2021 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/IOP/SCI) | 978-1-6654-1236-0/21/$31.00 ©2021 IEEE | DOI: 10.1109/SWC50871.2021.00053

Noise Generation GAN Based Identity Privacy Protection for Smart City

Jishen Yang*, Yan Huang†, Madhuri Siddula‡, Zhipeng Cai§
*Department of Computer Science, Georgia State University, Atlanta, GA, USA
†Department of Software Engineering & Game Development, Kennesaw State University, Atlanta, GA, USA
‡§Department of Computer Science, North Carolina Agricultural and Technical State University, Greensboro, NC, USA
jyang57@student.gsu.edu*, yhuang24@kennesaw.edu†, msiddula@ncat.edu‡, zcai@gsu.edu†

***Abstract*—The development of Internet of Things (IoT) infrastructure in the city leads to the emergence of the concept of smart city, an integrated solution to provide convenience for various applications in our daily life by understanding and analyzing the collected data from multi-sources. However, the collection of facial images collected from various IoT devices such as surveillance cameras, wearable, and mobile devices increases the risk of an individual's privacy leak. The facial recognition models augment this risk. These models retrieve facial data collected from IoT devices stored in smart city databases to get personal identity information. With extensive utilization of such IoT devices, which serve as a visual data collector, we compromise the person's identity. Therefore, to protect the privacy of image data from a database, we propose a Sensitivity Map Noise-Adding model based on generative adversarial networks to provide privacy for facial images against the malicious use of the face recognition models. The proposed models work as a black-box model that does not require any architectural information or the parameters of the target model. Additionally, the model runs at a real-time speed and the average run time for one operation is less than 12 milliseconds. The protection can be deployed for both local images and streaming videos. The data privacy protection is based on our proposed concept of the Sensitivity Maps, which summarizes the effectiveness and efficiency of adding noises on each pixel on the original image to interfere with the target model's performance. We have built a new dataset of facial images containing 102 celebrities for the proposed model to be trained and evaluated. The experimental results prove the advantage of the proposed method against protecting the identity information in facial images.**

***Index Terms*—Privacy, GAN, Face Recognition**

I. Introduction

With the beginning of the new information era, there has been a surge in IoT infrastructure. Many IoT devices with embedded sensors are currently deployed at every corner of the city, creating massive streams of real-time data. Additionally, with the rapid increase in computation power and the boom in data volume, researchers have developed numerous advanced artificial intelligence mechanisms to improve the efficiency of information processing. By combining the power of automatic information processing and escalation in IoT infrastructure, we enter the smart things era. A smart city is where we combine the data from various sensors in the city to understand, manage, and interpret information [20] [14] [8]. In the recent past, a lot of applications of smart city are proposed by the researchers in different fields such as intelligent transportation systems [24] [6], health condition monitoring systems [38] [7], and public safety and security solutions [13] [39] [32]. Among these applications, one of the most common and important data types is visual data [42] [13] [9] [5]. Recent facial recognition techniques have shown an excellent improvement in the performance and stability as they tend to use deep learning algorithms [28] [36] [17] [30]. Therefore, the facial images captured by the multi-camera video surveillance system and mobile smart devices can significantly improve applications such as human tracking and sensing [41], health monitoring [25], face payment, and public safety monitoring.

Although using a facial recognition system is beneficial and convenient, we are often posed with the risk of revealing personal identity information [11]. The fact that the efficiency of these recognition techniques is increasing escalates the risk of privacy leaks. If such image information is leaked, with this knowledge of a person, the scammers would perform a very persuasive and undiscernible fraud to the victim after they located the victim's identity from their facial images. As described in Fig. 1, without any protection of the original photos, the face recognition models can easily derive the identity of a user by pairing the faces contained in the image data with the face records from other sources like photo ID or social networks. Subsequently, with the development of smart city applications, the malicious utilization of artificial intelligence models intensifies the leak of sensitive information from public data.

An effective information encryption model for visual data against artificial intelligence models is imperative to solve the privacy concerns and protect the confidential content in visual data. The purpose of a visual private information protection model is to hide the sensitive information contained in the image data from the face recognition models but not from human observers. Thus, the procedure conducted has to alter the raw image precisely and yield an influential impact on the accuracy of the face recognition models. Then, the identity information contained in the image data is kept confidential from the computer vision models without evident alteration. For example, consider a scenario where apartments or other public areas equip cameras such as smart doorbells and security surveillance that record all the residents' and visitors' faces. By using the collected data and face recognition models, a

978-1-6654-1236-0/21/$31.00 ©2021 IEEE
DOI 10.1109/SWC50871.2021.00053

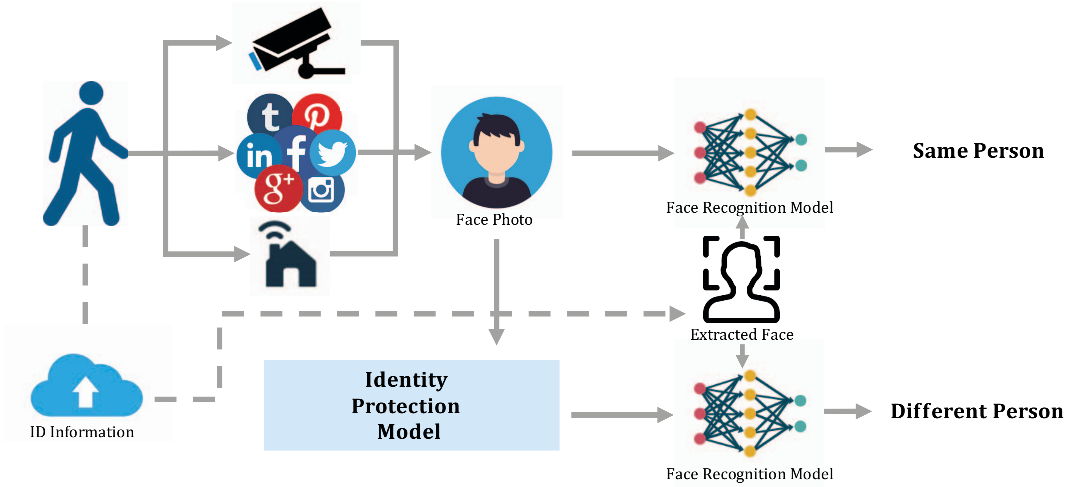

Fig. 1. The illustration of the identity protection model working on the captured face image. The face recognition models can infer the identity of the captured image by comparing the photo with the ID or other photos online. After the modification by the identity protection model, the target face recognition models should no longer be able to obtain the identity information of the captured image.

platform can discover the visitors' identities by comparing the recorded faces with labeled public visual data online. Additionally, any party with these data can track the peoples' routines and build a complete profile by sharing the data with other parties. Fortunately, all the loss caused by the privacy issue can be avoided if the camera of the digital doorbells could deploy a privacy protection system for face photos while all the rest of the services remaining functional.

There are different methods proposed in this field. Some works are proposed on the idea of adversarial example attack [10], [26]. These protection algorithms work as a white box on human faces assuming that the entire target face recognition model's structure and parameters are known, which is not commonly possible. Some other methods utilize the idea of exhaustive search to find an optimal noise to the target model with constraining the total magnitude of the alteration [31]. However, the time cost for such methods is too high to deal with the live streaming visual data. Besides, some methods use generative adversarial networks (GAN) [12], [18] model to generate disturbing noise. However, the training state needs queries to the target models or substitute models [16] [43]. In an ideal visual privacy protection model, the process should only work on raw image data without directly accessing or querying the face recognition models. Besides, a good privacy protection model requires as minimum as possible auxiliary information of any face recognition models to be applicable for universal purposes. Most importantly, the model should process the data fast enough to process the streaming data or deal with a mass amount of data in a short time.

In this paper, to achieve a better identity privacy protection model for visual data, we propose the Sensitivity Map Noise-Adding (SMNA) model. The proposed SMNA model is able to protect the identity information contained in images from face recognition models by adding inconspicuous noises in real-time speed and black-box setting.

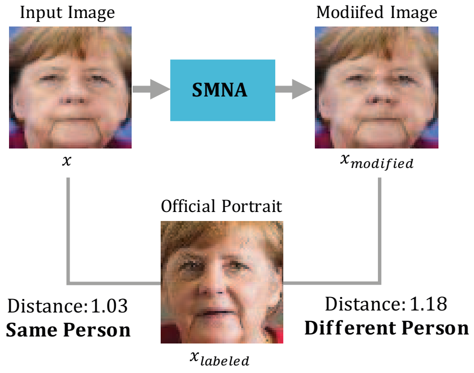

Fig. 2. Demonstration of the proposed model. After the protection process, the original image x is no longer recognized as the same person with the official portrait $x_{labeled}$ by the target face recognition model.

The overview of the protection process is illustrated in Fig. 1, where the identity information protection model works on the visual data from various sources. After the protection procedure, the identity information in the original data becomes obscure to the target face recognition models with minor noise added. So, we propose the SMNA model to play the role of the identity protection model as shown in Fig. 2. The original image x passes through the SMNA model and is transformed into $x_{modified}$ with added perturbations. Then the modified image $x_{modified}$ fools the target face recognition model to misclassify the data into a different person. In this way, the identity information contained in the image avoids being unjustifiably acquired.

In our design, the perturbations are produced based on our newly introduced concept of the sensitivity of pixels. The sensitivity of a pixel is a metric to measure how sensitive the change of the target model's final output confidence score is to the pixel's RGB value adjustments. In other words, the pixel's sensitivity is its influence on the overall output and efficiency to add noise. The sensitivity map of an image is a matrix of the same size as the image, and each element represents the sensitivity of the corresponding pixel. By this design, the sensitivity map well presents each pixel's importance to the final prediction of the target model. Also, the sensitivity maps can be visualized if turned into grey scale images as shown in Fig. 3.

Adding noise to the original image by the weights of the image's sensitivity map is effective and efficient, but the calculation of sensitivity maps demands a high time cost and a large number of queries to the target model. Hence, to avoid the drawbacks, we use the structure of GAN to generate the sensitivity maps swiftly and locally without any connections to the target model or the knowledge of the structure and parameters of the target model. In the proposed system, the noise generator runs entirely like a black-box and can work on the raw image data free from accessing the target model after and during the training. To adequately support the training and testing for the proposed model in a realistic scenario, we construct an entirely new dataset of faces with a restriction that each person has a reference photo of an official portrait from the cover photo in an authoritative webpage. The dataset includes 102 celebrities, and each person has 21 photos, including a labeled portrait and 20 other photos for testing or training use. Notably, for the entire system including the stage of forging the dataset, the need for knowledge of the architecture and the parameters of the target model is avoidable. The algorithm runs at a real-time speed and consumes acceptable computation power and thus is feasible to be deployed in various real-life scenarios.

The main contributions of this work are highlighted as follows:

- We propose a Sensitivity Map Noise-Adding (SMNA) model, a novel noise generating model to protect identity information from being inferred by deep learning face recognition models. The model provides privacy protection on face image data by reducing the prediction accuracy in a complete black-box mode without requiring any information from and queries to the target models.
- We introduce the concept of the sensitivity map to demonstrate the degree of sensitivity of pixels on original face images. The conversion of the original image and its corresponding sensitivity map is a procedure of projection from the distribution of RGB images to the distribution of sensitivity maps under given conditions. The transformation is learnable by using GAN models.
- We build an image dataset of faces from 102 celebrities with each celebrity's official portrait and 20 other images from social networks or media. The dataset can be employed for training and testing protection strategies against different target models. With the built dataset, the simulation validated that the proposed model can avoid identity information leakage from the face images.

II. Related Work

In this section, we review the literature work that is related to the proposed work.

A. Generative Adversarial Network

In 2014, Ian Goodfellow et al. proposed a novel learning model, generative adversarial networks (GAN) [18]. The network design enables the model to understand the training data distribution and generate data samples with the knowledge. GAN has two components: the generator and the discriminator. The generator is trained to fool the discriminator to misclassify generated samples as real samples, and the discriminator aims to distinguish between the generated samples and the real samples. Many studies demonstrate the performance of the GAN and the applications of GAN are now in diverse areas. Later, the researchers utilize GAN as the tool to perform black-box attacks on deep learning models. Hu et al. [21] and Xiao et al. [40] develop the adversarial example generators based on GAN in different areas. In 2019 song et al. [33] build a GAN network to generate the altered face to fool the face recognition models. The performance of the method does not reach a very high level.

B. Adversarial Example Attack

The research on the adversarial example attack to neural network models has seen much development in recent years. The idea of an Adversarial Example Attack is to reduce the performance of deep learning models significantly by adding noise on an input image that a human eye can hardly noise. Szegedy et al. [37] in 2014 discovered that many neural networks are vulnerable to *adversarial examples*. The term *adversarial examples* are the intentionally modified data samples based on normal raw data to fool the target models. Many state-of-art neural network models misclassify these instances into wrong label categories attributed to the imperceptible perturbations. Goodfellow et al. [19] proposed the Fast Gradient Sign Method to generate the adversarial examples relatively fast against the target models. However, the method requires the target model's structural information and full parameters, which is normally inaccessible. Nguyen et al. [26] in their 2015 work introduced a novel approach to conducting adversarial example attacks to fool the deep learning models. The examples are not developed from the actual raw images with slight modifications. Instead, the attack model is based on evolutionary algorithms, and the generated examples are not understandable by a human. Nonetheless, the learning models recognize the examples and classify bizarre images into different categories with high confidence scores. The work reveals the feasibility of misleading the deep learning models with delicately built data samples. However, most of the adversarial example generation processes require the full knowledge of the target model's structure and parameters. This strong assumption does not stand when the protection is designed to use on human faces against face recognition models since the target model is unknown and inaccessible directly.

C. Other Methods

Su et al. [35] proposed One Pixel Attack that interferes with the prediction of target classification neural network changing several to even just one pixel's RGB value in an input image. Under the extreme setting, the attack applies modification on only one pixel on the original image, and the noise causes the target model to make mistakes. However, it is a more heuristic idea than a practical solution. Only on a relatively low-resolution image input setting, 32 by 32 pixel image from $Cifar10$ [22] dataset, the model illustrates the best performance. On high-resolution images, the attack needs a longer length of perturbation and more pixels to be changed. The method uses differential evolution as the optimizer, and the optimization is not efficient. The attack is in a black-box manner and demonstrates the feature of many neural networks that some parts of the data are of more importance than others. Papernot et al. [27] proposed the concept of *Saliency Map* to illustrate the unbalance of the importance of the noises on different locations on the input images to the final performance of the target model. The concept is similar to our definition of the sensitivity map, but the *Saliency Map* is calculated based on the information of the target model's architecture and parameters. Also, the authors demonstrated the correlation of the amount of information known about the target model and the difficulty of successfully conducting an adversarial example attack. Later in 2017, the authors proposed a method running in a black-box manner by building a substitute model. Although the method shows competitive performance on MNIST [23] and GTSRD [34] dataset, building a substitute model is not always an optimal bypass because of the high costs, especially when the target models get more complicated. The very recent work of Shan et al. [31] illustrates the effective protection for identity information against the deep learning face recognition models. The methodology uses the feature vectors extracted from images of the closest wrong categories to form the noise to add on the target images. Besides the high performance, the process takes a relatively long time cost.

The mentioned methods for the privacy protection on visual data are either not practical for industrial use or not specially designed for identity information protection on facial images. Therefore, we propose a method that addresses the demand of identity privacy and is pragmatic for deployment requirements, which need the model to run real-time and in a black-box manner.

III. System Model

In this section, we introduce the optimization goal and approach for the identity protection process. Then, we explain the proposed noise-adding scheme, Sensitivity Map Noise-Adding (SMNA), to prevent malicious machine learning (ML) based computer vision models from gaining too much identity information from users' image data.

A. Problem Description

With the increased use of social media and everyone sharing their images, identity information leakage has escalated.

Many personal photos are uploaded by the users to the cloud for various reasons, including personal entertainment, daily life recordings, and social connections. Users also utilize the services of automatically synchronizing the photos to the service providers without users' explicitly allowing them every time. Additionally, numerous live surveillance cameras capture people's photos or record videos and upload to online data processors regardless of people's privacy concerns.

Admittedly, everyone has official profiles publicly accessible online, either a personal web page, a contact information page on the website of company where he works, or a social media account on Facebook or LinkedIn. This form of identity disclosure has been published by the users. However, the openly published profile photos are a key source of information for artificial intelligence models to discover a person's identity. Thus, the identity information loss from image data turns into the problem of how to prevent the high-performing face recognition models from accurately pairing the faces in the people's private images and the faces from other sources.

From this perspective, we propose Sensitivity Map Noise-Adding (SMNA) model to prevent malicious ML-powered computer vision models from gaining too much identity information based on the users' image data. The proposed noise-adding scheme generates noise on the photos, where the users do not want their identity to be recognized, to obstruct the functions of face recognition models. The unauthorized face recognition models to be protected from are referred to as the target models. The SMNA generates a subtle noise to add on the original face image, and the generated noise is hardly perceived by human observers. However, such a process causes an influential impact on the performance of the unauthorized face recognition models.

We define the objective of the proposed model as an optimization problem. The term to be optimized (minimized in this case) is the target face recognition models' output confidence score of comparison between two photos of the same person (one labeled photo and one photo processed by the noise-adding model).

We set the original image that needs privacy protection as x with the dimension of $H \times W \times 3$ where each of the elements is a pixel. The protection is performed directly and independently on x, where the target face recognition model is denoted by f. To reveal the identity information from x, the model f compares x with labeled image $x_{labeled}$ and decides whether to assign x the same label by a confidence score $f(x, x_{labeled})$. The confidence score $f(x, x_{labeled})$ indicates the closeness between the features extracted from the faces in $x_{labeled}$ and x. The target model f accounts the faces as an identical person if the $f(x, x_{labeled})$ is over a pre-set threshold, and therefore, the identity information from x is insecure. To secure the identity privacy, the proposed privacy protection model generates a noise vector $e(x)$ to be added on the original target image x, thus reducing the confidence score $f(x + e(x), x_{labeled})$ compared to $f(x, x_{labeled})$.

In our case, we aim to develop a universal noise generator

e for all images. The distortion needs to be in a limited size defined by an adjustable threshold δ. Also, the potency of the perturbation is expected to be maximized with the limitation of the magnitude of noises. Thus, the optimization goal can be formulated as Equation 1.

$$\begin{aligned} &\text{maximize}[f(x, x_{labeled}) - f(x + e(x), x_{labeled})] \\ &\qquad\qquad\qquad\qquad\text{where } e(x) \leq \delta \end{aligned} \tag{1}$$

B. Sensitivity Map

To generate noises effectively and constrained size, we introduce a new concept that assists the model in determining the intensity of the noises added to different locations on the image data at the pixel level. We propose our definition of conception of the sensitivity map. Each of the pixels in one input image has a different importance level for the final output, which is the pixel's sensitivity of noise to the target model.

The sensitivity map of the image x is denoted as $SM(x, x_{labeled}, \sigma, f)$ where σ is the hyper-parameter to set the type of noise adding on each of the pixel of the image x.

The value of each pixel in the sensitivity map is the sensitivity of the pixel on the original image x, that measures the weight and scale for the noise to be added on the original image. The algorithm to calculate $SM(x, x_{labeled}, \sigma, f)$ is explained in Algorithm 1. H and W are the height and width of the shape of x, and the SM is initialized as a matrix with the shape of $H \times W \times 1$. Each element of the SM matrix is noted as a and $a_{h,w}$ represents the element at row h and column w. The SM generating process runs on every pixel of the original image x and makes a modified image $x_{modified}$ by adding noise on the specific pixel. Then, the algorithm makes a query to f and records the output confidence score $f(x_{modified}, x_{labeled})$. line 4 of Algorithm 1 calculates the value of sensitivity $s_{h,w}$ as the change of confidence score by f before and after a noise is applied to pixel at h, w. The calculation of $SM(x, x_{labeled}, \sigma, f)$ is complete as every pixel is assigned its corresponding sensitivity.

Algorithm 1 Calculation of ground truth SM.

```
Input: $x$, $x_{labeled}$, $\sigma$ and $f$.
Output: $SM(x, x_{labeled}, \sigma, f)$
1: for $h \leftarrow 0$ $H$ do
2:     for $w \leftarrow 0$ $W$ do
3:         $x_{modified} \leftarrow x + \sigma$ Noise at the location $(h, w)$
4:         $s_{h,w} \leftarrow f(x, x_{labeled}) - f(x_{modified}, x_{labeled})$
5:     end for
6: end for
```

C. Sensitivity Map Generating GAN

The calculation of one single sensitivity map needs queries to the target model, and the number of queries is equal to the number of pixels of the original image. The larger size one image has, the longer time and the larger computation consumption the calculation costs. Thus, we introduce the Sensitivity Map Generating GAN(SMGG) for creating sensitivity maps of images quickly without any queries or other forms of contact to the target model. The SMGG model is different from regular traditional GAN models, which can only generate data samples with random noises. Specifically, the input of the SMGG's generator is the original image x, and the output is the predicted sensitivity map for the image. In this way, the SMGG model avoids the requirements for the knowledge and accessibility of the target model and can generate the sensitivity map in a black-box manner and in real-time. The generator of the GAN, as shown in Fig. 4, has a U-shaped structure to scale down and up the feature maps between the convolution layers. The reduction and increase of the feature map help the generator extract the features of the input image in both shallow and deep levels. The input layer takes the image data is of size $H * W * 3$, which is the data shape of an RGB original image. Then, the convolutional layers keep down-sampling until a bottleneck as the layer of $Conv6$ in Fig. 4. After the bottleneck, the outputted feature map is sent to the up-sampling layers to be reconstructed into the original image size. Notably, the U-shaped architecture uses skip connection, that each of the up-sampling layers takes the concatenated combination of the feature map from the previous layer and the feature map

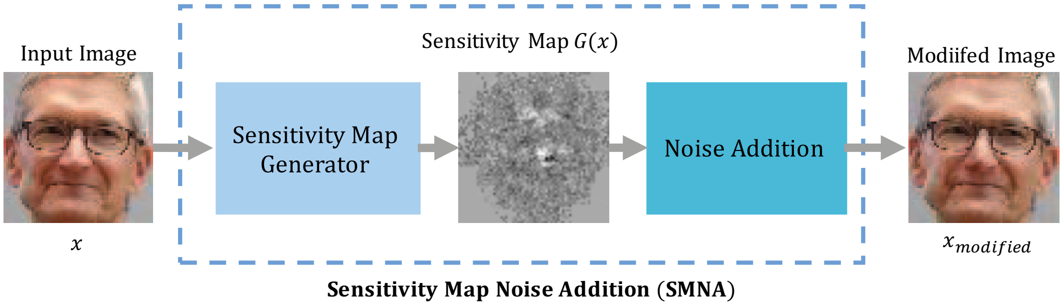

Fig. 3. The process of the privacy protection of the proposed model. The input image x is an original RGB image. During the privacy protection process, the sensitivity map of the input image $G(x)$ is generated. Finally, the noise is added to the original image with the $G(x)$ as the weights.

from the corresponding down-sampling layers as the input. By this design, the generator implements an encoder in the first half of the U structure to extract the deep and shallow features from the input image, and the outputted feature maps of down-sampling layers store the feature information and send them to the later parts of the net. The second half is a decoder that reconstructs the image using the features extracted by the encoder. Also, the decoder learns to deceive the discriminator, so the final output predicted sensitivity map has sufficient resemblance to the ground truth sensitivity map.

The discriminator plays the key role that helps the generator learn the transformation process of the original image to its sensitivity map. Fig. 4 shows the design of the GAN and the connections between the generator and the discriminator. We denote the ground truth sensitivity map $SM(x, x_{labeled}, \sigma, f)$ by y, the raw images by x, and the generated sensitivity map by $G(x)$ The optimization goal of the discriminator is to successfully distinguish the ground truth sensitivity maps and the predicted ones by the generator. The output of the discriminator is a decision matrix indicating the confidence score of the sensitivity map to be from the original distribution. If the input to the discriminator is the combination of the original image x and the ground truth sensitivity map y, the elements in the output matrix $D(x, y)$ is trained to be close to 1. On the contrary, with the input of the original image x and the sensitivity map predicted by the generator $G(x)$, the elements in the output matrix of the discriminator $D(x, G(x))$ is trained to be close to 0.

In the training steps of the SMGG model, the sensitivity maps are the labels for the GAN to be trained in a supervised setting. The loss function of the discriminator is similar to those of the regular GANs as shown in equation 4, where A and B are the matrices of 1s and 0s with the same shape as the discriminator's output matrix. The loss function of the generator has one more item besides the normal generator loss. The total generator loss is the sum of two parts as shown in equation 7. The first part is the binary cross entropy of 1 and $D(x, G(x))$, the output of the discriminator, which takes a predicted sensitivity map in input tuple. The second part is the $L1$ Manhattan distance loss between the predicted sensitivity map $G(x)$ and the ground truth y of the input image. The λ is the hyper-parameter to adjust the weight of the $L1$ loss to balance of the importance of losses for the better training result.

$$\mathcal{L}_{real} = E_x[CrossEntropy(D(x, y), A)] \tag{2}$$

$$\mathcal{L}_{generated} = E_x[CrossEntropy(D(x, G(x)), B)] \tag{3}$$

$$\mathcal{L}_D = \mathcal{L}_{real} + \mathcal{L}_{generated} \tag{4}$$

$$\mathcal{L}_{GAN} = E_x[CrossEntropy(D(x, G(x)), A)] \tag{5}$$

$$\mathcal{L}_{L1} = E_x[|G(x) - y|] \tag{6}$$

$$\mathcal{L}_G = \mathcal{L}_{GAN} + \lambda * \mathcal{L}_{L1} \tag{7}$$

The SMGG learns the projection relationship between the original images and their sensitivity maps under certain circumstances as the preset other parameters in the sensitivity map along with the image x.

Algorithm 2 The Training Process of SMGG.

```
1: while Not converging do
2:     Sample a minibatch of the data tuples $[\mathbf{x}, \mathbf{y}]$ from the dataset.
3:     Generate adversarial examples $G(\mathbf{x})$ by the generator $G$.
4:     Calculate the decision matrix $D(\mathbf{x}, \mathbf{y})$ and $D(\mathbf{x}, G(\mathbf{x}))$ by the discriminator $D$.
5:     Update the generator's weight $\theta_g$ by descending along the gradient $\nabla_{\theta g} Loss_G$.
6:     Update the discriminator's weight $\theta_d$ by descending along the gradient $\nabla_{\theta d} Loss_D$.
7: end while
```

D. Training of SMGG

The data for training the SMGG to generate sensitivity maps is formed by pairs of images x and its corresponding ground truth sensitivity map y calculated by direct queries to a face recognition model. y works as the label to the image and helps the SMGG to understand the pattern of translating data from the space of RGB images into the space of sensitivity maps. The dataset is the essential source that the SMGG learns the features of the images and features of target face recognition models' systematic operations on images generally. Algorithm 2 show the detailed process of training SMGG. For each training step, we first sample a mini-batch of data as tuples of a target image and its sensitivity map ($[x, y]$ as shown in line 1). Then, the generator G takes the input of x from the data sample $[x, y]$ to forge the fabricated sensitivity map $G(x)$. With input $[x, y]$ and $[x, G(x)]$, the discriminator D outputs different decision matrix $D(x, y)$ and $D(x, G(x))$ respectively. Next, we calculate the loss for D and G according to equations [2-7]. Then, the weights θ_g of G and θ_d of D are updated according to the losses by backpropagation. The trained SMGG can produce the sensitivity map of a given image for the assured noise type and the target face recognition model.

E. Noise Addition

The final step of the SMNA model is to add a subtle noise to the original image to be protected. The integrated noise addition algorithm is illustrated in Algorithm 3. The noise is mainly decided by the predicted sensitivity map $G(x)$ for both the noise type and the weights of noise on each pixel. We first define a $NoiseMap$ as the weight matrix of noises by the normalized $G(x)$. Then, the hyper-parameter δ is used to adjust the overall density of the noise as defined in equation 1. Also, the $NoiseLevel$ is defined as an indicator of the heaviness of overall noises, calculated as δ over the absolute value of $NoiseMap$. The adjustment of hyper-parameter is a trade-off game. A higher δ makes the noise more visible to the human

TABLE I
THE ARCHITECTURE OF THE GENERATOR

Layer Name	Layer Operation	Feature Map Shape
Conv1	$4 \times 4 \times 64$, Conv2D, LeakyReLU	$32 \times 32 \times 64$
Conv2	$4 \times 4 \times 128$, Conv2D, BatchNorm, LeakyReLU	$16 \times 16 \times 128$
Conv3	$4 \times 4 \times 256$, Conv2D, BatchNorm, LeakyReLU	$8 \times 8 \times 256$
Conv4	$4 \times 4 \times 512$, Conv2D, BatchNorm, LeakyReLU	$4 \times 4 \times 512$
Conv5	$4 \times 4 \times 512$, Conv2D, BatchNorm, LeakyReLU	$2 \times 2 \times 512$
Conv6	$4 \times 4 \times 512$, Conv2D, BatchNorm, LeakyReLU	$1 \times 1 \times 512$
Conv7	$4 \times 4 \times 512$, Conv2DTranspose, BatchNorm, LeakyReLU	$2 \times 2 \times 1024$
Conv8	$4 \times 4 \times 512$, Conv2DTranspose, BatchNorm, LeakyReLU	$4 \times 4 \times 1024$
Conv9	$4 \times 4 \times 256$, Conv2DTranspose, BatchNorm, LeakyReLU	$8 \times 8 \times 512$
Conv10	$4 \times 4 \times 128$, Conv2DTranspose, BatchNorm, LeakyReLU	$16 \times 16 \times 256$
Conv11	$4 \times 4 \times 64$, Conv2DTranspose, BatchNorm, LeakyReLU	$32 \times 32 \times 128$
Conv12	$4 \times 4 \times 1$, Conv2DTranspose	$64 \times 64 \times 1$

eye and more secure from information loss to the target model. In contrast, a lower δ means a smaller noise to be made and therefore is more unnoticeable but less effective. Finally, the overall noise is the product of $NoiseLevel$ and $NoiseMap$. The modified image from original image x is denoted by $x_{modified}$.

Algorithm 3 Noise Addition.

```
Input: $x$, $G$, $\delta$.
Output: $x_{modified}$
$NoiseMap = G(x) - Mean(G(x))$
$NoiseLevel = \frac{\delta}{|NoiseMap|}$
$x_{modified} = x + NoiseLevel * NoiseMap$
```

IV. Experiments

In this section, we introduce the experimental settings and design of the simulation. We then show the experimental results and discuss the performance. In the subsection, Target Model and Dataset, we show the detailed settings of the target model and the constitution of the dataset for training and testing. In the subsection, GAN Model Structure, we elaborate the shape and connection relationship of each layer of the GAN-based model. In the subsection, Test Results, we present the overall performance of the proposed model on the dataset and two examples for illustration.

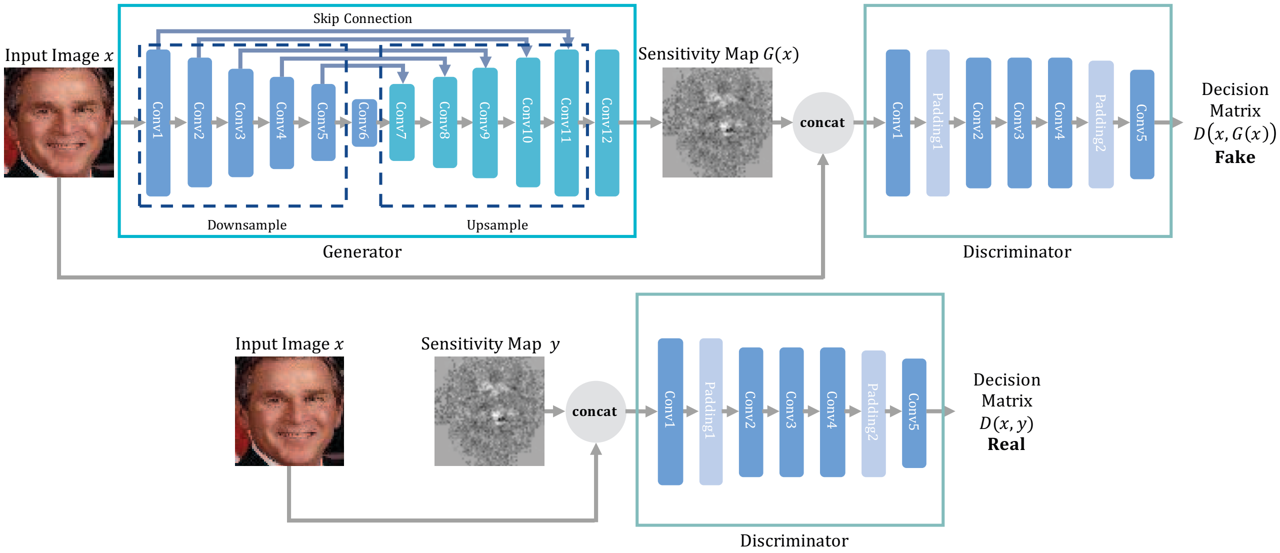

Fig. 4. The structure of the Sensitivity Map Generating GAN. For the generator, the input is the original image x to be protected with the size of $64 \times 64 \times 3$, and the output is the predicted sensitivity map $G(x)$ with the size of $64 \times 64 \times 1$. The structure of the generator consists of 12 convolution layers. The layers $Conv1$ to $Conv5$ and the layers $Conv7$ to $Conv11$ are connected correspondingly by skip connection and shape a U-Net where the layer $Conv6$ is the bottleneck. During the training process, the discriminator takes inputs of the concatenation of the image and the sensitivity map with the size of $64 \times 64 \times 4$ and determines whether the input is from the generator or the real dataset. When the input is the tuple of $[x, G(x)]$, the output of the discriminator $D(x, G(x))$ goes to 1s. Otherwise, if the input is $[x, y]$, the $D(x, y)$ goes to 0s.

TABLE II
THE ARCHITECTURE OF THE DISCRIMINATOR

Layer Name	Layer Operation	Feature Map Shape
Conv1	$4 \times 4 \times 256$, Conv2D	$32 \times 32 \times 256$
Padding1	0 Padding	$34 \times 34 \times 256$
Conv2	$4 \times 4 \times 512$, Conv2D, BatchNorm, LeakyReLU	$31 \times 31 \times 512$
Conv3	$4 \times 4 \times 512$, Conv2D, BatchNorm, LeakyReLU	$31 \times 31 \times 512$
Conv4	$4 \times 4 \times 512$, Conv2D, BatchNorm, LeakyReLU	$31 \times 31 \times 512$
Padding2	0 Padding	$33 \times 33 \times 512$
Conv5	$4 \times 4 \times 1$, Conv2D	$30 \times 30 \times 1$

A. Target Model and Dataset

To test the performance of the noise generation model, we set up the experiment in the following settings. The target face recognition model for the experiment is the FaceNet [30]. FaceNet is one of the most famous face verification, and recognition models and frequently takes the role of the baseline algorithm in the field. The integrated target model includes a Multi-task Cascaded Convolutional Network (MTCNN) [44] to detect the faces in the original input image, and then a convolution neural network to extract and compare the features vector. The target FaceNet model [29] is pretrained on dataset VGGFace2 [15] and reaches the accuracy of 99.65% on the dataset of Labeled Faces in the Wild (LFW). The integrated FaceNet model takes inputs images and outputs the verification results as the distance between the embedding of the images.

With the target model, we construct a new celebrity face photo dataset. The dataset contains faces of 102 celebrities, and each person has 1 openly accessible labeled target official portrait $x_{labeled}$ and 20 other face photos x that need protection. The dataset has a total of 2163 images where 102 of them are labeled images, and the rest of 2060 images are to be used as training and testing data samples. The image data x has the size of 64 by 64 pixels and 3 channels as RGB values. All of the images have the corresponding sensitivity maps calculated by testing the sensitivity of noise on every pixel of the image to the target model as in algorithm 1. The sensitivity maps y have the shape of $64 \times 64 \times 1$, which matches the resolution of the images. The dataset is formed by data tuples, and each tuple is the pair of the image and its sensitivity map as (x, y).

B. GAN Model Settings

The generator of the Sensitivity Map Generating GAN in this particular experiment is designed to take the input of raw image with a size of $64 \times 64 \times 3$ and output the generated sensitivity map. The detailed structure of the generator for this experiment is shown in the Table I. The generator uses the down-sampling 2D convolutional layers, from $Conv1$ to $Conv5$ for the first half of the network to shrink the size of the feature map to the bottleneck layer $Conv6$. Then, the up-sampling transpose 2D convolutional layers, from $Conv7$ to $Conv11$, are in the last half to reform the feature maps back to the size of the resolution of the original image. In different layers, the model extracts the features of the input image of lower or higher levels. Also, the design of skip connection curtails the risk of losing information during the reshaping of the feature maps.

The discriminator of the GAN has a more straightforward structure. The illustration of the architecture is in the Table II. The input is the tuple of the original image and its sensitivity map either generated by the generator or from the ground truth as (x, y). The image and the sensitivity map are concatenated, and the tuple has the shape of $64 \times 64 \times 4$. The output of the discriminator is a 30×30 decision matrix, where each of the elements is a score between 0 and 1, indicating whether the sensitivity map part of the tuple is from the ground truth or produced by the generator.

The integrated model is trained on the dataset with two settings of the sensitivity map of different types of noises added to the raw image. We set the noise type to be the white noise and the black noise, and the noise is applied to the pixel by changing the RGB value to either all 0s or 255s. The noise generator produced sensitivity maps $G(x)$ work as the weights for the noises added to the raw images with a hyper-parameter δ, which controls the noise level. A larger value of δ, which is a higher level of protection, leads to a greater impact on the target model's accuracy of correct verification of the face. However, it also causes a more perceivable alteration from the unadjusted image. The training of the GAN takes 200 epochs on the training set of 95% of the entire dataset.

TABLE III
AVERAGE DISTANCES OF DATA PROTECTED ON DIFFERENT LEVELS

	Black y	Black $G(x)$	White y	White $G(x)$
NoiseLevel 500	1.033	0.831	0.973	0.815
NoiseLevel 750	1.107	0.865	1.042	0.837
NoiseLevel 1000	1.156	0.901	1.084	0.867
NoiseLevel 1250	1.188	0.935	1.120	0.896

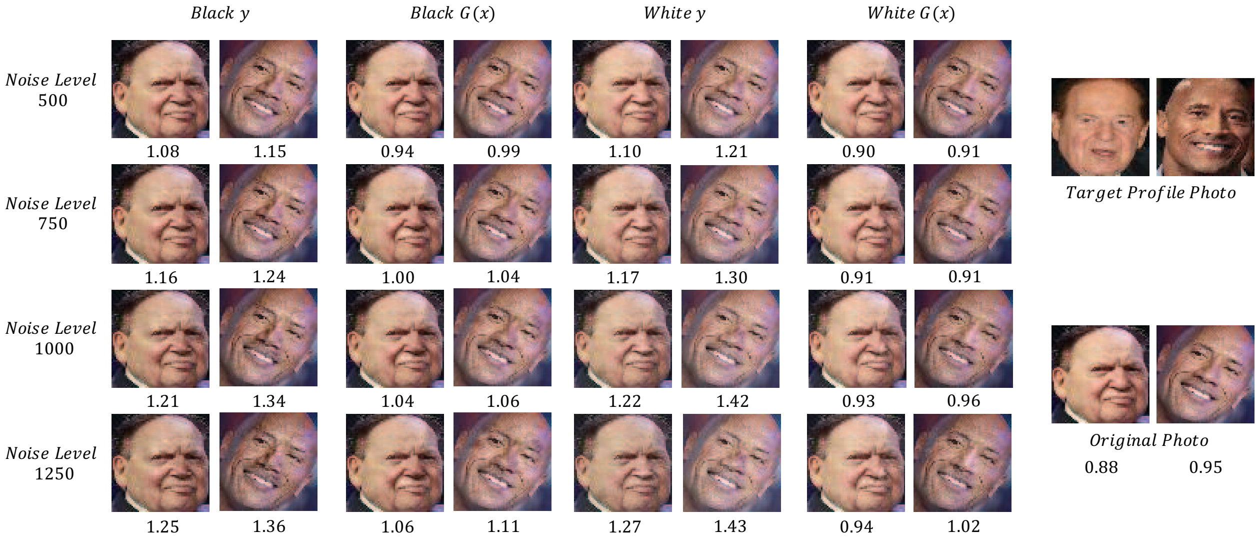

Fig. 5. The case example of the protection results. The protection results are generated by four different kinds of sensitivity maps. From left to right, $Blacky$ is the ground truth sensitivity map of black noise calculated by calling the target model. The $BlackG(x)$ is the sensitivity map of black noise generated by the proposed model. Similarly, the $Whitey$ and $WhiteG(x)$ are those of white noises. The $NoiseLevel$s are set from 500 to 1250. The larger the $NoiseLevel$, the more easily the perturbation is noticeable. Below the images are the distance between the image and the target profile photo shown beside.

C. Test Results

The test results are shown in the following Table III. The original average distance output by target $FaceNet$ model between the images in the test set with their $x_{labeled}$ is 0.788. The numbers in the table are the average distances after the protections by the noises based on the sensitivity maps from different sources. The columns Black y and the White y are the protection results by the sensitivity maps from the ground truth in the dataset, and the Black $G(x)$ and the White $G(x)$ are by the generated sensitivity maps by our model. The $NoiseLevel$ is the indicator defined in algorithm 3. We can see the noises added to the raw photo effectively interfere with the target model's performance on a large scale. With a higher $NoiseLevel$, the distances have a higher increase and achieve more secure protection. The trade-off always exists, that δ can be set higher with a more visible noise sacrificing the information details to achieve more strict protection of privacy. On the $NoiseLevel$ of 1250, the average distance increases to 0.935 with the noises generated based on the sensitivity maps by the GAN. The distance has a more than 18.5% increase while the total amount of noise added to the original image is still acceptable by human observers. The protections using the ground truth sensitivity maps from the dataset have better performance than those using the generated ones. However, the calculation for those ground truth sensitivity maps is not affordable for practical uses, and SMGG has an average sensitivity map generation cost of less than 11.3 milliseconds. The protection by the proposed model achieves a good level of protection based on the fact that the operation can be conducted on any images at a fast speed.

Fig. 5 shows the example of the protection on the photos of Sheldon Adelson and Dwayne Johnson. The target profile photos are the profile images of their Wikipedia pages [1] [2]. The protected image of Sheldon Adelson is from a report of ABC News [3], and Dwayne Johnson's protected image is from the Hollywood Reporter News [4]. The original distances by FaceNet between the two faces are 0.88 and 0.95. The four columns in Fig. 5 show the results of the protections by the noises based on the sensitivity maps from different sources. The numbers in columns Black y and the White y are the output distances after the protection by the sensitivity maps from the ground truth in the dataset in noise type settings of black and white noises. The Black $G(x)$ and the White $G(x)$ are by the generated sensitivity maps by our model. The distances increase in decisive amount after the protection, while the overall alteration of the original data is constrained in an acceptable level.

V. Conclusion

Aiming to solve identity information privacy from face photos to the face recognition models, we propose a fully black-box privacy protection model for face image data, the Sensitivity Map Noise-Adding model. The model adds subtle noise to the original data based on the sensitivity maps, which are produced by a GAN structured generator. The concept of sensitivity of an image works impressively for the task, and the design of the noise generation model functions well for learning the features in the ground truth sensitivity maps to understand the weak points on the images to the target face recognition model. The whole process is free from requesting information from the target face recognition model, and only calls the target model at the stage of the preparation of dataset for training the sensitivity map generation GAN. The noise generation procedure is effective and efficient. The overall performance of the privacy protection model is good against the benchmark face recognition model FaceNet, and the algorithm can run on a real-time level for video data at 60 frames per second.

ACKNOWLEDGE

This work is partially supported by National Science Foundation: NSF 1912753, 1704287, 1829674.

REFERENCES

[1] https://en.wikipedia.org/wiki/Sheldon_Adelson. Accessed April 4, 2020.

[2] https://upload.wikimedia.org/wikipedia/commons/f/f1/Dwayne_Johnson_2%2C_2013.jpg. Accessed April 4, 2020.

[3] https://s.abcnews.com/images/Politics/GTY_sheldon_adelson_mm_160720_hpMain_2_16x9_992.jpg. Accessed April 4, 2020.

[4] https://cdn1.thr.com/sites/default/files/2020/06/dwayne_johnson_-_the_rock_-_dwayne_the_rock_johnson_-_getty_-_h_2020_.jpg. Accessed April 4, 2020.

[5] K. Abas, C. Porto, and K. Obraczka. Wireless smart camera networks for the surveillance of public spaces. *Computer*, 47(5):37–44, 2014.

[6] Apoorv Agha, Rishabh Ranjan, and Woon-Seng Gan. Noisy vehicle surveillance camera: A system to deter noisy vehicle in smart city. *Applied Acoustics*, 117:236–245, 2017. Acoustics in Smart Cities.

[7] Zaheer Allam and David S. Jones. On the coronavirus (covid-19) outbreak and the smart city network: Universal data sharing standards coupled with artificial intelligence (ai) to benefit urban health monitoring and management. *Healthcare*, 8(1), 2020.

[8] Zaheer Allam and Peter Newman. Redefining the smart city: Culture, metabolism and governance. *Smart Cities*, 1(1):4–25, 2018.

[9] A. Alshammari and D. B. Rawat. Intelligent multi-camera video surveillance system for smart city applications. In *2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC)*, pages 0317–0323, 2019.

[10] Zhipeng Cai and Zaobo He. Trading private range counting over big iot data. In *39th IEEE International Conference on Distributed Computing Systems, ICDCS 2019, Dallas, TX, USA, July 7-10, 2019*, pages 144–153. IEEE, 2019.

[11] Zhipeng Cai, Zaobo He, Xin Guan, and Yingshu Li. Collective data-sanitization for preventing sensitive information inference attacks in social networks. *IEEE Transactions on Dependable and Secure Computing*, 15(4):577–590, 2018.

[12] Zhipeng Cai, Zuobin Xiong, Honghui Xu, Peng Wang, Wei Li, and Yi Pan. Generative adversarial networks: A survey toward private and secure applications. 54(6), July 2021.

[13] Lorena Calavia, Carlos Baladrón, Javier M. Aguiar, Belén Carro, and Antonio Sánchez-Esguevillas. A semantic autonomous video surveillance system for dense camera networks in smart cities. *Sensors*, 12(8):10407–10429, 2012.

[14] Andrés Camero and Enrique Alba. Smart city and information technology: A review. *Cities*, 93:84–94, 2019.

[15] Qiong Cao, Li Shen, Weidi Xie, Omkar M Parkhi, and Andrew Zisserman. Vggface2: A dataset for recognising faces across pose and age. In *2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018)*, pages 67–74. IEEE, 2018.

[16] Debayan Deb, Jianbang Zhang, and Anil K. Jain. Advfaces: Adversarial face synthesis, 2019.

[17] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4690–4699, 2019.

[18] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[19] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

[20] Tai hoon Kim, Carlos Ramos, and Sabah Mohammed. Smart city and iot. *Future Generation Computer Systems*, 76:159–162, 2017.

[21] Weiwei Hu and Ying Tan. Generating adversarial malware examples for black-box attacks based on gan, 2017.

[22] Alex Krizhevsky. Learning multiple layers of features from tiny images. pages 32–33, 2009.

[23] Yann LeCun and Corinna Cortes. MNIST handwritten digit database. 2010.

[24] H. Menouar, I. Guvenc, K. Akkaya, A. S. Uluagac, A. Kadri, and A. Tuncer. Uav-enabled intelligent transportation systems for the smart city: Applications and challenges. *IEEE Communications Magazine*, 55(3):22–28, 2017.

[25] Ghulam Muhammad, Mansour Alsulaiman, Syed Umar Amin, Ahmed Ghoneim, and Mohammed F Alhamid. A facial-expression monitoring system for improved healthcare in smart cities. *IEEE Access*, 5:10871–10881, 2017.

[26] Anh Mai Nguyen, Jason Yosinski, and Jeff Clune. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *CVPR*, pages 427–436. IEEE Computer Society, 2015.

[27] Nicolas Papernot, Patrick D. McDaniel, Somesh Jha, Matt Fredrikson, Z. Berkay Celik, and Ananthram Swami. The limitations of deep learning in adversarial settings. In *EuroSP*, pages 372–387. IEEE, 2016.

[28] Omkar M Parkhi, Andrea Vedaldi, and Andrew Zisserman. Deep face recognition. 2015.

[29] David Sandberg. Face recognition using tensorflow. https://github.com/davidsandberg/facenet, 2018.

[30] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823, 2015.

[31] Shawn Shan, Emily Wenger, Jiayun Zhang, Huiying Li, Haitao Zheng, and Ben Y. Zhao. Fawkes: Protecting privacy against unauthorized deep learning models, 2020.

[32] Vijender Kumar Solanki, Somesh Katiyar, Vijay BhashkarSemwal, Poorva Dewan, M. Venkatasen, and Nilanjan Dey. Advanced automated module for smart and secure city. *Procedia Computer Science*, 78:367–374, 2016.

[33] Qing Song, Yingqi Wu, and Lu Yang. Attacks on state-of-the-art face recognition using attentional adversarial attack generative network, 2018.

[34] J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel. Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition. *Neural Networks*, (0):–, 2012.

[35] Jiawei Su, Danilo Vasconcellos Vargas, and Kouichi Sakurai. One pixel attack for fooling deep neural networks. *IEEE Trans. Evolutionary Computation*, 23(5):828–841, 2019.

[36] Yi Sun, Ding Liang, Xiaogang Wang, and Xiaoou Tang. Deepid3: Face recognition with very deep neural networks. *arXiv preprint arXiv:1502.00873*, 2015.

[37] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In Yoshua Bengio and Yann LeCun, editors, *ICLR (Poster)*, 2014.

[38] Jaganathan Venkatesh, Baris Aksanli, Christine S Chan, Alper Sinan Akyurek, and Tajana Simunic Rosing. Modular and personalized smart health application design in a smart city environment. *IEEE Internet of Things Journal*, 5(2):614–623, 2017.

[39] Shuo Wan, Jiaxun Lu, Pingyi Fan, and Khaled B Letaief. To smart city: Public safety network design for emergency. *IEEE access*, 6:1451–1460, 2017.

[40] Chaowei Xiao, Bo Li, Jun-Yan Zhu, Warren He, Mingyan Liu, and Dawn Song. Generating adversarial examples with adversarial networks, 2018.

[41] Hirozumi Yamaguchi, Akihito Hiromori, and Teruo Higashino. A human tracking and sensing platform for enabling smart city applications. In *Proceedings of the Workshop Program of the 19th International Conference on Distributed Computing and Networking*, Workshops ICDCN '18, New York, NY, USA, 2018. Association for Computing Machinery.

[42] Jiachen Yang, Bin Jiang, and Houbing Song. A distributed image-retrieval method in multi-camera system of smart city based on cloud computing. *Future Generation Computer Systems*, 81:244–251, 2018.

[43] Lu Yang, Qing Song, and Yingqi Wu. Attacks on state-of-the-art face recognition using attentional adversarial attack generative network. *Multimedia Tools and Applications*, 80(1):855–875, 2021.

[44] Kaipeng Zhang, Zhanpeng Zhang, Zhifeng Li, and Yu Qiao. Joint face detection and alignment using multitask cascaded convolutional networks. *IEEE Signal Processing Letters*, 23(10):1499–1503, Oct 2016.