Supplement

1 Details on the Computation of Fisher Vector Embeddings

The representation of an image data set as FVs starts with the computation of a Gaussian Mixture Model (GMM) as a soft vocabulary of visual prototypes. The K components $\lambda = \{(\pi_k, \mu_k, \Sigma_k)\}_{k=1..K}$ of the GMM are fitted onto a set of local descriptors l extracted from the training images, where μ_k is the mean vector of the kth mixture component and Σ_k its covariance matrix which is assumed to be diagonal, e.g. $diag\,(\Sigma_k) = \sigma_k$ and all off-diagonal entries are 0. The parameter π_k is the mixture weight of component k, with $\sum_k \pi_k = 1$ and $\forall k : \pi_k \geq 0$. A *full* FV descriptor of an image measures the average 0th (soft mapping weight), 1st (deviation from mean) and 2nd (variance) moment of all local descriptors sampled from the image area [10].

$$\begin{aligned}
\Psi_{\pi_k}(l) &= \frac{1}{\sqrt{\pi_k}} \left(\gamma_k(l) - \pi_k\right) \\
\Psi_{\mu_k}(l) &= \frac{1}{\sqrt{\pi_k}} \gamma_k(l) \left(\frac{l - \mu_k}{\sigma_k}\right) \\
\Psi_{\sigma_k}(l) &= \frac{1}{\sqrt{\pi_k}} \gamma_k(l) \frac{1}{\sqrt{2}} \left(\frac{(l - \mu_k)^2}{\sigma_k^2} - 1\right)
\end{aligned} \tag{1}$$

with $\Psi_{\pi_k}(l) \in \mathbb{R}$, $\{\, l,\ \Psi_{\mu_k}(l),\ \Psi_{\sigma_k}(l)\ \} \in \mathbb{R}^D$ and $\gamma_k(l)$ returning the soft assignment of l to the kth mixture component [10]. The FV embedding $\Psi_\lambda(l)$ of a local descriptor l is achieved by concatenating the mapping outputs for all K components into a $(1 + 2D)K$ dimensional vector, such that

$$\Psi_\lambda(l) = \Bigg[\underbrace{\Psi_{\pi_1}(l)\ ,\ \ldots\ , \Psi_{\pi_K}(l)}_{K \text{ scalar entries}},\ \underbrace{\Psi_{\mu_1}(l),\ \ldots\ , \Psi_{\mu_K}(l)}_{K \text{ concatenated } D\text{-dimensional vectors}}\ ,\ \underbrace{\Psi_{\sigma_1}(l),\ \ldots\ , \Psi_{\sigma_K}(l)}_{K \text{ concatenated } D\text{-dimensional vectors}} \Bigg] \tag{2}$$

The final FV of an image is obtained by averaging over all $\Psi_\lambda(l)$ resultung in the *raw* FV representation[1], followed by power normalization to reduce the sparsity of the descriptor and and ℓ_2-normalization to improve prediction performance [8]:

$$\begin{aligned}
x &= \frac{1}{|L|} \sum_{l \in L} \Psi_\lambda(l) && \text{mapping aggregation} \\
x &\leftarrow sign(x)|x|^{\frac{1}{2}} && \text{power normalization} \\
x &\leftarrow \frac{x}{\|x\|_2} && \ell_2\text{-normalization}
\end{aligned}$$

2 On the Equality of the Normalization Steps and the Hellinger's Kernel

The work of [10] references the equality of the application of the Hellinger's (Bhattacharyya) kernel function to a raw FV descriptor and the improved FV (power- and ℓ_2-normalized FV) with a linear kernel

[1] *raw* FV in contrast to the *improved* FV, as used in the main document to distinguish between the both.

function on top. Here, we explicitly show this equality. Assume a raw FV x, the component-wise absolute of an input variable $|\cdot|$, and the ℓ_p-norm of a vector $\|\cdot\|_p$. The improved FV defines itself as the application of power-normalization followed by ℓ_2-normalization on top of a raw FV, e.g. interpreted as a mapping function $\Phi(\cdot)$ we receive

$$\begin{aligned}\Phi(x) &= \frac{sign(x)|x|^{\frac{1}{2}}}{\|sign(x)|x|^{\frac{1}{2}}\|_2} = \frac{sign(x)\sqrt{|x|}}{\sqrt{\sum_d \underbrace{sign(x_{(d)})^2}_{=1}|x_{(d)}|\underbrace{^{\frac{1}{2}\cdot 2}}_{=1}}} \\ &= sign(x)\sqrt{\frac{|x|}{\sqrt{\sum_d |x_{(d)}|}^2}} = sign(x)\sqrt{\frac{|x|}{\|x\|_1}}\end{aligned} \tag{3}$$

and with that

$$k(x,y) = \sum_d \Phi(x)_d\Phi(y)_d = \sum_d sign(x_{(d)}y_{(d)})\underbrace{\sqrt{\frac{|x_{(d)}|}{\|x\|_1}\cdot\frac{|y_{(d)}|}{\|y\|_1}}}_{\text{Hellinger's kernel}} \tag{4}$$

which enables us to compute component-wise relevance scores for each FV dimension d by attributing the power normalization to the kernel function as

$$R_d^{(3)} = \sum_i \alpha_i\Phi(x_i)_d\Phi(x)_d + \frac{b}{D} \tag{5}$$

In the case of an improved FV mapping (or the Hellinger's (Bhattacharyya) kernel function), the support vectors need not to be known explicitly, since $w = \sum_i \alpha_i\Phi(x_i)$, further simplifying Equation to

$$R_d^{(3)} = w_d\Phi(x)_d + \frac{b}{D} \tag{6}$$

3 Details on Relevance Decomposition for Fisher Vectors

As already stated within the main document, the relevance decomposition for FV embeddings follows the definitions and constraints of [1]. See [6] for a more in depth explanation. As already stated in Section 2, relevance values $R_d^{(3)}$ for each dimension d of the FV representation of an image can be easily computed as

$$R_d^{(3)} = \sum_i \alpha_i\Phi(x_i)_d\Phi(x)_d + \frac{b}{D} \tag{7}$$

The next step towards local explanations in pixel space is the computation of relevalce scores $R_l^{(2)}$ for each local descriptor l contributing to the computation of the FV used for prediction. The work of [1] introduces the notion of a mapping function $m_{(d)}(l)$ relating l to dimension d in output space for Bag of Feature (BoF) models assuming positive mapping outputs exclusively. Reformulating $\Psi_\lambda(l)$ in terms of $m_{(d)}(l)$ to fit it into the LRP framework facilitates the computation of local feature relevance scores proportionally to their forward mapping contributions

$$m_{(d)}(l) = \begin{cases} \frac{1}{\sqrt{\pi_k}}(\gamma_k(l) - \pi_k) & ;\ d = k,\ k \in [1,K] \\ \frac{1}{\sqrt{\pi_k}}\gamma_k(l)\left(\frac{l_{(r)}-\mu_{k,(r)}}{\sigma_{k,(r)}}\right) & ;\ d = K + D(k-1) + r, k \in [1,K], r \in [1,D] \\ \frac{1}{\sqrt{\pi_k}}\gamma_k(l)\frac{1}{\sqrt{2}}\left(\frac{(l_{(r)}-\mu_{k,(r)})^2}{\sigma^2_{k,(r)}} - 1\right) & ;\ d = (1+D)K + D(k-1) + r, k \in [1,K], r \in [1,D] \end{cases} \tag{8}$$

With that the proposed decomposition formula for BoF mappings can directly be applied with only a minor adaption of $Z(x)$ from Equation (26) in [1] to consider that $m_d(l)$ may output mapping weights of both positive and negative signs for FVs :

$$R_l^{(2)} = \sum_{d \notin Z(x)} R_d^{(3)} \frac{m_{(d)}(l)}{\sum_{l' \in L} m_{(d)}(l')} + \xi \tag{9}$$

where

$$Z(x) = \left\{ d \,|\forall l : m_{(d)}(l) = 0 \right\} \text{ and } \xi = \sum_{d \in Z(x)} R_d^{(3)} \frac{1}{|L|}$$

Pixel-wise relevance scores $R_p^{(1)}$ are then computed by uniformly distributing for all local features l the relevance scores $R_l^{(2)}$ onto the set of pixels p covered by the receptive field of l

$$\begin{aligned} L(p) =& \{l | p \in \text{area}(l)\} \\ R_p^{(1)} =& \sum_{l \in L(p)} \frac{R_l^{(2)}}{|\,\text{area}(l)|} \end{aligned} \tag{10}$$

resulting in a *heatmap* which can be visualized.

Above approach to compute values $R_l^{(2)}$ can in general – when $m_{(d)}(l)$ outputs values of both signs – be expected to be numerically problematic when $\sum_{l'} m_{(d)}(l')$ becomes very small due to individual mappings $m_{(d)}(l')$ cancelling each other out. Divisions close to zero would then pronounce otherwise insignificant local descriptor weights for relevance distribution. This problem is known in the context of neural network relevance decompositions and has been discussed in [1], which we will follow for extending Equation 9 to better satisfy the task at hand. Figure 1 will then compare how well the computed heatmap resulting from each decomposition method represents the classifier's perception according to the measurement procedure described in Section 4.

3.1 ϵ-Stabilized Decomposition

The first and probably most straight forward adaption of the decomposition formula introduced in [1] is the introduction of a numerical stabilizer $\epsilon > 0$ to prevent (near) zero divisions

$$R_l^{(2)} = \sum_{d \notin Z(x)} R_d^{(3)} \frac{m_{(d)}(l)}{\sum_{l' \in L} m_{(d)}(l') + \gamma} + \xi \tag{11}$$

with $\gamma = \epsilon \cdot sgn\left(\sum_{l' \in L} m_{(d)}(l')\right)$, $sgn(y) = 1$ if $y \geq 0$ and -1 else. An obvious drawback of this approach is, however, that it violates the conservation constraints set by LRP, as varying amounts of relevance can be absorbed or generated by ϵ.

3.2 Absolute Value Decomposition

Our second alternative to resolve the issue of mapping weight cancellation is to only consider the absolute of a descriptor's mapping contribution, effectively removing all cases of division by zero not already covered by Equation 9. As a further benefit, no additional parameters need to be defined and no relevance is lost as in Eq. 11

$$R_l^{(2)} = \sum_{d \notin Z(x)} R_d^{(3)} \frac{|m_{(d)}(l)|}{\sum_{l' \in L} |m_{(d)}(l')|} + \xi \tag{12}$$

The drawback of this method is the loss of information encoded in the mapping signs.

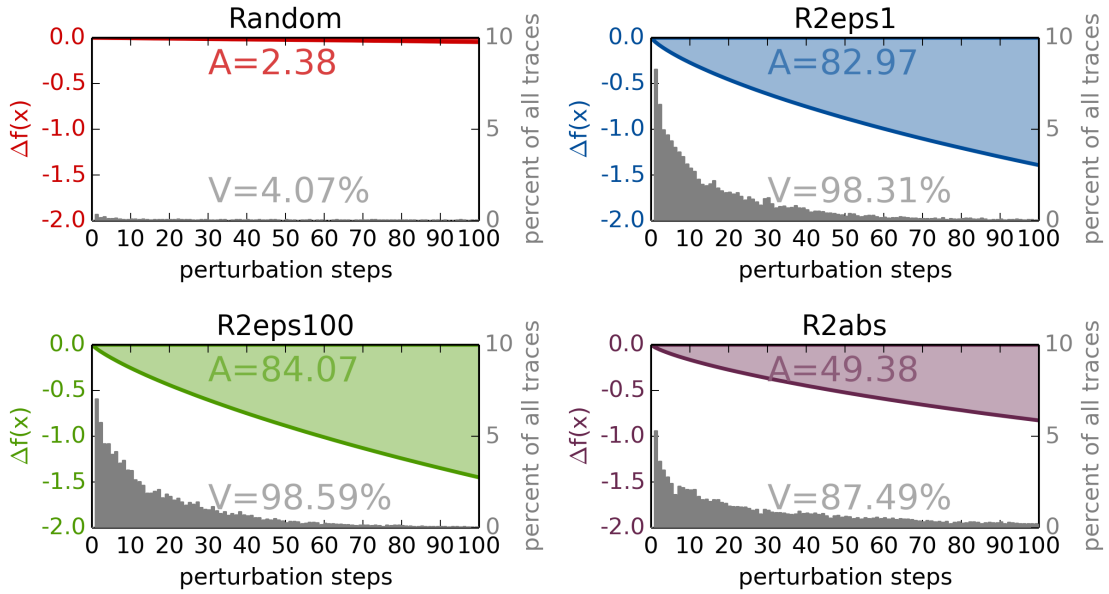

Figure 1: Heatmap Quality Measurements for Fisher Vectors. The value A measures the area above the curve between the original prediction $f(x)$ and the averaged perturbed prediction at step i in the sequence of regions. $f(x) - f(x^{(i)}_{\text{MoRF}})$. V represents the fraction of all perturbation sequences for which the prediction switched sign at some step in the sequence, with the gray bar chart showing how many sample traces changed class at each point of measurement. Despite violating the relevance conservation principle we have found the ϵ-stabilized decomposition to produce the best heatmaps for the FV classifer. Therefore heatmaps computed with Equation 11 and $\epsilon = 100$ for all further evaluations.

4 Measuring the Quality of a Heatmap

With *pixel flipping* a procedure has been introduced in [1], which demonstrated that the heatmaps computed for the MNIST [5] data set are able to successfully identify the properties of the input image determining the decision of the classifier. The idea behind this data deterioration approach guided by relevance scores has then been extended in [9] to an evaluation procedure to compare alternatives for LRP decomposition and image perturbation approaches applied to a deep neural network classifier.

Assessing the quality of a heatmap using image deterioration guided by the heatmap itself follows the semantic behind LRP: Let us assume a classifier which predicts a learned target if $f(x) > 0$. The application of LRP to $f(\cdot)$ with an input point x should then attribute (the highest) positive values $R_i^{(k)}$ to components of x which are (most) important to the positive decision of $f(\cdot)$. By intuition, removing that information should cause the predictor output to become less positive. Considering the values $R_i^{(k)}$ as a *ranking* of importance of components of the input , e.g. the most important components receive the largest proportions of relevance, then this ranking is optimal if the output of $f(\cdot)$ decreases faster than any other ranking with ongoing removal of input information with a removal order $\vec{o}$ determined by ordering the values $R_i^{(k)}$ descendingly. Since different decomposition methods (e.g. randomly picked component orders) can produce different orderings of importance, an algorithm of relevance-guided data deterioration can be used to compare how well each decomposition variant explains the reasoning of the classifier. We formally define this deterioration algorithm as

$$
\begin{aligned}
x^{(0)}_{\text{MoRF}} &= x \\
\forall\, 1 \leq i \leq I \;:\; x^{(i)}_{\text{MoRF}} &= g\left(x^{(i-1)}_{\text{MoRF}}, \vec{o}_i\right)
\end{aligned}
\tag{13}
$$

with $g(\cdot)$ being a perturbation function to remove or exchange data, $x^{(i)}_{\text{MoRF}}$ representing a data point after i

steps of removal of the **mo**st **r**elevant information **f**irst, determined by an ordering $\vec{o}$, have been performed. We compute a measure of quality of a heatmap A by comparing the prediction $f(x)$ to the prediction on the altered input $x^{(i)}_{\text{MoRF}}$ and then computing the area between those two curves by integrating over the points of measurement:

$$A = \frac{1}{I}\sum_{i=1}^{I}\Big(f(x) - f(x^{(i)}_{\text{MoRF}})\Big) \tag{14}$$

The value A increases with the reaction of the classifier to a change in its input and therefore higher values for A encode a better representation of the classifier decision in terms of back propagated relevance.

4.1 Local Feature Replacement

Previously used variants of pixel flipping have – hence the name – operated on pixel level exclusively to evaluate heatmaps for neural network type classifiers. Those classifiers – namely deep neural networks – directly received the pixel values of an image as inputs and the application of LRP produce pixel-accurate heatmaps. While [1] exchanges almost binary pixel values of the 28×28 pixel sized MNIST pixels by inverting the state of each pixel, [9] studies different image perturbation strategies on 227×227 pixel large color images showing photographic scenes and structures. Here, questions such as the number of pixels to exchange, their spatial grouping and how to best replace those pixels have been raised and the overall complexity of the problem of finding the *right* perturbation strategy is discussed.

When a feature extraction pipeline (in the classical sense: dense local feature extraction $\rightarrow$ mapping $\rightarrow$ pooling) is part of the predictor operating on images we face a set of distinct problems. For example, exchanging one pixel causes a recalculation and mapping of all local descriptors covering that pixel in order to measure the effect to the predictor output. Assuming a meaningful heatmap, pixel positions with leading relevance scores are in general not sparsely scattered all over the image area but rather grouped in the same image area over the extend of a patch of neighbouring pixels. This is even more so true due to the computation of pixel relevance scores $R^{(1)}_p$. The same local descriptors are very likely to be recalculated over and over again, even though the change in a single pixel will not have much of an effect. This causes a considerable computational cost to assess a heatmap, especially when local descriptors are sampled densely and at multiple scales. An obvious solution to this problem is to exchange a group of pixels at a time, yet this raises the questions of how many pixels to exchange, due to which grouping and with what replacement strategy? Removing single pixels is out of the question, since this would effectively damage the integrity of the image itself. Also the type of local descriptor needs to be considered, e.g. do SIFT features encode shape information well. By blurring an area important structural information might be removed, and a bias towards another class (or even the image's true class) might be introduced, by the blurred area resembling e.g. sky, water or cloth, interfering with the evaluation for classes *aeroplane*, *boat* and *person*, respectively. Vice versa, setting selected pixels to random color values might create high contrast gradients in the image, affecting the evaluation of certain object classes identifying themselves via sharp and well pronounced edges.

We aim to avoid all those problems by not exchanging pixel values, but local descriptors instead. Firstly, we assume the descriptors forming the orderless descriptor set L serving as an intermediate image representation to be the result independent events which can also be exchanged one at a time without affecting each other's meaning. This is not the case when individual pixels are exchanged. Secondly, in contrast to pixel scores $R^{(1)}_p$ the local feature scores $R^{(2)}_l$ are also the direct result of the relevance decompositions of the mapping function $m_d(l)$ of which we wish to compare the alternative options. Se can use the GMM λ representing the distribution of local descriptors as a generative model to draw (already dimensionality-reduced) replacements for the features to be exchanged, resulting in believable data points $x^{(i)}_{\text{MoRF}}$ close to the data manifold. What remains to do is to create an ordering of local descriptors to replace and to update the FV x. Algorithm 1 outlines this perturbation and evaluation strategy. Note, that the transformation $\Phi(\cdot)$ is applied to $x^{(i)}_{\text{MoRF}}$ before feeding it as an input to $f(\cdot)$ to compute A in Equation 13. To measure the impact of Algorithm 1 when applied to an image, we replace the first 10000 (w.r.t. to the ordering $\vec{o}$) local descriptors extracted from the image in batches of 100, resulting in 100 points of measurement. As

Algorithm 1 Local Feature Resampling

Input: local features L, GMM λ, number of replacements I, replacement order $\vec{o}$
$x_{\text{MoRF}}^{(0)} \leftarrow \frac{1}{|L|} \sum_{l \in L} \Psi_\lambda(l)$
for i from 1 to I **do**
    $l' \leftarrow$ feature to be replaced: $L(\vec{o_i})$
    $l_\lambda \leftarrow \lambda$.generate()
    $x_{\text{MoRF}}^{(i)} \leftarrow x_{\text{MoRF}}^{(i-1)} + \frac{1}{|L|}\Psi_\lambda(l_\lambda) - \frac{1}{|L|}\Psi_\lambda(l')$
end for

a rough estimate, from about 15000 up to 100000 features are computed for each image of the PASCAL VOC 2007 test set using the dense sampling approach of the encoding evaluation toolbox [2], with 69000 local features being sampled per image on average.

To reduce random effects influencing the measurement process, we repeat the experiment five times, in total recording 28870 image perturbation traces on all true positive predictions [2] after optimizing the prediction threshold w.r.t. to the EER measure.

5 Details on the Neural Network Retraining

The starting point was the BVLC reference caffe net as provided with the caffe package [4]. Training mode was multi-label training instead of the usual competitive multi-class training because for PASCAL VOC multiple classes can be present in one image. The training criterion was the sum of hinge losses over all 20 classes in the Pascal VOC data set. Note that this required to use a customized image data layer and a customized hinge loss layer.

One general problem for training and testing is the question how to score an image and what data to use for training. A second problem is how to generate patches matching the square receptive field size from non-quadratic images. One general approach to generate non-quadratric images is to ignore the aspect ratio and to use warping in order to transform a non-quadratic patch into a quadratic one such as in [3].

In order to have maximal comparability to Fisher vectors which do not use warping and process an image as a whole we decided for a simpler setup which is close to the setup used by Fisher vectors and which preserved the aspect ratio of patches used during training and testing, irrespective of the fact that other setups may have resulted in somewhat higher performance of the neural network.

Training data:

As we were interested in training a setup such that the neural network is able to use context, we refrained from training the network with image patches around the scale of a bounding box and smaller. We decided not to use the information about object bounding boxes for generating training data because for Fisher vectors this information was not used for generating training data. Instead each image was rescaled such that the largest side had 256 pixels. The smaller side was padded at its boundaries by the nearest pixel. From this modified image 4 edge and one center crop was taken. After mirroring the image, this was repeated. This resulted in 10 images per training image. This is a compromise to ensure a sufficiently large sample size for retraining, as it is known that neural networks excel typically at higher training sample sizes.

Testing data:

The heatmap was computed using one center crop only.

As for measurement of mean average precision, results depend on how to score one image at test time. Note that it is common to compute an average score over many crops of the image.

[2] 5774 TP predictions in total over all 20 classes with 5 repetitions for each case. Single images may show multiple and correctly detected object classes at once.

Resizing the largest side of the image to 256 pixels and using the 227×227 center crop only for each image resulted in the 72.12 mAP reported in the main paper. Note that this setup corresponded to the setup used for computing the heatmaps, so this was used for the sake of comparability.

Using a different test strategy, namely resizing the smallest side of the image to 256 pixels, then computing an average over a sliding window with stride of 20 pixels, resulted in an increase of ranking performance to 75.9 mAP. This strategy used merely 12 - 35 test patches per image. Using approaches with several hundred test windows, such as 500 windows in [7] would probably have resulted in much higher mAP scores, however we did not consider a higher score relevant for the main message of the paper focused on context usage. In view of the considerably increased computation time for such approaches we refrained from them.

6 More examples for heatmaps for deep neural networks for various architectures.

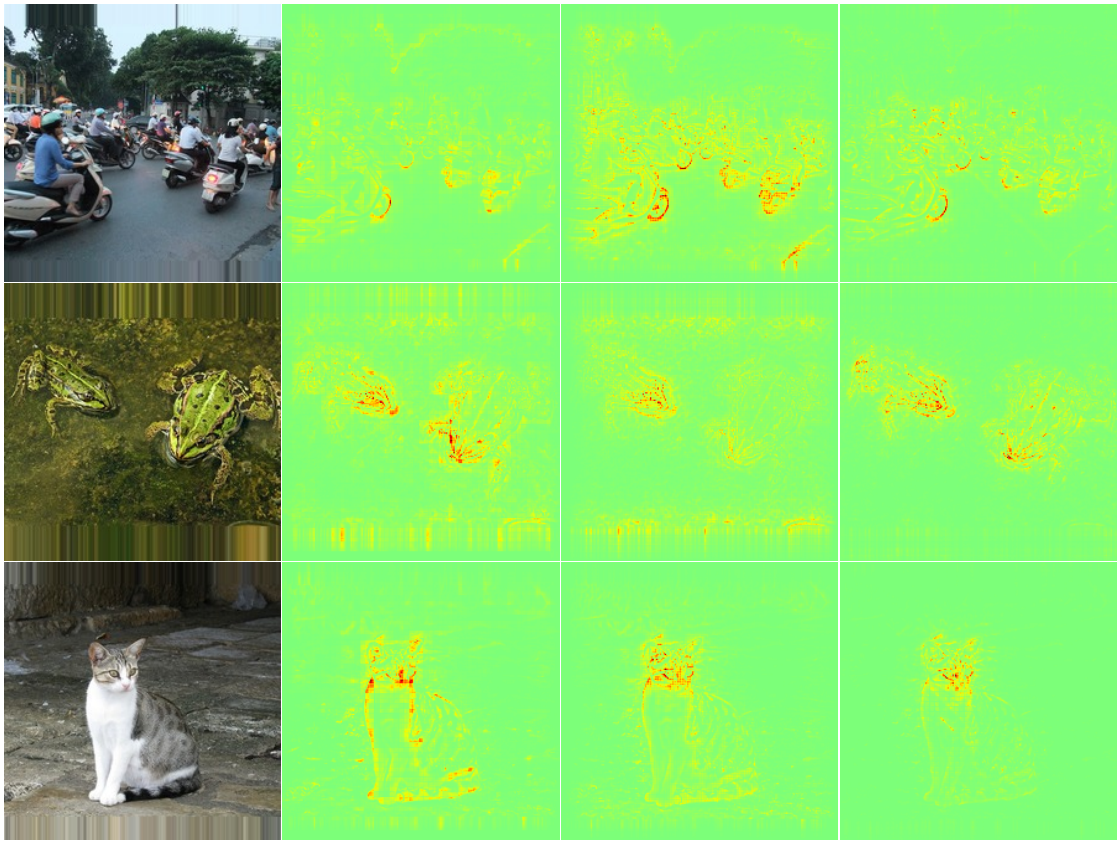

Figure 2: Comparison of different pretrained models on ImageNet for classes “scooter”, “frog”, and “cat”. From left to right: Input, heatmaps for BVLC CaffeNet, VGG CNN S and GoogleNet.

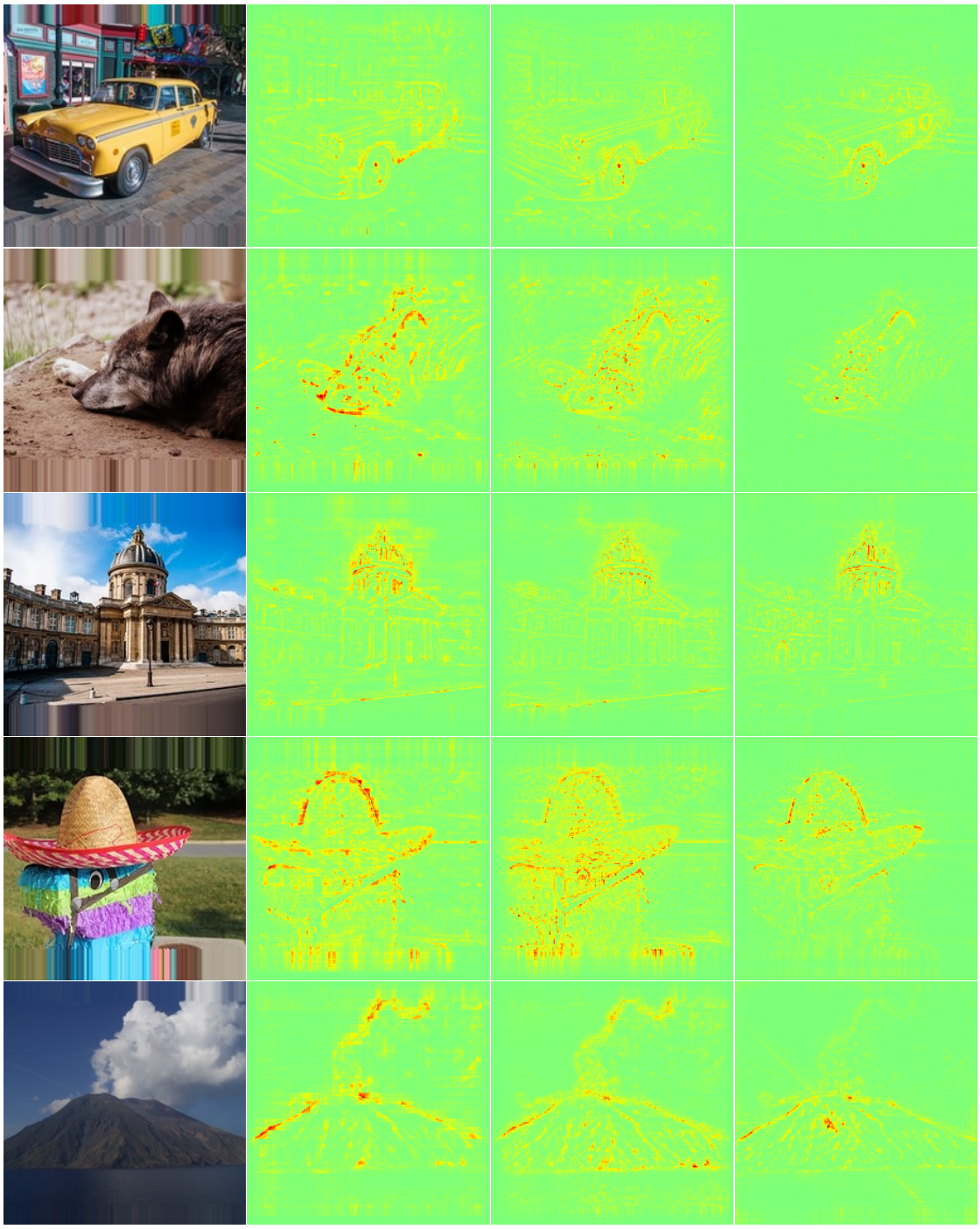

Figure 3: Comparison of different pretrained models on ImageNet for classes “taxi”, “wolf”,“palace”, “sombrero” and “volcano”. From left to right: Input, heatmaps for BVLC CaffeNet, VGG CNN S and GoogleNet.

References

[1] S. Bach, A. Binder, G. Montavon, F. Klauschen, K.-R. Müller, and W. Samek. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PLOS ONE*, 10(7):e0130140, 2015.

[2] K. Chatfield, V. S. Lempitsky, A. Vedaldi, and A. Zisserman. The devil is in the details: an evaluation of recent

feature encoding methods. In *BMVC*, page 8, 2011.

[3] R. B. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *2014 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2014, Columbus, OH, USA, June 23-28, 2014*, pages 580–587, 2014.

[4] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. B. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. In *Proceedings of the ACM International Conference on Multimedia, MM '14, Orlando, FL, USA, November 03 - 07, 2014*, pages 675–678, 2014.

[5] Y. LeCun and C. Cortes. The MNIST database of handwritten digits. http://yann.lecun.com/exdb/mnist/, 1998.

[6] G. Montavon, S. Bach, A. Binder, W. Samek, and K.-R. Müller. Explaining nonlinear classification decisions with deep taylor decomposition. *arXiv preprint arXiv:1512.02479*, 2015.

[7] M. Oquab, L. Bottou, I. Laptev, and J. Sivic. Learning and transferring mid-level image representations using convolutional neural networks. In *CVPR*, pages 1717–1724, 2014.

[8] F. Perronnin, J. Sánchez, and T. Mensink. Improving the fisher kernel for large-scale image classification. In *Computer Vision–ECCV 2010*, pages 143–156. Springer, 2010.

[9] W. Samek, A. Binder, G. Montavon, S. Bach, and K. Müller. Evaluating the visualization of what a deep neural network has learned. *CoRR*, abs/1509.06321, 2015.

[10] J. Sánchez, F. Perronnin, T. Mensink, and J. Verbeek. Image classification with the fisher vector: Theory and practice. *IJCV*, 105(3):222–245, 2013.