*, vol. abs/1606.00061, 2016.

[5] M. Ren, R. Kiros, and R. S. Zemel, "Image question answering: A visual semantic embedding model and a new dataset," *CoRR*, vol. abs/1505.02074, 2015.

[6] H. Ben-younes, R. Cadène, M. Cord, and N. Thome, "MUTAN: multimodal tucker fusion for visual question answering," *CoRR*, vol. abs/1705.06676, 2017.

[7] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1532–1543, 2014.

[8] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2014.

[9] M. D. Zeiler, "ADADELTA: an adaptive learning rate method," *CoRR*, vol. abs/1212.5701, 2012.

[10] T. Tieleman and G. Hinton, "Lecture 6.5—RmsProp: Divide the gradient by a running average of its recent magnitude." COURSERA: Neural Networks for Machine Learning, 2012.

[11] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *CoRR*, vol. abs/1502.03167, 2015.

[12] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015.

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

[14] 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*, pp. 770–778, 2016.

[15] R. Kiros, Y. Zhu, R. R. Salakhutdinov, R. Zemel, R. Urtasun, A. Torralba, and S. Fidler, "Skip-thought vectors," in *Advances in neural information processing systems*, pp. 3294–3302, 2015.

[16] H. Ben-younes, R. Cadene, M. Cord, and N. Thome, "Mutan: Multimodal tucker fusion for visual question answering," *arXiv preprint arXiv:1705.06676*, 2017.