arXiv:1312.6034v2 [cs.CV] 19 Apr 2014

Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps

Karen Simonyan **Andrea Vedaldi** **Andrew Zisserman**
Visual Geometry Group, University of Oxford
{karen,vedaldi,az}@robots.ox.ac.uk

Abstract

This paper addresses the visualisation of image classification models, learnt using deep Convolutional Networks (ConvNets). We consider two visualisation techniques, based on computing the gradient of the class score with respect to the input image. The first one generates an image, which maximises the class score [5], thus visualising the notion of the class, captured by a ConvNet. The second technique computes a class saliency map, specific to a given image and class. We show that such maps can be employed for weakly supervised object segmentation using classification ConvNets. Finally, we establish the connection between the gradient-based ConvNet visualisation methods and deconvolutional networks [13].

1 Introduction

With the deep Convolutional Networks (ConvNets) [10] now being the architecture of choice for large-scale image recognition [4, 8], the problem of understanding the aspects of visual appearance, captured inside a deep model, has become particularly relevant and is the subject of this paper.

In previous work, Erhan *et al.* [5] visualised deep models by finding an input image which maximises the neuron activity of interest by carrying out an optimisation using gradient ascent in the image space. The method was used to visualise the hidden feature layers of unsupervised deep architectures, such as the Deep Belief Network (DBN) [7], and it was later employed by Le *et al.* [9] to visualise the class models, captured by a deep unsupervised auto-encoder. Recently, the problem of ConvNet visualisation was addressed by Zeiler *et al.* [13]. For convolutional layer visualisation, they proposed the Deconvolutional Network (DeconvNet) architecture, which aims to approximately reconstruct the input of each layer from its output.

In this paper, we address the visualisation of deep image classification ConvNets, trained on the large-scale ImageNet challenge dataset [2]. To this end, we make the following three contributions. First, we demonstrate that understandable visualisations of ConvNet classification models can be obtained using the numerical optimisation of the input image [5] (Sect. 2). Note, in our case, unlike [5], the net is trained in a supervised manner, so we know which neuron in the final fully-connected classification layer should be maximised to visualise the class of interest (in the unsupervised case, [9] had to use a separate annotated image set to find out the neuron responsible for a particular class). To the best of our knowledge, we are the first to apply the method of [5] to the visualisation of ImageNet classification ConvNets [8]. Second, we propose a method for computing the spatial support of a given class in a given image (image-specific class saliency map) using a single back-propagation pass through a classification ConvNet (Sect. 3). As discussed in Sect. 3.2, such saliency maps can be used for weakly supervised object localisation. Finally, we show in Sect. 4 that the gradient-based visualisation methods generalise the deconvolutional network reconstruction procedure [13].

ConvNet implementation details. Our visualisation experiments were carried out using a single deep ConvNet, trained on the ILSVRC-2013 dataset [2], which includes 1.2M training images, labelled into 1000 classes. Our ConvNet is similar to that of [8] and is implemented using their

`cuda-convnet` toolbox[1], although our net is less wide, and we used additional image jittering, based on zeroing-out random parts of an image. Our weight layer configuration is: conv64-conv256-conv256-conv256-conv256-full4096-full4096-full1000, where convN denotes a convolutional layer with N filters, fullM – a fully-connected layer with M outputs. On ILSVRC-2013 validation set, the network achieves the top-1/top-5 classification error of $39.7\%/17.7\%$, which is slightly better than $40.7\%/18.2\%$, reported in [8] for a single ConvNet.

2 Class Model Visualisation

In this section we describe a technique for visualising the class models, learnt by the image classification ConvNets. Given a learnt classification ConvNet and a class of interest, the visualisation method consists in numerically *generating* an image [5], which is representative of the class in terms of the ConvNet class scoring model.

More formally, let $S_c(I)$ be the score of the class c, computed by the classification layer of the ConvNet for an image I. We would like to find an L_2-regularised image, such that the score S_c is high:

$$\arg\max_I S_c(I) - \lambda\|I\|_2^2, \tag{1}$$

where λ is the regularisation parameter. A locally-optimal I can be found by the back-propagation method. The procedure is related to the ConvNet training procedure, where the back-propagation is used to optimise the layer weights. The difference is that in our case the optimisation is performed with respect to the input image, while the weights are fixed to those found during the training stage. We initialised the optimisation with the zero image (in our case, the ConvNet was trained on the zero-centred image data), and then added the training set mean image to the result. The class model visualisations for several classes are shown in Fig. 1.

It should be noted that we used the (unnormalised) class scores S_c, rather than the class posteriors, returned by the soft-max layer: $P_c = \frac{\exp S_c}{\sum_c \exp S_c}$. The reason is that the maximisation of the class posterior can be achieved by minimising the scores of other classes. Therefore, we optimise S_c to ensure that the optimisation concentrates only on the class in question c. We also experimented with optimising the posterior P_c, but the results were not visually prominent, thus confirming our intuition.

3 Image-Specific Class Saliency Visualisation

In this section we describe how a classification ConvNet can be queried about the spatial support of a particular class in a given image. Given an image I_0, a class c, and a classification ConvNet with the class score function $S_c(I)$, we would like to rank the pixels of I_0 based on their influence on the score $S_c(I_0)$.

We start with a motivational example. Consider the linear score model for the class c:

$$S_c(I) = w_c^T I + b_c, \tag{2}$$

where the image I is represented in the vectorised (one-dimensional) form, and w_c and b_c are respectively the weight vector and the bias of the model. In this case, it is easy to see that the magnitude of elements of w defines the importance of the corresponding pixels of I for the class c.

In the case of deep ConvNets, the class score $S_c(I)$ is a highly non-linear function of I, so the reasoning of the previous paragraph can not be immediately applied. However, given an image I_0, we can approximate $S_c(I)$ with a linear function in the neighbourhood of I_0 by computing the first-order Taylor expansion:

$$S_c(I) \approx w^T I + b, \tag{3}$$

where w is the derivative of S_c with respect to the image I at the point (image) I_0:

$$w = \left.\frac{\partial S_c}{\partial I}\right|_{I_0}. \tag{4}$$

Another interpretation of computing the image-specific class saliency using the class score derivative (4) is that the magnitude of the derivative indicates which pixels need to be changed the least

[1] http://code.google.com/p/cuda-convnet/

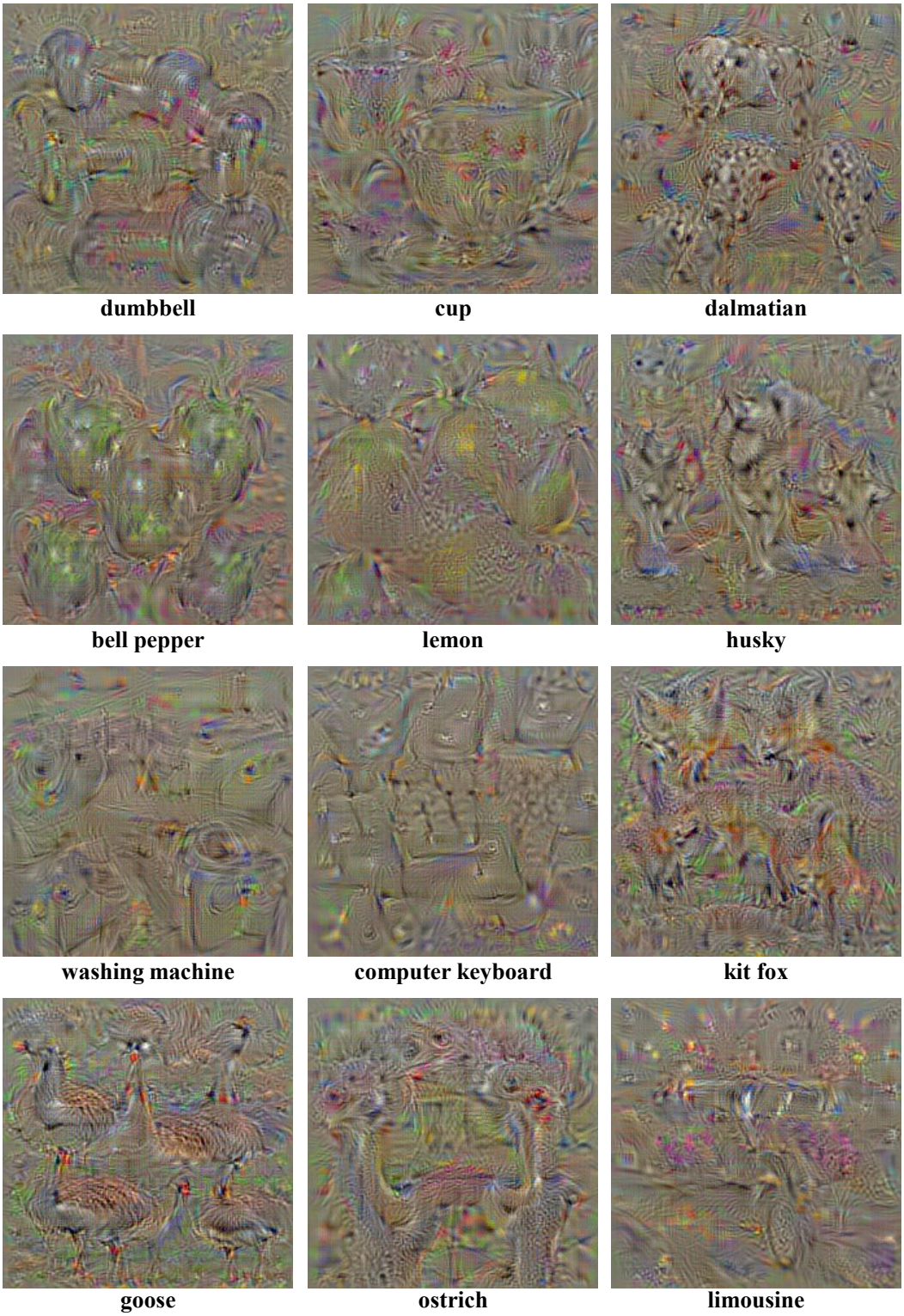

Figure 1: **Numerically computed images, illustrating the class appearance models, learnt by a ConvNet, trained on ILSVRC-2013.** Note how different aspects of class appearance are captured in a single image. Better viewed in colour.

to affect the class score the most. One can expect that such pixels correspond to the object location in the image. We note that a similar technique has been previously applied by [1] in the context of Bayesian classification.

3.1 Class Saliency Extraction

Given an image I_0 (with m rows and n columns) and a class c, the class saliency map $M \in \mathcal{R}^{m \times n}$ is computed as follows. First, the derivative w (4) is found by back-propagation. After that, the saliency map is obtained by rearranging the elements of the vector w. In the case of a grey-scale image, the number of elements in w is equal to the number of pixels in I_0, so the map can be computed as $M_{ij} = |w_{h(i,j)}|$, where $h(i,j)$ is the index of the element of w, corresponding to the image pixel in the i-th row and j-th column. In the case of the multi-channel (*e.g.* RGB) image, let us assume that the colour channel c of the pixel (i,j) of image I corresponds to the element of w with the index $h(i,j,c)$. To derive a single class saliency value for each pixel (i,j), we took the maximum magnitude of w across all colour channels: $M_{ij} = \max_c |w_{h(i,j,c)}|$.

It is important to note that the saliency maps are extracted using a classification ConvNet trained on the image labels, so *no additional annotation is required* (such as object bounding boxes or segmentation masks). The computation of the image-specific saliency map for a single class is extremely quick, since it only requires a single back-propagation pass.

We visualise the saliency maps for the highest-scoring class (top-1 class prediction) on randomly selected ILSVRC-2013 test set images in Fig. 2. Similarly to the ConvNet classification procedure [8], where the class predictions are computed on 10 cropped and reflected sub-images, we computed 10 saliency maps on the 10 sub-images, and then averaged them.

3.2 Weakly Supervised Object Localisation

The weakly supervised class saliency maps (Sect. 3.1) encode the location of the object of the given class in the given image, and thus can be used for object localisation (in spite of being trained on image labels only). Here we briefly describe a simple object localisation procedure, which we used for the localisation task of the ILSVRC-2013 challenge [12].

Given an image and the corresponding class saliency map, we compute the object segmentation mask using the GraphCut colour segmentation [3]. The use of the colour segmentation is motivated by the fact that the saliency map might capture only the most discriminative part of an object, so saliency thresholding might not be able to highlight the whole object. Therefore, it is important to be able to propagate the thresholded map to other parts of the object, which we aim to achieve here using the colour continuity cues. Foreground and background colour models were set to be the Gaussian Mixture Models. The foreground model was estimated from the pixels with the saliency higher than a threshold, set to the 95% quantile of the saliency distribution in the image; the background model was estimated from the pixels with the saliency smaller than the 30% quantile (Fig. 3, right-middle). The GraphCut segmentation [3] was then performed using the publicly available implementation[2]. Once the image pixel labelling into foreground and background is computed, the object segmentation mask is set to the largest connected component of the foreground pixels (Fig. 3, right).

We entered our object localisation method into the ILSVRC-2013 localisation challenge. Considering that the challenge requires the object bounding boxes to be reported, we computed them as the bounding boxes of the object segmentation masks. The procedure was repeated for each of the top-5 predicted classes. The method achieved 46.4% top-5 error on the test set of ILSVRC-2013. It should be noted that the method is weakly supervised (unlike the challenge winner with 29.9% error), and the object localisation task was not taken into account during training. In spite of its simplicity, the method still outperformed our submission to ILSVRC-2012 challenge (which used the same dataset), which achieved 50.0% localisation error using a fully-supervised algorithm based on the part-based models [6] and Fisher vector feature encoding [11].

4 Relation to Deconvolutional Networks

In this section we establish the connection between the gradient-based visualisation and the DeconvNet architecture of [13]. As we show below, DeconvNet-based reconstruction of the n-th layer input X_n is either equivalent or similar to computing the gradient of the visualised neuron ac-

[2] http://www.robots.ox.ac.uk/~vgg/software/iseg/

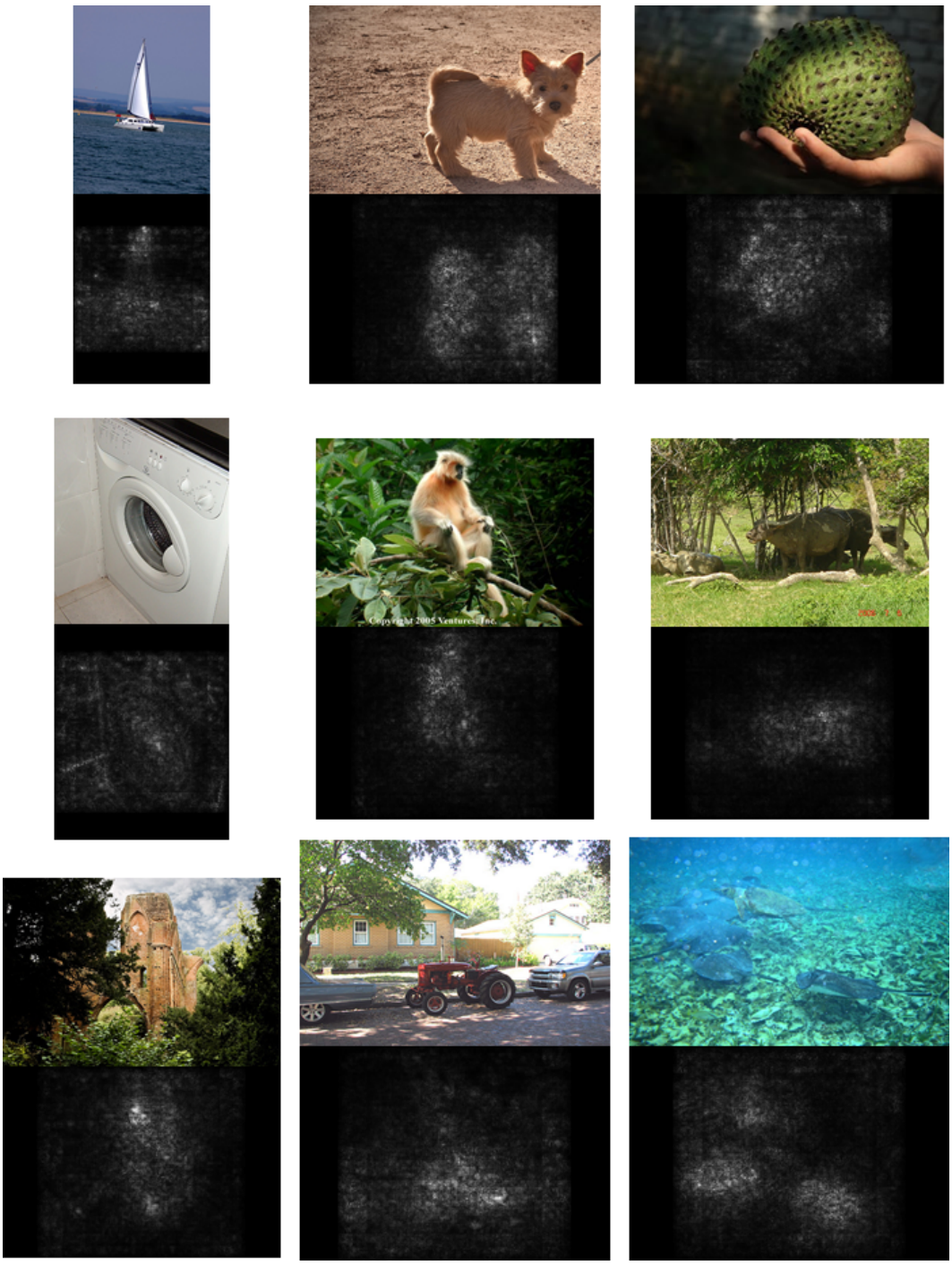

Figure 2: **Image-specific class saliency maps for the top-1 predicted class in ILSVRC-2013 test images.** The maps were extracted using a single back-propagation pass through a classification ConvNet. No additional annotation (except for the image labels) was used in training.

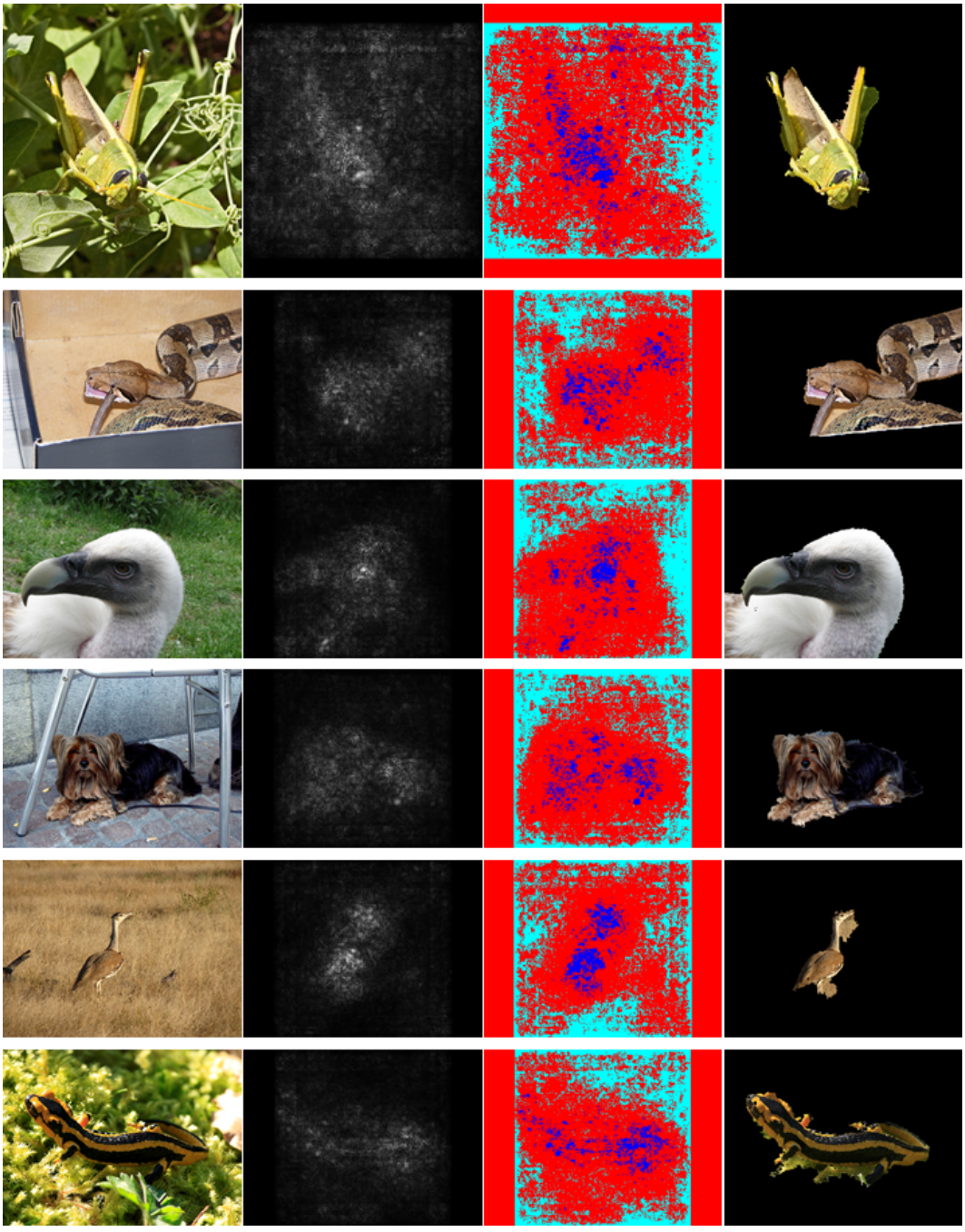

Figure 3: **Weakly supervised object segmentation using ConvNets (Sect. 3.2).** *Left:* images from the test set of ILSVRC-2013. *Left-middle:* the corresponding saliency maps for the top-1 predicted class. *Right-middle:* thresholded saliency maps: blue shows the areas used to compute the foreground colour model, cyan – background colour model, pixels shown in red are not used for colour model estimation. *Right:* the resulting foreground segmentation masks.

tivity f with respect to X_n, so DeconvNet effectively corresponds to the gradient back-propagation through a ConvNet.

For the convolutional layer $X_{n+1} = X_n \star K_n$, the gradient is computed as $\partial f / \partial X_n = \partial f / \partial X_{n+1} \star \widehat{K_n}$, where K_n and $\widehat{K_n}$ are the convolution kernel and its flipped version, respectively. The convolution with the flipped kernel exactly corresponds to computing the n-th layer reconstruction R_n in a DeconvNet: $R_n = R_{n+1} \star \widehat{K_n}$.

For the RELU rectification layer $X_{n+1} = \max(X_n, 0)$, the sub-gradient takes the form: $\partial f / \partial X_n = \partial f / \partial X_{n+1} \, \mathbf{1}\left(X_n > 0\right)$, where $\mathbf{1}$ is the element-wise indicator function. This is slightly different from the DeconvNet RELU reconstruction: $R_n = R_{n+1} \, \mathbf{1}\left(R_{n+1} > 0\right)$, where the sign indicator is computed on the output reconstruction R_{n+1} instead of the layer input X_n.

Finally, consider a max-pooling layer $X_{n+1}(p) = \max_{q \in \Omega(p)} X_n(q)$, where the element p of the output feature map is computed by pooling over the corresponding spatial neighbourhood $\Omega(p)$ of the input. The sub-gradient is computed as $\partial f / \partial X_n(s) = \partial f / \partial X_{n+1}(p) \, \mathbf{1}(s = \arg\max_{q \in \Omega(p)} X_n(q))$. Here, $\arg\max$ corresponds to the max-pooling "switch" in a DeconvNet.

We can conclude that apart from the RELU layer, computing the approximate feature map reconstruction R_n using a DeconvNet is equivalent to computing the derivative $\partial f / \partial X_n$ using back-propagation, which is a part of our visualisation algorithms. Thus, gradient-based visualisation can be seen as the generalisation of that of [13], since the gradient-based techniques can be applied to the visualisation of activities in any layer, not just a convolutional one. In particular, in this paper we visualised the class score neurons in the final fully-connected layer.

It should be noted that our class model visualisation (Sect. 2) depicts the notion of a class, memorised by a ConvNet, and is not specific to any particular image. At the same time, the class saliency visualisation (Sect. 3) is image-specific, and in this sense is related to the image-specific convolutional layer visualisation of [13] (the main difference being that we visualise a neuron in a fully connected layer rather than a convolutional layer).

5 Conclusion

In this paper, we presented two visualisation techniques for deep classification ConvNets. The first generates an artificial image, which is representative of a class of interest. The second computes an image-specific class saliency map, highlighting the areas of the given image, discriminative with respect to the given class. We showed that such saliency map can be used to initialise GraphCut-based object segmentation without the need to train dedicated segmentation or detection models. Finally, we demonstrated that gradient-based visualisation techniques generalise the DeconvNet reconstruction procedure [13]. In our future research, we are planning to incorporate the image-specific saliency maps into learning formulations in a more principled manner.

Acknowledgements

This work was supported by ERC grant VisRec no. 228180. We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Tesla K40 GPU used for this research.

References

[1] D. Baehrens, T. Schroeter, S. Harmeling, M. Kawanabe, K. Hansen, and K.-R. Müller. How to explain individual classification decisions. *JMLR*, 11:1803–1831, 2010.

[2] A. Berg, J. Deng, and L. Fei-Fei. Large scale visual recognition challenge (ILSVRC), 2010. URL http://www.image-net.org/challenges/LSVRC/2010/.

[3] Y. Boykov and M. P. Jolly. Interactive graph cuts for optimal boundary and region segmentation of objects in N-D images. In *Proc. ICCV*, volume 2, pages 105–112, 2001.

[4] D. C. Ciresan, U. Meier, and J. Schmidhuber. Multi-column deep neural networks for image classification. In *Proc. CVPR*, pages 3642–3649, 2012.

[5] D. Erhan, Y. Bengio, A. Courville, and P. Vincent. Visualizing higher-layer features of a deep network. Technical Report 1341, University of Montreal, Jun 2009.

[6] P. Felzenszwalb, D. Mcallester, and D. Ramanan. A discriminatively trained, multiscale, deformable part model. In *Proc. CVPR*, 2008.

[7] G. E. Hinton, S. Osindero, and Y. W. Teh. A fast learning algorithm for deep belief nets. *Neural Computation*, 18(7):1527–1554, 2006.

[8] A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet classification with deep convolutional neural networks. In *NIPS*, pages 1106–1114, 2012.

[9] Q. Le, M. Ranzato, R. Monga, M. Devin, K. Chen, G. Corrado, J. Dean, and A. Ng. Building high-level features using large scale unsupervised learning. In *Proc. ICML*, 2012.

[10] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[11] F. Perronnin, J. Sánchez, and T. Mensink. Improving the Fisher kernel for large-scale image classification. In *Proc. ECCV*, 2010.

[12] K. Simonyan, A. Vedaldi, and A. Zisserman. Deep Fisher networks and class saliency maps for object classification and localisation. In *ILSVRC workshop*, 2013. URL http://image-net.org/challenges/LSVRC/2013/slides/ILSVRC_az.pdf.

[13] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. *CoRR*, abs/1311.2901v3, 2013.