Where to Look and How to Answer: Using Regions and Answer Types for Visual Question Answering

Surbhi Goel
University of Texas at Austin
surbhi@cs.utexas.edu

Ashwini Venkatesh
University of Texas at Austin
ashuven6@cs.utexas.edu

Abstract

We attempt to solve the task of Visual Question Answering. We propose an architecture where we leverage regions of the image along with the question to model attention. We also propose a modification to the output layer gradient updates which takes into account the type of the answer predicted and promotes correct answer types. We evaluate the system against the VQA existing baselines and other models on the VQA dataset for the open-ended question task.

1. Introduction

Visual Question Answering is the task of accurately answering free-form and open-ended natural language questions based on a given image. The task poses a heavy challenge as it requires strong understanding of the image as it combines various vision tasks, such as scene recognition, object recognition/localization, knowledge-base reasoning and commonsense, into one task.

As the results of [1] show, the accuracy of the model with just the question is only a few percent lower than the model which gets access to both the question and image. This shows that the techniques used are unable to exploit information from the image. Intuitively the next step is to examine the question and figure out which region in the image is of importance for answering the same. In this paper, we use region proposals for the image and learn to weight important regions that are essential for answering questions.

Often the answers to the questions are semantically incorrect. A question "Which car is in the image?" expects the answer to be a type of car but the model often is not able to decipher this. In this paper, we look at modifying the output layer gradient updates to penalize the prediction of answers of a wrong type. We modify the loss function in order to promote answers that are semantically correct. By doing so, we wish to promote the model to learn what type of answer to generate for a given question.

Figure 1: Visual Question Answering Task [1]

The following section discusses the related work in the domain of VQA. Following that we present the technical aspects of the paper including the proposed model and gradient update technique along with implementation details. Subsequently we show experimental results on the VQA dataset. In the final section, we conclude and suggest directions of future work.

2. Related Work

[1] introduced the VQA dataset and proposed the corresponding task. The dataset contains both open ended and Multiple Choice Questions. The real scene images are based on COCO dataset. The dataset also contains abstract scenes which is constructed from a clipart. In our experiments we will only focus on open ended questions and real scene images. They evaluate several baselines out of which 2 are mentioned in Table 1. [1] uses a 2-channel image and question model that culminates with a softmax over K possible outputs as shown in Figure 2. The work experiments

Model	All	Y/N	Number	Other	All*
VQA [1] - deeper LSTM Q + norm I	57.75	80.50	36.77	43.08	NA
VQA [1] - BoW Q + C	54.70	75.82	40.12	42.56	NA
Where To Look [7]$^{\$}$	62.44	77.62	34.28	55.84	62.43
Spatial Attention [10]	57.99	80.87	37.32	43.12	58.24
Compositional Memory [3]	52.62	78.33	35.93	34.46	NA
Stacked Networks [11]	58.70	79.30	36.60	46.10	58.90
Dynamic Parameter Prediction [6]	57.22	80.71	37.24	41.69	57.36
Knowledge Based Reasoning [9]	59.22	81.02	38.47	45.30	59.50

Table 1: Performance comparison of existing models. * - performance on test-standard. $^{\$}$ - evaluated on MCQ.

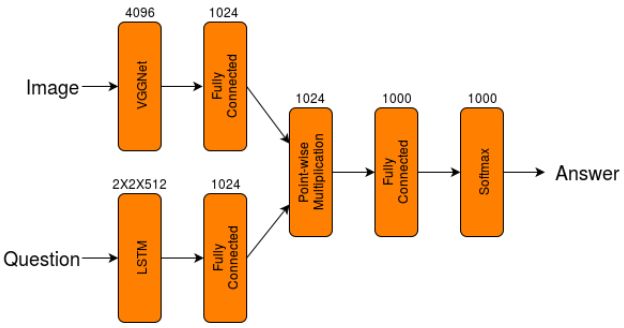


Figure 2: Existing Deeper LSTM and Normalized CNN VQA Model [1]

with two alternatives for the image channel - the last hidden layer of VGGNet and l_2 normalized activations from the last hidden layer of VGGNet. The questions channel is experimented with three approaches - Bag of Words, LSTM and a deeper LSTM with 2 hidden layers. The model is described in further detail in the next section.

Following this basic model, various different approaches have been tried to augment the basic model. The succeeding models have focused separately on Vision and NLP aspects of the problem. In the Vision direction, models have tried to incorporate attention through various techniques to focus on the area of importance in the image. On the NLP front, models have tried leveraging existing knowledge bases and using captions to include additional information useful for answering the question. We give a brief overview of these approaches.

Various papers [2, 7, 10, 3, 11] attempt to solve this task by focusing attention on a specific regions in the image to answer the question. [6] uses a convolutional neural network (CNN) with a dynamic parameter layer whose weights are determined adaptively based on a question. The adaptive parameter prediction, a separate parameter prediction network, consists of gated recurrent units (GRU) which takes a question as its input and a fully-connected layer generates a set of candidate weights as its output. Since the dynamic parameter layer is a fully connected layer, it is challenging to predict a large number of parameters in the layer to construct the CNN for ImageQA and use a hashing technique, where the candidate weights given by the parameter prediction network are selected using a predefined hash function to determine individual weights in the dynamic parameter layer. Our original plan was to use this architecture to build attention into our model. The plan was to shift the dynamic parameter weight layer to the last max pooling layer of the CNN of the classification network. The last max pooling layer has dimension $512 \times 14 \times 14$ where the 14×14 is the reduced representation of the original image with each region has 512 dimensional feature. Shifting the layer here will learn the weights of different regions in the image. Thus we won't have an explicit region selection algorithm for creating a attention based network. However we were unsuccessful in setting up this model due to limitation of GPU resources available.

[9] uses an automatically generated description of an image with information extracted from an external Knowledge Base to provide an answer to a general question about the image. The image description takes the form of a set of captions, and the external knowledge is text based information mined from a Knowledge Base. Given an image-question pair, a CNN is first employed to predict a set of attributes of the image. The attributes cover a wide range of high-level concepts. An existing image captioning model is applied to generate a series of captions based on the attributes. They then use the detected attributes to extract relevant information from the KB. Specifically,for each of the top-5 attributes detected in the image a query is generated which is applied to a Resource Description Framework KB, such as DBpedia. While this method uses the attributes and captions along with the question, they finally don't use the images features itself in predicting the final answer.

[7] models selecting image regions relevant to the text-based query. To do this learn an embedding to project question and potential features into a shared subspace to determine relevance with an inner product. However this model is evaluated on Multiple Choice Questions.

3. Technical Approach

We improve on the best performing baseline model provided in [1]. We also borrow ideas from the model provided in [7] to model attention. For completeness of the paper, we first describe these existing models in detail. Following this, we describe our two key approaches.

3.1. Deeper LSTM and Normalized CNN Model

The model computes both image and question embedding and fuses them via element-wise multiplication to obtain a 1024-dim vector that is fed into a MLP with 1000 hidden units (0.5 dropout, $\tanh$ non-linearity) followed by

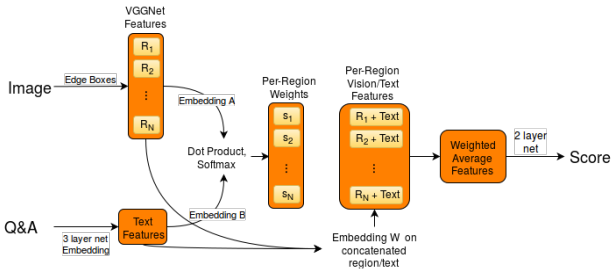


Figure 3: Attention Model of [7]

a softmax layer over the top K answers. For image embedding, the normalized activations from the last hidden layer of VGGNet [8] as the 4096-dim image embedding followed by a transformation to 1024-dim embedding using a fully connected layer with $\tanh$ non-linearity to match the question embedding. For the question embedding, an LSTM with two hidden layers is used to obtain 2048-dim vector which is then fed into a fully-connected layer with $\tanh$ non-linearity to transform to 1024-dim. Each question word is encoded with 300-dim embedding by a fully-connected layer with $\tanh$ non-linearity which is then fed to the LSTM. The input vocabulary consists of all the words in the questions seen in the training set. Figure 1 gives a gives a pictorial representation of the model. This model achieves 58.16% on Open-Ended and 63.09% on Multiple-Choice on test-standard split.

3.2. Where to Look Model

The model attempts to learn an embedding for the textual question and answer pair and an embedding for the set of image regions into a latent space such that the inner product of the two yields an importance weighting for each region. They use a BoW approach for encoding the question and answer using $word2vec$ followed by a two-layer network. The visual features for each region are encoded using the top two layers (including the output layer) of VGGNet, a model trained on ImageNet. The language and vision features are then embedded and compared with a dot product, which is softmaxed to output a per-region importance weighting. Subsequently, a weighted average of concatenated vision and language features is inputted to a 2-layer network that outputs a score to evaluate the correctness of the answer. The score is computed for each question-answer pair (total 18). This model is evaluated for MCQ questions. Each image is given 99 regions along with the image itself. Figure 3 shows the architecture of the model.

3.3. Our Model

We propose a model that leverages the region based weighting idea of [7] (WTL) and incorporates it into the model of [1] (VQA) to solve the VQA open-ended task.

The VQA model benefits from using LSTM to capture structure of question but does not capture attention directly. Though the point-wise multiplication is supposed to weakly capture attention but is not a robust way of doing so. The WTL model is able to learn region based weighting to highlight important regions of the image but works only for MCQ. Also, it uses the question-answer pair embedding to choose region which could be misleading for wrong answers. Another weakness of the model is that it uses a BoW approach to model the question which might not be able to capture the structure of the question.

We combine the positive aspects of both models into our model (shown in Figure 4) in the hope to learn better. Our model works as follows:

- We extract the top-ranked N Edge Boxes from the image after performing non-max suppression with a 0.2 intersection over union overlap criterion. [12] finds object proposals in images using the following observation: the number of edges that are wholly enclosed by a bounding box is indicative of the likelihood of the box containing an object.

- We use VGGNet features [8] of these regions along with the original image as the stacked input. We perform a convolution to learn an embedding to a lower space.

- Similar to the VQA model, we model the question using an LSTM followed by a FC layer to learn an embedding in the same space.

- Following WTL model, we get per region weights by taking a dot product of question embedding and region embedding and applying a softmax over it.

- Subsequently, we use the region weights to take a weighted average of point-wise multiplication products of question and region embedding followed by a FC layer with softmax to produce a distribution across top 1000 answers.

3.4. Weighted Gradient Updates

Figure 6 shows examples of images and corresponding questions for which the original model gave inconsistent answers. To counter this and ensure consistency, we define a new gradient update for the last layer. The new gradient update modifies the existing gradient update based on cross-entropy loss by scaling depending on the type of the actual answer. The intuition is to penalize the right type of answer less harshly than the wrong types in order to feed in information into the model of the type of answers that a question fits. The baseline model often produces incorrect types in the top answers for a question. This gradient will let the

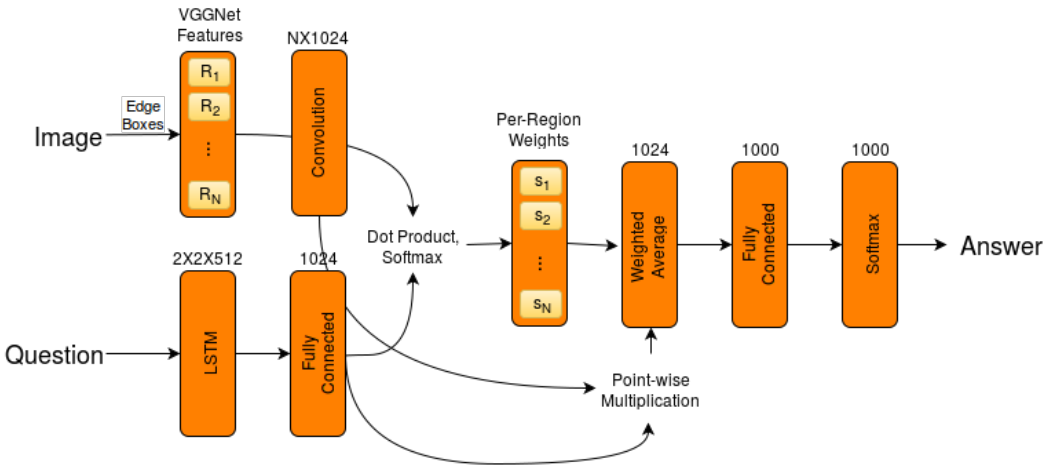


Figure 4: Our Proposed Model based on [1, 7]

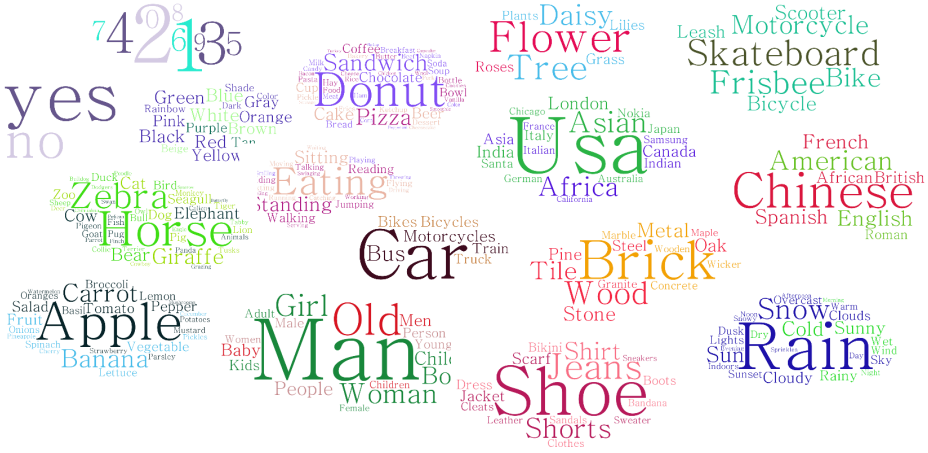


Figure 5: Types used for Weighted Gradient Update

model move towards producing the correct type of answer. Mathematically,

$$\hat{\Delta}_i(x,y) = \begin{cases} \alpha\Delta_i(x,y), & \text{if } type(x_i) = type(y), x \neq y \\ \Delta_i(x,y), & \text{otherwise} \end{cases}$$

where $type()$ gives the type of the answer label and α is the boosting constant.

We consider two categories of types:

- We consider three types: binary ('yes/no'), numeric ('one', 'two', ...) and other.
- We cluster the $word2vec$ embedding of the answer space using k-means clustering to get 50 clusters. We manually filter out incorrect clusters and finally take 17 representative clusters. Our clusters are visualized in Figure 5.

The model is able to mostly learn the first category of types from the structure of the question hence we looked into clustering semantically similar answers instead. As can be seen in Figure 5, the various types group numbers, food, activities, clothes and so on which seem to be semantically coherent.

4. Implementation Details

The code is based on the baseline VQA model code available on Github [4] written in Torch. The model was trained on Maverick (TACC) using 20 clusters for 12 hours each using standard back propagation. The number of iterations run vary because of this time restriction. The questions and answers are encoded using One-Hot encoding before sending to the LSTM whereas the image features are extracted after appropriate scaling using the Caffe model of VGGNet and stored as an HD5 file.

Model	All	Y/N	Number	Other
[1]	54.19	79.77	33.42	40.26
[7]*	62.44	77.62	34.28	55.84
Ours (Low)	52.74	78.43	32.47	38.62
Ours (High)	53.14	78.11	32.92	39.53

Table 2: Comparison of performance on VQA validation set. * Evaluated on MCQ. Others are evaluated on Open Ended questions. Our model ran only 56000 iterations in comparison to 150000 iterations for the VQA model. The value in the bracket indicates the low/high initialization weights of the embedding layers.

For the attention model, due to file size limit, we used $N = 3$ regions per image. The regions were extracted using the Edge-box detector from [12] coded in MatLab using IoU of 0.2. The torch model ran for 50000 iterations with a batch size of 500.

For the weighted gradient update, for 3 types, we update only for type binary and numeric in the last layer and then let it propagate. The other type constitutes most of the answer space and updating those weights in each iteration would not be beneficial and increase the computation cost greatly hence we focus on the binary and numeric types. Similarly, for the $word2vec$ types we do not modify updates for answers that are not in any cluster. We experimented with varying α. The update is done to the gradients of the last layer and back-propagated through the layers. Each model ran for 120000 iterations with a batch size of 500.

5. Experimental Results

We evaluate our models trained using the two approaches proposed in this paper. We analyze the results both quantitatively and qualitatively.

5.1. Quantitative Results

Table 2 gives a comparison of our models following the two approaches to the baseline [1] using the overall accuracy on the validation set of the VQA dataset as the metric. We cannot directly compare our result to that of [7] since the latter is evaluated on Multiple Choice Questions. Our model with attention almost comes close to the baseline in terms of performance. However, out model is trained for 1/3 the number of iterations as the baseline. If we train the model further we may achieve good results against the baseline. Also to evaluate the true potential of the attention based model we need to use more regions per input image. We also experimented with initializing with lower starting weights to prevent the softmax from saturating early on. However we did not observe any appreciable gains using this.

Model	All	Y/N	Number	Other
[1]	54.19	79.77	33.42	40.26
3 Types	54.24	79.75	33.17	40.45
W2V Types (0.90)	53.65	79.63	32.96	39.41
W2V Types (0.95)	54.37	79.96	33.12	40.54
W2V Types (0.99)	54.27	79.83	33.62	40.33

Table 3: Comparison of performance on different word clusters. 3 types uses Binary, Number and Other answer categories to update the gradients on VQA validation set. W2V types uses clusters obtained by running $word2vec$ on top 1000 answers. This uses 17 labels to update the gradients. The value indicated in the bracket indicates the value of α used.

Table 3 gives the comparison of using weighted gradient update on the existing VQA model using the overall accuracy on the validation set of the VQA dataset as the metric. When we use only 3 labels to update the gradient, we don't gain much. This is because the existing model in [1] has already learned the mapping from the question type to the answer type to a great extent. evaluate the $word2vec$ models with different values of α in the gradient update. It is observed that the model gives the best performance when α is 0.95. It does better on both Binary and Other question types. We also evaluate the accuracy of the model to get the right type of answer based on our type categories. Using the types that we clustered, we evaluate the percentage of answers that our model generates that fall in the correct category. We consider the non-clustered answers as a cluster. This gives us 85.7% accuracy of falling in the right type.

5.2. Qualitative Results

To show the successful learnings of the model, we show examples of questions where our models give the correct result while the original VQA model outputted an incorrect result.

Figure 6 highlights the success of weighted gradient update. The model is able to answer the right type of answer where the original model failed. We can see from the examples illustrated that the model learns the type it should or shouldn't answer for a particular question. For example in the picture with the cat on the bookshelf, the original model gave Yes as the answer. However the right answer to this question would be a verb. Similarly in the surfer picture, the model learns that wet suit is a clothing object and hence it is likely that the question which asks what the subject is wearing should have the answer type as some clothing object.

Figure 7 shows the actual image, question and the most important region for answering. It can be clearly scene that if the region proposals are informative, the model is able to answer correctly. Our model is able to pick out features from these images and answer challenging questions. For example the question in the first image asks what is lit in the image? The top region for this image shows the wick of the candle. Unfortunately due to file limit size on Maverick, we were unable to include sufficient region proposals to encompass the whole image. We use 3 top regions proposed by [12] and the entire image itself.

6. Conclusions

Visual Question Answering is a challenging domain and enhancing performance through higher visual information is a hard task. Our model shows a potential way of modeling attention and extracting regions of importance for answering questions. Given sufficient regions, the model shows potential to learn embedding in a common latent space for the question and image such that the combined embedding incorporates information about the areas of focus in the image. Though we have not improved on the baseline overall, we have reached the same accuracy in one-third the iterations showing the success of the model.

Another aspect of the problem is to help the model learn semantic correctness of the answer relative to a question. This ensures that the answer proposed is at least relevant. The model implicitly learns this partly, but we show how we can provide supervision externally to promote the same through our weighted gradient update method. The qualitative results show that the model has been pushed towards the correct direction, however we do get a slight improvement on the baseline in lesser number of iterations, it is hard to say whether we have fully exploited the potential of this approach.

7. Future Work

The work presented in this paper can be further extended in multiple directions. We suggest few possible directions.

For the attention model, using higher number of regions per image would help capture regions of importance over the entire image. With 3 images, presently our model is not able to encompass all salient objects (areas of focus). Generating better region proposals using the question information could be useful too. To further enhance the performance, adding additional region specific information through captions (possibly using Dense captioning [5]) can better performance. Another possibility is to experiment with the network architecture. Adding more complexity through additional FC layers and doing parameter tuning could potentially benefit the model. Also, running a higher number of iterations to help the model converge would be useful. Also, trying different approaches to concatenate region and questions embedding such as appending, point-wise sum or some other linear transformation could help leverage more from the input. In this paper we have experimented only with point-wise multiplication of the two embedding.

On the other hand, for weighted gradient update, making the parameter α dynamic could help with propagating the correct loss. Finding a better clustering of the answers, possibly as a function of the question would be useful for a higher improvement. In all, there is a lot of scope for improvement on this task as the potential of image features is far from being exploited and NLP techniques for answering the question have not been fully explored.

References

[1] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. Lawrence Zitnick, and D. Parikh. Vqa: Visual question answering. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2425–2433, 2015.

[2] K. Chen, J. Wang, L. Chen, H. Gao, W. Xu, and R. Nevatia. ABC-CNN: an attention based convolutional neural network for visual question answering. *CoRR*, abs/1511.05960, 2015.

[3] A. Jiang, F. Wang, F. Porikli, and Y. Li. Compositional memory for visual question answering. *CoRR*, abs/1511.05676, 2015.

[4] D. B. Jiasen Lu, Xiao Lin and D. Parikh. Deeper lstm and normalized cnn visual question answering model. `https://github.com/VT-vision-lab/VQA_LSTM_CNN`, 2015.

[5] J. Johnson, A. Karpathy, and L. Fei-Fei. Densecap: Fully convolutional localization networks for dense captioning. *arXiv preprint arXiv:1511.07571*, 2015.

[6] H. Noh, P. H. Seo, and B. Han. Image question answering using convolutional neural network with dynamic parameter prediction. *CoRR*, abs/1511.05756, 2015.

[7] K. J. Shih, S. Singh, and D. Hoiem. Where to look: Focus regions for visual question answering. *CoRR*, abs/1511.07394, 2015.

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

[9] Q. Wu, P. Wang, C. Shen, A. van den Hengel, and A. R. Dick. Ask me anything: Free-form visual question answering based on knowledge from external sources. *CoRR*, abs/1511.06973, 2015.

[10] H. Xu and K. Saenko. Ask, attend and answer: Exploring question-guided spatial attention for visual question answering. *CoRR*, abs/1511.05234, 2015.

[11] Z. Yang, X. He, J. Gao, L. Deng, and A. J. Smola. Stacked attention networks for image question answering. *CoRR*, abs/1511.02274, 2015.

[12] C. L. Zitnick and P. Dollár. Edge boxes: Locating object proposals from edges. In *Computer Vision–ECCV 2014*, pages 391–405. Springer, 2014.

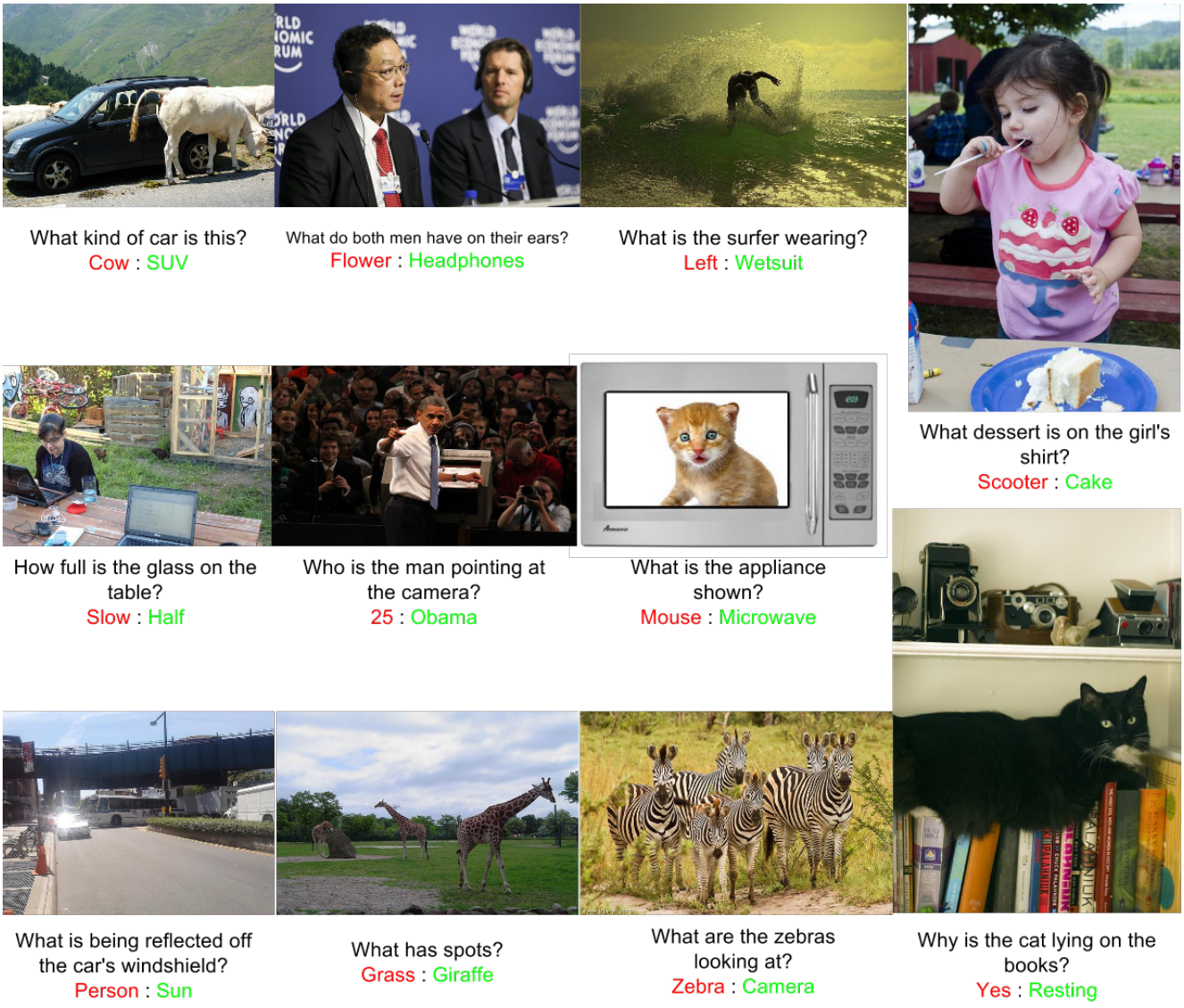
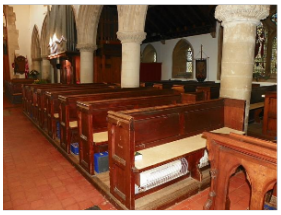


Figure 6: Qualitative Results for Model trained using Weighted Gradient Update. The answers marked in red indicate the prediction made by the baseline model and the answers marked in green indicate the predictions made by W2V (0.95) model

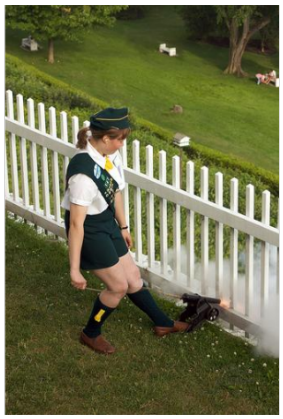

What is lit?
Wine : Candle


If the girls are not wearing white socks what color socks are they wearing?
White : Blue




What structure is this?
Museum : Church


What type of utensil is leaning on the edge of the plate?
Spoon : Fork




Is she a girl scout
No : Yes


What is the woman holding to shield the snow?
Camera : Umbrella

Figure 7: Qualitative Results for Model trained using image regions. The image on the left is the input image whereas the image on the right is the top weighted region. The answers marked in red indicate the prediction made by the baseline model and the answers marked in green indicate the predictions made by our attention model