Understanding Deep Networks via Extremal Perturbations and Smooth Masks

facebook Artificial Intelligence

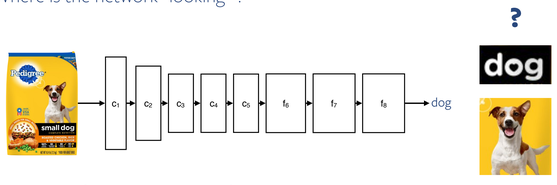
Ruth Fong*† Mandela Patrick*† Andrea Vedaldi†‡

†University of Oxford ‡Facebook AI Research *equal contribution

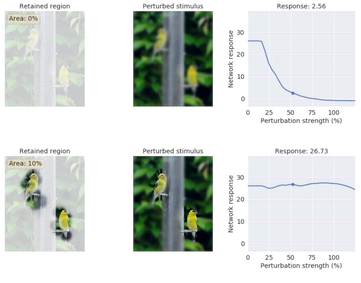
University of Oxford

Attribution

Where is the network "looking" ?



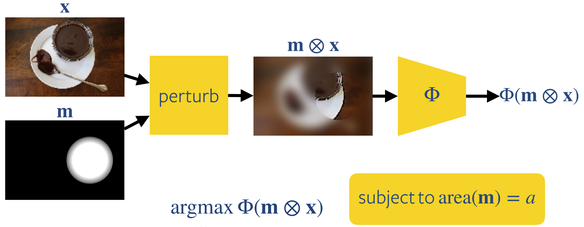
Perturbation analysis



- We test a region counterfactually by blurring it's complement and observing the change in the network's response
- **Extremal region**: the region that, for a given area, maximises the response

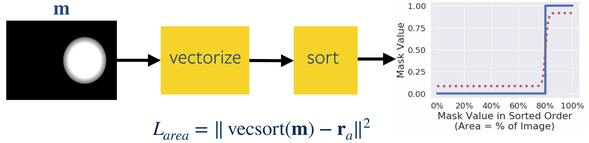
Formalization

We optimize the mask $\mathbf{m}$ to maximize the response of the network Φ on the blurred image $\mathbf{m} \otimes \mathbf{x}$ for a given area a:



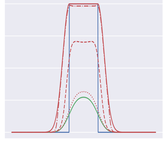
Area constraint

Optimizing for a given area size is non-trivial. We do it by sorting the mask values and comparing the result to the desired o-1 distribution $\mathbf{r}_a$:



$$L_{area} = \| \text{vecsort}(\mathbf{m}) - \mathbf{r}_a \|^2$$

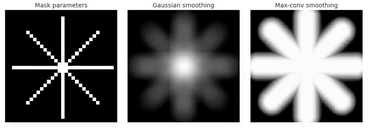
Smooth mask



- $m(v)$: mask
- $\text{conv}(u; m; k) = \frac{1}{Z} \sum_{v \in \Omega} k(u - v) m(v)$
- $\text{maxconv}(u; m; k) = \max_{v \in \Omega} k(u - v) m(v)$
- $\text{smoothconv}(u; m; k; T) = \text{smax}_{v \in \Omega; T} \, k(u - v) m(v)$

$$\text{smax}_{u \in \Omega; T} \, f(u) = \frac{\sum_u f(u) \exp(f(u)/T)}{\sum_u \exp(f(u)/T)}$$

Right: comparison between original mask (L), mask after conv (M), and maxconv (R).



Algorithm

Pick area a and perform SGD to optimize:

$$\underset{\mathbf{m}}{\text{argmax}} \, \Phi(\text{smoothconv}(\mathbf{m}) \otimes \mathbf{x}) - \lambda \| \text{vecsort}(\text{smoothconv}(\mathbf{m})) - \mathbf{r}_a \|^2$$

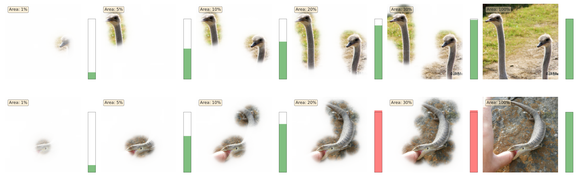
Results

Pointing Game

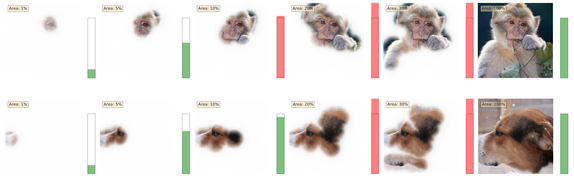
Quantitative evaluation is difficult; that said, the method achieves competitive results on pointing game [6].

	VOC07 Test (All/Diff)		*COCO14 Val (All/Diff)*	
Method	*VGG16*	*ResNet50*	*VGG16*	*ResNet50*
Grad	76.3/56.9	72.3/56.8	37.7/31.4	35.0/29.4
DConv	67.5/44.1	68.6/44.7	30.7/23.0	30.0/21.9
Guid.	75.9/53.0	77.2/59.5	39.1/31.4	42.1/35.3
MWP	77.1/56.6	84.4/70.8	39.8/32.8	49.6/43.9
cMWP	79.9/66.5	**90.6**/82.2	49.7/44.3	**58.5/53.6**
RISE*	87.3/—	88.9/—	50.7/—	55.6/—
GCAM	86.6/74.0	90.4/**82.3**	**54.2/49.0**	57.3/52.3
Ours	**88.7/75.5**	86.3/73.4	53.4/47.7	55.7/46.9

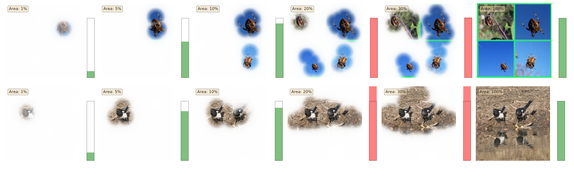
Foreground evidence is usually sufficient



Objects are recognized by their details

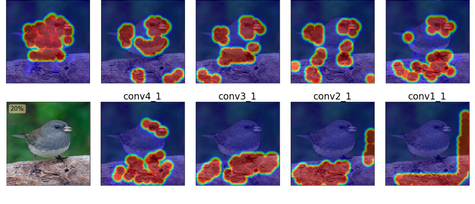


Multiple objects contribute cumulatively

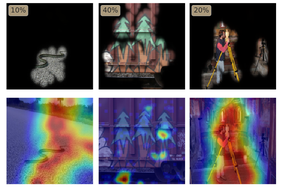


Sanity Checks

Our method is sensitive to model weights. Below: model weights are progressively randomized [7].

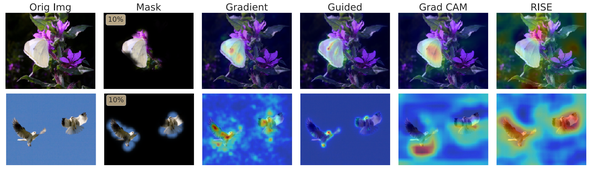


Comparisons

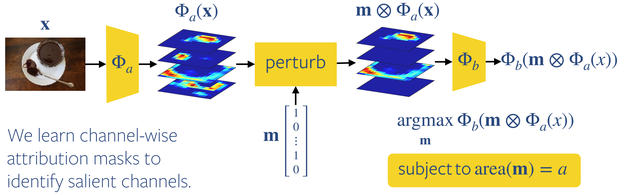


Left: Extremal perturbations (ours) vs. meaningful perturbations [1]. We show that our method is more sensitive and stable than prior work.

Below: Comparison with related works.



Channel attribution



We learn channel-wise attribution masks to identify salient channels.

Right: "Diff" activations with feature inversions (M & R) to inspect which features are highlighted and minimized by our channel attribution mask.

TorchRay

Interpretability library: github.com/facebookresearch/torchray

PyTorch

References

[1] Fong, Vedaldi, ICCV17, [2] Simonyan et al., ICLRW14, [3] Springenberg et al., ICLRW15, [4] Selvaraju et al., ICCV17, [5] Petsiuk et al., BMVC17, [6] Zhang et al., ECCV16, [7] Adebayo et al., NeuRIPS18.