Supplementary Material: Top-down Neural Attention by Excitation Backprop

Jianming Zhang[1,2] Sarah Adel Bargal[1] Zhe Lin[2] Jonathan Brandt[2]
Xiaohui Shen[2] Stan Sclaroff[1]

[1]Boston University [2]Adobe Research

Contents:

1 Probabilistic Winner-Take-All and Absorbing Markov Chain

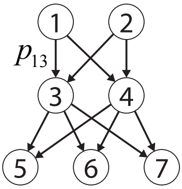


Fig. 1. An example Absorbing Markov Chain process in the feedforward network. The number in the circle denotes the index of the neuron. $p_{ij} := P(a_j|a_i)$ is the transition probability from i to j.

The top-down probabilistic Winner-Take-All process in a neural network can be interpreted as an Absorbing Markov Chain process [1].

A Markov Chain is an absorbing chain if 1) there is at least one absorbing state and 2) it is possible to go from any state to at least one absorbing state in a finite number of steps. Any walk will eventually end at one of the absorbing states. Non-absorbing states are called Transient States. For an absorbing Markov Chain, the canonical form of the transition matrix P can be represented by

$$P = \begin{bmatrix} Q & R \\ \mathbf{0} & I_r \end{bmatrix}, \tag{1}$$

where the entry p_{ij} is the the transition probability from state i to j. Each row sums up to one and I_r is an $r \times r$ identity matrix corresponding to the r absorbing states.

In our case, each random walk starts from an output neuron and ends at some absorbing node in the network. The neurons at the bottom layer are all absorbing nodes as they have no outgoing edges (in top-down order we invert the edges' direction in the network). An example is shown in Fig. 1. We can write down the transition matrix for this example as follows:

$$P = \begin{bmatrix} 0 & 0 & p_{13} & p_{14} & 0 & 0 & 0 \\ 0 & 0 & p_{23} & p_{24} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & p_{35} & p_{36} & p_{37} \\ 0 & 0 & 0 & 0 & p_{45} & p_{46} & p_{47} \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}. \tag{2}$$

For a feedforward network, the corresponding transition matrix can be represented by an upper triangular matrix, as shown above. The fundamental matrix can then be computed by

$$N = \sum_{k=0}^{\infty} Q^k = (I_t - Q)^{-1}, \tag{3}$$

where I_t is the $t \times t$ identity matrix, and

$$Q = \begin{bmatrix} 0 & 0 & p_{13} & p_{14} \\ 0 & 0 & p_{23} & p_{24} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}. \tag{4}$$

In our example, N is simply

$$N = \begin{bmatrix} 1 & 0 & p_{13} & p_{14} \\ 0 & 1 & p_{23} & p_{24} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{5}$$

The (i, j) entry of N can be interpreted as the the expected number of visits to node j, given that the walker starts at i. Let p_{01} and p_{02} denote the prior distribution over the starting node, then the expected numbers of visits to the transient nodes are

$$\begin{aligned} V &= [p_{01}, p_{02}, 0, 0]N \\ &= [p_{01}, p_{02}, p_{01}p_{13} + p_{02}p_{23}, p_{01}p_{14} + p_{02}p_{24}] \end{aligned} \tag{6}$$

for neuron 1, 2, 3 and 4 respectively. This is consistent with the definition the Marginal Winning Probability (MWP) in our formulation. The expected number of visits for absorbing nodes can also easily computed by $V \cdot R$.

In theory, all the hidden neuron's MWP can be computed based on the fundamental matrix, and the MWP is a linear function of the top-down signal vector. In practice, our Excitation Backprop does the computation in a layer-wise fashion, without the need to explicitly construct the fundamental matrix. This layer-wise propagation is possible due to the acyclic nature of the feedforward network.

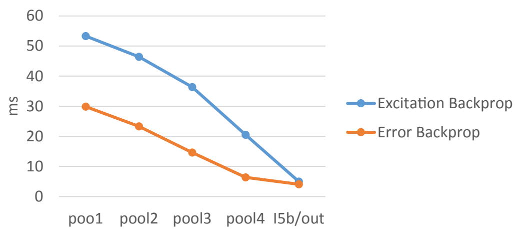


Fig. 2. Speed performance of our implementation of Excitation Backprop compared with error backpropagation in GPU mode. The speed is measured on a NVIDIA K40c GPU for a single 224X335 image (without using batch mode). The x-axis represents the layer at which the tested method terminates.

2 Speed Performance

The most time-consuming operations in Excitation Backprop are the second and fourth steps in Alg.1 of the main manuscript, which correspond to the `forward` and `backward` operations of the layer in Caffe. Therefore, the computational complexity of Excitation Backprop is about twice the complexity of error backpropagation, but in practice we only perform the Excitation Backprop to some intermediate layer. The speed performance of our implementation of Excitation Backprop is reported in Fig. 2 for GoogleNet.

3 Pointing Game

3.1 Classifier Training Details

To train classifiers on COCO [2] and VOC07 [3], we follow the basic fine-tuning procedure for image classification. We fine-tune the output layer of the model using the multi-label cross-entropy objective function on the training split of COCO and VOC07. Images are padded to square shape by mirror padding and up-sampled to 256×256. Random flipping and cropping are used for data augmentation. No multi-scale training [4] is used. We fix the learning rate to be 0.01 for all the architectures and optimize the parameters using SGD. The training batch size is set as 64, 32 and 64 for VGGS, VGG16 and GoogleNet respectively. We stop the training when the training error plateaus.

3.2 Per Category Performance

We report the per category accuracy using the GoogleNet classifier on COCO and VOC07 in Figs. 3 and 4 respectively. Our method c-MWP outperforms competitors in 69/80 categories on COCO and in 9/20 categories on VOC07. Our c-MWP is particularly more accurate than other methods for small objects such as `tie`, `kite`, `baseball bat`, `skateboard`, `bottle` on COCO.

3.3 Qualitative Evaluation

We provide qualitative attention map comparisons in Figs. 5-10 for c-MWP, CAM [5], LRP [6], Deconv [7] and Grad [8].

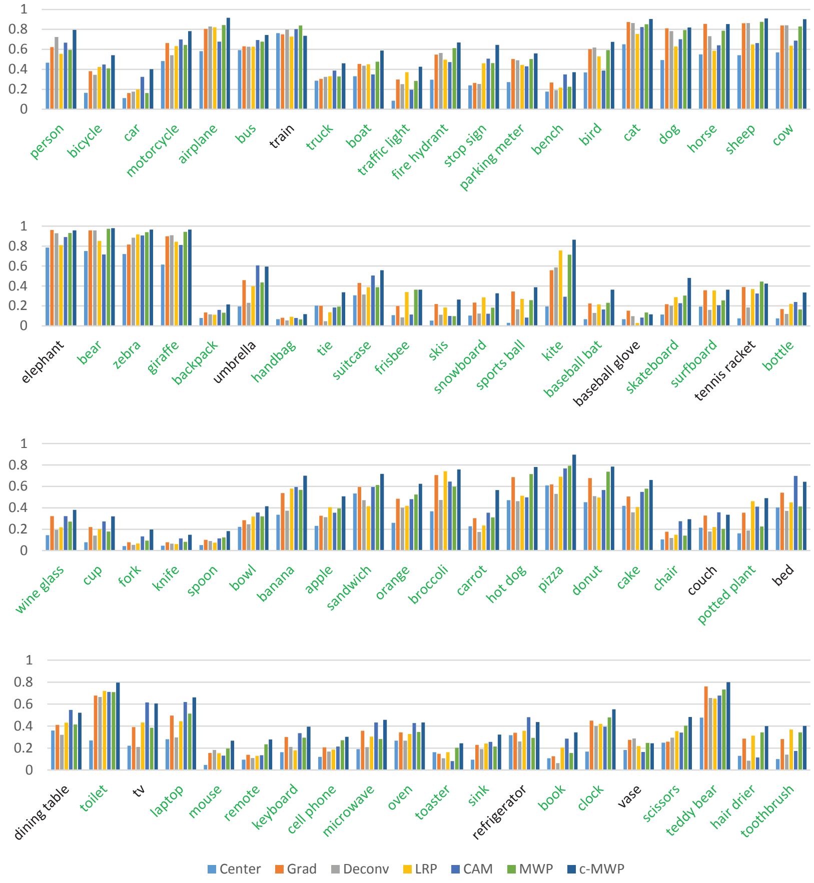


Fig. 3. Pointing Game: mean accuracy per category on COCO using GoogleNet. Categories where c-MWP gives the highest score are marked in green. c-MWP achieves the best performance in 69/80 categories.

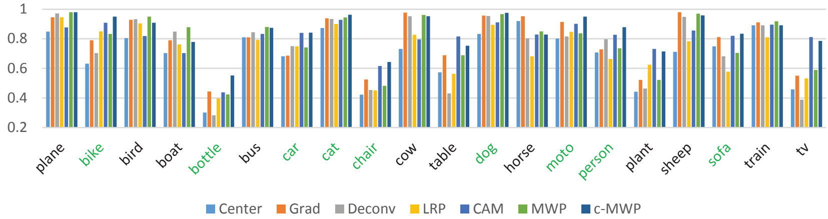


Fig. 4. Pointing Game: mean accuracy per category on VOC07 using GoogleNet. Categories where c-MWP gives the highest score are marked in green. c-MWP achieves the best performance in 9/20 categories.

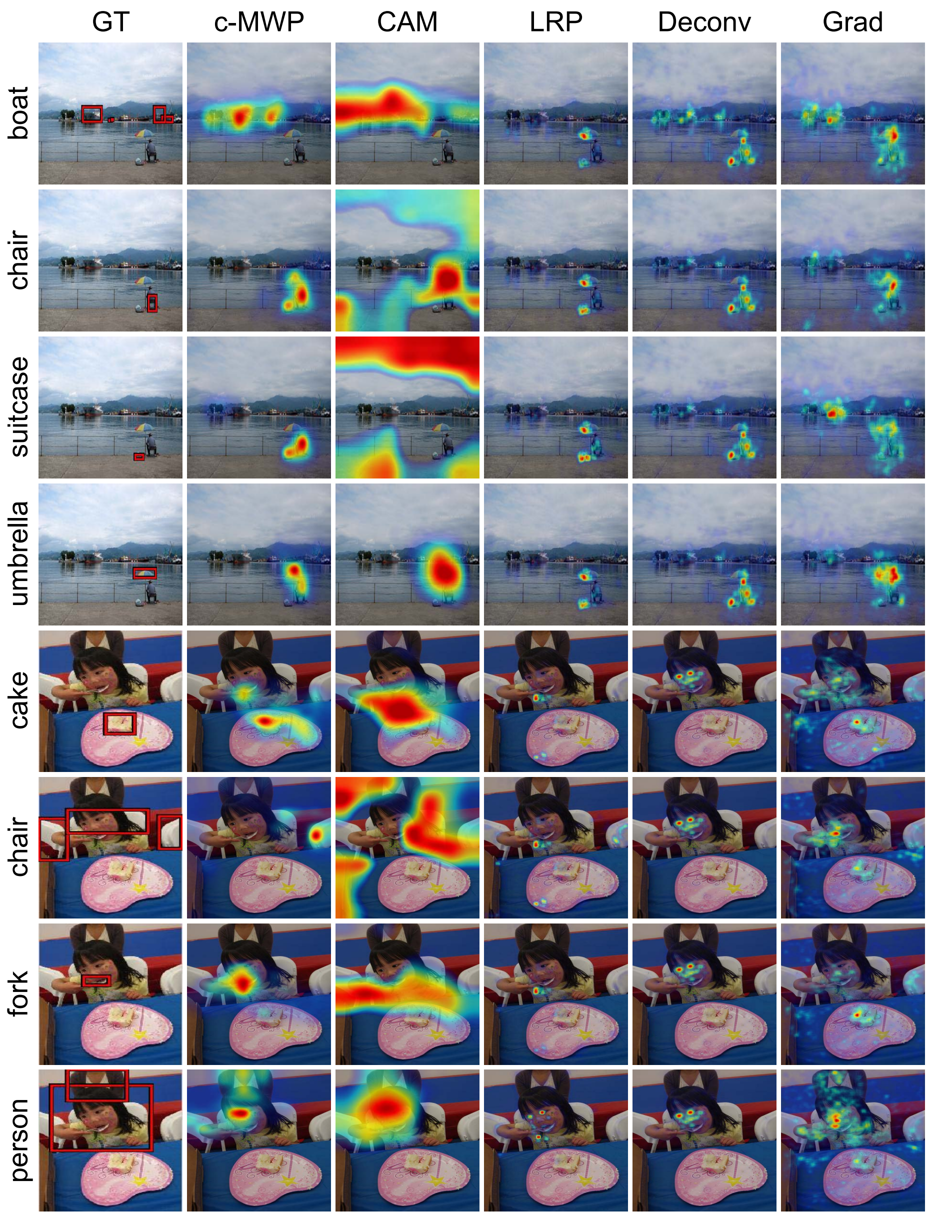


Fig. 5. Pointing Game: example attention maps using GoogleNet on COCO for c-MWP, CAM [5], LRP [6], Deconv [7] and Grad [8].

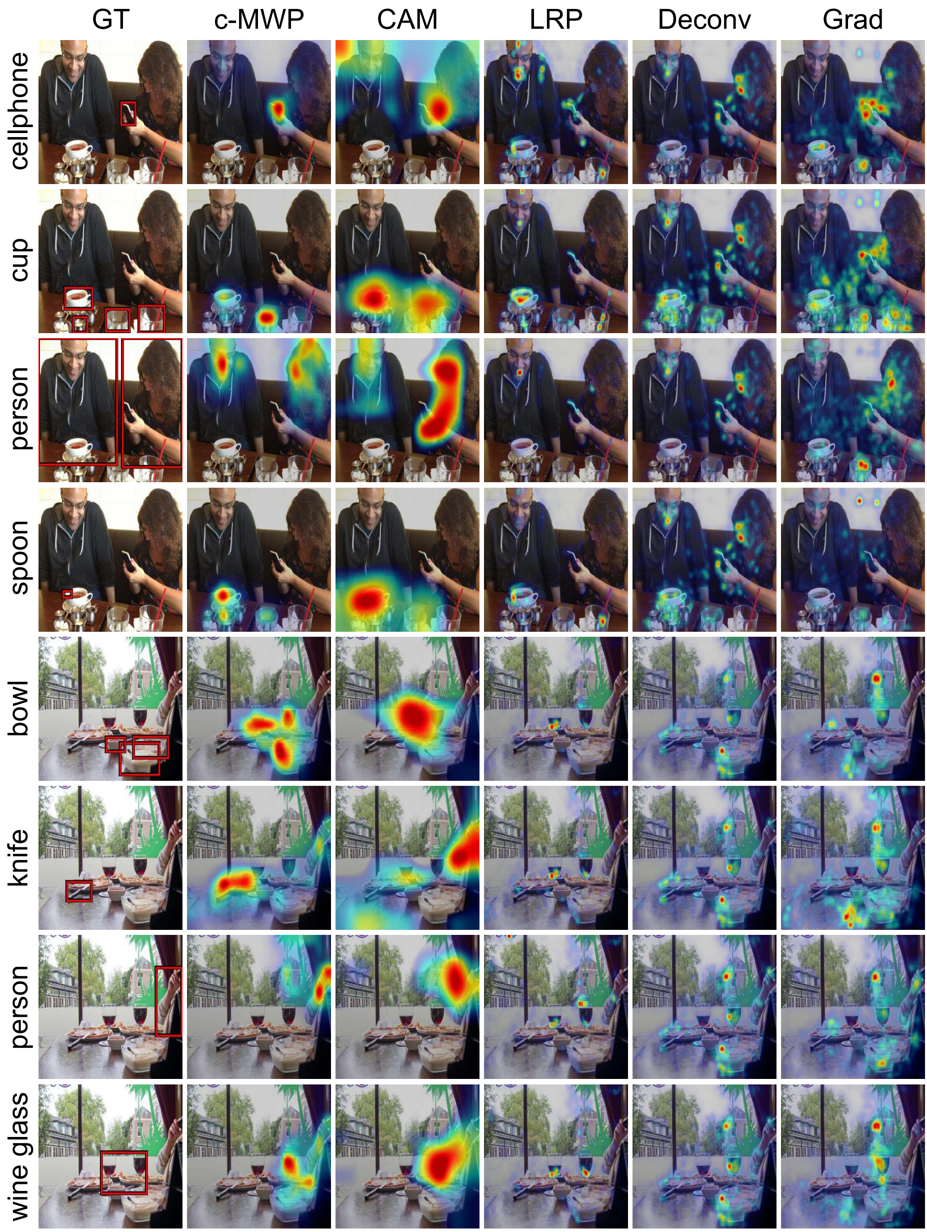


Fig. 6. Pointing Game: example attention maps using GoogleNet on COCO for c-MWP, CAM [5], LRP [6], Deconv [7] and Grad [8].

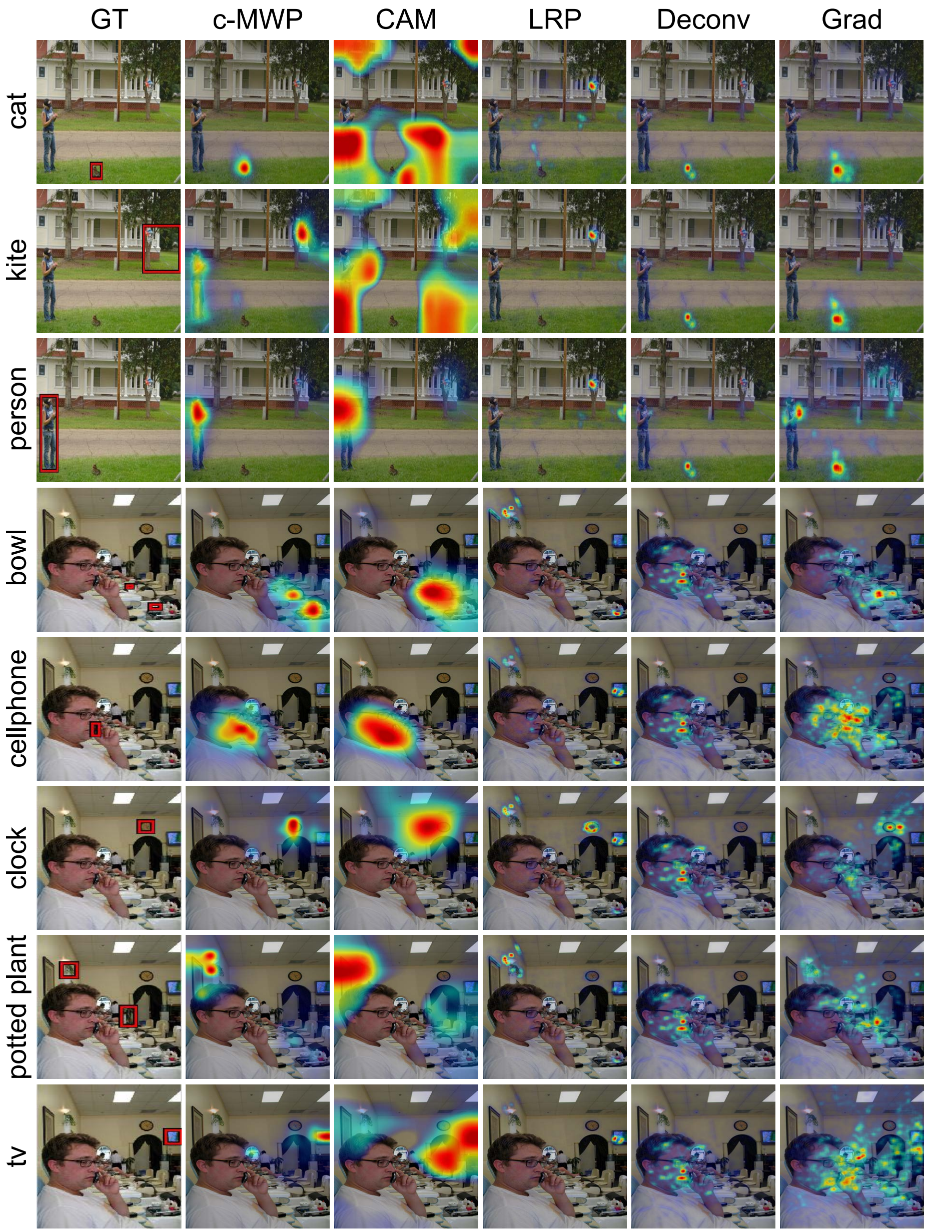


Fig. 7. Pointing Game: example attention maps using GoogleNet on COCO for c-MWP, CAM [5], LRP [6], Deconv [7] and Grad [8].

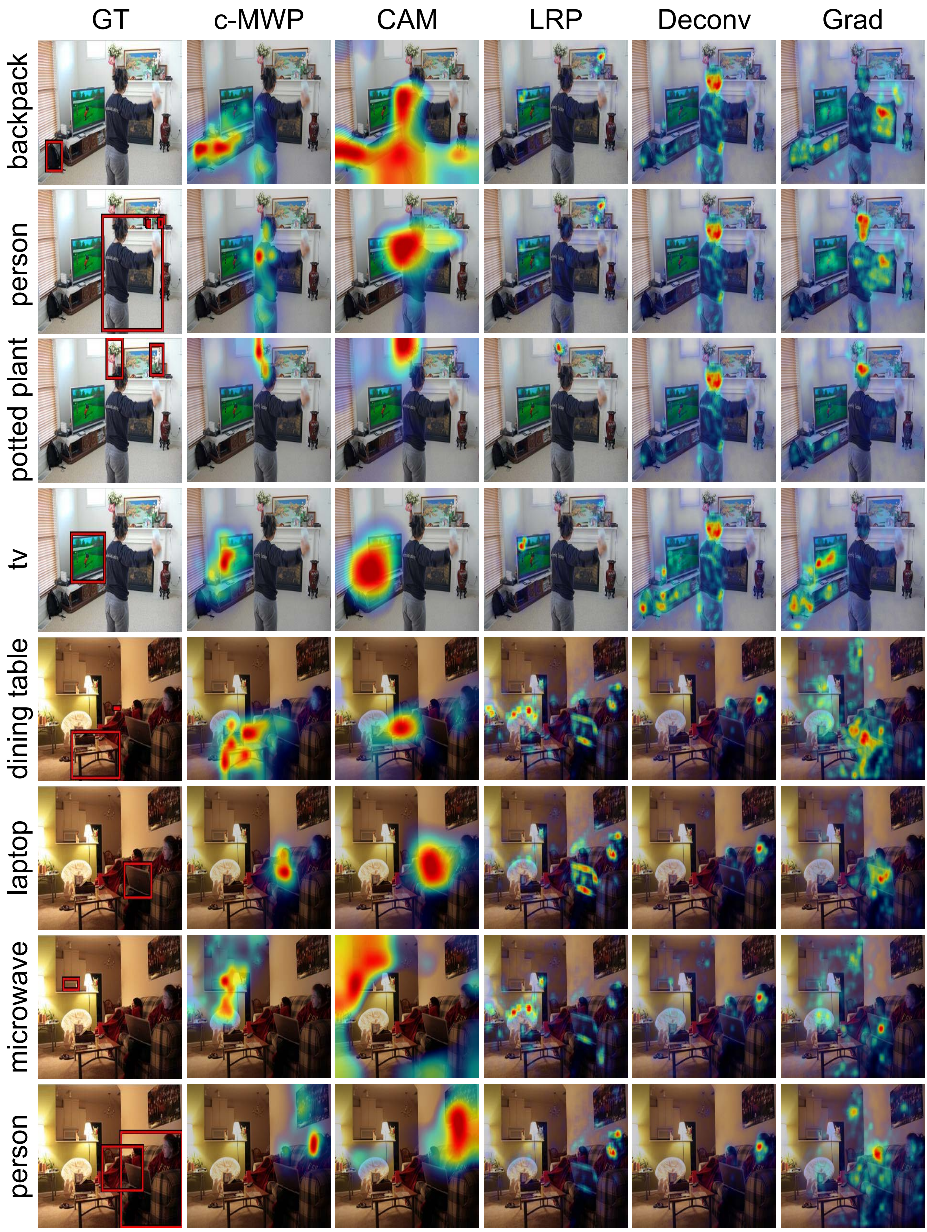


Fig. 8. Pointing Game: example attention maps using GoogleNet on COCO for c-MWP, CAM [5], LRP [6], Deconv [7] and Grad [8].

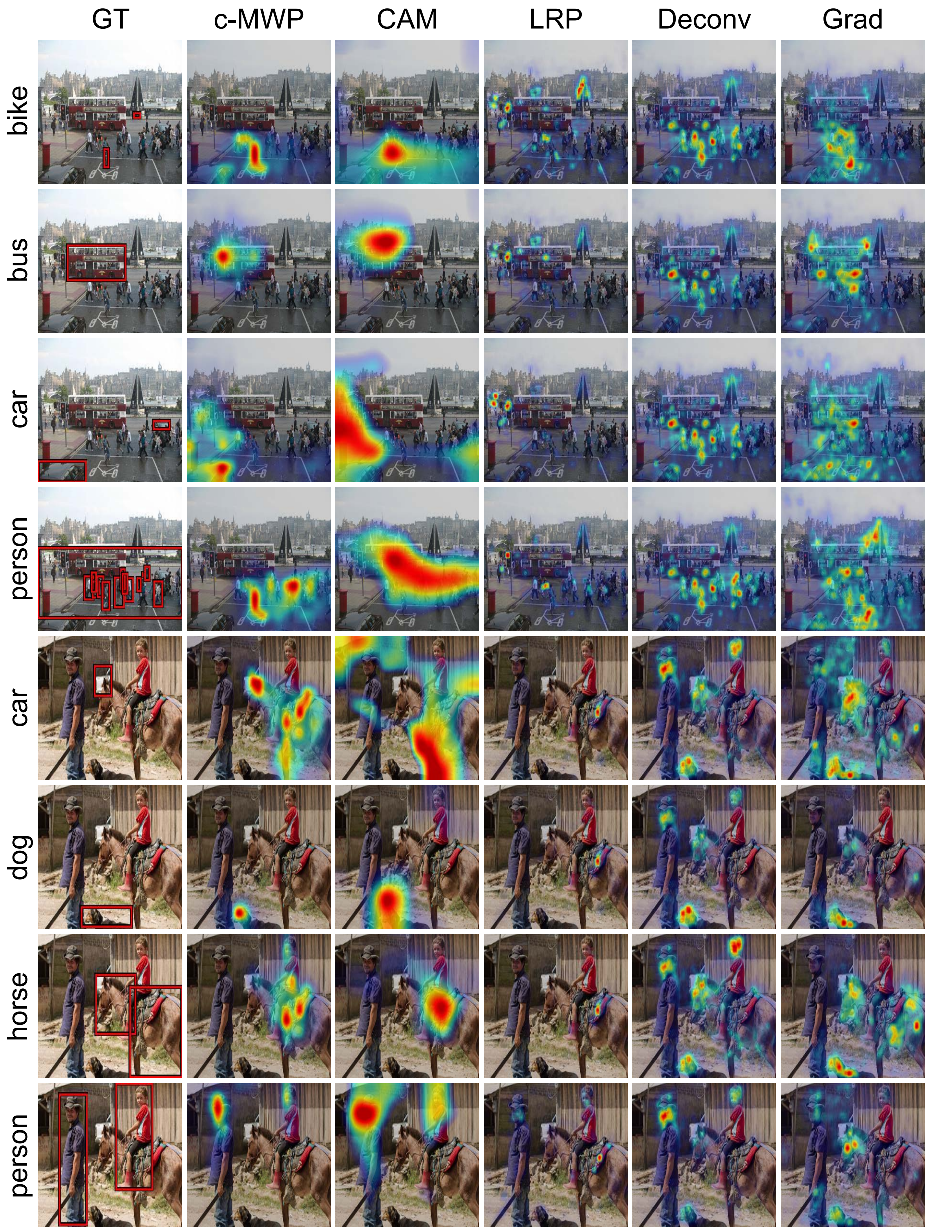


Fig. 9. Pointing Game: example attention maps using GoogleNet on COCO for c-MWP, CAM [5], LRP [6], Deconv [7] and Grad [8].

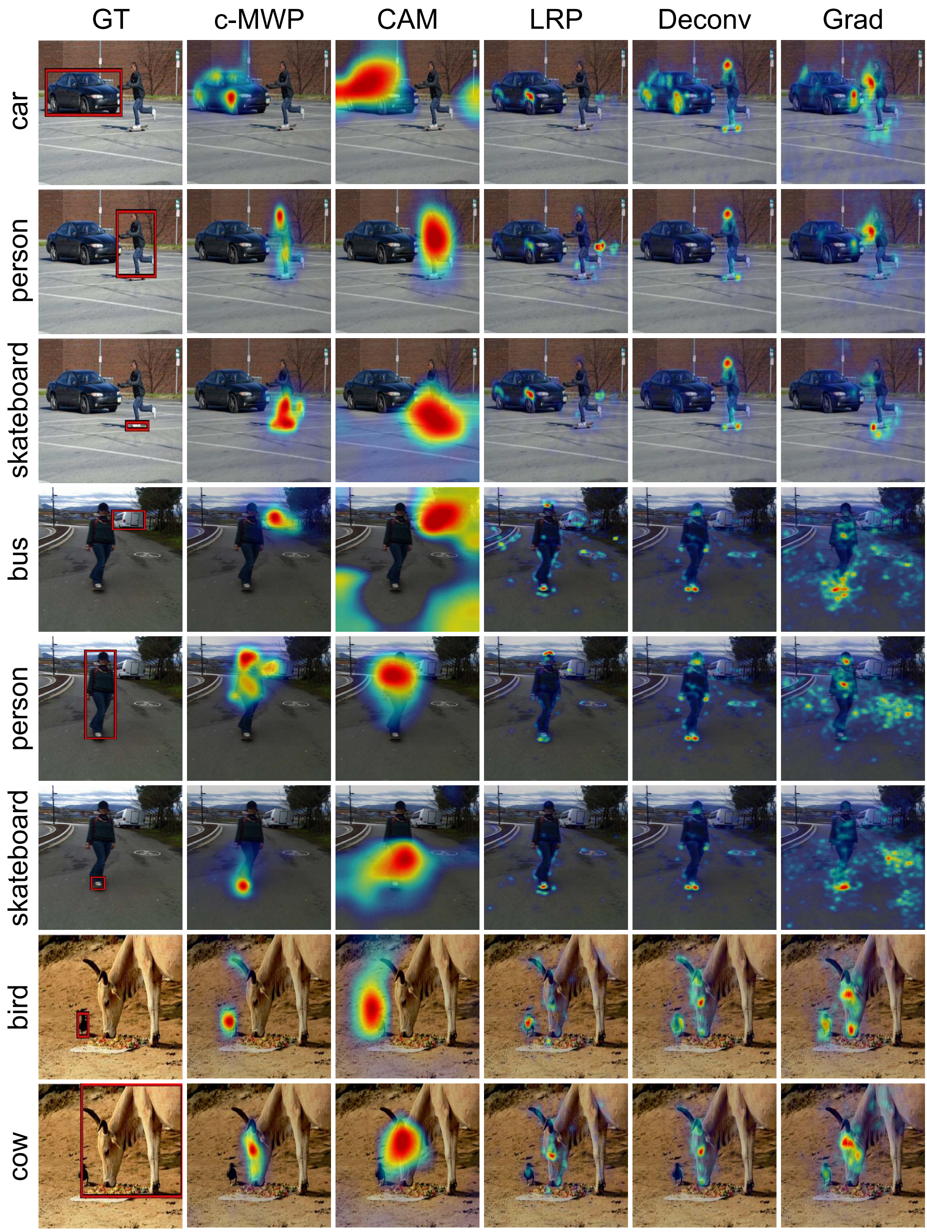


Fig. 10. Pointing Game: example attention maps using GoogleNet on COCO for c-MWP, CAM [5], LRP [6], Deconv [7] and Grad [8].

4 Text-to-Region Association

4.1 Details about the Stock6M Dataset

We provide more details about the Stock6M dataset used for training the image tag classifier.

Data collection and cleaning. We crawl an initial set of about 17M thumbnail images and their tags from a stock image website. This website provides professional photos and illustrations for commercial usage. Each image on the website has a list of tags used for text-based image search. Then we use the most frequent 18157 tags for our dictionary using a frequency threshold of 1000. Most of these tags are unigrams. We remove images with fewer than five tags. We empirically find that some images' tags are in alphabetical order, and the quality of these tags is usually poor. Thus, we remove these images, too. We further perform a duplicate detection based on tag information and user ids, since many images uploaded by the same user can be very similar. For each user id, we check the tag list of each of its images. We remove an image if its first five tags are very similar to the first five tags of a previously seen image of the same user id. The two sets of tags are considered to be similar if they have more than three overlaps. After all these steps, we end up with a dataset of about 6M images.

Frequent Tags and Example Images. We visualize the most frequent tags in a word cloud (Fig. 11). We can see that many frequent tags are related to humans, for example `woman`, `man`, `beautiful`, `happy`, *etc.* There are also a lot of non-visual tags like `healthy`, `business`, `holiday` and `lifestyle`. Some example images and the corresponding user tags are shown in Fig 12.

4.2 Example Word Attention Maps

Example word attention maps are shown in Figs. 13-17.

Fig. 11. Stock6M: visualization of the most frequent tags in Stock6M.

adult, alluring, american, banknote, business, businessman, buying, cash, catch, caucasian, credit, currency, dollar, economy, excited, finance, financial, happy, holding, hundred, loan, male, man, money, ...

smiling, friends, men, women, travel, vacation, book, guide, holidays, traveling, tourism, tourists, travelers, trip, journey, sightseeing, tour, bus, together, summer, friendship, sunglasses, eyewear, shades, happiness, young, people, person, team, hanging, out, having, fun, lifestyle, ...

book, bunch, correspondence, erudition, group, heap, information, journal, journalist, knowledge, library, literature, magazine, media, old, page, paper, periodical, pile, press, publish, reading, stack, subscription, ...

Fig. 12. Stock6M: example images and tags from the stock image website. Note that the first and third images' tags are in alphabetical order. In contrast, the second image's tags are not in alphabetical order and their ordering roughly reflects the relevance to the image. Therefore, we remove images whose tags are in alphabetical order.

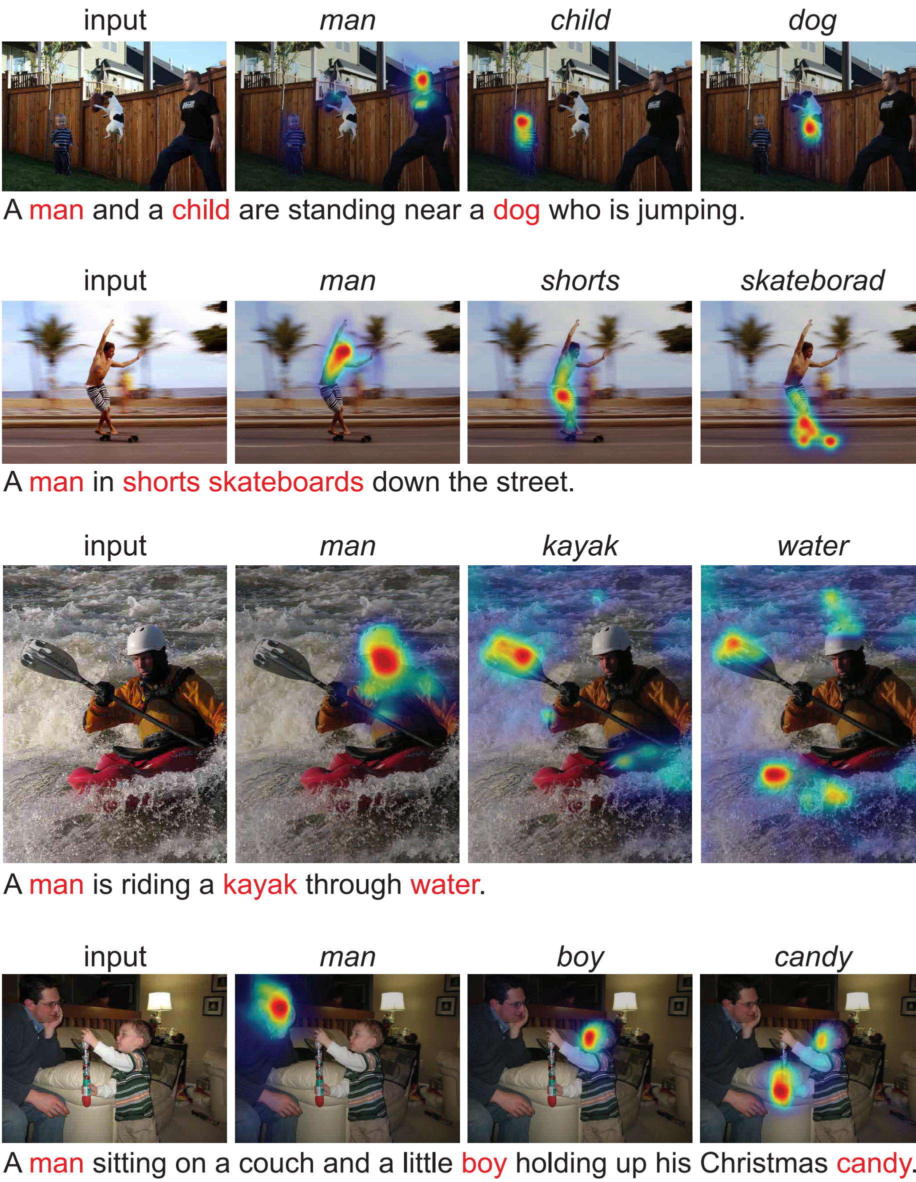
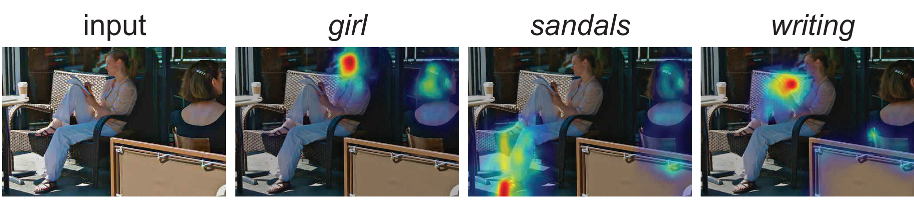


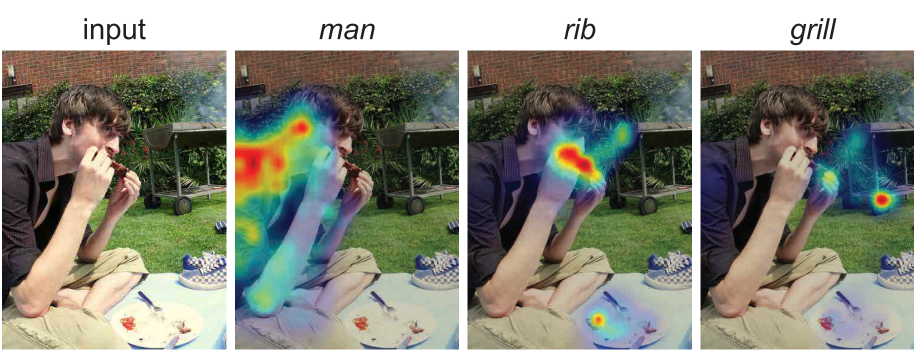
Fig. 13. Text-to-Region Association: word attention maps obtained by c-MWP using our image tag classifier. For each test image, one of its caption annotations from Flickr30k Entities is displayed below. We display the attention maps for the words in red in each caption.

A woman in a red business suit sits on a step sharing food with a man in a leather jacket.

A couple is sitting at a restaurant in front of a big fish sign.



A girl wearing sandals is writing in a notebook while sitting in a chair outside.



A man is eating barbecue ribs outside next to a grill.

Fig. 14. Text-to-Region Association: word attention maps obtained by c-MWP using our image tag classifier. For each test image, one of its caption annotations from Flickr30k Entities is displayed below. We display the attention maps for the words in red in each caption.

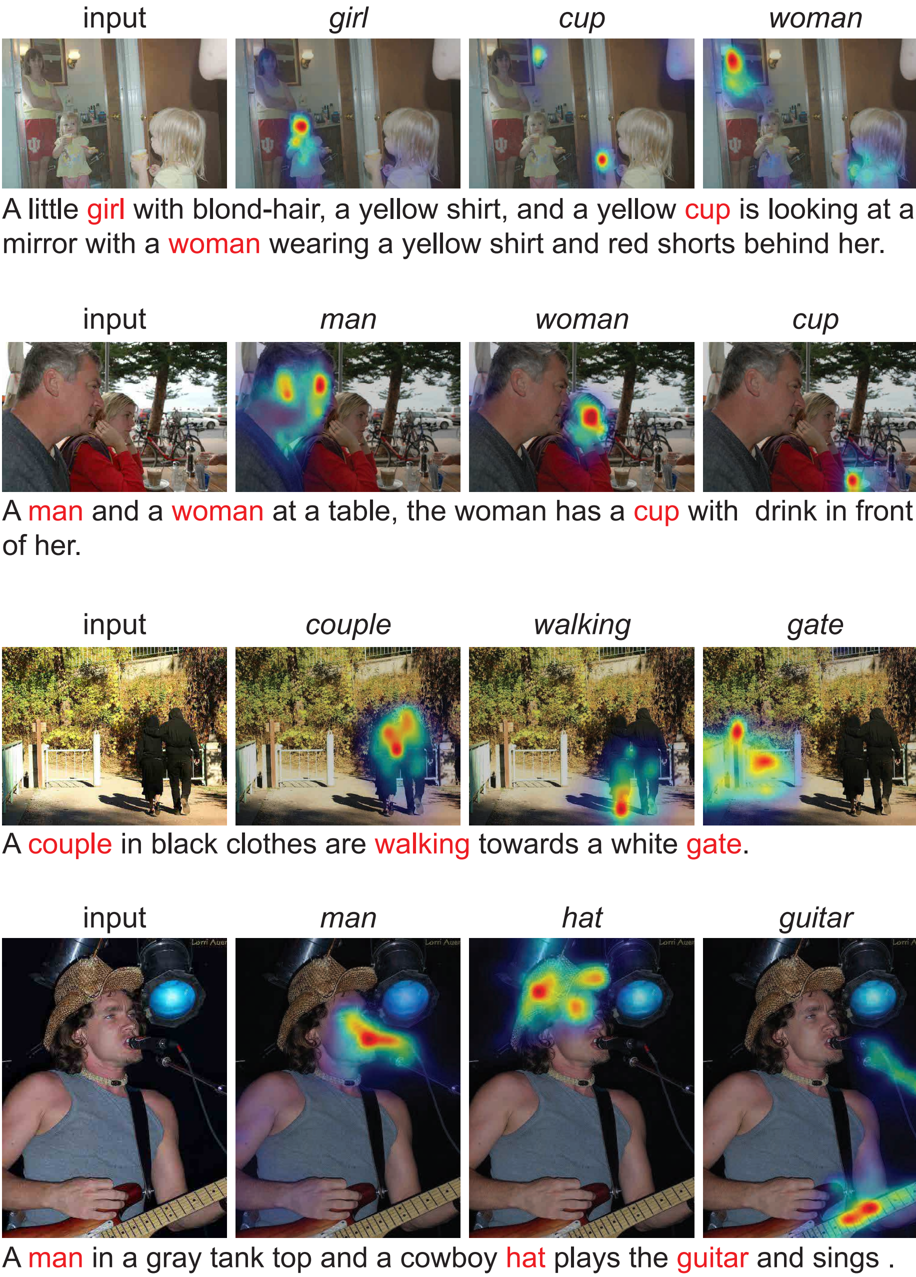


Fig. 15. Text-to-Region Association: word attention maps obtained by c-MWP using our image tag classifier. For each test image, one of its caption annotations from Flickr30k Entities is displayed below. We display the attention maps for the words in red in each caption.

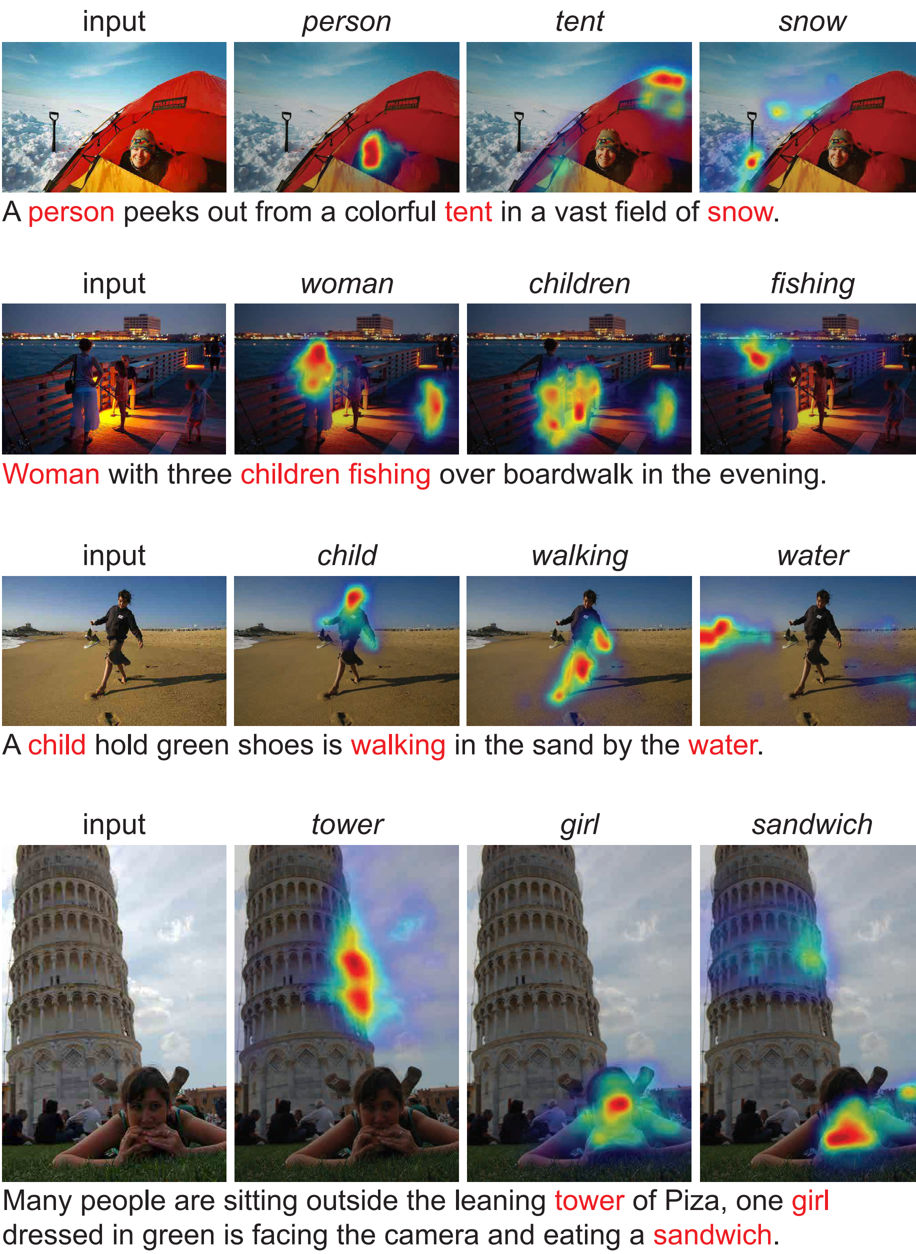


Fig. 16. Text-to-Region Association: word attention maps obtained by c-MWP using our image tag classifier. For each test image, one of its caption annotations from Flickr30k Entities is displayed below. We display the attention maps for the words in red in each caption.

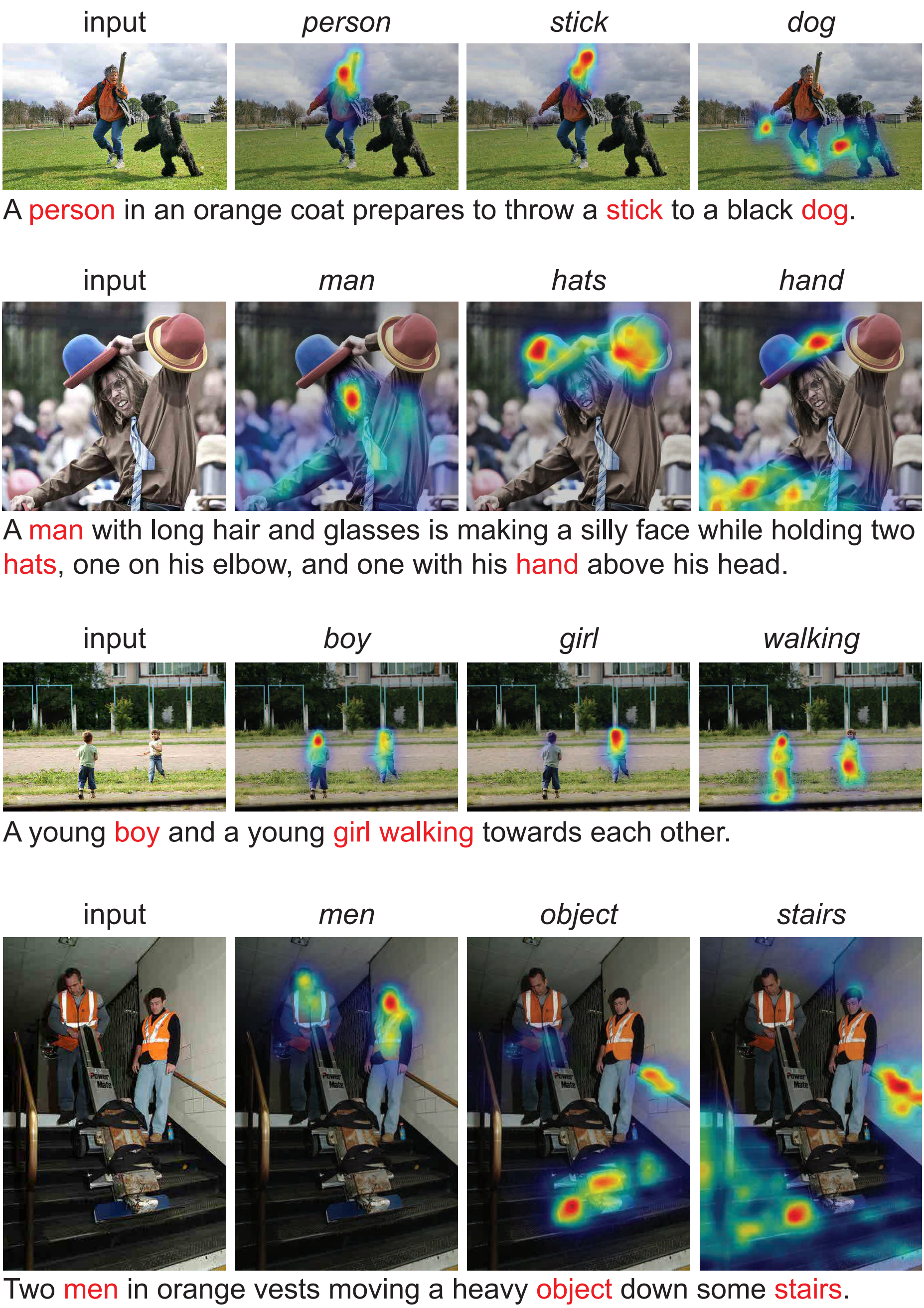


Fig. 17. Text-to-Region Association: word attention maps obtained by c-MWP using our image tag classifier. For each test image, one of its caption annotations from Flickr30k Entities is displayed below. We display the attention maps for the words in red in each caption.

5 Other Discussions

5.1 Effects of the Layer Selection

As we show in the paper, the effect of the layer selection on our method is quite marginal in the pointing game when the spatial resolution of the selected layer is about 14×14 or above. We observe the same trend in the phrase localization experiment. In the phrase localization experiment, the attention maps are used to rank the object proposals. The ranking function is based on the sum of the pixel values inside a proposal, and thus is not sensitive to the spatial resolution of the selected layer.

Table 1. Pool2 *vs.* pool3 using `GoogleNet` in localizing dominant objects on ImageNet.

	pool2	pool3
Loc. Error (%)	38.7	41.2

However, we find that using lower level layers is more critical in the experiment of Sec. 4.2, where the object localization is based on thresholded attention maps. Attention maps of low resolution cannot clearly define the object boundaries, and thus result in less accuracy of the resultant bounding boxes. In Table 1, ~~we compare the performance of pool2 and pool3 of the GoogleNet model. The spatial resolution of the attention map is 28 × 28 for pool2 and 14 × 14 for pool3.~~

References

1. Kemeny, J.G., Snell, J.L., et al.: Finite Markov chains. New York Berlin Heidelberg Tokyo: Springer-Verlag (1960)
2. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft COCO: Common objects in context. In: ECCV. (2014)
3. Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The pascal visual object classes (VOC) challenge. International Journal of Computer Vision **88**(2) (June 2010) 303–338
4. Oquab, M., Bottou, L., Laptev, I., Sivic, J.: Is object localization for free?-weakly-supervised learning with convolutional neural networks. In: CVPR. (2015)
5. Zhou, B., Lapedriza, A., Xiao, J., Torralba, A., Oliva, A.: Learning deep features for scene recognition using places database. In: NIPS. (2014)
6. Bach, S., Binder, A., Montavon, G., Klauschen, F., Müller, K.R., Samek, W.: On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. PloS One **10**(7) (2015) e0130140
7. Zeiler, M.D., Fergus, R.: Visualizing and understanding convolutional networks. In: ECCV. (2014)
8. Simonyan, K., Vedaldi, A., Zisserman, A.: Deep inside convolutional networks: Visualising image classification models and saliency maps. In: ICLR Workshop. (2014)