Supplementary Material: Learning Privacy-preserving Optics for Human Pose Estimation

Carlos Hinojosa[1], Juan Carlos Niebles[2], Henry Arguello[1]
[1]Universidad Industrial de Santander [2]Stanford University
https://carloshinojosa.me/project/privacy-hpe/

1. Sketch of the implemented optical system

We show the optics diagram of our implemented proof-of-concept optical system in Fig. 1. Our prototype camera has a main objective lens coupled with a $4f$ system with a phase modulating element at $2f$. The intermediate image plane is formed by an 8mm objective lens (L1), which is relayed by a pair of 75mm Fourier transforming lenses (L2 and L3). L1 corresponds to a NAVITAR MVL8M23 lens, and L2 and L3 are two Thorlabs AC254-075-A-ML lenses in our setup. Using a beamsplitter (BS, Thorlabs CCM1-BS013), we placed a deformable mirror (DM, Thorlabs DMP40-P01) at the pupil plane at a distance of $2f = 150$mm from the intermediate image plane. Finally, we place a CANON EOS REBEL T5i at a distance of $2f = 150$mm from the DM, corresponding to the optical setup's image plane.

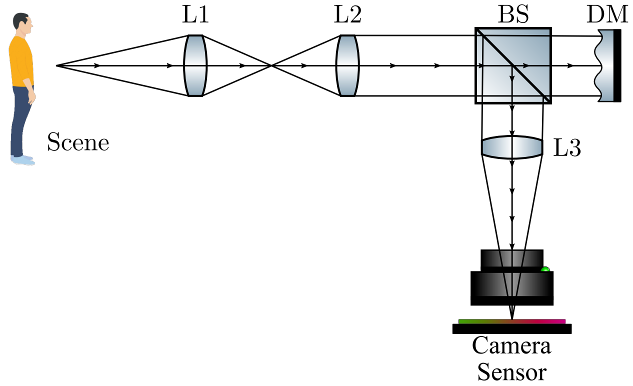

Figure 1. Sketch of the implemented proof-of-concept optical system. In our setup, we have three lenses (L1, L2, L3), one beam splitter (BS), and one deformable mirror (DM). Here, the DM allows us to use 15 aberrations via Zernike parameterization.

2. Face Recognition as Privacy Measure

In general, defining a specific metric to measure privacy is not an easy task. In this work, we measure privacy using face recognition. We use a Tensorflow implementation[1] of the Additive Angular Margin Loss for Deep Face Recognition (ArcFace) network[2]. ArcFace is a recently published, efficient, and highly effective face recognition network that incorporates margins in its loss function to obtain highly discriminative features for face recognition.

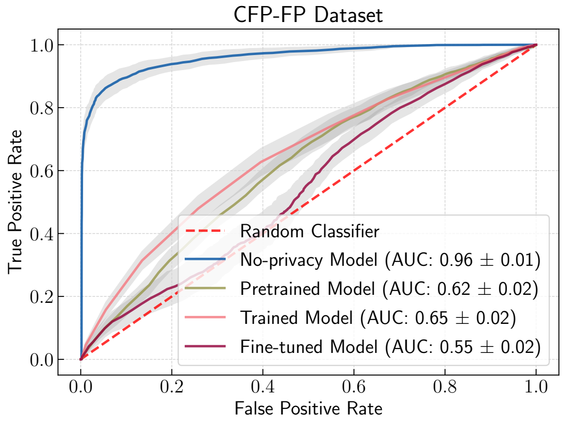

Figure 2. Face recognition performance on images from CFP-FP acquired with our optimized lens.

Face Recognition results on CFP-FP dataset. We use the ArcFace network to test the face recognition performance on images acquired with our optimized lens. We experimented on three datasets: LFW, AgeDB-30, and CFP-FP datasets. We generate ROC curves using three testing approaches for each dataset and compare them with the original ArcFace model tested on the original "non-private" images. We refer to the first approach as the "Pretrained model", which uses the pretrained ArcFace model to test the "private" version of each dataset. The second approach consists of training the ArcFace model from scratch using the private version of the MS-Celeb-1M dataset; we refer to such an approach as the "Trained model". Finally, in the "Finetuned model" approach, we first load the pretrained weights of the ArcFace model on original "non-private" images; then, we performed fine-tuning on the network with

[1] https://github.com/peteryuX/arcface-tf2

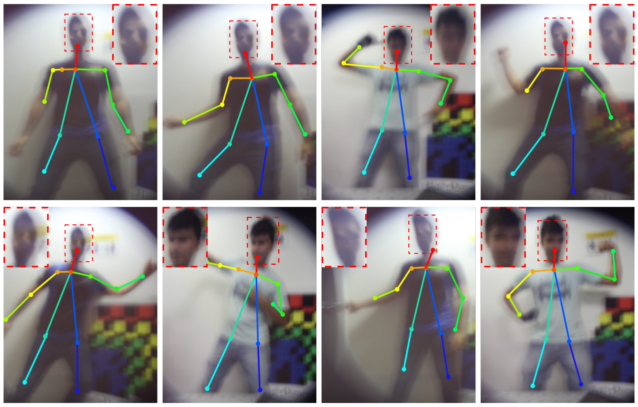

Figure 3. Qualitative results on some example images acquired by the prototype camera.

the private version of the MS-Celeb-1M dataset. We presented the results on the LFW and AgeDB-30 dataset in Fig. 4 of the main manuscript, and the results on the CFP-FP dataset are shown in Fig. 2. Similar to the main manuscript results, the ArcFace model performs poorly on the "private" images generated by our optimized lens.

Compute L_F in Eq. 18. In addition to the proposed privacy-preserving loss, we explore the two additional loss function approaches: L_{P_1}, and L_{P_2}, shown in Eq. 16 and Eq. 17 of the main manuscript. In particular, for L_{P_2} we need to compute L_F loss in Eq. 18 of the main manuscript, which measures face similarity/dissimilarity between original or input image $\mathbf{x}$ and "private" image $\mathbf{y}$. To extract the face regions from the images, we use the RetinaFace [3] detector with a ResNet50 backbone. We generate three files containing face labels for COCO 2017 dataset for the training, validation, and testing sets, respectively [2]. Each file contains lines with the COCO image filename and coordinates $(x_1, y_1), (x_2, y_2)$ specifying the upper-left and lower-right corners of the face rectangle. All coordinates are presented as floating-point numbers in the range [0, 1] relative to the specific image's width and height. Then, with these face region annotations, we crop faces from both $\mathbf{x}$ and $\mathbf{y}$ and extract the embeddings using the ArcFace model with pretrained weights loaded. We obtain the L_F by comparing both embedding vectors using the cosine similarity.

3. Additional Visual Results

Figure 5 present additional visual results when using our optimized lens and compare them with the results from the original OpenPose, which provides No-privacy pose estimation and works on images acquired with standard cameras. In the last two columns of the figure, we show example failure cases of our method, which fails to estimate distant people's pose as we reported in the main manuscript. However, when a person is far from the camera, less is the privacy concern; hence, our method's privacy protection is still useful in most cases. Also, Fig. 3 shows more qualitative results acquired by our prototype camera.

4. Lightweight OpenPose

In general, any human pose estimation (HPE) network can be adopted as the CNN-decoder of our proposed privacy-preserving approach. This section reports some results with the Lightweight OpenPose network (LOPPS) as the CNN-decoder. The authors of the LOPPS aims at decreasing the computational burden of the OpenPose[1] (OPPS) Network by proposing three main changes: to use a lighter backbone; using a single branch for PAF and keypoints predictions instead of the two branches of OPPS; and replacing expensive 7×7 convolutions with 3×3, 1×1, and 3×3 with dilation of 2 convolutions blocks [9]. The original paper of LOPPS replaces the VGG-19 backbone network of OPPS with MobileNet family networks [5]. MobileNets are built primarily from depthwise separable convolutions [10] to reduce the computation in the first few layers. However, in this work, we use the well-known ResNet-50 network [4] as a replacement for the VGG-19 backbone. As the LOPPS does not modify the OPPS loss function, we the same loss function proposed in the main manuscript.

Training Details. Similar to the procedure described in the main manuscript, we assume an aberration-free freeform lens and use the pretrained weights of a Tensorflow implementation of LOPPS [7] as a starting point. Once initializing the network with the previously learned weights, we freeze the single branch of LOPPS and fine-tune the first 68 trainable layers of the Resnet backbone to learn to extract human body features from the privacy image. We simulate a sensor with a pixel size of 3.40μm and a resolution of 864×864 pixels. We consider the first $q = 350$ Zernike coefficients in Noll notation to shape the surface profile ϕ. The fourth Zernike coefficient (the defocus term) is initialized, such that the lens has a focal length of $f = 25$mm. The optical element is discretized with a 3.40μm feature size on an 864×864 grid. We trained the end-to-end model using Adam optimizer with a batch size of 24 and an initial learning rate of 4×10^{-5}. We applied an exponential learning rate decay with a decay factor of 0.666 that is triggered after 15K, 20K, 25K, and 28K training steps. We trained the network for 50K steps (gradient updates), taking about 20 hours on an Nvidia TESLA V100-SXM2-32GB GPU.

Quantitative and Qualitative Results. Figure 6 shows a visual comparison of our proposed method using our optimized lens against the results from the original Lightweight OpenPose (No-privacy pose estimation), which works on images acquired with the standard lens. As shown, our proposed privacy-preserving approach using LOPPS achieves

[2] https://carloshinojosa.me/files/coco2017_RetinaFace_annotations.zip

Method	Fine-tuned Layers	PSNR	SSIM	AP	AP^{50}	AP^{75}	AP^{M}	AP^{L}	AR
LOPPS [8]	-	-	-	0.37	0.629	0.373	0.382	0.376	0.45
PP-LOPPS	68	16.34	0.631	0.237	0.487	0.259	0.266	0.228	0.302

Table 1. Comparisons on the COCO validation set. We compare our method against the lightweight OpenPose (LOPPS) network. The PP prefix stands for our proposed privacy-preserving approach.

less degradation and keypoint prediction accuracy than our proposed privacy-preserving approach using OPPS. However, the obtained low-quality images still provide privacy protection for people while achieves good human pose estimation. Besides, Table 1 reports the COCO keypoints evaluation results and the average of the PSNR and SSIM image quality metrics among all images from the COCO 2017 validation set. In the table, PP-LOPPS stands for our proposed privacy-preserving approach for LOPPS.

5. Blind Deconvolution using GANs

In this work, we assume that the attacker has no access to the camera hardware. The same assumption is made by hardware-level privacy protection approaches we found in the literature. Indeed, if the attacker has access to the camera, the PSF can be estimated by imaging a point light source. In Fig. 4 of the main manuscript, we explore the worst scenario when an attacker exactly knows the Zernike coefficients that lead to the PSF. In that case, a simple *non-blind* deconvolution method can reconstruct the visual details from the defocus approach while our method is more robust. Besides, our proposed approach achieves higher HPE performance despite having more blur than, for instance, a defocus lens. On the other hand, here we explore the case when an attacker has no access to the hardware but has a large set of blurred images with its respective non-blurred version. In such a case, the attacker can train a blind deconvolution network to try to recover the people's identities from the private images.

We trained a blind deconvolution network (*DeblurGAN*[3]) [6] with 16000 sharp and blur images (ours) from the COCO dataset during 500 epochs. Fig. 4 shows the results (third row) of recovering the people identities (first row) from ours blur images (second row) using the trained network. As observed, reconstruction is challenging. The network can reconstruct some objects (e.g., the fifth column's image); however, the face details seem to be missed, and the network cannot recover people's identities.

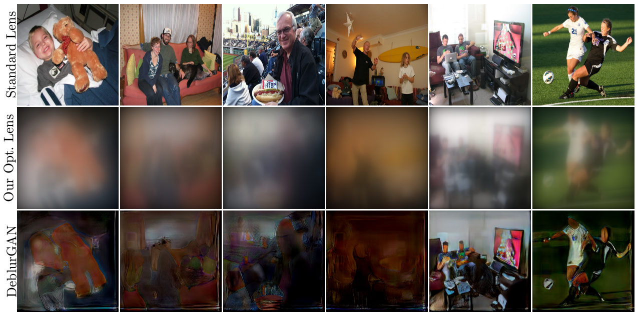

Figure 4. Blind deconvolution of private images acquired with our optimized lens, assuming the PSF is unknown but exists a large collection of acquired images with the proposed camera. The images acquired with our lens are robust to blind deconvolution, and the trained network cannot recover people's identities.

References

[1] Zhe Cao, Gines Hidalgo Martinez, Tomas Simon, Shih-En Wei, and Yaser A Sheikh. Openpose: Realtime multi-person 2d pose estimation using part affinity fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019. 2, 4

[2] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4690–4699, 2019. 1

[3] Jiankang Deng, Jia Guo, Yuxiang Zhou, Jinke Yu, Irene Kotsia, and Stefanos Zafeiriou. Retinaface: Single-stage dense face localisation in the wild. *arXiv preprint arXiv:1905.00641*, 2019. 2

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 2

[5] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017. 2

[6] Orest Kupyn, Tetiana Martyniuk, Junru Wu, and Zhangyang Wang. Deblurgan-v2: Deblurring (orders-of-magnitude) faster and better. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8878–8887, 2019. 3

[7] Jiawei Liu, Yixiao Guo, Guo Li, Luo Mai, and Dong Hao. Hyperpose: Real-time human pose estimation. https://github.com/tensorlayer/hyperpose, 2020. 2

[8] Daniil Osokin. Real-time 2d multi-person pose estimation on cpu: Lightweight openpose. *arXiv preprint arXiv:1811.12004*, 2018. 3

[9] Daniil Osokin. Real-time 2d multi-person pose estimation on CPU: lightweight openpose. In Maria De Marsico, Gabriella Sanniti di Baja, and Ana L. N. Fred, editors, *Proceedings of the 8th International Conference on Pattern Recognition Applications and Methods, ICPRAM 2019, Prague, Czech Republic, February 19-21, 2019*, pages 744–748. SciTePress, 2019. 2, 4

[10] Laurent Sifre and Stéphane Mallat. Rigid-motion scattering for image classification. *Ph. D. thesis*, 2014. 2

[3] https://github.com/VITA-Group/DeblurGANv2

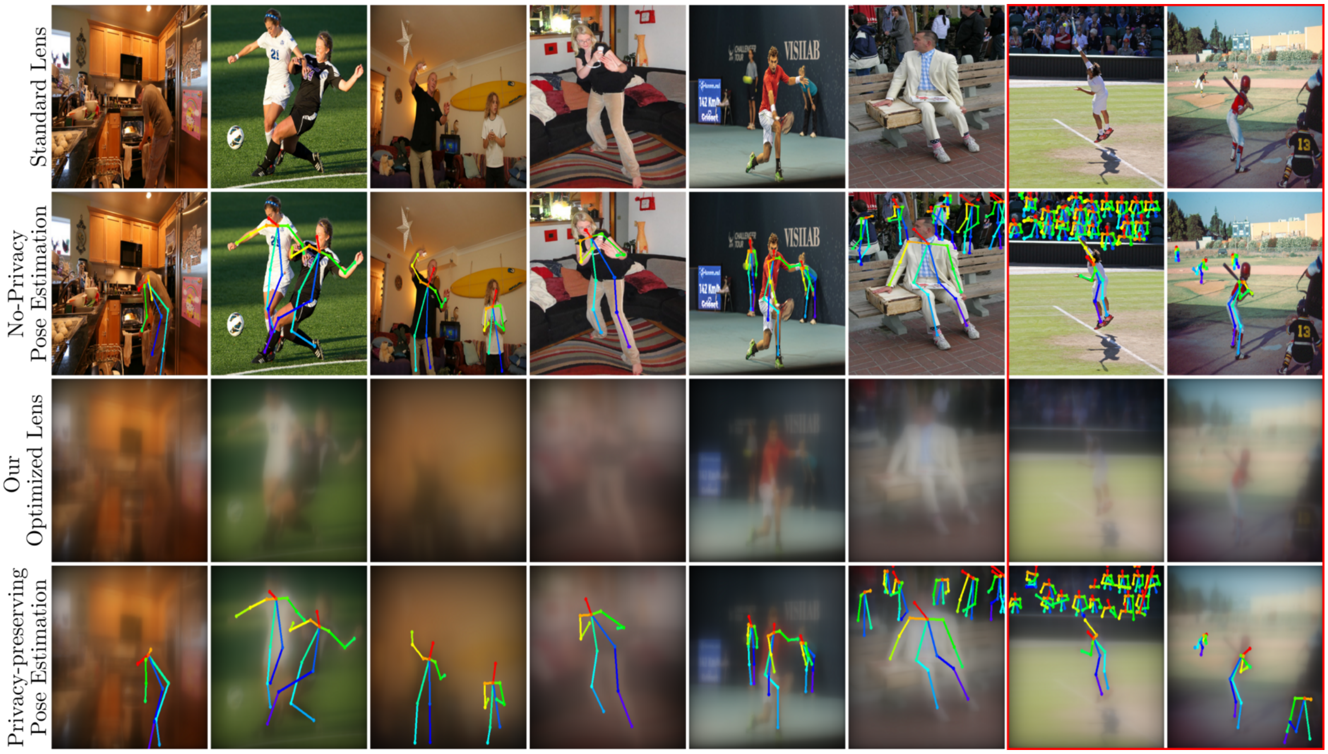
Standard Lens
No-Privacy Pose Estimation
Our Optimized Lens
Privacy-preserving Pose Estimation

Figure 5. Qualitative results on example COCO 2017 images using the OpenPose [1] (OPPS) network as the CNN-decoder. We compare our proposed privacy-preserving pose estimation results using our optimized lens with the Non-privacy approach using a standard lens. The last two columns depict failure cases where we fail to estimate the pose of far distant people.

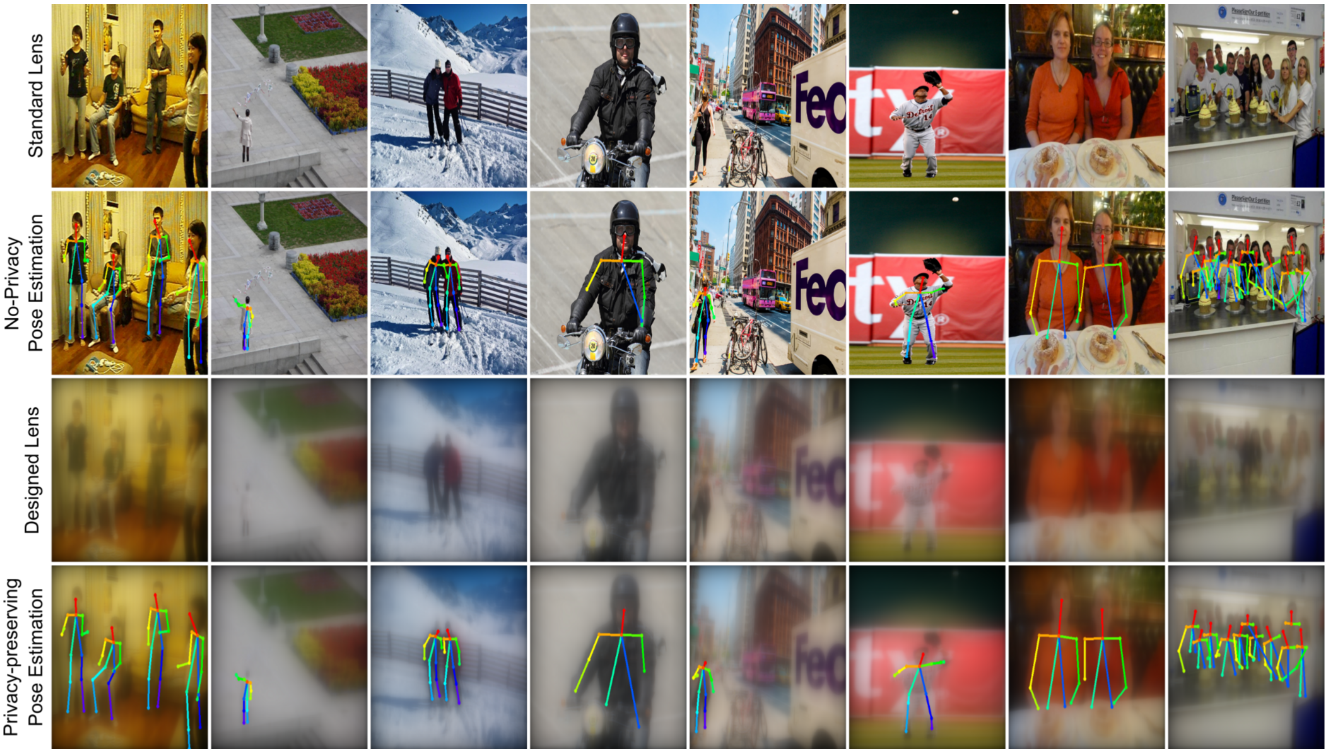
Standard Lens
No-Privacy Pose Estimation
Designed Lens
Privacy-preserving Pose Estimation

Figure 6. Qualitative results of some example images in the COCO 2017 dataset using the Lightweight OpenPose [9] (LOPPS) network as the CNN-decoder. We compare our proposed privacy-preserving pose estimation results using our optimized lens with the Non-privacy approach using a standard lens.