A Proofs

A.1 Gradient Projection

Proof. The primal problem Equation (4) could be rewritten as

$$\begin{aligned} \arg\min_{\boldsymbol{g}} \quad & f(\boldsymbol{g}) := \frac{1}{2}\boldsymbol{g}_o^\top \boldsymbol{g}_o - \boldsymbol{g}_o^\top \boldsymbol{g} + \frac{1}{2}\boldsymbol{g}^\top \boldsymbol{g}, \\ s.t. \quad & \boldsymbol{g}^\top \boldsymbol{g}_f \leq 0, \end{aligned} \tag{6}$$

where $\boldsymbol{g}_o^\top \boldsymbol{g}_o$ is a constant, and we can remove this constant term. Then we have the Lagrange dual function as

$$L(\boldsymbol{g}, v) = -\boldsymbol{g}_o^\top \boldsymbol{g} + \frac{1}{2}\boldsymbol{g}^\top \boldsymbol{g} + v(\boldsymbol{g}^\top \boldsymbol{g}_f), \tag{7}$$

where v is the Lagrange multiplier and $v \geq 0$. Thus, we have the equivalence problem to Equation (6):

$$\min_{\boldsymbol{g}} f(\boldsymbol{g}) = \inf_{\boldsymbol{g}} \sup_{v} L(\boldsymbol{g}, v) \tag{8}$$

We define the dual problem as $h(v) = \inf_{\boldsymbol{g}} L(\boldsymbol{g}, v)$, and the solution to the dual problem is obtained via $h^* = \sup_v h(v)$.

Lemma 1. *If the primal problem has the optimal solution f^* and its dual problem has the optimal solution h^*, then $h^* = \sup_v \inf_{\boldsymbol{g}} L(\boldsymbol{g}, v) \leq \inf_{\boldsymbol{g}} \sup_v L(\boldsymbol{g}, v) = f^*$.*

As such, instead of directly solving Equation (6) whose computational complexity is based on the number of parameters in the network, we attempt to solve its dual problem $h(v)$. First, to find the minimum of the Lagrange dual function w.r.t. $\boldsymbol{g}$, Let $\nabla_{\boldsymbol{g}} L(\boldsymbol{g}, v) = -\boldsymbol{g}_o + \boldsymbol{g} + v\boldsymbol{g}_f \equiv 0$, we can get $\boldsymbol{g} = \boldsymbol{g}_o - v\boldsymbol{g}_f$. Then substitute $\boldsymbol{g}$ back into the Lagrange dual function and we can have

$$\begin{aligned} L(v) &= -\boldsymbol{g}_o^\top (\boldsymbol{g}_o - v\boldsymbol{g}_f) + \frac{1}{2}(\boldsymbol{g}_o - v\boldsymbol{g}_f)^\top (\boldsymbol{g}_o - v\boldsymbol{g}_f) - v(\boldsymbol{g}_o - v\boldsymbol{g}_f)^\top \boldsymbol{g}_f, \\ &= -\frac{1}{2}\boldsymbol{g}_o^\top \boldsymbol{g}_o - v(\boldsymbol{g}_o^\top \boldsymbol{g}_f) + \frac{1}{2}v^2(\boldsymbol{g}_f^\top \boldsymbol{g}_f), \end{aligned} \tag{9}$$

where $\boldsymbol{g}_o^\top \boldsymbol{g}_o$ is a constant. Therefore, the dual problem could be written as

$$\begin{aligned} \sup_{v} \quad & h(v) := \frac{1}{2}\boldsymbol{g}_f^\top \boldsymbol{g}_f v^2 - \boldsymbol{g}_o^\top \boldsymbol{g}_f v, \\ s.t. \quad & v \geq 0. \end{aligned} \tag{10}$$

which gives Equation (5).

A.2 Connection Sensitivity

To effectively identify salient parameters based on the forgetting data $\mathcal{D}_f$, we adopt the approach proposed in [31] to compute the connection sensitivity of a network:

$$\mathbf{s}_j(\mathcal{D}) \coloneqq \mathbb{E}_{\boldsymbol{x},\boldsymbol{y}\sim\mathcal{D}}\Big[\ell(\boldsymbol{\theta};\boldsymbol{x},\boldsymbol{y}) - \ell((\mathbf{1}_d - \boldsymbol{e}_j)\odot\boldsymbol{\theta};\boldsymbol{x},\boldsymbol{y})\Big] \tag{11}$$

$$\approx \mathbb{E}_{\boldsymbol{x},\boldsymbol{y}\sim\mathcal{D}}\Big[\frac{\partial\ell(\boldsymbol{\theta};\boldsymbol{x},\boldsymbol{y})}{\partial\boldsymbol{\theta}_j}\boldsymbol{\theta}_j\Big]\,. \tag{12}$$

which measures the influence of parameter $j \in \{1,\ldots,d\}$ on a model in terms of the empirical risk for a given dataset $\mathcal{D}$.

Proof. Eq. (11) is approximated using the gradient of the loss w.r.t. that connection [23,31]. $\mathbf{s}_j(\mathcal{D})$ would be viewed to measure the sensitivity of the loss w.r.t. an infinitesimal additive change δ in the parameters $\boldsymbol{\theta}$, thereby probing the importance of the j-th parameter:

$$\begin{aligned}
\mathbf{s}_j(\mathcal{D}) \coloneqq& \mathbb{E}_{\boldsymbol{x},\boldsymbol{y}\sim\mathcal{D}}\Big[\ell(\boldsymbol{\theta};\boldsymbol{x},\boldsymbol{y}) - \ell((\mathbf{1}_d - \boldsymbol{e}_j)\odot\boldsymbol{\theta};\boldsymbol{x},\boldsymbol{y})\Big]\\
\approx& \mathbb{E}\Big[\lim_{\delta\to 0}\frac{\ell(\mathbf{m}\odot\boldsymbol{\theta};\boldsymbol{x},\boldsymbol{y}) - \ell((\mathbf{m}-\delta\boldsymbol{e}_j)\odot\boldsymbol{\theta};\boldsymbol{x},\boldsymbol{y})}{\delta}\Big]\\
=& \mathbb{E}_{\boldsymbol{x},\boldsymbol{y}\sim\mathcal{D}}\Big[\frac{\partial\ell(\mathbf{m}\odot\boldsymbol{\theta};\boldsymbol{x},\boldsymbol{y})}{\partial m_j}\Big|_{\mathbf{m}=\mathbf{1}}\Big]\\
=& \mathbb{E}_{\boldsymbol{x},\boldsymbol{y}\sim\mathcal{D}}\Big[\frac{\partial\ell(\mathbf{m}\odot\boldsymbol{\theta};\boldsymbol{x},\boldsymbol{y})}{\partial(m_j\odot\boldsymbol{\theta}_j)}\Big|_{\mathbf{m}=\mathbf{1}}\odot\boldsymbol{\theta}_j\Big]\\
=& \mathbb{E}_{\boldsymbol{x},\boldsymbol{y}\sim\mathcal{D}}\Big[\frac{\partial\ell(\boldsymbol{\theta};\boldsymbol{x},\boldsymbol{y})}{\partial\boldsymbol{\theta}_j}\boldsymbol{\theta}_j\Big]\,.
\end{aligned}$$

which gives Equation (12).

B Details and Additional results

B.1 Details

Image Classification. We mainly follow the settings in [11] for image classification. For all methods, we employ the SGD optimizer. Batch size is 256 for SVHN, CIFAR-10 and CIFAR-100 experiments. On SVHN, the original model and retrained model are trained over 50 epochs with a cosine-scheduled learning rate initialized at 0.1. On CIFAR-10 and CIFAR-100, the original model and retrained model are trained over 182 and 160 epochs, respectively, and both adopt a cosine-scheduled learning rate initialized at 0.1. On CelebAMask-HQ, the batch size is 8 and a model pre-trained with ImageNet1K is employed. The original model and retrained model are trained over 10 epochs with a cosine-scheduled learning rate initialized at 10^{-3}. FT trains for 10 epochs with a fixed learning rate of 0.1 on SVHN, CIFAR-10, and CIFAR-100, trains for 5 epochs with a fixed learning rate of 10^{-4} on CelebAMask-HQ. GA trains for 5 epochs for the former three datasets and 3 epochs for CelebAMask-HQ, and its learning rate $lr \in [10^{-6}, 10^{-4}]$. The hyper-parameter α in IU is within the range $[1, 20]$, and the hyper-parameter γ in ℓ_1-sparse is within the range $[10^{-6}, 10^{-4}]$ with a fixed learning rate of 0.1. The FGSM step size is 0.1 for BS. Both BS and BE train for 10 epochs for the former three datasets and 5 epochs for CelebAMask-HQ, and their learning rate $lr \in [10^{-6}, 10^{-4}]$. SalUn and *Scissorhands* are trained for 10 epochs for the former three datasets and 5 epochs for CelebAMask-HQ. SalUn's learning rate $lr \in [5 \times 10^{-3}, 5 \times 10^{-2}]$ and sparsity ratio is within the range $[0.2, 0.6]$. *Scissorhands*'s learning rate $lr \in [10^{-4}, 5 \times 10^{-3}]$, percent value is within the range $[0.9, 1.0)$ and $\lambda \in [0.01, 1.0]$. When evaluating the relearn time, the learning rate is 10^{-3} on CIFAR-10 and CIFAR-100. The original model achieves an accuracy of 100% on the forgetting data.

Image Generation. We use the open-source SD v1.4 checkpoint as the pre-trained model and perform sampling with 50 time steps. We generate ∼400 images with the prompts c_f ={'nudity', 'naked', 'erotic', 'sexual'} as $\mathcal{D}_f$ and ∼400 images with the prompt c_r ={'a person wearing clothes'} as $\mathcal{D}_r$ for performing the unlearning algorithms. For the unlearning process, we employ Adam optimizer and a learning rate of 10^{-5}. We fine-tune models with SalUn and *Scissorhands* for 5 epochs with a batch size of 16. Then we evaluate on 1K generated images with prompts c_f = and 4703 generated images with I2P [44] using the open-source NudeNet classifier, with the default probability threshold of 0.6 for identifying instances of nudity.

Dataset Agreement

CelebAMask-HQ dataset and the generations by stable diffusion models might contain identification information about the personal/human subjects. We evaluate on these data for non-commercial and research purposes only.

B.2 Additional results

Table 4: Quantitative results for forgetting class on SVHN. Although ℓ_1-sparse achieves the smallest average gap performance, SalUn and our *Scissorhands* achieve higher test accuracy (better generalization) than ℓ_1-sparse when all these methods have an accuracy of 0 on the forgetting data (erase data influence).

Method	$\text{Acc}_{\mathcal{D}_f}(\downarrow)$	$\text{Acc}_{\mathcal{D}_t}(\uparrow)$	$\text{Acc}_{\mathcal{D}_r}(\uparrow)$	MIA(↑)	Avg. Gap
Retrain	0.00±0.00	92.36±1.51	97.81±0.73	100.0±0.00	-
FT [53]	82.78±8.27	95.42±0.07	100.0±0.00	93.72±10.14	23.58
GA [50]	3.77±0.16	90.29±0.08	95.92±0.25	99.46±0.05	2.07
IU [27]	64.84±0.70	92.55±0.01	97.94±0.02	72.96±0.33	23.05
BE [4]	11.93±0.42	91.39±0.05	96.89±0.28	97.91±0.13	3.98
BS [4]	11.95±0.28	91.39±0.04	96.88±0.28	97.78±0.15	4.02
ℓ_1-sparse [25]	0.00±0.00	93.83±1.47	99.41±0.90	100.0±0.00	**0.77**
SalUn [11]	0.00±0.00	95.79±0.03	100.0±0.00	100.0±0.00	1.41
Ours	0.00±0.00	95.18±0.06	99.84±0.03	100.0±0.00	1.21

Table 5: Quantitative results for forgetting 50% identities on the CelebAMask-HQ.

Method	$\text{Acc}_{\mathcal{D}_f}(\downarrow)$	$\text{Acc}_{\mathcal{D}_t}(\uparrow)$	$\text{Acc}_{\mathcal{D}_r}(\uparrow)$	MIA(↑)	Avg. Gap
Retrain	0.00±0.00	88.09±1.37	99.98±0.03	100.0±0.00	-
FT [53]	99.98±0.03	90.71±1.27	99.98±0.03	3.08±0.24	49.46
GA [50]	99.96±0.02	88.41±0.40	99.98±0.03	2.44±0.43	49.46
IU [27]	90.37±8.78	68.40±7.91	94.80±6.61	30.10±9.65	46.29
BE [4]	99.94±0.02	83.12±1.68	99.97±0.02	3.62±0.52	50.33
BS [4]	99.98±0.03	87.80±0.95	99.98±0.03	2.76±0.35	49.38
ℓ_1-sparse [25]	76.14±3.63	90.29±1.05	99.92±0.10	99.86±0.19	19.64
SalUn [11]	54.90±2.60	90.92±1.66	99.98±0.03	99.95±0.00	14.45
Ours	0.76±0.52	81.64±3.75	99.14±0.95	100.0±0.00	**2.01**

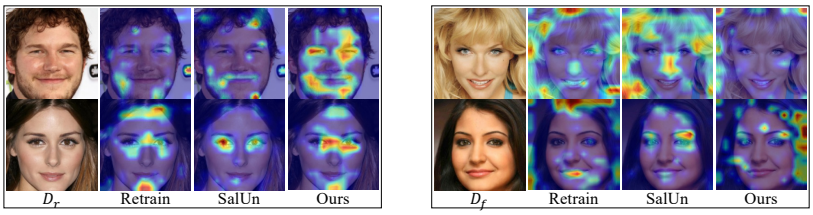

Fig. 5: Visualizations of regions where models focus on generated by GradCAM [47]. Best viewed in color.

Table 6: Quantitative results for forgetting 20% data on the SVHN, CIFAR-10 and CIFAR-100 datasets.

	Method	$\text{Acc}_{\mathcal{D}_f}(\downarrow)$	$\text{Acc}_{\mathcal{D}_t}(\uparrow)$	$\text{Acc}_{\mathcal{D}_r}(\uparrow)$	MIA(↑)	Avg. Gap
CIFAR-100	Retrain	73.25±0.53	72.95±0.28	99.98±0.01	52.58±0.64	-
	FT [53]	98.11±1.24	75.31±0.16	99.97±0.01	9.43±2.88	17.60
	GA [50]	98.11±1.26	75.55±0.12	98.23±1.16	4.91±1.97	19.22
	IU [27]	95.92±4.51	72.58±4.84	96.32±4.28	8.73±6.51	17.64
	BE [4]	97.95±1.37	72.81±0.42	97.98±1.32	8.41±2.68	17.75
	BS [4]	97.17±1.32	71.45±0.18	97.35±1.31	9.70±2.30	17.73
	ℓ_1-sparse [25]	94.35±2.64	72.57±0.80	98.80±0.57	19.11±3.56	14.03
	SalUn [11]	90.53±1.50	69.74±0.45	99.18±0.46	68.62±0.02	9.33
	Ours	67.93±2.37	70.62±0.30	97.31±0.56	43.71±1.08	**4.80**
CIFAR-10	Retrain	94.26±0.25	93.79±0.23	100.0±0.00	13.95±0.74	-
	FT [53]	99.37±0.36	94.10±0.12	99.91±0.03	2.53±0.75	4.23
	GA [50]	99.63±0.25	94.56±0.03	99.62±0.25	0.92±0.35	4.89
	IU [27]	98.58±1.49	92.39±1.92	98.64±1.41	3.49±2.69	4.39
	BE [4]	97.89±0.77	92.01±0.53	97.87±0.80	18.55±0.01	3.04
	BS [4]	99.55±0.29	94.19±0.02	99.55±0.29	6.67±0.42	3.36
	ℓ_1-sparse [25]	95.11±0.67	91.16±0.62	97.41±0.61	10.78±0.69	2.31
	SalUn [11]	98.58±0.43	93.82±0.12	99.85±0.09	15.94±1.18	1.63
	Ours	94.30±1.56	91.50±0.36	97.59±0.91	12.41±0.03	**1.57**
SVHN	Retrain	92.37±3.62	92.05±4.42	97.78±3.43	16.53±2.67	-
	FT [53]	99.52±0.24	95.12±0.11	100.0±0.00	4.02±0.38	6.24
	GA [50]	98.22±0.28	92.66±0.02	98.44±0.31	6.19±0.24	4.37
	IU [27]	95.39±1.13	89.88±0.89	96.14±1.23	11.47±1.99	**2.97**
	BE [4]	98.12±0.29	92.03±0.06	98.19±0.34	8.27±0.28	3.61
	BS [4]	97.87±0.31	91.60±0.09	97.96±0.34	8.56±0.25	3.53
	ℓ_1-sparse [25]	98.37±0.43	94.17±0.59	99.69±0.27	6.89±0.58	4.92
	SalUn [11]	99.33±0.26	95.26±0.26	99.76±0.12	13.03±1.21	3.91
	Ours	91.07±0.63	91.71±1.01	96.66±1.55	25.92±4.80	3.04

Table 7: Quantitative results for forgetting 50% data on the CIFAR-10 and CIFAR-100 datasets. Notice that while our scrubbed models are not the closest ones to the retrained models (evidenced by the average gap performance), ours achieve higher test accuracy (better generalization) and lower forget accuracy (more effective in erasing data influence) than SalUn.

	Method	$\text{Acc}_{\mathcal{D}_f}(\downarrow)$	$\text{Acc}_{\mathcal{D}_t}(\uparrow)$	$\text{Acc}_{\mathcal{D}_r}(\uparrow)$	MIA(↑)	Avg. Gap
CIFAR-100	Retrain	67.17±0.14	67.27±0.45	99.99±0.01	60.76±0.21	-
	FT [53]	98.17±1.20	75.36±0.36	99.97±0.01	9.26±2.84	22.65
	GA [50]	98.15±1.23	75.50±0.10	98.22±1.17	4.94±1.96	24.20
	IU [27]	96.86±2.19	72.08±2.41	97.17±2.00	8.20±4.10	22.47
	BE [4]	97.35±1.60	67.84±0.58	97.27±1.62	8.62±2.19	21.40
	BS [4]	95.31±1.47	68.12±0.18	95.41±1.46	10.07±1.99	21.07
	ℓ_1-sparse [25]	90.17±2.43	69.73±1.27	97.35±0.89	21.72±1.44	16.79
	SalUn [11]	84.81±0.91	64.94±0.48	98.89±0.48	73.86±1.98	**8.54**
	Ours	79.73±2.28	67.58±1.76	84.64±2.79	28.68±2.53	15.08
CIFAR-10	Retrain	92.17±0.26	91.71±0.30	100.0±0.00	19.13±0.55	-
	FT [53]	99.50±0.33	94.32±0.07	99.96±0.03	2.31±1.08	6.70
	GA [50]	99.60±0.27	94.55±0.06	99.62±0.26	0.96±0.40	7.20
	IU [27]	97.54±1.99	91.10±5.25	97.62±1.98	5.25±3.01	5.56
	BE [4]	99.57±0.28	94.28±0.04	99.59±0.28	10.82±0.89	4.67
	BS [4]	99.58±0.28	94.44±0.03	99.60±0.27	1.99±0.08	6.92
	ℓ_1-sparse [25]	97.42±0.60	92.10±0.24	98.89±0.15	6.59±0.80	4.82
	SalUn [11]	92.15±1.18	88.15±0.90	95.02±0.98	19.30±2.81	**2.18**
	Ours	92.02±5.31	88.32±4.24	94.00±4.87	15.52±6.43	3.29
SVHN	Retrain	93.45±1.69	93.85±1.61	99.69±0.62	19.25±2.80	-
	FT [53]	99.50±0.25	95.08±0.10	100.0±0.00	4.49±0.33	5.59
	GA [50]	97.72±0.34	91.82±0.07	97.90±0.39	7.36±0.44	5.00
	IU [27]	97.37±0.62	91.80±0.64	97.94±0.66	8.24±0.78	4.68
	BE [4]	94.60±4.71	88.03±5.54	94.60±4.77	13.47±8.70	4.46
	BS [4]	97.51±0.31	90.87±0.06	97.55±0.36	10.12±0.51	4.58
	ℓ_1-sparse [25]	92.77±0.40	92.16±0.57	97.54±0.40	15.81±0.88	**1.99**
	SalUn [11]	98.67±0.28	93.66±0.07	98.83±0.27	14.89±0.36	2.66
	Ours	97.23±0.31	94.47±0.07	99.66±0.07	10.85±0.92	3.21

Table 8: Relearn time and overhead when forgetting 10% data on CIFAR-10. Relearn time denotes the epochs to regain performance on $\mathcal{D}_f$, measured over four runs. RTE is defined as the ratio of the time needed for forgetting to the time for retraining. Memory is computed via the module Memory Profiler to monitor the memory consumption of algorithms. Although *Scissorhands* outperforms SalUn in terms of relearn time (i.e., the effectiveness of forgetting), our method introduces more computational cost than SalUn. This is because, during the repair process, SalUn only fine-tunes the specific model parameters identified via the saliency scores, while ours fine-tunes the whole network.

Method	Relearn time (↑)		Overhead	
	CIFAR-10	CIFAR-100	Memory (MiB)	RTE
SalUn	24.25	41.50	1968.4	0.075
Ours	>200	>200	2002.7	0.182

Table 9: Evaluation on the class and nudity erasure. We use scrubbed models that forget 'nudity' to generate images with COCO-30K prompts and measure FID, and CLIP scores to show the generated image quality. RTE is not provided as retrained models in these cases can not be easily obtained.

Method	Imagenette			COCO-30K	
	FID↓	CLIP↑	UA↑	FID↓	CLIP↑
SalUn	1.49	31.92	100%	25.06	28.91
Ours	1.09	31.02	100%	19.45	30.73

Fig. 6: Sample images with the I2P prompt generated by SDs **w**/ and **w**/**o** machine unlearning algorithms (SD v1.4 [42], SD v2.1 that is trained on a dataset filtered for nudity, ESD-u [14] and SalUn [11]). Best viewed in color.

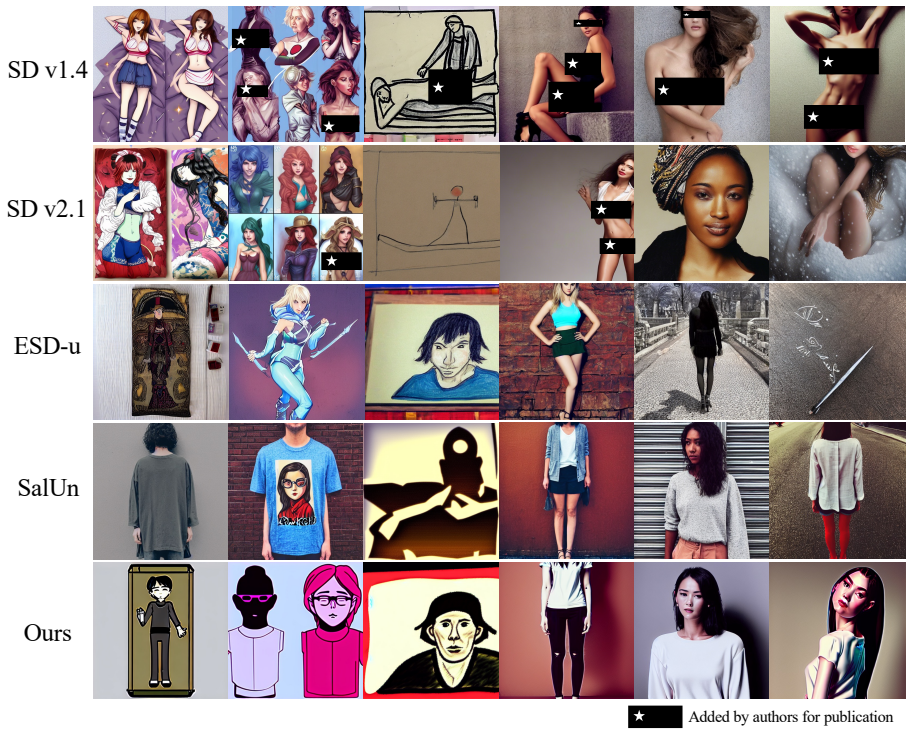

Fig. 7: Sample images with the I2P prompt generated by SDs **w/** and **w/o** machine unlearning algorithms. Best viewed in color.

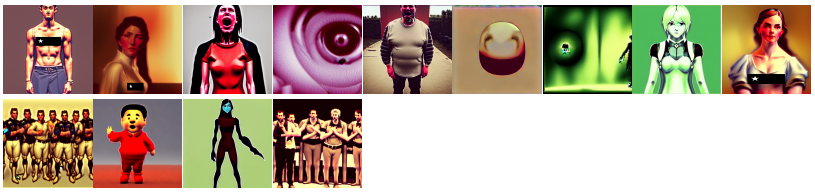

Fig. 8: The flagged images detected as exposed female breast (top)/genitalia (bottom) by the NudeNet classifier.