DuetFace: Collaborative Privacy-Preserving Face Recognition via Channel Splitting in the Frequency Domain

Yuxi Mi  
yxmi20@fudan.edu.cn  
Fudan University  
Shanghai, China

Yuge Huang  
yugehuang@tencent.com  
Tencent Youtu Lab  
Shanghai, China

Jiazhen Ji  
royji@tencent.com  
Tencent Youtu Lab  
Shanghai, China

Hongquan Liu  
hqliu21@m.fudan.edu.cn  
Fudan University  
Shanghai, China

Xingkun Xu  
xingkunxu@tencent.com  
Tencent Youtu Lab  
Shanghai, China

Shouhong Ding  
ericshding@tencent.com  
Tencent Youtu Lab  
Shanghai, China

Shuigeng Zhou  
sgzhou@fudan.edu.cn  
Fudan University  
Shanghai, China

ABSTRACT

With the wide application of face recognition systems, there is rising concern that original face images could be exposed to malicious intents and consequently cause personal privacy breaches. This paper presents DuetFace, a novel privacy-preserving face recognition method that employs collaborative inference in the frequency domain. Starting from a counterintuitive discovery that face recognition can achieve surprisingly good performance with only visually indistinguishable high-frequency channels, this method designs a credible split of frequency channels by their cruciality for visualization and operates the server-side model on non-crucial channels. However, the model degrades in its attention to facial features due to the missing visual information. To compensate, the method introduces a plug-in interactive block to allow attention transfer from the client-side by producing a feature mask. The mask is further refined by deriving and overlaying a facial region of interest (ROI). Extensive experiments on multiple datasets validate the effectiveness of the proposed method in protecting face images from undesired visual inspection, reconstruction, and identification while maintaining high task availability and performance. Results show that the proposed method achieves a comparable recognition accuracy and computation cost to the unprotected ArcFace and outperforms the state-of-the-art privacy-preserving methods. The source code is available at https://github.com/Tencent/TFace/tree/master/recognition/tasks/duetface.

CCS CONCEPTS

- Security and privacy → Privacy protections;  
- Computer systems organization → Neural networks.

KEYWORDS

face recognition, data privacy, deep learning, channel splitting

1 INTRODUCTION

Face recognition (FR) has been a phenomenal biometric method for identity authentication, with lots of remarkable breakthroughs gained in recent years. As face recognition is widely incorporated into applications in such as finance, health, and public security, there rises a growing concern about the privacy of sensitive facial images. The privacy-preserving face recognition (PPFR) technique thus has arrested the attention of academia and the industry.

In a typical privacy-preserving face recognition scenario, a query face image is collected and held by local devices such as cell phones and webcams. As often constrained by computation power, they outsource the recognition task to a third-party service provider, who infers on a SOTA model pre-trained on massive labeled datasets. For clarity, we thereafter refer to the image holder as the client and the service provider as the server. As the query face here is considered private, the client is not willing to share the raw image with others, including the server.

In the recent decade, vast advancements have been incorporated into PPFR applications. We coarsely categorize them into two branches: the encrypt-based methods build the privacy of the face image on the bricks of cryptographic primitives and security...
We exploit this discovery as a starting point for our work. We propose a novel way to effectively solve the privacy-accuracy paradox by addressing them successively through a collaboration between the server and the client. Concretely, let the server possess a SOTA model, whereas the client holds a lightweight model that is an order of magnitude smaller in complexity and cost. We first split the frequency channels into two parts by their cruciality for visualization, and let the server infer the non-crucial part. The visual information is therefore concealed from the server and, by transferring the client-side attention to the server through a proposed interactive block, a recognition is performed through the collaboration of the server and client.

Figure 1: A paradigm comparison among unprotected FR (top), prior PPFR methods (middle) and our method (bottom). The images are directly shared in unprotected FR, while are encrypted or transformed in PPFR. The server in prior works performs recognition alone, which usually suffers from a downgrade in accuracy. In our method, the recognition is performed through the collaboration of the server and client.

Our Method
Unprotected FR
Prior PPFR Methods

In summary, the contributions of our paper are three-fold:

1. We propose a novel face recognition paradigm by combining the collaborative efforts of two parties: the client and the server. We develop a novel PPFR framework, referred to as DuetFace, which is satisfactory in recognition accuracy, good in cost-efficiency, and reliable in privacy protection;
2. We introduce a channel splitting scheme to derive appropriate image break-ups and devise an interactive block based on ROI-refined feature masks to allow attention transfer;
3. We conduct extensive experiments on multiple datasets, which validate the effectiveness and superiority of the proposed method.

2 RELATED WORK

2.1 Privacy-Preserving Face Recognition

Significant advances of privacy-preserving face recognition (PPFR) has been achieved significant progress in the past decade, which can be roughly divided into two categories:

Encrypt-based methods. In this branch of work, face recognition is carried out on encrypted domains. Necessary computations such as feature extraction and similarity calculation are either accomplished straightly on the encrypted images or by executing certain security protocols. Pioneering works [9, 15, 27] apply homomorphic encryption (HE) and garbled circuits (GC) to the Eigenface recognition algorithm to hide raw image features from undesired parties. Methods with similar intentions also employ other cryptographic primitives including matrix encryption [17], one-time-pad [8], and functional encryption [1]. Dedicated secure multiparty computation (MPC) schemes are introduced in [19, 38, 41] to perform certain operations (e.g., parameter comparison) in a protected manner, or to outsource parts of the job to a trusted third party.

Encrypt-based methods have very little degradation on the recognition accuracy since almost all the operations involved are lossless. Their effectiveness is also strongly guaranteed by the provable or computational security of the employed cryptographic primitives. However, the practical usages of these methods are limited as they mostly bear intolerable computation costs and communication overhead. Moreover, these methods have low generalizability since most of them are tightly coupled with very specific face recognition schemes. Our work is more generalizable and can achieve competitive accuracy to the standard ArcFace with much less cost.

Transform-based methods. Another active line of research transforms face images into perturbed or regenerated representations to reduce their distinguishability from untrusted parties. For perturbation, differential privacy is employed by lots of works [2, 18, 20, 42] where raw images are overlaid by noise mechanisms to make them less visually differentiable. To increase anonymity, Honda et al. [13] proposed a clustering-based method by mapping raw images to their class representations. As for means to regenerate images into new representations, some works exploit deep-learning-based methods such as adversarial generative network (GAN) [18, 24] and
We here describe the proposed privacy-preserving face recognition with lower frequency, which in the meantime are those with larger work. Further, PPFR-FD realizes privacy mainly by randomizing works study the contribution of channels on visualization and recognition. To illustrate this, we train multiple recognition models, each with different numbers of lowest-frequency channels discarded in every work. However, PPFR-FD begins with the premise that the lowest frequency channel contributes not much to the recognition task, which is quite different from ours; (2) The high-frequency channels are removed in PPFR-FD as they are believed to contribute little to distinguishability. These channels are instead retained and fully exploited in our work. Further, PPFR-FD realizes privacy mainly by randomizing the order of channels, while ours by removing visual components.

2.2 Learning in the Frequency Domain
Learning in the frequency domain is traditionally leveraged for image compression, which allows retaining meaningful patterns for image understanding tasks through compressed representations. Prior arts in face recognition [30, 40] train autoencoder-based networks to perform compression and inference tasks simultaneously. [6] first performs image classification in the frequency domain directly. [39] proposes an accuracy-retaining image down-sampling method, where spatial images are reorganized in the frequency domain to remove the non-crucial channels.

3 METHODOLOGY
3.1 Overview
We here describe the proposed privacy-preserving face recognition method, referred to as DuetFace. In the world of art, a duet is a performance by two singers, instrumentalists, or dancers. Similarly, in DuetFace, the inference is carried out together by two parties, i.e., the server and the client.

Our discovery and motivation. Our work starts from a counterintuitive discovery. Previous research in the frequency domain [33] suggests that the recognition is mainly determined by channels with lower frequency, which in the meantime are those with larger amplitude [39], as they contribute most of the visual information. Yet, those prior arts mostly ignore the value of the rest “non-crucial” channels. We, on the contrary, surprisingly find that the models can also perform recognition with quite acceptable accuracy by using only the visually indistinguishable high-frequency channels. To illustrate this, we train multiple recognition models, each with different numbers of lowest-frequency channels discarded in every color component, then evaluate the trained models on 5 public datasets. Results in Fig. 3(b)(c) show that the models are able to maintain a quite decent performance even the remaining channels possess <10% energy, although there certainly exists an accuracy gap due to the lack of visual information. This discovery allows us to construct our PPFR method from a completely different view.

Concretely, we introduce a collaborative paradigm between the server and the client. We first design an appropriate split of the query image in the frequency domain, and let the server train on the high-frequency components in a privacy-preserving manner with tolerable accuracy loss. Then, we let the client further refine the server-side performance by compensating for the missing information without revealing the image itself.

The ability of the server and the client. To better explain the motivation of our method, we start by characterizing the parties. We assume the server to be semi-honest and the client to be resourcesensitive. A semi-honest server is one who honestly follows the face recognition protocol and provides correct results while trying to learn as much as possible from the messages sent by the client. The server is an abstraction of corrupted or unregulated service providers, who may collect, use, and redistribute face images unauthorized. A resource-sensitive client is one bounded by limited storage, bandwidth, and computation power. The real-world clients are often reified as personal devices such as cell phones and webcams, whose owners may be unwilling to download and store large models locally or perform complex inference tasks.

The security goals of DuetFace. We address the inference-time privacy between the semi-honest server $S$ and the client $C$. The client possesses a query face image $X$ that it wants the server to identify. The image is considered private. We denote all the information related to $X$ that the client could share without causing a privacy breach as $I(X)$. Our privacy consideration is to prevent the unauthorized collection, use, and redistribution of $X$, which we concretize into three security goals:

(1) Visual privacy. As the very basis, the server should be unable to collect useful information from the visual appearance of the face image $X$;

(2) Privacy against reconstruction. The server may try to reconstruct the information it misses about $X$. We ergo prescribe, by leveraging $I(X)$, the server cannot effectively produce a reconstruction $X'$ of $X$;

(3) Privacy against identity inference. The server could redistribute the reconstructed image to a third party. Plus, the message could be intercepted during transmission. To prevent potential privacy leakage, we require that, without accessing the recognition model, acquiring either $I(X)$ or $X'$ should be insufficient to infer the identity of $X$.

The paradigm of DuetFace. Let the server and the client each hold a local model $M_s$ and $M_c$, respectively. Here, $M_s$ is a full-size state-of-the-art model used to answer the recognition requests, whereas $M_c$ is a lightweight model applied as an aid. Both models are pre-trained by the server and $M_c$ is downloaded by the client. To infer a query image $X$, the client first splits $X$ into two appropriate parts $X_s$ and $X_c$, where visual information is removed from $X_s$ but retained in $X_c$. The client shares $X_s$ with the server and keeps $X_c$ to itself. The server identifies $X_s$ via its model $M_s$. Note that the loss of visual information in $X_s$ will surely degrade the performance of the server-side model. To compensate for the information loss, the client infers the other part $X_c$ of the image $X$ on $M_c$ at a very low computation cost, to obtain a concise supplementary representation $R(X_c)$, which will be revealed to the server. Finally, the server leverages $R(X_c)$ to refine its judgment and produce better results.
3.2 Splitting Channels in Frequency Domain
As the very first step of our proposed method, the image \( X \) is split by channel frequency. To transform the face image to the frequency domain, we first follow the common data pre-processing protocol in the spatial domain to crop, resize and horizontally flip the face image, and obtain an input shape of \( H \times W \times 3 \). Then, we convert the image from RGB to YCbCr color space, and subsequently to the frequency domain by carrying out the block discrete cosine transform (BDCT) following the same way in JPEG compression [32]. We also perform an 8-fold bilinear up-sampling right before BDCT. As standard BDCT maps each 8×8 pixel block into one frequency channel (here, maps \( 8H \times 8W \times 3 \) to \( H \times W \times 192 \)), the up-sampling enables us to maintain the channel shape and minimize the modifications to the recognition backbone.

We construct a credible split \( \{ X_s, X_c \} \) of \( X \) by the amplitude of channels. Here, we measure amplitude by channel energy: given a channel, its energy is the mean of absolute values of all its elements. We first split channels by energy on the luma (Y) component of YCbCr color space, as it carries most of the visual profiles and features [39]. As shown in Fig. 3(a)(b), the low-frequency channels on the top-left corner contribute to >90% of total energy. We select \( K \) channels with the highest energy as the crucial channels and regard the rest as non-crucial ones. To achieve spatial consistency, the same selection is also applied to the chroma (Cb, Cr) components. Then, we produce \( \{ X_s, X_c \} \) utilizing the split channels. We form \( X_c \) by concatenating the crucial channels (in the shape of \( H \times W \times 3K \)), and \( X_s \) by the rest. Therefore, visual information is convincingly removed from \( X_s \) but retained in \( X_c \), as illustrated in Sec. 4.3.

We adjust the input shapes of the models \( M_s \) and \( M_c \) to meet the shapes of \( X_s \) and \( X_c \), respectively. It is widely observed that discarding non-crucial frequency channels affects the model very slightly [33, 35, 39], so we expect no performance change on \( M_c \). As of \( M_s \), results in Sec. 4.5 show that the model suffers a tolerable accuracy gap in the absence of visual information, which we are to compensate for in the following subsections.

3.3 Attention Transfer
We ascribe the downgrade of \( M_s \) to the inaccuracy of model attention. To explain, we visualize the top-down attention maps for each layer of \( M_s \) and \( M_c \) via Grad-CAM [28]. As shown in Fig. 4, after discarding the crucial channels, the attention of the server-side model \( M_s \) could not focus on the effective visual features such as facial contours and the positions of eyes, noses, and lips, which are generally considered the indispensable information for high-quality
We subsequently refine the feature mask by removing noisy features. To explain, Fig. 5(b) illustrates some counter-examples that the feature mask can sometimes be ineffective as it incorrectly highlights the hair, the headgear, and the surrounding area, rather than the face itself. Although these features are inherent and could be harmless in the standard face recognition process, in our case, they bring in undesirable noise during attention transfer.

We denoise the mask by obtaining a region of interest (ROI) on the raw image. Facial landmark detection is a proven technique that detects and tracks key points in a human face and is widely adopted in applications such as augmented reality. A general facial landmark detector produces a sequence of points that mark the positions of main facial features between facial contours and eyebrows, which specifies the interesting regions of our feature mask. Therefore, we utilize the point sequence to derive the facial ROI.

Specifically, right before performing BDCT, we pass the image $X$ through a pre-trained facial landmark detector to obtain the sequence of landmark points $P$. Here, we employ an open-source PFLD model [10]. Note that PFLD can be replaced by an arbitrary lightweight 2D landmark detector. Subsequently, we derive the facial ROI, which wraps the facial region, by calculating the convex hull $H(P)$ of $P$ using Delaunay triangulation. Features outside the ROI are considered useless for attention transfer. To remove them, we turn $H(P)$ into a mask by marking all the pixels inside as 1 and the rest as 0, and overlay it on its feature mask:

$$R'(X_c) = H(P) \odot R_i(X_c).$$

Ergo, a clean mask is produced. Finally, we normalize $R'(X_c)$ and use it as a replacement of $R_i(X_c)$.

Fig. 6 show sample masks before and after applying the facial ROI. It can be clearly observed that the ROI removes the features of the surroundings, resulting in focused attention to the face. Results in Sec. 4.5 also show a performance lift after employing the ROI.
We set its accuracy-privacy parameter to 100; (5) which almost fills in the accuracy gap with the standard ArcFace, Training datasets. We employ MS1Mv2 as training set, plus BUPT-10K faces, each with 98 manual annotated landmarks. To illustrate the effectiveness of our method, we randomly sample a query image and the refined feature masks \(X_c\) at each stage. To evaluate the effectiveness of our method, we randomly sample face images and visualize their \(X_c\), \(X_s\) and \( R_i(X_c) \) at each stage. As \(X_s\) is in the frequency domain, we pad its removed channels with zero and convert it back to RGB by performing inverse DCT. We perform the same on \(X_c\) for comparison. The results in Fig. 7(a) show that most of the perceptible visual information is removed from \(X_c\) but retained in \(X_s\), which prevents the server from inspecting \(X_c\) while allowing the client to infer \(X_c\) (in private) normally. As for the feature masks, the results show that they reveal only very limited visual information such as the approximate facial contours, which would not sabotage our security goal.

4.3 Visual Privacy

We validate that our proposed method provides reliable privacy protection that satisfies the security goals stated in Sec. 3.1. The basic concern among our goals is visual security, i.e., the server should not be able to visually inspect the query image by the information it acquires from the client. **Visualization of the client/server-side components and feature masks.** During the whole inference process, the information the server obtains including the non-crucial components \(X_c\) of the query image and the refined feature masks \( R_i(X_c) \) at each stage.

4.4 Privacy Against Malicious Intents

Ever since its invention, face recognition is under the threat of malicious attacks. Our security goal in Sec. 3.1 addresses concerns...
Table 1: Comparisons with State-of-the-Art Methods

| Method             | PPFR | LFW | CFP-FP | AgeDB | CPLFW | CALFW | IJB-B(TPR@FPR) | IJB-C(TPR@FPR) |
|--------------------|------|-----|--------|-------|-------|-------|----------------|----------------|
| ArcFace [5]        | No   | 99.77 | 98.30  | 97.88 | 92.77 | 96.05 | 94.13          | 95.60          |
| ArcFace-FD [6]     | No   | 99.78 | 98.04  | 98.10 | 92.48 | 96.03 | 94.08          | 95.64          |
| PEEP [2]           | Yes  | 98.41 | 74.47  | 87.47 | 79.58 | 90.06 | 5.82           | 6.02           |
| Cloak [22]         | Yes  | 98.91 | 87.97  | 92.60 | 83.43 | 92.18 | 33.58          | 33.82          |
| InstaHide [15]     | Yes  | 96.53 | 83.20  | 79.58 | 81.03 | 86.24 | 61.88          | 69.02          |
| CPGAN [31]         | Yes  | 98.87 | 94.61  | 96.98 | 90.43 | 94.79 | 92.67          | 94.31          |
| PPFR-FD [35]       | Yes  | 99.68 | 95.04  | 97.37 | 90.78 | 95.72 | *             | 94.10          |
| DuetFace (ours)    | Yes  | 99.82 | 97.79  | 97.93 | 92.35 | 96.10 | 93.66          | 95.30          |

* The results of PPFR-FD are quoted from [34] due to the lack of source code. Please note that its experimental condition may be different slightly from ours.

Table 2: Effectiveness Against Reconstruction and Malicious Identity Inference

| Target                      | SSIM↓ | PSNR↓ | Accuracy↓ |
|-----------------------------|-------|-------|-----------|
| Raw image                   | 0.9993| 48.30 | 99.80     |
| Reconstruction of Xs         | 0.5395| 13.01 | 51.52     |
| Reconstruction of R_0'(Xc)   | 0.4422| 12.45 | 57.18     |
| Reconstruction of F_0'(Xs)   | 0.4555| 12.63 | 61.97     |

Effectiveness against reconstruction. Autoencoders are widely used for reconstruction. In our case, the server may attempt to reconstruct the face image by an autoencoder-based network. Accessible information for the server including the non-crucial channels, the feature masks and their combination, which equals to the masked server-side feature map $F_0'(X_s)$. Ergo, we first infer images randomly picked from MS1Mv2 on a pre-trained DuetFace to collect their $X_s$, $R_0'(X_c)$ and $F_0'(X_s)$. Note that in our case, as the resolution of feature maps is successively divided by half in each stage, the reconstruction is most likely to succeed on stage 0. We train three autoencoder-based U-Net models, denoted as $U_1$, $U_2$ and $U_3$, on $X_s$, $R_0'(X_c)$ and $F_0'(X_s)$, respectively, then use the trained model for reconstruction. We quantify the quality of the reconstructed images by structural similarity index (SSIM, as compared to the raw image) and peak signal-to-noise ratio (PSNR). As shown in Fig. 8, none of $U_1$, $U_2$ and $U_3$ manages to effectively reconstruct an image that embodies distinguishable features. And in Tab. 2, we can see that the SSIM and PSNR values are low for all three cases. These verify the robustness of our method against reconstruction.

Effectiveness against identity inference. We subsequently feed $X_s$ (after transformed into the spatial domain) and the reconstructed
To demonstrate this, we remove the interactive block as well as the facial ROI to focus its attention on the facial region. We remove the facial ROI to refine the feature mask by focusing its attention on the facial region. We remove the interactive block, ROI and server-side model, respectively. Effect of the interactive block. The server leverages client-side features to compensate for its inaccurate attention to facial features. To demonstrate this, we remove the interactive block as well as the client-side model completely and train the server-side model only accounts for a 6.7% increase of the whole model size and takes about 10M of storage space at the client-side. Inference time. We perform inference on images with a batch size of 64 and record the average per-batch time. Here, the inference is performed in an asynchronous manner, i.e., the client completes the local computations on all query images, then hands it over to the server for the rest. As compared to ArcFace, our total inference time increased by ×2.6. We think such an increase is marginal as it is the communication time that predominates in real-world practice. The overall time cost is still within a decent scope.

4.6 Complexity and Cost
To demonstrate the resource-friendliness of our method, we summarize the model size, time cost, and communication overhead required for inference in Tab. 4. As the availability of our method mainly depends on the client-side budget, for a clear demonstration, we list the space and time costs of the client and server separately. The PFLD model is included when evaluating the client’s cost. Model size. We list the model size by the number of parameters. Since both MobileFaceNet and PFLD are an order of magnitude smaller than the server-side model, the employment of the local model only accounts for a 6.7% increase of the whole model size and takes about 10M of storage space at the client-side.

| Year | LFW | CFP-FP | AgeDB | CPLFW | CALFW |
|------|-----|-------|-------|-------|-------|
| 2022 | 99.39 | 93.46 | 95.12 | 89.03 | 94.86 |

Table 3: Ablation Study Results

| Method | LFW | CFP-FP | AgeDB | CPLFW | CALFW |
|--------|-----|-------|-------|-------|-------|
| IR-50 + MobileFaceNet, on MS1Mv2 | 99.77 | 98.30 | 97.88 | 92.77 | 96.05 |
| ArcFace [5] | 99.82 | 97.79 | 97.93 | 92.35 | 96.10 |
| DuetFace | 99.70 | 95.86 | 97.57 | 90.82 | 95.86 |
| w/o IB | 99.78 | 97.23 | 98.75 | 92.07 | 96.06 |
| w/o ROI | 99.48 | 93.91 | 96.10 | 89.68 | 95.08 |
| w/o Msc | 99.82 | 97.79 | 97.93 | 92.35 | 96.10 |

| Method | LFW | CFP-FP | AgeDB | CPLFW | CALFW |
|--------|-----|-------|-------|-------|-------|
| IR-18 + MobileFaceNet, on BUPT | 99.52 | 94.06 | 94.95 | 90.05 | 95.07 |
| ArcFace [5] | 99.40 | 93.79 | 95.07 | 89.67 | 95.03 |
| DuetFace | 99.16 | 91.96 | 94.18 | 87.67 | 94.27 |
| w/o IB | 99.39 | 93.46 | 95.12 | 89.03 | 94.86 |
| w/o ROI | 99.32 | 92.43 | 93.42 | 89.15 | 93.62 |

We analyze the effects of the major components of DuetFace by testing the performance when removing one of them. Results are reported in Tab. 3 on LFW, CFP-FP, AgeDB, CPLFW, and CALFW. To validate the generality of our method, we also report the results on the combination of the IR-18 model and BUPT dataset. Effect of the interactive block. The server leverages client-side features to compensate for its inaccurate attention to facial features. To demonstrate this, we remove the interactive block as well as the client-side model completely and train the server-side model only accounts for a 6.7% increase of the whole model size and takes about 10M of storage space at the client-side. Inference time. We perform inference on images with a batch size of 64 and record the average per-batch time. Here, the inference is performed in an asynchronous manner, i.e., the client completes the local computations on all query images, then hands it over to the server for the rest. As compared to ArcFace, our total inference time increased by ×2.6. We think such an increase is marginal as it is the communication time that predominates in real-world practice. The overall time cost is still within a decent scope.

Communication overhead. We calculate by adding up the number of elements in the server-side components of all 4 stages \( R_i(X) \). The baseline ArcFace transfers the RGB images of \( 3 \times 112 \times 112 \) directly, resulting in \( 37,632 \) elements. In our original purpose, each image is up-sampled before BDCT (as means to preserve the same input height and width), which causes a larger communication cost. We can overcome this by moving the up-sampling later to the server-side. Therefore, the server-side components of all 4 stages \( R_i(X) \). The baseline ArcFace transfers the RGB images of \( 3 \times 112 \times 112 \) directly, resulting in \( 37,632 \) elements. In our original purpose, each image is up-sampled before BDCT, and then it is transferred to the server as a facial ROI. Extensive experiments show the proposed method is satisfactory in recognition accuracy, with good cost-efficiency, and achieves high reliability in privacy protection.
In 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020. Computer Vision Foundation / IEEE, 8641–8649. https://doi.org/10.1109/CVPR42600.2020.00871

[34] Mei Wang and Weihong Deng. 2019. Mitigate Bias in Face Recognition using Skewness-Aware Reinforcement Learning. (2019). arXiv:1911.10692 http://arxiv.org/abs/1911.10692

[35] Yinggui Wang, Jian Liu, Man Luo, Le Yang, and Li Wang. 2022. Privacy-Preserving Face Recognition in the Frequency Domain. (2022). (in press).

[36] Cameron Whitelam, Emma Taborsky, Austin Blanton, Brianna Maze, Jocelyn C. Adams, Tim Miller, Nathan D. Kalka, Anil K. Jain, James A. Duncan, Kristen Allen, Jordan Cheney, and Patrick Grother. 2017. IARPA Janus Benchmark-B Face Dataset. In 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2017, Honolulu, HI, USA, July 21-26, 2017. IEEE Computer Society, 592–600. https://doi.org/10.1109/CVPRW.2017.87

[37] Wayne Wu, Chen Qian, Shuo Yang, Quan Wang, Yici Cai, and Qiang Zhou. 2018. Look at Boundary: A Boundary-Aware Face Alignment Algorithm. In 2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018. Computer Vision Foundation / IEEE Computer Society, 2129–2138. https://doi.org/10.1109/CVPR.2018.00227

[38] Can Xiang, Chunming Tang, Yunlu Cai, and Quixia Xu. 2016. Privacy-preserving face recognition with outsourced computation. Soft Comput. 20, 9 (2016), 3735–3744. https://doi.org/10.1007/s00500-015-1759-5

[39] Kai Xu, Minghui Qin, Fei Sun, Yuhao Wang, Yen-Kuang Chen, and Fengbo Ren. 2020. Learning in the Frequency Domain. In 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020. Computer Vision Foundation / IEEE, 1737–1746. https://doi.org/10.1109/CVPR42600.2020.00181

[40] Kai Xu, Zhikang Zhang, and Fengbo Ren. 2018. LAPFRAN: A Scalable Laplacian Pyramid Reconstructive Adversarial Network for Flexible Compressive Sensing Reconstruction. In Computer Vision – ECCV 2018 – 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part X (Lecture Notes in Computer Science, Vol. 11214), Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss (Eds.). Springer, 491–507. https://doi.org/10.1007/978-3-030-01249-6_30

[41] Xiaopeng Yang, Hui Zhu, Xiaolong Li, and Hui Li. 2018. Efficient and Privacy-Preserving Online Face Recognition Over Encrypted Outsourced Data. In IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), iThings/GreenCom/CPSCom/SmartData 2018, Halifax, NS, Canada, July 30 – August 3, 2018. IEEE, 366–373. https://doi.org/10.1109/Cybermatics_2018.2018.00089

[42] Chen Zhang, Xiongwei Hu, Yu Xie, Maoguo Gong, and Bin Yu. 2019. A Privacy-Preserving Multi-Task Learning Framework for Face Detection, Landmark Localization, Pose Estimation, and Gender Recognition. Frontiers Neurorobotics 13 (2019), 112. https://doi.org/10.3389/fnbot.2019.00112

[43] T. Zheng and W. Deng. 2018. Cross-pose LFW: A database for studying cross-pose face recognition in unconstrained environments. Technical Report 18-01. Beijing University of Posts and Telecommunications.

[44] Tianyue Zheng, Weihong Deng, and Jiani Hu. 2017. Cross-Age LFW: A Database for Studying Cross-Age Face Recognition in Unconstrained Environments. (2017). arXiv:1708.08197 http://arxiv.org/abs/1708.08197

A IMPLEMENTATION DETAILS

A.1 Backbone

We use the adapted ResNet50 with an improved residual unit (IR-50) [12] as the server-side backbone $M_s$, which has better convergence in the early training stages, and the MobileFaceNet [4] as the client-side $M_c$. For the client-side facial landmark detection, we apply a PFLD [10] network. Note that both MobileFaceNet and PFLD are lightweight backbones dedicated to resource-constraint devices such as cell phones. We also adapt a smaller IR-18 backbone for ablation study, to explain the generality of our method on different network architectures.

A.2 Preparation

We crop and resize each image to $112 \times 112$ pixels and add random flipping as image augmentation, then apply BDCT to obtain 192 frequency components by an open-source TorchJPEG library [7]. After transforming into the frequency domain, we calculate the energy, and select 10 channels from each of the Y, Cb, Cr components as stated in Sec. 3.2. This results in 30 channels selected in total. Note that we choose 30 without loss of generality and it is not the only choice. Therefrom, we form an $X_i$ in the shape of $112 \times 112 \times 162$ and an $X_c$ of $112 \times 112 \times 30$.

We alter the model input channels to meet the shapes of $X_i$ and $X_c$. Each of the IR-18/50 and MobileFaceNet models contains 4 stages. For attention transfer, we insert an interactive block $IB$ at the end of each stage. Aligning with the feature map shapes in $X_i$, in our case, the 4 feature masks are resized to the height and width of 56, 28, 14 and 7, respectively. The other parts of the models remain unchanged.

A.3 Training

The IR-18/50 and MobileFaceNet models are trained for 24 epochs on the same dataset (either MS1Mv2 or BUPT), using the ArcFace [5] loss. We use the stochastic gradient descent (SGD) optimizer, which is applied with an initial learning rate of 0.1, a momentum of 0.9, and a weight decay of 5e-4. As of PFLD, we apply a learning rate and a weight decay of 1e-4 and 1e-6, respectively, at a batch size of 512. We successively divide the learning rate by 10 at stages 10, 18, and 22. As of PFLD, we apply a learning rate and a weight decay of 1e-4 and 1e-6, respectively, and train it until convergence. Experiments are conducted on 8 NVIDIA Tesla V100 GPU under the PyTorch framework. The same random seed is sampled for all experiments for fairness.