Interpreting Neural Network Classifications with Variational Dropout Saliency Maps

Chun-Hao Chang
University of Toronto
Vector Institute
kingsley@cs.toronto.edu

Elliot Creager
University of Toronto
Vector Institute
creager@cs.toronto.edu

Anna Goldenberg
University of Toronto
Vector Institute
anna.goldenberg@utoronto.ca

David Duvenaud
University of Toronto
Vector Institute
duvenaud@cs.toronto.edu

Abstract

Deep neural networks are effective at classification across many domains, but they are also opaque in the sense of being seen as "black boxes" even when the training data and model architecture are available. Saliency maps are a tool for interpreting neural network classification that, given a particular input example and output class, score the relevance of each input dimension to the resulting classification. Recent work defined salient inputs as those yielding the greatest change in the classification output when replaced with some reference value; this value is chosen heuristically (e.g., background color, Gaussian noise, or blurred version of the input) and thus biases the saliency computation. We generalize this approach by extending the notions of 'replacement' and 'reference': first we cast the input replacement in a dropout framework and use variational inference to learn a distribution over dropped-out (replaced) inputs from the data; then we express the reference value as the output of a generative model, which can be learned from data to mitigate the effect of the reference value biasing the saliency map. We then propose a new model-agnostic saliency map that uses both extensions in tandem. We show the resulting saliency maps for a digit classification network pre-trained on MNIST and compare our results against other methods both qualitatively and quantitatively.

1 Introduction

Deep neural networks (DNN) recently surpassed human-level performance in image classification (Russakovsky et al., 2015). However, despite promising results from the computer vision community, the opacity of DNN has limited their adoption in domains that rely on highly-trained human experts such as biology and health care. These application areas require that the experts can *interpret* and *trust* models that yield high prediction accuracy. Here we focus on interpretability via *saliency maps*, which compute a per-input-dimension relevancy score given a particular input example and classification outcome.

Early saliency maps for convolutional DNNs (CDNNs) used the gradients of the classification scores (or softmax inputs) with respect to the input pixels (Simonyan et al., 2013). This notion of saliency captures a sensitivity of the in-class activations to small local changes to the input. However, these methods lead to bias when saturation or gradient discontinuity happens in the nonlinear DNN (Shrikumar et al., 2017).

More recent work considered saliency as the change of the in-class probability that results when the substituting the input for (or convexly mixing it with) some *reference* value, which may be input-dependent. Reference values were chosen heuristically, e.g., graying (Zhou et al., 2014), blurring, the mean of the image (Fong and Vedaldi, 2017) or some value based on human understanding (Shrikumar et al., 2017).

Zintgraf et al. (2017) instead considered the saliency of one component of the input as the resulting difference in classification output when that component is treated as unobserved and marginalized out. Along the way they implicitly express a generative model for inputing the missing component as the conditional distribution of the missing component given rest of the input. Our work follows this approach, but we sample from a generative model, which may be learned from data instead of than marginalizing over its support. The goal is to choose a reasonably likely value for the imputed component rather than relying on a heuristic, which hopefully mitigates the bias in the saliency map.

Dabkowski and Gal (2017) proposed two objectives for optimizing the saliency map:

- Smallest deletion region (SDR) considers a saliency map as the answer to a question like "what is the smallest region of the input that could be swapped for (or mixed with) reference values such that the in-class probability is minimized?"
- Smallest supporting region (SSR) instead asks the question "what is the smallest region of the input that that could be substituted into the reference such that the in-class probability is maximized?"

We show these two different objectives have advantages in different settings.

In this paper, we make several contributions. First, we provide a new model-agonistic framework to visualize any differentiable classifier based on variational Bernoulli dropout (Gal and Ghahramani, 2016). Second, we consider the role of a generative model in imputing the reference value. Third, we combine these contributions to produce saliency maps for a digit-classifying DNN trained on MNIST; we demonstrate matched performance with existing approaches when the generative model is simple (the regime in which existing approaches operate), and show how our approach is amenable to more expressive generative imputation of inputs.

2 Related work

Gradient-based approaches (Simonyan et al., 2013; Selvaraju et al., 2016) derive a saliency map for a given input example and class target by computing the gradient of the classification probability (or softmax input) with respect to each component (e.g., pixel) of the input. However, the reliance on the local gradient information induces a biases due to gradient saturation or discontinuity in the DNN activations (Shrikumar et al., 2017).

Reference-based approaches compute saliency according to how the output classification scores change when inputs are swapped (or convexly mixed) with a reference value, as described above. For example, Shrikumar et al. (2017) considers the reference as the background color (e.g., black in MNIST) and locally linearly approximates the classification difference before and after swapping the reference using an algorithm that resembles backpropagation. This method runs efficiently and addresses some problems of gradient discontinuity. However, the reliance on a heuristic reference is a drawback; it is not obvious how to select such a reference for datasets beyond MNIST, especially in applications besides image processing. Also, the linear approximation of the activations biases the saliency estimate.

Zintgraf et al. (2017) computes the saliency of a component (or region of components, e.g., an image patch) by treating it as unobserved and marginalizing it out. This probabilistic interpretation of imputation is attractive from a modeling perspective and it yields qualitatively compelling saliency maps, but at great computational cost since the algorithm iterates over components with a marginalization in the inner loop. To mitigate the cost of marginalization, Zintgraf et al. (2017) impose a factorization on the conditional distribution such that inputs are only conditionally depend on nearby inputs. However, this approximation isn't well suited to the case where multiple distant components or regions jointly support the classification, which occurs in medical imaging applications, e.g., MRI.

Fong and Vedaldi (2017) proposes to solve for saliency by optimizing over perturbations that "meaningfully" change the classification outcome; we refer to their method as the Black Box Meaningful

Perturbation saliency map, or BBMP. They expresses the perturbed input as a pixel-wise convex combination of the original input with a reference, and offer three heuristics for choosing the reference: mean input pixel value (typically grey), Gaussian noise, or blurring the input. They then solve for the saliency map as the parameters of the pixel-wise mixture (the "mask") that optimize the SDR objective, with the optimization further regularized to discourage trivial solutions ("artifacts"). Using the blurred input as reference yields qualitatively interesting maps; however, this heuristic biases the saliency estimate because it contains low-frequency information from the original input image.

Dabkowski and Gal (2017) similarly cast saliency as an optimization, where their objective contains both SDR and SSR terms, plus regularization. Rather than inferring the mask per image, they amortize the cost of inference by learning parameters of an auxiliary neural network that efficiently computes the saliency map at test time, which they claim supports real-time applications. They also uses heuristic references such as random constant color with high-frequency noise.

3 Proposed method

3.1 Saliency

We consider two definitions of saliency corresponding to the SDR and SSR objectives (Dabkowski and Gal, 2017), which were mentioned in section 1; we now formalize these objectives. We define a mask $\boldsymbol{z}$ as a binary vector with value 1 when the input image pixel value is used and 0 when the reference image pixel value is used. Consider an input image $\boldsymbol{x}$, mask $\boldsymbol{z}$, class c, reference $\hat{\boldsymbol{x}}$ and CDNN classifier $\mathcal{M}$ with output probabilities $p_{\mathcal{M}}(c|\boldsymbol{x})$. The SSR saliency map solves the problem

$$\underset{\boldsymbol{z}}{\text{maximize}}\, p_{\mathcal{M}}(c|\boldsymbol{z} \odot \boldsymbol{x} + (\mathbf{1} - \boldsymbol{z}) \odot \hat{\boldsymbol{x}}) \tag{1}$$

while the SDR saliency map solves the problem

$$\underset{\boldsymbol{z}}{\text{minimize}}\, p_{\mathcal{M}}(c|(\boldsymbol{z} \odot \hat{\boldsymbol{x}} + (\mathbf{1} - \boldsymbol{z}) \odot \boldsymbol{x}). \tag{2}$$

We will need to place appropriate priors on $\boldsymbol{z}$ to discourage trivial solutions.

3.2 Generative model

Solving SSR and SDR necessitate a reference $\hat{\boldsymbol{x}}$, which was previously chosen heuristically, e.g., by blurring $\boldsymbol{x}$. Because the heuristic induces a bias on the saliency map, we instead compute the saliency for each component by considering its imputation in a probabilistic framework (Robnik-Šikonja and Kononenko, 2008; Zintgraf et al., 2017). Here describe such a framework for an image application where the input comprises pixels organized into regions, but our method is more broadly applicable to any domain where the classifier is differentiable.

Consider an input image $\boldsymbol{x}$ with $\boldsymbol{x}_{\backslash r}$ denoting that input with pixels from the region r removed (unobserved). Then to compute the classifier output we must marginalize out the removed pixels $\boldsymbol{x}_r$ like

$$p_{\mathcal{M}}(c|\boldsymbol{x}_{\backslash r}) = \mathbb{E}_{\boldsymbol{x}_r \sim p(\boldsymbol{x}_r|\boldsymbol{x}_{\backslash r})}(p_{\mathcal{M}}(c|\boldsymbol{x}_{\backslash r}, \boldsymbol{x}_r)), \tag{3}$$

where we approximate $p(\boldsymbol{x}_r|\boldsymbol{x}_{\backslash r})$ by some generative model G with distribution $p_G(\boldsymbol{x}_r|\boldsymbol{x}_{\backslash r})$. Then given a binary mask $\boldsymbol{z}$ and the original image $\boldsymbol{x}$, we define a perturbation function ϕ as a convex mixture of the input and reference with binary weights:

$$\phi(\boldsymbol{x}, \boldsymbol{z}) = \boldsymbol{z} \odot \boldsymbol{x} + (\mathbf{1} - \boldsymbol{z}) \odot \hat{\boldsymbol{x}} \;\text{ where } \hat{\boldsymbol{x}} \sim p_G(\hat{\boldsymbol{x}}|\boldsymbol{x}_{\boldsymbol{z}=\mathbf{0}}). \tag{4}$$

3.3 Objective

Here we consider inference of $\boldsymbol{z}$ under the two objectives previously discussed. Following the SSR objectives, we aim to find a minimum region that supports our classification. Therefore, we encourage our mask distribution $q_\theta(\boldsymbol{z})$ to output mask that increases its classification score but also penalizes the size of the kept region. Specifically, given a minibatch with size M, a classification score function $s_{\mathcal{M}}(c|\boldsymbol{x})$, our objective function can be written as:

$$\arg\min_\theta L(\theta) = -\frac{1}{M}\sum_{i=1}^{M} s_{\mathcal{M}}(\boldsymbol{c_i}|\phi(\boldsymbol{x}_i, \boldsymbol{z_i})) + \frac{\lambda}{M}\sum_{i=1}^{M} \|\mathbf{1} - \boldsymbol{z}_i\|_1, \;\text{ where } \boldsymbol{z}_i \sim q_\theta(\boldsymbol{z}). \tag{5}$$

On the other hand, following the SDR objective, we encourage mask $\boldsymbol{z}$ to have low classification score with smallest removal region. This optimizes a very similar objective:

$$\arg\min_{\theta} L(\theta) = \frac{1}{M}\sum_{i=1}^{M} s_{\mathcal{M}}(\boldsymbol{c_i}|\phi(\boldsymbol{x}_i, \boldsymbol{z_i})) + \frac{\lambda}{M}\sum_{i=1}^{M}\|\boldsymbol{z}_i\|_1, \text{ where } \boldsymbol{z}_i \sim q_\theta(\boldsymbol{z}). \tag{6}$$

The classification score function s represents the classification confidence. In our experiment, we take s as a log-odds score of the class probability:

$$s_{\mathcal{M}}(c|\boldsymbol{x}) = \log p_{\mathcal{M}}(c|\boldsymbol{x}) - \log(1 - p_{\mathcal{M}}(c|\boldsymbol{x})) \tag{7}$$

Naively searching over all possible $\boldsymbol{z}$ is exponentially costly in the number of pixels U, which is clearly computationally intractable. Therefore we specify a variational distribution over masks $q_\theta(\boldsymbol{z})$ as a factorized Bernoulli

$$q_\theta(\boldsymbol{z}) = \prod_{u=1}^{U} q_{\theta_u}(z_u) = \prod_{u=1}^{U} \mathrm{Bern}(z_u|\theta_u). \tag{8}$$

This corresponds to applying Bernoulli dropout (Srivastava et al., 2014) to the input pixels and optimizing the per-pixel dropout rate. We thus call θ_u the Variational Dropout Saliency Map (VDSM), which can be visualized since it has same dimensionality as the input.

3.4 Optimization

To optimize the parameter $\boldsymbol{\theta}$ through the discrete random mask $\boldsymbol{z}$, we use a continuous relaxation (Maddison et al., 2016) to the distribution over masks (Gal et al., 2017), which permits gradient-based optimization by reparameterization[1]. We initialize all our dropout rate θ to be 0.5. We optimize using Adam (Kingma and Ba, 2014) with learning rate 0.02 and linearly decay the learning rate for 500 epochs in all our experiments. Our PyTorch and takes about one minute in a eight-core CPU machine to finish one single image.

3.5 Relation to BBMP

BBMP (Fong and Vedaldi, 2017) can be seen as a continuous relaxation of our method. Specifically, their mask is continuous in $[0, 1]$ while ours is discrete in $\{0, 1\}$. And they optimize a single mask initialized in the center, while our approach optimizes a mask distribution. BBMP has two drawbacks compared to our method. First, they need to have a static reference value such as blurred images, while our method can dynamically impute the reference value by generative model from the preserved pixels. Second, our method can utilize the mini-batch advantage to explore the huge space of masks, while BBMP can only do a local search of the current single mask.

4 Experiments

In the first experiment, we compare our saliency map against existing methods in the regime where they operate: simple heuristic generative models of the reference. In particular used the rudimentary generative model from Shrikumar et al. (2017) that imputes missing values using the background color (black in MNIST) in order to facilitate comparison with that method. In the second experiment we consider the use of a more sophisticated generative model, a convolutional variational autoencoder (VAE), in our method.

Our baselines are three recently proposed methods: Prediction Difference Analysis (PDA) (Zintgraf et al., 2017), DeepLift (Shrikumar et al., 2017), and BBMP (Fong and Vedaldi, 2017). PDA requires a window size parameter which we set to 1 since empirically we find increasing window size decreases the performance on MNIST. For both our method (VDSM) and BBMP method, we solve for saliency maps according to two different objectives: SSR and SDR. All methods are used to interpret a convolution network trained on MNIST with architecture specified by Shrikumar et al. (2017) with inputs normalized between 0 and 1.

[1] Admittedly, this relaxation biases the gradient estimator; we hypothesize that optimizing with low-variance unbiased gradient estimators (Tucker et al., 2017; Grathwohl et al., 2017) may improve our results.

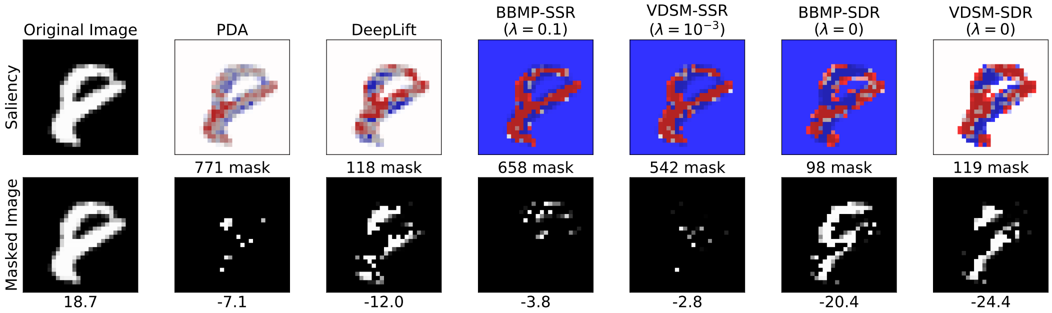


Figure 1: Evaluation of Experiment 1 using the Removal procedure. Best viewed in color. In the saliency maps (top row), red and blue signify salient and non-salient to the classification, respectively. The second row shows the image that has the lowest overall log-odds difference when pixels are successively masked and imputed in order from most to least salient. Here imputation is replacement with the background color (black). For each image, the number of masked-and-imputed pixels used the log-odds is specified in the bottom.

4.1 Evaluation procedures

We consider evaluation according to two procedure, which we call Removal and Preservation. Both procedures involve successively removing pixels and impute their values according to the generative model, either filling in background color or draw a sample from the VAE. We record the change in log-odds classification score at each step relative to the original input and and evaluate each method according to the maximum log-odds change. The two procedures differ only in the order of removed-and-imputed pixels: Removal goes from most to least salient pixels, while preservation goes from least to most. Intuitively, the SDR objective is well matched for the Removal procedure while the SSR objective is well matched for the Preservation procedure. For each evaluation procedure, we evaluate 100 images in MNIST and use the grid search to determine λ for both our method (VDSM) and the BBMP.

4.2 Background color imputation

Figure 1 shows an example of our experiment. In the saliency map (top row), red and blue means important and non-important to the classification respectively, while we set VBD and BBMP's saliency value 0.5 as white since in both methods the saliency ranges between 0 and 1. In the bottom row, we show the digit when it has minimum log-odds as digit 8 during the removal of pixels. In figure 2, we show the box plot of maximum log-odds decreases for 100 MNIST images. It shows that our method (VDSM) with SDR objective performs the best, with BBMP with SDR objective slightly worse. PDA method slightly wins DeepLift, and the BBMP with SSR and VDSM with SSR are the worst two methods.

We show an example in figure 3. We demonstrate VDSM and BBMP with SSR objective are better than other methods by removing the evidence of other classes. In figure 4, we show the box plot of the maximum log-odds increases for 100 MNIST images. We see BBMP and VDSM with SSR objective clearly win other methods. PDA and DeepLift are still in the middle, with VDSM with SDR objective slightly worse. Note that BBMP with SDR is mostly close to 0, which means it is not able to find any mode that is better than the original image.

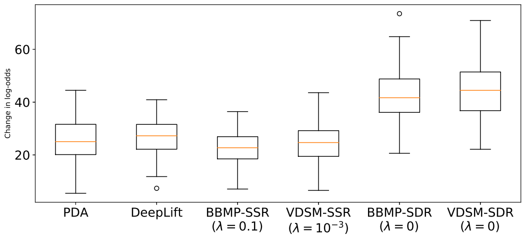


Figure 2: Box plot for minimum log-odds during masking-imputing of pixels in the Removal procedure for 100 images. The higher the better.

4.3 Imputation by VAE

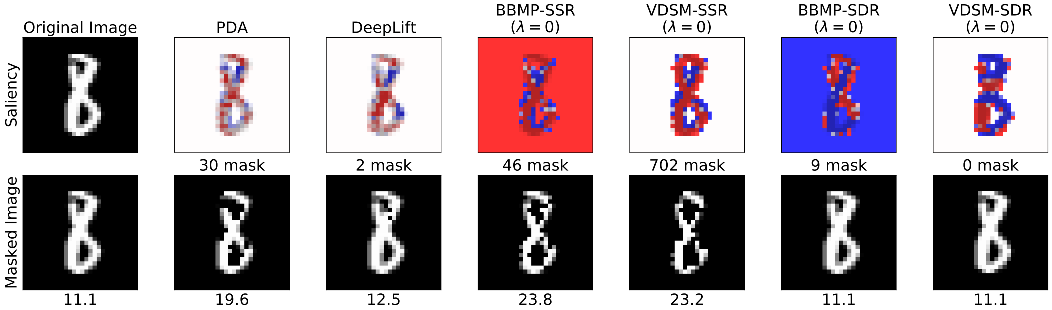


Figure 3: Evaluation of experiment 1 using the Preservation procedure. We use the same coloring scheme as Figure 1. Preservation differs from Removal in that it masks and imputes pixels in the opposite order.

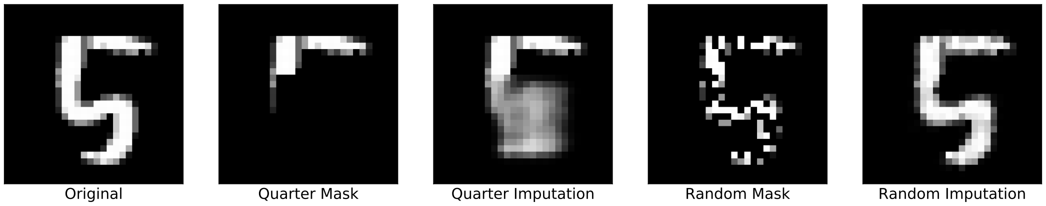


Figure 5: VAE for imputing missing (set to background color) pixels in MNIST images. The training data comprises two kinds of pixel removals: The Quarter mask training data removes a rectangular region with $\frac{1}{4}$ of the pixel area and random center position, while the Random mask training data randomly removes 50% of the pixels. We show example imputations for each type of training example.

VAE Training To understand the role of generative model, we train a simple 3-layer convolution VAE (Kingma and Welling, 2013) to impute the missing region in MNIST. We concatenate the mask vector as an additional channel to the input image, and construct binary cross entropy loss only on the masked region. During training, we randomly produce two kinds of masks: quarter mask and random mask. Quarter mask randomly removes a $\frac{1}{4}$ square region in the image, and random mask removes randomly 50% of the pixels. We use Adam (Kingma and Ba, 2014) optimizer to optimize 100 epochs on MNIST training set. We demonstrate examples of imputation in the figure 5.

Comparison of background-color and VAE imputation in Removal We compare the background-color and VAE imputation effect in the Removal procedure described in section 4.1. We compare with PDA, VDSM with SSR objective (VDSM-SSR) and VDSM with SDR objective (VDSM-SDR) since DeepLift and BBMP do not use a learned generative model. In figure 6, we visualize the differences between background-color imputation and the VAE imputation. In the saliency map, since VAE does not only consider the white stroke part, it highlights some adjacent black regions and highlights their importance. For example, VDSM-SDR-VAE considers the black region inside the lower circle of 8 as important, since removing it makes it look like a 7. Regrettably, the saliency maps from VSDM with VAE aren't apparently more interpretable than the competing methods. However, sampling from the imputation generative model give us a sense for how VDSM-VAE fools the classifier, for example by highlighting pixels that, when missing, cause the VAE to generate a image that resembles a 7.

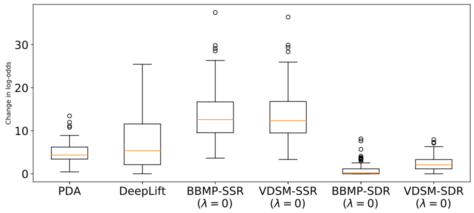


Figure 4: Box plot for maximum log-odds during removal of pixels in Preservation procedure for 100 images. The higher the better.

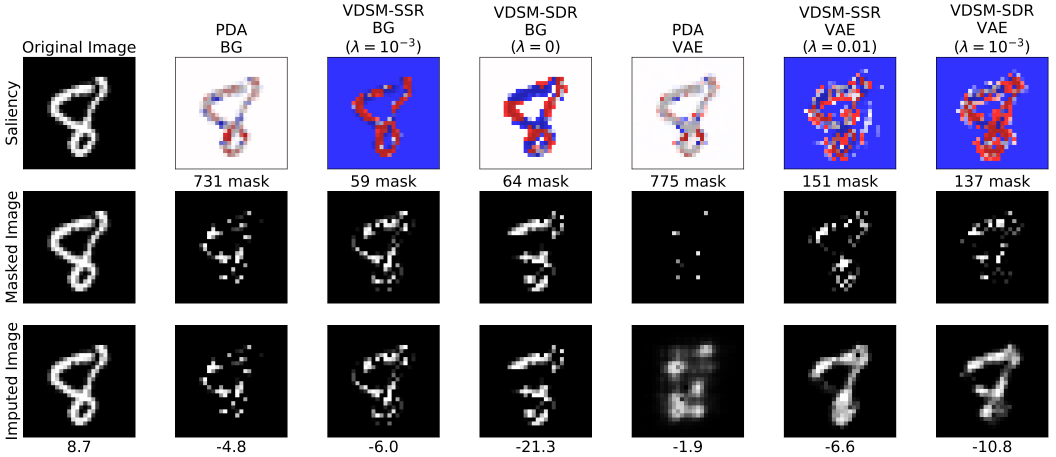


Figure 6: Comparison of imputation methods (Background vs. VAE) in the Removal evaluation procedure. We show saliency map, masked image and imputation image for first, second and third row. The "BG" or "VAE" in the column name indicates the generative model used for imputation.

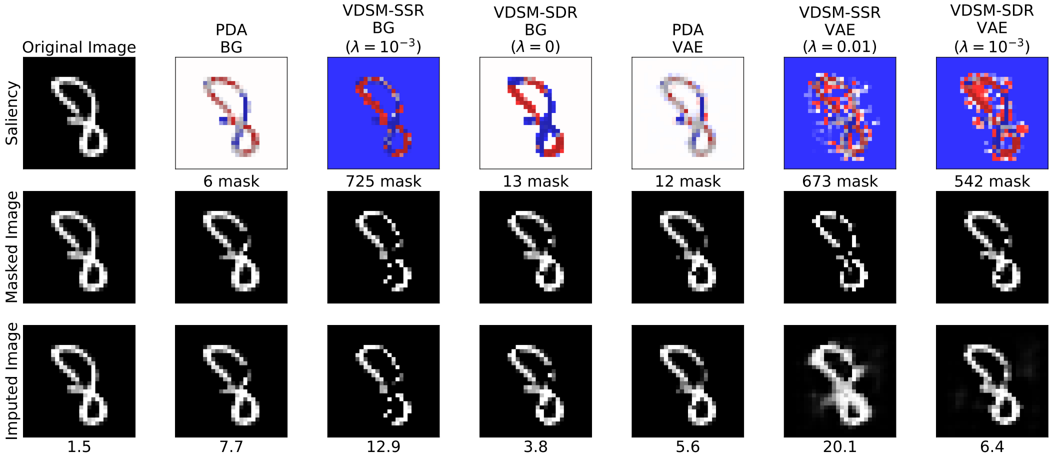


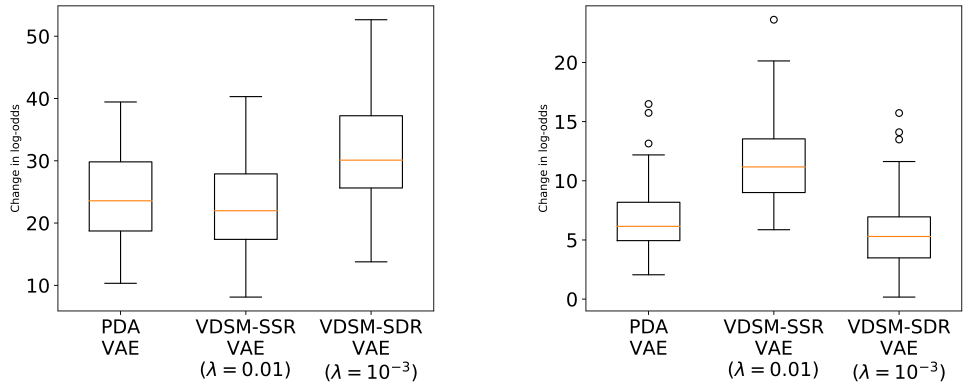
Figure 7: Comparison of imputation methods in the Preservation evaluation procedure. We follow the same formatting as Figure 6.

Comparison of background-color and VAE imputation in Preservation In figure 7, we visualize the background-color and VAE imputation effect in the Preservation procedure mentioned in section 4.1. The VAE provides more reasonable imputations than the background-color method. In VDSM-SSR with VAE, VAE closes the loop of 8 and has higher log-odds by removing pixels that might resemble other classes. On the other hand, VDSM-SSR with background-color imputation seems to find an unrealistic digit that increases the log-odds of 8, indicating that the lack of a generative model yields saliency maps that suffer from artifacts.

In figure 8, we demonstrate box plot comparison with both procedures. We show that our method VDSM-SDR still performs the best in Removal, while and VDSM-SSR still performs the best in Preservation.

5 Conclusion and future work

We extended previously proposed saliency maps for convolutional network image classification by modeling the mask as Bernoulli distributed and solving with variational inference. We call the resulting Bernoulli parameters per input the Variational Dropout Saliency Map. Previous work



(a)

(b)

Figure 8: Box plot for VAE imputation in the Removal and Preservation evaluation procedures. (a) and (b) are Removal and Preservation, respectively.

considered two objectives for optimizing the saliency map: smallest supporting region and smallest destroying region; we discuss the merits of each objective with qualitative examples from MNIST. We also quantitatively compare our method against previous work on both metrics and conclude that our method is competitive. We make explicit the role of imputation by a generative model and compare background-color imputation and VAE imputation. We also plan to extend to larger dataset such as ImageNet.

References

Dabkowski, P. and Y. Gal (2017). Real time image saliency for black box classifiers. *arXiv preprint arXiv:1705.07857*.

Fong, R. and A. Vedaldi (2017). Interpretable explanations of black boxes by meaningful perturbation. In *Proceedings of the International Conference on Computer Vision (ICCV)*.

Gal, Y. and Z. Ghahramani (2016). Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pp. 1050–1059.

Gal, Y., J. Hron, and A. Kendall (2017). Concrete dropout. *arXiv preprint arXiv:1705.07832*.

Grathwohl, W., D. Choi, Y. Wu, G. Roeder, and D. Duvenaud (2017). Backpropagation through the void: Optimizing control variates for black-box gradient estimation. *arXiv preprint arXiv:1711.00123*.

Kingma, D. and J. Ba (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kingma, D. P. and M. Welling (2013). Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

Maddison, C. J., A. Mnih, and Y. W. Teh (2016). The concrete distribution: A continuous relaxation of discrete random variables. *arXiv preprint arXiv:1611.00712*.

Robnik-Šikonja, M. and I. Kononenko (2008, May). Explaining Classifications For Individual Instances. *IEEE Transactions on Knowledge and Data Engineering*.

Russakovsky, O., J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei (2015). ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV) 115*(3), 211–252.

Selvaraju, R. R., A. Das, R. Vedantam, M. Cogswell, D. Parikh, and D. Batra (2016). Grad-cam: Why did you say that? *arXiv preprint arXiv:1611.07450*.

Shrikumar, A., P. Greenside, and A. Kundaje (2017). Learning important features through propagating activation differences. *arXiv preprint arXiv:1704.02685*.

Simonyan, K., A. Vedaldi, and A. Zisserman (2013). Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*.

Srivastava, N., G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov (2014). Dropout: a simple way to prevent neural networks from overfitting. *Journal of machine learning research 15*(1), 1929–1958.

Tucker, G., A. Mnih, C. J. Maddison, J. Lawson, and J. Sohl-Dickstein (2017). Rebar: Low-variance, unbiased gradient estimates for discrete latent variable models. In *Advances in Neural Information Processing Systems*, pp. 2624–2633.

Zhou, B., A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba (2014). Object detectors emerge in deep scene cnns. *arXiv preprint arXiv:1412.6856*.

Zintgraf, L. M., T. S. Cohen, T. Adel, and M. Welling (2017). Visualizing deep neural network decisions: Prediction difference analysis. *arXiv preprint arXiv:1702.04595*.