Visual QA Assistance for Visual Impairment

Akanksha Malhotra **Piyush Mishra** **Pratik Revankar**

College of Engineering
University of Colorado at Boulder
Email: {Akanksha.Malhotra , Piyush.Mishra, Pratik.Revankar}@colorado.edu

Abstract

In recent years, with the advent of deep learning, the AI community is trying to move towards multidisciplinary approach where we combine various facets of AI like natural language processing, computer vision, reinforcement learning, etc. One such example is Visual Question Answering (VQA) which combines Natural Language Processing and Computer Vision. Several such VQA models have been proposed in recent years, which combine textual information and visual data, that allow user interaction in the form of natural language question-answers. While these works have a more general task, we aim to use this model to assist those who suffer from visual impairments. We will implement a VQA model that uses a *co-attention* mechanism to explain both the image and the question attention. For the user's trust on the model, we decided to make it explainable, i.e, we will provide proofs from both visual and textual side that our model is working perfectly by generating similar images to the given image and explanation-*by elaboration*, to show that the given answer is correct. We will be using the VQA v2 dataset and VQA-E dataset to train our model.

Keywords: Visual Question Answering, Co-Attention, Visual Impairment

1 Introduction

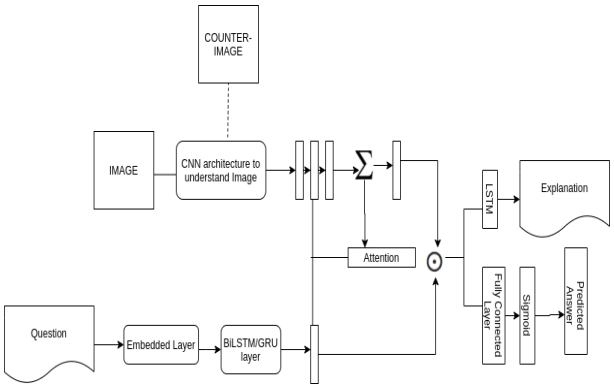
A picture may truly be worth a thousand words, but what purpose does that serve if one cannot perceive it to its full entirety? People who suffer from visual impairments like loss of *central* or *peripheral* vision, *visual agnosia*, *night blindness* or *blurred vision*, do not perceive their environment completely and require some corrective measures. Vision is important to sense and understand the world around us - like people's faces and their expressions, what different things look like and understand their dimensions, and our physical environment where we live and move, including approaching hazards. We thus aim to use a VQA model to help those that need the assistance.

Visual Question Answering [1] [2] is an emerging multi-disciplinary research problem in the industry and had generated numerous possibilities for applications in visual assistance and measuring attention in visual information. To answer questions about an image, the model needs to pay attention and understand both the visual and textual information. In recent years several models [2] [3] [4] have been developed to understand VQA, which generally create a feature map that have both text and image feature encodings. We propose a model that borrows from these models, with specific enhancements based on textual and visual information to provide an assisted tool for visual impairment.

Our solution also aims to make the model *explainable*, in keeping with the spirit of "Explainable AI" [5]. To achieve this, we have two modalities for explanations - *textual explanation* [6]; where the answer to the questions is more elaborate, rather than a simple "yes or no" or a "number", etc., and *visual explanation* [2]; in the form of counter-examples which are drawn from the dataset that are close to the input image (nearest-neighbor) but have the opposite or wrong answer to the question. These "counter-examples" are retrieved from the VQA v2 dataset [2] using the complementary image mapping.

2 System Design and Implementation

2.1 Architecture



The VQA system is divided into three parts:

2.1.1 Understanding Image

To understand the image, we used a pre-trained network - ResNet152, to get the image feature vector before the final 'softmax' layer.

2.1.2 Understanding Question

To understand the question, we need to get the word embeddings for each word in the question. We used a pre-trained word embedding - Glove (glove.GB.300d); and passed these embeddings through a 'Sequential' model, in TensorFlow, using a GRU layer. This gave us the final feature embedding for the question.

2.1.3 Applying Attention and creating a Joint-Embedding

We used a *question-guided soft attention* mechanism which most modern VQA models use. This was inspired from the VQA-E approach proposed by Li et.

al. [6]. For each patch in the image, the feature vector v_i and the question embedding q are first projected by non-linear layers to the same dimension. Next we use the Hadamard product (i.e., element-wise multiplication) to combine the projected representations and input to a linear layer to obtain a scalar attention weight associated with the input image patch. The attention weights τ are normalized over all patches with a 'softmax' function. Finally, the image features from all patches are weighted by the normalized attention weights and summed into a single vector v as the representation of the attended image. [6]

$$\tau_i = w^T(Relu(W_v v_i) \odot Relu(W_q q_i))$$

$$\alpha = softmax(\tau)$$

$$v = \Sigma_{i=1}^{P} \alpha_i v_i$$

Next, the representations of the question q and the image v are projected to the same dimension by non-linear layers and then fused by a Hadamard product:

$$h = Relu(W_q h_q) \odot Relu(W_v h_v)$$

where h is a joint representation of the question and the image, and then fed to the subsequent modules for answer prediction and explanation generation. [6]

2.2 Answer Generation

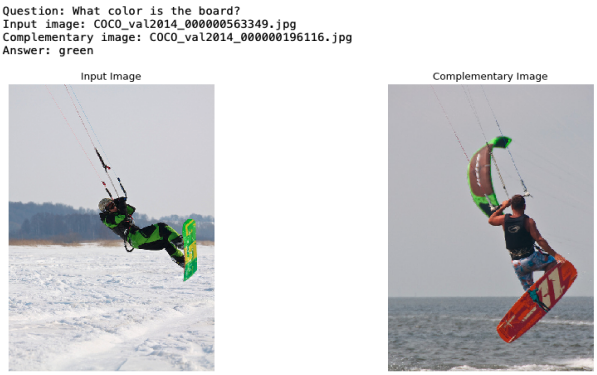
In our model architecture, we have treated the problem of answer generation as a multi-label classification problem. The joint representation from the attention layer is given as an input to a Dense layer with output dimensions of 1024. This again passes through the Dense layer with softmax as the activation function and output dimension equal to answer vocab.

2.3 Textual Explanation Generation

We use a GRU as a 'generator' to generate the explanation for the predicted answer. The GRU input is the joint representation of the question and the image from the attention layer to create the explanation. The collective representation is then passed through a dense layer, with a maximum length of the explanation vector as the output dimension. The output of dense vector goes through an embedding layer. The embedding is an input to a GRU, with return_sequence parameter set to True. The output of GRU passes through the Dense layer with output dimensions equal to the encoding unit of GRU. This output again passes through a Dense layer to produce the final explanation. The loss used was categorical cross entropy.

2.4 Counter Image for Model Validation

In order to validate our model predictions and and make it more 'explainable", we decided to use the complementary images provided in the updated, VQA v2 dataset - "complementary pairs list". These image pairs were created based on a images which had the same applicable question, but different answers. These were manually annotated using Amazon Mechanical Turk, where ten images were shown to an individual, with a given input image and question, and were asked to vote the best counter/negative example. We used the VQA-E dataset to first find common question mappings from the VQA v2 dataset and used the question ids, and their associated image ids to query from the MS-COCO images.



3 Dataset

We utilised the VQA v2 dataset [2], which is a enhanced, balanced dataset built on top of the VQA dataset, built by Antol et.al. [1]. This dataset includes complementary images (for visual explanation). The original VQA dataset was built using the image and captions from MS-COCO dataset [7]. We also make use of the VQA-E dataset built by Li et.al [6] which contains explanations for each image as annotations, and is able to better localize and understand the important regions in the images than the original VQA model.

VQA v2 dataset contains 204K images and 1100K questions along with 295K complementary pairs lists.

VQA-E dataset contains 108K images, 269K questions and 269K explanations.

3.1 Custom Dataset

In order to train our model for our proposed architecture, we combined the data labels from VQA v2 and VQA-E datasets, creating a new dataset which contains both textual and image explanations/validations. We have made this dataset public for future work and research Visual Question Answering, where the model outputs are more "explainable" to the user.

The custom VQA-EI dataset contains 160K images, 269K questions and 269K explanations.

4 Results

Although the number of epochs were small, the model showed consistent decrease in the loss. An with this increase in loss our accuracy of predicting the answer kept on increasing. While the improvement in explanation accuracy was small due to the small set of dataset (due to hardware issue) and fewer epochs, the gradual increase was a positive sign to the model per-

formance. We achieved an accuracy of 70% on the train set.

Model: "model_1"

Layer (type)	Output Shape	Param #	Connected to
ques (InputLayer)	[(1, 16)]	0	
embedding (Embedding)	(1, 16, 300)	2700300	ques[0][0]
img (InputLayer)	[(1, 224, 224, 3)]	0	
gru (GRU)	(1, 1024)	4073472	embedding[0][0]
resnet152 (Model)	multiple	58370944	img[0][0]
dense (Dense)	(1, 1024)	1049600	gru[0][0]
dense_1 (Dense)	(1, 1024)	2098176	resnet152[1][0]
attention (Attention)	(1, 1024)	0	dense_1[0][0] dense_1[0][0] dense[0][0]
dense_2 (Dense)	(1, 1024)	1049600	attention[0][0]
multiply (Multiply)	(1, 1024)	0	dense[0][0] dense_2[0][0]
dense_3 (Dense)	(1, 45)	46125	multiply[0][0]
embedding_1 (Embedding)	(1, 45, 300)	13500	dense_3[0][0]
gru_1 (GRU)	(1, 45, 1024)	4073472	embedding_1[0][0]
dense_4 (Dense)	(1, 1024)	1049600	multiply[0][0]
dense_5 (Dense)	(1, 45, 1024)	1049600	gru_1[0][0]
ans (Dense)	(1, 2545)	2608625	dense_4[0][0]
exp (Dense)	(1, 45, 9001)	9226025	dense_5[0][0]

Total params: 87,409,039
Trainable params: 26,337,795
Non-trainable params: 61,071,244

Figure 1: Model Summary

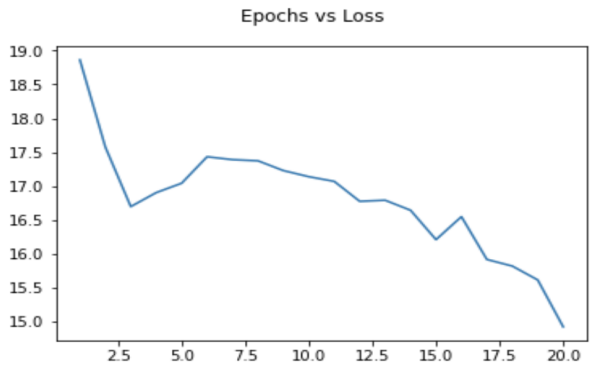


Figure 2: Change loss with epochs

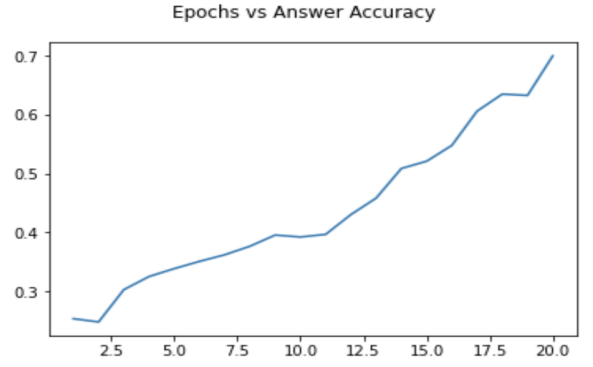


Figure 3: Change loss with epochs

5 Challenges

The workload for the dataset manipulations in the custom VQA-EI dataset is computationally expensive. To resolve this issue, we setup a virtual machine on Google Cloud Platform with 96 vCPUs and backed by a 50GB SSD drive. We saw significant increase in performance by parallelizing the code using the multiprocessing package in Python.

6 Conclusion and Future Work

While we used counter examples, annotated by Mechinacal Turkers, for explaining a model's prediction, this functionality can be automated by feeding in the question along with the image for finding the closest counter example. The model should be able to find the image similar to the input image while the answer to the question should be different. Also an enhanced dataset will be needed with more image and question pairs for better model training.

In this project, we extended the explainability of a model by including explanations which is generated from the given captions and images from the provided counter examples. While the paper doesn't make any claim about the efficiency of the model, it gives some insight to how a model can be better explained and be more reliable from a user perspective.

Github Link: https://github.com/AkankshaMalhotra/VQA-E

References

[1] Antol, S., Agrawal, A., Lu, J., Mitchell, M., Batra, D., Lawrence Zitnick, C. and Parikh, D. *Vqa: Visual question answering.*, Proceedings of the IEEE international conference on computer vision (pp. 2425-2433), 2015.

[2] Goyal, Y., Khot, T., Summers-Stay, D., Batra, D. and Parikh, D. *Making the V in VQA matter: Elevating the role of image understanding in Visual Question Answering.*, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 6904-6913), 2017.

[3] Lu, J., Yang, J., Batra, D. and Parikh, D. *Hierarchical question-image co-attention for visual question answering.*, Advances in neural information processing systems (pp. 289-297), 2016.

[4] Fukui, A., Park, D.H., Yang, D., Rohrbach, A., Darrell, T. and Rohrbach, M. *Multimodal compact bilinear pooling for visual question answering and visual grounding.*, arXiv preprint arXiv:1606.01847, 2016.

[5] Gunning, D. *Explainable artificial intelligence (xai).*, Defense Advanced Research Projects Agency (DARPA), nd Web, 2., 2017.

[6] Li, Q., Tao, Q., Joty, S., Cai, J. and Luo, J. *Vqa-e: Explaining, elaborating, and enhancing your answers for visual questions.*, Proceedings of the European Conference on Computer Vision (ECCV) (pp. 552-567). Vancouver, 2018.

[7] Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P. and Zitnick, C.L. *Vqa-e: Microsoft coco: Common objects in context.*, European conference on computer vision (pp. 740-755). Springer, Cham. Vancouver, 2014.

GitHub: