Where To Look: Focus Regions for Visual Question Answering

Kevin J. Shih, Saurabh Singh, and Derek Hoiem

University of Illinois at Urbana-Champaign

{kjshih2, ss1, dhoiem}@illinois.edu

Abstract

We present a method that learns to answer visual questions by selecting image regions relevant to the text-based query. Our method maps textual queries and visual features from various regions into a shared space where they are compared for relevance with an inner product. Our method exhibits significant improvements in answering questions such as "what color," where it is necessary to evaluate a specific location, and "what room," where it selectively identifies informative image regions. Our model is tested on the recently released VQA [1] dataset, which features free-form human-annotated questions and answers.

1. Introduction

Visual question answering (VQA) is the task of answering a natural language question about an image. VQA includes many challenges in language representation and grounding, recognition, common sense reasoning, and specialized tasks like counting and reading. In this paper, we focus on a key problem for VQA and other visual reasoning tasks: knowing where to look. Consider Figure 1. It's easy to answer "What color is the walk light?" if the light bulb is localized, while answering whether it's raining may be dealt with by identifying umbrellas, puddles, or cloudy skies. We want to learn where to look to answer questions supervised by only images and question/answer pairs. For example, if we have several training examples for "What time of day is it?" or similar questions, the system should learn what kind of answer is expected and where in the image it should base its response.

Learning where to look from question-image pairs has many challenges. Questions such as "What sport is this?" might be best answered using the full image. Other questions such as "What is on the sofa?" or "What color is the woman's shirt?" require focusing on particular regions. Still others such as "What does the sign say?" or "Are the man and woman dating?" require specialized knowledge or reasoning that we do not expect to achieve. The system needs to learn to recognize objects, infer spatial relations, determine relevance, and find correspondence between natural language and visual features. Our key idea is to learn a non-linear mapping of language and visual region features into a common latent space to determine relevance. The relevant regions are then used to score a specific question-answer pairing. The latent embedding and the scoring function are learned jointly using a margin-based loss supervised solely by question-answer pairings. We perform ex-

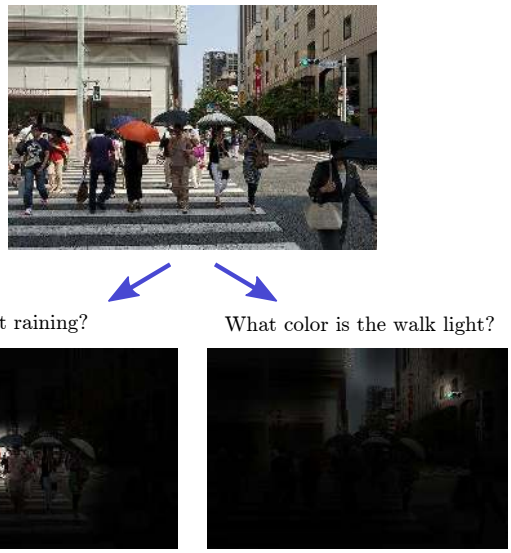

Figure 1. Our goal is to identify the correct answer for a natural language question, such as "What color is the walk light?" or "Is it raining?" We particularly focus on the problem of learning where to look. This is a challenging problem as it requires grounding language with vision and learning to recognize objects, use relations, and determine relevance. For example, whether it is raining may be determined by detecting the presence of puddles, gray skies, or umbrellas in the scene, whereas the color of the walk light requires focused attention on the light alone. The above figure shows example attention regions produced by our proposed model.

What color are the dots on the handle of the utensil? | Is it raining? | Do children like this object?

Figure 2. Examples from VQA [1]. From left to right, the above examples require focused region information to pinpoint the dots, whole image information to determine the weather, and abstract knowledge regarding relationships between children and stuffed animals.

periments on the VQA dataset [1] because it features open-ended language, with a wide variety of questions (see Figure 2). We focus on its multiple-choice format because its evaluation is much less ambiguous than open-ended answer verification.

We focus on learning where to look and provide useful baselines and analysis for the task as a whole. Our contributions are as follows:

- We present an image-region selection mechanism that learns to identify image regions relevant to questions.
- We present a learning framework for solving multiple-choice visual QA with a margin-based loss that significantly outperforms provided baselines from [1].
- We provide a detailed comparison with various baselines to highlight exactly when our region selection model improves VQA performance

2. Related Works

Many recent works in tying text to images have explored the task of automated image captioning [10, 7, 22, 12, 13, 14, 6, 4, 21]. While VQA can be considered as a type of directed captioning task, our work relates to some [22, 7] in that we learn to employ an attention mechanism for region focus, though our formulation makes determining region relevance a more explicit part of the learning process. In Fang et al. [7], words are detected in various portions of the image and combined together with a language model to generate captions. Similarly, Xu et al. [22] uses a recurrent network model to detect salient objects and generate caption words one by one. Our model works in the opposite direction of these caption models at test time by determining the relevant image region given a textual query as input. This allows our model to determine whether a question-answer pair is a good match given evidence from the image.

Partly due to the difficulty of evaluating image captioning, several visual question answering datasets have been proposed along with applied approaches. We choose to experiment on VQA [1] due to the open ended nature of its question and answer annotations. Questions are collected by asking annotators to pose a difficult question for a smart robot, and multiple answers are collected for each question. We experiment on the multiple-choice setting as its evaluation is less ambiguous than that of open-ended response evaluation. Most other visual question answering datasets [17, 23] are based on reformulating existing object annotations into questions, which provides an interesting visual task but limits the scope of visual and abstract knowledge required.

Our model is inspired by End-to-End Memory Networks [19] proposed for answering questions based on a series of sentences. The regions in our model are analogous to the sentences in theirs, and, similarly to them, we learn an embedding to project question and potential features into a shared subspace to determine relevance with an inner product. Our method differs in many details such as the language model and more broadly in that we are answering questions based on an image, rather than a text document. Ba et al. [2] also uses a similar architecture, but in a zero-shot learning framework to predict classifiers for novel categories. They project language and vision features into a shared subspace to perform similarity computations with inner products like us, though the score is used to guide the generation of object classifiers rather than to rank image regions.

Approaches in VQA tend to use recurrent networks to model language and predict answers [17, 1, 23, 15], though simpler Bag-Of-Words (BOW) and averaging models have been shown to perform roughly as well if not better than sequence-based LSTM [17, 1]. Yu et al. [23], which proposes a Visual Madlibs dataset for fill-in-the-blank and question answering, focuses their approach on learning latent embeddings and finds normalized CCA on averaged word2vec representations [10, 16] to outperform recurrent networks for embedding. Similarly, we find a fixed-length averaged representation of word2vec vectors for language to be highly effective and much simpler to train, and our approach differs at a high level in our focus on learning where to look.

3. Approach

Our method learns to embed the textual question and the set of visual image regions into a latent space where the inner product yields a relevance weighting for each region. See Figure 3 for an overview. The input is a question, potential answer, and image features from a set of automatically selected candidate regions. We encode the parsed question and answer using word2vec [16] and a three-layer network. Visual features for each region are encoded using the top two layers (including the output layer) of a CNN trained on ImageNet [18]. The language and vision features are then embedded and compared with a dot product, which is softmaxed to produce a per-region relevance weighting. Using these weights, a weighted average of concatenated vision

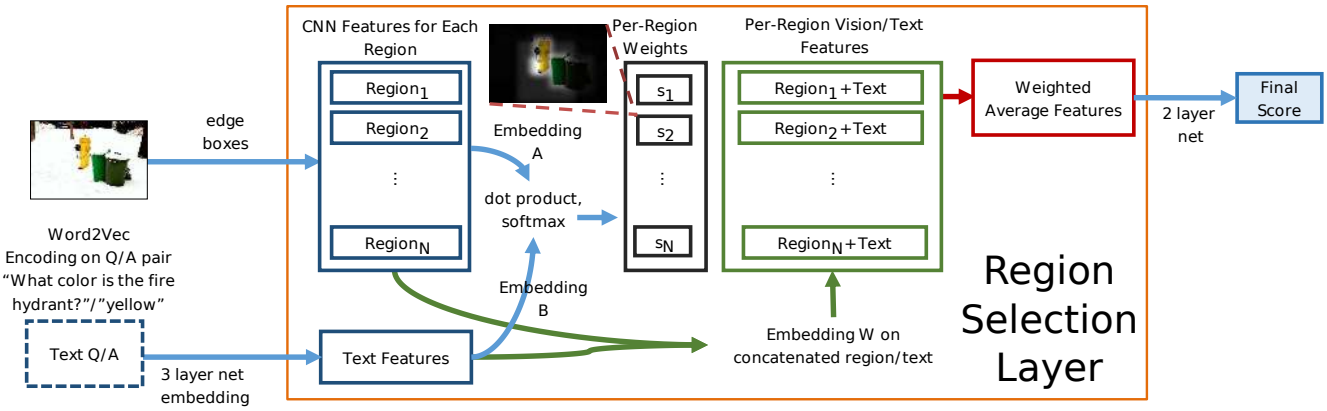

Figure 3. Overview of our network for the example question-answer pairing: "What color is the fire hydrant? Yellow." Question and answer representations are concatenated, fed through the network, then combined with selectively weighted image region features to produce a score.

and language features is the input to a two-layer network that outputs a confidence for the answer candidate.

3.1. QA Objective

Our model is trained for the multiple choice task of the VQA dateset. For a given question and its corresponding choices, the objective of our network aims to maximize a margin between correct and incorrect choices in a structured-learning fashion. We achieve this by using a hinge loss over predicted confidences y.

In our setting, multiple answers could be acceptable to varying degrees, as correctness is determined by the consensus of 10 annotators. For example, most may say that the color of a scarf is "blue" while a few others say "purple". To take this into account, we scale the margin by the gap in number of annotators returning the specific answer:

$$\mathcal{L}(y) = \max_{\forall n \neq p}(0, y_n + (a_p - a_n) - y_p). \quad (1)$$

The above objective requires that the score of the correct answer (y_p) is at least some margin above the score of the highest-scoring incorrect answer (y_n) selected from the set of incorrect choices ($n \neq p$). For example, if $\frac{6}{10}$ of the annotators answer p ($a_p = 0.6$) and 2 annotators answer n ($a_n = 0.2$), then y_p should outscore y_n by a margin of at least 0.4.

3.2. Region Selection Layer

Our region selection layer selectively combines incoming text features with image features from relevant regions of the image. To determine relevance, the layer first projects the image features and the text features into a shared N-dimensional space, after which an inner product is computed between each question-answer pair and all available regions.

Let $V = (\vec{v}_1, \vec{v}_2, ...\vec{v}_K)$ be a collection of visual features extracted from K image regions and $\vec{q}$ be the feature representation of the question and candidate answer pair. The forward pass to compute the relevance weighting of the jth region is computed as follows:

$$g_j = (A\vec{v}_j + \vec{b}^A)^\top (B\vec{q} + \vec{b}^B) \quad (2)$$

$$s_j = \frac{e^{g_j}}{\sum_k e^{g_k}} \quad (3)$$

Here, vectors $\vec{b}$ represent bias vectors for each affine projection. The inner product forces the model to compute region-question relevance (g_j) in a vector similarity fashion. Using softmax-normalization across 100 regions per image ($K = 100$) gives us a 100-dimensional vector $\vec{s}$ of normalized relevance weights.

The vector $\vec{s}$ is then used to compute a weighted average across all region features. We first construct a language-vision feature representation for each region by defining $\vec{d}_j$ as the concatenation of $\vec{v}_j$ with $\vec{q}$. Each feature vector is then projected with W and $\vec{b}^W$ before computing the weighted average feature vector $\vec{z}$.

$$\vec{z} = \sum_j \left(W\vec{d}_j + \vec{b}^W\right) s_j \quad (4)$$

We also tried learning to predict a relevance score directly from concatenated vision and language features, rather than computing the dot product of the features in a latent embedded space. However, the resulting model appeared to learn a salient region weighting scheme that varied little with the language component. The inner-product based relevance was the only formulation we tried that successfully varies with different queries given the same image.

3.3. Language Representation

We represent our words with 300-dimensional Google News dataset pre-trained word2vec vectors [16] for their simplicity and compact representation. We are also motivated by the ability of vector-based language representations to encode similar words with similar vectors, which

may aid answering open-ended questions. Using means of word2vec vectors, we construct fixed-length vectors for each question-answer pair, which our model then learns to score. In our results section, we show that our vector-averaging language model noticeably outperforms a more complex LSTM-based model from [1], demonstrating that BOW-like models provide very effective and simple language representations for VQA tasks.

We first tried separately averaging vectors for each word with the question and answer, concatenating them to yield a 600-dimensional vector, but since the word2vec representation is not sparse, averaging several words may muddle the representation. We improve the representation using the Stanford Parser [5] to bin the question into additional separate semantic bins. The bins are defined as follows:

Bin 1 captures the type of question by averaging the word2vec representation of the first two words. For example, "How many" tends to require a numerical answer, while "Is there" requires a yes or no answer.

Bin 2 contains the nominal subject to encode subject of question.

Bin 3 contains the average of all other noun words.

Bin 4 contains the average of all remaining words, excluding determiners such as "a," "the," and "few."

Each bin then contains a 300-dimensional representation, which are concatenated with a bin for the words in the candidate answer to yield a 1500-dimensional question/answer representation. Figure 4 shows examples of binning for the parsed question. This representation separates out important components of a variable-length question while maintaining a fixed-length representation that simplifies the network architecture.

How many birds are in the photo

| How many | birds | photo | are in |

Is there a cat on the car

| Is there | cat | car | on |

What animal is in the picture

| What animal | animal | picture | is in |

Figure 4. Example parse-based binning of questions. Each bin is represented with the average of the word2vec vectors of its members. Empty bins are represented with a zero-vector.

3.4. Image Features

The image features from 100 rectangular regions are fed directly into the region-selection layer from a pre-trained network. We first select candidate regions by extracting the top-ranked 99 Edge Boxes [25] from the image after performing non-maximum suppression with a 0.2 intersection over union overlap threshold. We found this aggressive thresholding to be important for selecting smaller regions that may be important for some questions, as the top-ranked regions tend to be highly overlapping large regions. Finally, a whole-image region is also added to ensure that the model at least has the spatial support of the full frame if necessary, bringing the total number of candidate regions to 100 per image. While we have not experimented with the number of regions, it is possible that the improved recall from additional regions may improve performance.

We extract features using the VGG-s network [3], concatenating the output from the last hidden layer (4096 dimensions) and the pre-softmax layer (1000 dimensions). The pre-softmax classification layer was included to provide a more direct signal for objects from the Imagenet [18] classification task.

3.5. Training

Our network architecture is a multi-layer network as seen in Figure 3, implemented in MatConvNet[20]. Our fully connected layers are initialized with Xavier initialization [8] and separated with a batch-normalization [11] and ReLU layer [9]. The word2vec text features are fed into the network's input layer, whereas the image region features feed in through the region selection layer.

Our network sizes are set as follows. The 1500 dimensional language features first pass through 3 fully connected layers with output dimensions 2048, 1500, and 1024 respectively. The embedded language features are then passed through the region selection layer to be combined with the vision features. Inside the region selection layer, projections A and B project both vision and language representations down to 900 dimensions before computing their inner product. The exiting feature representation passes through W with an output dimension of 2048. then finally through two more fully connected layers with output dimensions of 900 and 1 where the output scalar is the question-answer pair score.

The training was especially sensitive to the initialization of the region-selection layer. The magnitude of the projection matrices A, B and W are initialized to 0.001 times the standard normal distribution. We found that low initial values were important to prevent the softmax in selection from spiking too early and to prevent the higher-dimensional vision component from dominating early in the training.

4. Experiments

We evaluate the effects of our region-selection layer on the multiple-choice format of the MS COCO Visual Question Answering (VQA) dataset [1]. This dataset contains 82,783 images for training, 40,504 for validation, and 81,434 for testing. Each image has 3 corresponding questions with recorded free-response answers from 10 annotators. Any response that comes from at least 3 annotators is

Model	Overall (%)
Word Only	53.98
Word+Whole Image	57.83
Word+Ave. reg.	57.88
Word+Sal. reg.	58.45
Word+Region Sel.	**58.94**
LSTM Q+I [1]	53.96

Table 1. Overall accuracy comparison on Validation. Our region selection model outperforms our own baselines, demonstrating the benefits of selective region weighting.

considered correct. We evaluate on multiple choice task because its evaluation is much less ambiguous than the open-ended response task, though our method could be applied to the latter by treating the most common or likely M responses as a large M-way multiple choice task. We perform detailed baseline comparisons on the validation set and report final scores on the test set.

We evaluate and analyze how much our region-weighting improves accuracy compared to using the whole image or only language (Tables 1, 2, 3) and show examples in Figure 8. We also perform a simple evaluation on a subset of images showing that relevant regions tend to have higher than average weights (Figure 6). We also show the advantage of our language model over other schemes (Table 4).

4.1. Comparisons between region, image, and language-only models

We compare our region selection model with several baseline methods, described below. All models use a 10% held-out from train for model selection.

Word-only: We train a network to score each answer purely from the language representation. This provides a baseline to demonstrate improvement due to image features, rather than just good guesses.

Word+Whole image: We concatenate CNN features computed over the entire image with the language features and score them using a three-layer neural network, essentially replacing the region-selection layer with features computed over the whole image.

Word+Uniform averaged region features: To test that region weighting is important, we also try uniformly averaging features across all regions as the image representation and train as above.

Word+Salient region weighting: We include a baseline where each region's weight is computed independently of the language component. We replace the inner product computation between vision and language features with an affine transformation that projects just the vision features down to a scalar, followed by a softmax over all regions. The layer's output is the weighted combination of concatenated vision and language features as before, but using the salient weights.

Table 1 shows the comparison of overall accuracy on the validation set, where it is clear our proposed model performs best. The salient weighting baseline alone showed noticeable improvement over the simpler whole image and averaging baselines. We noticed it performed similarly to the whole image baseline on localization dependent categories such as "what color" due to its inability localize on mentioned subjects, but performed similarly to the proposed model in scene and sport recognition questions due to its ability to highlight discriminative regions. We also include the best-performing LSTM question+image model on val from the authors of [1]. This model significantly underperforms even our much simpler baselines, which could be partly because the model was designed for open-ended answering and adapted for multiple choice.

We evaluate our model on the test-dev and test-standard partitions in order to compare with additional models from [1]. In Table 2, we include comparisons to the best-performing question+image based models from the VQA dataset paper [1], as well as a competitive implementation of the whole image+language baseline from Zhou et al. [24]. Our model was retrained on train+val data using the same held-out set as before for model selection. Our model significantly outperforms the baselines in the "others" category, which contains the majority of the question types that our model excels at.

Table 3 offers a more detailed performance summary across various question types, with discussion in the caption. Figure 8 shows a qualitative comparison of results, highlighting some of the strengths and remaining problems of our approach. These visualizations are created by soft masking the image with a mask created by summing the weights of each region and normalizing to a max of one. A small blurring filter is applied to remove distracting artifacts that occur from multiple overlapping rectangles. On color questions, localization of the mentioned object tends to be very good, which leads to more accurate answers. On questions such as "How many birds are in the sky?" the system cannot produce the correct answer but does focus on the relevant objects. The third row shows examples of how different questions lead to different focus regions. Notice how the model identifies the room as a bathroom in the third row by focusing on the toilet, and, when confirming that "kite" is the answer to "What is the woman flying over the beach?" focuses on the kite, not the woman or the beach.

In Figure 5, we show additional qualitative examples of how the region selection varies with question-answer pairs. In the first row, we see the model does more than simply match answer choices to regions. While it does find a matching green region, the corresponding confidence is still low. In addition, we see that irrelevant answer choices tend to have less-focused attention weightings. For example, the kitchen recognition question has most of its weighting on

Figure 5. Comparison of attention regions generated by various question-answer pairings for the same question. Each visualization is labeled with its corresponding answer choice and returned confidence. We show the highlighted regions for the top multiple choice answers and some unrelated ones. Notice that in the first example, while the model clearly identified a green region within the image to match the "green" option, the corresponding confidence was significantly lower than that of the correct options, showing that the model does more than just match answer choices with image regions.

Model	All	Y/N	Num.	Others
test-dev				
LSTM Q+I [1]	57.17	**78.95**	35.80	43.41
Q+I [1]	58.97	75.97	34.35	50.33
iBOWIMG [24]	61.68	76.68	**37.05**	54.44
Word+Region Sel.	**62.44**	77.62	34.28	**55.84**
test-standard				
iBOWIMG [24]	61.97	76.86	**37.30**	54.60
Word+Region Sel.	**62.43**	**77.18**	33.52	**56.09**

Table 2. Accuracy comparison on VQA test sets.

Figure 6. Example image with corresponding region weighting. Red boxes correspond to manual annotation of regions relevant to the question: "Are the people real?"

what appears to be a discriminative kitchen patch for the correct choice, whereas the "blue" choice appears to have a more evenly spread out weighting.

4.2. Region Evaluation

We set up an informal experiment to evaluate the consistency of our region weightings with respect to various types

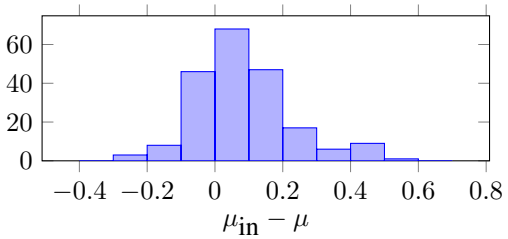

Figure 7. Histogram of differences between mean pixel weight within (μ_{in}) annotated regions and across the whole image (μ). Pixel weights are normalized by the maximum pixel weight. Often much more weight is assigned to the relevant region and very rarely much less.

of questions. We manually annotated 205 images from the validation set with bounding boxes considered relevant to answering the corresponding question. An example of the annotation and predicted weights can be seen in Figure 6. To evaluate, we compare the average pixel weighting within the annotated boxes with the average across all pixels. Pixel weighting was determined by cumulatively adding each region's selection weight to each of its constituent pixels. We observe that the the mean weighting within the annotated regions was greater than the global average in 148 of the instances (72.2%), often much greater and rarely much smaller (Figure 7).

We further investigate the effectiveness of ranking by our region scores in Figure 9 by retaining only the top K

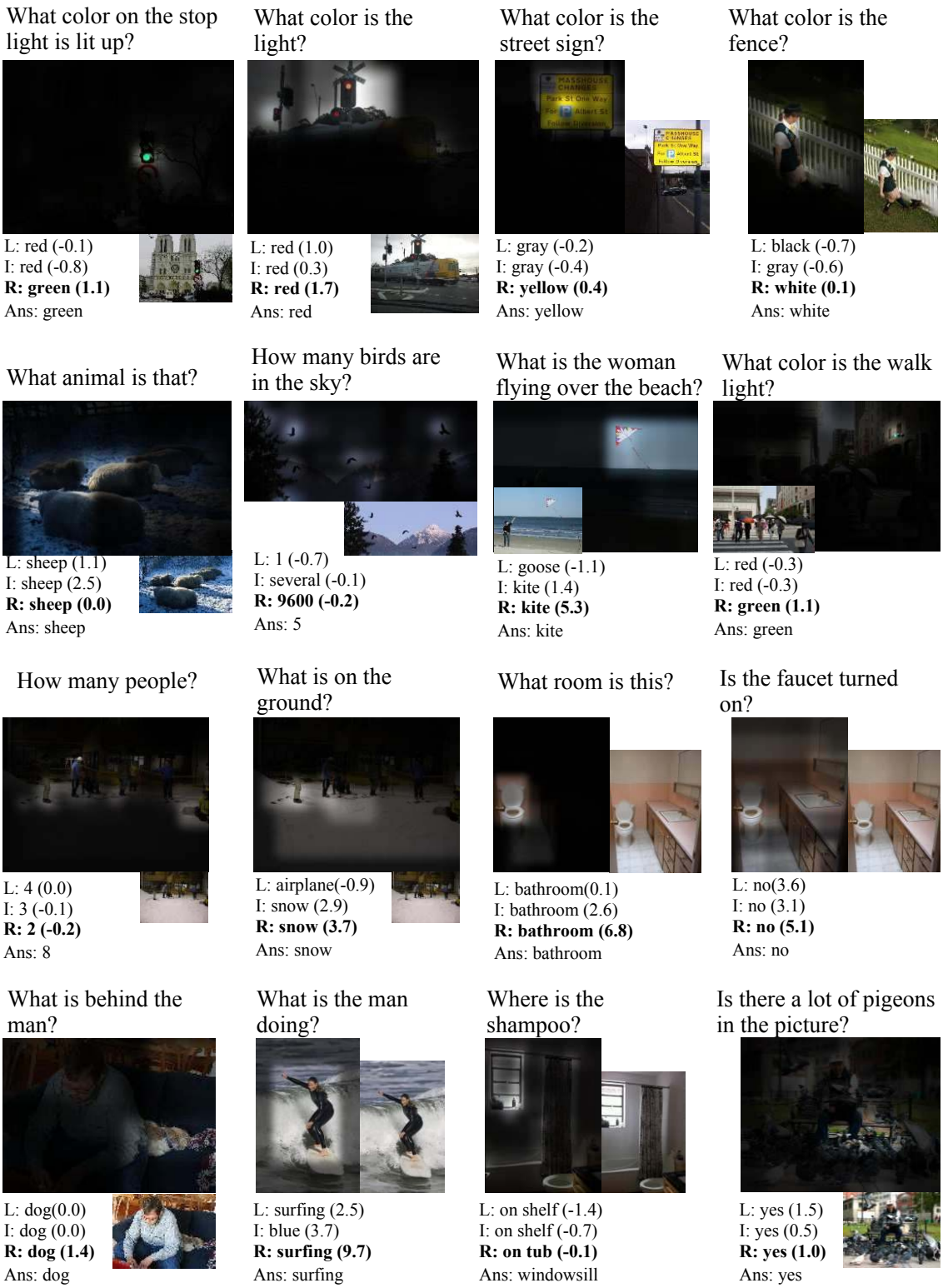

Figure 8. Comparison of qualitative results from Val. The larger image shows the selection weights overlaid on the original image (smaller). L: Word only model; I: Word+Whole Image; R: Region Selection. The scores shown are ground truth confidence - top incorrect. Note that the first row shows successful examples in which tight region localization allowed for an accurate color detection. In the third row, we show examples of how weighting varies on the same image due to differing language components.

	region	image	text	freq
overall	**58.94**	57.83	53.98	100.0%
is/are/was	**75.42**	74.63	75.00	33.3%
identify: what kind/type/animal	**52.89**	52.10	45.11	23.8%
how many	33.38	**36.84**	34.05	10.3%
what color	**53.96**	43.52	32.59	9.8%
interpret: can/could/does/has	**75.73**	74.43	**75.73**	4.6%
none of the above	45.40	44.04	**48.23**	4.1%
where	42.11	**42.43**	37.61	2.5%
why/how	26.31	28.18	**29.24**	2.2%
relational: what is the man/woman	**70.15**	67.48	56.64	2.0%
relational: what is in/on	54.78	**54.80**	45.41	1.8%
which/who	**43.97**	42.70	38.62	1.7%
reading: what does/number/name	**33.31**	31.54	30.84	1.6%
identify scene: what room/sport	**86.21**	76.65	61.26	0.9%
what time	**41.47**	37.74	38.64	0.8%
what brand	45.40	44.04	**48.23**	0.4%

Table 3. Accuracies by type of question on the validation set. Percent accuracy is shown for each subset for our region-based approach, classification using the whole image and question/answer text, and classification based only on text. We also show the frequency of each question type. Since there are 121,512 questions used for testing, there are hundreds or thousands of examples of even the rarest question types, so small gains are statistically meaningful. Overall, our region selection scheme outperforms use of whole images by 2% and text-only features by 5%. There is substantial improvement in particular types of questions. For example, questions such as "What is the woman holding?" are answered correctly 70% of the time vs. 67% for whole image and only 57% for text. "What color," "What room," and "What sport" also benefit greatly from use of image features and further from region weighting. Question types that have yes/no answers tend not to improve, in part because the prior is so reliable. E.g., someone is unlikely to ask "Does the girl have a lollipop?" if she is not so endowed. So "no" answers are unlikely and also more difficult to verify. We also note that reading questions ("What does the sign say?") and counting questions ("How many sheep?") are not greatly improved by visual features in our system because they require specialized processes.

Model	Accuracy (%)
Q+A (2-bin)	51.87
parsed(Q)+A (5-bin)	**53.98**

Table 4. Language model comparison. The 2-bin model is the concatenation of the question and answer averages. The parsed model uses the Stanford dependency parser to further split the question into 4 bins.

weighted regions (retained weights are L1 normalized) or only the Kth (1-hot weighting of Kth region). We observe that performance on color-type questions does not improve significantly beyond the first 10 regions, and that performance drops off sharply in the Kth-only experiment. This provides further evidence that our model is able to score relevant regions above the rest.

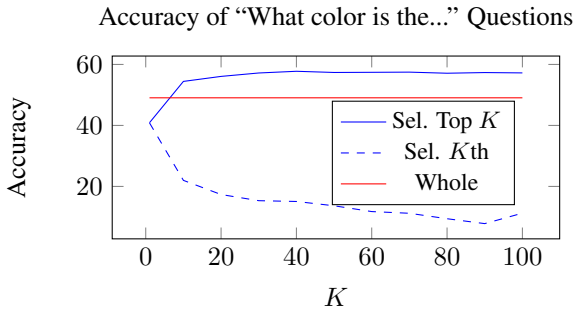

Figure 9. Plot of color-based question accuracy with varying number of regions sampled at every 10. The experiment was run on a 10% held-out set on train. We look at using the weighted average of only the top K scoring regions, as well as only the Kth. We include the whole image baseline's accuracy in this category for comparison.

4.3. Language Model

We also compare our parsed and binned language model with a simple two-binned model (one bin averages word2vec of question words; the other averages answer words) to justify our more complex representation. Each model is trained on the train set and evaluated on the validation set of the VQA real-images subset. The comparison results are shown in Table 4 and depict a significant performance improvement using the parsing.

5. Conclusion

We presented a model that learns to select regions from the image to solve visual question answering problems. Our model outperforms all baselines and existing work on the MS COCO VQA multiple choice task [1], with substantial gains for some questions such as identifying object colors that require focusing on particular regions. One direction for future work is to learn to perform specialized tasks such as counting or reading. Other directions are to incorporate and adapt pre-trained models for object and attribute detectors or geometric reasoning, or to use outside knowledge sources to help learn what is relevant to answer difficult questions. We are also interested in learning where to look to find small objects and to recognize activities.

6. Acknowledgements

This work is supported by NSF CAREER award 1053768, NSF Award IIS-1029035, and ONR MURI Awards N00014-10-1-0934 and N00014-16-1-2007. We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Tesla K40 GPUs used for this research.

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. 1, 2, 4, 5, 6, 8

[2] J. Ba, K. Swersky, S. Fidler, and R. Salakhutdinov. Predicting deep zero-shot convolutional neural networks using textual descriptions. In *International Conference on Computer Vision (ICCV)*, 2015. 2

[3] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman. Return of the devil in the details: Delving deep into convolutional nets. In *British Machine Vision Conference*, 2014. 4

[4] X. Chen and C. Lawrence Zitnick. Mind's eye: A recurrent visual representation for image caption generation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015. 2

[5] M.-C. De Marneffe, B. MacCartney, C. D. Manning, et al. Generating typed dependency parses from phrase structure parses. In *Proceedings of LREC*, volume 6, pages 449–454, 2006. 4

[6] J. Donahue, L. A. Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell. Long-term recurrent convolutional networks for visual recognition and description. *arXiv preprint arXiv:1411.4389*, 2014. 2

[7] H. Fang, S. Gupta, F. Iandola, R. Srivastava, L. Deng, P. Dollár, J. Gao, X. He, M. Mitchell, J. Platt, et al. From captions to visual concepts and back. *arXiv preprint arXiv:1411.4952*, 2014. 2

[8] X. Glorot and Y. Bengio. Understanding the difficulty of training deep feedforward neural networks. In *International conference on artificial intelligence and statistics*, pages 249–256, 2010. 4

[9] X. Glorot, A. Bordes, and Y. Bengio. Deep sparse rectifier neural networks. In *International Conference on Artificial Intelligence and Statistics*, pages 315–323, 2011. 4

[10] Y. Gong, L. Wang, M. Hodosh, J. Hockenmaier, and S. Lazebnik. Improving image-sentence embeddings using large weakly annotated photo collections. In *Computer Vision–ECCV 2014*, pages 529–545. Springer, 2014. 2

[11] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015. 4

[12] A. Karpathy and L. Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In *CVPR*, 2015. 2

[13] J. Mao, W. Xu, Y. Yang, J. Wang, Z. Huang, and A. Yuille. Deep captioning with multimodal recurrent neural networks (m-rnn). *ICLR*, 2015. 2

[14] J. Mao, W. Xu, Y. Yang, J. Wang, and A. L. Yuille. Explain images with multimodal recurrent neural networks. *NIPS Deep Learning Workshop*, 2014. 2

[15] M. F. Mateusz Malinowski, Marcus Rohrbach. Ask your neurons: A neural-based approach to answering questions about images. In *ICCV*, 2015. 2

[16] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013. 2, 3

[17] M. Ren, R. Kiros, and R. Zemel. Exploring models and data for image question answering. *arXiv preprint arXiv:1505.02074v3*, 2015. 2

[18] 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)*, pages 1–42, April 2015. 2, 4

[19] S. Sukhbaatar, A. Szlam, J. Weston, and R. Fergus. Weakly supervised memory networks. *CoRR*, abs/1503.08895, 2015. 2

[20] A. Vedaldi and K. Lenc. Matconvnet – convolutional neural networks for matlab. 2015. 4

[21] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan. Show and tell: A neural image caption generator. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015. 2

[22] K. Xu, J. Ba, R. Kiros, A. Courville, R. Salakhutdinov, R. Zemel, and Y. Bengio. Show, attend and tell: Neural image caption generation with visual attention. *arXiv preprint arXiv:1502.03044*, 2015. 2

[23] L. Yu, E. Park, A. C. Berg, and T. L. Berg. Visual madlibs: Fill in the blank image generation and question answering. *arXiv preprint arXiv:1506.00278*, 2015. 2

[24] B. Zhou, Y. Tian, S. Sukhbaatar, A. Szlam, and R. Fergus. Simple baseline for visual question answering. *arXiv preprint arXiv:1512.02167*, 2015. 5, 6

[25] C. L. Zitnick and P. Dollár. Edge boxes: Locating object proposals from edges. In *Computer Vision–ECCV 2014*, pages 391–405. Springer, 2014. 4