IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received August 21, 2021, accepted September 10, 2021, date of publication September 24, 2021, date of current version October 4, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3113186

Development of a Privacy-Preserving UAV System With Deep Learning-Based Face Anonymization

HARIM LEE[1], MYEUNG UN KIM[2], YEONGJUN KIM[3], HYEONSU LYU[4], AND HYUN JONG YANG[4], (Member, IEEE)

[1]School of Electronic Engineering, Kumoh National Institute of Technology, Gumi, Gyungbuk 39177, South Korea
[2]Korea Aerospace Research Institute (KARI), Daejoen 34133, South Korea
[3]Department of Electrical Engineering, Ulsan National Institute of Science and Technology (UNIST), Ulsan 44919, South Korea
[4]Department of Electrical Engineering, Pohang University of Science and Technology (POSTECH), Pohang 37673, South Korea

Corresponding author: Hyun Jong Yang (hyunyang@postech.ac.kr)

This work was supported in part by the Ministry of Science and ICT (MSIT), South Korea, under the Information Technology Research Center (ITRC) Support Program supervised by the Institute of Information and Communications Technology Planning and Evaluation (IITP) under Grant IITP-2021-0-02048, in part by IITP Grant by the Korean Government (MSIP) (Development of Joint Electrical/Mechanical Drone Beamforming based on Target Detection and Precise Attitude Control) under Grant 2018-0-00958, and in part by the Project titled "Development of Technology for Impact Assessment and Management of HNS discharged from Marine Industrial Facilities," by the Ministry of Oceans and Fisheries, South Korea.

ABSTRACT In this paper, we develop a privacy-preserving UAV system that does not infringe on the privacy of people in the videos taken by UAVs. Instead of blurring or masking the face parts of the videos, we want to exquisitely modify only the face parts so that the people in the modified videos still look like humans, but they become anonymous. Doing so, the semantic information of the videos can be preserved even with the anonymization. Specifically, based on the latest generative adversarial network architecture, we propose a deep learning-based face-anonymization scheme so that each modified face part looks like the face of a person who does not actually exist. The trained face-anonymizer is then mounted on the UAV system we have implemented. Through experiments, we confirm that the developed privacy-preserving UAV system anonymizes UAV's first-person videos so that the people in the video are not recognized as anyone in the dataset used. In addition, we show that even with such anonymized videos, the perception performance required for performing UAV's essential functions such as simultaneous localization and mapping is not degraded.

INDEX TERMS Privacy infringement, privacy-preserving vision, deep learning, security robot, UAV patrol system.

I. INTRODUCTION

As one of the most important systems in the 4-th industrial era, unmanned aerial vehicles (UAVs) are expanding their use in all directions, ranging from transportation, delivery, surveillance, security, exploration, military, public safety, agriculture, and smart factories. In particular, UAV systems capable of performing missions autonomously have boundless potential in many applications. The recent rapid advancement of UAV systems is attributed to recent deep learning-based computer vision techniques. As UAV's cognitive ability has soared, it has become possible to autonomously find paths, avoid obstacles, and perform missions stably in various situations. However, as UAVs become ubiquitous around us, UAV's high-performance vision function may raise serious concerns about privacy breaches by exposing us to unwanted recordings. The constant recording of UAVs by itself causes public anxiety and can be easily exploited. In 2015, a Kentucky man shot down a UAV hovering over his property [1], [2]. He argued that the UAV was spying on his 16-year-old daughter who was sunbathing in the garden. An article reported that a Knightscope security robot was suspended from its job of patrolling a San Francisco animal shelter after a few residents complained that the robot was taking unnecessary pictures of them [3].

A few studies have analyzed privacy concerns due to the computer vision function of robots [4], and proposed to resolve the privacy concern by recording videos at extreme

The associate editor coordinating the review of this manuscript and approving it for publication was Pedro R. M. Inácio.

low resolution [5], [6] or by blurring the face parts of the recorded videos for anonymization [7]. However, it becomes very difficult to extract the original facial expression information from the deformed video, if we record the video at extreme low resolution or if we blur or mask the face parts as suggested by the previous studies. Even removing facial expressions is not suitable for patrol robot systems because it can interfere with the perception of someone's situation and cause security problems. In [8], the authors endeavor to convert a whole face part with two aims: i) the converted face should still look like a human face, and ii) the converted face should not look like the original face. However, since the face anonymization method proposed in [8] puts changes on the pixel values of the original face image, the modified face could still have the information of the original face. Therefore, there is a possibility that such technique cannot completely hide private information in the original image.

Recently, DeepFake has attracted great interest not only from academia but also from the general public [9]. The purpose of DeepFake is to make an arbitrary human face that resembles someone. However, the face anonymizer proposed in this study is to completely anonymize someone's identity by converting someone's face into the face of a human who does not actually exist, not in the training dataset.

In this paper, we develop and implement a privacy-preserving UAV system with deep learning-based face anonymization. Our proposed face anonymizer is designed to achieve the following two goals:

- *Anonymization:* Our proposed face anonymizer should anonymize faces without observing any pixel value of the original face images.
- *Preservation of the Original Semantic Information:* The transformed face should still look like a human face while preserving facial expressions as much as possible.

The main features of our proposal are three-fold.

- In the scheme proposed in [8], there is a possibility that a face part could be transformed to resemble someone's face in the training datasets. We design a face anonymizer using two generative adversarial networks, so that the deformed face does not resemble anyone's face in training, and design a training structure and loss function to train our proposed face anonymizer.
- We propose a face-anonymizing approach, where a face image is transformed to an intermediate image by eliminating the privacy-sensitive information, and then the intermediate image is converted to a photorealistic face image. Note that unlike in [8] by using the intermediate image our anonymization algorithm can create an anonymized face without observing the pixel values of the original face. In other words, our anonymization method is suitable for privacy-preserving systems because the anonymizer uses only semantic information that has no inherent features such as face color, wrinkles, eyelids, etc.
- We construct a UAV system equipped with the developed face anonymization feature by using hardware consisting of Pixhawk4, Nvidia Xavier, and others that we have chosen for ourselves, and by using open source-based software such as PX4 and ROS. Experiment results show that the developed UAV system anonymizes the face parts of its recording sufficiently well. As an illustrative example, we present that the developed UAV system performs SLAM well even with anonymized videos.

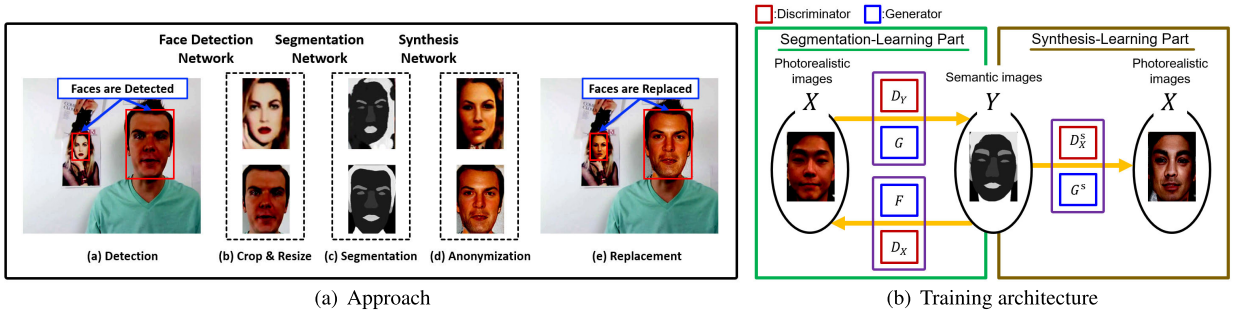
II. RELATED WORK

A. REMOVAL OF PRIVACY SENSITIVE INFORMATION FOR ROBOT SYSTEM

In robot vision areas, privacy infringement has attracted increasing research attention, which has led to the development of methods to eliminate privacy-sensitive information in images [7], [8], [10]–[13].

Jason *et al.* [8] developed a method for privacy-preserving action detection via a face modifier by using generative adversarial networks (GANs). They proposed pixel-level modifications to change each person's face with minimal effect on the action recognition performance. The proposed generator modifies each pixel in an original face to eliminate the features of the face. However, the generator observes the pixels of a face for modification, which implies that the modified face is generated based on the original face. This approach still leaves some information of an original face in the modified face. That is, apart from the ability to anonymize faces, there is a limitation in methodological aspects for privacy protection. The authors in [14] investigated the inversion attacks to a deep neural network, and then revealed that a deep neural network indeed could have the information of the training dataset. They also showed that the training dataset could be extracted from a trained deep neural network. The training data leakage problem through the inversion attack is because a neural network is trained to capture the relationship between the input image and the output image. Hence, if a neural network is trained to generate an anonymized face using the original image, the anonymizing neural network can also have information from the face training dataset and thus can be subjected to the inversion attack. On the other hand, our synthesis network generates an anonymized face using a semantic image, and thus the inversion attack cannot extract original face images from the synthesis network. In addition, as in [15], our training approach that does not expose the original training dataset directly to a target neural network will be encouraged in future studies related to privacy protection.

In [7], the authors proposed a dynamic resolution face detection architecture to blur faces. The framework detects faces from extremely low-resolution images via the proposed deep learning-based algorithm. Except for the detected faces, other privacy-insensitive pixels are enhanced to high resolution. Hence, in resultant images, only faces are blurred, which protects privacy-sensitive parts while preserving the performance of robot perception. However, in the case where the faces are big in a frame, the privacy protection becomes not strong enough for complete anonymization even with



(a) Approach (b) Training architecture

FIGURE 1. The proposed approach for anonymizing faces in a video frame and training architecture for segmentation and synthesis networks.

blurring. More importantly, the initial face detection stage is often unsuccessful, since it should detect faces at extreme-low resolution.

In [10], the authors introduced a scene recognition method from an image. The scheme determines if a person is in a privacy-sensitive location. If an image is taken in a privacy-sensitive place, the scheme enables the camera device to be automatically turned off. However, faces remain exposed in privacy-insensitive places, and thus this scheme is not suitable for privacy-preserving UAV visions.

The works [11]–[13] proposed face-regenerating schemes. In [11], to regenerate a face, the proposed procedure linearly mixed an input image and a network's transformed image with a weight mask. However, this scheme still uses the pixels of an input image. The works [12], [13] utilized landscape features in faces to make regenerated faces. However, the techniques for self-driving of cars and UAVs exploit semantic images to recognize objects around cars and UAVs. Hence, those schemes based on landscape features can be difficult directly to use with autonomous driving technologies.

B. GENERATIVE ADVERSARIAL NETWORKS

GANs have had impressive success in generating realistic images [16]. The goal of this learning framework is to train a neural network to model an image distribution in an unsupervised manner. The trained network can generate a fake image that is indistinguishable from a real image. This training approach has been adopted for image-to-image translation [17]–[25]. Those works learn a mapping from input to output images, meaning that an input image is translated to an image in a different image distribution. To construct our training architecture for obtaining deep-learning networks for our purpose, we have adopted the latest two works [23], [25]. By using the training framework in [23], we create a generator that translates a photorealistic image to a segmentation mask, and [25] is used to train another generator that converts the resultant segmentation mask into a photorelistic image.

C. SIMULTANEOUS LOCALIZATION AND MAPPING (SLAM)

By using SLAM, in an unknown environment, a robot constructs a map around itself and localizes itself in the resultant map. Hence, in a video frame, the manipulation of a few pixels could affect the performance of SLAM, since a map is drawn by extracting feature points of lines, edges, and corners of objects in images. In this work, ORB-SLAM2 [26] is implemented in our system, which is one of the most popular algorithms for vision-based SLAM. Via experiments, it shall be confirmed that our proposal has little effect on the feature point extraction. Hence, our anonymization method has no effect on vision-based robot perception.

III. APPROACH, LOSS FUNCTIONS, AND ALGORITHM

This section introduces our approach for anonymizing faces via neural-type networks. For these networks, we present a training architecture, where two up-to-date GANs are combined. Then, we describe the loss functions for our purpose. Finally, we present and explain our face-anonymizing algorithm.

Approach: Fig. 1(a) illustrates the face-anonymizing approach for our system. To anonymize faces, a companion computer has three networks: a face detection network, segmentation network, and synthesis network. The face detection network operates to detect faces whenever a video frame is fetched to the companion computer. Through the segmentation network, the images of the detected faces are converted to semantic images. Note that the semantic images only have the outline information of faces with no privacy-sensitive information. The synthesis network generates photorealistic images based on the semantic images. Finally, the photorealistic images replace the original faces.

A person's facial expression is determined by the angle at which the eyebrows are bent, the position of the corners of the mouth, and how the person opens his/her mouth and eyes. For each facial expression, the relationship between the states of the facial components are actually determined to some extent. To exploit this relationship for each expression, the proposed approach utilizes a semantic image that preserves the states of eyes, eyebrows, mouth, and lips.

Fig. 1(b) presents our training architecture for segmentation and synthesis networks. In order to train these networks, we combine CycleGAN and GauGAN, each of which is a spotlight image-to-image translation framework using GAN. The CycleGAN is good at translating photorealistic images to semantic images while the GauGAN is proposed

for converting semantic images to photorealistic images. The segmentation network is trained by the CycleGAN, where the segmentation generator called Generator G generates a semantic image from a photorealistic image. The synthesis network is obtained from the GauGAN. Generator G^{s} makes a photorealistic image from the output of the segmentation generator G.

To obtain well-trained networks suitable for anonymization, we modify the losses of generators and discriminators in both GANs.

Training Architecture Model: For our modifications, we formulate our training architecture in the following manner.

For training samples, X and Y denote a photorealistic domain and a semantic domain, respectively. For each domain, training samples are represented by $\{x_i\}_{i=1}^{N}$ and $\{y_j\}_{j=1}^{N}$. Samples of each domain follow a data distribution, which is denoted as $x \sim p_{\mathrm{data}}(x)$ and $y \sim p_{\mathrm{data}}(y)$. In this architecture, we have three generators, G, F, and G^{s}. Each generator is a mapping function: $G : X \to Y$, $F : Y \to X$, and $G^{\mathrm{s}} : Y \to X$. For the adversarial networks of these generators, there are discriminators D_X, D_Y, and D_X^{s}, each of which distinguishes if an input is from the data distribution or is generated by the generator. D_Y, D_X, and D_X^{s} examine the output of G, F, and G^{s}, respectively.

Well-Known Basic Definitions: To describe our modifications, this subsection introduces well-known losses [23], [25].

Adversarial Loss: For each generator and discriminator pair, the *adversarial loss* is defined in the following manner.

$$\mathcal{L}_{\mathrm{adv}}(G, D_Y, X, Y) = \mathbb{E}_{x\sim p_{\mathrm{data}}(x)}[\log(1 - D_Y(G(x)))] + \mathbb{E}_{y\sim p_{\mathrm{data}}(y)}[\log(D_Y(y))], \quad (1)$$

where $G(x)$ is a generated image by the generator G with a image sample x. G tries to make D_Y as difficult as possible to distinguish generated samples $G(x)$ from real samples y, whereas D_Y must not be deceived by G. The relationship can be formulated as $\min_G \max_{D_Y} \mathcal{L}_{\mathrm{adv}}(G, D_Y, X, Y)$. Hence, the generator must actually minimize the following loss.

$$\mathcal{L}_{\mathrm{adv},G}(G, D_Y, X, Y) = \mathbb{E}_{x\sim p_{\mathrm{data}}(x)}[\log(1-D_Y(G(x)))]. \quad (2)$$

where $D_Y(\cdot)$ is between 0 and 1. As $D_Y(G(x))$ is closer to 1, $G(x)$ looks similar to images from the domain Y. For other pairs, (F, D_X) and (G^{s}, D_X^{s}), the adversarial loss can be obtained by replacing (G, D_Y) in (1) with (F, D_X) and (G^{s}, D_X^{s}). In addition, in (1), X and Y are replaced with Y and X.

Cycle-Consistency Loss: This loss [23] is defined as

$$\mathcal{L}_{\mathrm{cyc}}(G, F) = \mathcal{L}_{\mathrm{cyc},G}(F) + \mathcal{L}_{\mathrm{cyc},F}(G), \quad (3)$$

where $\mathcal{L}_{\mathrm{cyc},G}(F)$ and $\mathcal{L}_{\mathrm{cyc},F}(G)$ are cycle-consistency losses for G and F. The losses are defined as

$$\mathcal{L}_{\mathrm{cyc},G}(F) = \mathbb{E}_{x\sim p_{\mathrm{data}}(x)}[\|F(G(x))-x\|_1], \quad (4)$$

$$\mathcal{L}_{\mathrm{cyc},F}(G) = \mathbb{E}_{y\sim p_{\mathrm{data}}(y)}[\|G(F(y))-y\|_1]. \quad (5)$$

This loss is used to induce a sample x_i to be mapped to a desired sample y_j. Note that the adversarial loss guarantees that via a learned mapping function, samples in the domain X are mapped to samples in the domain Y; however, the learned mapping function cannot translate a sample x_i to an intended y_j because the adversarial loss can guarantee translation only between data distributions. Hence, to obtain a mapping function between individual samples, the cycle-consistency loss should be used in the training procedure for the generators.

Multi-Scale Discriminators' Feature Loss: In [25], M multiple discriminators are trained to distinguish y_j and $G(x_i)$ at M different scales, which enables each discriminator to examine y_j and $G(x_i)$ from a different view. As the size of y_j and $G(x_i)$ becomes smaller, a discriminator has a wider view of y_j and $G(x_i)$, as the receptive field sizes of all the discriminators are the same. By using multiple discriminators, for a generator G^{s}, a GAN feature matching loss is defined in the following manner.

$$\mathcal{L}_{\mathrm{FM},G^{\mathrm{s}}}(D_{X,1}^{\mathrm{s}}, \ldots, D_{X,M}^{\mathrm{s}}) = \sum_{k=1}^{M} \frac{1}{M} \mathbb{E}_{(y,x)}\Big[\sum_{i=1}^{T} \frac{1}{N_i} \|D_{X,k}^{\mathrm{s},i}(x) - D_{X,k}^{\mathrm{s},i}(G^{\mathrm{s}}(y))\|_1\Big], \quad (6)$$

where $\mathbb{E}_{(y,x)} \triangleq \mathbb{E}_{(y,x)\sim p_{\mathrm{data}}(y,x)}$ for simplicity. M is the number of discriminators, $D_{X,k}^{\mathrm{s}}(\cdot)$ is the k-th discriminator, and $D_{X,k}^{\mathrm{s},i}$ denotes the i-th layer feature extractor of $D_{X,k}^{\mathrm{s}}(\cdot)$. N_i means the number of elements in each layer, and T is the number of feature layers. Note that G^{s} can learn how to translate a semantic image to a photorealistic image at both coarse and fine views, since discriminators distinguish x and $G^{\mathrm{s}}(y)$ at M different views.

VGG Perceptual Loss: The VGG perceptual loss [25], [27] is obtained by a VGG network, which is defined as

$$\mathcal{L}_{\mathrm{VGG}}(\psi, x, G^{\mathrm{s}}(y)) = \sum_{i\in S_{\mathrm{I}}} \frac{\|\psi_i(G^{\mathrm{s}}(y)) - \psi_i(x)\|_1}{C_i H_i W_i}, \quad (7)$$

where S_{I} is the set including VGG's layer indices, ψ is the VGG network, and ψ_i is denoted as the i-th layer of ψ. For ψ_i, C_i, H_i, and W_i are the number of channels, the height, and the width, respectively. By minimizing $\mathcal{L}_{\mathrm{VGG}}(\cdot)$, G^{s} can generate a photorealistic image $G^{\mathrm{s}}(y)$, thereby visually indistinguishable from x in the feature-level perspective.

A. MODIFICATIONS OF LOSS FUNCTIONS FOR ANONYMIZATION

1) MODIFICATION ON SEGMENTATION GENERATOR'S LOSS

The goal of a segmentation generator G is to make semantic images with which a synthesis generator G^{s} creates well-synthesized images.

a: MODIFICATION

In order to consider the performance of G^{s} in the loss of G, we define the loss of G as follows:

$$\mathcal{L}_G(G, F, D_Y, G^{\mathrm{s}}) = \underbrace{\mathcal{L}_{\mathrm{adv},\,\mathrm{G}}(G, D_Y, X, Y) + \lambda_{\mathrm{cyc}}\mathcal{L}_{\mathrm{cyc},G}(F)}_{\text{original loss of } G} + \underbrace{\lambda_{\mathrm{s}}\mathcal{L}_{G^{\mathrm{s}}}(G^{\mathrm{s}}, D_{X,1}^{\mathrm{s}}, \ldots, D_{X,M}^{\mathrm{s}}, G) + \lambda_{\mathrm{dist}}\mathcal{L}_{\mathrm{dist}}(G)}_{\text{newly added term}}, \quad (8)$$

where $\mathcal{L}_{\text{dist}}(G) = \mathbb{E}_{x\sim p_{\text{data}}(x)}[\|G(x) - y\|_1]$, and $\mathcal{L}_{\text{dist}}(G)$ could further reduce the size of the space of possible mapping functions with $\mathcal{L}_{\text{cyc},G}(F)$. $\mathcal{L}_{G^{\text{s}}}$ is the loss of a synthesis generator G^{s} and will be explained in Section III-A2 in detail. In addition, λ_{cyc}, λ_{s}, and λ_{dist} control the relative importance of each loss.

By adding $\mathcal{L}_{G^{\text{s}}}(G^{\text{s}}, D^{\text{s}}_{X,1}, \ldots, D^{\text{s}}_{X,M}, G)$, the generator G is trained to generate a semantic image minimizing the loss of G^{s}.

b: OBJECTIVE OF SEGMENTATION LEARNING PART

The complete loss of the segmentation-learning part is defined as

$$\begin{aligned}&\mathcal{L}_{\text{seg}}(G, F, D_X, D_Y, G^{\text{s}})\\ &\quad= \mathcal{L}_G(G, F, D_Y, G^{\text{s}}) + \mathcal{L}_F(G, F, D_X)\\ &\qquad+ \mathbb{E}_{y\sim p_{\text{data}}(y)}[\log(D_Y(y))] + \mathbb{E}_{x\sim p_{\text{data}}(x)}[\log(D_X(x))],\end{aligned} \tag{9}$$

where $\mathcal{L}_F(G, F, D_X) = \mathcal{L}_{\text{adv},F}(F, D_X, Y, X) + \lambda_{\text{cyc}}\mathcal{L}_{\text{cyc},F}(G)$.

The segmentation learning part finds G and F as follows:

$$G^*, F^* = \underset{G,F}{\text{argmin}} \max_{D_X, D_Y} \mathcal{L}_{\text{seg}}(G, F, D_X, D_Y, G^{\text{s}}). \tag{10}$$

2) MODIFICATIONS OF SYNTHESIS GENERATOR'S LOSS

There are two main challenges that hinder the learning of a face anonymizing synthesis generator. We introduce the loss of the synthesis generator G^{s} in [25], and explain each challenge. Then, to obtain a synthesis generator that works for the objective of our system, we modify the loss of G^{s} and the loss of $D^{\text{s}}_{X,k}$, $\forall k$. In addition, we modify the means to train the discriminator $D^{\text{s}}_{X,k}$.

In [25], by using (2), (6), and (7), the loss of the synthesis generator is defined as

$$\begin{aligned}&\mathcal{L}_{G^{\text{s}}}(G^{\text{s}}, D^{\text{s}}_{X,1}, \ldots D^{\text{s}}_{X,M}, G)\\ &\quad= \mathcal{L}_{\text{VGG}}(\psi, x, G^{\text{s}}(G(x))) + \sum_{k=1}^{M}\frac{1}{M}\\ &\qquad\times\left\{\mathcal{L}_{\text{adv},G^{\text{s}}}(G^{\text{s}}, D^{\text{s}}_{X,k}, Y, X, G(x)) + \mathcal{L}_{\text{FM},G^{\text{s}}}(D^{\text{s}}_{X,k})\right\},\end{aligned} \tag{11}$$

where we introduce, for simplicity, $\mathcal{L}_{\text{FM},G^{\text{s}}}(D^{\text{s}}_{X,k}) = \mathbb{E}_{(y,x)}\left[\sum_{i=1}^{T}\frac{1}{N_i}\|D^{\text{s},i}_{X,k}(x) - D^{\text{s},i}_{X,k}(G^{\text{s}}(y))\|_1\right]$ in (6). Additionally, $\mathcal{L}_{\text{adv},G^{\text{s}}}(G^{\text{s}}, D^{\text{s}}_{X,k}, Y, X, G(x))$ is defined as

$$\begin{aligned}&\mathcal{L}_{\text{adv},G^{\text{s}}}(G^{\text{s}}, D^{\text{s}}_{X,k}, Y, X, G(x))\\ &\qquad= \mathbb{E}_{\hat{y}\sim p_{\text{data}}(y)}\left[\log(1 - D^{\text{s}}_{X,k}(G^{\text{s}}(G(x))))\right].\end{aligned} \tag{12}$$

where $\hat{y} = G(x)$.

a: CHALLENGE IN VGG PERCEPTUAL LOSS

In the synthesis-learning part, the loss (11) should be minimized in order to train the generator G^{s}. The minimization leads to the reduction of $\mathcal{L}_{\text{VGG}}(\psi, x, G^{\text{s}}(G(x)))$, and thus the distance between features of x and $G^{\text{s}}(G(x))$ is also reduced during the training of G^{s}. Consequently, the generator G^{s} is trained to generate a photorealistic image $G^{\text{s}}(G(x))$ that is almost the same as the original image x. This trained generator should not be used for our privacy-preserving UAV system.

b: MODIFICATION OF VGG PERCEPTUAL LOSS

To prevent the distance between $G^{\text{s}}(G(x))$ and x from being reduced to a very small value, we introduce margins to (7) as

$$\begin{aligned}&\mathcal{L}_{\text{VGG}}(\psi, x, G^{\text{s}}(\hat{y}), \Upsilon)\\ &\quad= \sum_{i\in S_{\text{I}}}\max\left(0, \frac{\|\psi_i(G^{\text{s}}(\hat{y})) - \psi_i(x)\|_1}{C_iH_iW_i} - \Upsilon(m(i))\right),\end{aligned} \tag{13}$$

where the set $\Upsilon = \{\epsilon_1, \ldots, \epsilon_{|S_{\text{I}}|}\}$ includes margins corresponding to each VGG's layer ψ_i. $|S_{\text{I}}|$ is the number of elements in S_{I}. $m(i)$ is a mapping function to find, for Υ, an index corresponding to the VGG's layer index i in S_{I}. Then, ϵ_i allows the distance between the i-th VGG layer for x and $G^{\text{s}}(G(x))$ to be at least ϵ_i. Hence, a photorealistic image $G^{\text{s}}(G(x))$ can have different features from features of the original image x, which could make $G^{\text{s}}(G(x))$ appear different from x.

c: CHALLENGE IN ADVERSARIAL LOSS AND MULTI-SCALE DISCRIMINATORS' FEATURE LOSS

First, we need to comprehend how the discriminator $D^{\text{s}}_{X,k}$ works. Based on the understanding, we describe a hindrance to the learning of our synthesis generator G^{s}. Then, we modify the adversarial losses of G^{s} and $D^{\text{s}}_{X,k}$, and the multi-scale discriminators' feature loss of G^{s}.

To minimize (12), G^{s} makes a synthesized face $G^{\text{s}}(G(x))$ look like a face in the training dataset, and thus tends to translate $G(x)$ to x, which is not allowed for anonymization. Specifically, in (12), a discriminator $D^{\text{s}}_{X,k}$ examines $G^{\text{s}}(G(x))$ to determine if $G^{\text{s}}(G(x))$ is from the training dataset X or is arbitrarily generated. The generated image $G^{\text{s}}(G(x))$ contains an entire face, and thus $D^{\text{s}}_{X,k}$ is trained to determine whether the entire face in $G^{\text{s}}(G(x))$ is from the training dataset.

Consequently, to deceive $D^{\text{s}}_{X,k}$, G^{s} is trained to generate x from $G(x)$, which greatly reduces the size of the space of possible mapping from the domain Y to the domain X.

d: MODIFICATIONS TO ADVERSARIAL LOSS AND MULTI-SCALE DISCRIMINATORS' FEATURE LOSS

In order to prevent G^{s} from regenerating the almost same face as x, we modify the adversarial losses of G^{s} and $D^{\text{s}}_{X,k}$, which expands the space of possible mapping. To do it, we limit a discriminator $D^{\text{s}}_{X,k}$ to investigate each facial component and not the entire face. By applying the idea, the adversarial loss is rewritten in the following manner.

$$\begin{aligned}&\mathcal{L}^{s}_{\text{adv}}(G^{\text{s}}, D^{\text{s}}_{X,1}, \ldots, D^{\text{s}}_{X,M}, Y, X, S_\xi)\\ &\quad= \sum_{k=1}^{M}\frac{1}{M}\left\{\mathbb{E}_{\hat{y}\sim p_{\text{data}}(y)}\left[\sum_{i\in S_\xi}\frac{1}{|S_\xi|}\log(1 - D^{\text{s}}_{X,k}(\xi_i(G^{\text{s}}(\hat{y}))))\right]\right\}\\ &\qquad+ \sum_{k=1}^{M}\frac{1}{M}\left\{\mathbb{E}_{x\sim p_{\text{data}}(x)}\left[\sum_{i\in S_\xi}\frac{1}{|S_\xi|}\log(D^{\text{s}}_{X,k}(\xi_i(x)))\right]\right\},\end{aligned} \tag{14}$$

where we denote the output of our segmentation generator as $\hat{y} = G(x)$, ξ_i is to extract pixels corresponding to the label index i, S_ξ is the set including extracted labels' index, and $|S_\xi|$ is the number of elements in S_ξ. For example, if $i = 2$ and the label index 2 indicates the nose in a face, all pixels in $\xi_2(G^{\mathrm{s}}(\hat{y}))$ become zero, except for the pixels corresponding to the nose.

According to (14), our modification enables a discriminator $D^{\mathrm{s}}_{X,k}$ to examine a part of a face instead of observing all facial parts together. This approach enables discriminators to learn the distribution of each facial component instead of learning the distribution of an entire face. That is, a discriminator attempts to distinguish each part of a face in $G^{\mathrm{s}}(\hat{y})$ from the same part of a face in x, which could widen the space of possible mapping from the perspective of the entire face. By setting $|S_\xi| < N_{\mathrm{f}}$, where N_{f} denotes the number of labels in a face, we make discriminators observe certain parts of an entire face, and thus the generator G^{s} could have wider space for possible mapping of the other parts that are not examined by discriminators.

In the same vein, the feature loss of the multi-scale discriminators can be also redefined as (15).

Despite our modifications to $\mathcal{L}^{\mathrm{s}}_{\mathrm{adv}}(\cdot)$ and $\mathcal{L}_{\mathrm{FM},G^{\mathrm{s}}}(\cdot)$, there is still room for G^{s} to regenerate x because $\mathcal{L}_{\mathrm{FM},G^{\mathrm{s}}}(D^{\mathrm{s}}_{X,k}, S_\xi)$ in (15) still compares features of $G^{\mathrm{s}}(G(x))$ and x. Hence, we modify (15) to (16), as shown at the bottom of the page. In (16), $\mathbb{E}_{\hat{y},x,\tilde{x}} \triangleq \mathbb{E}_{(\hat{y},x,\tilde{x})\sim p_{\mathrm{data}}(x,y,x)}$ and $\tilde{x} \neq x$, but $\tilde{x}$ is from the same training dataset of x. By the modification, $\mathcal{L}_{\mathrm{FM},G^{\mathrm{s}}}(D^{\mathrm{s}}_{X,k}, S_\xi)$ compares features of $G^{\mathrm{s}}(G(x))$ to features of $\tilde{x}$, which can help G^{s} learn to generate a different face from x.

e: OBJECTIVE OF THE SYNTHESIS LEARNING PART

Based on (13), (14), and (16), our complete objective of the synthesis-learning part is defined as

$$\begin{aligned}\mathcal{L}_{\mathrm{syn}}(G^{\mathrm{s}}, D^{\mathrm{s}}_{X,1}, \ldots, D^{\mathrm{s}}_{X,M}, S_\xi, G) &= \mathcal{L}_{\mathrm{VGG}}(\psi, x, G^{\mathrm{s}}(\hat{y}), S_{\mathrm{I}}) \\ &+ \mathcal{L}^{\mathrm{s}}_{\mathrm{adv}}(G^{\mathrm{s}}, D^{\mathrm{s}}_{X,1}, \ldots, D^{\mathrm{s}}_{X,M}, Y, X, S_\xi) + \mathcal{L}_{\mathrm{cyc},G^{\mathrm{s}}}(G) \\ &+ \mathcal{L}_{\mathrm{FM},G^{\mathrm{s}}}(D^{\mathrm{s}}_{X,1}, \ldots, D^{\mathrm{s}}_{X,M}, S_\xi), \end{aligned} \tag{17}$$

where $\mathcal{L}_{\mathrm{cyc},G^{\mathrm{s}}}(G) = \mathbb{E}_{x\sim p_{\mathrm{data}}(x)}[\|G(G^{\mathrm{s}}(G(x))) - G(x)\|_1]$ enables a synthesized image $G^{\mathrm{s}}(G(x))$ to maintain the shape and location of each facial part in x. Since (16) compares features of $G^{\mathrm{s}}(G(x))$ to those of $\tilde{x}$, the generator G^{s} can cause a synthesized image to retain shape and location of each facial part in $\tilde{x}$ not in x. Hence, $\mathcal{L}_{\mathrm{cyc},G^{\mathrm{s}}}(G)$ helps G^{s} generate synthesized images to retain the shape and location of facial parts in x. By $\mathcal{L}_{\mathrm{cyc},G^{\mathrm{s}}}(G)$ and (16), G^{s} can generate a synthesized face $G^{\mathrm{s}}(G(x))$ including the facial features of $\tilde{x}$ while maintaining the shape and location of facial components in x.

Finally, the synthesis learning part solves (17) as

$$(G^{\mathrm{s}})^* = \underset{G^{\mathrm{s}}}{\operatorname{argmin}} \max_{D^{\mathrm{s}}_{X,k},\forall k} \mathcal{L}_{\mathrm{syn}}(G^{\mathrm{s}}, D^{\mathrm{s}}_{X,1}, \ldots, D^{\mathrm{s}}_{X,M}, S_\xi, G). \tag{18}$$

Algorithm 1 Training Procedure for One Epoch

Output: Generators, G^* and $(G^{\mathrm{s}})^*$
for i = 1: N_{data}
1) Select x, y, $\tilde{x}$ from dataset X and Y
2) Update G, F, D_X, D_Y with G^{s}
(9): $\underset{G,F}{\operatorname{argmin}} \max_{D_X,D_Y} \mathcal{L}_{\mathrm{seg}}(G, F, D_X, D_Y, G^{\mathrm{s}})$
3) Update G^{s}, $D^{\mathrm{s}}_{X,k}$, $\forall k$ with G, F, D_X, D_Y
(17): $\underset{G^{\mathrm{s}}}{\operatorname{argmin}} \max_{D^{\mathrm{s}}_{X,k},\forall k} \mathcal{L}_{\mathrm{syn}}(G^{\mathrm{s}}, D^{\mathrm{s}}_{X,1}, \ldots, D^{\mathrm{s}}_{X,M}, S_\xi, G)$

Algorithm 2 Face-Anonymizing Algorithm

Input: Face detector D; Segmentation generator G; Synthesis generator G^{s}; Video frame v
Output: Anonymized video frame $\tilde{v}$
1) $D(v) \rightarrow f_{\mathrm{D}}, N_{\mathrm{face}}$ // Face detection in v
2) **if** $N_{\mathrm{face}} > 0$ **then** // Faces exist
for $k \leftarrow 1$ to N_{face} **do**
- $G^{\mathrm{s}}(G(f_{\mathrm{D}}(k))) \rightarrow f_{\mathrm{A}}$ // Anonymization
- $f_{\mathrm{A}} \circ B^{-1}(G(f_{\mathrm{D}})(k))$
$+ f_{\mathrm{D}}(k) \circ B(G(f_{\mathrm{D}}(k))) \rightarrow f_{\mathrm{A}} \cdots$ ①
- $(v - f_{\mathrm{D}}(k)) + f_{\mathrm{A}} \rightarrow \tilde{v}$ // Face's replacement
end for
else // No faces to anonymize
$v \rightarrow \tilde{v}$
3) Terminate face anonymization in a video frame

B. TRAINING PROCEDURE

The overall training procedure is summarized in Algorithm 1, where N_{data} is the number of data in the dataset X and Y.

$$\mathcal{L}_{\mathrm{FM},G^{\mathrm{s}}}(D^{\mathrm{s}}_{X,1}, \ldots, D^{\mathrm{s}}_{X,M}, S_\xi) = \sum_{k=1}^{M} \frac{1}{M} \underbrace{\mathbb{E}_{(\hat{y},x)}\Big[\sum_{i=1}^{T} \frac{1}{N_i} \sum_{j\in S_\xi} \frac{1}{|S_\xi|} \|D^{\mathrm{s},i}_{X,k}(\xi_j(x)) - D^{\mathrm{s},i}_{X,k}(\xi_j(G^{\mathrm{s}}(\hat{y})))\|_1\Big]}_{\mathcal{L}_{\mathrm{FM},G^{\mathrm{s}}}(D^{\mathrm{s}}_{X,k}, S_\xi)}, \tag{15}$$

$$\mathcal{L}_{\mathrm{FM},G^{\mathrm{s}}}(D^{\mathrm{s}}_{X,1}, \ldots, D^{\mathrm{s}}_{X,M}, S_\xi) = \sum_{k=1}^{M} \frac{1}{M} \underbrace{\mathbb{E}_{(\hat{y},x,\tilde{x})}\Big[\sum_{i=1}^{T} \frac{1}{N_i} \sum_{j\in S_\xi} \frac{1}{|S_\xi|} \|D^{\mathrm{s},i}_{X,k}(\xi_j(\tilde{x})) - D^{\mathrm{s},i}_{X,k}(\xi_j(G^{\mathrm{s}}(\hat{y})))\|_1\Big]}_{\mathcal{L}_{\mathrm{FM},G^{\mathrm{s}}}(D^{\mathrm{s}}_{X,k}, S_\xi)}. \tag{16}$$

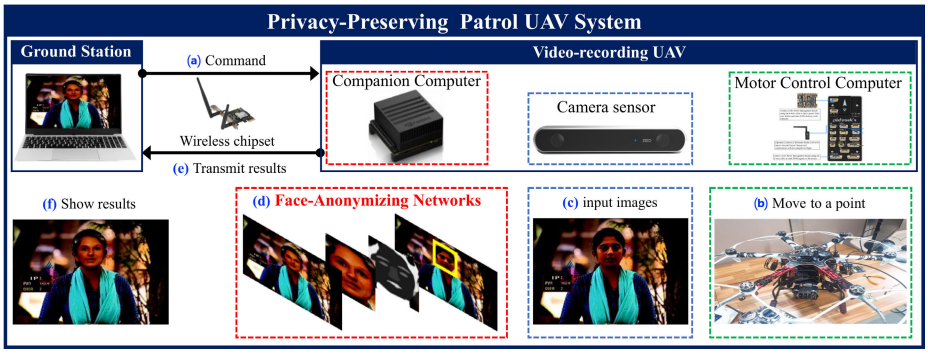


FIGURE 2. Composition of the developed privacy-preserving UAV system with face anonymization.

By repeating the training procedure, we obtain the optimized G^* and $(G^S)^*$.

C. FACE-ANONYMIZING ALGORITHM

With the optimized segmentation and synthesis generators, our face-anonymizing procedure is conducted as in Algorithm 2. A companion computer conducts Algorithm 2 whenever it receives a video frame. Here, f_D denotes box images of detected faces, N_{face} is the number of detected faces, and f_A is a box image of an anonymized face. Then, f_D has N_{face} box images.

In Algorithm 2, $B(\cdot)$ is a function that converts all nonzero elements of an input semantic image to 0 and all zero elements to 1. In a semantic image, zero indicates the background, and thus $B(\cdot)$ is used to extract a background image. Then, $B^{-1}(\cdot)$ extracts only a face image from a semantic image. The operator $\circ$ is the Hadamard product.

Hence, in Algorithm 2, ① creates an image that mixes the anonymized face in f_A and the background in f_D, which could preserve the background in the original image. In ①, the first term generates an image that includes only the anonymized face in f_A and the second term produces an image that includes only the background in f_D.

D. UAV SYSTEM COMPOSITION

In Fig. 2, our face-anonymizing UAV patrol system comprises a ground station and a video-recording UAV.

1) GROUND STATION

This component has two roles: (1) commander and (2) viewer. The ground station is connected to the video-recording UAV via Wi-Fi. This part runs a command program, and then controls the location of the UAV. In addition, this ground station receives anonymized video frames from the UAV, and then shows these frames via our viewer.

2) VIDEO-RECORDING UAV

This consists of a high-resolution camera, a companion computer, and a motor control computer; the camera and motor control computer are connected to the companion computer. The companion computer fetches video frames from the camera, and then anonymizes faces in the received frames by executing our face-anonymizing networks. Then, the ORB-SLAM2 is processed in the anonymized videos.

As in Fig. 2, the companion computer is connected with the UAV control computer, the wireless chipset, and the high-resolution camera. We utilize ROS to enable all the components to communicate with each other. The companion computer communicates with the ground station through wireless communication. The ground station can transmit a command message to the companion computer. The companion computer sends the received message to the UAV control computer. The UAV continuously records images via the camera, which are passed to our face-anonymzing networks implemented in the companion computer. The anonymized images are sent back to the ground station via the wireless chipset, and thus we can immediately check the results on the screen of the ground station. Simultaneously, the anonymized images are processed by the ORB-SLAM2 algorithm in the companion computer.

TABLE 1. Parameters used in each learning part.

Segmentation		Synthesis	
Parameter	Value	Parameter	Value
Image size	256×256	G^S	SPADE
G and F	9 Resnet blocks	M	3
D_X and D_Y	70×70 PatchGAN	λ_{cyc}, λ_s, λ_{dist}	10, 10, 1

IV. EVALUATION

A. TRAINING DETAILS

In the segmentation learning part, for the segmentation generator G, we adopt the network architecture in [27] that is known for powerful neural-type transfer. For the discriminators D_X and D_Y, we use 70 × 70 PatchGANs [18], [23], [28], [29]. For our synthesis generator, we use the SPADE generator in [25]. Table 1 summarizes the parameters used in each learning part.

We conduct our training procedure with CelebA-HQ dataset [30]. This dataset contains 30,000 high-resolution face images with 19 semantic classes. In this work, we modify the semantic dataset by extracting 9 main facial components. In our modified semantic dataset, the semantic classes include skin, nose, eyes, eyebrows, ears, mouth, lip, hair, and neck.

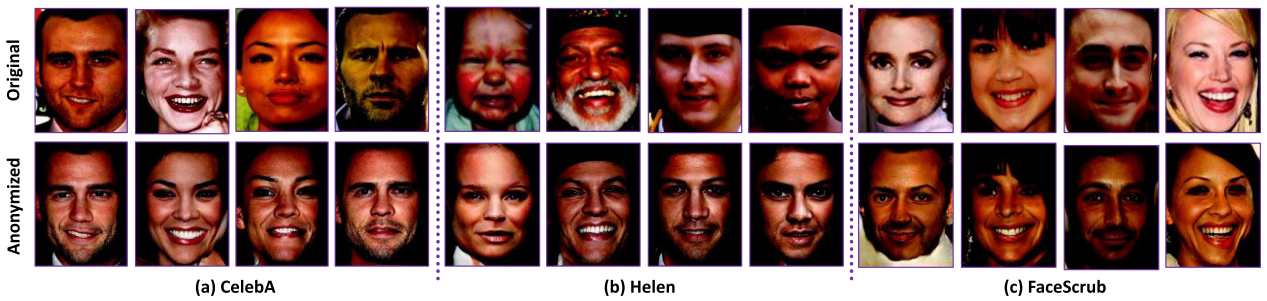


FIGURE 3. Test results of our face anonymization networks with the CelebAMask-HQ dataset, the Helen, and the FaceScrub.

TABLE 2. De-identification performance.

Recall@1	Test dataset		
	CelebA Mask-HQ	Helen	FaceScrub
Ours	3.5	2.3	2.0
CIAGAN	2.0	1.5	2.0

B. EVALUATION OF FACE-ANONYMIZING GENERATORS

This section presents the evaluation of our face-anonymizing generators in both qualitative and quantitative perspectives. In addition, we also show our proposed scheme works well in the situation of a patrol drone. Our system utilizes a lightweight but accurate face detector called FaceBoxes [31]. The computing speed is invariant irrespective of the number of faces in an image. We test our face-anonymizing generators on several datasets: (1) CelebAMask-HQ, (2) Helen, and (3) FaceScrub [32], [33]. For CelebAMask-HQ, the test data is different from the training data. Note that it takes about 120 ms to process one 256×256 face image with NVIDIA GeForce GTX 1080 Ti.

Fig. 3 provides the quality of our face-anonymizing generators, which confirms that our method produces well anonymized faces. For each dataset, additional results are provided in [34]. For quantitative evaluation of our face-anonymizing generators, we utilize the Siamese network architecture [35] that is widely used for re-identification [12]. Through this network, we measure how much an original face is different from its anonymized version. By using the CASIA-WebFace dataset [36], a Siamese network based on the inception Resnet [37] is trained to evaluate the de-identification performance. By using the trained network, we obtain the standard recall-at-1 (Recall@1) metric for re-identification [12]. This metric is the ratio of samples whose nearest neighbor is from the same class, and takes values between 0 and 100. The closer the metric value is to 0, the more complete de-identification. In addition, we show the facial expression comparison, which verifies how well our proposed scheme and a state-of-the-art (SOTA) preserve the facial expressions of original images.

For each dataset, Table 2 shows the Recall@1 value of our proposed scheme and the SOTA CIAGAN method. With a recall of $2 - 3.5$ %, our proposed anonymization algorithm severely degrades the identification performance. In addition, with a performance difference of only 1.5 %, the proposed

FIGURE 4. Facial expression comparison of our proposed scheme and the SOTA CIAGAN.

FIGURE 5. Feature points extracted by ORB-SLAM2 from an original video frame and our anonymized version obtained by the proposed scheme.

method is competitive with the SOTA CIAGAN method. Fig. 4 shows the facial expression comparison of our proposed scheme and the SOTA CIAGAN. The proposed scheme using semantic images can better preserve the facial expressions of original images than CIAGAN using landscape features. This is because semantic images contain more detailed shapes of facial components than landscape features, and thus they can contain more information about facial expressions than landscape features.

As a result, our anonymization algorithm can indeed make an anonymized face appear different from the original face, and thus the generated images are almost unrecognizable by the famous Siamese-based identification system. In addition, by using semantic images, the proposed scheme can better retain facial expressions with limited de-identification performance reduction. In the case of a patrol robot scenario, it is also important to accurately identify facial expressions in order to fully understand a person's situation. Hence, we can confirm that the proposed scheme is a promising solution

FIGURE 6. Video frames with varying sizes of a face, taken by the developed UAV system.

FIGURE 7. Feature points extracted by ORB-SLAM2 from the same videos in Fig. 6.

TABLE 3. Products for the customized UAV system.

Component	Product name
Frame	DJI 550
Propeller	1137 T-motor V2 carbon fiber
Motor	T-motor MN3110 KV 780
ESC	T-motor Air 40A
Control computr	Holybro Pixhawk4
Companion computer	NVIDIA Jetson Xavier
Batter	8800mah 4S1P LiPo

for the patrol robot scenario that needs to accurately grasp a human's situation while protecting individual privacy.

C. EVALUATION OF FACE-ANONYMIZING ALGORITHM IN REAL-WORLD ENVIRONMENTS

In this subsection, we investigate the performance of our proposed face-anonymizing algorithm under static and dynamic environments. The dynamic scenario is the UAV scenario where the UAV records a person, which is our target scenario while a static scenario is that a person is sitting in front of the desk. In addition, we also discuss the impact of our proposal on vision-based robot perception, ORB-SLAM2. Through both real-world scenarios, we show that our technique can work well in a variety of environments.

1) STATIC ENVIRONMENT: DESK SCENARIO

Fig. 5 presents frames with SLAM's features that are obtained via ORB-SLAM2. In the resultant images, green boxes are feature points extracted by ORB-SLAM2. The feature points indicate corners of objects in an image.

Fig. 5 shows the detected feature points of an original video frame and our anonymized version obtained by the proposed scheme, respectively. The extracted feature points of our anonymized frame are almost the same as those of the original frame. In addition, it is evident from the figure that the face is also anonymized well by the proposed scheme. From the experiments, it is confirmed that our anonymization scheme is designed not to degrade UAVs' perception performance while preserving privacy of the people in the video.

2) DYNAMIC ENVIRONMENT: UAV SCENARIO

We have built a customized UAV system with a ZED stereo camera. The products selected for the UAV system are summarized in Table 3.

Fig. 6 shows snapshots taken by our UAV system. The snapshots in the first and second rows are original images and their corresponding anonymized versions, respectively. The UAV is hovering in our laboratory, and a person is walking in front of the UAV. From this experiment, we can confirm that our system well anonymizes a face with varying size. Fig. 7 presents the example of SLAM with an original and anonymized videos. As shown by the results, feature points are well extracted for various sizes of the face also with the anonymized videos obtained by the proposed scheme. Finally, a video example of ORB-SLAM2 by our developed

privacy-preserving UAV system can be found in the following link: https://youtu.be/8S5jikJQltc.

V. CONCLUSION

To protect the privacy of people in the recorded video, we have proposed a privacy-preserving UAV system with deep-learning-based face anonymization. The proposed face-anonymizing neural networks generate human faces based on semantic images without observing the pixels of original faces. The generated faces are totally different from the original faces, but preserve the original semantic information and facial expressions. Hence, in the proposed privacy-preserving UAV system, the trained neural networks transform all original faces in every snapshot to different faces which are not in any dataset used. Privacy of the people in every snapshot can be fundamentally protected. Moreover, the proposed UAV system can preserve the vision-based UAV's perception performance even with anonymized videos. Since the proposed face anonymization utilizes semantic images of faces, we expect that the proposed privacy-preserving UAV system can be simply applied to the autonomous driving techniques using semantic images for recognizing objects around cars and UAVs.

REFERENCES

[1] D. Whiter. (2015). *Kentucky Man Arrested for Shooting Down a Drone Over His Property*. [Online]. Available: https://time.com/3977166/drone-shooting-down-kentucky/

[2] S. Meyer. (2018). *Eye in the Sky—Drone Surveillance and Privacy*. [Online]. Available: https://www.cpomagazine.com/data-privacy/eye-in-the-sky-drone-surveillance-and-privacy/

[3] J. Littman. (2018). *7 Sightings That Prove the Robot Invasion is Already Here*. [Online]. Available: https://www.bisnow.com/national/news/technology/7-incidents-that-prove-% the-robot-invasion-is-already-here-85607/?utm_source=CopyShare&utm_medium=B% rowser

[4] E. Zeng, S. Mare, and F. Roesner, "End user security and privacy concerns with smart Homes," in *Proc. 13th Symp. Usable Privacy Secur.*, 2017, pp. 65–80.

[5] M. S. Ryoo, K. Kim, and H. J. Yang, "Extreme low resolution activity recognition with multi-siamese embedding learning," in *Proc. 32nd AAAI Conf. Artif. Intell. (AAAI)*, New Orleans, LA, USA, Feb. 2018, pp. 7315–7322.

[6] M. S. Ryoo, B. Rothrock, C. Fleming, and H. J. Yang, "Privacy-preserving human activity recognition from extreme low resolution," in *Proc. 31st AAAI Conf. Artif. Intell.*, San Francisco, CA, USA, Feb. 2017, pp. 4255–4262.

[7] M. U. Kim, H. Lee, H. J. Yang, and M. S. Ryoo, "Privacy-preserving robot vision with anonymized faces by extreme low resolution," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Nov. 2019, pp. 462–467.

[8] Z. Ren, Y. J. Lee, and M. S. Ryoo, "Learning to anonymize faces for privacy preserving action detection," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2018, pp. 620–636.

[9] F. Juefei-Xu, R. Wang, Y. Huang, Q. Guo, L. Ma, and Y. Liu, "Countering malicious DeepFakes: Survey, battleground, and horizon," 2021, *arXiv:2103.00218*. [Online]. Available: https://arxiv.org/abs/2103.00218

[10] R. Templeman, M. Korayem, D. Crandall, and A. Kapadia, "PlaceAvoider: Steering first-person cameras away from sensitive spaces," in *Proc. Netw. Distrib. Syst. Secur. Symp.*, 2014, pp. 23–26.

[11] O. Gafni, L. Wolf, and Y. Taigman, "Live face de-identification in video," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 9377–9386, doi: 10.1109/ICCV.2019.00947.

[12] M. Maximov, I. Elezi, and L. Leal-Taixe, "CIAGAN: Conditional identity anonymization generative adversarial networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Seattle, WA, USA, Jun. 2020, pp. 5446–5455, doi: 10.1109/CVPR42600.2020.00549.

[13] Q. Sun, L. Ma, S. J. Oh, L. V. Gool, B. Schiele, and M. Fritz, "Natural and effective obfuscation by head inpainting," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Salt Lake City, UT, USA, Jun. 2018, pp. 5050–5059. [Online]. Available: https://openaccess.thecvf.com/content_cvpr_2018/html/Sun_Natural_and% _Effective_CVPR_2018_paper.html

[14] M. Fredrikson, S. Jha, and T. Ristenpart, "Model inversion attacks that exploit confidence information and basic countermeasures," in *Proc. 22nd ACM SIGSAC Conf. Comput. Commun. Secur.*, I. Ray, N. Li, and C. Kruegel, Eds. Denver, CO, USA, Oct. 2015, pp. 1322–1333.

[15] M. Gong, Y. Xie, K. Pan, K. Feng, and A. K. Qin, "A survey on differentially private machine learning [review article]," *IEEE Comput. Intell. Mag.*, vol. 15, no. 2, pp. 49–64, May 2020.

[16] J. I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 2672–2680.

[17] X. Huang, M.-Y. Liu, S. Belongie, and J. Kautz, "Multimodal unsupervised image-to-image translation," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2018, pp. 172–189.

[18] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1125–1134.

[19] L. Karacan, Z. Akata, A. Erdem, and E. Erdem, "Learning to generate images of outdoor scenes from attributes and semantic layouts," 2016, *arXiv:1612.00215*. [Online]. Available: https://arxiv.org/abs/1612.00215

[20] L. Karacan, Z. Akata, A. Erdem, and E. Erdem, "Manipulating attributes of natural scenes via hallucination," 2018, *arXiv:1808.07413*. [Online]. Available: https://arxiv.org/abs/1808.07413

[21] M.-Y. Liu, T. Breuel, and J. Kautz, "Unsupervised image-to-image translation networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 700–708.

[22] B. Zhao, L. Meng, W. Yin, and L. Sigal, "Image generation from layout," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 8584–8593.

[23] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2223–2232.

[24] J.-Y. Zhu, R. Zhang, D. Pathak, T. Darrell, and A. A. Efros, "Toward multimodal image-to-image translation," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 465–476.

[25] T. Park, M.-Y. Liu, T.-C. Wang, and J.-Y. Zhu, "Semantic image synthesis with spatially-adaptive normalization," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 2337–2346.

[26] R. Mur-Artal and J. D. Tardós, "ORB-SLAM2: An open-source SLAM system for monocular, stereo and RGB-D cameras," *IEEE Trans. Robot.*, vol. 33, no. 5, pp. 1255–1262, Jun. 2017.

[27] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2016, pp. 694–711.

[28] C. Li and M. Wand, "Precomputed real-time texture synthesis with Markovian generative adversarial networks," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2016, pp. 702–716.

[29] C. Ledig, L. Theis, F. Huszar, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi, "Photo-realistic single image super-resolution using a generative adversarial network," 2016, *arXiv:1609.04802*. [Online]. Available: https://arxiv.org/abs/1609.04802

[30] C.-H. Lee, Z. Liu, L. Wu, and P. Luo, "Maskgan: Towards diverse and interactive facial image manipulation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Seattle, WA, USA, Jun. 2020, pp. 5548–5557.

[31] S. Zhang, X. Zhu, Z. Lei, H. Shi, X. Wang, and S. Z. Li, "Faceboxes: A CPU real-time face detector with high accuracy," in *Proc. IEEE Int. Joint Conf. Biometrics (IJCB)*, Oct. 2017, pp. 1–9.

[32] V. Le, J. Brandt, Z. Lin, L. D. Bourdev, and T. S. Huang, "Interactive facial feature localization," in *Proc. 12th Eur. Conf. Comput. Vis. (ECCV)*, vol. 7574, 2012, pp. 679–692.

[33] H.-W. Ng and S. Winkler, "A data-driven approach to cleaning large face datasets," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2014, pp. 343–347.

[34] H. Lee, M. U. Kim, Y. Kim, H. Lyu, and H. Jong Yang, "Privacy-protection drone patrol system based on face anonymization," 2020, *arXiv:2005.14390*. [Online]. Available: https://arxiv.org/abs/2005.14390

[35] K. Gregory, R. Zemel, and R. Salakhutdinov, "Siamese neural networks for one-shot image recognition," in *Proc. Int. Conf. Mach. Learn. (ICML) Deep Learn. Workshop*, vol. 2, 2015.

[36] S. L. D. Yi, Z. Lei, and S. Z. Li, "Learning face representation from scratch," 2014, *arXiv:1411.7923*. [Online]. Available: https://arxiv.org/abs/1411.7923
[37] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1–9.

HARIM LEE received the B.S. degree in electrical engineering from Kyungpook National University, Daegu, South Korea, in 2013, the M.S. degree in IT convergence engineering from Pohang University of Science and Technology (POSTECH), Pohang, South Korea, in 2015, and the Ph.D. degree from the School of Electrical and Computer Engineering, Ulsan National Institute of Science and Technology (UNIST), Ulsan, South Korea, in August 2020. From September 2020 to August 2021, he worked as a Postdoctoral Researcher with the Department of Electrical Engineering, POSTECH. Since September 2021, he has been an Assistant Professor with the School of Electronic Engineering, Kumoh National Institute of Technology, Gumi, Gyungbuk, South Korea. His research interests include autonomous UAV systems, privacy-protecting deep learning, PHY & MAC for the next generation mobile networks, and embedded systems and robotics, such as licensed assisted access in LTE (LTE-LAA), wireless networking with deep neural networks, and radar systems. He received Kwanjeong Educational Foundation Fellowship, from 2013 to 2014.

MYEUNG UN KIM received the B.S. degree in computer science engineering from Ulsan National Institute of Science and Technology (UNIST), Ulsan, South Korea, in 2015, and the combined M.S. and Ph.D. degree in electrical engineering from UNIST, in August 2020. From August 2019 to February 2020, she studied as a Visiting Student with the School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, USA, supported by the Institute for Information and Communication Technology Planning and Evaluation (IITP), Seoul, South Korea. From May 2020 to June 2020, she worked as a Postdoctoral Researcher at Korea Institute of Science and Technology (KIST), Seoul. Since July 2020, she has been a Researcher at Korea Aerospace Research Institute (KARI), Daejeon, South Korea. Her research interests include robot vision and deep learning.

YEONGJUN KIM received the B.S. degree in electrical engineering from Ulsan National Institute of Science and Technology (UNIST), Ulsan, South Korea, in 2016, where he is currently pursuing the combined M.S. and Ph.D. degree with the School of Electrical and Computer Engineering. His research interests include software defined radio, wireless networking with deep neural networks, and UAV systems.

HYEONSU LYU received the B.S. degree in mathematical science and the M.S. degree in electrical engineering from Ulsan National Institute of Science and Technology (UNIST), Ulsan, South Korea, in 2018 and 2020, respectively. He has been a Ph.D. student in the Department of Electrical Engineering, POSTECH, since March 2021. His research interests include intelligent UAV system and optimization theory.

HYUN JONG YANG (Member, IEEE) received the B.S., M.S., and Ph.D. degrees in electrical engineering from Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Republic of Korea, in 2004, 2006, and 2010, respectively. From August 2010 to August 2011, he was a Research Fellow at Korea Institute Ocean Science Technology (KIOST), Daejoen. From October 2011 to October 2012, he worked as a Postdoctoral Researcher with the Electrical Engineering Department, Stanford University, Stanford, CA, USA. From October 2012 to August 2013, he was a Staff II Systems Design Engineer at Broadcom Corporation, Sunnyvale, CA, USA, where he developed physical-layer algorithms for LTE-A MIMO receivers. In addition, he was a delegate of Broadcom in 3GPP standard meetings for RAN1 Rel-12 technologies. From September 2013 to July 2020, he was an Assistant/an Associate Professor with the School of Electrical and Computer Engineering, UNIST, Ulsan, South Korea. Since July 2020, he has been an Associate Professor with the Department of Electrical Engineering, Pohang University of Science and Technology (POSTECH), Pohang, South Korea. His research interests include privacy-preserving robot systems, deep-learning theory and algorithms, and signal processing.