Towards Improved Input Masking for Convolutional Neural Networks: Appendix

1. Implementation details

In order to fairly compare the masking techniques, we fix the number of segments that the segmentation algorithm partitions the image into to approximately equal around 200. We use the `sklearn` implementation for SLIC and quickshift. For SLIC, we fix the approximate number of segments to 196. For quickshift, we set `kernel_size=2`, `max_dist=200`, `ratio=0.2`, which produces approximately 200 segments per image. For LIME, we use 500 random samples to train the linear classifier.

For the token dropping variant of Vision Transformers (ViT and DeiT), we use code from https://github.com/MadryLab/missingness.

2. Comparison of layer masking with partial convolution

Partial convolution is a method for image inpainting introduced by Liu et al, 2018. Partial convolution handles convolution over images with irregular holes by using a method similar to layer masking. However, instead of doing neighbor padding as in layer masking, the convolutions over the edge is **scaled up** by a factor of $\frac{k^2}{\boldsymbol{m} \odot \mathbf{1}_{k \times k}}$ (where $\boldsymbol{m}$ is the binary mask corresponding to the field of the convolution and k is the size of the filter). This means that the edge convolutions are given a *higher* weight than normal. While this may be useful for inpainting purposes, where most of the important information is concentrated around the edges and parameters of the neural network can be trained, it is exactly the opposite of what we want, as this worsens the edge artifact problem which we cannot fix by training. Thus, naively using partial convolution is worse than even zero padding as far as accuracy or unchanged predictions are concerned. We thus find that the AUC for the accuracy (or class entropy) vs fraction of masked image is only **0.1922** (or **3.8589**) when we use partial convolution layers, which is much lower than corresponding numbers for layer masking (see Fig. 1).

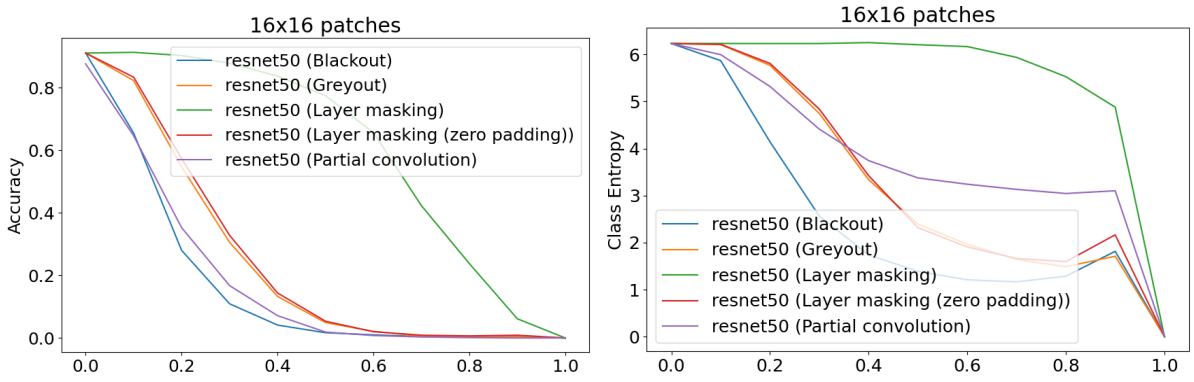


Figure 1. Accuracy and class entropy vs fraction of 16×16 patches of the image masked out in random order using various masking methods on ResNet-50

3. Ablation study

To further investigate the effect of layer masking and neighbor padding on model behavior, we construct 3 variants of layer masking: (a) With zero padding instead of neighbor padding (b) Masking and padding only the first two residual blocks, (c) Masking and padding only the first convolutional layer, ReLU and BatchNorm layer

Using a similar setup as in Sec. 4.1, we compute the area under curve (AUC) for each plot of metric vs fraction of segments dropped. The AUC values are averaged over different segmentation algorithms (SLIC, quickshift, etc) and masking orders (random, salient first, etc)(refer Tab. 1).

We find that both neighbor padding and masking all layers are important to the masking technique. Layer masking with zero padding is still better than blackout or greyout, but much worse than with neighbor padding. Layer masking only the first two residual blocks is also inferior to masking through all layers, but we find that there are diminishing returns, as we are able to obtain much of the improvement by masking only half of the layers.

	Accuracy	Class Entropy	Wordnet Similarity	Unchanged Predictions
Blackout	0.3881	4.6473	0.6930	0.4094
Greyout	0.4398	4.9408	0.7167	0.4636
Layer masking:				
On all layers	**0.5604**	**5.6021**	**0.7881**	**0.5907**
On 1st and 2nd residual blocks	0.5103	5.0962	0.7616	0.5391
Zero padding	0.4502	5.0388	0.7262	0.4747

Table 1. Average AUC for different variants of layer masking alongside the black out and grey out baselines (model: ResNet-50). Higher the better

4. Extended results for segment masking experiments (Section 4.1)

We also measure the degradation of WordNet similarity and change in predictions as segments are removed for models like ResNet-50, ResNet-50 with augmentations, DenseNet, SqueezeNet, AlexNet, EfficientNet and MobileNet. Note that EfficientNet and MobileNet are also trained with grey missingness data augmentations, thus greyout is disproportionately more robust for these models. Most significant differences are found in random 16×16 patch removal

ResNet-50	Quickshift segments			16×16 patches			SLIC superpixels		
	Random	Most sal. first	Least sal. first	Random	Most sal. first	Least sal. first	Random	Most sal. first	Least sal. first
Unchanged preds									
Blackout	0.415	0.134	0.832	0.193	0.359	0.608	0.362	0.213	0.699
Greyout	0.484	0.146	0.853	0.253	0.393	0.648	0.437	0.247	0.746
Layer masking	**0.580**	**0.169**	**0.877**	**0.608**	**0.511**	**0.741**	**0.569**	**0.311**	**0.808**
Wordnet Sim									
Blackout	0.705	0.547	0.889	0.571	0.672	0.802	0.669	0.591	0.838
Greyout	0.748	0.517	0.892	0.604	0.696	0.821	0.718	0.606	0.857
Layer masking	**0.785**	**0.549**	**0.904**	**0.800**	**0.757**	**0.865**	**0.779**	**0.642**	**0.884**
Accuracy									
Blackout	0.395	0.124	0.767	0.181	0.340	0.582	0.347	0.200	0.657
Greyout	0.463	0.137	0.787	0.240	0.374	0.621	0.418	0.234	0.702
Layer masking	**0.551**	**0.159**	**0.806**	**0.577**	**0.484**	**0.703**	**0.542**	**0.294**	**0.756**
Class entropy									
Blackout	4.988	2.224	5.824	2.574	4.570	5.406	4.815	3.976	5.661
Greyout	5.327	2.362	5.875	3.289	4.807	5.642	5.022	4.229	5.808
Layer masking	**5.698**	**2.572**	**5.892**	**5.651**	**5.782**	**5.879**	**5.607**	**4.763**	**5.876**

Table 2. AUC of **(From top)** Fraction of unchanged predictions, Wordnet similarity of the predictions to true label, the accuracy of predictions, and class entropy of the predictions vs fraction of segments masked out for plain ResNet-50

ResNet-50 (augmented)	Quickshift segments			16×16 patches			SLIC superpixels		
	Random	Most sal. first	Least sal. first	Random	Most sal. first	Least sal. first	Random	Most sal. first	Least sal. first
Unchanged preds									
Blackout	0.552	0.169	0.872	0.671	0.632	0.831	0.513	0.297	0.791
Greyout	0.734	0.232	0.894	0.746	0.644	0.838	0.708	0.417	0.854
Layer masking	0.621	0.186	0.884	0.622	0.557	0.775	0.603	0.348	0.828
Wordnet Sim									
Blackout	0.787	0.577	0.912	0.838	0.823	0.903	0.770	0.654	0.886
Greyout	0.866	0.610	0.919	0.872	0.830	0.906	0.858	0.714	0.910
Layer masking	0.789	0.493	0.904	0.795	0.761	0.862	0.782	0.624	0.887
Accuracy									
Blackout	0.534	0.162	0.831	0.651	0.612	0.799	0.498	0.289	0.760
Greyout	0.708	0.225	0.851	0.723	0.624	0.807	0.687	0.405	0.821
Layer masking	0.603	0.180	0.843	0.605	0.540	0.749	0.585	0.338	0.798
Class entropy									
Blackout	5.520	2.458	5.878	5.742	5.807	5.892	5.438	4.511	5.834
Greyout	5.875	2.628	5.895	5.860	5.865	5.898	5.862	4.849	5.896
Layer masking	5.603	2.258	5.893	5.554	5.630	5.789	5.554	4.305	5.887

Table 3. AUC of **(From top)** Fraction of unchanged predictions, Wordnet similarity of the predictions to true label, the accuracy of predictions, and class entropy of the predictions vs fraction of segments masked out for ResNet-50 trained with data augmentations

WideResNet-50	Quickshift segments			16×16 patches			SLIC superpixels		
	Random	Most sal. first	Least sal. first	Random	Most sal. first	Least sal. first	Random	Most sal. first	Least sal. first
Unchanged preds									
Blackout	0.445	0.129	0.862	0.226	0.401	0.666	0.389	0.227	0.731
Greyout	0.515	0.141	0.877	0.293	0.447	0.707	0.472	0.263	0.778
Layer masking	0.596	0.158	0.891	0.620	0.526	0.769	0.584	0.323	0.823
Wordnet Sim									
Blackout	0.718	0.544	0.906	0.602	0.695	0.826	0.693	0.603	0.855
Greyout	0.757	0.529	0.909	0.624	0.727	0.846	0.733	0.618	0.874
Layer masking	0.798	0.569	0.919	0.810	0.768	0.878	0.793	0.661	0.897
Accuracy									
Blackout	0.435	0.125	0.835	0.220	0.394	0.652	0.379	0.222	0.711
Greyout	0.504	0.137	0.851	0.288	0.440	0.694	0.461	0.258	0.758
Layer masking	0.587	0.154	0.864	0.609	0.516	0.754	0.574	0.317	0.802
Class entropy									
Blackout	5.161	2.102	5.865	2.706	4.855	5.585	4.967	4.224	5.706
Greyout	5.372	2.177	5.884	3.299	5.204	5.718	5.165	4.408	5.815
Layer masking	5.734	2.354	5.891	5.668	5.790	5.884	5.664	4.819	5.878

Table 4. AUC of **(From top)** Fraction of unchanged predictions, Wordnet similarity of the predictions to true label, the accuracy of predictions, and class entropy of the predictions vs fraction of segments masked out for WideResNet-50

AlexNet	Quickshift segments			16×16 patches			SLIC superpixels		
	Random	Most sal. first	Least sal. first	Random	Most sal. first	Least sal. first	Random	Most sal. first	Least sal. first
Unchanged preds									
Blackout	0.278	0.104	0.789	0.128	0.224	0.431	0.234	0.147	0.573
Greyout	0.364	0.111	0.830	0.197	0.268	0.501	0.326	0.183	0.663
Layer masking	0.437	0.120	0.853	0.487	0.358	0.597	0.458	0.226	0.733
Wordnet Sim									
Blackout	0.629	0.526	0.856	0.499	0.584	0.704	0.590	0.551	0.774
Greyout	0.681	0.523	0.874	0.536	0.615	0.745	0.652	0.577	0.815
Layer masking	0.721	0.527	0.882	0.734	0.676	0.794	0.729	0.601	0.845
Accuracy									
Blackout	0.256	0.091	0.696	0.115	0.205	0.397	0.215	0.132	0.517
Greyout	0.337	0.098	0.732	0.181	0.245	0.462	0.302	0.166	0.599
Layer masking	0.405	0.107	0.753	0.449	0.334	0.547	0.424	0.208	0.662
Class entropy									
Blackout	4.587	1.917	5.790	2.794	4.444	5.243	4.342	3.930	5.522
Greyout	5.059	2.031	5.870	3.577	4.903	5.472	4.852	4.164	5.717
Layer masking	5.406	2.230	5.889	5.196	5.439	5.690	5.360	4.432	5.825

Table 5. AUC of **(From top)** Fraction of unchanged predictions, Wordnet similarity of the predictions to true label, the accuracy of predictions, and class entropy of the predictions vs fraction of segments masked out for AlexNet

SqueezeNet	Quickshift segments			16×16 patches			SLIC superpixels		
	Random	Most sal. first	Least sal. first	Random	Most sal. first	Least sal. first	Random	Most sal. first	Least sal. first
Unchanged preds									
Blackout	0.285	0.103	0.794	0.141	0.243	0.469	0.240	0.147	0.578
Greyout	0.355	0.110	0.823	0.182	0.279	0.515	0.312	0.175	0.648
Layer masking	0.408	0.113	0.844	0.544	0.362	0.592	0.405	0.203	0.694
Wordnet Sim									
Blackout	0.627	0.524	0.842	0.515	0.579	0.721	0.599	0.545	0.767
Greyout	0.680	0.534	0.855	0.529	0.605	0.747	0.656	0.573	0.801
Layer masking	0.694	0.483	0.853	0.750	0.666	0.775	0.689	0.568	0.810
Accuracy									
Blackout	0.251	0.083	0.645	0.117	0.207	0.416	0.208	0.124	0.488
Greyout	0.311	0.090	0.669	0.154	0.237	0.456	0.270	0.148	0.546
Layer masking	0.357	0.093	0.683	0.472	0.314	0.518	0.357	0.174	0.585
Class entropy									
Blackout	4.593	1.832	5.801	2.360	4.379	5.183	4.410	3.667	5.491
Greyout	5.017	1.935	5.865	2.931	4.755	5.483	4.776	3.903	5.673
Layer masking	5.052	2.018	5.879	5.266	5.065	5.506	4.926	4.074	5.708

Table 6. AUC of **(From top)** Fraction of unchanged predictions, Wordnet similarity of the predictions to true label, the accuracy of predictions, and class entropy of the predictions vs fraction of segments masked out for SqueezeNet

DenseNet	Quickshift segments			16×16 patches			SLIC superpixels		
	Random	Most sal. first	Least sal. first	Random	Most sal. first	Least sal. first	Random	Most sal. first	Least sal. first
Unchanged preds									
Blackout	0.423	0.124	0.852	0.184	0.387	0.629	0.373	0.217	0.709
Greyout	0.481	0.134	0.865	0.272	0.403	0.652	0.449	0.247	0.751
Layer masking	0.483	0.127	0.873	0.503	0.434	0.671	0.497	0.256	0.774
Wordnet Sim									
Blackout	0.712	0.543	0.888	0.555	0.686	0.805	0.686	0.600	0.838
Greyout	0.746	0.552	0.894	0.607	0.700	0.821	0.726	0.619	0.858
Layer masking	0.733	0.550	0.897	0.743	0.726	0.829	0.737	0.625	0.864
Accuracy									
Blackout	0.400	0.115	0.776	0.173	0.366	0.594	0.351	0.204	0.656
Greyout	0.456	0.124	0.790	0.256	0.381	0.618	0.423	0.232	0.696
Layer masking	0.456	0.118	0.796	0.477	0.412	0.632	0.469	0.243	0.718
Class entropy									
Blackout	4.987	1.999	5.861	2.606	5.090	5.555	4.868	4.092	5.667
Greyout	5.321	2.160	5.880	3.549	5.176	5.660	5.234	4.365	5.798
Layer masking	5.051	2.017	5.885	5.152	5.391	5.661	5.050	4.215	5.773

Table 7. AUC of **(From top)** Fraction of unchanged predictions, Wordnet similarity of the predictions to true label, the accuracy of predictions, and class entropy of the predictions vs fraction of segments masked out for DenseNet

MobileNet-v3	Quickshift segments			16×16 patches			SLIC superpixels		
	Random	Most sal. first	Least sal. first	Random	Most sal. first	Least sal. first	Random	Most sal. first	Least sal. first
Unchanged preds									
Blackout	0.525	0.144	0.871	0.278	0.406	0.646	0.495	0.271	0.761
Greyout	0.661	0.181	0.888	0.633	0.576	0.779	0.633	0.355	0.815
Layer masking	0.516	0.135	0.873	0.619	0.441	0.644	0.531	0.271	0.767
Wordnet Sim									
Blackout	0.769	0.557	0.910	0.612	0.702	0.822	0.751	0.636	0.872
Greyout	0.829	0.542	0.911	0.817	0.791	0.878	0.816	0.665	0.889
Layer masking	0.753	0.531	0.907	0.804	0.715	0.813	0.764	0.623	0.870
Accuracy									
Blackout	0.515	0.139	0.837	0.272	0.398	0.632	0.486	0.266	0.738
Greyout	0.644	0.176	0.853	0.621	0.562	0.757	0.617	0.345	0.788
Layer masking	0.504	0.131	0.838	0.605	0.430	0.627	0.520	0.265	0.744
Class entropy									
Blackout	5.560	2.204	5.897	4.069	5.319	5.679	5.490	4.504	5.848
Greyout	5.813	2.324	5.897	5.776	5.805	5.886	5.771	4.753	5.895
Layer masking	5.336	2.152	5.887	5.532	5.309	5.540	5.341	4.406	5.801

Table 8. AUC of **(From top)** Fraction of unchanged predictions, Wordnet similarity of the predictions to true label, the accuracy of predictions, and class entropy of the predictions vs fraction of segments masked out for MobileNet

EfficientNet	Quickshift segments			16×16 patches			SLIC superpixels		
	Random	Most sal. first	Least sal. first	Random	Most sal. first	Least sal. first	Random	Most sal. first	Least sal. first
Unchanged preds									
Blackout	0.591	0.162	0.883	0.261	0.439	0.666	0.581	0.326	0.805
Greyout	0.729	0.204	0.896	0.686	0.605	0.804	0.723	0.411	0.849
Layer masking	0.553	0.146	0.881	0.581	0.476	0.692	0.572	0.302	0.802
Wordnet Sim									
Blackout	0.804	0.569	0.910	0.595	0.716	0.824	0.800	0.667	0.887
Greyout	0.860	0.594	0.915	0.842	0.807	0.891	0.859	0.709	0.903
Layer masking	0.769	0.533	0.906	0.788	0.737	0.839	0.778	0.637	0.880
Accuracy									
Blackout	0.570	0.154	0.835	0.252	0.424	0.641	0.559	0.314	0.768
Greyout	0.697	0.194	0.847	0.661	0.580	0.772	0.692	0.394	0.809
Layer masking	0.532	0.139	0.834	0.561	0.459	0.666	0.549	0.290	0.766
Class entropy									
Blackout	5.642	2.222	5.893	3.772	5.378	5.661	5.654	4.610	5.868
Greyout	5.879	2.430	5.893	5.848	5.857	5.897	5.880	4.941	5.895
Layer masking	5.356	2.099	5.891	5.410	5.464	5.682	5.367	4.390	5.844

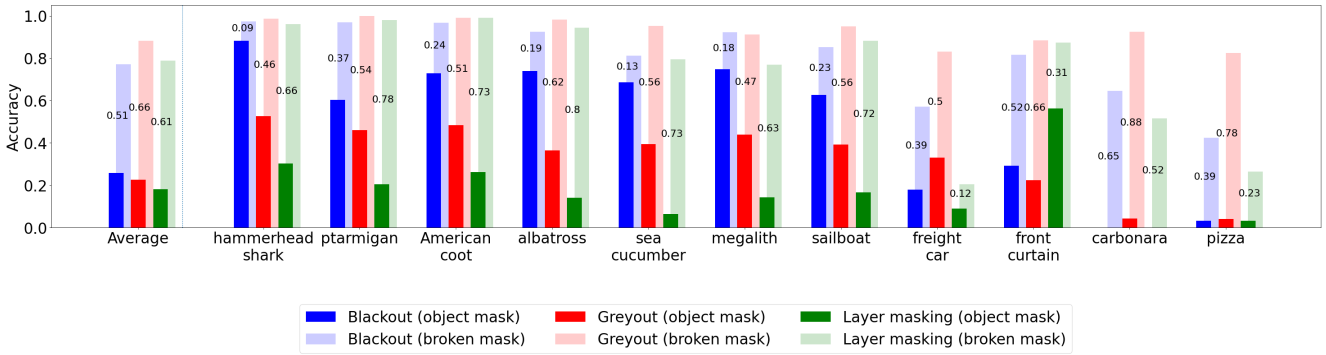
Table 9. AUC of **(From top)** Fraction of unchanged predictions, Wordnet similarity of the predictions to true label, the accuracy of predictions, and class entropy of the predictions vs fraction of segments masked out for EfficientNet

5. Extended experiments on shape bias (Section 4.2 and Section 4.3)

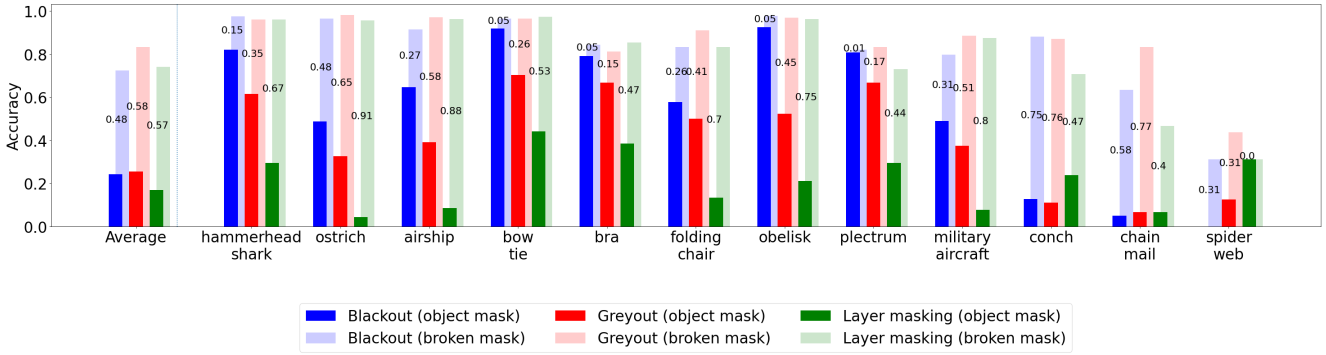
We now show the bar plots for object masked and broken masked cases for different CNN architectures. Consistent with the previous section, we observe that layer masking is more robust as compared to black-out or grey-out for Wide ResNet-50, AlexNet, SqueezeNet and DenseNet (Fig. 3). Also, the object masked accuracy is typically lower on average. Looking at specific classes, we see similar trends as mentioned in Section 4.2. There are many classes for like megalith, obelisk, sunglasses, etc in which the object's true color is very close to the masking color, and the shape of the object mask conveys a lot of information about the class itself. Conversely, other classes like priarie grouse, bee eater, southern black widow, etc get misclassified as other related classes when masked out using black or grey baseline colors at a higher -than-ideal rate as compared to layer masking.

However, for EfficientNet and MobileNet-v3 (Fig. 2), we find that owing to its pretraining on data augmentations, it is more robust grey-out masking, even compared to layer masking. Still, consistent with Section 4.3, we find classes like megalith and hammerhead shark where layer masking can be more helpful, but also classes like pizza or carbonara where it is not.

In conclusion, we should be cognizant of the missingness biases of a masking method when applied to a model, both shape and color, when evaluating a model's dependence on various image features. Layer masking can be particularly useful in cases where the object to be masked has a distinctive shape with its color also being similar to the baseline color (for e.g: obelisk, megalith, sunglasses). It may also be useful in situations where there exists another class closely related to the true class which has a similar shape but different color which closely resembles the masking color (for e.g:). It may not be so useful in situations where shape is not very indicative of object and model is already robust to some color replacing masking method like greyout (e.g: pizza, crate, carbonara).

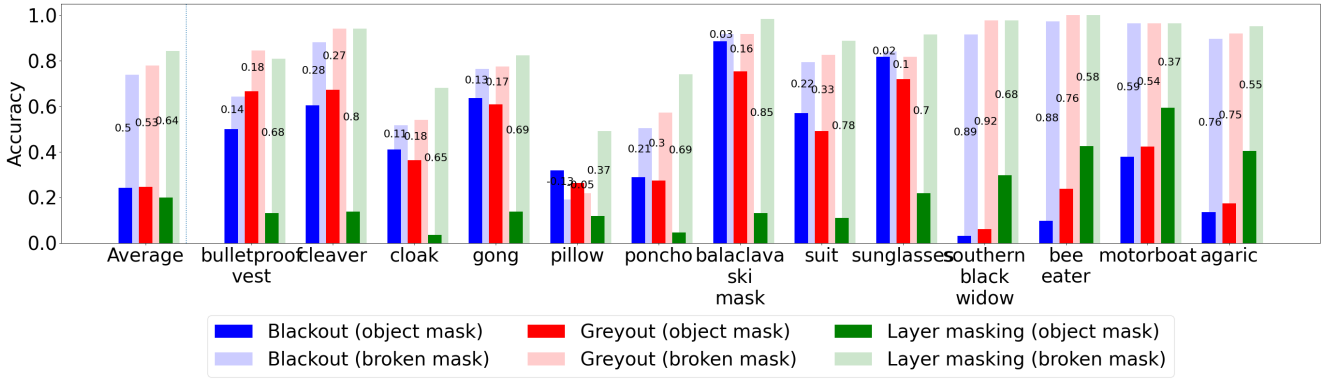


(a) EfficientNet

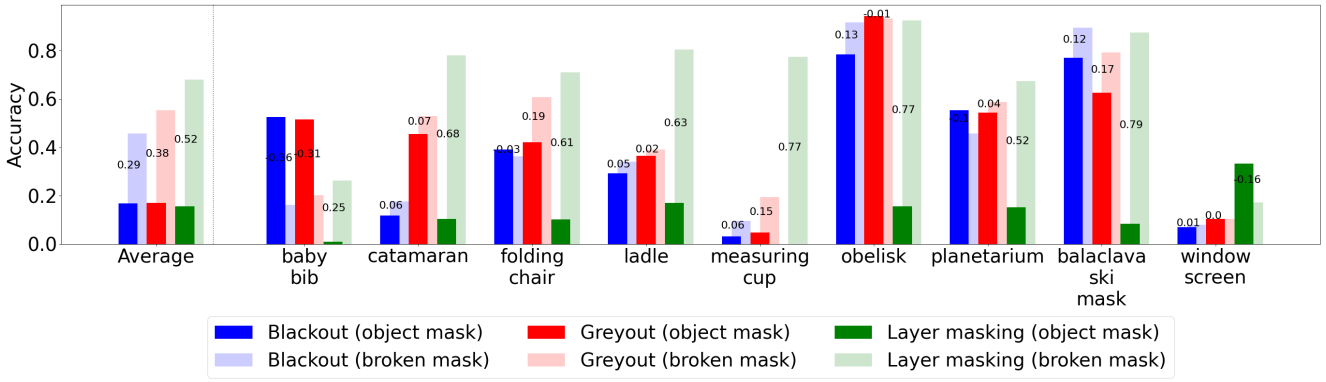


(b) MobileNet

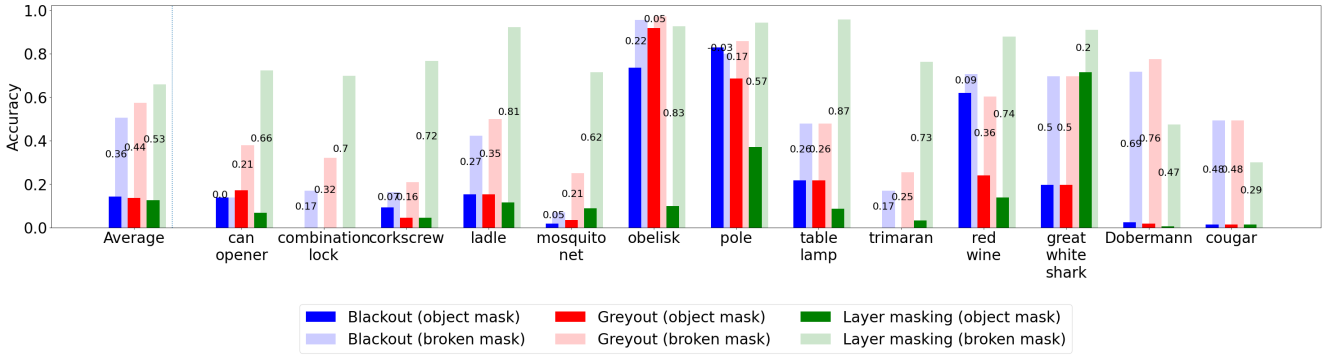
Figure 2. Effect of shape bias (measured as in Section 4.2) for EfficientNet and MobileNet ()



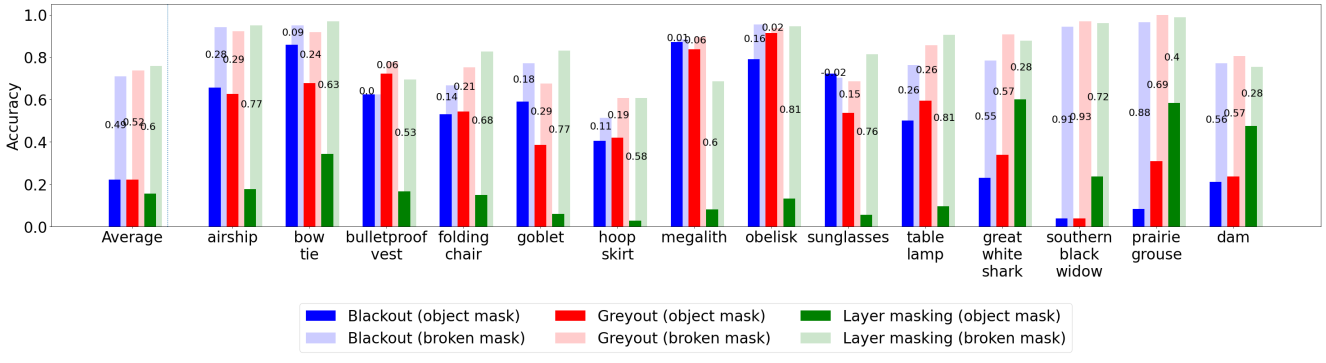
(a) Wide ResNet-50



(b) AlexNet



(c) SqueezeNet



(d) DenseNet

Figure 3. Effect of shape bias (measured as in Section 4.2) for Wide ResNet-50, AlexNet, SqueezeNet and DenseNet

6. Extended experiments on LIME (Section 4.4)

6.1. Qualitative

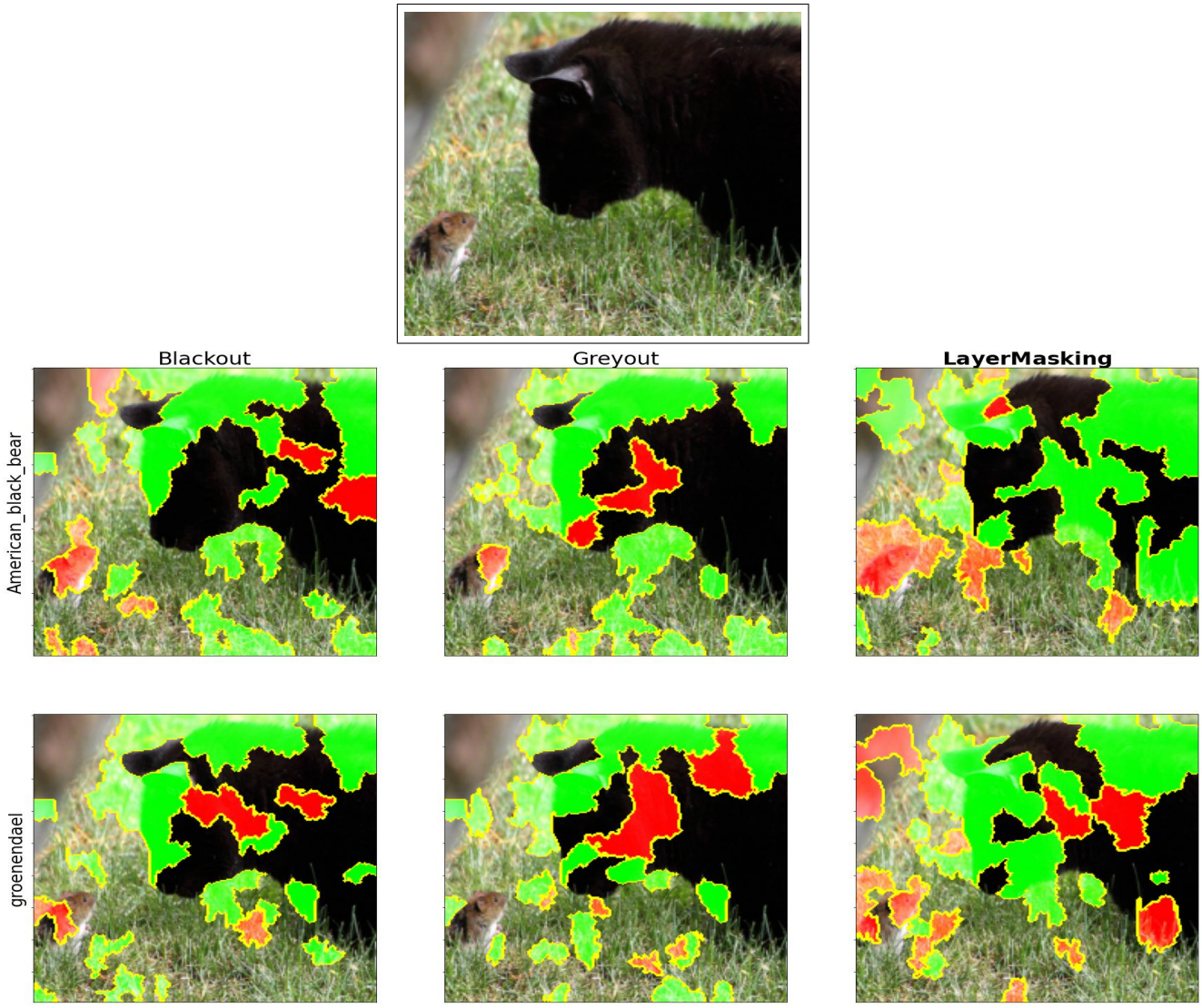


Figure 4. Visualization of LIME scores for the top two predictions of ResNet-50 on a sample image of a cat and a mouse. Columns correspond to the masking techniques (blacking out, greying out, and layer masking), rows are the top 2 predictions. The top two predictions are American black bear and mouse. Green regions contribute to the prediction, red regions detract from the prediction.

We also include some more visualizations of LIME scores on random images from ImageNet, with most important segments highlighted in green (positive score) or red (negative score). These are **not** cherrypicked.

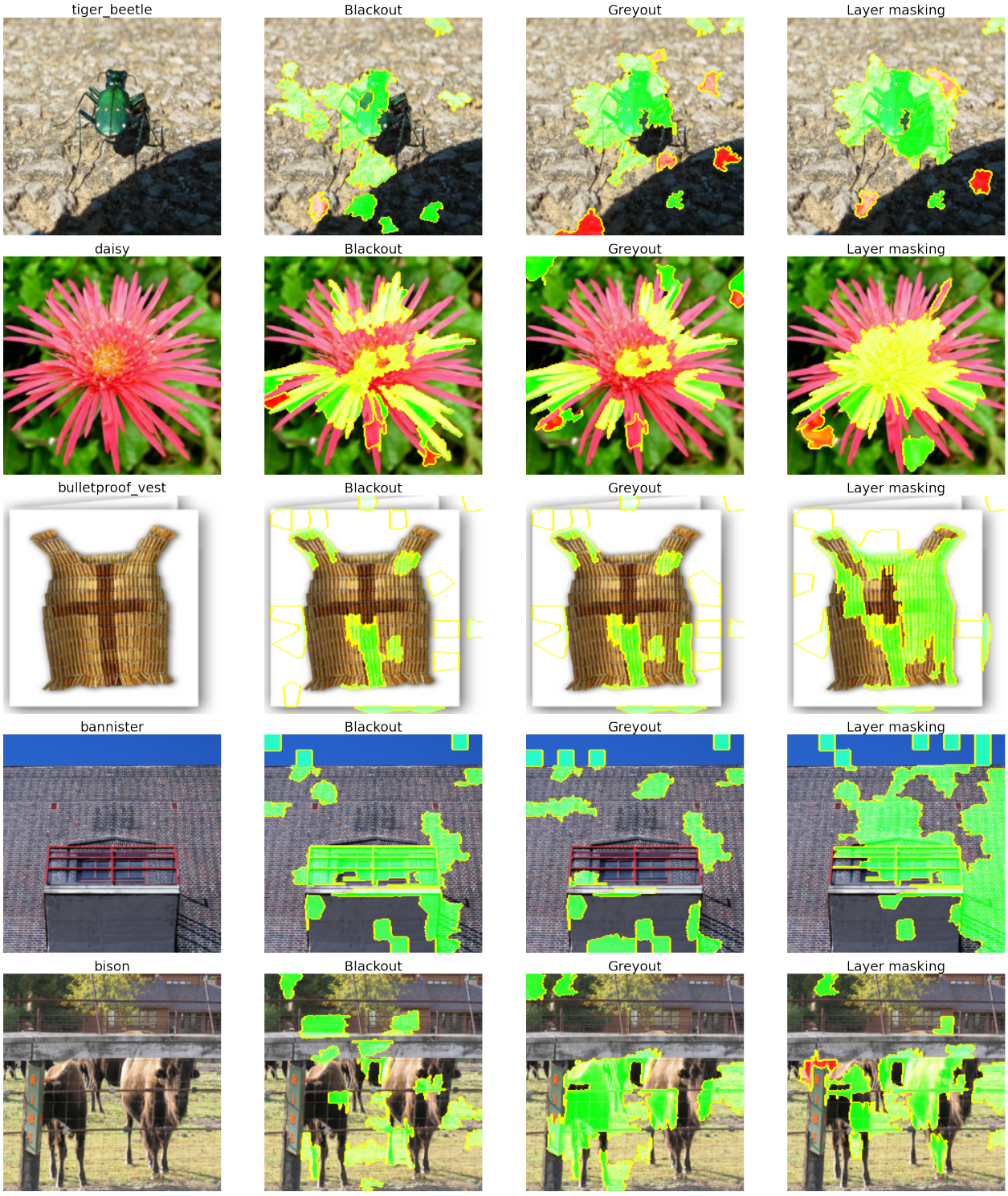


Figure 5. LIME scores using SLIC segmentation (5 samples). Top 15 segments are highlighted

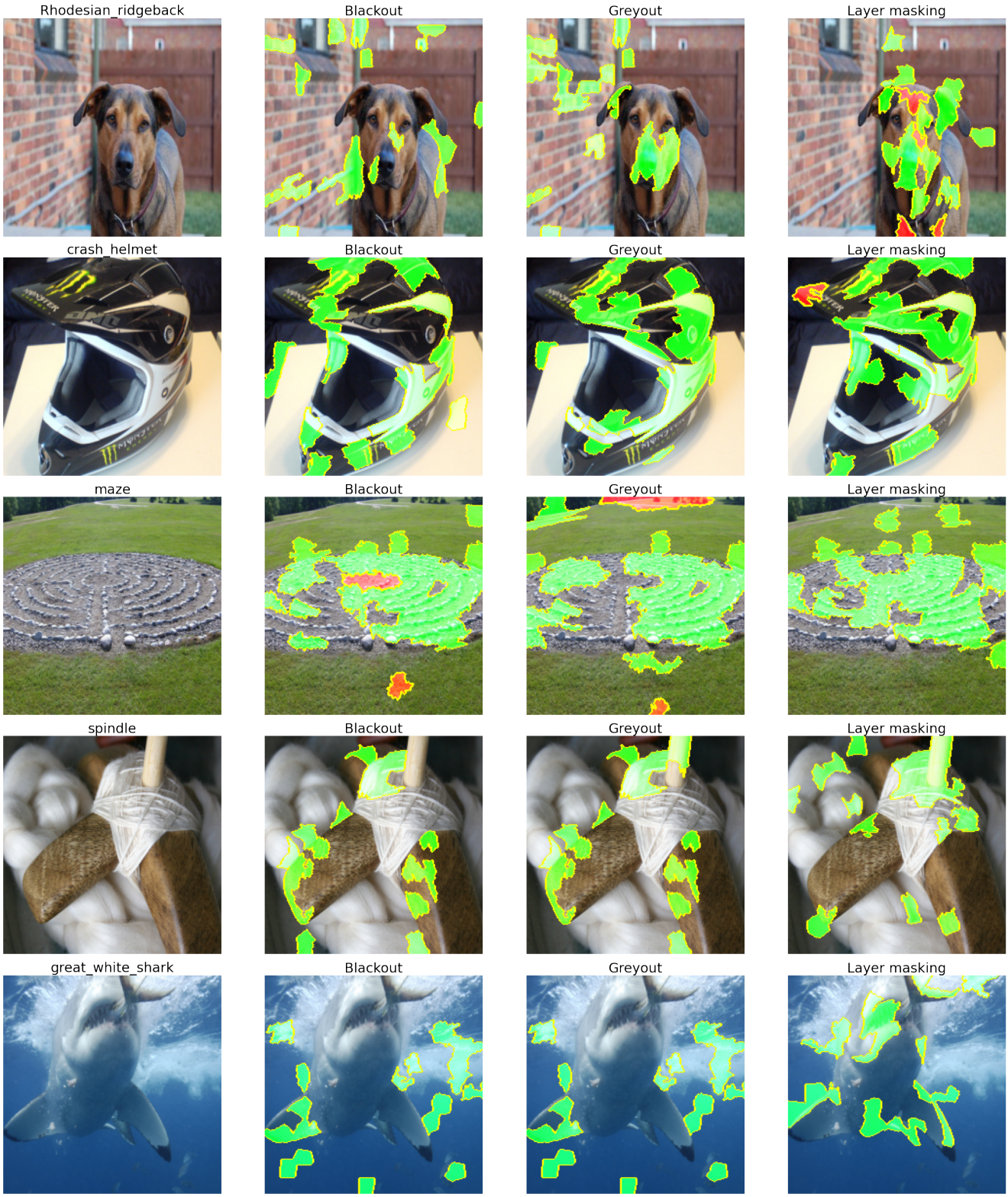


Figure 6. LIME scores using SLIC segmentation (5 samples). Top 15 segments are highlighted

Figure 7. LIME scores using 16×16 segmentation (5 samples). Top 20 segments are highlighted

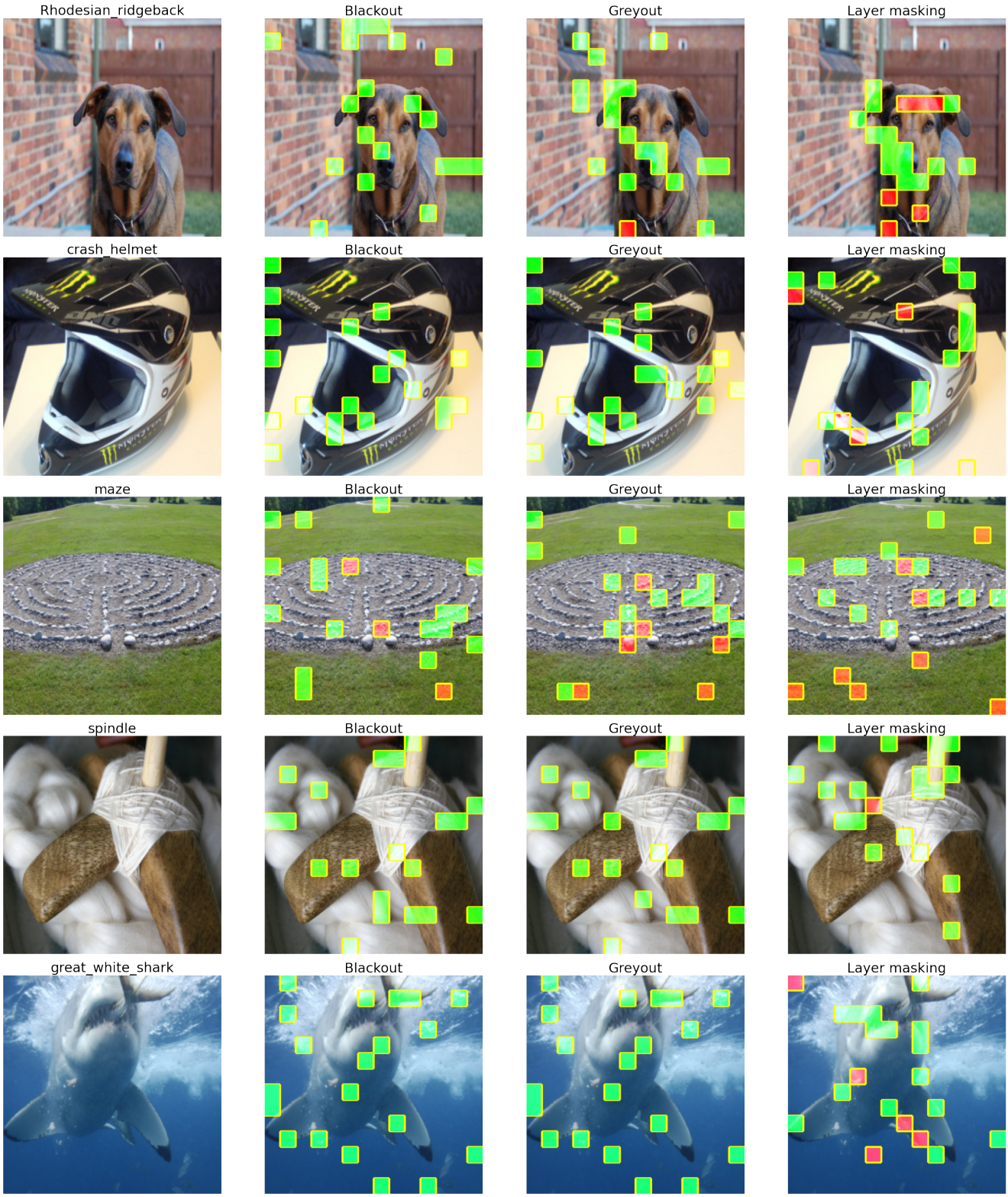


Figure 8. LIME scores using 16×16 segmentation (5 samples). Top 20 segments are highlighted

6.2. Quantitative

We compute the same metrics (Top-20 ablation accuracy, Alignment score, Top-20 Jaccard similarity) for different architectures and segmentation algorithms. The metrics are computed as follows:

1. **Top-k ablation accuracy**: As described in Strumfels et al, we choose the k most important segments according to the explanation, remove them by substituting with a missingness approximation (we use grey), and compute the accuracy on the masked images. The more the accuracy drops, the better the explanations. Let $\boldsymbol{m}'$ be a mask such that if pixel (u, v) lies in the top k features, then $\boldsymbol{m}'[u, v] = 1$ otherwise 0. Then, the top-k ablation accuracy is the accuracy when images are masked by $\boldsymbol{m}'$ using a missingness approximation t (we use grey):

$$\mathbb{E}_{(x,y)\sim D}[\mathbf{1}[f(x \odot (1 - \boldsymbol{m}') + \boldsymbol{m}' \odot t) = y]]$$

2. **Alignment score**: Given a segmentation mask $\boldsymbol{m} \in [0, 1]^{d\times d}$ for an image of dimension d, we derive the "ground truth" $\boldsymbol{g}$ for the explanation such that $\boldsymbol{g}_i = \sum_{(u,v)\in\text{patch } i} (\boldsymbol{m}[u, v] - m_{avg})$ where m_{avg} is the mean of the segmentation mask. We can then measure how aligned the explanations are with the ground truth by computing the *alignment score*, which is the cosine similarity between g_i and s_i, or

$$\cos(\boldsymbol{g}, \boldsymbol{s}) = \frac{\sum_i g_i s_i}{\sqrt{(\sum_i g_i^2)(\sum_i s_i^2)}}$$

   The alignment score will be 1 if the LIME explanation $\boldsymbol{s}$ is perfectly aligned with $\boldsymbol{g}$, and -1 if it is completely misaligned.

3. **Top-k Jaccard similarity**: Take the top-k most contributing features according to the explanation and compute a mask $\boldsymbol{m}'$ such that if pixel (u, v) lies in the top k features, then $\boldsymbol{m}'[u, v] = 1$ otherwise 0. Then, we compute Jaccard similarity between the segmentation mask $\boldsymbol{m}$ and $\boldsymbol{m}'$ as

$$\text{JaccSim}(\boldsymbol{m}, \boldsymbol{m}') = \frac{\sum_{u,v} \boldsymbol{m}[u, v] \cdot \boldsymbol{m}'[u, v]}{\sum_{u,v} \mathbf{1}[\boldsymbol{m}[u, v] + \boldsymbol{m}'[u, v] > 0]}$$

All of these metrics have their pros and cons. Top k ablation accuracy does not require any supervision or ground truth, but has an undesirable dependence on the missingness approximation used to compute it. The alignment score is designed such that random attributions get a score of 0, but has an undesirable dependence on scale of the explanations. Top k Jaccard similarity is not dependent on the scale, but only the relative ordering of importance of the features, but has a non-zero value for random features. Together, they give a more complete picture of the performance of LIME.

We report our results in Tab. 10. For Wide ResNet-50, AlexNet, SqueezeNet, and DenseNet, the performance of layer masking is the best across all metrics. For EfficientNet and MobileNet-v3, performance of layer masking is worse than greyout in top-k ablation accuracy, but better in alignment score and top-k Jaccard similarity.

	Top-20 ablation accuracy (↓)			Alignment score (↑)			Top-20 Jaccard similarity (↑)		
	Quickshift	16×16	SLIC	Quickshift	16×16	SLIC	Quickshift	16×16	SLIC
Wide ResNet-50									
Blackout	0.668	0.736	0.767	0.128	0.028	0.091	0.177	0.089	0.128
Greyout	0.395	0.642	0.611	0.246	0.084	0.195	0.232	0.113	0.180
Layer masking	0.315	0.392	0.429	0.319	0.252	0.276	0.267	0.188	0.216
AlexNet									
Blackout	0.550	0.506	0.681	0.039	0.006	0.020	0.139	0.085	0.097
Greyout	0.375	0.488	0.531	0.114	0.014	0.074	0.189	0.089	0.124
Layer masking	0.181	0.256	0.331	0.209	0.200	0.187	0.240	0.167	0.188
SqueezeNet									
Blackout	0.479	0.552	0.615	0.058	0.002	0.031	0.154	0.081	0.101
Greyout	0.307	0.547	0.568	0.124	0.015	0.075	0.195	0.087	0.129
Layer masking	0.224	0.234	0.281	0.197	0.194	0.186	0.235	0.167	0.189
DenseNet									
Blackout	0.562	0.745	0.682	0.156	0.029	0.122	0.203	0.089	0.149
Greyout	0.276	0.589	0.495	0.273	0.099	0.234	0.259	0.122	0.196
Layer masking	0.312	0.359	0.500	0.301	0.261	0.290	0.277	0.195	0.220
MobileNet									
Blackout	0.562	0.896	0.719	0.214	0.072	0.173	0.225	0.108	0.168
Greyout	0.365	0.526	0.536	0.237	0.167	0.207	0.231	0.159	0.182
Layer masking	0.547	0.656	0.599	0.258	0.203	0.241	0.249	0.168	0.201
EfficientNet									
Blackout	0.703	0.901	0.771	0.251	0.084	0.231	0.246	0.119	0.199
Greyout	0.500	0.646	0.604	0.236	0.175	0.198	0.244	0.167	0.192
Layer masking	0.661	0.688	0.750	0.291	0.231	0.266	0.268	0.185	0.216

Table 10. Top-20 ablation accuracy, alignment score, and top-20 Jaccard similarity of LIME scores over 200 random images

7. Other interesting properties of layer masking

In this section, we identify some more properties of layer masking that are important for model interpretability.

7.1. Linearity in masking:

Consider a model equipped with a masking technique f_m which acts on an input - mask pair $(\boldsymbol{x}, \boldsymbol{m})$ and returns an output $\boldsymbol{y}$ which depends only on the unmasked parts of the input. Then, we say that the model f_m is linear in masking if $f_m(\boldsymbol{x}, \boldsymbol{m}_1 + \boldsymbol{m}_2) = f_m(\boldsymbol{x}, \boldsymbol{m}_1) + f_m(\boldsymbol{x}, \boldsymbol{m}_2)$ for any two binary masks $\boldsymbol{m}_1, \boldsymbol{m}_2$ such that $\boldsymbol{m}_1 \cdot \boldsymbol{m}_2 = 0$. This property is useful for interpretability methods like LIME which train a linear model on $(\boldsymbol{m}, \boldsymbol{y})$ pairs and use its weights to explain the model prediction. Modern vision models like CNNs and Vision Transformers are non-linear and include cross-interactions between features in $\boldsymbol{m}_1$ and $\boldsymbol{m}_2$. Thus, it is not possible to design a perfectly linear masking technique for these architectures, which means that only approximate linearity is possible. However, we can attempt to design more linear masking methods for each model architecture, and thus obtain more interpretable masking techniques.

We measure linearity by sampling random images from ImageNet and dividing it into N smaller square patches. We can then compute the cosine similarity between $f(\boldsymbol{x})$ and $\sum_{i=1}^{N} f_m(\boldsymbol{x}, \boldsymbol{m}_i)$ where $\boldsymbol{m}_i$ corresponds to patch i (Tab. 11). We find that layer masking is much more linear as compared to greying out or blacking out pixels, and in general, ResNet masking methods are more linear than corresponding methods for ViTs. Because the attention heads in ViTs introduce a lot of cross terms right from the beginning, including cross terms between distant patches, linearity in vision transformer masking is much lower than CNN masking.

We also find that in layer masking, $\mathbb{E}_{\boldsymbol{x}} \|f_m(\boldsymbol{x}, \boldsymbol{m})\|$ scales linearly with $|\boldsymbol{m}|$. We test this by measuring the magnitude of $f_m(\boldsymbol{x}, \boldsymbol{m})$ with $\boldsymbol{m}$ as a mask for square patches of side length n, so that $\|\boldsymbol{m}\| \propto n^2$. We observe in Fig. 10 that layer masking closely tracks the n^2 curve, which implies that $\mathbb{E}_{\boldsymbol{x}} \|f_m(\boldsymbol{x}, \boldsymbol{m})\|$ scales almost linearly with $\|\boldsymbol{m}\|$ for layer masking. However, the magnitude for ViT features remain approximately constant.

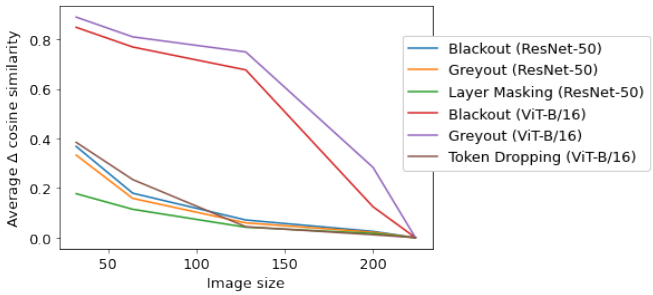


Figure 9. Average difference in cosine similarity vs image size. Since model features of ViTs can be negative unlike ResNet-50, cosine similarity can vary from -1 to 1

Patch size	ResNet-50 Blackout	ResNet-50 Greyout	ResNet-50 **Layer masking**	ViT-B/16 Blackout	ViT-B/16 Greyout	ViT-B/16 Token dropping
112	0.7975	0.8284	**0.9485**	0.6707	0.7063	0.7043
56	0.5124	0.5842	**0.8310**	0.2202	0.2506	0.1929
32	0.4282	0.4878	**0.7094**	0.1377	0.1365	0.1426
16	0.3848	0.4371	**0.6490**	0.0912	0.0876	0.0877

Table 11. Average cosine similarity between image features and their linear approximation

7.2. Avoidance of output collapse:

As the fraction of masked input approaches 1, it is desirable to avoid the model output collapsing to the same vector and thus not being sensitive enough to the unmasked features. To test for this, we take two random images $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ of size 224×224 and compute the cosine similarity between their model features, $c = \cos(f(\boldsymbol{x}_1), f(\boldsymbol{x}_2))$. Then, these images are resized to a smaller size n, and padded with zeros to recover the original size . We now have images $\boldsymbol{x}_{1,n}$ and $\boldsymbol{x}_{2,n}$ of size 224×224 and a mask of the same shape $\boldsymbol{m}_n$ which is 1 for a region of size $n \times n$ and 0 elsewhere. We then measure the cosine similarity between $\boldsymbol{x}_{1,n}$ and $\boldsymbol{x}_{2,n}$ as $c_n = \cos(f_m(\boldsymbol{x}_{1,n}, \boldsymbol{m}_n), f_m(\boldsymbol{x}_{2,n}, \boldsymbol{m}_n))$ and plot $\mathbb{E}_{\boldsymbol{x}_1,\boldsymbol{x}_2}[c_n - c]$ as function of n in Fig. 9. We clearly see that as the image size is decreased, the cosine similarity changes much more for greyout or blackout as compared to layer masking for ResNet-50 or token dropping for ViTs.

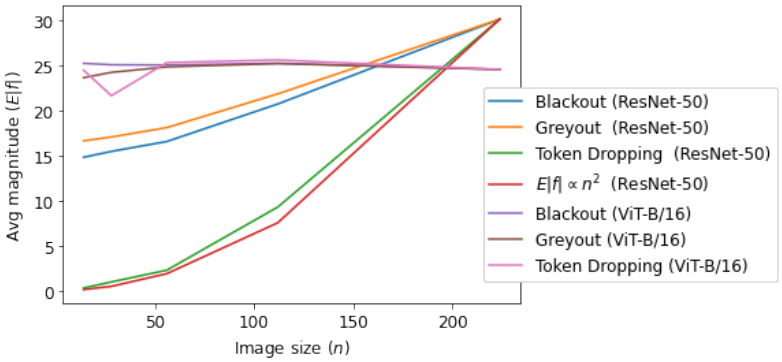
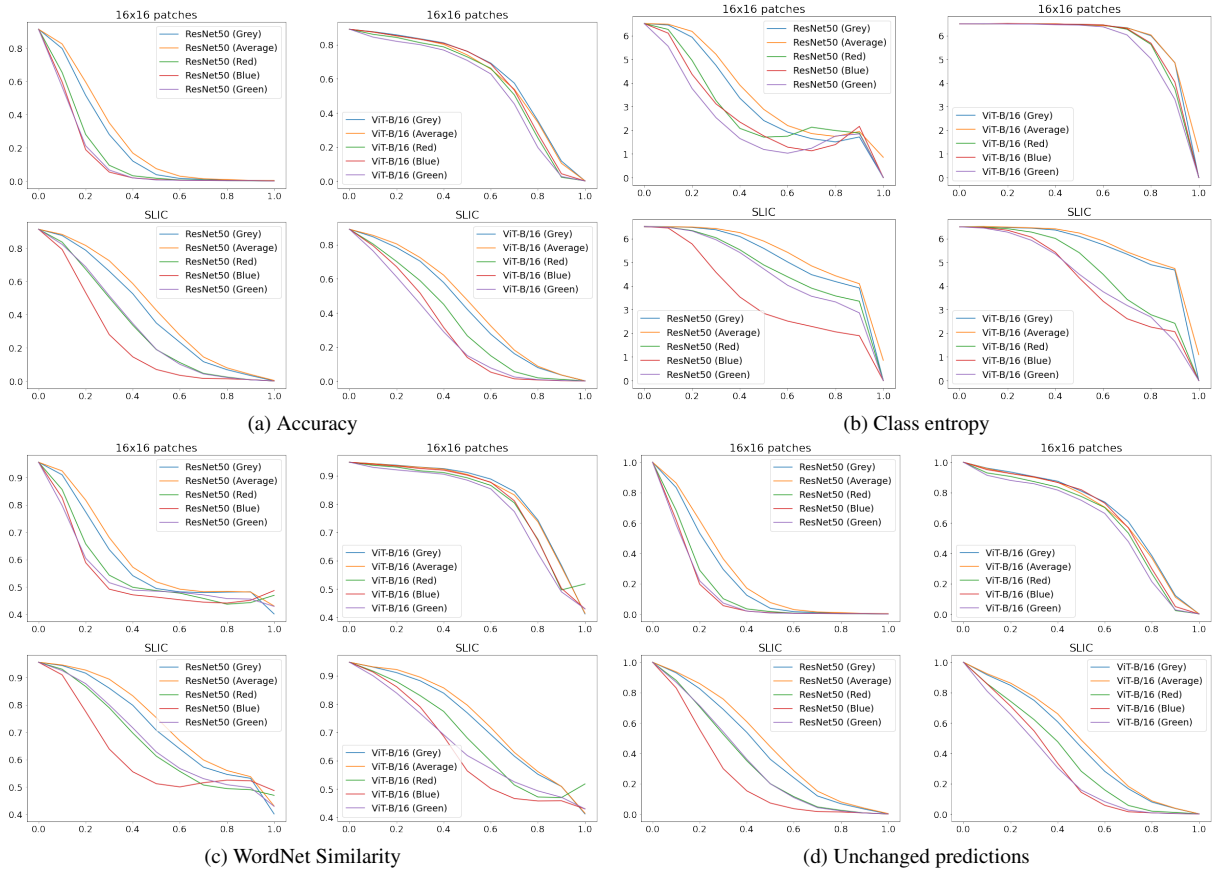


Figure 10. Mean magnitude of output feature vectors vs image size

8. Other baseline colors

We also repeat the experiments in Section 4.1 with other baseline colors like red, blue and green. Grey baseline is included for reference. Segments are removed out in random order. We find that the best constant baseline is either greyout or average color of that image for both ResNet-50 and transformers.



(a) Accuracy (b) Class entropy

(c) WordNet Similarity (d) Unchanged predictions

Figure 11. Changes in model prediction for different model architectures