Adversarial Image Perturbation for Privacy Protection
A Game Theory Perspective
Supplementary Materials

1. Contents

The supplementary materials contain auxiliary experiments for the empirical analyses in the main paper. In particular, we include:

- Score loss for adversarial image perturbation (AIP).
- AIP performance at different L_2 norms.
- Experiments for the non-GoogleNet architectures.
- More qualitative results.

As in the main paper, we mark the optimal entry in each column (row) for the user (recogniser) with orange (blue).

2. Score Loss for AIPs

In the main paper, we have reviewed variants of AIPs according to the loss functions and the optimisation algorithms. Algorithms `FGV`, `FGS`, `BI`, and `GA` use the softmax-log loss $-\log \hat{f}^y$. The DeepFool (`DF`) and our `GAMAN` variants use the difference of two scores (*e.g.* $f^{y^\star} - f^y$). This section includes an auxiliary analysis for the effect of the loss type: softmax-log loss $-\log \hat{f}^y$ versus score loss $-f^y$. We denote the score loss analogues with the suffix `-S` (*e.g.* `FGS-S`). We also include `FGMAN` (Fast Gradient – Maximal Among Non-GT), the single iteration analogue of `GAMAN`, for completeness. See table 1 for a summary.

The corresponding empirical performances are shown in table 2 and 4. Since single-iteration AIPs are significantly outperformed by the multi-iteration AIPs, we have focused on the latter in the main paper, and so do we here. In table 2, we observe that the choice of the loss function does not make much difference. Table 4 further supports this view against image processing techniques, although the softmax-log loss does perform marginally better.

3. AIP Performance at Different L_2 Norms

In the main paper, we have used the L_2 norm constraint $\epsilon = 1000$ as the default choice. In this section, we examine the behaviour of AIP performance at varying ϵ values.

Variants	Loss $\mathcal{L}$	Stopping condition	Step size
`FGS`[1]	$-\log \hat{f}^y$	1 iteration	Fixed
`FGV`[4]	$-\log \hat{f}^y$	1 iteration	Fixed
`FGS-S`	$-f^y$	1 iteration	Fixed
`FGV-S`	$-f^y$	1 iteration	Fixed
`FGMAN`	$f^{y^\star} - f^y$	1 iteration	Fixed
`BI`[2]	$-\log \hat{f}^y$	K iterations	Fixed
`GA`	$-\log \hat{f}^y$	K iterations	Fixed
`BI-S`	$-f^y$	K iterations	Fixed
`GA-S`	$-f^y$	K iterations	Fixed
`DF`[3]	$f^{y^c} - f^y$	K it.$\vee$ fooled	Adaptive
`GAMAN`	$f^{y^\star} - f^y$	K iterations	Fixed

Table 1: Extended version of table **??** in the main paper; additional methods are denoted as gray cells. $f^{y'}$ is the model score for class y', and $\hat{f}$ denotes the softmax output of f. y is the ground truth label, and $y^\star$ is the most likely label among wrong ones. y^c is the label with the closest linearised decision boundary. $\tilde{y}$ is the least likely label.

See figure 1 for the plot. The performances are post-Proc (§5.3 in the main paper). We fix the step size to $\gamma = 10^4$ (5×10^3 for `GAMAN`), and the maximal number of iterations to $K = 100$; we choose the norm constraint ϵ from $\{100, 200, 500, 1000, 2000\}$. The norm of the resulting AIP is upper bounded by ϵ, but may not necessarily be exactly ϵ. The average norm across the test set is plotted.

We observe that the AIP variants are much more effective than Noise, Blur, or Eye Bar, achieving the same degree of obfuscation at $1 \sim 2$ orders of magnitude smaller perturbations. At the same norm level, the multi-iteration variants (`BI`,`GA`) are more effective than the single-iteration analogues (`FGS`,`FGV`). Taking gradient signs decreases the obfuscation performance at small L_2 norms (≤ 1000), but they converge to a similar performance at $\epsilon = 2000$. Deep-

	Perturbation	AlexNet	VGG	Google	ResNet
	None	83.8	86.1	87.8	91.1
Image Proc.	Noise	≥83	≥85	≥87	≥90
	Blur	≥82	≥85	≥86	≥90
	Eye Bar	≥81	≥84	≥84	≥87
1-Iter. AIP	FGS[1]	23.6	16.0	5.9	20.2
	FGV[4]	13.3	11.5	4.6	20.0
	FGS-S	27.8	6.2	1.0	4.3
	FGV-S	21.0	5.5	3.5	8.0
	FGMAN	4.4	3.9	2.8	11.5
K-Iter. AIP	BI[2]	1.2	0.5	0.0	0.0
	GA	0.2	0.0	0.0	0.0
	BI-S	1.2	0.3	0.0	0.0
	GA-S	0.2	0.0	0.0	0.0
	DF[3]	0.0	0.0	0.0	0.0
	GAMAN	0.0	0.0	0.0	0.0

Table 2: Extended version of table **??** in the main paper; new entries are denoted as gray cells. Recognition rates after image perturbation. In all methods, the perturbation is restricted to $||\cdot||_2 \leq 1000$. For the baseline image processing perturbations, we only report lower bounds (denoted $\geq \cdot$).

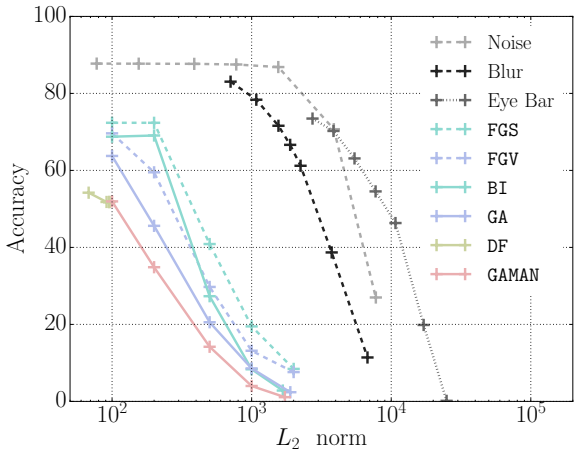

Figure 1: GoogleNet accuracy after various perturbations methods at different L_2 norms. All results are after Proc.

Fool (DF) outputs have small norms ≤ 100 due to early stopping. Our variant GAMAN performs best across all norm levels, achieving nearly zero recognition at $\epsilon = 2000$.

4. Non-GoogleNet Experiments

In the main paper, we have focused on the GoogleNet results for the AIP robustness analysis and the game theoretic studies (table 3 and 4 in the main paper). We extend the experiments to AlexNet, VGG, and ResNet152.

4.1. Robustness Analysis

See table 4 for the robustness analyses for all four networks. We confirm here again that GAMAN shows overall best robustness, across image processing techniques (Proc, T, N, B, C, and TNBC), across architectures. For AlexNet and ResNet, cropping (C) is the most powerful neutralisation, while for VGG and GoogleNet blurring (B) is. We observe that the effects are particularly strong for ResNet; C boosts the performance from 0.0 to 31.8 against GAMAN.

4.2. Game Analysis for Various Networks

See table 5 for the payoff tables for all four networks. We summarise the optimal user strategy $\theta^{u\star}$ and the corresponding guarantee on the recognition rate in table 3. Note that against all but AlexNet architecture, the optimal strategy $\theta^{u\star}$ is given as a mixture of /B and /TNBC.

Network	Optimal Strategy $\theta^{u\star}$	Bound on Rec. Rate
AlexNet	(/B : 100%)	≤6.4
VGG	(/B : 86%, /TNBC : 14%)	≤4.9
GoogleNet	(/B : 61%, /TNBC : 39%)	≤7.3
ResNet	(/B : 31%, /TNBC : 69%)	≤8.5

Table 3: Optimal strategies and the corresponding guaranteed upper bounds on the recognition rate for different networks. We write $\leq \cdot$ to denote the upper bound.

5. Additional Qualitative Results

We include more qualitative results (equivalent to figure 3 in the main paper). See figures 2, 3, 4, 5.

References

[1] I. J. Goodfellow, J. Shlens, and C. Szegedy. Explaining and harnessing adversarial examples. In *ICLR*, 2015.

[2] A. Kurakin, I. Goodfellow, and S. Bengio. Adversarial examples in the physical world. *CoRR,abs/1607.02533*, 2016.

[3] S.-M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard. Deepfool: a simple and accurate method to fool deep neural networks. In *CVPR*, 2016.

[4] A. Rozsa, E. M. Rudd, and T. E. Boult. Adversarial diversity and hard positive generation. In *CVPRW*, 2016.

AlexNet

Perturb	∅	Proc	T	N	B	C	TNBC
None	83.8	83.8	83.7	77.8	78.7	80.1	83.9
BI[2]	1.2	10.0	29.7	20.8	26.6	34.3	23.3
GA	0.2	4.8	13.6	11.6	17.7	17.8	12.2
BI-S	1.2	10.1	31.2	21.0	27.2	35.7	23.3
GA-S	0.2	5.0	15.4	12.6	19.0	19.3	12.8
DF[3]	0.0	62.1	76.5	68.5	69.4	75.0	74.7
GAMAN	0.0	1.4	6.4	9.2	13.5	12.3	5.6

VGG

Perturb	∅	Proc	T	N	B	C	TNBC
None	86.1	86.1	84.8	77.2	81.5	84.1	85.8
BI[2]	0.5	6.8	11.1	18.1	23.2	16.8	14.4
GA	0.0	4.2	5.5	11.2	17.2	10.2	8.2
BI-S	0.3	7.1	11.2	19.2	23.8	17.3	14.3
GA-S	0.0	4.8	5.9	11.9	18.6	11.3	8.8
DF[3]	0.0	53.3	66.3	65.9	69.4	69.2	71.4
GAMAN	0.0	1.6	2.1	8.5	11.8	5.6	3.5

GoogleNet

Perturb	∅	Proc	T	N	B	C	TNBC
None	87.8	87.8	87.6	64.0	81.2	85.4	87.3
BI[2]	0.0	8.3	15.8	16.8	28.6	27.4	17.6
GA	0.0	8.6	13.2	14.1	28.4	23.7	16.4
BI-S	0.0	8.8	17.2	17.7	29.3	28.8	18.8
GA-S	0.0	9.1	14.9	15.2	29.3	25.5	18.0
DF[3]	0.0	51.8	75.6	56.5	72.5	76.9	75.5
GAMAN	0.0	4.0	6.6	15.0	22.2	16.7	9.9

ResNet

Perturb	∅	Proc	T	N	B	C	TNBC
None	91.1	91.1	90.6	72.0	87.2	89.3	90.8
BI[2]	0.0	10.9	36.8	24.8	32.8	45.3	26.3
GA	0.0	15.2	37.3	24.4	36.9	43.7	28.9
BI-S	0.0	13.0	43.4	27.4	35.8	51.5	29.9
GA-S	0.0	19.4	45.0	27.1	40.2	50.3	33.3
DF[3]	0.0	52.9	83.1	65.0	76.8	84.2	80.9
GAMAN	0.0	7.3	23.4	23.3	28.2	31.8	18.4

Table 4: Extended version of table **??** in the main paper for all four network architectures; additional AIP entries are denoted as gray cells. Robustness analysis of AIPs for various convnet architectures. AIPs are restricted to $||\cdot||_2 \leq 1000$. $(\mathsf{T}, \mathsf{N}, \mathsf{B}, \mathsf{C}) =$ (Translate, Noise, Blur, Crop).

AlexNet

	Recogniser Θ^r					
User Θ^u	Proc	T	N	B	C	TNBC
GAMAN	1.4	6.4	9.2	13.5	12.3	5.6
/T	0.9	0.8	6.2	10.5	2.7	2.2
/N	1.2	4.2	4.8	11.7	9.5	3.9
/B	0.8	3.5	6.3	6.4	6.0	2.6
/C	2.4	2.5	9.2	13.1	1.3	3.4
/TNBC	0.6	1.2	4.5	7.8	2.9	1.9

VGG

	Recogniser Θ^r					
User Θ^u	Proc	T	N	B	C	TNBC
GAMAN	1.6	2.1	8.5	11.8	5.6	3.5
/T	1.5	1.2	8.1	12.3	3.2	2.8
/N	2.0	2.5	3.9	12.6	6.7	3.9
/B	0.3	0.7	5.0	4.5	2.2	1.2
/C	2.0	1.6	9.5	14.0	1.9	3.1
/TNBC	0.6	0.7	4.3	7.3	2.3	1.4

GoogleNet

	Recogniser Θ^r					
User Θ^u	Proc	T	N	B	C	TNBC
GAMAN	4.0	6.6	15.0	22.2	16.7	9.9
/T	2.5	2.3	11.6	18.5	7.2	4.9
/N	5.8	7.6	4.6	23.6	16.6	9.1
/B	0.4	0.8	8.6	5.8	3.1	1.4
/C	2.6	2.2	11.8	18.1	3.4	4.3
/TNBC	0.7	0.9	5.2	9.5	3.2	2.0

ResNet

	Recogniser Θ^r					
User Θ^u	Proc	T	N	B	C	TNBC
GAMAN	7.3	23.4	23.3	28.2	31.8	18.4
/T	2.9	2.8	16.6	19.0	5.4	5.8
/N	5.3	12.9	4.2	23.5	20.1	10.2
/B	0.6	3.1	13.0	6.8	5.3	2.4
/C	3.5	3.1	17.0	18.8	3.2	5.4
/TNBC	0.7	1.2	6.5	9.3	2.9	2.3

Table 5: Extended version of table **??** in the main paper for all four network architectures. Recogniser's payoff table p_{ij}, $i \in \Theta^u$, $j \in \Theta^r$, for various convnet architectures. The user's payoff is given by $100 - p_{ij}$.

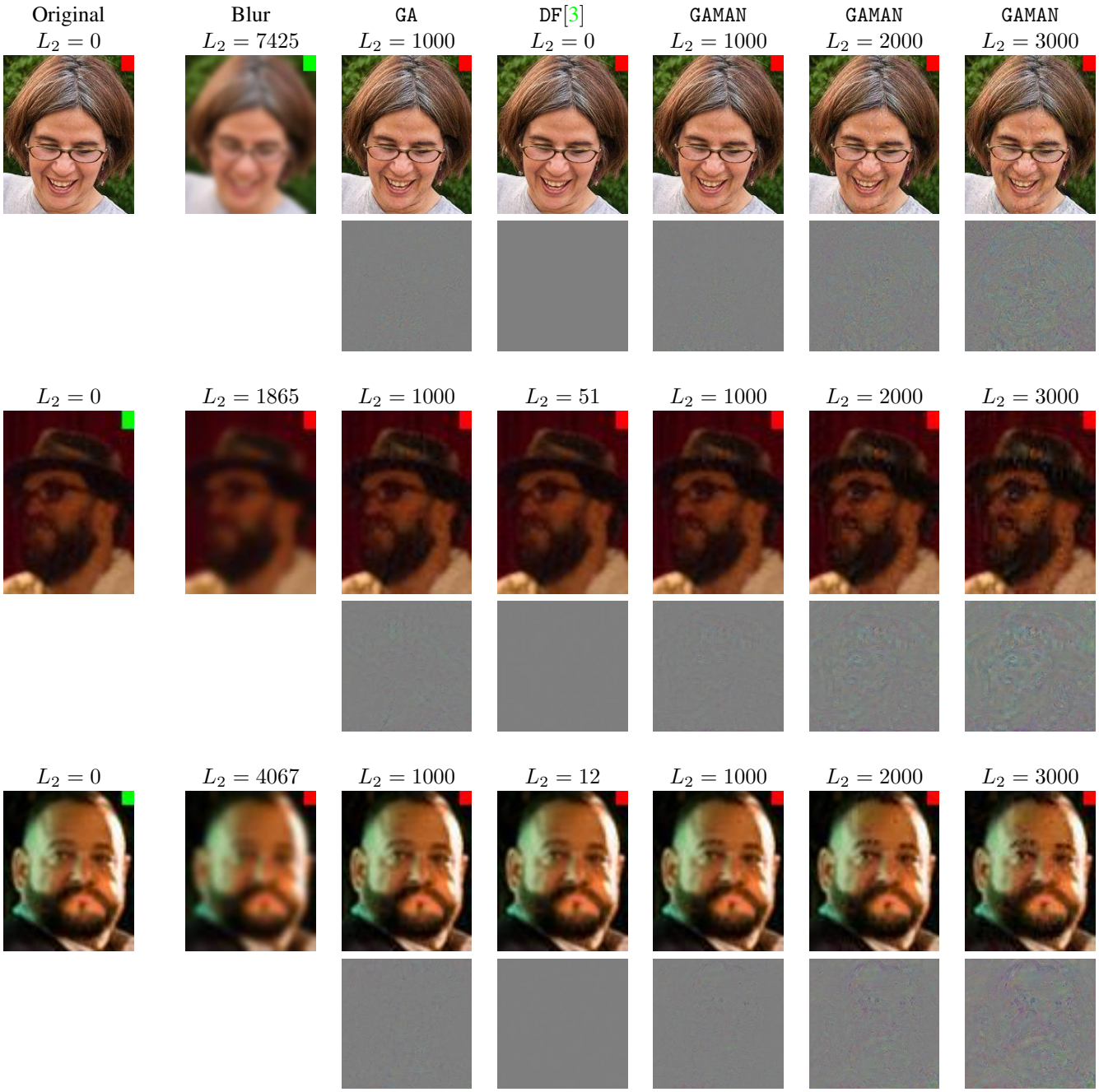

Figure 2: Randomly chosen perturbed images after Proc and the corresponding GoogleNet predictions (green for correct, red for wrong). Perturbations are visualised with gray background. GA and GAMAN reliably confuse the classifier at almost no cost on the aesthetics. As the L_2 norm increases, artifacts become more visible. Perturbations may be too small to be visible when printed; zoom in in electronic version for better visibility.

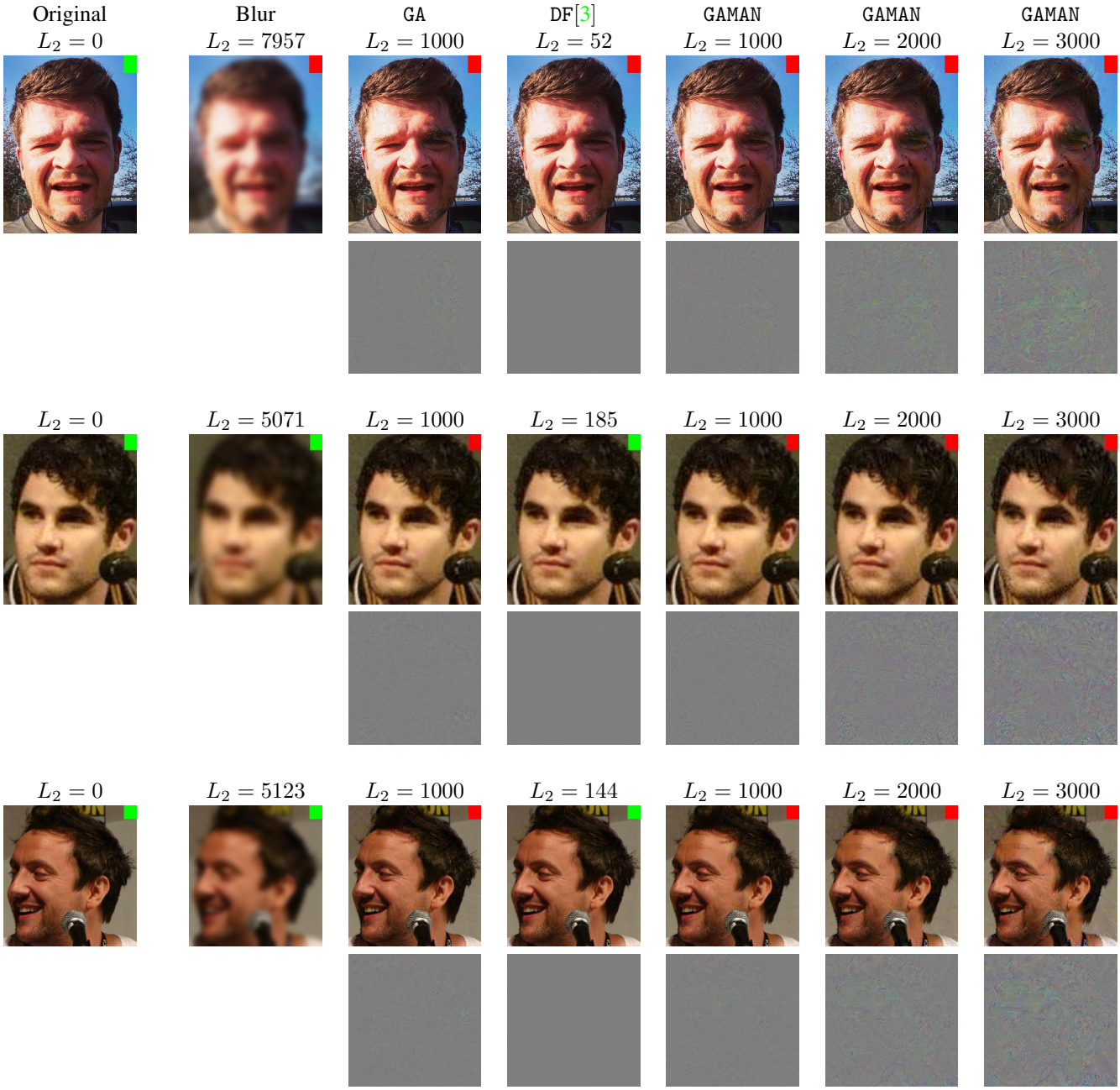

Figure 3: Randomly chosen perturbed images after Proc and the corresponding GoogleNet predictions (green for correct, red for wrong). Perturbations are visualised with gray background. GA and GAMAN reliably confuse the classifier at almost no cost on the aesthetics. As the L_2 norm increases, artifacts become more visible. Perturbations may be too small to be visible when printed; zoom in in electronic version for better visibility.

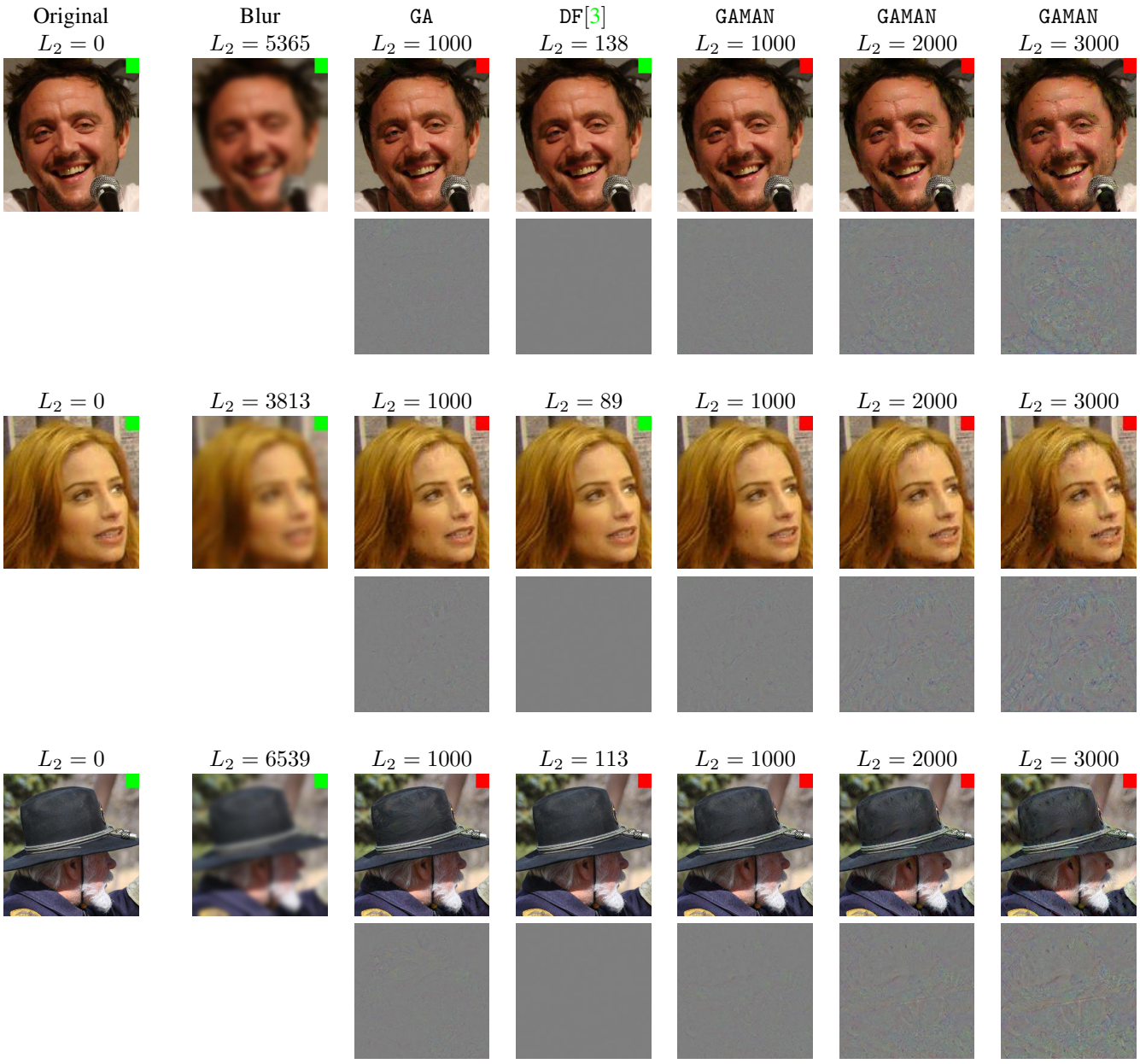

Figure 4: Randomly chosen perturbed images after Proc and the corresponding GoogleNet predictions (green for correct, red for wrong). Perturbations are visualised with gray background. GA and GAMAN reliably confuse the classifier at almost no cost on the aesthetics. As the L_2 norm increases, artifacts become more visible. Perturbations may be too small to be visible when printed; zoom in in electronic version for better visibility.

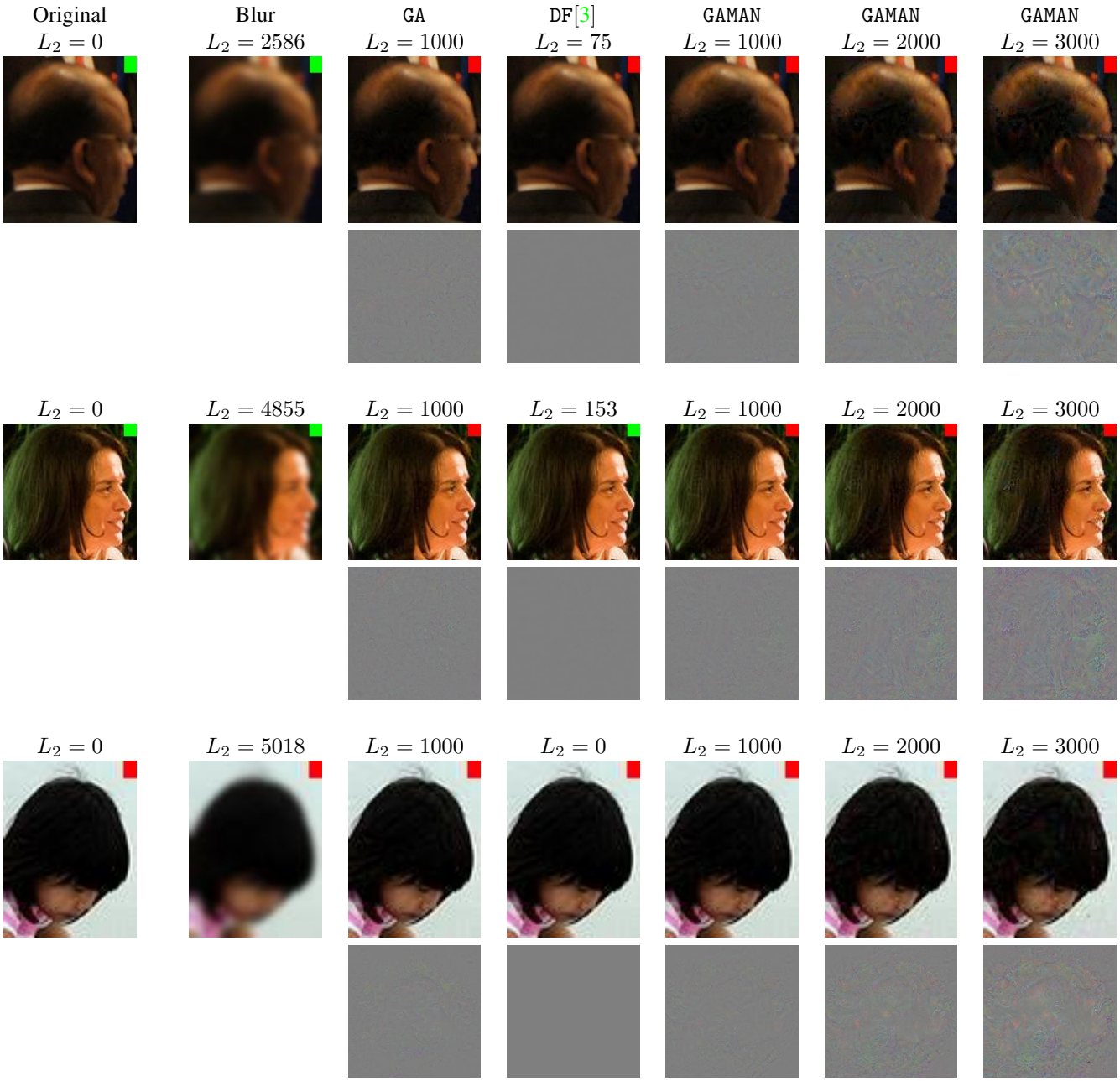

Figure 5: Randomly chosen perturbed images after Proc and the corresponding GoogleNet predictions (green for correct, red for wrong). Perturbations are visualised with gray background. GA and GAMAN reliably confuse the classifier at almost no cost on the aesthetics. As the L_2 norm increases, artifacts become more visible. Perturbations may be too small to be visible when printed; zoom in in electronic version for better visibility.