Grad-CAM:
Visual Explanations from Deep Networks via Gradient-based Localization
Supplementary Material

Ramprasaath R. Selvaraju[1*] Michael Cogswell[1] Abhishek Das[1] Ramakrishna Vedantam[1*]
Devi Parikh[1,2] Dhruv Batra[1,2]
[1]Georgia Institute of Technology [2]Facebook AI Research
{ramprs, cogswell, abhshkdz, vrama, parikh, dbatra}@gatech.edu

In this supplementary document, we provide

- Section 1: Derivation to show that Grad-CAM is a generalization to CAM for *any* CNN-based architecture and hence doesn't require any architectural change or retraining.
- Section 2: Qualitative results showing Grad-CAM and Guided Grad-CAM visualizations for image classification, image captioning, and visual question answering (VQA). For image captioning and VQA, our visualizations (Grad-CAM, and Guided Grad-CAM) expose the somewhat surprising insight that even non-attention based CNN + LSTM models can often be good at localizing discriminative input image regions despite not being trained on grounded image-text pairs.
- Section 3: We make a slight modification to Grad-CAM that can provide *Counterfactual explanations*- which highlight the support for the regions that would make the network change its decision.
- Section 4: We provide Grad-CAM explanations for the two models described in Section 6.3 (Identifying dataset bias).
- Section 5: Ablation studies to explore and validate our design choices for computing Grad-CAM visualizations.
- Section 6: Weakly-supervised segmentation results on PASCAL VOC 2012 by using weak-localization cues from Grad-CAM as a seed for SEC [24].
- Section 8: Comparison to existing visualization techniques, CAM and c-MWP on PASCAL and COCO, where we find that our visualizations are superior, while being faster to compute and at the same time being possible to visualize a *wide variety of CNN-based models, including but not limited to, CNNs with fully-connected layers, CNNs stacked with Recurrent Neural Networks (RNNs), ResNets* etc..
- Section 9: Analysis of Grad-CAM visualizations for 200-layer Residual Network.

1. Grad-CAM as generalization of CAM

In this section we formally prove that Grad-CAM is a generalization of CAM, as mentioned in Section 3 in the main paper.
Recall that the CAM architecture consists of fully-convolutional CNNs, followed by global average pooling, and linear classification layer with softmax.
Let the final convolutional layer produce K feature maps A^k, with each element indexed by i, j. So A^k_{ij} refers to the activation at location (i, j) of the feature map A^k.
CAM computes a global average pooling (GAP) on A^k_{ij}. Let us define F^k to be the global average pooled output,
So,

$$F^k = \frac{1}{Z} \sum_i \sum_j A^k_{ij} \tag{1}$$

*Work done at Virginia Tech.

CAM computes the final scores by,

$$Y^c = \sum_k w_k^c \cdot F^k \tag{2}$$

where w_k^c is the weight connecting the k^{th} feature map with the c^{th} class.
Taking the gradient of the score for class c (Y^c) with respect to the feature map F^k we get,

$$\text{(From Chain Rule)} \frac{\partial Y^c}{\partial F^k} = \frac{\frac{\partial Y^c}{\partial A_{ij}^k}}{\frac{\partial F^k}{\partial A_{ij}^k}} \tag{3}$$

Taking partial derivative of (1) w.r.t. A_{ij}^k, we can see that $\frac{\partial F^k}{\partial A_{ij}^k} = \frac{1}{Z}$. Substituting this in (3), we get,

$$\frac{\partial Y^c}{\partial F^k} = \frac{\partial Y^c}{\partial A_{ij}^k} \cdot Z \tag{4}$$

From (2) we get that, $\frac{\partial Y^c}{\partial F^k} = w_k^c$. Hence,

$$w_k^c = Z \cdot \frac{\partial Y^c}{\partial A_{ij}^k} \tag{5}$$

Now, we can sum both sides of this expression in (5) over all pixels (i, j) to get:

$$\sum_i \sum_j w_k^c = \sum_i \sum_j Z \cdot \frac{\partial Y^c}{\partial A_{ij}^k}, \quad \text{which can be rewritten as} \tag{6}$$

$$Z w_k^c = Z \sum_i \sum_j \frac{\partial Y^c}{\partial A_{ij}^k} \quad \text{(Since } Z \text{ and } w_k^c \text{ do not depend on } (i, j)) \tag{7}$$

Note that Z is the number of pixels in the feature map (or $Z = \sum_i \sum_j 1$). Thus, we can re-order terms and see that:

$$w_k^c = \sum_i \sum_j \frac{\partial Y^c}{\partial A_{ij}^k} \tag{8}$$

We can see that up to a proportionality constant ($1/Z$) that is normalized out during visualization, the expression for w_k^c is identical to α_k^c used by Grad-CAM (as described in the main paper).
Thus Grad-CAM is a generalization of CAM to arbitrary CNN-based architectures, while maintaining the computational efficiency of CAM.

2. Experimental Results

In this section we provide more qualitative results for Grad-CAM and Guided Grad-CAM applied to the task of image classification, image captioning and VQA.

2.1. Image Classification

We use Grad-CAM and Guided Grad-CAM to visualize the regions of the image that provide support for a particular prediction. The results reported in Fig. A1 correspond to the VGG-16 [41] network trained on ImageNet.
Fig. A1 shows randomly sampled examples from COCO [27] validation set. COCO images typically have multiple objects per image and Grad-CAM visualizations show precise localization to support the model's prediction.
Guided Grad-CAM can even localize tiny objects. For example our approach correctly localizes the predicted class "torch" (Fig. A1.a) inspite of its size and odd location in the image. Our method is also class-discriminative – it places attention *only* on the "toilet seat" even when a popular ImageNet category "dog" exists in the image (Fig. A1.e).

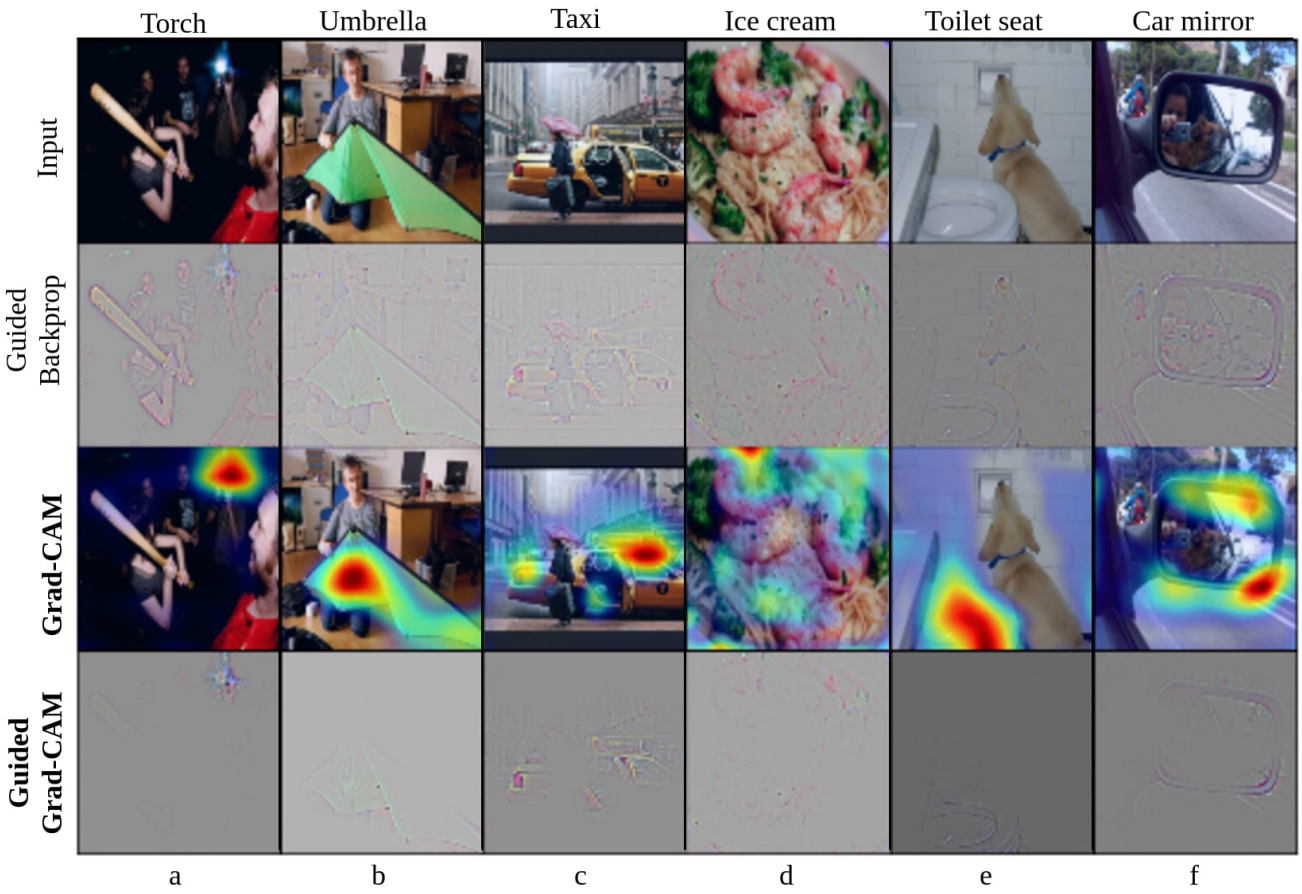


Figure A1: Visualizations for randomly sampled images from the COCO validation dataset. Predicted classes are mentioned at the top of each column.

2.2. Image Captioning

We use the publicly available Neuraltalk2 code and model[1] for our image captioning experiments. The model uses VGG-16 to encode the image. The image representation is passed as input at the first time step to an LSTM that generates a caption for the image. The model is trained end-to-end along with CNN finetuning using the COCO [27] Captioning dataset. We feedforward the image to the image captioning model to obtain a caption. We use Grad-CAM to get a coarse localization and combine it with Guided Backpropagation to get a high-resolution visualization that highlights regions in the image that provide support for the generated caption.

2.3. Visual Question Answering (VQA)

We use Grad-CAM and Guided Grad-CAM to explain why a publicly available VQA model [30] answered what it answered. The VQA model by Lu *et al*. uses a standard CNN followed by a fully connected layer to transform the image to 1024-dim to match the LSTM embeddings of the question. Then the transformed image and LSTM embeddings are pointwise multiplied to get a combined representation of the image and question and a multi-layer perceptron is trained on top to predict one among 1000 answers. We show visualizations for the VQA model trained with 3 different CNNs - AlexNet [25], VGG-16 and VGG-19 [41]. Even though the CNNs were not finetuned for the task of VQA, it is interesting to see how our approach can serve as a tool to understand these networks better by providing a localized high-resolution visualization of the regions the model is looking at. Note that these networks were trained with no explicit attention mechanism enforced.

Notice in the first row of Fig. A3, for the question, "*Is the person riding the waves?*", the VQA model with AlexNet and VGG-16 answered "No", as they concentrated on the person mainly, and not the waves. On the other hand, VGG-19 correctly answered "Yes", and it looked at the regions around the man in order to answer the question. In the second row, for the question, "*What is the person hitting?*", the VQA model trained with AlexNet answered "Tennis ball" just based on context

[1] https://github.com/karpathy/neuraltalk2

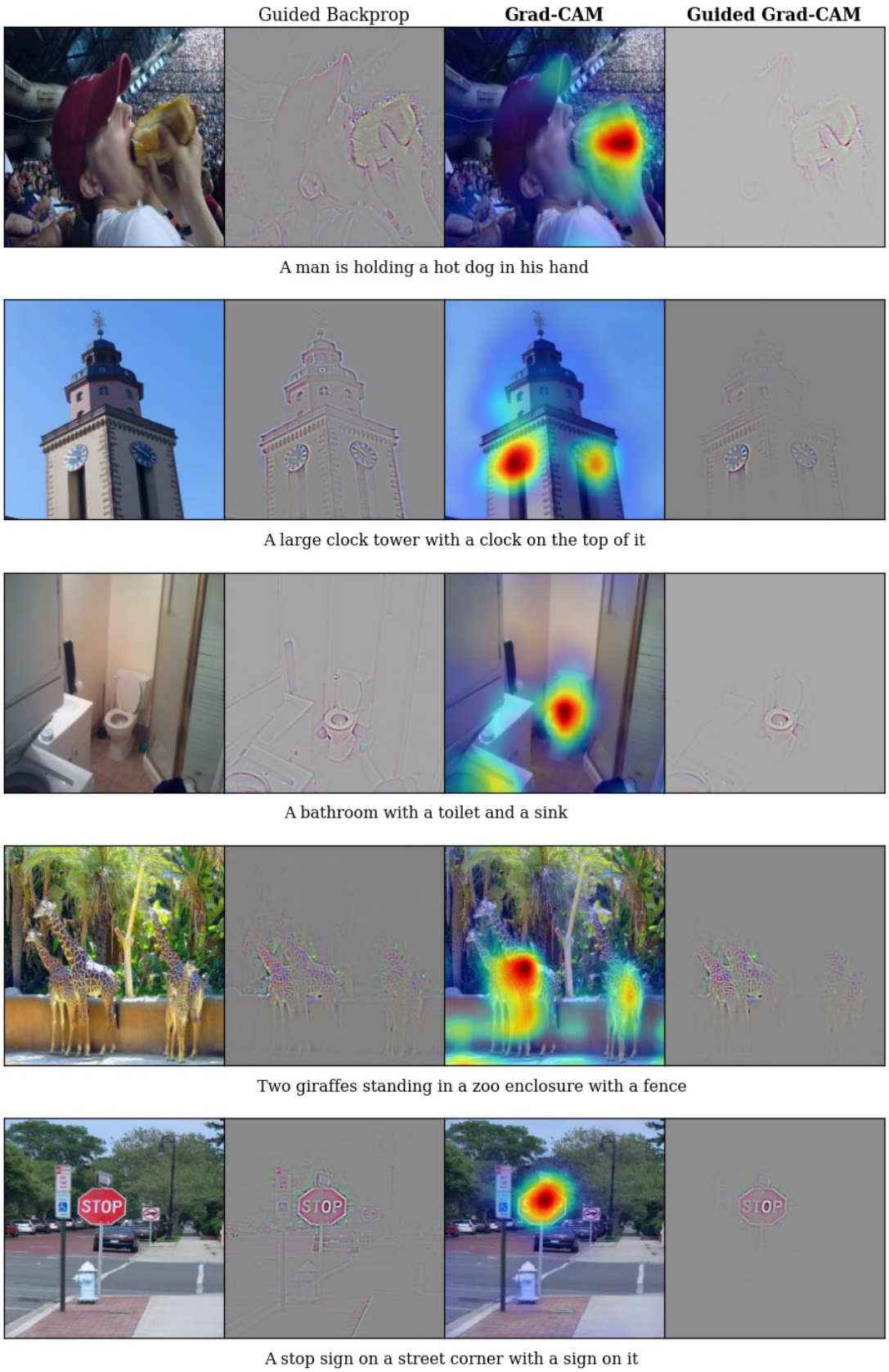


Figure A2: Guided Backpropagation, Grad-CAM and Guided Grad-CAM visualizations for the captions produced by the Neuraltalk2 image captioning model.

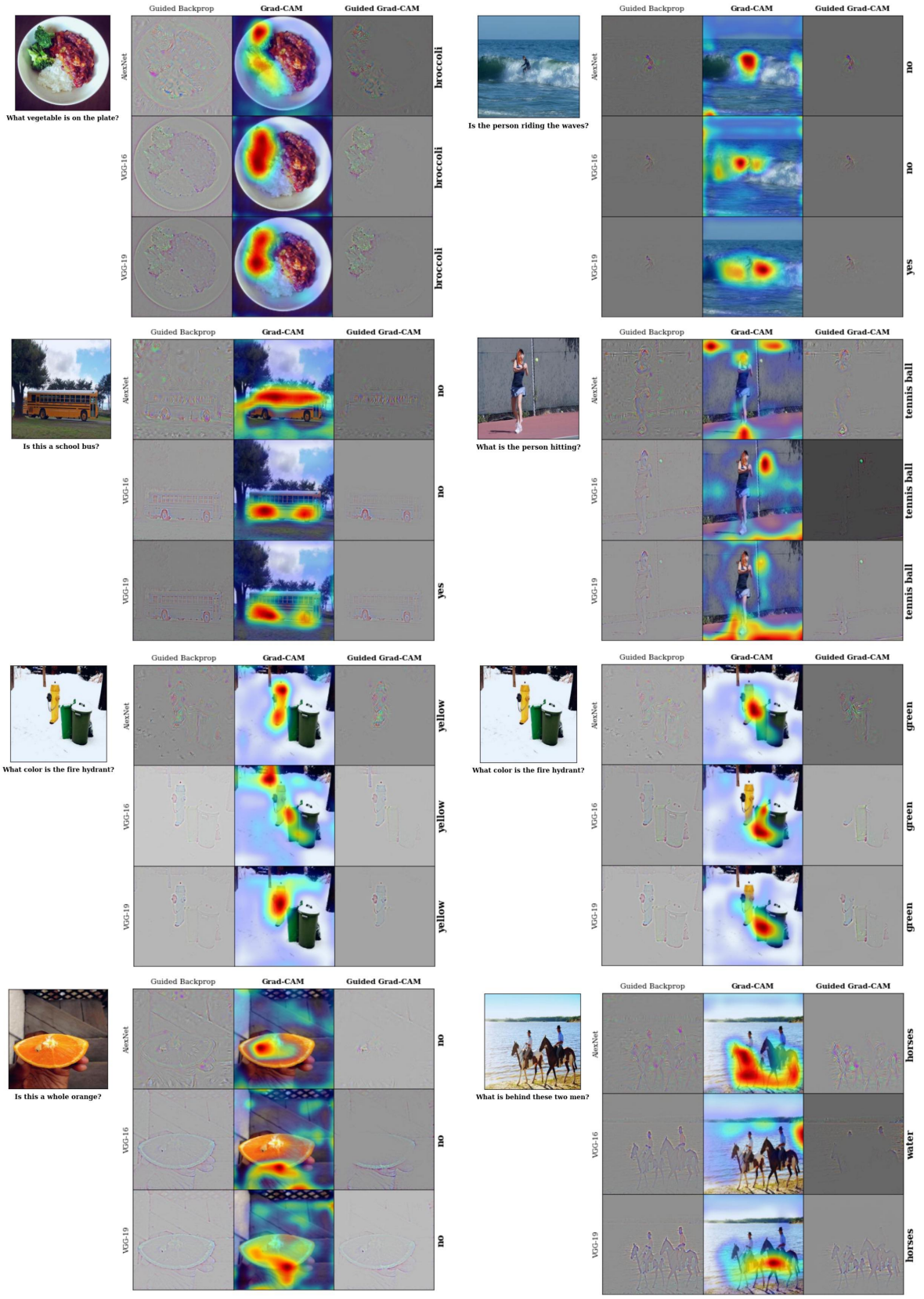


Figure A3: Guided Backpropagation, Grad-CAM and Guided Grad-CAM visualizations for the answers from a VQA model. For each image-question pair, we show visualizations for AlexNet, VGG-16 and VGG-19. Notice how the attention changes in row 3, as we change the answer from *Yellow* to *Green*.

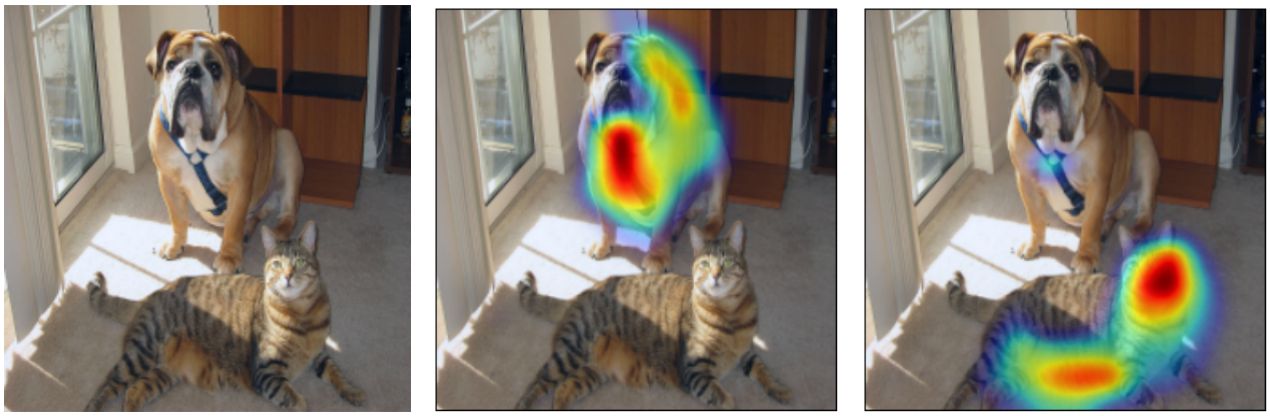
without looking at the ball. Such a model might be risky when employed in real-life scenarios. It is difficult to determine the trustworthiness of a model just based on the predicted answer. Our visualizations provide an accurate way to explain the model's predictions and help in determining which model to trust, without making any architectural changes or sacrificing accuracy. Notice in the last row of Fig. A3, for the question, "*Is this a whole orange?*", the model looks for regions around the orange to answer "No".

3. Counterfactual Explanations

Using a slight modification to Grad-CAM we obtain counterfactual explanations, which highlight the support for the regions that would make the network change its decision. Removing concepts occurring in those regions would make the model more confident about the given target decision.
Specifically, we negate the gradient of y^c (score for class c) with respect to feature maps A of a convolutional layer. Thus the importance weights α_k^c, now become,

$$\alpha_k^c = \overbrace{\frac{1}{Z}\sum_i\sum_j}^{\text{global average pooling}} \underbrace{-\frac{\partial y^c}{\partial A_{ij}^k}}_{\text{Negative gradients}} \tag{9}$$

As in (2), we weighted sum the forward activation maps, A with weights α_k^c, and follow it by a ReLU to obtain *counterfactual explanations* as shown in Fig. A4.



(a) Original Image (b) Cat Counterfactual exp (c) Dog Counterfactual exp

Figure A4: Counterfactual Explanations with Grad-CAM

4. Identifying and removing bias in datasets

In this section we provide qualitative examples showing the explanations from the two models trained for distinguishing doctors from nurses- model1 which was trained on images (with an inherent bias) from a popular search engine, and model2 which was trained on a more balanced set of images from the same search engine.
As shown in Fig. A5, Grad-CAM visualizations of the model predictions show that the model had learned to look at the person's face / hairstyle to distinguish nurses from doctors, thus learning a gender stereotype.
Using the insights gained from the Grad-CAM visualizations, we balanced the dataset and retrained the model. The new model, model2 not only generalizes well to a balanced test set, it also looks at the right regions.
Statistics for the two models can be found in Table. A1a and Table. A1b

5. Ablation studies

In this section we provide details of the ablation studies we performed.

5.1. Varying mask size for occlusion

Fig. 1 (e,k) of main paper show the results of occlusion sensitivity for the "cat" and "dog" class. We compute this occlusion map by repeatedly masking regions of the image and forward propagate each masked image. At each location of the occlusion

Ground Truth \ Predicted	doctor	nurse	total
doctor	79	34 (22 female)	113
nurse	7 (6 male)	106	113
total	86	140	

(a) Confusion Matrix for model trained with biased examples from search engine. Note that the text in () indicate the number of mistakes made due to gender bias.

Ground Truth \ Predicted	doctor	nurse	total
doctor	101	12 (6 female)	113
nurse	10 (3 male)	103	113
total	111	115	

(b) Confusion Matrix for model trained after correcting dataset bias learned from Grad-CAM visualizations. See that mistakes due to gender bias has reduced significantly.

Table A1: Statistics for biased and unbiased model

(a) Original image (b) Grad-CAM for biased model (c) Grad-CAM for unbiased model

(d) Original Image (e) Grad-CAM for biased model (f) Grad-CAM for unbiased model

(g) Original Image (h) Grad-CAM for biased model (i) Grad-CAM for unbiased model

Figure A5: Grad-CAM explanations for biased and unbiased model. In (a-c) even though both made the right decision, the biased model was looking at the face of the person to make the decision (b), whereas the unbiased model was correctly looking at the short sleeves (c). For (d) and (g) the biased model made the wrong prediction (misclassifying doctor as a nurse) by looking at the face/ hairstyle (e, h), unlike the unbiased model which made the right prediction by looking at the white coat and the stethoscope (f, i).

map we store the difference in the original score for the particular class and the score obtained after forward propagating the masked image. Our choices for mask sizes include (10×10, 15×15, 25×25, 35×35, 45×45, and 90×90). We zero-pad the images so that the resultant occlusion map is of the same size as the original image. The resultant occlusion maps can be found in Fig. A6. Note that blue regions correspond to a decrease in score for a particular class ("tiger cat" in the case of Fig. A6) when the region around that pixel is occluded. Hence it serves as an evidence *for* the class. Whereas the red regions correspond to an increase in score as the region around that pixel is occluded. Hence these regions might indicate existence of other confusing classes. We observe that 35×35 is a good trade-off between sharp results and a smooth appearance.

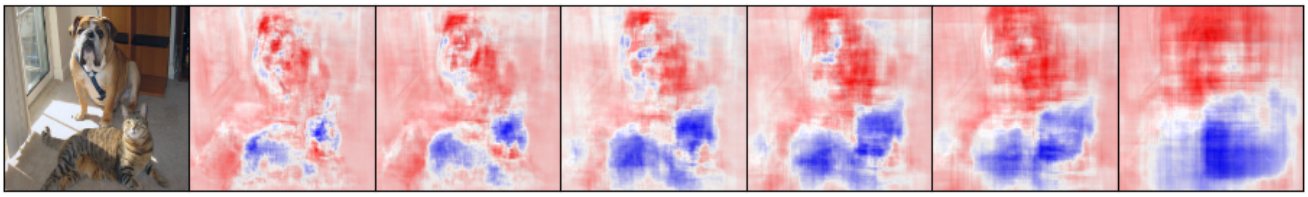


Figure A6: Occlusion maps with different mask sizes for the "tiger cat" category.

5.2. Guided Grad-CAM on different layers

We show results of applying Grad-CAM for the "Tiger-cat" category on different convolutional layers in AlexNet and VGG-16 CNN. As expected, the results from Fig. A7 show that localization becomes progressively worse as we move to shallower convolutional layers. This is because the later convolutional layers capture high-level semantic information and at the same time retain spatial information, while the shallower layers have smaller receptive fields and only concentrate on local features that are important for the next layers.

5.3. Design choices

Method	Top-1 error
Grad-CAM	59.65
Grad-CAM without ReLU in Eq.1	74.98
Grad-CAM with Absolute gradients	58.19
Grad-CAM with GMP gradients	59.96
Grad-CAM with Deconv ReLU	83.95
Grad-CAM with Guided ReLU	59.14

Table A2: Localization results on ILSVRC-15 val for the ablation studies. Note that the localizations reported in this table were created for a single-crop, compared to the 10-crop evaluation reported in the main paper.

We evaluate design choices via top-1 localization error on the ILSVRC15 val set [9].

5.3.1 Importance of ReLU in Eq. 2 in main paper

Removing ReLU (Eq. 1 in main paper) increases error by 15.3%. See Table. A2. Negative values in Grad-CAM indicate confusion between multiple occurring classes. Thus, localization improves when we suppress them (see Fig. A9).

5.3.2 Absolute value of each derivative in Eq. 1 in main paper

Taking the absolute value of each derivative in Eq. 1 in main paper decreases the error by 1.5% (see Table. A2). But qualitatively maps look a bit worse (see Fig. A9), and this evaluation does not fully capture class discriminability (most ImageNet images have only 1 class).

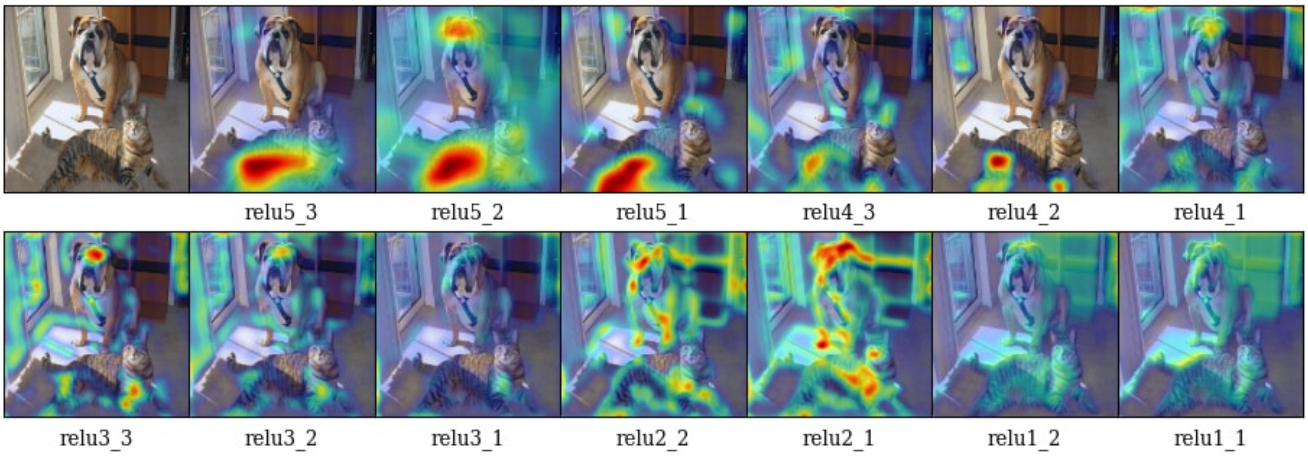


Figure A7: Grad-CAM at different convolutional layers for the 'tiger cat' class. This figure analyzes how localizations change qualitatively as we perform Grad-CAM with respect to different feature maps in a CNN (VGG16 [41]). We find that the best looking visualizations are often obtained after the deepest convolutional layer in the network, and localizations get progressively worse at shallower layers. This is consistent with our intuition described in Section 3 of main paper.

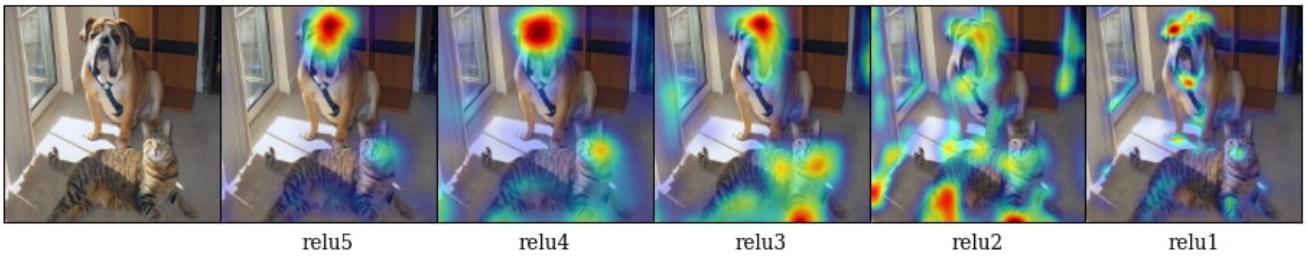


Figure A8: Grad-CAM localizations for "tiger cat" category for different rectified convolutional layer feature maps for AlexNet.

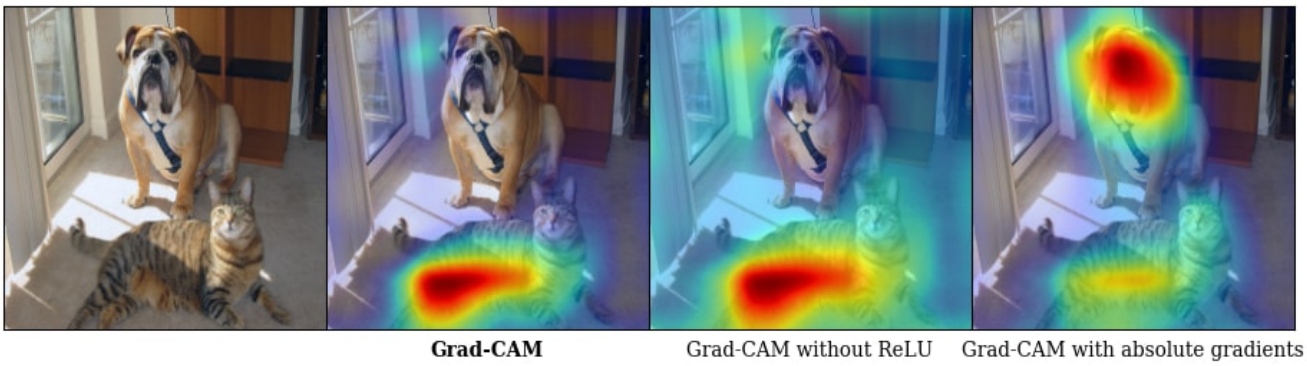


Figure A9: Grad-CAM for "tiger cat" category stating the importance of ReLU and effect of using absolute gradients in Eq. 1 of main paper.

5.3.3 Global Average Pooling *vs*. Global Max Pooling

Instead of Global Average Pooling (GAP) the incoming gradients to the convolutional layer, we tried Global Max Pooling (GMP) them. We observe that using GMP lowers the localization ability of our Grad-CAM technique. An example can be found in Fig. A10 below. This observation is also summarized in Table. A2. This may be due to the fact that *max* is statistically less robust to noise compared to the *averaged* gradient.

5.3.4 Effect of different ReLU backward on Grad-CAM

We experiment with different modifications to the backward pass of ReLU - using Guided-ReLU [42] and Deconv-ReLU [45].

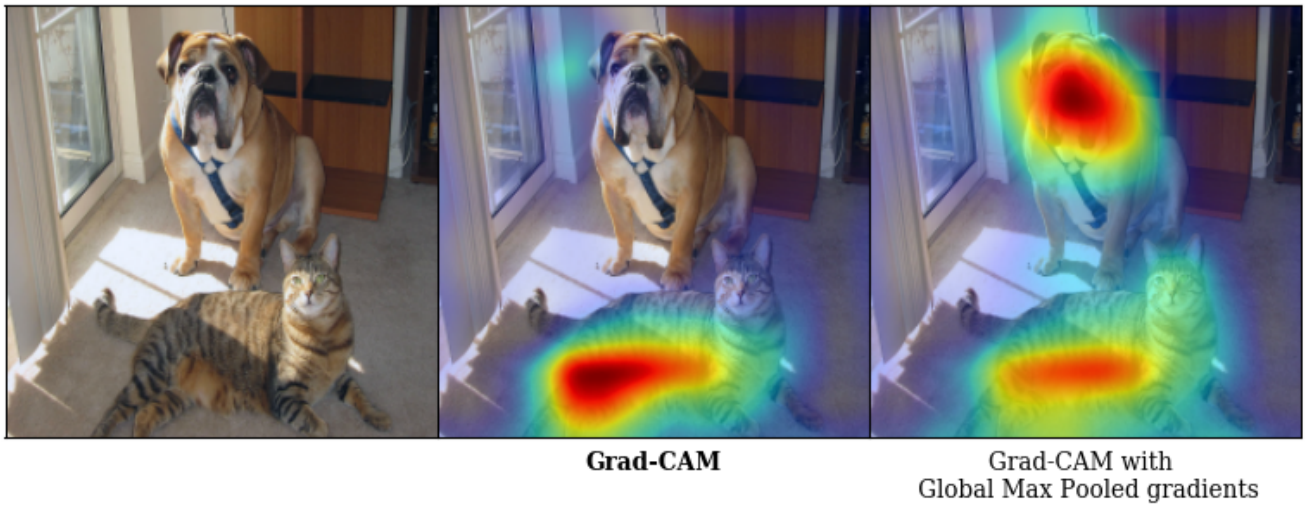


Figure A10: Grad-CAM visualizations for "tiger cat" category with Global Average Pooling and Global Max Pooling.

Effect of Guided-ReLU:
Springenberg *et al*. [42] introduced Guided Backprop, where they modified the backward pass of ReLU to pass only positive gradients to regions with positive activations. Applying this change to the computation of our Grad-CAM maps introduces a drop in the class-discriminative ability of Grad-CAM as can be seen in Fig. A11, but it gives a slight improvement in the localization ability on ILSVRC'14 localization challenge (see Table. A2).

Effect of Deconv-ReLU:
Zeiler and Fergus [45] in their Deconvolution work introduced a slight modification to the backward pass of ReLU, to pass only the positive gradients from higher layers. Applying this modification to the computation of our Grad-CAM gives worse results as shown in Fig. A11.

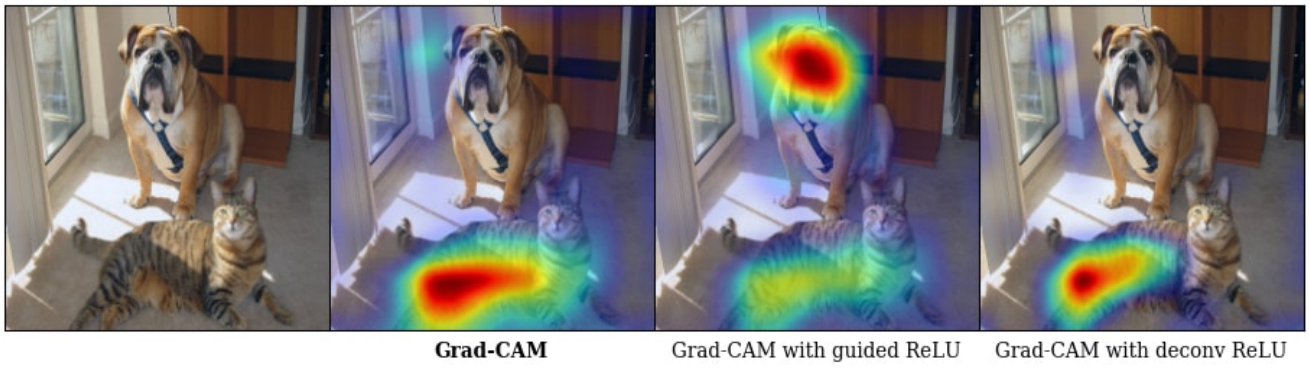


Figure A11: Grad-CAM visualizations for "tiger cat" category for different modifications to the ReLU backward pass. The best results are obtained when we use the actual gradients during the computation of Grad-CAM.

6. Weakly-supervised segmentation

In recent work Kolesnikov *et al*. [24] introduced a new loss function for training weakly-supervised image segmentation models. Their loss function is based on three principles: 1. to seed with weak localization cues, 2. to expand object seeds to regions of reasonable size, 3. to constrain segmentations to object boundaries. They showed that their proposed loss function leads to better segmentation.

They showed that their algorithm is very sensitive to seed loss, without which the segmentation network fails to localize the objects correctly [24]. We used Grad-CAM as seed (weak-supervision for localizing foreground classes), and train their segmentation architecture. This model achieves an accuracy (intersection/union measure) of 49.6 %. Note that to obtain the Grad-CAM localization, we take off-the-shelf classification CNN (VGG-16), trained only with class labels and not bounding box annotations. We show qualitative results in Fig. A12. The last row shows 2 failure cases. In the bottom left image, the

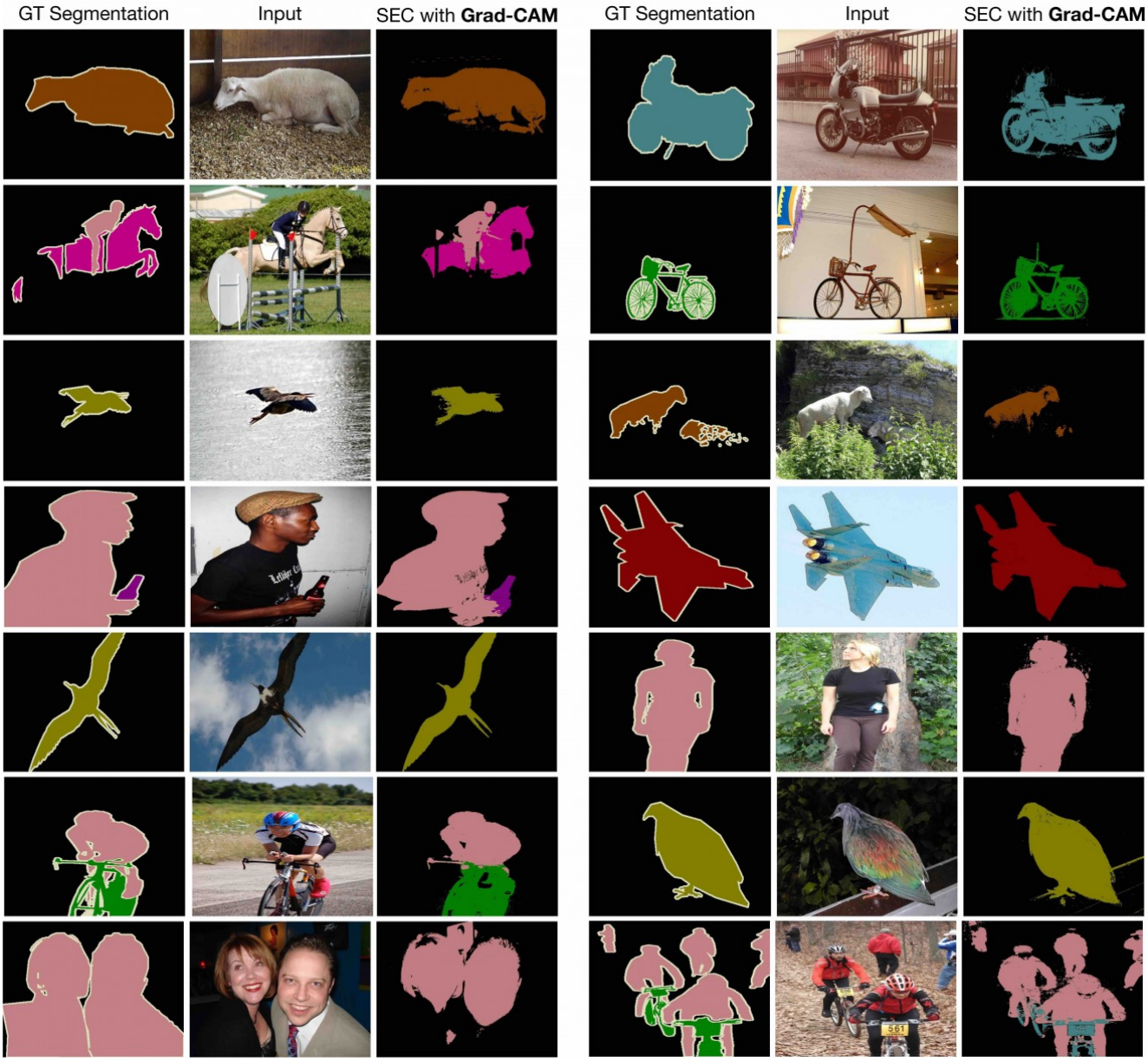


Figure A12: PASCAL VOC 2012 Segmentation results with Grad-CAM as seed for SEC [24].

clothes of the 2 person weren't highlighted correctly. This could be because the most discriminative parts are their faces, and hence Grad-CAM maps only highlights those. This results in a segmentation that only highlights the faces of the 2 people. In the bottom right image, the bicycles, being extremely thin aren't highlighed. This could be because the resolution of the Grad-CAM maps are low (14×14) which makes it difficult to capture thin areas.

7. More details of Pointing Game

In [46], the pointing game was setup to evaluate the discriminativeness of different attention maps for localizing ground-truth categories. In a sense, this evaluates the precision of a visualization, *i.e.* how often does the attention map intersect the segmentation map of the ground-truth category. This does not evaluate how often the visualization technique produces maps which do not correspond to the category of interest. For example this evaluation does not penalize the visualization in Fig. A14 top-left, for highlighting a zebra when visualizing the bird category.

Hence we propose a modification to the pointing game to evaluate visualizations of the top-5 predicted category. In this case the visualizations are given an additional option to reject any of the top-5 predictions from the CNN classifiers. For each of the two visualizations, Grad-CAM and c-MWP, we choose a threshold on the max value of the visualization, that can be used to determine if the category being visualized exists in the image.

We compute the maps for the top-5 categories, and based on the maximum value in the map, we try to classify if the map is of the GT label or a category that is absent in the image. As mentioned in Section 4.2 of the main paper, we find that our approach Grad-CAM outperforms c-MWP by a significant margin (70.58% vs 60.30%). Fig. A14 shows the maps computed for the top-5 categories using c-MWP and Grad-CAM.

8. Qualitative comparison to Excitation Backprop (c-MWP) and CAM

In this section we provide more qualitative results comparing Grad-CAM with CAM [47] and c-MWP [46].

8.1. PASCAL

We compare Grad-CAM, CAM and c-MWP visualizations from ImageNet trained VGG-16 models finetuned on PASCAL VOC 2012 dataset. While Grad-CAM and c-MWP visualizations can be directly obtained from existing models, CAM requires an architectural change, and requires re-training, which leads to loss in accuracy. Also, unlike Grad-CAM, c-MWP and CAM can only be applied for image classification networks. Visualizations for the ground-truth categories can be found in Fig. A13.

8.2. COCO

We compare Grad-CAM and c-MWP visualizations from ImageNet trained VGG-16 models finetuned on COCO dataset. Visualizations for the top-5 predicted categories can be found in Fig. A14. It can be seen that c-MWP highlights arbitrary regions for predicted but non-existent categories, unlike Grad-CAM which seem much more reasonable. We quantitatively evaluate this through the pointing experiment.

9. Analyzing Residual Networks

In this section, we perform Grad-CAM on Residual Networks (ResNets). In particular, we analyze the 200-layer architecture trained on ImageNet[2].

Current ResNets [16] typically consist of residual blocks. One set of blocks use identity skip connections (shortcut connections between two layers having identical output dimensions). These sets of residual blocks are interspersed with downsampling modules that alter dimensions of propagating signal. As can be seen in Fig. A15 our visualizations applied on the last convolutional layer can correctly localize the cat and the dog. Grad-CAM can also visualize the cat and dog correctly in the residual blocks of the last set. However, as we go towards earlier sets of residual blocks with different spatial resolution, we see that Grad-CAM fails to localize the category of interest (see last row of Fig. A15). We observe similar trends for other ResNet architectures (18 and 50-layer ResNets) including ones finetuned for other tasks such as image captioning or VQA.

References

[1] A. Agrawal, D. Batra, and D. Parikh. Analyzing the Behavior of Visual Question Answering Models. In *EMNLP*, 2016.

[2] H. Agrawal, C. S. Mathialagan, Y. Goyal, N. Chavali, P. Banik, A. Mohapatra, A. Osman, and D. Batra. CloudCV: Large Scale Distributed Computer Vision as a Cloud Service. In *Mobile Cloud Visual Media Computing*, pages 265–290. Springer, 2015.

[3] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. Lawrence Zitnick, and D. Parikh. VQA: Visual Question Answering. In *ICCV*, 2015.

[4] L. Bazzani, A. Bergamo, D. Anguelov, and L. Torresani. Self-taught object localization with deep networks. In *WACV*, 2016.

[5] Y. Bengio, A. Courville, and P. Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013.

[2]We use the 200-layer ResNet architecture from https://github.com/facebook/fb.resnet.torch.

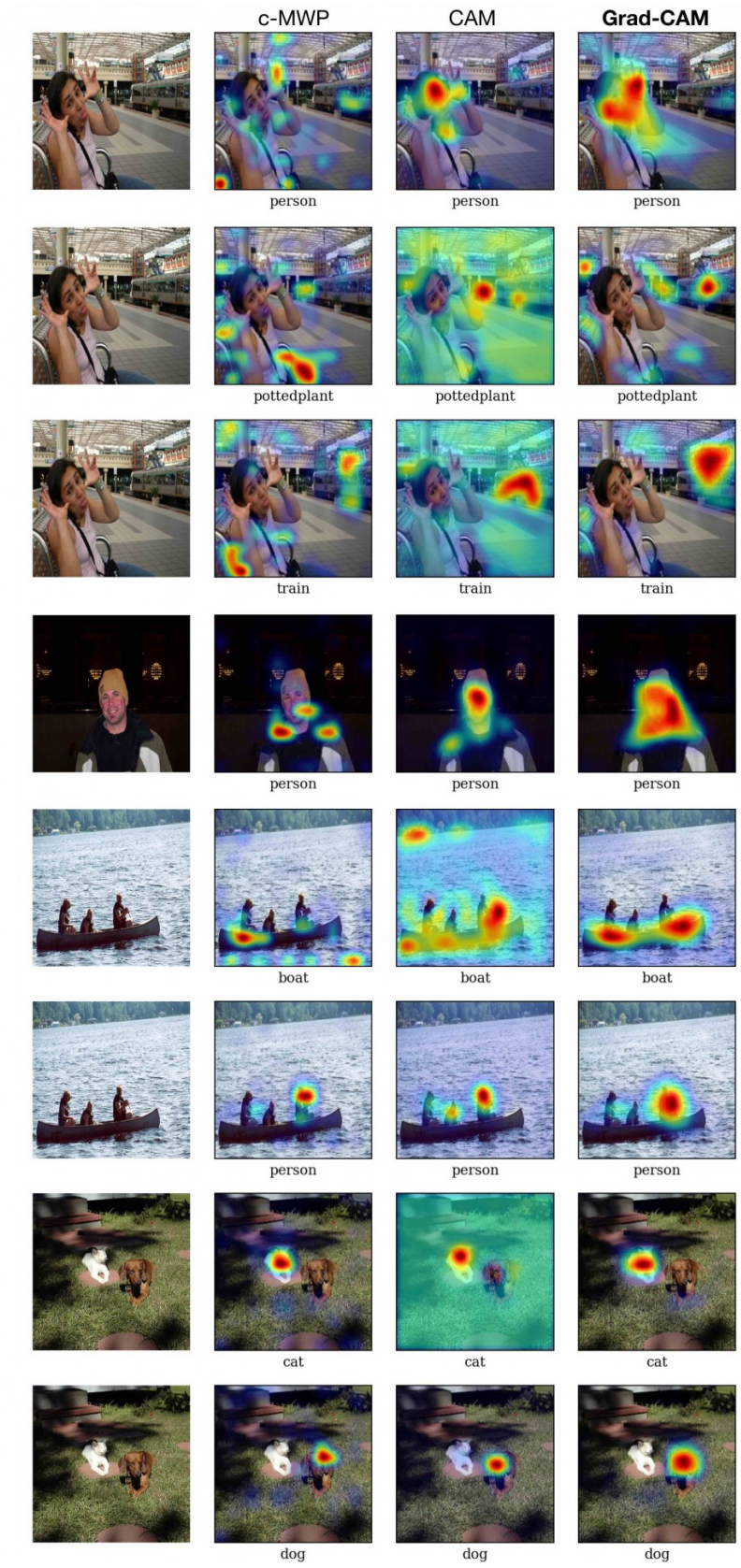


Figure A13: Visualizations for ground-truth categories (shown below each image) for images sampled from the PASCAL validation set.

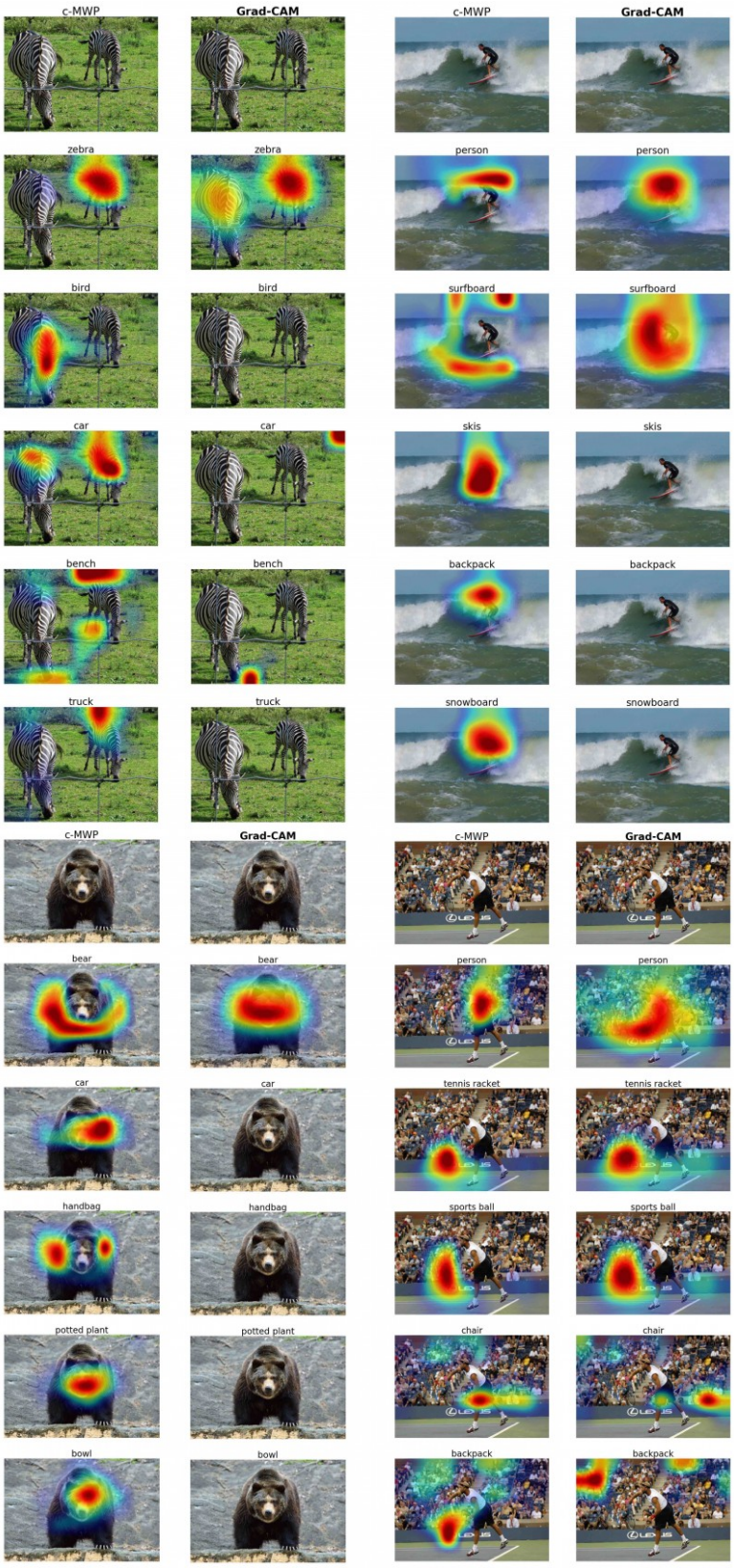
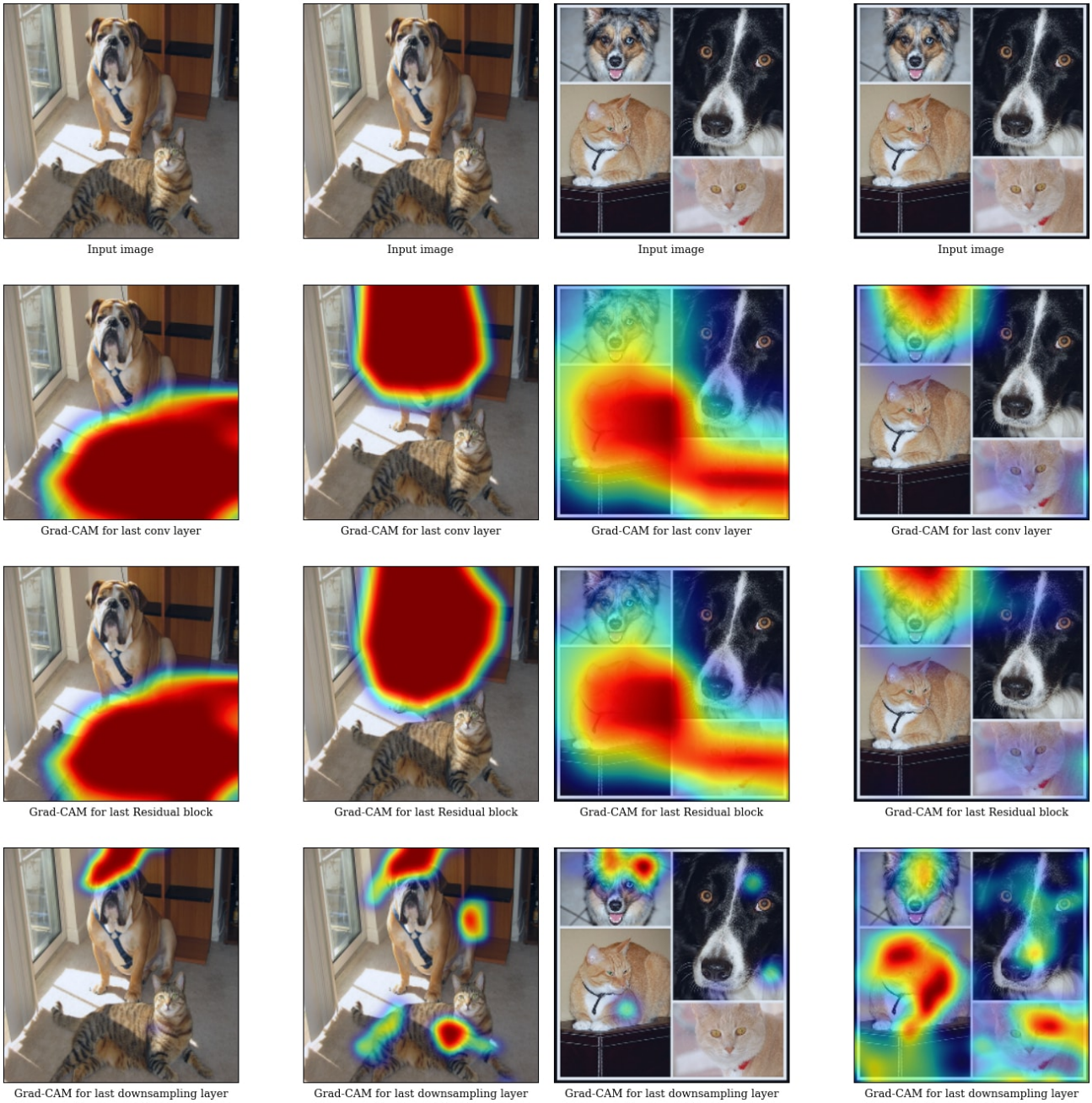


Figure A14: c-MWP and Grad-CAM visualizations for the top-5 predicted categories (shown above each image) for images sampled from the COCO validation set.



(a) Grad-CAM visualizations for the ResNet-200 layer architecture for 'tiger cat'(left) and 'boxer'(right) category.

(b) Grad-CAM visualizations for the ResNet-200 layer architecture for 'tabby cat'(left) and 'boxer'(right) category.

Figure A15: We observe that the discriminative ability of Grad-CAM significantly reduces as we encounter the downsampling layer.

[6] X. Chen, H. Fang, T.-Y. Lin, R. Vedantam, S. Gupta, P. Dollár, and C. L. Zitnick. Microsoft COCO captions: Data Collection and Evaluation Server. *arXiv preprint arXiv:1504.00325*, 2015.

[7] R. G. Cinbis, J. Verbeek, and C. Schmid. Weakly supervised object localization with multi-fold multiple instance learning. *IEEE transactions on pattern analysis and machine intelligence*, 2016.

[8] A. Das, H. Agrawal, C. L. Zitnick, D. Parikh, and D. Batra. Human Attention in Visual Question Answering: Do Humans and Deep Networks Look at the Same Regions? In *EMNLP*, 2016.

[9] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR*, 2009. 8

[10] A. Dosovitskiy and T. Brox. Inverting Convolutional Networks with Convolutional Networks. In *CVPR*, 2015.

[11] D. Erhan, Y. Bengio, A. Courville, and P. Vincent. Visualizing Higher-layer Features of a Deep Network. *University of Montreal*, 1341, 2009.

[12] H. Fang, S. Gupta, F. Iandola, R. K. Srivastava, L. Deng, P. Dollár, J. Gao, X. He, M. Mitchell, J. C. Platt, et al. From Captions to Visual Concepts and Back. In *CVPR*, 2015.

[13] C. Gan, N. Wang, Y. Yang, D.-Y. Yeung, and A. G. Hauptmann. Devnet: A deep event network for multimedia event detection and evidence recounting. In *CVPR*, 2015.

[14] H. Gao, J. Mao, J. Zhou, Z. Huang, L. Wang, and W. Xu. Are You Talking to a Machine? Dataset and Methods for Multilingual Image Question Answering. In *NIPS*, 2015.

[15] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In *CVPR*, 2014.

[16] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 12

[17] D. Hoiem, Y. Chodpathumwan, and Q. Dai. Diagnosing Error in Object Detectors. In *ECCV*, 2012.

[18] P. Jackson. *Introduction to Expert Systems*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 3rd edition, 1998.

[19] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional Architecture for Fast Feature Embedding. In *ACM MM*, 2014.

[20] E. Johns, O. Mac Aodha, and G. J. Brostow. Becoming the Expert - Interactive Multi-Class Machine Teaching. In *CVPR*, 2015.

[21] J. Johnson, A. Karpathy, and L. Fei-Fei. DenseCap: Fully Convolutional Localization Networks for Dense Captioning. In *CVPR*, 2016.

[22] A. Karpathy. What I learned from competing against a ConvNet on ImageNet. http://karpathy.github.io/2014/09/02/what-i-learned-from-competing-against-a-convnet-on-imagenet/, 2014.

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

[24] A. Kolesnikov and C. H. Lampert. Seed, expand and constrain: Three principles for weakly-supervised image segmentation. In *ECCV*, 2016. 1, 11

[25] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012. 3

[26] M. Lin, Q. Chen, and S. Yan. Network in network. In *ICLR*, 2014.

[27] 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 *ECCV*. 2014. 2, 3

[28] Z. C. Lipton. The Mythos of Model Interpretability. *ArXiv e-prints*, June 2016.

[29] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015.

[30] J. Lu, X. Lin, D. Batra, and D. Parikh. Deeper LSTM and normalized CNN Visual Question Answering model. https://github.com/VT-vision-lab/VQA_LSTM_CNN, 2015. 3

[31] J. Lu, J. Yang, D. Batra, and D. Parikh. Hierarchical question-image co-attention for visual question answering. In *NIPS*, 2016.

[32] A. Mahendran and A. Vedaldi. Visualizing deep convolutional neural networks using natural pre-images. *International Journal of Computer Vision*, pages 1–23, 2016.

[33] M. Malinowski, M. Rohrbach, and M. Fritz. Ask your neurons: A neural-based approach to answering questions about images. In *ICCV*, 2015.

[34] M. Oquab, L. Bottou, I. Laptev, and J. Sivic. Learning and transferring mid-level image representations using convolutional neural networks. In *CVPR*, 2014.

[35] M. Oquab, L. Bottou, I. Laptev, and J. Sivic. Is object localization for free? – weakly-supervised learning with convolutional neural networks. In *CVPR*, 2015.

[36] P. O. Pinheiro and R. Collobert. From image-level to pixel-level labeling with convolutional networks. In *CVPR*, 2015.

[37] M. Ren, R. Kiros, and R. Zemel. Exploring models and data for image question answering. In *NIPS*, 2015.

[38] M. T. Ribeiro, S. Singh, and C. Guestrin. "Why Should I Trust You?": Explaining the Predictions of Any Classifier. In *SIGKDD*, 2016.

[39] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. Van Den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *Nature*, 529(7587):484–489, 2016.

[40] K. Simonyan, A. Vedaldi, and A. Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *CoRR*, abs/1312.6034, 2013.

[41] K. Simonyan and A. Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. In *ICLR*, 2015. 2, 3, 9

[42] J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. A. Riedmiller. Striving for Simplicity: The All Convolutional Net. *CoRR*, abs/1412.6806, 2014. 9, 10

[43] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan. Show and tell: A neural image caption generator. In *CVPR*, 2015.

[44] C. Vondrick, A. Khosla, T. Malisiewicz, and A. Torralba. HOGgles: Visualizing Object Detection Features. *ICCV*, 2013.

[45] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *ECCV*, 2014. 9, 10

[46] J. Zhang, Z. Lin, J. Brandt, X. Shen, and S. Sclaroff. Top-down Neural Attention by Excitation Backprop. In *ECCV*, 2016. 12

[47] B. Zhou, A. Khosla, L. A., A. Oliva, and A. Torralba. Learning Deep Features for Discriminative Localization. In *CVPR*, 2016. 12