Chapter 5

Enhanced Resolution & Sanity preserving saliency maps

ABSTRACT

Backpropagation image saliency aims at explaining model predictions by estimating model-centric importance of individual pixels in the input. However, class-insensitivity of the earlier layers in a network only allows saliency computation with low resolution activation maps of the deeper layers, resulting in compromised image saliency. Remedying this can lead to sanity failures. We propose CAMERAS\(^1\), a technique to compute high-fidelity backpropagation saliency maps without requiring any external priors and preserving the map sanity. Our method systematically performs multi-scale accumulation and fusion of the activation maps and backpropagated gradients to compute precise saliency maps. From accurate image saliency to articulation of relative importance of input features for different models, and precise discrimination between model perception of visually similar objects, our high-resolution mapping offers multiple novel insights into the black-box deep visual models, which are presented in the paper. We also demonstrate the utility of our saliency maps in adversarial setup by drastically reducing the norm of attack signals by focusing them on the precise regions identified by our maps. Our method also inspires new evaluation metrics and a sanity check for this developing research direction.

5.1 Introduction

Deep visual models are fast surpassing human-level performance for various vision tasks, including image classification \([3],[4]\), object detection \([23],[24]\), and semantic segmentation \([25],[26]\). However, they hardly offer any explanation of their decisions, and are rightfully considered black-boxes. This is problematic for their practical deployment, especially in high-risk emerging applications where transparency is vital, e.g. in healthcare, self-driving vehicles and smart surveillance \([15]\). The problem is exacerbated by the push of ‘right to explanation’ by algorithmic regulatory authorities and their objection to black-box models in safety-critical applications \([11]\).

Addressing this issue for deep visual models, techniques are emerging to offer input-agnostic \([95]\) and input-specific \([16],[15],[17]\) explanation of model predictions. This work subscribes to the latter, where the

---

\(^1\)Code is available at https://github.com/AsimJalwana/CAMERAS.git
CHAPTER 5. ENHANCED RESOLUTION & SANITY PRESERVING SALIENCY MAPS

Figure 5.1: (Top) CAMERAS meticulously fuses activation maps and backpropagated gradients of a layer for multi-scale copies of an input. After passing the resulting saliency map through ReLU and normalising it \( f \), the map is embedded on the original image. (Bottom) By avoiding influence of external factors, CAMERAS easily passes the sanity checks for image saliency. Shown are the results of cascading randomization [101] on ResNet. Progressive randomising of layer weights randomises the output right from the logits layer which identifies preservation of sanity. Thus, the CAMERAS maps do not sacrifice their sanity for high-resolution, achieving the best of both worlds.

The ultimate goal is to identify the contribution of each pixel in an input to the output prediction. The popular techniques to achieve this adopt one of two strategies. The first, systematically modifies the input image pixels (i.e. image regions) and analyzes the effects of those perturbations on the output predictions [16], [83], [96], [97]. The underlying search nature of this perturbation-based formulation offers high-fidelity model-centric importance attribution to the input pixels, albeit at a high computational cost. Hence, tractability is achieved under heuristics or external priors over the computed importance maps. This is undesired because the eventual maps may be influenced by these external factors, which compromises the model-fidelity of the maps.

The second strategy relies on the activation maps of the internal layers and gradients of the models. Commonly known as backpropagation saliency methods [17], [15], [98], [99], [97], [100] approaches adopting this strategy are computationally efficient, thereby offering the possibility of avoiding unnecessary heuristics or priors. However, for the visual neural models, the layers closer to the input are class-insensitive [15]. This limits the ammunition of backpropagation saliency methods to the deeper layers of the networks, where the size of activation maps is very small, e.g. \( 10^{-3} \times \) of the input size. Projecting the saliency computed with those maps onto the original image grid results in intrinsically low-resolution image saliency. On the other hand, using heuristics or priors to sharpen those projections inadvertently compromise the sanity of the maps [101]. Not to mention, employing activation signals of multiple internal layers for resolution enhancement takes us back to a combinatorial search problem of choosing the best layers, under a pre-specified heuristic.

Addressing the above issues, we introduce CAMERAS - an Enhanced Resolution And Sanity preserving...
Class Activation Mapping for backpropagation image saliency. The proposed technique (Fig. 5.1-top) systematically accumulates and fuses multi-scale activation maps and backpropagated gradients of a model to construct precise saliency maps. By avoiding the influence of any external factor, e.g. heuristics, priors, thresholds, the saliency maps of CAMERAS easily pass the sanity checks for image saliency (Fig. 5.1-bottom). Moreover, the technique allows saliency estimation with a single network layer, not requiring any layer search for map resolution enhancement. Contributions of the paper are summarised below:

- We propose CAMERAS for precise backpropagation image saliency while preserving the sanity. Our method outperforms the state-of-the-art saliency methods by a large margin, achieving up to $27.5\%$ error reduction for the popular pointing game metric [100].
- Exploring the newly found precision saliency mapping with CAMERAS, we visualise differences in the semantic understanding of different architectures that govern their performance. We also highlight model-centric discrimination of input features for visually similar objects in never-before-seen details.
- Considering the equivalent treatment of deep models as differentiable programs by the fast-developing parallel field of adversarial learning, we enhance the widely considered strongest adversarial attack PGD [22] with our saliency technique - drastically improving the efficacy of the attack.
- The ability of precise saliency computation allowed by CAMERAS calls for new quantitative metrics and sanity checks. We contribute two new evaluation metrics and a sanity check to advance this research direction.

5.2 Related work

The literature has seen multiple techniques that perturb input pixels and measure its effects on the model outputs to identify salient regions in input images [83], [16], [96]. Perturbing every possible combination of pixels has an exponential complexity. Therefore, such techniques often rely on fixed subsets of pixel combinations for tractability. Moreover, high non-linearity of deep models further restricts the saliency map to be reliable under fixed devised perturbation subsets. RISE [96] and Occlusion [97] generate attribution maps by weighing perturbation masks corresponding to the changes in the output scores. Other techniques, such as Meaningful perturbations [83], Extremal perturbations [16], Real-time saliency [102] and [103] cast the problem into an optimization objective. Though effective, these methods face a common issue of allowing a channel for external influence on the resulting maps in the form of e.g. heuristics, external constraints, priors or threshold etc.

Backpropagation saliency methods [98], [17], [97] aim at extracting information from within the model to identify model-centric salient regions in an input image. Relying on layer activations and model gradients, these methods are also known to be computationally more effective [104], [105]. Simoyan et al. [98] first used model gradients as a possible explanation of output predictions. Different adaptations have since been proposed to mitigate the inherent noise sensitivity of model gradients. Guided back prop [99] and DeConvNet [97] alter the backpropagation rules of model ReLU layers, while SmoothGrad [106] computes the average gradients over samples in the close vicinity of the original one. Similarly, DeepLIFT [107], LRP [108] and Excitation Backprop [100] recast the backpropagation rules such that the sum of attribution signal becomes unity. Sundararajan et al. [109] interpolated multiple attribution maps to reduce the signal noise. There are also instances of exploring saliency computation using various layers of deep models by merging the layer activation maps with the gradient information. Such methods include CAM [110], its generalized adaption GradCAM [17], linear approximation...
and NormGrad [15]. For such methods, it is found that the layers closer to output generate better saliency maps because those layers are more sensitive to the high level class features.

Beyond the visual quality of saliency maps, a few works have also critically explored the reliability of these maps for different methods [112], [113]. Adebayo et al. [101] first introduced sanity checks for image saliency methods, highlighting that visual appeal of the maps alone can be misleading. They evaluated sensitivity of the results of popular techniques to model parameters. Surprisingly, ‘model-centric saliency maps’ computed by multiple methods were found insensitive to the model - failing the sanity check. The techniques avoiding external influences on the map, e.g. Grad-CAM [17] easily passed the test. This finding also resonated with other subsequent sanity checks [15].

Evaluation of image saliency methods is a challenging problem because deep model representation is not always aligned with the human visual system [114]. Hence, indirect evaluation of image saliency is often done by analysing its weak localization performance. For instance, Pointing game score [100] is a commonly used metric for quantitative evaluation of image saliency results [16], [15]. It measures the correlation between the maximal point in a saliency map with the semantic labels of the pixel. Its later adaptation [83] measures the overlap in the bounding boxes from saliency maps and the ground truth. Petsiuk et al. [96] proposed insertion-deletion metrics to measure the impact of perturbations over image patches in the order of importance to quantify saliency accuracy. Nevertheless, these metrics are meant to be weak indicators of saliency maps due to the imprecise nature of the maps computed by the earlier methods, which is no longer the case for CAMERAS.

5.3 Proposed Approach

Before discussing the details of our technique, we first provide a closer look at the broader paradigm of backpropagation saliency computation. We use Grad-CAM [17] - a popular technique - as a test case to motivate the proposed method. The text below highlights only the relevant aspects of the test case for intuition.

5.3.1 Saliency computation with backpropagation

Let \( I \in \mathbb{R}^{c \times h \times w} \) be an input image with ‘c’ channels. A deep visual classifier \( K(I) \) maps \( I \) to a prediction vector \( y^\ell \in \mathbb{R}^L \), where ‘L’ is the total number of classes. Here, ‘\( \ell \)’ indicates the predicted label of \( I \) under the premise that the \( \ell \)th coefficient of \( y \) has the largest value. It is well-known that a neural network is a hierarchical composition of representation layers. Rebuffi et al. [15] demonstrated that among these layers, those closer to the input learn class-insensitive features. Thus, the deeper layers hold more promise for computing image saliency for a model. Grad-CAM [17] takes a pragmatic approach to single out the last convolutional layer to estimate the saliency map.

Let us denote the \( k \)th activation map of the last convolutional layer of a network as \( A_k(I) \in \mathbb{R}^{m \times n} \). Focusing on Grad-CAM, the technique first computes an intermediate representation \( S(I) \in \mathbb{R}^{m \times n} \), such that \( S^{(i,j)}(I) = \sum_k w_k A_k^{(i,j)}(I) \), where \( w_k \) is given by Eq. (5.1). Henceforth, we ignore the argument \( I \) for clarity, unless required.

\[
w_k = \frac{1}{(m + n)} \sum_{i=1}^{m} \sum_{j=1}^{n} \left( \frac{\partial y^\ell}{\partial A_k^{(i,j)}} \right).
\] (5.1)

In the above expressions, \( X^{(i,j)} \) is the \((i, j)\) coefficient of \( X \). The computed \( S \) is later extended to the final
saliency map $\Psi \in \mathbb{R}^{h \times w}$, as $f(S) : S \rightarrow \Psi$, where the function $f(.)$ must account for interpolating an $m \times n$ matrix for a $h \times w$ grid (along other complementary transformations).

5.3.2 Sub-optimality of backpropagation methods

Observing Grad-CAM (and similar methods) from the above perspective reveals two performance drain-holes in backpropagation image saliency computations. 

(a) Over-simplification of the weights $w_k$: Since $A_k$ is an activation map, individual coefficients of this matrix should have different importance for the final prediction. Indeed, this is also reflected in the values of individual backpropagated gradients for these coefficients - computed with the expression in the parenthesis in Eq. (5.1). Since the ultimate objective of image saliency is to compute importance of individual pixels, losing information with over-simplification of $w_k$ is not conducive. Grad-CAM takes an extreme approach of representing $w_k$ with a scalar value. The main reason for that is, it is actually detrimental to plainly replace $w_k$ with an encoding $W_k \in \mathbb{R}^{m \times n}$ such that $S = \sum_k W_k \circ A_k$. Here, $\circ$ is the point-wise product and $W_k$ encodes individual backpropagated gradients. Gradients are extremely sensitive to signal variations. Hence, even a small activation change can result in (a misleading) exaggerated weight alteration for the activation map, resulting in incorrect image saliency. Grad-CAM is able to mitigate this problem by averaging the gradients in Eq. (5.1). However, this remedy comes at the cost of losing the fine-grained information about the gradients.

Though centered around Grad-CAM, the above discussion points to a simple, yet powerful generic notion for effective backpropagation saliency computation. That is, to better leverage the backpropagated gradients, the differential information of the gradients (in $W_k$) is still at our disposal to exploit. Fusing activation maps with this information promises more precise image saliency.

(b) Large interpolated segments: Typically, the activation maps of the deeper convolutional layer in visual classifiers are (spatially) much smaller than the input images. For instance, in ResNet [58], the $7 \times 7$ maps of the last convolutional layer are 1024 times smaller than the $224 \times 224$ inputs. Thus, for $m \ll h$ and $n \ll w$, a saliency map $\Psi$ computed with the class sensitive deeper layers must be mainly composed of interpolated segments. To contextualize, in the above ResNet example, 99.9% of the values are generated by the function $f(S) : S \rightarrow \Psi$ in Grad-CAM. This automatically renders $\Psi$ a low-resolution map, leaving alone the issue of correctness of the importance assigned to the individual pixels in the eventual saliency map. The low resolution of saliency maps has also spawned methods to improve $f(.)$ [101], [17]. However, those techniques inevitably rely on external information (including heuristics) for the transform $S \rightarrow \Psi$ due to unavailability of further useful information from the model itself. This leads to sanity check failures because the operands are no longer purely grounded in the original model.

5.3.3 The room for improvement

From the above discussion, it is clear that whereas useful techniques exist for backpropagation image saliency, the paradigm is yet to fully harness the backpropagated gradients and resolution enhancement of the activation maps for precise image saliency. Both limitations are rooted in the very nature of the underlying signals. Leveraging these signals from multiple layers can potentially help in partially overcoming the issues. However, this possibility is also restricted by the class-insensitivity of the earlier network layers and combinatorial nature of the problem. Moreover, there is evidence that multi-layer fusion can often adversely affect image saliency [15]. Hence, a technique specifically targeting the class-sensitive last layer, while allowing minimal loss
of differential information across backpropagated gradients and improving activation map upsampling, holds significant promises for better saliency computation.

5.3.4 CAMERAS

Building directly on the insights in § 5.3.3, we devise CAMERAS - an enhanced resolution and sanity preserving class activation mapping scheme. The approach is illustrated in Fig. (5.1-top) and explained below in a top-down manner, keeping the flow of the above discussion.

Our method eventually computes a saliency map as:

$$\Psi = f\left(\text{ReLU}\left(\sum_k W^*_k \odot A^*_k\right)\right), \forall k,$$  \hspace{1cm} (5.2)

where $W^*_k \in \mathbb{R}^{h \times w}$ encodes the differential information of the backpropagated gradients for the $k$th activation map in a network layer, $A^*_k \in \mathbb{R}^{h \times w}$ is an enhanced resolution encoding for the activation map itself, and $f(.)$ performs an element-wise normalisation in the range $[0, 1]$. The $W^*_k$ and $A^*_k$ are defined as follows

$$W^*_k = \mathbb{E}_t[\varphi_t(W_k(\varphi_t(I, \zeta_t)), \zeta_o)],$$

$$A^*_k = \mathbb{E}_t[\varphi_t(A_k(\varphi_t(I, \zeta_t)), \zeta_o)],$$  \hspace{1cm} (5.3)

where $\varphi_t(X, \zeta_t)$ is the $t$th up-sampling applied to resize $X$ to the dimensions $\zeta_t$ - provided as a tuple. We fix $\zeta_o = (h, w)$ for $I \in \mathbb{R}^{c \times h \times w}$. This will be explained shortly. We compute the $(i, j)$ coefficient of $W_k$ as $\frac{\partial y^\ell}{\partial A^{(\kappa, \ell)}_{k(i,j)}}$. The overall process of generating an image saliency map with CAMERAS is summarized as Algorithm 6.

The algorithm computes the desired saliency map $\Psi$ by an iterative multi-scale accumulation of activation maps and gradients for the $\kappa$th layer of the model. In the $t$th iteration, the input image $I$ gets up-sampled to $\zeta_t$ based on the maximum desired size $\zeta_m$ and the number of steps $N$ allowed to reach that size (lines 4,5). Provided that the input up-scaling does not alter the model prediction, the activation maps and backpropagated gradients to the $\kappa$th layer are also up-sampled and stored. We show this on lines 6-10 of the algorithm. Notice that, we use calligraphic symbols to distinguish 3D tensors from matrices (e.g. $A$ instead of $A$ for activations) in the algorithm for clarity. The newly introduced symbol $\nabla J(\kappa, \ell)$ on line 9 denotes the collective backpropagated gradients to the $\kappa$th layer w.r.t. the predicted label $\ell$. Also notice, on lines 8, 9, up-sampling of the activation maps and gradients and performed to match the original image size $\zeta_o$. This is because, the same accumulated signals are eventually transformed into the saliency map of the original image. We iteratively accumulate the up-sampled activation maps and gradients, and finally compute their averages on lines 12 and 13. On line 14, we compute the saliency map by solving Eq. (5.2). Here, matrix notation is intentionally used to match the original equation.

In Algorithm 6, CAMERAS is shown to expect four input parameters, along the classifier and the image. We discuss the choice of $\varphi(.)$ in § 5.3.4 where we eventually propose to keep this function fixed. The algorithm optionally allows $\kappa$ for computing saliency maps using layers other than the last convolutional layer of the model. For all the experiments presented in the main paper, we keep $\kappa$ fixed to the last convolutional layer. This is due
5.3. PROPOSED APPROACH

Algorithm 6 CAMERAS algorithm

\textbf{Input:} Classifier $\mathcal{K}$, image $I \in \mathbb{R}^{c \times h \times w}$, maximum size $\zeta_m$, steps $N$, interpolation function $\varphi(\cdot)$, layer $\kappa$

\textbf{Output:} Image saliency map $\Psi \in \mathbb{R}^{h \times w}$

1: Initialize $\mathcal{A}_0$, $\mathcal{W}_0$ to 0 tensors, $\zeta_0 = (h, w)$ and $t = t_m = 0$, $\ell = \mathcal{K}(I)$
2: while $t \leq N$ do
3: \hspace{1em} $t \leftarrow t + 1$
4: \hspace{1em} $\zeta_t \leftarrow \zeta_{t-1} + \lfloor \frac{\zeta_m}{N} \rfloor (t - 1)$
5: \hspace{1em} $I_t \leftarrow \varphi_t(I, \zeta_t)$
6: \hspace{1em} if $\mathcal{K}(I_t) \rightarrow \ell$ then
7: \hspace{2em} $t_m \leftarrow t_m + 1$
8: \hspace{2em} $\mathcal{A}_t \leftarrow \mathcal{A}_{t-1} + \varphi_t(A(I_t, \kappa), \zeta_o)$
9: \hspace{2em} $\mathcal{W}_t \leftarrow \mathcal{W}_{t-1} + \varphi_t(\nabla J(\kappa, \ell), \zeta_o)$
10: \hspace{1em} end if
11: end while
12: $\mathcal{A} \leftarrow \mathcal{A}_t/t_m$
13: $\mathcal{W} \leftarrow \mathcal{W}_t/t_m$
14: $\Psi = f \left( \text{ReLU} (\sum_k W_k \odot A_k^*) \right), \forall k$
15: return

...to the well-known class-sensitivity of the deeper layers of CNNs [15]. Essentially, the only choice to be made is for the values of parameters $I_m$ and $N$, which are related as $\zeta_m = cN\zeta_o$, where $c$ is the step size. Trading-off performance with efficiency, the choice of these parameters is mainly governed by the available computational resources. For larger $\zeta_m$ and $N$ values, performance of CAMERAS roughly improves monotonically - generally saturating at $\zeta_m \approx (1K, 1K)$ for $\zeta_o = (224, 224)$ for the popular ImageNet models. For this $\zeta_m$ range, the performance is largely insensitive for $N \in [5, 10]$. We give further analysis of parameter values in the supplementary material. In the presented experiments, we empirically choose $\zeta_m = (1K, 1K)$ and $N = 7$.

Sanity preservation and strength

In CAMERAS, we do not impose any prior over the saliency map, nor we use any heuristic to guide its computation process. The technique also preserves model fidelity by requiring no structural (or any other) alteration to the original model. It mainly relies on primitive arithmetic operations over the model signals. These attributes also characterize those other methods that pass the popular sanity checks for backpropagation saliency [101], [15], albeit resulting in low-resolution saliency maps. In CAMERAS, the only source of any ‘potential’ external influence on the resulting map is through the interpolation function $\varphi(\cdot)$. We conjecture that as long as $\varphi(\cdot)$ is a first-order function defined over the signals originating in the model itself, CAMERAS maps will always preserve their sanity because the maps would fully originate in the model. To preclude any unintentional prior over the maps, our formulation dictates the use of simpler functions as $\varphi(\cdot)$. Hence, we choose to fix bi-linear interpolation as $\varphi(\cdot)$.

To analyse the reasons of extraordinary performance of CAMERAS (see § 5.4), we provide a brief theoretical perspective on the accumulation of multi-scale interpolated signals exploited by our method, using the results below.

\textbf{Lemma 5.1:} For $\mathbf{X}^* \in \mathbb{R}^{h \times w}$ and its interpolated approximation $\tilde{\mathbf{X}} = \varphi(\mathbf{X})$ s.t. $\mathbf{X} \in \mathbb{R}^{m \times n}$ and $\mathbf{X} \neq \tilde{\mathbf{X}}$, $||\mathbf{X} - \tilde{\mathbf{X}}|| = f ((h - m), (w - n))$ for $m < h$ and $n < w$, where $\varphi(\cdot)$ denotes bi-linear interpolation and $f(\cdot)$.
is a monotonic function over its arguments.

Lemma 5.2: For \( \tilde{X}_z = \varphi(X_z) \), where \( X_z \in \mathbb{R}^{m \times n} \) and \( \tilde{X}_p = \varphi(X_p) \), where \( X_p \in \mathbb{R}^{p \times q} \) s.t. \( p < m \) and \( n < q \), \( ||\hat{X} - \tilde{X}_z|| - ||\hat{X} - \tilde{X}_p|| \leq 0 \).

Corollary: \( E[||\hat{X} - \tilde{X}_z||] \leq ||\hat{X} - \tilde{X}_p||, \forall \tilde{X}_z \cup \tilde{X}_p \).

The lemma 5.1 states that bi-linear interpolation tends to be more accurate when the difference in the dimensions of the source signal and the target grid is smaller. The lemma 5.2 can also be easily verified as \( \tilde{X}_z \) is at least as accurate a projection of \( \hat{X} \) as \( \tilde{X}_p \), according to lemma 5.1. This necessarily makes its error equal to or smaller than the error of \( \tilde{X}_p \). In the light of lemma 5.2, the corollary affirms that the expected error of a set of interpolated signals is upper-bounded by the error of the least accurate signal in the set.

The above analysis highlights an important aspect. The CAMERAS results will necessarily be as accurate as operating our algorithm on the original input size, and will improve monotonically thereof with the up-sampled inputs. This is because the activations and gradients of the up-sampled inputs would map more accurately on the original image grid, as per the above results. This is significant because it allows CAMERAS to use the differential information in the backpropagated gradient maps while accounting for the noise-sensitivity of the gradients by averaging out these signals across multiple scales. This is the key strength of the proposed technique.
5.4 Evaluation

We perform a thorough qualitative and quantitative evaluation of CAMERAS on large-scale models and compare the performance with the state-of-the-art methods.

5.4.1 Qualitative results

In Fig. 5.2, we qualitatively compare the results of our technique with the existing saliency methods for ImageNet models of ResNet, DenseNet and Inception. Representative maps of randomly chosen images are provided. See the supplementary material for more visualisations. The high quality of CAMERAS maps is apparent in the figure. A quick inspection reveals that our technique maintains its performance across a variety of scenarios, including clear objects (Loudspeaker, Mountain tent), occluded objects (Bulbul), and multiples instances of objects (Accordion, Basket). Observing carefully, CAMERAS maps provide precise maps even for small and relatively complex geometric shapes, e.g. Volleyball, Spotlight, Loafer, Hognose snake. Interestingly, our method is able to attach appropriate importance even to the reflection of the Hognose snake. Adoption of saliency maps to complex geometric shapes is a direct consequence of enabling precise saliency mapping while sealing-off any external influence on the maps. CAMERAS is able to maintain its characteristic precision across different models and images. These are highly promising results for explainability of modern deep visual classifiers.

5.4.2 Quantitative results

A quantum leap in performance with CAMERAS is also observed in our quantitative results. Saliency maps are typically evaluated by measuring their correlation with the semantic annotations of the image. Pointing game [100] is a popular metric for that purpose, which considers the computed saliency for every object class in the image. If the maximal point in the saliency map is contained within the object, it is considered a hit; otherwise, a miss. The performance is measured as the percentage of successful hits. We refer to [100] for more details on the metric. Table 5.1 benchmarks the performance of CAMERAS for pointing game on 4,952 images of PASCAL VOC test set [115], and ~ 50K images of COCO 2014 validation set [116]. Our technique consistently shows superior performance, achieving up to 27.5% error reduction. The gain is higher for ResNet as compared to VGG due the better performance of the original ResNet that permits better saliency.

The pointing game generally disregards precision of the saliency maps by focusing only on the maximal points. Arguably, crudeness of the saliency maps computed by the earlier methods influenced this evaluation metric. The possibility of precise saliency computation (by CAMERAS) calls for new metrics that account for the finer details of saliency maps. We propose ‘positive map density’ and ‘negative map density’ as two suitable metrics, respectively defined as:

\[
\rho_{\text{map}}^+ = P(K(I \odot \Psi))/\sum_{i} \sum_{j} \Psi(i,j) \times (h \times w),
\]

and

\[
\rho_{\text{map}}^- = P(K(I \odot 1 - \Psi))/\sum_{i} \sum_{j} 1 - \Psi(i,j) \times (h \times w).
\]

Here, \(P(\cdot)\) is the predicted probability of the actual label of the object. Other notations follow the conventions from above. For an estimated saliency map, the value of \(\rho_{\text{map}}^+\) improves if higher importance is attached to a smaller number of pixels that retain higher confidence of the model on the original label. In the extreme case of all the pixels deemed maximally important (saliency value 1), the score depicts the model’s confidence on the object label. On the other hand, the value of \(\rho_{\text{map}}^-\) decreases if lesser importance is attached by the saliency method to more pixels that do not influence the prediction confidence on the original label. Lower value of this metric is more desired.
Table 5.1: Mean accuracy on pointing game over the full data (All) splits and subset of difficult images (Diff), as specified in [100]. The results of other schemes are generated with TorchRay package [16], and ‘*’ denotes an average over 3 runs for improved performance.

<table>
<thead>
<tr>
<th>Method</th>
<th>VOC07 Test (All/Diff)</th>
<th>COCO14 Val (All/Diff)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>VGG16</td>
<td>ResNet50</td>
</tr>
<tr>
<td>Center [16]</td>
<td>69.6/42.4</td>
<td>69.6/42.4</td>
</tr>
<tr>
<td>Gradient [98]</td>
<td>76.3/56.9</td>
<td>72.3/56.8</td>
</tr>
<tr>
<td>DeConv. [97]</td>
<td>67.5/44.2</td>
<td>68.6/44.7</td>
</tr>
<tr>
<td>Guid [99]</td>
<td>75.9/53.0</td>
<td>77.2/59.4</td>
</tr>
<tr>
<td>MWP [100]</td>
<td>77.1/56.6</td>
<td>84.4/70.8</td>
</tr>
<tr>
<td>cMWP [100]</td>
<td>79.9/66.5</td>
<td>90.7/82.1</td>
</tr>
<tr>
<td>RISE* [96]</td>
<td>86.9/75.1</td>
<td>86.4/78.8</td>
</tr>
<tr>
<td>GradCAM [17]</td>
<td>86.6/74.0</td>
<td>90.4/82.3</td>
</tr>
<tr>
<td>Extremal* [16]</td>
<td><strong>88.0/76.1</strong></td>
<td>88.9/78.7</td>
</tr>
<tr>
<td>NormGrad [15]</td>
<td>81.9/64.8</td>
<td>84.6/72.2</td>
</tr>
<tr>
<td>CAMERAS</td>
<td><strong>86.2/76.2</strong></td>
<td><strong>94.2/88.8</strong></td>
</tr>
</tbody>
</table>

Figure 5.3: Proposed CAMERAS allows verification of label attribution even more precisely than the input perturbation-based methods. Meaningful [83] and Extremal [16] perturbation methods require image-specific fine tuning of parameters. The latter is confined to 8% and 9% of the image area to achieve the results.

Combined $\rho_{map}^+$ and $\rho_{map}^-$ provide a comprehensive quantification of the quality of the saliency map. We provide results of our technique, Grad-CAM [17] and the recent Norm-Grad [15] on our metrics in Table 5.3. Due to page limits, we provide further discussion on the proposed metrics in the supplementary material.

5.5 CAMERAS for analysis

The precise CAMERAS results allow model analysis with backpropagation saliency in unprecedented details. Below, we present a few interesting examples.

The label attribution problem: It is known that deep visual classifiers sometimes learn incorrect association of labels with the objects in input images [83], [16]. For instance, for the image of Chocolate Sauce in Fig. 5.3, Inception is found to associate the said label to spoon instead of the sauce in the cup [83], [16]. This revelation was possible only through the input perturbation-based methods for attribution due to their precise nature, however, only after fine-tuning a list of parameters for the specific images. CAMERAS is the first backpropagation saliency method to achieve this result, without requiring any image-specific fine tuning. Our method verifies the original results of the perturbation-based methods with an even better precision.
5.6. ADVERSARIAL ATTACK ENHANCEMENT

**Figure 5.4:** Precise saliency of CAMERAS reveals similarity in the level of attention on fine-grained features causes similarity in the prediction confidence (given as percentages) of different models.

**Figure 5.5:** CAMERAS saliency maps reveal the differences of features learned by models to distinguish visually similar objects.

**Prediction confidence:** In Fig. 5.14, CAMERAS results reveal that prediction confidence on individual images is often strongly influenced by a model’s attention on fine-grained features. Different visual models may pay similar attention to the same features to achieve similar confidence scores.

**Discrimination of similar objects:** Precise saliency mapping of CAMERAS also reveals clear differences of the features learned by the models for visually similar objects. In Fig. 5.5, we show the saliency maps for multiple examples of ‘Chain-link Fence’ and ‘Swing’ for two high performing models. Notice how the models pay high attention to the individual chain knots (left) as compared to the larger chain structures (right) to distinguish the two classes. These results also reinforce the importance of ‘not enforcing’ any priors on the map (e.g. smoothness [16]). The shown results provide the first instance of clear saliency differences between similar object features under backpropagation saliency mapping without external priors.

### 5.6 Adversarial Attack Enhancement

Similar to the backpropagation saliency methods, most of the adversarial attacks on deep visual classifiers [2] treat the models as differentiable programs. Using model gradients, they engineer additive noise (i.e. perturbations) that alter model predictions on an input. To avoid attack suspicion, the perturbations must be kept norm-bounded. Projected Gradient Descent (PGD) [22] is considered one of the strongest attacks [2] that computes holistic
perturbations to fool the models. Using PGD as an example, we show that precision saliency of CAMERAS can significantly enhance these attacks by confining the perturbations to the regions considered more salient by our method, see Fig. 5.6.

We iteratively solve for the following using the PGD

$$\min_p \left( J(I_p, \ell_{ll}) + \beta \left\| p \odot (1 - \Psi) \right\|_2 \right), \quad (5.4)$$

where $I_p$ is the perturbed image, $J(\cdot)$ is the cross entropy loss, $\ell_{ll}$ is the least likely label of the clean image, $p$ is the perturbation, and $\beta = 50$ is an empirically chosen scaling factor. In (5.4), we allow the perturbation signal to grow freely for our salient regions while restricting it in the other regions. By focusing only on the most important regions, we are able to drastically reduce the required perturbation norm. Maintaining 99.99% fooling confidence for ResNet-50 on all images of ImageNet validation set, we successfully reduced the PGD perturbation norm by 56.5% on average with our CAMERAS enhancement. See the supplementary material for more details. Other adversarial attacks can also be enhanced similarly with CAMERAS.

5.6.1 Sanity test with adversarial perturbation

Gradient-based adversarial attacks algorithmically compute minimal perturbations to image pixels to maximally change the model prediction. This objective coincides with the objective of image saliency computation, thereby providing a natural sanity check for the saliency methods. That is, the effects of corruption with an adversarial perturbation to image pixels should correspond to the importance of the pixels identified by the saliency method. Leveraging this fact, we develop a sanity check for saliency methods that operates on pixel-by-pixel basis, which is more suited for precise image saliency. We provide details of the test in the supplementary material due to page limits.

5.7 Metrics and additional experiments

5.7.1 Computing time

We compare computational time of the proposed CAMERAS with other techniques in Table 5.3 that reports the average processing time of an image, estimated over 1000 images for three different visual models. For a fair comparison, all of the experiments are performed on NVIDIA Titan V GPU with Pytorch framework, keeping...
Table 5.3: Average computational time for saliency map generation for RISE [20], Extremal Perturbation (Extremal) [8], NormGrad [21], GradCAM [25] and the proposed CAMERAS. GradCAM is the fastest. CAMERAS is generally orders of magnitude faster than the remaining methods.

<table>
<thead>
<tr>
<th>Model</th>
<th>RISE</th>
<th>Extremal</th>
<th>NormGrad</th>
<th>GradCAM</th>
<th>CAMERAS</th>
</tr>
</thead>
<tbody>
<tr>
<td>ResNet</td>
<td>26.4s</td>
<td>35.8s</td>
<td>7.0s</td>
<td>0.07s</td>
<td>0.44s</td>
</tr>
<tr>
<td>DenseNet</td>
<td>79.7s</td>
<td>64.1s</td>
<td>2.5s</td>
<td>0.08s</td>
<td>0.57s</td>
</tr>
<tr>
<td>Inception</td>
<td>55.9s</td>
<td>60.9s</td>
<td>5.0s</td>
<td>0.08s</td>
<td>0.54s</td>
</tr>
</tbody>
</table>

The batch size 1. Publicly available implementations are used for NormGrad [21], Extremal Perturbations [8] and RISE [20]. For GradCAM [25], we use the Torchray package [8]. For the Extremal perturbations, we fix the hyper-parameters ‘area’ to 30% and iterations to 1000. Similarly, RISE has a fixed maximum number of 6000 iterations. These settings generally achieve visually comparable results to CAMERAS.

Table 5.3 shows that CAMERAS computational time is slightly higher than Grad-CAM, which is a highly efficient method for saliency computation. This happens because CAMERAS requires multiple forward/backward passes to construct the saliency map. However, the overall computational cost resulting from these passes are far less than the other methods. This is reflected in the order of magnitudes higher processing times required by RISE and Extremal perturbations.

5.7.2 CAMERAS parameters

Table 5.4: Average difference between CAMERAS saliency maps generated with different resolution steps (‘N’) and fixed maximum resolution $\zeta_m = (1K, 1K)$. The results were computed for ResNet-50 over 1000 randomly sampled images from ILSVRC 2012 validation dataset.

<table>
<thead>
<tr>
<th>Steps $\downarrow$</th>
<th>$N = 5$</th>
<th>$N = 6$</th>
<th>$N = 7$</th>
<th>$N = 8$</th>
<th>$N = 9$</th>
<th>$N = 10$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$N = 5$</td>
<td>0.00</td>
<td>0.11</td>
<td>0.09</td>
<td>0.11</td>
<td>0.10</td>
<td>0.11</td>
</tr>
<tr>
<td>$N = 6$</td>
<td>0.11</td>
<td>0.00</td>
<td>0.13</td>
<td>0.08</td>
<td>0.15</td>
<td>0.10</td>
</tr>
<tr>
<td>$N = 7$</td>
<td>0.09</td>
<td>0.13</td>
<td>0.00</td>
<td>0.11</td>
<td>0.07</td>
<td>0.09</td>
</tr>
<tr>
<td>$N = 8$</td>
<td>0.11</td>
<td>0.08</td>
<td>0.11</td>
<td>0.00</td>
<td>0.12</td>
<td>0.07</td>
</tr>
<tr>
<td>$N = 9$</td>
<td>0.10</td>
<td>0.15</td>
<td>0.07</td>
<td>0.12</td>
<td>0.00</td>
<td>0.10</td>
</tr>
<tr>
<td>$N = 10$</td>
<td>0.11</td>
<td>0.10</td>
<td>0.09</td>
<td>0.07</td>
<td>0.10</td>
<td>0.00</td>
</tr>
</tbody>
</table>

CAMERAS accepts resolution steps (‘N’) as a hyper-parameter. The performance of our technique remains largely insensitive for $N \in [5, 10]$. In order to demonstrate that, we report the average difference between the saliency maps of our method for different values of $N$. A small difference signifies only slight variations in the maps and hence insensitivity of CAMERAS to the hyper-parameter. In Table 5.4, we report the average difference values, where the value for a step size between $N_1$ and $N_2$ is computed as:

$$E \left[ \frac{||\Psi_1 - \Psi_2||_2}{||\Psi_1||_2 + ||\Psi_2||_2}/2 \right] \quad \text{s.t} \quad \Psi_1 = \text{CAMERAS}(N_1), \ \Psi_2 = \text{CAMERAS}(N_2),$$

(5.5)

From the table 5.4, it can be observed that the difference between the saliency maps stays within 15% of the average norm of the original maps. This is true even with the reduction in number of steps by half. This validates that the performance of CAMERAS is reasonably stable for different values of ‘N’.
CHAPTER 5. ENHANCED RESOLUTION & SANITY PRESERVING SALIENCY MAPS

Figure 5.7: Saliency maps for an image from Pascal VOC 2007 [7] over ResNet-50 model. The second column includes the result for Grad-CAM (top-row), its score for the pointing game (middle row) and its score on our proposed metrics score (last row). The corresponding results for CAMERAS are given in the last column. Our metrics account for the overall precision on the saliency map.

5.7.3 Further discussion on the proposed evaluation metrics

In the main paper, we indicated that due to the crudeness of the saliency maps resulting from the earlier methods, existing evaluation metrics are also meant to evaluate the maps imprecisely. For instance, Pointing game [34] evaluates a saliency map by checking if the maximal point in the saliency map lies within the bounding box of an object. If this is true, it assigns the map the best score. However, no attention is paid to the rest of the region identified by the map. For example, in Figure 5.7 (top-row) Pointing game assigns equal score to the saliency maps of ‘GradCAM’ ($\Psi_{GradCAM}$) and ‘CAMERAS’ ($\Psi_{CAMERAS}$), ignoring the fact that Grad-CAM map also includes the background region outside the bounding box. The binary nature of the score does not reveal much about the actual quality of the attribution maps. Therefore, a more precise metric is needed that scores the attribution maps corresponding to model confidence over the salient region. Our proposed metrics capture this notion adequately. This is reflected in Figure 5.7 where Grad-CAM clearly gets penalized for inclusion of the irrelevant pixels under our evaluation metrics.

We further scrutinize the behaviour of the proposed metrics over the inclusion of irrelevant pixels. Inclusion of irrelevant pixels is generally the result of blobiness caused by interpolation in the existing methods. This imprecision should result in reduction of an appropriate metric score. We empirically verify this for our metrics by linearly increasing the CAMERAS saliency map to the GradCAM map. This is performed by setting different values of $\gamma$ in Equation 5.6

$$\Psi_{interpolated} = \Psi_{CAMERAS} + \gamma(\Psi_{GradCAM} - \Psi_{CAMERAS}) \quad \text{s.t} \quad \gamma \in [0, 1].$$  (5.6)

We gather a number of interpolated saliency maps by incremental increase in the salient region. The proposed metric is computed over these interpolated saliency maps and their trend is analyzed. Figure 5.8 shows some of the interpolated saliency maps (left) and reports the metric variation in a plot (right). The graph matches our intuition that as more and more irrelevant pixels are included, the scores decreases monotonically. This demonstrates that the metric adequately captures the attribution from all the pixels. We note that, for all of these interpolated saliency maps, the pointing game score stays the same. This reinforces our argument for the need of more precise evaluation metrics for image saliency.
5.7. METRICS AND ADDITIONAL EXPERIMENTS

Figure 5.8: Interpolated saliency maps computed under Equation 5.6 with changing value of $\gamma$. For $\gamma = 0$, the map corresponds toCAMERAS. For $\gamma = 1$, the map resembles to that computed by GradCAM. The graph on the right shows the corresponding impact on $\rho_{map}$ score.

5.7.4 Adversarial attacks inspired sanity check

Adversarial attacks alter image pixels to change the prediction of a model. These alterations are systematic - in the directions that maximally alter the model output. Intuitively, cleaning the adversarially attacked pixels by replacing them with the original pixels, should restore the model’s confidence on the original label of the image. This observation inspires a simple sanity check for the image saliency methods - a test that implicitly accounts for the precision of the map.

Saliency methods ultimately aim at tallying the model-centric importance of individual pixels of an image. Therefore, for a ‘sane’ method that assigns correct importance to the individual pixels, iteratively cleaning the adversarial version of those pixels in a sorted manner should result in a steep rise of model’s confidence on the original image label. For a saliency map that fails to compute the correct importance for the individual pixels, restoration of the confidence with this process is expected to be slow and unstable. This behavior indicates the method to be ‘less’ sane. We make two observations here. First, our sanity check does not have a binary outcome. Though less acknowledged, this fact also holds for the other popular sanity checks. For instance, the model layer randomisation of [2] also does not offer a binary decision. Thus, a method can be less sane than another under the proposed sanity check. Second, our definition of sanity is based on individual pixel importance. This is a tough criterion, emulating the hardest possible scenario for the perfect sanity. That is, only the method that computes correct relative importance for all the pixels in an image can perfectly pass this test.

To observe a map’s sanity, we suggest iterative cleaning of the adversarial pixels in an attacked version of the image in the descending order of pixel importance, as stipulated by the computed saliency map. An ideal saliency map will raise the ground truth confidence of the model almost instantly under this cleaning process. Therefore, the area under probability-pixels curve for an ideal saliency map will be

$$\text{Area}_{\text{Ideal}} \approx P_{\text{GroundTruth}} \times H \times W \quad \text{s.t} \quad P_{\text{GroundTruth}} = P(K(I)), \quad I \in \mathbb{R}^{H \times W \times C}, \quad (5.7)$$

Any deviation from the ideal saliency mask will introduce a corresponding change in the area under the curve, which can be used as an indicator of sanity. The deviation can be computed by observing the difference in the area under probability-pixels graph of an ideal and a given saliency map. A smaller difference will be more desirable.

To implement this sanity check, we corrupt a given image by the proposed enhanced PGD scheme and afterward iteratively clean the resulting adversarial image in the order of importance indicated by the saliency map. The computational complexity of this process is directly proportional to number of pixels involved.
CHAPTER 5. ENHANCED RESOLUTION & SANITY PRESERVING SALIENCY MAPS

Figure 5.9: Probability-pixels curves (last column) of Grad-CAM (Brown) and CAMERAS (Blue) for the image shown in first column. The saliency map from CAMERAS is shown in second column and Grad-CAM in the third column respectively. The sanity scores (smaller is better) are indicated over the saliency maps in ‘yellow’ fonts. Scores are evaluated with the ideal area composed of 10% image pixels.
Therefore, we fix the maximum number of pixels to be cleaned to a fraction of the original image size. We evaluate the sanity score as,

$$Sanity = \frac{\text{Area}_{\text{Ideal}} - \text{Area}_\Psi}{\text{Area}_{\text{Ideal}}} ,$$

(5.8)