Deep Compressive Sensing for Visual Privacy Protection in FlatCam Imaging

Thuong Nguyen Canh and Hajime Nagahara
Insitute for Datability Science, Osaka University, Osaka, Japan
{ngcthuong, nagahara}@ids.osaka-u.ac.jp

Abstract

Detection followed by projection in conventional privacy cameras is vulnerable to software attacks that threaten to expose image sensor data. By multiplexing the incoming light with a coded mask, a FlatCam camera removes the spatial correlation and captures visually protected images. However, FlatCam imaging suffers from poor reconstruction quality and pays no attention to the privacy of visual information. In this paper, we propose a deep learning-based compressive sensing approach to reconstruct and protect sensitive regions from secured FlatCam measurements. We predict sensitive regions via facial segmentation and separate them from the captured measurements. Our deep compressive sensing network was trained with simulated data, and was tested on both simulated and real FlatCam data.

1. Introduction

Cameras have become ubiquitous these days, from closed-circuit television (CCTV) cameras in public spaces and those in smartphones everywhere, to cameras in wearable devices like Google Glass and Virtual Reality/Augmented Reality headsets. It has thus become remarkably easy to capture and distribute high-quality pictures/videos without the knowledge or permission of others [10, 20]. We sacrifice our privacy in exchange for security and utilities for safety. For surveillance application, personal identity is required only in the case of accidents or crimes [24]. It is possible to perform computer vision tasks without knowing the object identity. Therefore, there are increasing concern about visual privacy, which is driven by the conventional approaches [12, 23, 29] of (i) capturing high-quality images, (ii) detecting thend (iii) protecting sensitive visual information (by blurring, blanking, and scrambling). However, the conventional method suffers from the following practical problems.

C1. Low-power encryption. Simultaneously sampling, detecting, and protecting data consumes significant resources, whereas low-power protection is desirable for applications of the Internet of Things (IoT).

C2. Camera-side protection. Outsourcing computation to cloud severs has become a popular approach for low-complexity end-user applications. Cameras pre-process visual signals with low-complexity operations and transmit the related information to the cloud server for computationally intensive tasks. Thus, protection at the camera is preferred as it is relatively safe from transmission attacks [29].

C3. Sensor-level protection. Verifying all software components to avoid software attacks [12, 29] is cumbersome. The capture first and then protect scheme is sensitive to software attacks because attackers can still obtain high quality visual data before privacy protection is implemented.

C4. Comprehensive protection. The entire visual privacy system can collapse owing to a single misdetection that reveals sensitive data and object identity [29]. Thus, powerful deep learning based detection technique should be used.

C5. Multi-class privacy. Advanced encryption methods can protect visually sensitive regions but also disrupts the viewing of the original content as different users may have different access rights [5, 12, 29]. Authorized users have the right to access the original content in case of emergency.

Compressive sensing (CS) [4, 7–9, 11] can help capture signals in a secure format at low complexity, and can partially solve C1, C2, and C3. A considerable amount of research has sought to study security-based CS [5, 32] but mostly in a simulated setting that is challenging to implement. On the contrary, some hardware is available for secure cameras, such as the single-pixel camera [11] and lenless imaging [2, 13, 18, 26] (e.g., coded aperture, FlatCam, and diffuser.) by multiplex the incoming light to destroy the spatial information. FlatCam can capture single-shot secure images but suffers from poor reconstruction quality.

This paper proposes an efficient sensor-level privacy protection method that uses FlatCam imaging and a deep convolutional neural network. The image is simultaneously captured and secured at the sensor level with a FlatCam camera, thus enabling (C1) low-cost (C2, C3) sensor-level protection. We introduce a deep privacy framework to recover/detect and protect the visually sensitive region of image, thereby, achieving (C4) comprehensive protection as well as multi-class protection (C5).

2. Related Work

2.1. Software Visual Privacy Protection

TrustEYE [28] is a secure sensing unit that handles the detection and encryption of sensitive information. Protected visual data are delivered from the camera's host system to other computer vision applications, the operating system, and software frameworks. However, the image details are still captured and thus sensitive to software attacks.

With a random projection, CS measurements are equally important. CS thus ensures computational security but pays no attention to visual privacy. Cambareri et al. [5] proposed multi-level encryption based on CS to protect sensitive text information. Sensitive regions are detected from a recovered high-resolution [5] or low-resolution image [6], and thus are still vulnerable to software attacks. Researchers have also applied computer vision directly from CS measurements with or without the reconstructed image signals [15, 16, 27]. Beside, learning from CS is possible with low complexity without requiring complex reconstruction [4] .

2.2. Hardware Privacy Protection

To provide stronger level of protection, Pittaluga et al. [22] extracted features for computer vision applications directly from the thermal image and defocused images [21]. Because it does not capture an RGB image, their solution prevents attackers from stealing the original content, and is resilient to software attacks, but fails to deliver the original content in case of emergency.

Another approach to protecting visual data is lensless imaging, such as single-pixel coded imaging [2, 11]. FlatCam [2] is a simple approach that involves placing a printed mask very close to the image sensor plane. Each pixel gathers multiplexed light with a unique pattern to destroy the spatial information in conventional imaging. FlatCam measurements are used to protect visual content, such as in face detection/recognition in [25] that requires reconstruction, and action recognition without reconstruction as proposed in [26]. However, these studies are not concerned with protecting sensitive visual information. Moreover, surveillance applications also demand the availability of the original content at high quality, which they cannot provide.

2.3. Deep Compressive Sensing

Recently, advanced deep learning method has been applied to compressive sensing (DCS) to provide non-iterative and fast reconstruction as well as learned sampling matrix. The DCS reduces complexity via convolution [17, 31], or separable sampling with Kronecker layers [7] in the single-scale sampling. DCS was extended to multi-scale scheme in [8, 9] utilizing image decomposition. This work focused on deep learning for FlatCam imaging and followed single-scale, separable compressive sensing.

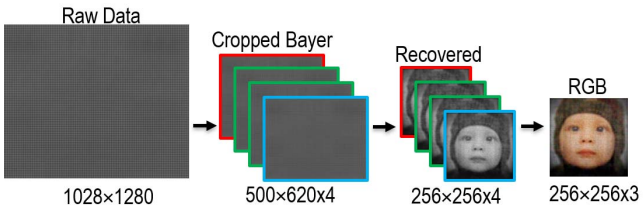

Figure 1. FlatCam processing pipeline [25].

3. FlatCam Imaging

FlatCam [2] places a coded mask very close to the image sensor for a thin camera design. Each pixel is a combination of multiple light rays similar to that in the frame-based CS, and this is computationally expensive. Sampling matrices are required to perform the reconstruction. Therefore, given a coded mask and FlatCam settings (sensor, distance, etc.), the authors [2] estimated the captured measurement by a separable sampling scheme via calibration. A color image $X = [X_R, X_G, X_B]$ with four Bayer measurements $Y = [Y_R, Y_{Gr}, Y_{Gb}, Y_B]$ is modeled as

$$\begin{aligned} Y_R &= P_R \cdot X_R \cdot Q_R^T + \mathcal{N}_R, \\ Y_{Gr} &= P_{Gr} \cdot X_G \cdot Q_{Gr}^T + \mathcal{N}_{Gr}, \\ Y_{Gb} &= P_{Gb} \cdot X_G \cdot Q_{Gb}^T + \mathcal{N}_{Gb}, \\ Y_B &= P_B \cdot X_B \cdot Q_B^T + \mathcal{N}_B, \end{aligned} \tag{1}$$

where $\mathcal{N}_i$ is Gaussian noise and P_i/Q_i are the vertical/horizontal sampling matrices for each channel. As in Fig. 1, FlatCam uses a color Bayer sensor at 1024×1280 equivalent to four channels of size 512×640. Bayer data are normalized, rotated, and cropped to 500×620 to reduce the boundary effect. Each channel is independently reconstructed at a resolution of 256×256 to improve light efficiency. $P_i \in \mathbb{R}^{256\times500}$ and $Q_i \in \mathbb{R}^{256\times620}$ are estimated through calibration. A color channel is recovered by

$$\arg\min_{X_i} \left\| P_i \cdot X_i \cdot Q_i^T - Y_i \right\|_2^2 + \lambda \mathcal{R}(X_i), \tag{2}$$

where X_i, P_i, Q_i, and Y_i denote the parameters for each Bayer channel, $i \in \{R, Gr, Gb, B\}$. $\mathcal{R}(\cdot)$ represents a regularization term such as the least squares $\|X\|_2^2$. Finally, four Bayer channels are converted into RGB.

FlatCam imaging suffers from poor reconstruction quality. *First*, the closer the distance is between the image sensor and the printed mask, the more the number of light rays multiplexed, and thus the more challenge recovery is. *Second*, there is limited light rays near the image boundary that leads to the vignetting effect. *Third*, determining the measurement matrices P_i, Q_i highly accurately is challenging, and a high correlation between channel measurements (as Table 1) suggests joint channel reconstruction.

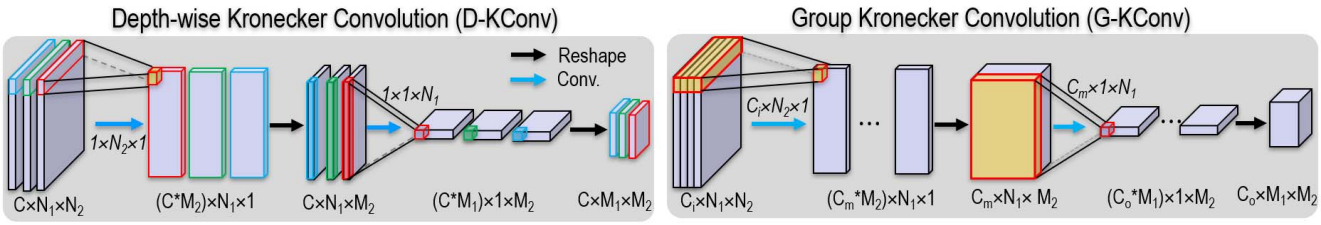

Figure 2. Depth-wise (D-KConv) and Group Kronecker convolution (G-KConv). G-KConv utilizes information across channels with more flexible parameters (C_m and C_o represent the numbers of intermediate and output features, respectively) than D-KConv.

Table 1. Average correlation between color channels of real (upper-right) and simulated (lower-left) data

S \ R	Y_R	Y_{Gr}	Y_{Gb}	Y_B
Y_R		0.857	0.860	0.702
Y_{Gr}	0.835		0.893	0.814
Y_{Gb}	0.834	0.849		0.7907
Y_B	0.801	0.805	0.805	

4. FlatCam Restoration with Deep Learning

It is straightforward to improve the quality of reconstruction of FlatCam with deep image enhancement given pairs of the ground truth and the simulated initial reconstruction. This naive approach can be used to recover image details, but linear processing in the initial reconstruction (i.e., normalization and rotation) introduces and/or propagates distortions. The literature has shown that directly recovering images from sensor measurements is better than multiple image processing pipelines [30]. We, therefore, propose reconstructing FlatCam with deep learning by modeling the initial and the reconstruction networks.

4.1. Sampling

Even though we can learn the capturing matrices by modeling separable sampling [7] as a Kronecker convolution (called KConv) [7, 33], FlatCam imaging uses calibration to deliver fixed sampling matrices with a given coded pattern and camera settings. It is difficult to match the learned separable sampling matrix with the coded pattern because of the nonlinearity of the calibration process. Therefore, to simulate FlatCam imaging, we follow Eq. (1) and add Gaussian noise to achieve a signal-to-noise ratio (SNR) of 10 (see Table 1). We do not perform the calibration and reuse the calibrated matrices from [25].

4.2. Multi-Phase Reconstruction

Similar to previous work [7–9, 17], we build network to mimic the process of FlatCam reconstruction. However, our previous work [7] proposed only for gray scale imaging. We propose a multiple-phase reconstruction for multi-channel images with (i) initial reconstruction (Phase 1–no bias, activation) with depth-wise and group Kronecker reconstruction; (ii) enhancement of the quality of reconstruction (Phase 2–complex layers) with multi-level wavelet convolution (MWCNN) [14].

4.2.1 Depth-wise Kronecker Initial Reconstruction

To obtain the initial reconstruction, it is straight forward to implement simple matrix inversion at for each channel as

$$\hat{X}_i = W_P^i \cdot Y_i \cdot W_Q^i, \quad i \in \{R, Gr, Gb, B\}. \tag{3}$$

which is modeled as a KConv layer with learned convolution W_P^i and W_Q^i. We apply KConv independently for both horizontal and vertical convolutions; thus the name Depth-wise KConv (D-KConv). It is shown in Fig. 2. Each of the three KConv has 256 convolutions of size $1 \times 500 \times 1$ and 256 convolutions of size $1 \times 1 \times 620$ for W_P^i and W_Q^i, followed by a reshape function.

4.2.2 Group Kronecker Initial Reconstruction

As there was a high correlation between the simulated and sensor data measurements (see Table 1), it is better to use information across multiple channels. We therefore propose using group Kronecker convolution (G-KConv). As in Fig. 2, G-KConv performs horizontal and vertical convolutions on all channel measurements and jointly recovers all channels. By doing so, G-KConv requires four times the number of convolution parameters as D-KConv. For instance, G-KConv requires $4 \times (4 \times 500 \times 1)$ horizontal and $4 \times (4 \times 1 \times 620)$ vertical convolution parameters, compared with the $4 \times (1 \times 500 \times 1)$ horizontal and $4 \times (1 \times 1 \times 620)$ vertical convolutions required by D-KConv.

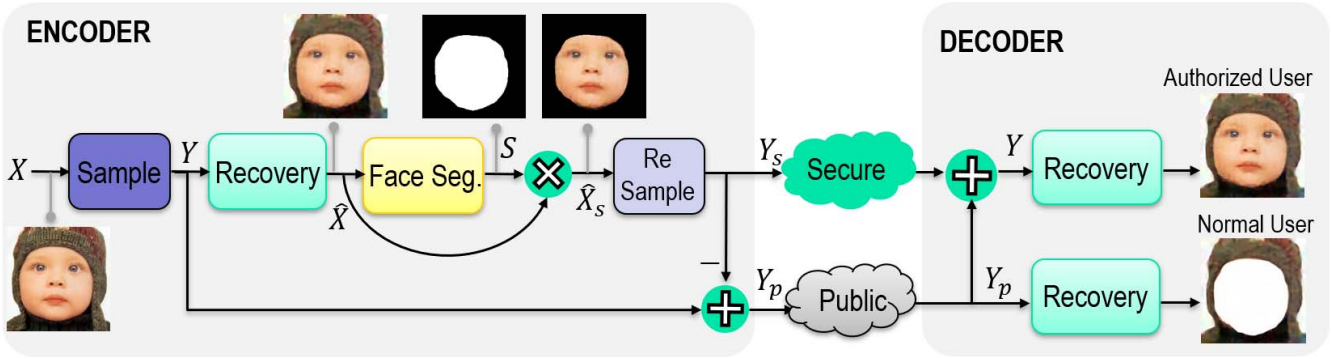

Figure 3. Our deep visual privacy protection network. The facial region in reconstructed image of normal user is protected (it was converted to white for better visualization).

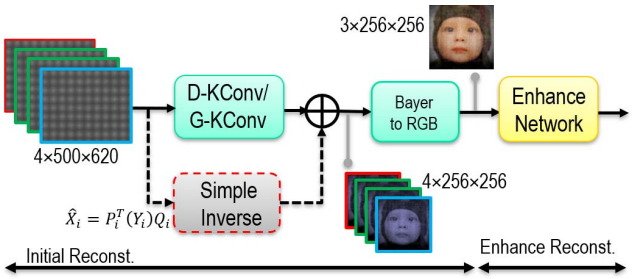

Figure 4. Our FlatCam reconstruction network with and without (including dashed-line block) residual learning.

4.2.3 Enhancing Reconstruction

Because of the difficulty of recovering high-quality images through few convolution layers in Phase 1, we enhance the quality of reconstruction by taking advantage of multi-level wavelet reconstruction (MWCNN) [14] in Phase 2. MWCNN replaces the up-/down-sampling layer in the U-Net architecture by wavelet decomposition and achieves high performance on image restoration tasks.

4.3. Proposed Network Architecture

Even though we can recover images with learned convolutions, we should utilize information in the calibrated sampling matrices P_i, Q_i. We thus propose a residual learning scheme that learns the residual output to compensate for simple reconstruction by the matrix inversion of

$$\hat{X}_i = W_P^i \cdot Y_i \cdot W_Q^i + P_i^T \cdot Y_i \times Q_i, \tag{4}$$

where W_P^i, W_Q^i are learnable parameters for the corresponding channel $i \in \{R, Gr, Gb, B\}$. The generalized framework is shown in Fig. 4 for both normal and residual learnings. We used the L2 norm as the image restoration loss [9, 14] as

$$\min \frac{1}{N} \sum_{i=1}^{N} ||\mathcal{F}(Y, \theta) - X||_2^2, \tag{5}$$

where N denotes the total number of samples and $\mathcal{F}$ is network function with input Y and parameter θ.

We use DIV2K dataset [1] to generate $32 \times 32,000$ color images of size 256×256 for training. The batch size was set to 16. The learning rate was set to 5×10^{-4} and gradually reduced by half every 50 epochs for a total of 200 epochs. Adam was used as optimizer.

5. Deep Visual Privacy Protection

Even though the FlatCam measurements revealed no spatial information of the image, they did not attend to the privacy of visual data. This section introduces a deep neural network to predict and protect sensitive information among FlatCam measurements. We define the facial region from the chin to the forehead as sensitive information

We define the problem of detecting sensitive regions as a semantic segmentation problem with a binary label S (0 for background and 1 for the facial region). A straightforward approach is to use reconstructed image by employing existed segmentation frameworks. We can protect sensitive regions by random scrambling, blurring, whitening, and encryption. However, standard encryption is difficult to apply to models as a deep layer, and blurring is not secure owing to advanced deblurring techniques. In this work, we define the reconstructed image $\widehat{X}$ as a combination of a secure $\widehat{X}_s$ and a protected image $\widehat{X}_p$ as

$$\widehat{X} = \widehat{X}_p + \widehat{X}_s, \quad s.t. \quad \widehat{X}_s = S \odot \widehat{X}, \tag{6}$$

where $\odot$ represents pixel-wise multiplication. Sensitive images are resampled using calibrated sampling matrices as

$$\begin{aligned} Y_s &= P \cdot \widehat{X}_s \cdot Q^T, \\ \Rightarrow Y_p &= P \cdot (\widehat{X} - \widehat{X}_s) \cdot Q^T = Y - Y_s. \end{aligned} \tag{7}$$

Table 2. Reconstruction performance (PSNR[dB]/SSIM) of various algorithms

Test Set	Flat Cam [25]	Phase 1 - Initial Reconstruction				Phase 2 - Enhance Reconstruction			
		Standard		Residual		Standard		Residual	
		D-KConv	G-KConv	D-KConv	G-KConv	D-KConv	G-KConv	D-KConv	G-KConv
Set5	14.43	21.83	22.53	21.83	22.63	24.01	24.09	24.17	24.04
	0.654	0.624	0.699	0.624	0.712	0.778	0.786	0.786	0.785
Set14	13.89	21.37	21.95	21.36	22.05	23.44	23.46	23.55	23.41
	0.555	0.624	0.683	0.621	0.691	0.752	0.758	0.759	0.757
Kodak	14.39	21.98	22.69	21.97	22.81	24.11	24.20	24.28	24.17
	0.567	0.623	0.698	0.622	0.710	0.771	0.779	0.779	0.777

Table 3. Prediction accuracy of Face Segmentation [19] in spatial domain. Org - labels are predicted from clean images

Quality Index	Org	Flat Cam [25]	Phase 1 - Initial Reconstruction				Phase 2 - Enhance Reconstruction			
			Standard		Residual		Standard		Residual	
			D-KConv	G-KConv	D-KConv	G-KConv	D-KConv	G-KConv	D-KConv	G-KConv
FPA	0.9340	0.6429	0.4090	0.9417	0.4082	0.9427	0.9386	0.9631	0.9608	0.9646
IoU	0.8765	0.5798	0.3854	0.8521	0.3847	0.8527	0.8483	0.8678	0.8678	0.8766

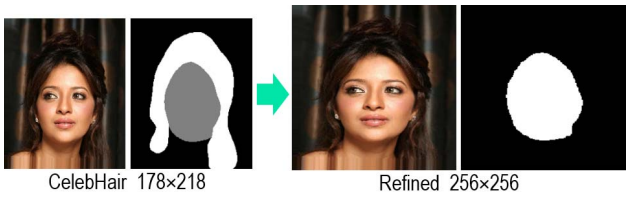

Figure 5. Refining the FaceHair image dataset [3] by removing the hair class and resizing it to 256×256.

5.1. Proposed Privacy Protection Network

As shown in Fig. 3, we propose a unified deep network with multiple sub-networks. First, image X is captured by a FlatCam sensor in the secured form of Y. Second, reconstruction is performed $\widehat{X}$ and the sensitive region S is predicted via a facial segmentation network. The sensitive image $\widehat{X}_s$ is extracted by element-wise multiplication and resampled to deliver the visually sensitive measurement Y_p. Third, we subtract the captured measurements Y from Y_s to obtain the protected measurements Y_p.

Measurements Y_s and Y_p are sent through a private and a public channel, respectively. Depending on user type, they receive corresponding FlatCam data. Normal users receive protected data from the public channel that can be recovered the protected image (i.e., the sensitive region is removed). The authorized user combines the visually sensitive and the protected measurements to reconstruct the original content. Both users can use the same reconstruction. Note that the availability of calibrated sampling matrices in the residual scheme is vulnerable to software attacks.

At the decoder side, attackers can access the public channel for the protected images. They can gain access to the private channel, but without knowing the sampling matrix they have to solve blind image restoration which is an NP-hard problem. At the encoder side, we assume that the attacker can perform plain attacks (i.e., manipulate the input/output of the encoder) but cannot extract the intermediate deep features. This is a reasonable assumption as the deep learning framework calculates layer by layer in a single chip and discards intermediate features for memory efficiency. As long as the input and output are secured, so is the deep encoder network. Thus, our framework is resilient to transmission and attacks from the encoder side.

6. Experimental Results

6.1. Reconstruction Performance

We trained our networks D-KConv-P1 and G-KConv-P1 in the initial reconstruction (Phase 1), and for enhanced reconstructions D-KConv-P2 and G-KConv-P2 (Phase 2) under the standard and residual schemes, as shown in Fig. 4. We removed activation and bias in Phase 1. Test images Set5, Set14, and Kodak were cropped to a square shape and resized to 256×256 to simulate FlatCam data. Moreover, noise was added to simulate FlatCam data at an SNR of 10. For real FlatCam data, we used face measurements in [25].

Table 2 shows the superior performance of G-KConv-P1 to D-KConv-P1 on both the standard and the residual networks, with approximate gains of 0.8–1 dB, 0.4–0.6 dB, and 0.7–0.8 dB for Set5, Set14, and Kodak, respectively. The residual scheme, however, slightly favored joint channel reconstruction with G-KConv-P1, with a gain of 0.1 dB

Figure 6. Quality of visual reconstruction for simulated Baby FlatCam data. First row - ground truth and reconstructed images from standard network. Second row - conventional FlatCam [25] and reconstructed images from residual network

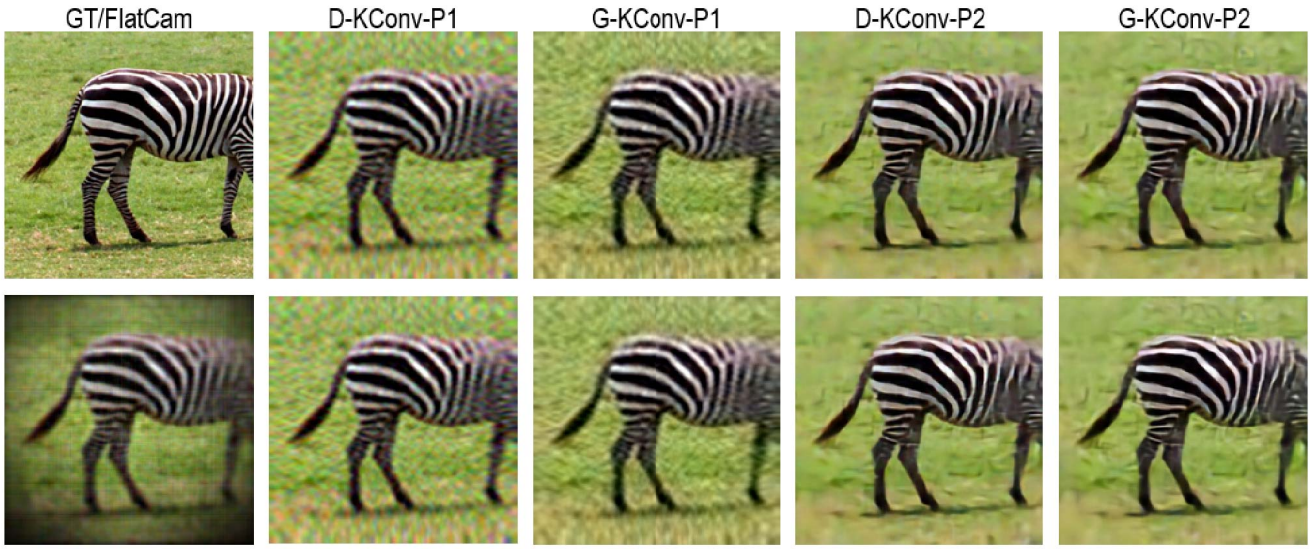

Figure 7. Quality of visual reconstruction for simulated Zebra FlatCam data. First row - ground truth and reconstructed images from standard network. Second row - conventional FlatCam [25] and reconstructed images from residual network

over D-KConv-P1. As in Fig. 6, 7, there was a strong color artifact in both the conventional and the residual schemes of D-KConv-P1. On the contrary, from the perspective of visual quality, both learned reconstruction with D-KConv-P1 and G-KConv-P1 to eliminate the vignetting effect (i.e., pixels darkened at the boundary pixels), and thus significantly gained in PSNR (with a 7-dB gain on average over the conventional FlatCam [25]). They smoothed the corners as a way of compensation.

In Phase 2, the MWCNN yielded a gain of 2.20 dB and generated better visual quality than in Phase 1. G-KConv performed similarly to D-KConv in terms of PSNR but was better on the SSIM index. Again, residual reconstruction outperformed the conventional scheme with a gain of 0.11 0.18 dB in D-KConv-P2 but a slight loss in G-KConv-P2. As shown in Fig. 7, the residual scheme tended to preserve edges better than the conventional one. The poor SNR led to an oversmoothed image and generated fake edges.

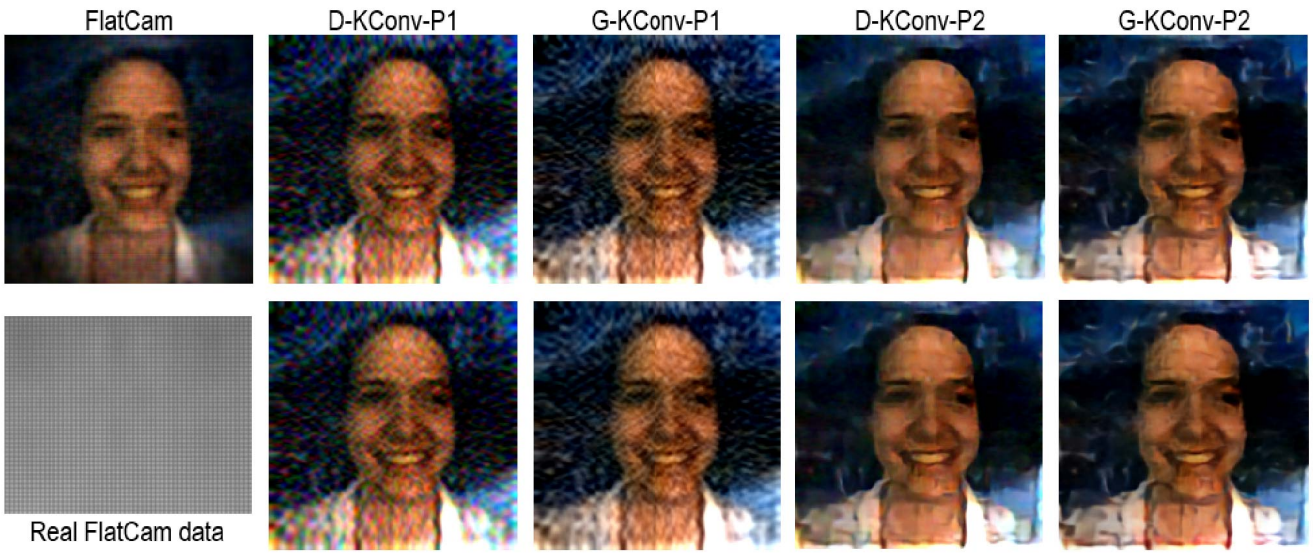

Figure 8. Quality of visual reconstruction quality of real FlatCam data (Girl 75-13). First row - conventional FlatCam [25] and reconstructed image from standard network. Second row - real FlatCam data and reconstructed image from residual network.

There was a mismatch between the simulated and real FlatCam data. First, despite the calibration process, estimating high-quality sampling matrices was still challenging. Second, additional noise (i.e., quantization noise and sensor noise) was present. The FlatCam data [25] were captured when the frontal face images were weakly lit while the simulated training data were at a high resolution. Therefore, the proposed methods improved the visual quality of the images but also introduce fake edges. The test images were taken indoor, and featured smooth objects. Therefore, they did not highlight the edge-preserving quality of deep learning. As a result, instead of G-KConv-P2, the D-KConv-P2 produced the most visually pleasing images as shown in Fig. 8.

6.2. Visual Privacy Protection

To evaluate the quality of the proposed method at protection of visual images, we example the accuracy of sensitive region prediction instead. We used two popular quality indices—Facial Pixel Accuracy (FPA) and Intersection over Union (IoU). The FPA index is defined as

$$FPA = \frac{\sum_{i=0}^{N} n_{f,f}^{i}}{\sum_{i=0}^{N} n_{f}^{i}}, \tag{8}$$

where $n_{f,f}^{i}$ denotes the number of face pixels which is classified correctly and n_{f}^{i} represents the total number of pixels in class face of a given image index i th. The second quality index of the Intersection over Union (IoU) is represented by

$$IoU = \frac{\sum_{i=0}^{N} n_{f,f}^{i}}{\sum_{i=0}^{N} (n_{f}^{i} + n_{b,f}^{i} - n_{f,f}^{i})}, \tag{9}$$

which defines the overlapped ratio between the ground truth and the predicted facial region.

Dataset. While semantic segmentation is an active subject of research, limited work has been published on facial segmentation owing to the limited number of datasets available. The closest dataset to the one used here was reported in [19]. However, it is not yet available to the public, and thus we used FaceSeg [19] without fine tuning. To evaluate quality, we used the CelebHair [3] dataset with some corrections (i.e., fixed or removed incorrectly classified labels of images, and removed the hair label). We also resized the images (by bicubic) and label masks (by nearest neighbor) to 256×256 to match with the resolution of FlatCam imaging as shown in Fig. 5. We selected 100 images from the [3] dataset to measure prediction accuracy.

The results of the segmentation of the FPA and IOU indices with the ground truth, conventional FlatCam reconstruction [2, 25], initial (D-KConv-P1, G-KConv-P2, and their residual), and enhanced reconstructions (D-KConv-P2, G-KConv-P2, and their residual) as input. We used FaceSeg [19] without further fine tuning. Table 3 shows that the conventional FlatCam reconstruction was poor in quality [25], and heavily degraded the accuracy of facial segmentation with an approximate 30% reduction in the FPA.

As suffering color artifacts that impacted the facial skin quality, D-KConv yielded poor FPA and IoU. Not fine-tuning the FaceSeg was another reason for poor segmen-

Figure 9. Results of facial segmentation with FPA/IoU indexes. First row — (left to right) original RGB image, protected image detected by [19] with the original RGB and reconstructed images from standard networks. Second row — (left to right) protected image with ground truth face label and protected image detected by [19] from reconstructed image of FlatCam [25] and residual networks.

tation. Interestingly, even with the initial G-KConv-P1, we obtained slightly better FPA and a 2.5% IoU reduction compared to the original. As G-KConv-P1 preserved edges well and FaceSeg was resilient against noise, both D-KConv-P2 and G-KConv-P2 had slightly better values of the FPA than the predictions of the original image. This might have been obtained because FaceSeg upscaled images multiple times (from 178×218 to 256×256, and then to 500×500) before segmentation. Owing to deep learning, D-KConv-P2 and G-KConv-P2 maintained sharp edges and led to a higher FPA index. The results from residual networks exhibited better values of the IoU than the standard scheme, with the best performance delivered by G-KConv-P2.

The results of the visualization segmentation in Fig. 9 show similar conclusions to the above. FaceSeg could not predict face regions using the D-KConv-P1 and FlatCam reconstructions very accurately. The loss of high frequency in the original images also led to poor prediction in the face boundary. With fewer color artifacts, G-KConv-P1 predicted face labels more accurately. Visually, we observed that the residual learning scheme returned better segmented results than the standard learning scheme as in Fig. 9.

While it is possible to recognize identities from FlatCam and D-KConv-P1 prediction results, it is nearly impossible form other schemes. While G-KConv-P1 cover only 80% of the faces, it still reveal no identity of the person. The visualization in Fig. 9 also reveals that both FPA and IoU are effective quality index reflects the order of visual protection quality.

7. Conclusion

This paper proposed a deep learning framework to protect visual data using secured FlatCam lensless measurements. To reconstruct the FlatCam measurements, we proposed a multi-phase deep network based on depth-wise or group Kronecker convolutions with or without residual connections. To protect sensitive information, we used a semantic segmentation network to detect and protect face regions in the reconstructed image. Our network protects visual data and supports the recovery of the original content, which is critical for surveillance applications.

8. Acknowledgment

This work was supported by JSPS KAKENHI Grant Number JP18K19818.

References

[1] E. Agustsson and R. Timofte. Ntire 2017 challenge on single image super-resolution: Dataset and study. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, July 2017. 4

[2] M. S. Asif, A. Ayremlou, A. Sankaranarayanan, A. Veeraraghavan, and R. G. Baraniuk. Flatcam: Thin, lensless cameras using coded aperture and computation. *IEEE Transaction on Computational Imaging (TCI)*, 3(3):384–397, September 2017. 1, 2, 7

[3] D. Borza, T. Ileni, and A. Darabant. A deep learning approach to hair segmentation and color extraction from facial images. In *Springer International Conference on Advanced Concepts for Intelligent Vision Systems: Advanced*

Concepts for Intelligent Vision Systems, pages 438–449. Springer, 2018. 5, 7

[4] R. Calderbank and et al. Compressed learning: Universal sparse dimensionality reduction and learning in the measurement domain. Technical report, 2009. 1, 2

[5] V. Cambareri, M. Mangia, F. Pareschi, R. Rovatti, and G. Setti. Low-complexity multiclass encryption by compressed sensing. *IEEE Transaction on Signal Processing (TSP)*, 63(9):2183–2195, May 2015. 1, 2

[6] T. N. Canh and B. Jeon. Privacy-preserving compressive sensing for still images. In *IEIE International Technical Conference on Circuit/System, Computer, and Communication (ICT-CSCC)*, pages 1–4, December 2017. 2

[7] T. N. Canh and B. Jeon. Deep learning-based kronecker compressive imaging. In *IEEE International Conference on Consumer Electronic - Asia (ICCE-A)*, pages 1–4, December 2018. 1, 2, 3

[8] T. N. Canh and B. Jeon. Multi-scale deep compressive sensing network. In *IEEE International Conference on Visual Communication and Image Processing (VCIP)*, pages 1–4, December 2018. 1, 2, 3

[9] T. N. Canh and B. Jeon. Difference of convolution for deep compressive sensing. In *IEEE International Conference on Image Processing (ICIP)*, September 2019. 1, 2, 3, 4

[10] A. Cavailaro. Privacy in video surveillance [in the spotlight]. 24(2):168–166, 2017. 1

[11] M. F. Duarte, M. A. Davenport, D. Takhar, J. N. Laska, T. Sun, K. F. Kelly, and R. G. Baraniuk. Single-pixel imaging via compressive sampling. *IEEE Signal Processing Magazine (SPM)*, 25(2):83–91, March 2008. 1, 2

[12] A. Erdélyi and et al. Privacy protection vs. utility in visual data. *Multimedia Tools Application*, 77(2):2285–2312, January 2018. 1

[13] Y. Inagaki, Y. Kobayashi, K. Takahashi, T. Fujii, and H. Nagahara. Learning to capture light fields through a coded aperture camera. In *The European Conference on Computer Vision (ECCV)*, September 2018. 1

[14] P. Liu, H. Zhang, K. Zhang, L. Lin, and W. Zuo. Multi-level wavelet-cnn for image restoration. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2018. 3, 4

[15] S. Lohit and et al. Reconstruction-free inference on compressive measurements. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshop*, pages 16–24, June 2015. 2

[16] S. Lohit and et al. Direct inference on compressive measurements using convolutional neural networks. In *IEEE International Conference on Image Processing (ICIP)*, pages 1913–1917, November 2016. 2

[17] S. Lohit and et al. Convolutional neural networks for noniterative reconstruction of compressively sensed images. *IEEE Transaction on Computational Imaging (TCI)*, 4(3):326–340, September 2018. 2, 3

[18] H. Nagahara and Y. Yagi. Lensless imaging for wide field of view. *Optical Engineering*, 54(2):1 – 8 – 8, 2015. 1

[19] . Nirkin, I. Masi, A. Tran Tuan, T. Hassner, and G. Medioni. On face segmentation, face swapping, and face perception. In *IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018)*, pages 98–105, May 2018. 5, 7, 8

[20] J. R. Padilla-Lopez, , A. A. Chaaraoui, and F. Florez-Revuelta. Visual privacy protection method: A survey. 42(9):4177–4195, 2015. 1

[21] F. Pittaluga and S. J. Koppal. Privacy preserving optics for miniature vision sensors. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 314–324, June 2015. 2

[22] F. Pittaluga, A. Zivkovic, and S. J. Koppal. Sensor-level privacy for thermal cameras. In *IEEE International Conference on Computational Photography (ICCP)*, pages 1–12, May 2016. 2

[23] J. Shu, R. Zheng, and P. Hui. Cardea: Context-aware visual privacy protection for photo taking and sharing. In *Proceeding of Multimedia Systems Conference*, pages 304–315. ACM. 1

[24] A. Srivastava, P. Jain, S. Demetriou, L. P. Cox, and K.-H. Kim. CamForensics: Understanding visual privacy leaks in the wild. In *Proceedings of the 15th ACM Conference on Embedded Network Sensor Systems - SenSys '17*, pages 1–13. ACM Press. 1

[25] J. Tan, L. Niu, J. K. Adams, V. Boominathan, J. T. Robinson, and R. G. Baraniuk. Face detection and verification using lensless cameras. *IEEE Transaction on Computational Imaging (TCI)*, 5(2):180–194, June 2019. 2, 3, 5, 6, 7, 8

[26] Z. W. Wang, V. Vineet, F. Pittaluga, S. N. Sinha, O. Cossairt, and S. Bing Kang. Privacy-preserving action recognition using coded aperture videos. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2019. 1, 2

[27] T. Wimalajeewa, H. Chen, and P. K. Varshney. Performance limits of compressive sensing-based signal classification. *IEEE Transaction on Signal Processing*, 60(6):2758–2770, June 2012. 2

[28] T. Winkler, A. Erdélyi, and B. Rinner. Trusteye.m4: Protecting the sensor — not the camera. In *IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, pages 159–164, August 2014. 2

[29] T. Winkler and B. Rinner. Privacy and security in video surveillance. In *Intelligent Multimedia Surveillance: Current Trends and Research*, pages 37–66. Springer. 1

[30] X. Xu, Y. Ma, and W. Sun. Towards real scene super-resolution with raw images. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 3

[31] H. Yao, F. Dai, S. Zhang, Y. Zhang, Q. Tian, and C. Xu. Dr2-net: Deep residual reconstruction network for image compressive sensing. *Neurocomputing*, 2019. 2

[32] P. Zhang and et al. A secure data collection scheme based on compressive sensing in wireless sensor networks. *Ad Hoc Networks*, 70:73 – 84, 2018. 1

[33] S. Zhou, J.-N. Wu, Y. Wu, and X. Zhou. Exploiting local structures with the kronecker layer in convolutional networks. *CoRR*, abs/1512.09194, February 2015. 3