Counting in Visual Question Answering

a concept detector based approach

M.H.T. de Boer
TNO and Radboud University
Oude Waalsdorperweg 63
2597AK, The Hague
maaike.deboer@tno.nl

S. Reitsma
Radboud University
Toernooiveld 212
6525 EC, Nijmegen
steven@properchaos.nl

K. Schutte
TNO
Oude Waalsdorperweg 63
2597AK, The Hague
klamer.schutte@tno.nl

ABSTRACT

Visual Question Answering is a field that combines computer vision techniques and natural language processing techniques. One of the most challenging question types in this field is counting, such as *How many sheep are in this picture*. In this paper, we focus on counting questions and improve upon the state-of-the-art method DPPnet. We train concept detectors on the MSCOCO dataset and use these detectors in addition to the pre-final layer from the original visual network. Additionally, we use a postprocessing technique to output the right type of answer to each type of question. Both the concept detectors, and the postprocessing slightly improve performance and is usable on current state-of-the-art methods.

CCS Concepts

•**Information systems** → *Information retrieval;* Image search;

Keywords

Visual Question Answering; Concept Detectors; Neural Networks

1. INTRODUCTION

One of the most common forms of visual question answering is one where a system answers natural language questions posed by human users about images [12]. In practice this could take recent developments such as Google Now, Siri and Cortana a step further by not only being able to answer questions on general topics that are searchable on the web, but also on the local user context using e.g. a smartphone's camera. This could be especially useful for visually impaired users, who can take a picture using their smartphone and ask their device questions about the local scene, such as *where is an empty seat in this train?* or *is there a pedestrian crossing here?*.

DOI: 10.1145/1235

A dataset that enables image question answering is the VisualQA task [1], set up by VirginiaTech and Microsoft Research after the release of the Microsoft Common Objects in Context (*MSCOCO*) dataset [7]. This MSCOCO dataset consists of more than 250,000 images. In the VisualQA task three questions for each image were posed together with 10 human answers to each question. The type of answers to the questions can be categorized into three major categories: *closed (yes / no)*, *numerical answers*, *categorical answers*.

In this paper, we focus on the Visual QA questions with numerical answers. Current state-of-the-art methods have low performance for this type of question compared to the questions with closed and categorical answers. We propose to count the amount of certain objects using concept detectors with object segmentations. In addition, we introduce a post-processing method to provide an answer that is in the right category.

Results show that the use of concept detectors improves performance. Post-processing slightly improves performance further.

2. RELATED WORK

According to Wu et al. [12], visual question answering solutions can be put into four categories: joint embedding, attention, compositional, knowledge bases. We focus on the first and biggest category. Approaches in this category use deep learning networks for both the image and the question and combine these in a classifier such as another neural network to predict the most probable answer. This is used in the baseline for the VisualQA task [1], but current state-of-the-art and a good performer in the task is DPPnet [8]. DPPnet uses the state-of-the-art VGGnet network [9], trained on the ImageNet images in the ILSVC-2012 [3], to understand the image. This pre-trained model is finetuned using the MSCOCO dataset [7] to create a network that is tailored to the VisualQA task. Instead of the 1000 concepts from ImageNet, the 4096 features in the pre-final layer are used. To understand the question, Gated Recurrent Units (GRU) are used. The question model is pre-initialized using the *skip-thought* vector model [5] which is trained on the BookCorpus dataset [13], containing 74 million sentences. To generate an answer DPPnet uses a dynamic parameter layer to combine the image and question features. The image features are used as input for this layer and the weights are determined by the question features using a hashing function [2]. Recently, the Multimodal Compact Bilinear Pooling (MCB) [4] further improves performance. This model combines the joint embedding with attention. The winning

submissions in the VisualQA challenge, linked to the VisualQA task[1], combine joint embedding with some type of attention model to focus on a specific part of the image.

3. METHOD

In our method, we build upon DPPnet [8]. We use *concept detectors* in addition to the 4096 pre-final layer for the part of the network with the dynamic parameter layer. To train the concept detectors we use masked images of the ground truth annotations of the MSCOCO dataset. The concept detectors can be applied on 1. the full image or 2. object proposals within the image and sum the activations to count. Additionally, we add *postprocessing* repair to make sure that only the answers of the same category as the question are proposed.

3.1 Concept Detection

We train concept detectors using the ground truth annotation of the MSCOCO dataset [7] for each of the 80 classes. These 80 classes are tailored to the test set, whereas the 1000 ImageNet concepts are not and thus we expect better performance for these classes. We use a pretrained GoogLeNet model [11] based on the Inception architecture. GoogLeNet was chosen for its high accuracy and the fact that it uses 12 times fewer parameters and thus fewer VRAM than the next-best ImageNet submission. From the ground truth annotations we use a masked version of each separate segmentation with a black background. This masked segmentation is fed through the convolutional neural network to obtain its features, similarly to the normal process for the unmasked images. A fixed amount of segmentations is chosen (25 in our experiments) and if an image has fewer segmentations, the concatenated feature vector is zero-padded.

The network is trained on the segmentations using gradient descent with Nesterov momentum [10] for 25 epochs. For the first 10 epochs, the weights of the convolutional layers are locked to prevent the noisy gradients from the randomly initialized fully-connected layers from changing the pretrained weights too much. Cross-entropy loss is used and Top 1 accuracy is used for validation. The segmentation masks are stretched to use the entire 224×224 image space (aspect ratio is retained), which improves validation accuracy from 57% to 87%. This removes scale variance and reduces overfitting. Furthermore, segmentations that have a surface smaller than 500 pixels are removed from training as they provide no meaningful information. The biases in the first convolutional layer are set to 0 to ensure the black background causes no activations. Finally, since the class balance is skewed – the most prevalent class occurs 185,316 times, while the rarest class occurs only 135 times – the amount of data per epoch is limited to 5000 per class. Note that if a class has more than 5000 samples, each epoch different data is shown to the network. Effectively, this means samples in underrepresented classes will be shown to the network more than samples in large classes.

These concept detectors can be used on either the full image or the object proposals within the image. Using the full image, we expect a deterioration of the activation with less objects (i.e. less pixels firing on the object), for which DPPnet can learn that for example an activation of 0.6 for a certain concept will most probably resemble a number 4. When we use object proposals, we expect that the activation will be high for objects that are in that proposal and summing over the proposals will resemble actual counting.

The object proposals can be obtained in several ways. First, ground truth segmentations as present in the MSCOCO dataset could be used. These segmentations are, however, not available in many datasets, so automatic object proposals can be obtained using *Edgebox* [14] or *Deepbox* [6] (with non-maximum supression), which are state-of-the-art segmentation methods. Edgebox generates bounding box object proposals, which makes it especially suited for objects that are rectangular. Often, the Edgebox strategy generates hundreds of object candidates. The algorithm scores and sorts these according to the number of contours that are wholly contained within the image. Deepbox [6] uses a different scoring metric: it trains a convolutional neural network that reranks the proposals that Edgebox made. Using Deepbox, the same recall is achieved with four times less proposals.

3.2 Postprocessing repair

For some questions, such as *how many...* questions, we know that the answer should be numerical. Often, the network will predict other answers as well, such as the string equivalents of the numeric digits, e.g. *one* instead of *1*. For questions that start with *are there...*, *does this...*, and so on, we expect as an answer *yes*, *no* or a word that exists in the question. For example, the question *does this image contain a cat?* always has to be answered by either yes or no, while the question *is there a cat or a dog in this image?* should be answered with either *cat*, *dog*, *yes* or *no*. Using a simple rule-based program, questions that start with *how many* always get the numerical answer that generates the highest softmax response in the network and the questions that have a closed answer are processed as explained above.

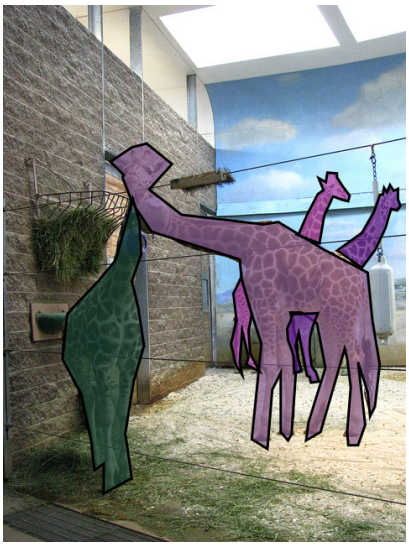
4. RESULTS

In our experiments, we first investigate the optimal performance gain using segmentations. We use the validation set to set the ground truth annotations (which are not available in the test set), the Deepbox and Edgebox methods. These methods do not yet use the concept detectors, but have as input a vector of 25*4096 + 4096, which resembles the output of the pre-final layer of the VGGnet on each of the 25 object proposals and the full image. Furthermore, we test classification and regression as last layer. Classification is similar to DPPnet, using a softmax over all possible answers and for *regression* we have one output node outputting a number. Because the use of finetuning and the large dynamic parameter layer require at least 12GB of VRAM (i.e. a GTX Titan X or Tesla M40), we remove the finetuning and the large dynamic parameter layer to make the network fit into 6GB of VRAM, enabling its use on state-of-the-art hardware. In the downsized network, the hash size is decreased from 40000 to 10240 and the amount of linear units in the dense part of the network is decreased from 2000 to 1024. Afterwards, we use the finetuned and the full network on our best run to make a submission in the VisualQA challenge. The results of these experiments are shown in Table 1. These results show that object proposals can increase performance by 3%. Classification works slightly better than regression for the counting questions and overall. The ground truth annotations obviously have highest performance, but the bounding boxes by

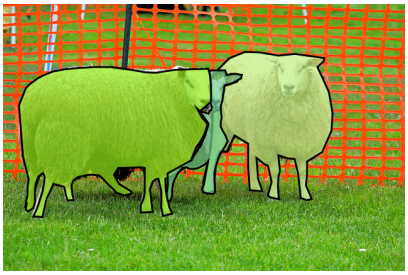
[1]http://visualqa.org/challenge.html

Method	All	Yes/No	Number	Other
DPPnet (downsized)	51.94	78.34	33.66	36.77
Ground truth annotations	**52.29**	78.34	**36.46**	36.77
Ground Truth Regression	52.23	**78.40**	34.84	**37.01**
Edgebox	52.08	78.34	34.97	36.77
Deepbox	52.16	78.34	35.36	36.77

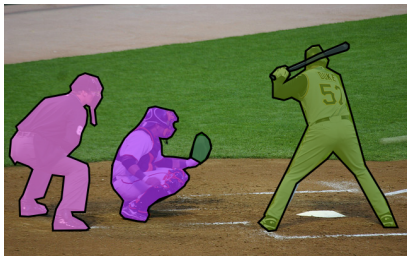
Table 1: Evaluation of segmentation methods on val2014.



(a) *How many giraffes are in this picture?*
DPPnet: 2
Ground truth annotations: 4
Edgeboxes: 2
Deepboxes: 3



(b) *How many sheep are in this picture?*
DPPnet: 1
Ground truth annotations: 3
Edgeboxes: 2
Deepboxes: 2



(c) *How many players can you see?*
DPPnet: 2
Ground truth annotations: 3
Edgeboxes: 3
Deepboxes: 3

Figure 1: Comparison of segmentation methods. Overlays are the ground truth proposals.

Deepbox can gain 2% performance and are slightly better than those produced by the Edgebox system. To gain some insight, we show a few images with the answers to a numerical question for the different methods in Figure 1. In 1a) and 1b), we see that automatic segmentation techniques such as Edgeboxes and Deepboxes have trouble segmenting the objects, while in 1c) this seems to be no problem. This makes sense when realizing that Edgeboxes and Deepboxes create rectangular object proposals, which suits the 1c) very well, but 1a) and 1b) less so.

Based on these results, we continue with the classification method and the Deepboxes, because the ground truth annotations are not available in the test set. We now use the sum the concept detector scores over all 25 object proposals and concatenate this vector to the 4096 vector of the original pre-final layer. In the full image runs, the concept detector scores over the whole image are concatenated with the 4096 vector. Results are shown in Table 2. Interestingly, using the full image is better than using the object proposals. As indicated before the deterioration of the concept detectors scores might be more insightful for the DPPnet system compared to the top 25 rectangular object proposals. The postprocessing repair slightly improves performance. In Figure 2 we can see some example questions and images with the answers given by the regular DPPnet and the DPPnet with concept detector information. In 2a), the bias of DPPnet towards more often occuring answers can be seen. Using concept detectors, the answer is closer to the truth, but still not correct. In 2b) and 2c), we view the disadvantage of building a scale invariant system. The concept detector activations for both images are almost equal, caused by the larger objects in 2c) compared to 2b). In the context of the VQA challenge, using concept detectors as an additional input to the classification network scores 5th place out of 30 on numerical answers, tested on the test-standard dataset split.

5. CONCLUSIONS

Using concept detectors to count in a visual question answering task improves performance upon only using the features. The postprocessing further improves performance. Object proposals intuitively should increase performance, but with current state-of-the-art methods, performance on the full image is higher. The top performing methods, such as Multimodal Compact Bilinear Pooling (MCB) [4], could use the concept detectors and postprocessing to improve their system.

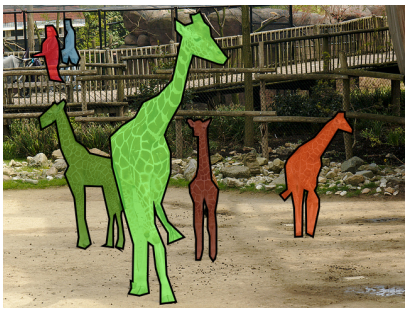
6. REFERENCES

[1] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra,

Method	All	Yes/No	Number	Other
LSTM Question + Image (baseline from Antol et al. [1])	53.74	78.94	35.24	36.42
DPPnet (finetuned, no downsizing from Noh et al. [8])	57.22	80.71	37.24	41.69
DPPnet (downsized)	56.11	79.88	36.81	40.18
Concept detectors on Deepbox segm. (downsized)	56.13	79.99	36.87	40.16
Concept detectors on full image (downsized)	56.34	79.99	37.31	40.45
Concept detectors on full image (downsized, +pp repair)	56.45	80.03	37.46	40.66
Concept detectors on full image (finetuned, no downsizing, +pp repair)	**58.01**	**80.89**	**38.03**	**42.44**

Table 2: Results on test-dev2015.

(a) *How many apples are in the picture?*
DPPnet: 3
Concept detectors: 8
Ground truth: 11



(b) *How many giraffes can you see?*
DPPnet: 2
Concept detectors: 4
Ground truth: 4

(c) *How many giraffes are there?*
DPPnet: 2
Concept detectors: 4
Ground truth: 3

Figure 2: Ground truth annotations and questions for some images in the MSCOCO dataset

C. L. Zitnick, and D. Parikh. Vqa: Visual question answering. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2425–2433, 2015.

[2] W. Chen, J. T. Wilson, S. Tyree, K. Q. Weinberger, and Y. Chen. Compressing neural networks with the hashing trick. In *Proceedings of The 32nd International Conference on Machine Learning*, pages 2285âĂŞ2294, 2015.

[3] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition, 2009. CVPR 2009.*, pages 248–255. IEEE, 2009.

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

[5] R. Kiros, Y. Zhu, R. R. Salakhutdinov, R. Zemel, R. Urtasun, A. Torralba, and S. Fidler. Skip-thought vectors. In *Advances in Neural Information Processing Systems*, pages 3276–3284, 2015.

[6] W. Kuo, B. Hariharan, and J. Malik. Deepbox: Learning objectness with convolutional networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2479–2487, 2015.

[7] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *European Conference on Computer Vision*, pages 740–755. Springer, 2014.

[8] H. Noh, P. H. Seo, and B. Han. Image question answering using convolutional neural network with dynamic parameter prediction. *arXiv preprint arXiv:1511.05756*, 2015.

[9] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

[10] I. Sutskever, J. Martens, G. Dahl, and G. Hinton. On the importance of initialization and momentum in deep learning. In *Proceedings of the 30th international conference on machine learning (ICML-13)*, pages 1139–1147, 2013.

[11] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–9, 2015.

[12] Q. Wu, D. Teney, P. Wang, C. Shen, A. Dick, and A. v. d. Hengel. Visual question answering: A survey of methods and datasets. *arXiv preprint arXiv:1607.05910*, 2016.

[13] Y. Zhu, R. Kiros, R. Zemel, R. Salakhutdinov, R. Urtasun, A. Torralba, and S. Fidler. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 19–27, 2015.

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