Segue: Side-information Guided Generative Unlearnable Examples for Facial Privacy Protection in Real World

Zhiling Zhang[1], Jie Zhang[2,*], Kui Zhang[1], Wenbo Zhou[1,*], Ting Xu[3], Daiheng Gao[1],
Zixian Guo[1], Qinglang Guo[1,4], Weiming Zhang[1], Nenghai Yu[1]
[1]University of Science and Technology of China, China
[2]Centre for Frontier AI Research, Agency for Science, Technology and Research (A*STAR), Singapore
[3]National University of Singapore, Singapore
[4]China Academic of Electronics and Information Technology, China

***Abstract*—The widespread adoption of face recognition has raised privacy concerns regarding the collection and use of facial data. To address this, researchers have explored "unlearnable examples" by adding imperceptible perturbations during model training to prevent the model from learning target features. However, current methods are inefficient and cannot guarantee transferability and robustness at the same time, causing impracticality in the real world. To remedy it, we introduce Side-information Guided Generative Unlearnable Examples (Segue). Using a once-trained multiple-used model to generate perturbations, Segue avoids the time-consuming gradient-based approach. To improve transferability, we introduce side information such as true or pseudo labels, which are inherently consistent across different scenarios. For robustness enhancement, a distortion layer is integrated into the training pipeline. Experiments show Segue is 1000× faster than previous methods, transferable across datasets and models, and resistant to JPEG compression, adversarial training, and standard augmentations.**

***Index Terms*—Information security, Face recognition, Privacy protections.**

I. Introduction

The rise of social media platforms like Twitter and Facebook has led to an increase in publicly shared facial data for fun or commercial purposes. This also facilitates the unauthorized collection of facial data, violating public privacy [1], [2]. Such data can be used to train face analysis models, including face recognition systems [3]–[6], posing a threat to security-critical applications such as authentication systems [7], [8]. Malicious actors can further exploit this technology using drones or surveillance cameras to track and monitor individuals, which not only infringes on personal privacy but also endangers personal safety. Thus, it's crucial to protect individual faces from unauthorized use.

Recent works utilize *unlearnable examples* [9]–[11] for facial privacy protection. In this approach, the defender applies perturbations to the original image, generating unlearnable examples. When facial recognition (FR) models are trained on these modified images, they fail to recognize individuals using drones or surveillance cameras accurately. This technique exploits neural networks' tendency to rely on shortcuts as discriminative features [12], with the added perturbation serving as such a shortcut. In conclusion, unlearnable examples provide an effective means of safeguarding facial privacy.

To generate practical unlearnable facial examples, five key requirements must be met: 1) *Effectiveness*: The facial recognition model should fail to recognize clean examples, with accuracy reduced to random guessing. 2) *Imperceptibility*: Perturbations must be invisible, making unlearnable examples indistinguishable from clean ones. However, some methods [13] generate visually noticeable perturbations. 3) *Transferability*: Perturbations should transfer across different datasets [5], [6], [14] and model architectures [15]–[17]. However, gradient-based methods [9]–[11] are limited to fixed categories and depend on target model gradients, making them impractical in black-box settings. 4) *Robustness*: Perturbations must resist transmission distortions (e.g., JPEG compression) and adaptive attacks like adversarial training [18], but many methods [9], [11] fail to do so. 5) *Efficiency*: Fast generation is crucial for online use, yet gradient-based methods [9]–[11] are computationally costly. Existing methods [9]–[11] fail to meet all these criteria, particularly in terms of efficiency, transferability, and robustness, limiting their practice for facial privacy protection.

To address the limitations of existing methods, we propose **Segue**, a **s**i**d**e-information-guided **g**enerative **u**nlearnable **e**xamples approach. Unlike previous iterative gradient optimization methods, Segue leverages an auto-encoder model to generate perturbations, enabling a once-trained, multi-use capability efficiently. Furthermore, we constrain the perturbation using L2 loss to ensure imperceptibility. To enhance transferability, we incorporate side information, utilizing true labels when available, or pseudo labels from K-means clustering [19] for unlabeled datasets. Moreover, a distortion layer is added to simulate real-world transmission distortions (e.g., JPEG compression, blurring) and adversarial training, further enhancing robustness.

We conduct extensive experiments demonstrating that **Segue** successfully meets the five requirements mentioned above. Our method significantly reduces the attacker's model recognition capability, achieving 11.5% accuracy on VGGFace10, compared to previous methods, which only reduce accuracy to 20.5%. Additionally, we assess transferability across six model architectures and five facial datasets, where **Segue** outperforms in most cases. Regarding robustness, Segue can withstand adversarial training with varying intensities and multiple distortions. Furthermore, **Segue** is significantly faster than existing methods [9]–[11] (1000×). Ablation studies further validate our design.

II. Preliminary

A. Formalized Description

We can divide a clean dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{n}$ into a training set $\mathcal{D}_{train}$ and a testing set $\mathcal{D}_{test}$. Huang et al. [9] propose a bi-level objective to generate perturbations to prevent the face recognition (FR) model from learning anything from the training data. They use the following objective function:

*Corresponding author: Jie Zhang and Wenbo Zhou. This work was supported in part by the Natural Science Foundation of China under Grant 62121002, U20B2047, U2336206, 62372423, and 62102386. Email: {zhilingzhang,zk19,gzxx,gql1993}@mail.ustc.edu.cn, zhang_jie@cfar.a-star.edu.sg, {welbeckz,zhangwm,ynh}@ustc.edu.cn, xuting@nus.edu.sg, samuel.gao023@gmail.com

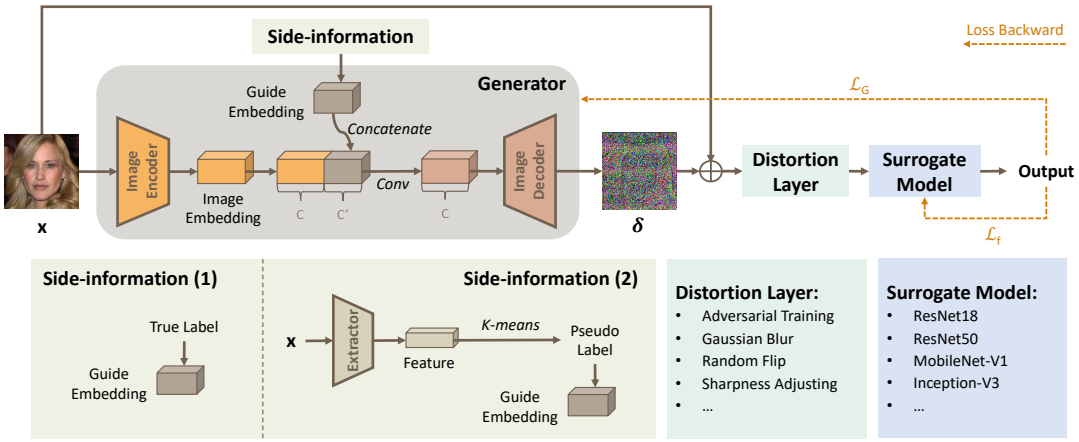

Fig. 1. The overall framework of Segue. The generator comprises an encoder and decoder, fusing the image and side information in the feature space. Side information is either the true label (supervised) or the pseudo label (unsupervised). The distortion layer enhances the robustness of δ. The generator and surrogate model are trained alternately.

$$\arg\min_{f} \mathbb{E}_{(x,y)\sim\mathcal{D}_{train}} \left[\min_{\delta} \mathcal{L}\left(f(x+\delta), y\right) \right], \tag{1}$$

where the modified image $x+\delta$ is called as the unlearnable example. f acts as a surrogate model for the target model. The perturbation δ is bounded by $\|G(x)\|_p \leq \epsilon$ to guarantee that it is imperceptible to human eyes.

B. Threat Model

Following current methods, we assume that the attacker wants to train a FR model but only has access to the unlearnable examples instead of $\mathcal{D}_{train}$. To boost the model performance, the attacker may apply data augmentation techniques, such as Cutout, Mixup, and CutMix. Besides, the attacker may also use adversarial training. Besides, we consider a black-box scenario, where the defender has no knowledge of the target model, including parameters and architectures, used by the attacker. Instead, we use a surrogate model f as an approximation. We can use the dataset labels as side information if we have them. However, the dataset labels are not necessary. We only need to know the number of identities K in the dataset, which is used to cluster the image features and obtain the pseudo labels. We describe this process in more detail in the next section. Our goal is to protect the user's facial privacy. To do this, we optimize δ following Eq. (1) and add δ to $\mathcal{D}_{train}$ to prevent the attacker from learning useful information from it. As a result, the FR model trained with the unlearnable dataset fails to recognize the images in $\mathcal{D}_{test}$ since the distributions have changed between them.

III. Method

This section covers side information, the once-trained multiple-use generator, the distortion layer, and the two-stage training strategy.

A. The Design of Side Information

Linear separability allows samples to be easily distinguished by linear models. Yu et al. [13] show that perturbations in training-stage availability attacks [9], [20] are linearly separable, making models learn the noise instead of image information.

Strong linear separability is achieved by adding class-wise perturbations, such as applying δ_i to all samples of class i. Huang et al. [9] note that random class-wise perturbations prevent models from learning useful information and enhance transferability. To improve transferability, increasing linear separability using true labels (supervised) or unlabeled facial data (unsupervised) is effective.

Algorithm 1 Two-stage Training Strategy

Input: image x, side information $\hat{y}$, dataset D, distortion layer T, generator G, surrogate model f, learning rate α_f and α_G
Output: generator G

for $epoch = 1$ to E **do**
  if $epoch\%5 = 1$ **then**
    for $i = 1$ to $maxiter$ **do**
      Sample $(x_i, y_i) \sim \mathcal{D}$
      $\delta_i \leftarrow Clip(G(x_i), -\epsilon, \epsilon)$
      $x_i' \leftarrow x_i + \delta_i$
      $\theta_{f,i+1} \leftarrow \theta_{f,i} - \alpha_f \nabla_{\theta_{f,i}} \mathcal{L}_f(f(x_i'), \hat{y}_i)$
    end for
  else
    for $i = 1$ to $maxiter$ **do**
      Sample $(x_i, y_i) \sim \mathcal{D}$
      $\delta_i \leftarrow Clip(G(x_i), -\epsilon, \epsilon)$
      $x_i' \leftarrow x_i + \delta_i$
      $\theta_{G,i+1} \leftarrow \theta_{G,i} - \alpha_G \nabla_{\theta_{G,i}} \mathcal{L}_G(f(x_i'), \hat{y}_i)$
    end for
  end if
end for

1) Supervised Scenario: In the supervised scenario, we use dataset labels as side information. Following [21], we concatenate label embeddings with image embeddings in the high-level feature space to guide generation and reduce the channel dimension $C+C'$ back to C using an extra convolution layer. The label embedding is a 16-bit binary vector, e.g., $0\ldots0101$ for label $y = 5$. This supports up to 2^{16} identities, adaptable to various datasets [5], [6], [14], and can be adjusted based on the number of classes.

2) Unsupervised Scenario: To address the lack of labels, we generate pseudo labels using an unsupervised approach [22]. We extract facial features with a model trained on CelebA [6], apply K-means clustering [19] to group them into K clusters, and assign pseudo labels. These pseudo labels are then concatenated with image features as in the supervised scenario. This eliminates manual labeling and only requires knowing the number of classes K. Our method remains effective as long as clustering accuracy exceeds 80%, ensuring robust transferability across datasets and models.

TABLE I
COMPARISON OF EFFECTIVENESS (CLEAN TEST ACC % ↓) AMONG METHODS USING RESNET18 ON FIVE DATASETS.

Methods	WebFace10	WebFace50	VGGFace10	CIFAR10	ImageNet10
CLEAN	75.00	80.60	83.00	91.67	71.00
UE [9]	12.50	3.50	20.50	19.93	30.00
LSP [13]	31.50	9.30	57.50	17.07	28.50
RUE [10]	11.50	7.40	30.00	15.18	24.50
TUE [11]	33.50	11.20	82.00	11.25	60.50
Ours	**10.50**	**2.50**	**11.50**	**10.12**	**14.00**

TABLE II
EFFECTIVENESS ON LARGER DATASETS (CLEAN TEST ACC % ↓).

Methods	WebFace500	VGGFace400	CelebA500
CLEAN	82.74	78.60	80.60
Ours	**0.23**	**1.09**	**3.73**

B. Once-trained Multiple-used Generator

As shown in Fig. 1, the generator G encodes the input into an embedding, fuses it with the guide embedding, and decodes it into a perturbation δ using 3×3 convolutions, batch normalization, and ReLU activation. Unlike previous methods [9], [11] that directly optimize perturbations, we optimize a generator to produce perturbations based on inputs. This allows generating perturbations for different datasets with one generator, improving efficiency over existing methods that require retraining for each dataset.

C. Distortion Layer

Inspired by RUE [10], which adopts a min-min-max framework to introduce adversarial training and increase the difficulty of perturbation generation, we incorporate a distortion layer to simulate potential distortions that may weaken the protective effect of perturbations.

In fact, adversarial training can be viewed as a form of data augmentation. Therefore, we enhance the data through a distortion layer that applies adversarial training, Gaussian blurring, random flips, and more (see Fig. 1).

D. Two-stage Training Strategy

As shown in Alg. 1, we alternately train the surrogate model f (ResNet18 by default) and the generator G. In the first stage, we train f for $maxiter$ iterations using the loss $\mathcal{L}_f$ to ensure the perturbed image $x + G(x)$ is correctly classified as $\hat{y}$:

$$\mathcal{L}_f = CE(f(x + G(x)), \hat{y}), \tag{2}$$

where $\hat{y}$ includes true or pseudo labels, and CE is the Cross-Entropy loss. In the second stage, we update G for four epochs. The loss function for G includes two terms: $\mathcal{L}_{G1}$, which reduces the loss of f on unlearnable examples, and $\mathcal{L}_{G2}$, which constrains the perturbation magnitude:

$$\mathcal{L}_G = \alpha \cdot \mathcal{L}_{G1} + \beta \cdot \mathcal{L}_{G2}, \tag{3}$$

$$\mathcal{L}_{G1} = CE(f(x + G(x)), \hat{y}), \quad \mathcal{L}_{G2} = \mathbb{E}_x(\|G(x)\|_2) \tag{4}$$

where α and β control the weight of each term. Training ends after 20 epochs or when the loss of f on the unlearnable examples drops below 0.001.

IV. EXPERIMENTS

We evaluate **Segue** on effectiveness, imperceptibility, transferability, robustness, and efficiency, comparing it with current methods to demonstrate its advantages. Ablation studies are also provided.

TABLE III
COMPARISON OF IMPERCEPTIBILITY ON WEBFACE10.

Methods	PSNR(↑)	SSIM(↑)	MSSSIM(↑)	LPIPS(↓)	MAE(↓)	RMSE(↓)
UE [9]	32.37	0.754	0.973	0.205	0.0219	0.0241
LSP [13]	31.53	**0.968**	0.974	0.049	0.0254	0.0265
RUE [10]	**32.45**	0.763	0.977	0.188	**0.0186**	**0.0212**
TUE [11]	30.18	0.651	0.952	0.310	0.0308	0.0309
Ours	30.54	0.673	**0.980**	**0.047**	0.0224	0.0248

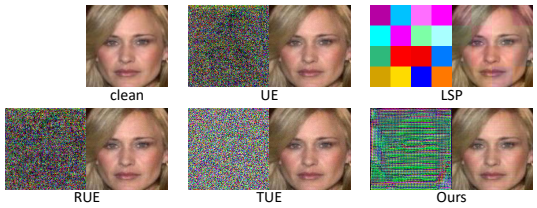

Fig. 2. Visualization of different unlearnable examples and the corresponding residual compared with the clean image.

A. Experimental Settings

1) Datasets: We use three facial datasets: WebFace [5], VGGFace2 [14], and CelebA [6] and two non-facial datasets: CIFAR10 [23] and ImageNet10 [24]. Besides, we randomly select categories for sub-datasets (e.g., WebFace10 with 10 categories) and resize all images to 224×224 (except for CIFAR10, which remains at 32×32).

2) Metric: We use *clean test accuracy*, which measures model performance on clean examples after training on unlearnable examples. Lower accuracy indicates more effective unlearnable examples. Accuracy close to $\frac{100\%}{\#IDs}$ suggests the model learns nothing, resembling random guessing.

3) Baselines: We compare our method with three grad-based approaches: UE [9], RUE [10], and TUE [11], and the model-agnostic LSP [13], using their official code.

4) Implementation Details: We constrain the perturbation to $\|\delta\|_\infty \leq \epsilon = \frac{8}{255}$ and employ the Adam optimizer with an initial learning rate of 0.0005. The parameters α and β in Eq. (3) are set to 1 and 0.001, respectively. ResNet18 and WebFace10 are used as defaults. The distortion layer includes adversarial training, Gaussian blur, sharpness adjustment, random horizontal flips, and random vertical flips. For adversarial training, we use a default perturbation size $\rho_d = \frac{1}{255}$, which is varied between $\rho_d = \frac{0}{255}$ and $\rho_d = \frac{4}{255}$ in robustness experiments. Gaussian blur uses a (3,3) kernel with sigma 0.2, and the sharpness factor is set to 2. The probabilities for horizontal and vertical flips are both 0.1.

B. Effectiveness and Imperceptibility

Tab. I shows that our method achieves the best performance across three facial datasets, successfully reducing accuracy to approximately $\frac{100\%}{\#IDs}$. Variations in results are due to dataset characteristics: WebFace50 has more categories, increasing classifier difficulty and lowering accuracy, while VGGFace10's higher image quality makes feature learning easier, resulting in higher accuracy. Tab. II presents results on larger datasets. We trained the noise generator on WebFace500 and applied it to WebFace500 and two unseen datasets, demonstrating that our method is both effective and transferable. Quantitative results are provided in Tab. III, and visual examples with perturbations magnified 30× are shown in Fig. 2. As indicated in Tab. III, our method achieves visual quality comparable to current baseline methods.

C. Transferability

1) Different Models: All methods generate unlearnable examples using ResNet18 as the surrogate model, testing transferability across five architectures. As shown in Tab. IV, our method consistently

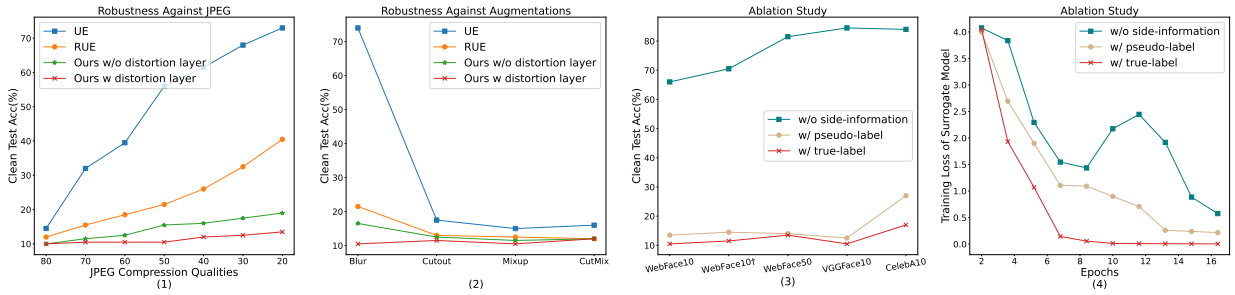

Fig. 3. (1) Robustness to JPEG compression (lower quality = higher compression). (2) Robustness to data augmentations. (3) Effect of side information on transferability. (4) Effect of side information on surrogate model training loss. Lower values indicate better performance.

TABLE IV
TRANSFERABILITY ACROSS MODELS (CLEAN TEST ACCURACY % ↓). DEFENDERS USE RESNET18, WHILE ATTACKERS USE FIVE MODELS: RS (RESNET), MN (MOBILENET), I (INCEPTION), EN (EFFICIENTNET), AND ST (SWIN TRANSFORMER).

Methods	RN18	RN50	MN	I-v3	EN-b1	ST
UE [9]	14.50	14.50	15.50	73.00	28.00	25.50
LSP [13]	31.50	32.50	18.50	56.00	52.50	16.50
RUE [10]	19.00	27.50	17.00	77.00	27.00	21.50
TUE [11]	33.50	70.00	15.50	69.00	67.50	16.00
Ours	**10.50**	**12.50**	**11.00**	**10.50**	**12.00**	**14.50**

TABLE V
TRANSFERABILITY ACROSS DATASETS (CLEAN TEST ACC % ↓). WEBFACE10 PERTURBATIONS APPLIED TO OTHER DATASETS.

Methods	WebFace10	WebFace10†	WebFace50	VGGFace10	CelebA10
UE [9]	12.50	14.50	\	21.50	44.00
LSP [13]	31.50	35.00	**9.30**	57.50	74.00
RUE [10]	17.00	26.50	\	78.50	78.50
TUE [11]	33.50	52.00	\	53.00	59.00
Ours	**10.50**	**11.50**	13.50	**13.00**	**17.00**

outperforms others, particularly on deeper networks like Inception-V3, where traditional methods fail in privacy protection. This is likely because Inception-V3 employs multiple convolutional kernels to capture multi-scale image features. In contrast, conventional unlearnable noise is constrained by the surrogate model's structure and overlooks multi-scale linear consistency. Our approach leverages side information to generate noise, maintaining linear consistency across different convolutional scales.

2) Different Datasets: In Tab. V, we generate perturbations using WebFace10 and evaluate them across various datasets. Despite CelebA10's smaller class sizes, which demand higher linear separability, our method reduces accuracy to 17%. In contrast, TUE and UE fail to transfer perturbations to WebFace50 due to their fixed perturbation size during training, restricting their transferability to smaller datasets and limiting their broader applicability.

D. Robustness

1) Adversarial Training: Adversarial training effectively removes non-robust noise from inputs [18]. The attacker applies adversarial training with ρ_a to mitigate perturbations, while we use ρ_d in the distortion layer to enhance perturbation robustness. When both ρ_a and ρ_d are 0, neither party employs adversarial training. As shown in Tab. VI, our method maintains strong performance, achieving 16.5% clean data accuracy even when the attacker uses $\rho_a = 4/255$.

2) JPEG Compression: Lower JPEG compression quality degrades image quality and weakens unlearnable perturbation protection. Fig. 3 (1) shows that our method remains effective across all quality levels, while others fail under low-quality conditions.

TABLE VI
ROBUSTNESS AGAINST ADVERSARIAL TRAINING (CLEAN TEST ACC % ↓). PERTURBATION BUDGET: ATTACKER ρ_a, DEFENDER ρ_d.

Adv. Train.	Clean	UE	RUE			Ours		
ρ_a			ρ_d=0	2/255	4/255	ρ_d=0	2/255	4/255
0	75.00	12.50	11.50	13.00	14.50	**10.50**	12.50	13.50
1/255	68.00	18.00	17.50	15.50	17.50	13.50	**11.50**	12.50
2/255	65.00	69.00	26.00	19.50	22.50	15.50	15.00	**12.50**
3/255	63.50	74.50	69.50	61.50	58.00	29.00	16.00	**14.50**
4/255	65.50	69.50	71.00	62.00	63.00	34.00	21.00	**16.50**

TABLE VII
COMPARISON OF EFFICIENCY ON WEBFACE10.

Methods	UE [9]	LSP [13]	RUE [10]	TUE [11]	Ours
Time (s)	∼2.1k	4.5	∼6.7k	∼7.4k	**2.2**

3) Data Augmentation: We apply augmentations as follows: Gaussian blurring with a kernel size of 5 and standard deviation of 1.0; Cutout [25] with two patches, each 112 pixels (half the image size); Mixup [26], mixing random image pairs with λ from a beta distribution [0,1]; and CutMix [27], using Mixup on the Cutout region with the same settings. Fig. 3 (2) demonstrates our method's robustness to all these augmentations.

E. Efficiency

We use a server with a single A6000 GPU and an Intel Xeon Gold 6130 CPU. Our method requires only one-step inference, while LSP needs no training. In contrast, UE, TUE, and RUE demand multiple costly SGD updates. Tab. VII shows our method is 1000× faster than gradient-based approaches.

F. Ablation Study

Fig. 3 (3) shows that side information enhances both the effectiveness and transferability of perturbations, with true labels outperforming pseudo-labels. Additionally, Fig. 3 (4) demonstrates that without side information, the generator struggles to converge, causing fluctuations in the surrogate model's training loss. We attribute this to side information serving as a prior, which narrows the generator's search space and accelerates convergence.

V. CONCLUSION

We propose **Segue**, a novel method for facial privacy protection using unlearnable examples that meet five key criteria: effectiveness, imperceptibility, transferability, robustness, and efficiency. By leveraging generative models with side information, Segue creates unlearnable examples that evade recognition by face recognition models. Our approach demonstrates strong transferability across datasets and models, resilience against attacks and distortions, and up to 1000× faster generation than existing methods. We believe our

work can provide a new perspective and a practical solution for facial privacy protection in the real world.

REFERENCES

[1] V. U. Prabhu and A. Birhane, "Large image datasets: A pyrrhic win for computer vision?" *2021 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 1536–1546, 2020.

[2] K. Hill, "The secretive company that might end privacy as we know it," in *Ethics of Data and Analytics*. Auerbach Publications, 2022, pp. 170–177.

[3] H. Jiang and E. G. Learned-Miller, "Face detection with the faster r-cnn," *2017 12th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017)*, pp. 650–657, 2016.

[4] R. Ranjan, S. Sankaranarayanan, C. D. Castillo, and R. Chellappa, "An all-in-one convolutional neural network for face analysis," *2017 12th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017)*, pp. 17–24, 2016.

[5] D. Yi, Z. Lei, S. Liao, and S. Z. Li, "Learning face representation from scratch," *arXiv preprint arXiv:1411.7923*, 2014.

[6] Z. Liu, P. Luo, X. Wang, and X. Tang, "Deep learning face attributes in the wild," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 3730–3738.

[7] B. Yin, W. Wang, T. Yao, J. Guo, Z. Kong, S. Ding, J. Li, and C. Liu, "Adv-makeup: A new imperceptible and transferable attack on face recognition," in *International Joint Conference on Artificial Intelligence*, 2021.

[8] X. Yang, C. Liu, L. Xu, Y. Wang, Y. Dong, N. Chen, H. Su, and J. Zhu, "Towards effective adversarial textured 3d meshes on physical face recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 4119–4128.

[9] H. Huang, X. Ma, S. M. Erfani, J. Bailey, and Y. Wang, "Unlearnable examples: Making personal data unexploitable," in *ICLR*, 2021.

[10] S. Fu, F. He, Y. Liu, L. Shen, and D. Tao, "Robust unlearnable examples: Protecting data against adversarial learning," in *International Conference on Learning Representations*, 2022.

[11] J. Ren, H. Xu, Y. Wan, X. Ma, L. Sun, and J. Tang, "Transferable unlearnable examples," in *The Eleventh International Conference on Learning Representations*, 2023.

[12] R. Geirhos, J.-H. Jacobsen, C. Michaelis, R. S. Zemel, W. Brendel, M. Bethge, and F. Wichmann, "Shortcut learning in deep neural networks," *Nature Machine Intelligence*, 2020.

[13] D. Yu, H. Zhang, W. Chen, J. Yin, and T.-Y. Liu, "Availability attacks create shortcuts," *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2021.

[14] Q. Cao, L. Shen, W. Xie, O. M. Parkhi, and A. Zisserman, "Vggface2: A dataset for recognising faces across pose and age," *2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018)*, pp. 67–74, 2017.

[15] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2015.

[16] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," *arXiv preprint arXiv:1704.04861*, 2017.

[17] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2818–2826, 2015. [Online]. Available: https://api.semanticscholar.org/CorpusID:206593880

[18] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *CoRR*, vol. abs/1412.6572, 2014.

[19] S. Z. Selim and M. A. Ismail, "K-means-type algorithms: A generalized convergence theorem and characterization of local optimality," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-6, pp. 81–87, 1984.

[20] J. Feng, Q.-Z. Cai, and Z.-H. Zhou, "Learning to confuse: Generating training time adversarial data with auto-encoder," in *Neural Information Processing Systems*, 2019.

[21] J. Han, X. Dong, R. Zhang, D. Chen, W. Zhang, N. Yu, P. Luo, and X. Wang, "Once a man: Towards multi-target attack via learning multi-target adversarial network once," *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 5157–5166, 2019.

[22] J. Zhang, X. Ma, Q. Yi, J. Sang, Y. gang Jiang, Y. Wang, and C. Xu, "Unlearnable clusters: Towards label-agnostic unlearnable examples," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023.

[23] A. Krizhevsky, "Learning multiple layers of features from tiny images," in *Technical report, University of Toronto*, 2009.

[24] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. S. Bernstein, A. C. Berg, and L. Fei-Fei, "Imagenet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, pp. 211–252, 2014.

[25] T. DeVries and G. W. Taylor, "Improved regularization of convolutional neural networks with cutout," *arXiv preprint arXiv:1708.04552*, 2017.

[26] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond empirical risk minimization," *arXiv preprint arXiv:1710.09412*, 2017.

[27] S. Yun, D. Han, S. J. Oh, S. Chun, J. Choe, and Y. J. Yoo, "Cutmix: Regularization strategy to train strong classifiers with localizable features," *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 6022–6031, 2019.