Appendix for Markpainting: Adversarial Machine Learning meets Inpainting

David Khachaturov \(^1\)  Ilia Shumailov \(^{1,2}\)  Yiren Zhao \(^1\)  Nicolas Papernot \(^2\)  Ross Anderson \(^1\)

1. Applying Markpainting to a Collection of Models

As illustrated in the markpainting algorithm, we find that we are able to markpaint a collection of models \(\Theta\) simultaneously using a single input image \(I\) as detailed in Algorithm 1 in the paper. This means a single adversarial image serves as an input to all considered models. An example application of this in a white-box setting is shown in Figure 1, where the same sample is perturbed to be producing a different face after inpainting by a 6 different inpainters. Figure 2 shows the benign samples from the inpainters, these are the infilling effect if there are no markpainting generated noises. The inpainter will naively fill the masked region without facial details, but the generated markpainted example can influence this filling to provide facial details of Obama.

Table 1 shows the details of how this technique works with different inpainters. In Table 1, the adversarial samples are generated using all models and evaluated on each model individually. The table reports the loss, L2 norms, peak signal to noise ratio (PSNR) and structural similarity index measure (SSIM) for accessing the inpainted image quality.

2. Targeted Application of Inpainting

Table 2 shows the results of the markpainting technique on different inpainter models and it is an extended version of Table 2 in the main paper. The markpainting technique is launched at each individual inpainter and evaluated on that inpainter with different perturbation budgets.

The table reports the loss, L2 norms, peak signal to noise ratio (PSNR) and structural index similarity (SSIM) for accessing the inpainted image quality.

3. Transferability of Markpainting

In Table 3, we assess the transferability of markpainting.

\(^{1}\)Equal contribution  \(^{1}\)Computer Laboratory, University of Cambridge  \(^{2}\)University of Toronto and Vector Institute. Correspondence to: Ilia Shumailov <ilia.shumailov@cl.cam.ac.uk>.

Figure 1. Markpainting Vladimir Putin’s face with former President Obama’s. The mix of colors in the image makes it significantly easier to attack. Attack performed over 500 iterations, with \(\epsilon = 0.1\). The original photo with Putin and Trump is taken from The Guardian. Photograph of Obama taken from Acclaim Images.

Figure 2. Benign infilling of the example image in Figure 1.
4. Parameter Choices

In the paper, we provided a visualization of having an increasing $\epsilon$ budgets in Figure 4. This term controls a loss trade-off between the network loss and the L2 loss. As we can see, in Figure 3, when the $\alpha$ value increases, the marked painted image gets closer to the target. We also show the original benign inpainting results in Figure 2 as a baseline for comparisons. It is worth to mention that the baseline simply fills the face with surrounding colors.

We also further study the effect of $\epsilon$ of the markpainting technique. In the evaluation, the effect of different epsilons are shown for the RN inpainter, we further illustrate the effect of epsilons on other inpainters (RFR and CRFILL), and they are Figure 7 and Figure 8 respectively.

5. Evaluation Details

The places_subset16 dataset that we used to evaluate our proposed method on – a series of 16 randomly-selected images from the Places2 dataset (Zhou et al., 2017) – is visualized in Figure 9.

We understand that it is of interest to readers to be able to visualize the numeric loss values we quote in our results. In Figure 4, we present visual examples of how numeric loss relates to the markpainted results for complex targets; and in Figure 5 we do the same for a solid-color target.

Figure 4. Correspondence between numeric loss to target image and the obtained markpainted results for a complex target.
Appendix for Markpainting: Adversarial Machine Learning meets Inpainting

(a) Applied to the RFR inpainter model.

(b) Applied to the EDGE CONNECT inpainter model.

Figure 5. Correspondence between numeric loss to target image and the obtained markpainted results for a solid-color target.

6. Watermark Removal

We show more results on the watermark removal with $\epsilon = 0.15$. The objective is to build an image that is resistant to watermark removals using markpainting. Figure 6 shows that markpainted images (top two rows) are in general more robust to different inpainters trying to fill the watermark.

References


Figure 7. Inpainting with increasing perturbation epsilon budget. Top row is the adversarial images generated using markpainting, and second row is the inpainted results of these adversarial images. We target the CRFILL inpainter with 500 iterations and a step size of $\epsilon/100$. Note that this example is really hard, because we are filling a black and white image with color.

Figure 8. Inpainting with increasing perturbation epsilon budget. Top row is the adversarial images generated using markpainting, and second row is the inpainted results of these adversarial images. We target the RFR inpainter with 500 iterations and a step size of $\epsilon/100$. Note that this example is really hard, because we are filling a black and white image with color.
Figure 9. The `places_subset16` dataset used for evaluation. A subset of Places2 (Zhou et al., 2017).
Appendix for Markpainting: Adversarial Machine Learning meets Inpainting

Table 1. Demonstration of a multi-model attack. Note that this produces just a single adversarial image, which all models subsequently adapt.

<table>
<thead>
<tr>
<th>Distance to</th>
<th>$\epsilon$</th>
<th>Original PSNR</th>
<th>Original SSIM</th>
<th>Adversarial PSNR</th>
<th>Adversarial SSIM</th>
<th>BGen</th>
<th>BGen</th>
<th>BGen</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0</td>
<td>0.00625</td>
<td>0.00000</td>
<td>4.09 (0.257)</td>
<td>0.00000</td>
<td>4.09 (0.257)</td>
<td>1.65</td>
<td>0.123</td>
<td>6.43</td>
</tr>
<tr>
<td>0.00625</td>
<td>0.00000</td>
<td>1.65 (0.257)</td>
<td>4.09 (0.257)</td>
<td>1.65 (0.257)</td>
<td>4.09 (0.257)</td>
<td>1.65</td>
<td>0.123</td>
<td>6.43</td>
</tr>
<tr>
<td>0.00125</td>
<td>0.00000</td>
<td>1.65 (0.257)</td>
<td>4.09 (0.257)</td>
<td>1.65 (0.257)</td>
<td>4.09 (0.257)</td>
<td>1.65</td>
<td>0.123</td>
<td>6.43</td>
</tr>
<tr>
<td>0.000625</td>
<td>0.00000</td>
<td>1.65 (0.257)</td>
<td>4.09 (0.257)</td>
<td>1.65 (0.257)</td>
<td>4.09 (0.257)</td>
<td>1.65</td>
<td>0.123</td>
<td>6.43</td>
</tr>
<tr>
<td>0.0003125</td>
<td>0.00000</td>
<td>1.65 (0.257)</td>
<td>4.09 (0.257)</td>
<td>1.65 (0.257)</td>
<td>4.09 (0.257)</td>
<td>1.65</td>
<td>0.123</td>
<td>6.43</td>
</tr>
<tr>
<td>0.00015625</td>
<td>0.00000</td>
<td>1.65 (0.257)</td>
<td>4.09 (0.257)</td>
<td>1.65 (0.257)</td>
<td>4.09 (0.257)</td>
<td>1.65</td>
<td>0.123</td>
<td>6.43</td>
</tr>
<tr>
<td>0.000078125</td>
<td>0.00000</td>
<td>1.65 (0.257)</td>
<td>4.09 (0.257)</td>
<td>1.65 (0.257)</td>
<td>4.09 (0.257)</td>
<td>1.65</td>
<td>0.123</td>
<td>6.43</td>
</tr>
<tr>
<td>0.0000390625</td>
<td>0.00000</td>
<td>1.65 (0.257)</td>
<td>4.09 (0.257)</td>
<td>1.65 (0.257)</td>
<td>4.09 (0.257)</td>
<td>1.65</td>
<td>0.123</td>
<td>6.43</td>
</tr>
</tbody>
</table>

Table 2. Impact of markpainting attack on each model individually: the model in each row is attacked, and the results are presented from evaluation on that same model. This example input outputs the same target/mask combination as Table 3. The results are after 100 iterations with a step size of $\epsilon$ and are presented in the form $\mu \pm \sigma$. 
Appendix for Markpainting: Adversarial Machine Learning meets Inpainting

Table 3. Key attribution based on the adversarial sample produced. RN was attacked. The results are after 100 iterations with a step size of \(\epsilon_{50}\). The results are presented in the form of \(\frac{\mu}{\pm \sigma}\).