In Proceedings of the NIPS 2017 workshop on Visually-Grounded Interaction and Language (ViGIL), December 2017.

Ensembling Visual Explanations for VQA

Nazneen Fatema Rajani
Department of Computer Science
University of Texas at Austin
nrajani@cs.utexas.edu

Raymond J. Mooney
Department of Computer Science
University of Texas at Austin
mooney@cs.utexas.edu

Abstract

Explanations make AI systems more transparent and also justify their predictions. The top-ranked Visual Question Answering (VQA) systems are ensembles of multiple systems; however, there has been no work on generating explanations for such ensembles. In this paper, we propose different methods for ensembling visual explanations for VQA using the localization maps of the component systems. Our crowd-sourced human evaluation indicates that our ensemble visual explanation is superior to each of the individual system's visual explanation, although the results vary depending on the individual system that the ensemble is compared against as well as the number of individual systems that agree with the ensemble model's answer. Overall, our ensemble explanation is better 63% of the time when compared to any individual system's explanation. Our algorithm is also efficient and scales linearly in the number of component systems in the ensemble.

1 Introduction

Visual Question Answering (VQA) [2] requires both language and image understanding, language grounding capabilities, as well as common-sense knowledge. A variety of methods to address these challenges have been developed in recent years [1, 4, 11, 7, 3]. The vision component of a typical VQA system extracts visual features using a deep convolutional neural network (CNN), and the linguistic component encodes the question into a semantic vector using a recurrent neural network (RNN). An answer is then generated conditioned on the visual features and the question vector. The top performing VQA systems are ensembles of neural networks that perform substantially better than any of the underlying individual models [4].

Although there have been several innovative and ground-breaking ideas deployed to solve VQA problems, the current state-of-the-art on real-world images is still approximately 15 points behind human accuracy.[1] One way to reduce this gap in performance would be to analyze how various neural architectures arrive at their predicted answers, and then design heuristics or loss functions that overcome the shortcomings of current networks. This has led to some work in generating explanations that help interpret the decisions made by CNNs [5, 6, 8]. However, previous work focuses on generating explanations for individual models even though the top performing systems on various computer vision and language tasks are ensembles of multiple models. This motivated us to explore the problem of generating explanations for an ensemble using explanations from underlying individual models as input. In this paper, we focus on ensembling *visual* explanations for VQA.

Deep learning models have been shown to attend to relevant parts of the image when answering a question [5]. The regions of an image on which a model focuses can be thought of as a visual explanation for that image-question (IQ) pair. The Guided Grad-CAM algorithm [10] highlights the regions in an image that the model focuses on by generating a heat-map with intensity gradients. We adapt the Guided Grad-CAM approach to generate heat-map visualizations for three different VQA

[1] Based on the performance reported on the CodaLab Leader-board and human performance reported on the task in [2].

31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA.

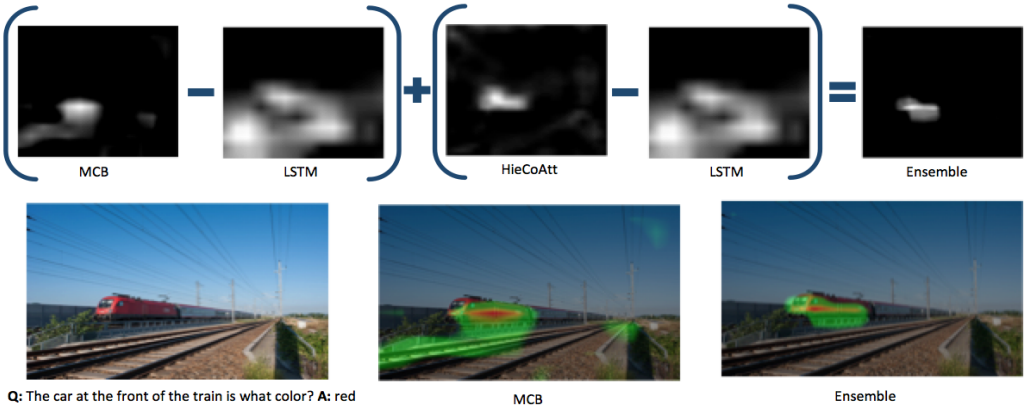

Figure 1: The top row shows the process of ensembling visual explanation for an IQ pair when the ensemble model agrees with the MCB and HieCoAtt models (ans: "red") and disagrees with the LSTM model (ans: "white"). The bottom row shows the reference IQ pair and the MCB vs ensemble visual explanation. The feature map is normalized to obtain the final ensemble visualization.

systems – LSTM [2], MCB [4] and HieCoAtt [7]. By manually analyzing some of the visualizations from each of these systems, we found that all of them had some degree of noise with high variance depending on the IQ pair under consideration. We also observed that there was high variance across visualizations for different models even when they agreed on the answer for a given IQ pair. This motivated us to ensemble visualizations of the individual models such that the ensembled visual explanation: (i) aggregates visualizations from appropriate regions of the image, (ii) discounts visualizations from regions that are not relevant, (iii) reduces noise as much as possible and (iv) is superior to any individual system's visualization on a manual evaluation. Results from a crowd-sourced human evaluation indicate that, on an average, our visual explanation ensemble is superior to each of the individual system's visual explanation 63% of the time, while an individual system's explanation is better only 35% of the time, and there is either no agreement or the answer is incorrect the remaining 2% of the time.

2 Algorithm for Ensembling Visual Explanations

Our strategy for ensembling visual explanations for VQA depends on the individual component models' answer and visualization for a given image-question (IQ) pair. We first build an ensemble model that uses Stacking With Auxiliary Features (SWAF) [9] to combine outputs of the three component systems. We then use our best ensemble model as the system for which we set out to generate visual explanations. We do this by ensembling the visual explanations of the component systems that agree with the ensemble answer for an IQ pair. We propose two heuristics for generating visual explanation ensemble – Weighted Average (WA) and Penalized Weighted Average (PWA).

2.1 Weighted Average Ensemble Explanation

The Weighted Average (WA) ensemble explanation is calculated as follows:

$$E_{i,j} = \begin{cases} \frac{1}{K} \sum_{k \in K} w_k A^k_{i,j}, & \text{if } A^k_{i,j} \geq t \\ 0, & \text{otherwise} \end{cases} \quad \text{subject to} \sum_{k \in K} w_k = 1 \tag{1}$$

Here, E is the localization map of the ensemble, i and j are used to index into the feature map entries, $K = 3$ is the number of component systems, w^k and A^k are the weights and localization feature maps respectively for each of the component systems, and t is a thresholding value for the feature maps. Thresholding the pixel values for the maps before or after averaging worked well for reducing noise as well as eliminating several low-intensity regions that arose as a result of combining multiple noisy maps. A weighted combination of the component feature maps worked better than using equal weights across all component systems. Since we had access to the performance of the component systems on validation data, we were able to rank them and use these ranks to determine weights for combining the feature maps. We weigh the maps of the component systems proportional to their performance on the validation set and subject to the constraint that the weights sum to one.

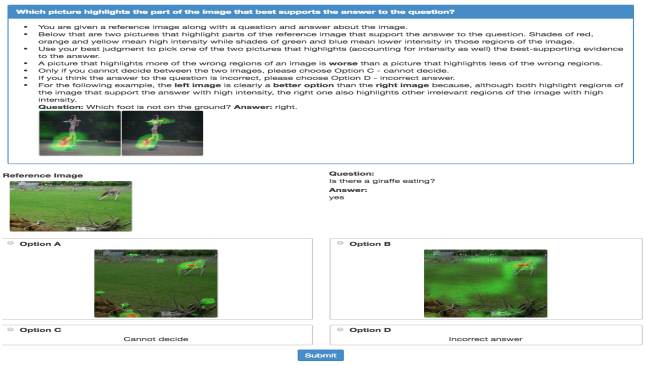

Figure 2: AMT interface for evaluating visual explanation.

2.2 Penalized Weighted Average Ensemble Explanation

The Penalized Weighted Average (PWA) ensemble explanation is calculated as follows:

$$E_{i,j} = \begin{cases} \frac{1}{K} \sum_{k \in K} w_k \left(A^k_{i,j} - I_{i,j} \right), & \text{if } \left(A^k_{i,j} - I_{i,j} \right) \geq t \\ 0, & \text{otherwise} \end{cases} \quad \text{subject to } \sum_{k \in K} w_k = 1 \tag{2}$$

Here, I is the localization map of the component system that does not agree with the ensemble. This assumes that the system that does not agree with the ensemble on an answer is highlighting regions of the image that are not relevant and so we down-weight those regions in the localization map for the ensemble. Another variation we explored is forcing the component model that does not agree with the other models to produce a localization map for the alternate answer picked by the ensemble. We then calculate the ensemble localization in a way similar to the previous section, when all systems agree on the output.

2.3 Agreement with N systems

The visual explanation ensemble can be generated for N number of component models using Equations 1 and 2 and it scales linearly with N. In this paper, we consider three component VQA systems and there are three scenarios that arise depending on whether the ensemble model agrees with all three, any two, or only one of the component systems on the answer for an IQ pair. Figure 1 shows the process of ensembling visual explanations for an IQ pair for one of the scenarios. For all the different scenarios, we first generate a gray-scale GradCam visualization using the approach described in [10] for each of the component systems. In their approach, given an image and category, the image is forward propagated through the CNN part of the model. The gradients are set to zero for all categories except the one under consideration, which is set to 1. This signal is then backpropagated to the convolutional feature maps of interest and is combined to compute the localization map. Thereafter, we generate the ensemble explanation using the aforementioned approaches depending on the scenario under consideration.

We observed that our ensemble model for VQA agreed with all the *three* systems on the answer for an IQ pair for approximately half of the VQA test set. We use the weighted average approach for generating the ensemble visualization map. On approximately one-fourth of the outputs on the test set, our ensemble model agreed with exactly *two* component systems. For this scenario, we combine the localization feature maps of the two systems using both the weighted average and the penalized weighted average approaches by ignoring and down-weighting the system that does not agree with the ensemble's answer respectively. When the ensemble model's output agreed with only *one* system's output, we generate the ensemble localization map in two ways. First, the ensemble localization was set equal to the localization of the system it agrees with, minus the localization of the systems it does not agree with, as in Equation 2. Second, we force the systems that do not agree with the ensemble to produce localization maps for the answer produced by the ensemble and then use those maps to calculate the ensemble localization map using Equation 1.

3 Experimental Results and Discussion

We used crowd-sourced human evaluation to compare the visual explanation of the ensemble to the individual systems' explanations. We showed two visual explanations side-by-side to workers on Amazon Mechanical Turk (AMT) along with the IQ pair as well as the ensemble model's answer and ask them "Which picture highlights the part of the image that best supports the answer to the question?", as shown in Figure 2. One image is the localization map of the ensemble while the other

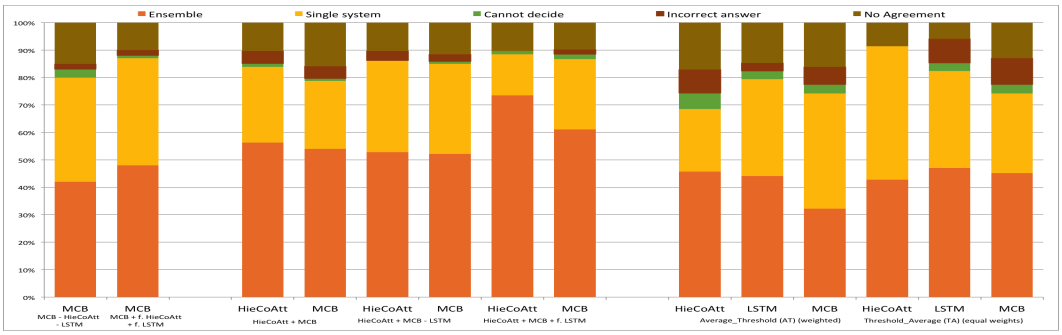

Figure 3: The figure shows results obtained when one system (MCB) agrees, two systems (MCB and HieCoAtt) agree and when all three systems (MCB, HieCoAtt and LSTM) agree with the ensemble's output respectively from left to right. The y-axis indicates the percentage of instances for each of the agreed upon options chosen by Turkers as well as when there was no agreement among the Turkers. The x-axis shows the individual system that the ensemble was compared to. Below that is the label indicating how the ensemble localization map was calculated; f. stands for forced output that agrees with the ensemble, AT stands for averaging the localization maps followed by thresholding and TA does the reverse, first thresholds the maps and then averages them.

one is the localization map of one of the systems that the ensemble agrees with, selected at random. We provide detailed instructions along with an example to show what a good visualization looks like. Apart from the two images as options, we also give two more options – "cannot decide" and "wrong answer" so that we can obtain more insights from the results.

Three workers evaluated each of the 100 random instances of IQ pairs for each of the different variations discussed in Section 2. We then aggregate the options chosen by the Turkers using voting and when there is no agreement among workers, we classify those instances under the "no agreement" category. Figure 3 shows the results obtained when only one system, two systems, and all three systems agree on the output of the ensemble. We used a threshold of 0.25 for all the cases and scenarios within each case. The pixel intensities > 0.25 are normalized to lie between zero and one. All the localization maps are converted from gray-scale to heat-maps based on pixel intensity ranges as a final step before evaluation. We found that, on an average, the Turkers considered our ensemble's explanation superior to any individual model's visualization 63% of time.

For the case when only one system agrees with the ensemble, we considered the IQ pairs in which only the top-ranked MCB system agrees with the ensemble and compare its localization map to the ensemble localization map obtained using our algorithm. We found that subtracting the thresholded localization maps of the LSTM and HieCoAtt systems worked slightly better than averaging the thresholded localization maps obtained by forcing the LSTM and HieCoAtt systems to produce maps for answers produced by the MCB model. For the case when two systems agree with the ensemble, we consider instances when the MCB and HieCoAtt models agree. We show results for three different scenarios for generating the ensemble localization maps – just using MCB and HieCoAtt localization maps, using the LSTM localization map, and finally using the localization map produced by LSTM for the output that agrees with the other two systems. For the case when all the three systems agree with the ensemble, we found that using the weighted average worked better than using equal weights when compared to the HieCoAtt and LSTM localization maps but performed slightly worse when compared to the MCB maps. We also experimented with taking the union and intersection of the component localization maps for various scenarios, but found that they were either too noisy or too minimal and thus do not report them in this paper.

4 Conclusions

Visual explanations can help us understand the decisions made by VQA systems and thereby aid error analysis and help build trust with human users. We have presented the first approaches to ensembling visual explanations for VQA. We proposed several methods for combining the heat-maps of multiple systems to produce improved ensemble explanations. We thoroughly examined various cases and scenarios based on the number of systems that agree with the ensemble model's output. Our evaluation using Mechanical Turk indicated that our visual explanation ensemble is superior to each of the component system's visual explanation. Our proposed approach for ensembling visual explanation is linearly scalable in the number of component systems.

Acknowledgement

This research was supported by the DARPA DEFT program under AFRL grant FA8750-13-2-0026.

References

[1] Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Dan Klein. Learning to compose neural networks for question answering. In *Proceedings of the Conference on Natural language learning (NAACL2016)*, pages 1545–1554, 2016.

[2] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C. Lawrence Zitnick, and Devi Parikh. VQA: Visual Question Answering. In *The IEEE International Conference on Computer Vision (ICCV2015)*, December 2015.

[3] Kan Chen, Jiang Wang, Liang-Chieh Chen, Haoyuan Gao, Wei Xu, and Ram Nevatia. ABC-CNN: An attention based convolutional neural network for Visual Question Answering. *arXiv preprint arXiv:1511.05960*, 2015.

[4] Akira Fukui, Dong Huk Park, Daylen Yang, Anna Rohrbach, Trevor Darrell, and Marcus Rohrbach. Multimodal Compact Bilinear pooling for Visual Question Answering and Visual Grounding. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP2016)*, 2016.

[5] Yash Goyal, Akrit Mohapatra, Devi Parikh, and Dhruv Batra. Towards Transparent AI Systems: Interpreting Visual Question Answering Models. In *International Conference on Machine Learning (ICML) Workshop on Visualization for Deep Learning*, 2016.

[6] Lisa Anne Hendricks, Zeynep Akata, Marcus Rohrbach, Jeff Donahue, Bernt Schiele, and Trevor Darrell. Generating Visual Explanations. *arXiv preprint arXiv:1603.08507*, 2016.

[7] Jiasen Lu, Jianwei Yang, Dhruv Batra, and Devi Parikh. Hierarchical question-image co-attention for visual question answering. In *Advances In Neural Information Processing Systems (NIPS2016)*, pages 289–297, 2016.

[8] Dong Huk Park, Lisa Anne Hendricks, Zeynep Akata, Bernt Schiele, Trevor Darrell, and Marcus Rohrbach. Attentive explanations: Justifying decisions and pointing to the evidence. *arXiv preprint arXiv:1612.04757*, 2016.

[9] Nazneen Fatema Rajani and Raymond J. Mooney. Stacking With Auxiliary Features. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence (IJCAI2017)*, Melbourne, Australia, August 2017.

[10] Ramprasaath R Selvaraju, Abhishek Das, Ramakrishna Vedantam, Michael Cogswell, Devi Parikh, and Dhruv Batra. Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization . 2017.

[11] Huijuan Xu and Kate Saenko. Ask, Attend and Answer: Exploring question-guided spatial attention for visual question answering. In *European Conference on Computer Vision (ECCV2016)*, pages 451–466. Springer, 2016.