k-Same-Siamese-GAN: k-Same Algorithm with Generative Adversarial Network for Facial Image De-identification with Hyperparameter Tuning and Mixed Precision Training

Yi-Lun Pan$^{1,2}$, Min-Jhih Haung$^1$, Kuo-Teng Ding$^1$, Ja-Ling Wu$^1$, Jyh-Shing Jang$^1$

$^1$Department of Computer Science & Information Engineering, National Taiwan University, Taiwan
$^2$National Center for High-performance Computing

Abstract

In recent years, advances in camera and computing hardware have made it easy to capture and store amounts of image and video data. Consider a data holder, such as a hospital or a government entity, who has a privately held collection of personal data. Then, how can we ensure that the data holder does conceal the identity of each individual in the imagery of personal data while still preserving certain useful aspects of the data after de-identification? In this work, we proposed a novel approach towards high-resolution facial image de-identification, called k-Same-Siamese-GAN (kSS-GAN), which leverages k-Same-Anonymity mechanism, Generative Adversarial Network (GAN), and hyperparameter tuning. To speed up training and reduce memory consumption, the mixed precision training (MPT) technique is also applied to make kSS-GAN provide guarantees regarding privacy protection on close-form identities and be trained much more efficiently as well. Finally, we dedicated our system to an actual dataset - RafD dataset for performance testing. Besides protecting privacy of high-resolution facial images, the proposed system is also justified for its ability in automating parameter tuning and breaking through the limitation of the number of adjustable parameters.

1 Introduction

The protection of facial images is getting higher and higher awareness recently, especially when personal images and videos are easily captured by pervasively equipped high-resolution visual devices, such as smart phone and surveillance cameras. These devices did simplify the image- and video footage capturing tasks a lot, nevertheless, attention should be paid to the misuse of the captured imagery data especially when they are stored on a datacenter. De-identification [Gellman, 2010] is one of the basic methods aiming at protecting the privacy of imagery data and granting their legal usage, at the same time.

In facial image de-identification problems, there are two kinds of dilemma. One is to make the surrogate images retain as much structural information in the original image as possible so that the image utility remains. Therefore, the existing facial image de-identification procedures based mostly on ksame algorithms [Newton et al., 2005; Gross et al., 2008] and leveraged some facial feature preserving techniques, such as the Active Appearance Models (AAMs) [Gross et al., 2006; Samarzija and Ribaric, 2014] and the Principal Component Analysis (PCA) method [Meng et al., 2014], to explicitly construct morphing-faces for preserving the facial utility attributes like age, race, and gender, as much as possible. Unfortunately, the results are too fake to be applied in real world, due to the corresponding poor visual quality and unnatural appearance.

The other one is to make surrogate images as different from the original image as possible so as to ensure the removal of any personal identifiable attribute. The work [Letournel et al., 2015] presented a facial image de-identification method with expression preservation capabilities based on variational adaptive filters, where the filtering process preserves most of the important facial image characteristics (i.e., eyes, gazed regions, lips, and their corners). Since only the traditional de-identification algorithms are adopted in [Letournel et al., 2015], there is still a large room left for performance enhancing with the aid of advanced Deep Neural Network (DNN)-related architectures, such as Generative Neural Networks (GNNs), Convolution Neural Networks (CNNs), and Generative Adversarial Networks (GANs). Moreover, even if DNN-related algorithms and/or architectures are applied, the associated large time spent and huge amount of memory consuming in GPGPU become the critical obstacles needed to be conquered.

To tackle the above-mentioned problems, a novel GAN-based facial image de-identification system is proposed, which relies on the formal privacy protection ability provided by the above-mentioned k-Same approaches. In which, we not only designed a novel label-scheme function for generating de-identified image set automatically and flexibly, but also took the appropriate hyperparameter tuning into account. In addition, for enhancing the re-identification ability, a recently proposed image recognition neural network, called the Siamese Network [Koch et al., 2015], has been modified and integrated into our kSS-GAN. The whole processing proce-
dures of our work can be divided into the following three stages:

1. Replacing all images within a k-sized cluster by the same surrogate image to ensure anonymity
2. Selecting the centroid of a cluster as the surrogate via the proposed labelling scheme to minimize information loss during the de-identification stage, and
3. Avoiding the risk of losing data utility with the aid of the k-Same algorithm.

Comparing with related works, the proposed facial image de-identification system provides better privacy protection ability, surrogate image quality, and training performance by leveraging the superiority of the mixed precision training technique. The main contributions of our work can then be summarized as follows.

1. Proposed a novel GAN-based de-identification algorithm for protecting the privacy and the utility of high-resolution facial images at the same time,
2. Proposed a novel labelling scheme to automatically and flexibly generate the de-identified image set,
3. Leveraged the k-Same algorithm to provide guarantees for privacy protection,
4. Speeded up the training process with the aid of mixed precision training, and
5. Integrated appropriate automatic Machine Learning (autoML) toolkit, such as Advisor [Golovin et al., 2017], to search for better hyperparameters.

2 Related Works

2.1 k-Anonymity and Face De-identification

The concept of k-Anonymity was addressed by Sweeney [Sweeney, 2002] and had been applied to de-identify the entries of a relational database. Sweeney said a table is conformed to the k-Anonymity (after de-identification) if and only if each sequence of values, for each attribute, in the table appears at least k times. Since there are k copies of the same sequence in the data set, the probability of connecting a single sequence to the original sequence is bounded by 1/k. The k-Anonymity model inspired a series of the so-called k-Same de-identification algorithms. After comprehending the basic concepts of k-Anonymity and the behavior of k-Same algorithms, later in this section, we will briefly review some highly relevant articles to our study and then analyze their pros and cons.

Gross, R., et al. [Gross et al., 2006] proposed a model-based face de-identification method. The weaknesses of this work include - (1) the group of surrogates cannot be extended, in other words, k-value is limited, (2) the problem of ghosting effects cannot be solved very well, and (3) the most fatal weakness is that the background information of the raw images must be removed so as to align each image with one another. To overcome these challenges, we leveraging the k-Same-M positive property (will be detailed later) and integrating AAM into the k-Anonymity parameter space for keeping the visual information and facial expression related attributes as much as possible. [Sweeney, 2002] addressed a very useful table-based anonymity scheme. It helps k-Anonymity algorithms provide guarantees to protect privacy with limited number of Quasi-identifiers. However, its main drawback lies in dealing only with specified personal data. That is, it focuses only on the text database and cannot be applied to media dataset directly. In our work, we imitate the idea of the formal table-based anonymity algorithm and design a novel labeling scheme for generating de-identified image set, automatically. Therefore, our work keeps the same merits as the k-Same algorithm and provides strong guarantees of privacy protection for image data.

The authors of [Meden et al., 2018] proposed the k-Same-Net scheme, whose key idea is to combine the k-Anonymity algorithm with the Generative Deep Neural Network (GNN) architectures. Although the obtained result is the state-of-the-art in this research topic, there are some weaknesses still. First, while selecting the cluster centroids (i.e., similar images), traditional PCA algorithm was used; therefore, certain amount of computation is inevitable. Second, it takes quite a long training time when GNN architecture is adopted. In other words, besides the long time spent, the required huge amount of computing and memory resources, such as GPGPUs, handicaps its applicability in practical applications. Another fatal weakness of this approach is that the original images must be down sampled during training and synthesizing, which impairs the quality of surrogate images a lot.

2.2 StarGAN

StarGAN is a multi-domain image-to-image translation framework, within a single or multiple dataset and has shown remarkable results as reported in [Choi et al., 2018]. As the name implied, the architecture of StarGAN is a modified from that of a general GNN. Target domain labels are added to its Discriminator for classifying whether an image belongs to a specific target domain, and then to the Generator for generating the desired image on the basis of the target domain. Moreover, Choi et al., also added a reconstruction process to StarGAN to make sure that the images generated by Generator can be transformed back to the original domain for preventing any non-consistency in image generation.

2.3 Siamese Net

Siamese Net is a special type of neural network architectures designed for recognizing one-shot images. Instead of being a model assimilating to classify its inputs, it learns to distinguish between two input images. A Siamese Net [Koch et al., 2015] consists of two identical neural networks, each taking care of one of the input images. The last layer outputs of the two networks are fed to a contrastive loss function, which is used to calculate the similarity of two images. During the training, the two sister networks share the weight and optimizes the contrastive loss function to make similar images as close as possible, and vice versa. In other words, the object of this network is not to classify the inputs, but differentiate between them. Hence, we use this network as our similarity arbitrator to prevent our Generator from producing images that alike to the original one too much.
2.4 Advisor - the Tool for Hyperparameter Tuning

Hyperparameter optimization is useful for finding the best model hyperparameters and selecting the proper model function. The cost paid for model tuning depends normally on the cardinality of the involved parameter set, which is a large number usually. Exhaustive search is the simplest heuristic method to tackle this problem, but it also stands for a variable amount and lengthy time spent is inevitable in the tuning process. To deal with this problem, we use the well-reputed parameter tuning tool - Advisor to search for best performed hyperparameter combinations. Advisor is an open-source implementation of Google Vizier [Golovin et al., 2017], which defined four data types for each hyperparameter feasible set (i.e., Double, Integer, Discrete and Categorical). A tuning work configuration, called Study, is assumed to have a list of parameters and the corresponding data types. Each list of the pending parameters, called Trial, will be calculated by using a delegation process, called Worker.

2.5 Mixed Precision Training

Deep Learning has enabled progresses in many different applications, ranging from speech recognition [Amodei et al., 2016] to image recognition [He et al., 2016], and from language modeling [Jozefowicz et al., 2016] to machine translation [Wu et al., 2016], just named a few. Two trends do contribute and are critical to these results: large training data sets and complex models. Although, increasing the size of a neural network will typically improve the accuracy rate and generate images with higher quality; however, it will also raise the memory and computation requirements for training the model. That’s to say, the cost to train or use the model will be grown along with the depth of the network. Therefore, to increase the value of our system in real applications, we have to find some ways to reduce our cost, especially in training. When the model demands to be cost-efficient for training or inference. The most intuitive way to build a cost-efficient model from its original complex form is to simplify the associated model architecture by pruning and/or quantization; nevertheless, in this way, relinquishing in performance is an inevitable side effect [Micikevicius et al., 2017].

MPT is the abbreviation of Mixed Precision Training, literally, MPT means there are more than one floating-point data types involved during model training. The key merit of MPT is that we can nearly halve the memory requirement and, on recent GPUs, speed up almost 2 to 3 times in the training process. Besides, with MTP, we don’t need to change any model structure and possess nearly the same performance as that of its original counterpart.

3 Implementation

3.1 The Implementation of kSS-GAN

Before discussing the proposed system in detail, as illustrated in Figure 1, an explanation of the architecture of kSS-GAN for face de-identification, is given. First of all, Radboud Faces Database (RaFD) [Langner et al., 2010] is chosen as our input image set for generating new groups of image sets. Due to the concatenation with the modified loss function of Siamese-GAN, the auto label-scheme helps generate a label for each image, according to some specific features, such as facial expression, gender, group id and so on. Finally, we can get the k-same/k-anonymity de-identified faces as the system outputs.

According to 1, kSS-GAN performs the following functions: distance measuring, image grouping (for selecting k-identity faces), auto-labelling, and concatenating with the modified StarGAN (for training with hyperparameter tuning and MPT mechanisms). In other words, the proposed kSS-GAN consists of the following three functional modules: Face Recognition module, Cluster Generating module, and Candidate Clustering modules.

The task of Face Recognition module is to align faces and call the distance measuring function to calculate the similarity of the target image to the whole image set. The Cluster Generating module is used to control the image grouping function for selecting k-identity faces and call the auto-labelling function to replace or add specific features for generating candidate clusters. This is the reason why, with the aid of auto-labelling scheme, we can generate de-identified image set automatically and flexibly. The last Candidate Clusters module is responsible for communicating with Siamese-GAN to conduct training. At the same time, the hyperparameter tuning tool, Advisor, is activated to find the most appropriate parameter combinations, and MPT is activated as well to speed up the training process and reduce the memory consumption. The schematic diagram of the whole proposed system is sketched in Figure 1.

3.2 The Implementation of Siamese-GAN

In Star-GAN [Choi et al., 2018], a generated image must go through the reconstruction process, as shown in Figure 2 (R1), to ensure its relevance to the original one. The authors of StarGAN then applied Cycle Consistency Loss, defined in [Kim et al., 2017; Zhu et al., 2017] to the generator, which is defined as

$$L_{rec} = \mathbb{E}_{x,c,c'}[\|x - G(x,c,c')\|_1]$$  (1)
where \( G \) takes the translated image \( G(x, c) \) and the original domain label \( c' \) as inputs and tries to reconstruct the original image \( x \). The L1 norm is adopted as the reconstruction loss, in Equation (1).

In our work, however, the generated images must be deviated from the original input so as to achieve the so-called K-Anonymity. The newly proposed Siamese-GAN is depicted in Figure 2. In the rest of this section, the involved feature-based similarity measure, reconstruction loss, adversarial loss, domain classification loss, and the overall loss function will be detailed in sequence.

**Feature-Based Similarity**

To overcome the shortage associated with Star-GAN, we modify its reconstruction process from (R1) to (R2), as shown in Figure 2, for ensuring that our translated image do distinguishable from the original one. Further, we apply the Cycle Consistency Loss to our generator and use the feature-based similarity kernel [Wang et al., 2017] to measure the reconstruction loss, that is

\[
S(f_{real}, f_{fake}) = \frac{1}{2}(1 + f_{real}^T f_{fake})
\]

(2)

Where \( f_{real} \) and \( f_{fake} \) are the feature maps of the real and the fake image, respectively. Since \( f_{real} \) and \( f_{fake} \) have been previously L2-norm normalized, the inner product \( f_{real}^T f_{fake} \) is actually the Cosine similarity between the two originally unnormalized feature vectors. The Cosine similarity is chosen because it is bounded and its gradient with respect to \( f_{real} \) and \( f_{fake} \), as compared with other bounded similarity functions like the Bhattacharyya coefficient, can be calculated much easily.

**Reconstruction Loss**

To get the feature vector of an image, as shown in Figure 2, we include the Siamese model [Koch et al., 2015] into our reconstruction process. Siamese model will learn facial features of the real and the fake images during the training process and then project the input and the reconstructed images back to the feature space. The so-obtained feature vectors are fed into the feature-based similarity measuring function, mentioned in the previous sub-section. On the bases of the calculated similarity and our new reconstruction loss function, as defined in Equation (3), a similarity score of the two images will be obtained.

\[
\mathcal{L}_{rec} = \mathbb{E}_{x,c,c'} [S(x, G(G(x,c), c'))]
\]

(3)

**Adversarial Loss**

To make the generated (fake) images indistinguishable from the original (real) images, we define an adversarial loss function, as given in Equation (4), where \( G \) generates an image \( G(x,c) \) according to the input image \( x \) and the target domain label \( c \), while \( D \) works to distinguish the real image from the fake one. The whole process behaves like a minmax game in the game theory. That is, the generator \( G \) attempts to minimize the distinguishable objective function, while the discriminator \( D \) tries its best to maximize it.

\[
\mathcal{L}_{adv} = \mathbb{E}_x [\log (D(x))] + \mathbb{E}_{x,c} [\log (1 - D(G(x,c)))]
\]

(4)

**Domain Classification Loss**

For a given input image \( x \) and a target domain label \( c \), we have to translate \( x \) into an output image \( y \) which is correctly classified into the target domain with label \( c \). To achieve this goal, we add an ancillary classifier into \( D \), just like StarGAN, and impose two domain classification loss functions, as defined in Equations (5) and (6), to optimize \( D \) and \( G \), respectively. The domain classification loss function for the discriminator \( D \) is defined as

\[
\mathcal{L}_{real_{cls}} = \mathbb{E}_{x,c} [\log (D_{cls}(c'|x))] 
\]

(5)

The domain classification loss function for the generator \( G \) is defined as

\[
\mathcal{L}_{fake_{cls}} = \mathbb{E}_{x,c} [\log (D_{cls}(c'|G(x,c)))] 
\]

(6)

where \( D_{cls}(c'|x) \) denotes the probability distribution over the domain labels computed by \( D \). In other words, \( G \) tries to minimize this objective function for generating images that can be classified as the target domain with label \( c \) and \( D \) learns to classify a real image \( x \) into its corresponding original domain with label \( c' \).
Algorithm 2: Siamese-GAN with gradient penalty with gradient penalty. The following default settings: $\lambda_{gp} = 10, \lambda_{rec} = 10, \lambda_{cls} = 1, n_{critic} = 5, \alpha = 0.0001, \beta_1 = 0.5, \beta_2 = 0.999$ are adopted.

**Input:** The gradient penalty coefficient $\lambda_{gp}$, the domain reconstruction loss coefficient $\lambda_{rec}$, the domain classification loss coefficient $\lambda_{cls}$, the number of critic iterations per generator iteration $n_{critic}$, the batch size $m$, and Adam hyperparameters $\alpha, \beta_1, \beta_2$.

**Parameter:** initial critic parameters $w_0$, initial generator parameters $\theta_0$.

1. **while** $\theta$ has not converged **do**
2. **for** $t = 1, ..., n_{critic}$ **do**
3. **for** $i = 1, ..., m$ **do**
4. Sample real data $x \sim \mathbb{P}_r$, latent variable $z \sim p(z)$, a random number $\epsilon \sim U[0, 1]$.
5. $\hat{x} \leftarrow c x + (1 - \epsilon) \hat{x}$
6. $L_{D(i)} \leftarrow D(\hat{x}) - D(x) + \lambda_{gp}(\|\nabla_{D, \hat{x}} D(\hat{x})\|_2 - 1)^2 + \lambda_{cls} E_{x,c}[\log D_{cls}(c(x)) - \log D_{cls}(c(\hat{x}))]
7. **end for**
8. $w \leftarrow \text{Adam}(\nabla_0 \frac{1}{m} \sum_{i=1}^{m} L_{D(i)}, w, \alpha, \beta_1, \beta_2)$
9. **end for**
10. Sample a batch of latent variables $\{z^{(i)}\}^m_{i=1} \sim p(z)$, real data $x \sim \mathbb{P}_r$.
11. $\hat{x} \leftarrow G_0(z)$
12. $L_{G(i)} \leftarrow D(\hat{x}) - D(x) + \lambda_{cls} E_{x,c}[\log D_{cls}(c(G(x, c))) - S(f_{real}, f_{fake})]
13. $\theta \leftarrow \text{Adam}(\nabla_0 \frac{1}{m} \sum_{i=1}^{m} L_{G(i)}, w, \alpha, \beta_1, \beta_2)$
14. **end while**

**Overall Loss Function**

Finally, the loss functions to optimize $G$ and $D$ are written, respectively, as

$$L_D = -L_{adv} + \lambda_{cls} * L_{realcls}$$

$$L_G = L_{adv} + \lambda_{cls} * L_{fakecls} + \lambda_{rec} * L_{rec}$$

where $\lambda_{cls}$ and $\lambda_{rec}$ respectively are hyperparameters controlling the relative importance of domain classification loss and the reconstruction loss, as compared with the adversarial loss. Without loss of generality, we use $\lambda_{cls} = 1$ and $\lambda_{rec} = 10$ in all our experiments.

### 3.3 The Implementation of MPT

As pre-described, MPT [Micikevicius et al., 2017] is used to reduce our training time and memory consumption. The key roles of MPT are the floating-point operation with 16-bit precision (FP16) and the Tensor-Core which can accelerate matrix operations and halve the GPU memory consumption. We first cast the input images to FP16 and ingest forward propagation through the model. After that, we convert the model output to FP32 for evaluating the loss and then scale the loss back to FP16 to cover wider representable range. The casting of forward model output to FP32 is a must for preserving small values that are, otherwise, quantized to zero in FP16.

### 4 Implementation Results and Evaluation of Hyperparameter Performance

#### 4.1 Preliminaries of Experiment and Dataset - RaFD

Table 1 summarizes the characteristics of our experimental environments, and a publicly available face image dataset, the Radboud Faces Database (RaFD) [Langner et al., 2010], is used to benchmark our work. RaFD contains high-quality images of 67 subjects, each of which has eight different facial expressions (i.e., anger, disgust, fear, happiness, sadness, surprise, contempt, and neutrality). RaFD is a suitable choice for training kSS-GAN because it includes aligned...
| CPU Model | CPU Memory | Frequency | # of CPUs | GPU Model | GPU Memory | # of GPUs |
|-----------|------------|-----------|-----------|-----------|------------|-----------|
| Intel(R) Xeon(R) Gold 6128 CPU | 192 GB | 3.4 GHz | 24 | Tesla V100 | 24 GB | 2 |
| Intel(R) Core(TM) i9-7980XE CPU | 64 GB | 2.6 GHz | 1 | Quadro GP100 | 16 GB | 1 |

Table 1: Characteristics of Our Experimental Environments.

4.2 The Evolution of Hyperparameter Performance

Three specific solvers of the open-source tool Advisor are used to test the proposed system, including Particle Swarm, Random Search, and Hyperopt [Golovin et al., 2017]. Since the Cluster Generator module in kSS-GAN is to choose specific features for generating proper candidate clusters, we use Advisor to search for better hyperparameter combinations. As illustrated in Figure 4, the x-axe stands for the number of tuning iteration and the y-axe denotes the value of the loss function; clearly, Particle Swarm and Hyperopt provide much lower loss than that of the Random Search. After getting the hyperparameter combination associated with the minimum loss, we use that particular parameter combination to train our model. For comparison, as shown in Figure 5, we take the loss of the default combination (i.e., without using any solver) into account. Based on the resultant loss of the training process, the combinations of hyperparameter recommended by Advisor do provide lower system loss than that of the naïve approach.

4.3 Experimental Results - Mixed Precision Training (MPT)

To illustrate the effects of MPT, we trained our model with the two different GPU platforms listed in Table 2, which are Volta V100 and Pascal P100, respectively. The memory consumption before including MPT is quite high; actually, the corresponding memory utility rate is approaching to 100%. In contrast, by using MPT, the memory utility rates of our system reduce to 36% for V100 and 37% for P100, respectively. In MPT, FP16 is one of the computing data types which enabled us to leverage the extremely high speed FP16 matrix operations in Tensor-Core. Roughly speaking, with FP16 computing in TensorCore, the training process can be speeded up almost 1.5x, as compared without using MPT in V100. Although not every GPU has Tensor-Core, such as P100, MPT can still reduce lots of memory consumption in training.

4.4 The Evolution of kSS-GAN for Face De-identification

For checking the effectiveness of kSS-GAN in facial image de-identification, we compare the quality of the generated (fake) images of K-Same-M, K-Same-Net, and kSS-GAN, in this Section. For the same criteria k=3, we listed the original and the generated images for K-Same-M, K-Same-Net, and kSS-GAN (from top to bottom) in Figure 6. One should aware of that the schemes differ not only in the generated images’ visual quality but also in the amount of information content retained.

In other words, the quality (represented by the naturalness and the smoothness) of the generated image plays the dominating role, if privacy protection (i.e., facial image deidentification) is the only application target. However, under certain application scenarios the identity re-identification is a must, then besides image quality, the amount of retained information associated with the original image becomes very crucial. For example, under the GDPR regulation, the storage of any Personal Identifiable Information (PII) data (such as facial images) must be protected carefully. Because of the maturity of Face Recognition technology, recently, facial image becomes the most widely used biometric feature for authenticating one’s identity (ID). Therefore, if a bank provides bank-
| Device          | Method     | Performance | GPU-RAM Consumption | Training Time  | Speed up | Lightweight |
|-----------------|------------|-------------|---------------------|----------------|----------|-------------|
| Tesla V100      | No MPT     | N/A         | Run out of memory   | N/A            | N/A      | N/A         |
| Tesla V100      | With MPT   | 3-4 s/10iter| 13.3GB/16GB (83.2%) | 19h 24m        | >1       | -           |
| Quadro GP100    | No MPT     | N/A         | Run out of memory   | N/A            | N/A      | N/A         |
| Quadro GP100    | With MPT   | 7 s/10iter  | 12.5GB/16GB (77%)   | 1d 14h 48m     | 1        | -           |

Table 2: The Characteristics and Performances of the Proposed kSS-GAN.

In terms of image quality, as shown in Figure 6, k-same-Net does provide the best result. However, its re-identification capability has never been addressed. Our approach, as shown in the bottom of Figure 6, achieves the goals of providing anonymity guarantee, producing nearly natural and realistic de-identification results, and supporting the ability for re-identifying the original ID if necessary. That is, even though the details of our generated facial image still have some defects, especially in the regions with lights and shadows, which impair the quality of our output image a few. Nevertheless, from Figure 6, we can see that the proposed kSS-GAN provides the highest possibility of ID re-identification from the output images. Moreover, we also find that there are no ghosting effects in our generated images. Thus, we apply Blurry Compression Test to compare the three k-same approaches with the blur detection from OpenCV [Adrian Rosebrock, ]. Figure 7 shows the comparison results, where the x-axe gives the value of k, and y-axe represents the degree of sharpness (i.e., the variance of Laplacian values). The higher the degree of sharpness is the more the clearness will be. From Figure 7, the image generated by kSS-GAN is the clearest one in each level of the same k.

5 Conclusion and Future Works

In this work, we proposed a novel approach towards facial image de-identification, called kSS-GAN, which leverages the techniques of k-Same-Anonymity mechanism, Generative Adversarial Network, and hyperparameter tuning. In order to speed the whole training process up and reduce the memory consumption, the well-performed mixed precision training and hyperparameter tuning are also included in the proposed system. From the experimental results, we proved that kSS-GAN achieved the goals of providing anonymity guarantee, producing nearly natural and realistic de-identification results, and supporting the ability for re-identifying the original ID if necessary. In the future, we will apply our approach
to video streams for generating the surrogate images in real time; in other words, we will tackle the following critical issues, such as real time object detection from videos, enhancing the quality of images generated by KSS-GAN, and finding effective methods to provide higher model compression. So far, we have realized a proof of concept (POC) prototype system, as illustrated in Figure 8. The bottom window shows the real image, the left depicts the referencing input image to our approach, and the top gives the synthesized agent image.

References

[Adrian Rosebrock, ] Adrian Rosebrock. Blur detection with opencv. https://www.pyimagesearch.com/2015/09/07/blur-detection-with-opencv/. Accessed: 2019-02-25.

[Amodei et al., 2016] Dario Amodei, Sundaram Ananthanarayan, Rishita Anubhai, Jingliang Bai, Eric Battenberg, Carl Case, Jared Casper, Bryan Catanzaro, Qiang Cheng, Guoliang Chen, et al. Deep speech 2: End-to-end speech recognition in english and mandarin. In International conference on machine learning, pages 173–182, 2016.

[Choi et al., 2018] Yunjey Choi, Minje Choi, Munyoung Kim, Jung-Woo Ha, Sunghun Kim, and Jaegul Choo. Star-gan: Unified generative adversarial networks for multi-domain image-to-image translation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 8789–8797, 2018.

[Gellman, 2010] Robert Gellman. The deidentification dilemma: a legislative and contractual proposal. Fordham Intell. Prop. Media & Ent. LJ, 21:33, 2010.

[Golovin et al., 2017] Daniel Golovin, Benjamin Solnik, Subhodeep Moitra, Greg Kochanski, John Karro, and D Sculley. Google vizier: A service for black-box optimization. In Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 1487–1495. ACM, 2017.

[Gross et al., 2006] Ralph Gross, Latanya Sweeney, Fernando De la Torre, and Simon Baker. Model-based face de-identification. In 2006 Conference on Computer Vision and Pattern Recognition Workshop (CVPR’06), pages 161–161. IEEE, 2006.

[Gross et al., 2008] Ralph Gross, Latanya Sweeney, Fernando De La Torre, and Simon Baker. Semi-supervised learning of multi-factor models for face de-identification. In 2008 IEEE Conference on Computer Vision and Pattern Recognition, pages 1–8. IEEE, 2008.

[He et al., 2016] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 770–778, 2016.

[Jozeffowicz et al., 2016] Rafal Jozeffowicz, Oriol Vinyals, Mike Schuster, Noam Shazeer, and Yonghui Wu. Exploring the limits of language modeling. arXiv preprint arXiv:1602.02410, 2016.

[Kim et al., 2017] Taeksoo Kim, Moonsu Cha, Hyunsoo Kim, Jung Kwon Lee, and Jiwon Kim. Learning to discover cross-domain relations with generative adversarial networks. In Proceedings of the 34th International Conference on Machine Learning-Volume 70, pages 1857–1865. JMLR. org, 2017.

[Koch et al., 2015] Gregory Koch, Richard Zemel, and Ruslan Salakhutdinov. Siamese neural networks for one-shot image recognition. In ICML Deep Learning Workshop, volume 2, 2015.

[Langner et al., 2010] Oliver Langner, Ron Dotsch, Gijsbert Bijlstra, Daniel HJ Wigboldus, Skyler T Hawk, and AD Van Knippenberg. Presentation and validation of the radboud faces database. Cognition and emotion, 24(8):1377–1388, 2010.

[Letournel et al., 2015] Geoffrey Letournel, Aurélie Bugeau, V-T Ta, and J-P Domenger. Face de-identification with expressions preservation. In 2015 IEEE International Conference on Image Processing (ICIP), pages 4366–4370. IEEE, 2015.

[Meden et al., 2018] Blaž Meden, Žiga Emeršič, Vitomir Štruc, and Peter Peer. k-same-net: k-anonymity with generative deep neural networks for face deidentification. Entropy, 20(1):60, 2018.

[Meng et al., 2014] Lily Meng, Zongji Sun, Aladdin Ariaieenina, and Ken L Bennett. Retaining expressions on de-identified faces. In 2014 37th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), pages 1252–1257. IEEE, 2014.

[Micikevicius et al., 2017] Paulius Micikevicius, Sharan Narang, Jonah Alben, Gregory Diamos, Eric linen, David Garcia, Boris Ginsburg, Michael Houston, Oleksii Kuchaiev, Ganesh Venkatesh, et al. Mixed precision training. arXiv preprint arXiv:1710.03740, 2017.

[Newton et al., 2005] Elaine M Newton, Latanya Sweeney, and Bradley Malin. Preserving privacy by de-identifying face images. IEEE transactions on Knowledge and Data Engineering, 17(2):232–243, 2005.

[Samarzija and Ribaric, 2014] Branko Samarzija and Slobo Dan Ribaric. An approach to the de-identification of faces in different poses. In 2014 37th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO), pages 1246–1251. IEEE, 2014.

[Sweeney, 2002] Latanya Sweeney. k-anonymity: A model for protecting privacy. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, 10(05):557–570, 2002.

[Wang et al., 2017] Jin Wang, Zheng Wang, Changxin Gao, Nong Sang, and Rui Huang. Deeplist: learning deep features with adaptive listwise constraint for person reidentification. IEEE Transactions on Circuits and Systems for Video Technology, 27(3):513–524, 2017.
[Wu et al., 2016] Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. Google’s neural machine translation system: Bridging the gap between human and machine translation. arXiv preprint arXiv:1609.08144, 2016.

[Zhu et al., 2017] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In Proceedings of the IEEE International Conference on Computer Vision, pages 2223–2232, 2017.