A Supplemental Material

A.1 Notation

I	Input image
x	Input tensor
$\mathcal{F}$	Optical transform family
$\hat{x}$	Keyed tensor
$\hat{x}_0$	Vectorized optically transformed image
x_0	Vectorized raw image
W	Layerwise linear transformation
$\hat{W}$	Keyed layer, AWA^{-1}
$\mathcal{N}$	Compositional conv-net function
$\mathcal{N}_i$	Layerwise conv-net function
$\mathcal{N}(x;W)$	Conv-net with input x and parameters W
$\hat{\mathcal{N}}$	Key-net with input Ax and parameters AWA^{-1}
A	Optical transformation matrix
g	Non-linear activation function
$\lvert\cdot\rvert_0$	L_0 norm
α	User specified privacy parameter
$\mathcal{F}_\alpha$	$\mathcal{F}_\alpha \subseteq \mathcal{F}$, given privacy parameter α
D	Photometric analog elementwise gain and bias
Π	Stochastic matrix, geometric optical shuffling

A.2 Proof of Commutativity

Lemma A.1. *The function composition $f(g(h(x))) = P^{-1}(ReLU(Px))$ is commutative for generalized permutation matrix $P = D\Pi$ with permutation matrix Π and diagonal matrix D, if $D \geq 0$.*

Proof. Let $g(x) = \text{ReLU}(x)$ amd $f(x) = P^{-1}x$ and $h(x) = Px$. The function composition $f(g(h(x))) = P^{-1}(\text{ReLU}(Px))$ is commutative if the equivalence relation $f(g(x))) = g(f(x))$ holds. Given a diagonal matrix $D \geq 0$ (i.e. has non-negative entries), the product $P = D\Pi$ for permutation matrix Π is non-negative, since permutation matrices are monomial and the product of non-negative matrices is non-negative. The function $y = \text{ReLU}(x)$ is computed elementwise as $y_i = \texttt{max}(0,\ x_i)$. Observe that any non-negative scale factor β is commutative such that $\text{ReLU}(\beta x_i) = \texttt{max}(0,\ \beta x_i) = \beta y_i = \beta\ \text{ReLU}(x_i)$, since a non-negative scaling does not change the sign of x_i. This can be written in matrix notation with β on the diagonal of D, then $\text{ReLU}(Dx) = D\ \text{ReLU}(x)$. Furthermore, since $\text{ReLU}(x)$ is computed elementwise and Π is a one-to-one mapping (e.g. a permutation), $\Pi^{-1}\text{ReLU}(\Pi x) = \text{ReLU}(x)$. Therefore, $\Pi^{-1}D^{-1}\text{ReLU}(D\Pi x) = \Pi^{-1}\text{ReLU}(D^{-1}D\Pi x) = \text{ReLU}(\Pi^{-1}D^{-1}D\Pi x) = \text{ReLU}(x)$. □

A.3 Proof of Sparsity Bound

Lemma A.2. *Given a sparse matrix W and any $A \in P$ and $B \in P$ where P is the family of generalized doubly stochastic matrices with privacy parameter α, there exists a sparsity upper bound $|AWB|_0 \leq \alpha^2|W|_0$.*

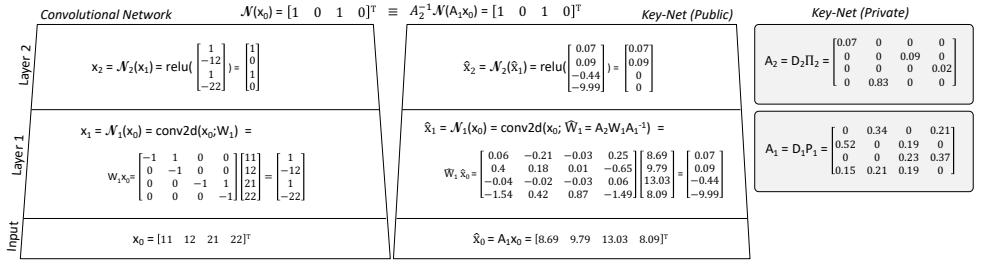


Figure 5: Optically Transformed Convolutional Networks. In this example, a 2x2 image $[11,12;21,22]$ is input to a 2 level convolutional network with a convolutional layer and ReLU layer, forming an inference x_2. The key-net is constructed from the conv-net using the private keys A, such that A_1 is a linear transformation implemented in the optics and analog processing of a custom vision sensor forming the sensor observation $\hat{x}_0$. This optically transformed sensor measurement is input to the key-net with output $\hat{x}_2$. The result of a forward pass in the conv-net is $x_2 = A_2^{-1}\hat{x}_2$, however the raw image x_0 is never observed or recovered to perform inference in the key-net.

Proof. Let W_k be a sparse matrix with exactly one non-zero element, then the decomposition $W = \sum_k W_k$ such that if $|W|_0 = N$ then the decomposition has N terms. Then, for any conformal matrices A and B, $AWB = \sum_k AW_kB$. Since $A \in P$ and $B \in P$, there exists a decomposition $A = D\sum_i \theta_i \Pi_i$ (resp. $B = D\sum_j \theta_j \Pi_j$). The sparsity pattern is upper bounded as $|A|_0 \leq |\sum_i \theta_i \Pi_i|_0$, when setting $D = I$. Each term AW_iB can be expanded into exactly α^2 terms of the form $\sum_{i,j} \theta_i\theta_j \Pi_i W_k \Pi_j$. The product $\Pi_i W_k \Pi_j$ is a permutation of W_k with sparsity $|\Pi_i W_k \Pi_j|_0 = 1$. Therefore, the sum of α^2 terms will have at most α^2 non-zero elements for every non-zero element in W, hence $|AWB|_0 \leq \alpha^2 |W|_0$. □

A.4 Keynet Example

Figure 5 shows a simple example of a key-net. In this example, there is a 2x2 raw image vectorized into a 4×1 vector (x_0) which is input to a two level convolutional network. This network includes a convolutional layer with kernel $[-1,1]$ (or equivalently a Toeplitz matrix W_1), followed by a ReLU layer. The output of this two layer convolutional network is a vector $[1,0,1,0]^T$. The key-net uses private keys A_1 and A_2 to transform the input and network weights, such that the weights $\hat{W}_1$ cannot be factored to recover either A or W. The key-net operates on the transformed input $\hat{x}_0$ which is observed in a custom designed vision sensor such that A_1 is equivalent to a physically realizable optical and analog transformation chain. Inference in a key-net operates equivalently to the conv-net with transformed weights of the form $\hat{W} = AWA^{-1}$, such that the key-net output is a vector $\hat{x}_2 = A_2 x_2$. This output is equivalent to the conv-net output, encrypted such that $x_2 = A_2^{-1}\hat{x}_2$. This is a homomorphism, enabled by an optical transformation A_1.

A.5 Optical Realization

The sufficient conditions for an optical transform in section 3.1 define a feasible family of transformations for use in a privacy preserving vision sensor. In this section, we show that the selected family of optical transforms based on generalized stochastic matrices can

be physically realized using an analog and optical processing chain based on *3D printed incoherent fiber bundle faceplates*.

An optical fiber bundle faceplate is an optical element constructed using a bundle of multi-micron-diameter optical fibers bundled into a thin plate with polished faces. An incoherent faceplate consists of fiber optic strands that are shuffled and rotated so that the faceplace will unfaithfully transmit an image from one face to the other, but in a deterministic manner. Recent work has demonstrated that optical fiber faceplates can be constructed using 3D printing of thermoplastic filaments [64]. This enables large scale manufacturing for design of privacy preserving vision sensors.

Figure 3 shows the design of the optical element to realize a generalized stochastic matrix. In this design strategy, a lens focuses the light field of the scene onto the optical fiber bundle. This fiber bundle is designed to implement the doubly stochastic matrix, which shuffles observed pixels and re-transmits them to an alternate location, which is then observed by the CMOS sensor. Next, during pixel readout, analog preprocessing applies an analog bias and gain. The resulting pixel readouts are converted from analog to digital (ADC) forming the observed sensor measurement. The combination of the fiber bundle to implement the stochastic matrix fractional shuffling (Π) and the analog processing to implement the pixelwise multiplicative scaling and additive bias (D) results in a physical realization of the optical transformation in (7). Figure 3 (right) shows the optical simulation of the fiber bundle for a mild permutation, without analog effects for visualization purposes. Simulation details are described in Section A.5.

A 3D printed incoherent fiber bundle faceplate includes the following primary design variables. First, 3D printed optics are *air clad* such that each fiber strand is separated from neighboring strands by an air gap. This introduces crosstalk due to cladding leakage between fiber strands which reduces the optical transmission fidelity. Second, 3D printed optics exhibits a *minimum fiber diameter* which limits the minimum size of the each optical strand. This minimum dimension is specified by the diameter of the 3D print head, which is on the order of 100μm on modern printers. This is two orders of magnitude larger than a pixel pitch on a CMOS sensor, which requires that each strand covers a pixel neighborhood. Finally, fiber optic transmission is specified by total internal reflection (TIR), which introduces a cone of projection from the end of the fiber to the CMOS sensor. This introduces *mixed pixels* where the observed intensity is a mixture of the contribution from all neighboring fibers. Figure 3 (right) shows examples of these modeling errors which must be addressed during sensor calibration.

In Section 4, we addressed these modeling errors by simulating the fiber optic bundle using parameterization demonstrated by Wang et al. [64] and re-training the key-net to be invariant to these physically realizable effects. In the remainder of this section, we will describe the simulation of the optical element and the CMOS sensor to simulate the physically realized optical transformation.

A.5.1 Optical Simulation

An optical fiber bundle is simulated as follows. An image of arbitrary size is input to the simulation tools. The image pixel size is designer defined. Next a padded mask is defined that is slightly larger than the input image size, the pad size is designer selectable. The designer then sets the fiber core dimension in the row direction and separately in the column direction. The simulator allows for a designer defined open area to cladding ratio which allows for image information to be lost due to non-imaging areas in the fiber bundle. Cladding

and fiber core sizes are converted to number of pixels (using pixel size defined above). A matrix of the centroids of each fiber core in the bundle is initialized. There is an option for the designer to set a shearing factor which simulates manufacturing tolerances on the array of fiber cores used to form the bundle. A masking matrix is defined such that for areas of the bundle entered on each centroid matrix element and within the defined fiber core diameter light is transmitted, all other areas are blocked to a designer defined value. The individual fibers are arranged in a brick-like pattern, i.e. the core centers are offset by one half of the core diameter as the rows go down. The image is then masked with the core and interstitial matrices. The script then rasters through the image and fiber bundle to see which parts of the image fall within allowed fiber cores and which parts are masked. All parts of the image that fall within a given core are intensity averaged which sets the image resolution to be that of the fiber core size. Lastly the designer can set crosstalk parameters for both the row and column directions of the bundle which enables the designer to input manufacturing tolerances and/or use of blocking materials between fibers. The crosstalk value operates like a kernel where the core image intensity is replaced by the vertical crosstalk factor times the sum of the four nearest neighbor vertical core elements plus the horizontal crosstalk factor times the sum of the two nearest neighbor horizontal core elements, normalized to the image maximum pixel value, for every fiber core in the bundle. This composite image is then taken as the input to the camera noise model defined below.

Figure 6 shows an example of this simulation. User configurable parameters for the fiber bundle simulation are:

1. Image size
2. Fiber core size row, column
3. Fiber core/cladding area ratio
4. Fiber bundle shearing factor
5. Fiber interstitial blocking factor
6. Vertical and horizontal fiber crosstalk coefficients

A.5.2 CMOS Sensor Simulation

The sensor noise model begins with a given photon intensity hitting a given pixel, this can be set by scaling the input image. The mean number of photons is given by μ_{ph} and assumes Poisson statistics to calculate the shot noise σ_{ph}^2 (in the limit of large numbers of photons we can use a Gaussian approximation to the Poisson distribution, this should be the case for this system). The sensor has a defined quantum efficiency depending on the wavelength, sensor materials, and sensor construction geometry, denoted by ν This then gives the number of photo electrons generated in the pixel μ_e which also follow Poisson statistics as: $\mu_e = \nu\mu_p$. Since the statistics are Poisson the variance is also μ_e. In addition to shot noise, the sensor also has a dark noise, i.e. photoelectrons are generated even in the absence of an optical signal. Usually this dark current is integration time (and temperature) dependent and is due to thermally induced electrons. The mean dark count is given by a constant term, μ_0 and the integration time dependent term $\mu_I \times t_{int}$ where μ_d is the sum of these two terms. Since the thermally induced electrons are also Poisson distributed the dark count variance can be written as: $\sigma_d^2 = \sigma_{d0}^2 + \mu_I t_{int}$.

Finally, the sum of these sources of photoelectrons charge a capacitor which turns the signal into a voltage, this gets amplified by a gain stage G and then is ultimately converted into a digital signal by the ADC. This process is assumed to be linear and the camera usually has some over system gain G_{sys} that converts electrons to digital counts out. The final signal

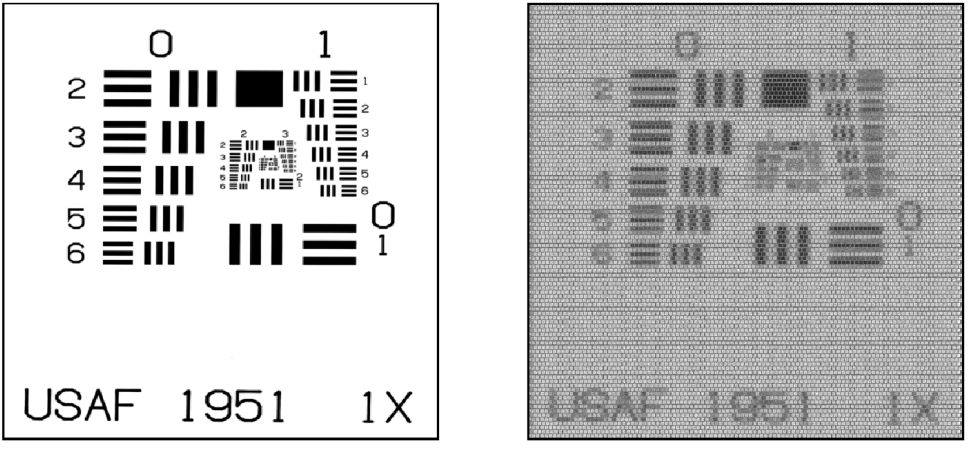


Figure 6: Optical Simulation of the 3D printed fiber bundle.

is then given by: $\mu_{gs} = G_{sys}(\mu_e + \mu_d)$.

Since the signal model is linear and the noise sources are independent, we can RSS the noise sources. As shown above the readout noise and amplifier noise can be lumped into a dark noise variance σ_d^2 and finally there is usually a noise associated with the final ADC stage, σ_q^2 that typically has a uniform distribution and is some camera dependent fraction of the digital scale output. Performing the RSS we get the total camera noise as: $\sigma_{gs}^2 = G_{sys}^2\sigma_d^2 + \sigma_q^2 + G_{sys}^2\mu_e$

Adjustable parameters for the camera noise simulation are:

1. Sensor pixel size and number – microns, $N \times N$
2. Sensor quantum efficiency - ν
3. Sensor dark noise – μ_d, σ_d^2
4. Sensor integration time - t_{int}
5. Sensor gain on a per pixel basis specified by a gain matrix - G_{ij}
6. ADC resolution and noise – depth, σ_q^2

As an example, the input image and final output image of the fiber bundle simulator and camera noise models is displayed in Figure 6.

A.6 Image Transformation Experiments

A privacy preserving vision system must obscure the image data from the perspective of a human observer, while allowing for vision based tasks. In this section, we provide ancillary experiments that demonstrate that off-the-shelf convolutional networks (and architectures) are subject to a privacy-performance trade-off, which is a limiting factor compared to the keynet architecture.

In our experiments, we consider face identification and (later) object detection as the target machine learning tasks. Face identification experiments were performed using the VGG-16 network architecture, with a pre-trained model [46]. Experiments for object detection were conducted using a PyTorch implementation of the Faster R-CNN object detector [52] trained on the MS-COCO dataset [38]. Where applicable, network weights were fine-

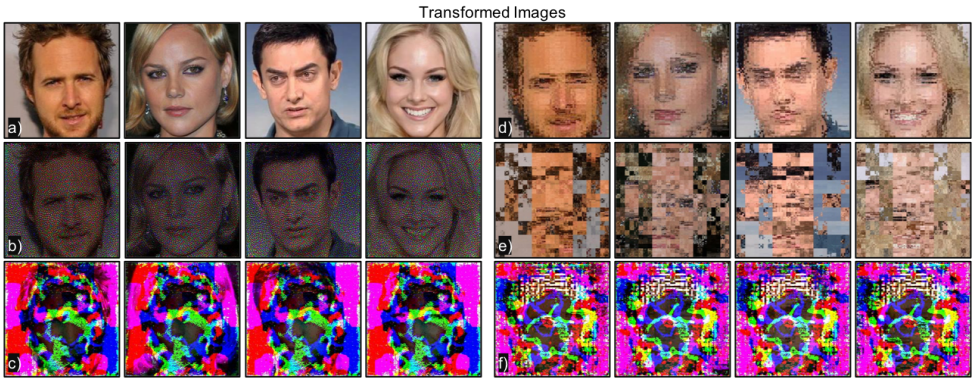


Figure 7: Image transformations. a) Reference images, b) Null-space learning, c) Trained-system learning, d) Geometric (permutation), low; e) Geometric (permutation), high; e) Combined geometric (low) and trained-system. Note that each transformation obscures the image differently.

tuned using the "training" subset and metrics are reported using the "validation" subset. To report the identification performance of baseline and learned networks, we utilize the Rank-1 classification accuracy. We acknowledge the limitations of the Rank-1 classification accuracy as a measure of matching performance [43], but for the purposes of this work it is only used as as an indicator of acceptable performance from the matcher, rather than a benchmark. Object detection performance was measured using average precision over 80 classes, using the standard COCO evaluation metrics [38]. Human perceptual loss in images is measured by the Structural Similarity Index (SSIM) [66]. The motivation for SSIM as a measurement to assess human observable changes in differences is owed to the fact that humans are much more likely to perceive structural changes in images. Therefore, the SSIM is used as a surrogate for image reconstruction fidelity between a reference (i.e., unmodified, "clean" image) and an optically transformed image. SSIM is a single-valued measurement and is defined in the range $[0,1]$, where 0 denotes no similarity (privacy preserving) and 1.0 denotes the images are equivalent (not privacy preserving).

A.6.1 Nullspace Learning for Frozen System

In Section 4, we pose a question asking if it is possible to identify an optical transformation that degrades an input image, while retaining performance of an off-the-shelf (i.e., pre-trained) ML system. In this experiment, we utilize a learned linear image transformation of image I to degraded image $\hat{I}$, with gain and bias parameters a and b, respectively. For simplicity, in this experiment, we restrict learning to only the gain parameter, a. Note that we also performed experiments with bias and gain/bias, which generated similar conclusions. We also enforce a constraint projection such that $\hat{I}$ is in the integer range of $[0,255]$.

$$\hat{I} = aI + b \tag{8}$$

Next, we define an adversarial loss that combines the primary task and an adversarial task, where the primary task is face identification and the adversarial task is image reconstruction (human perception). This loss measures the performance of the target task relative

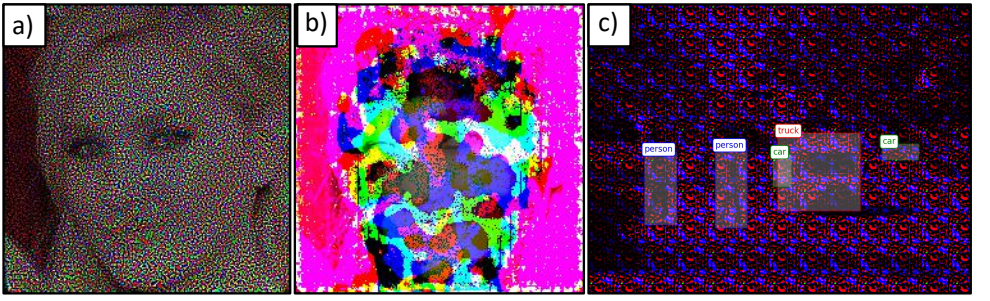


Figure 8: Adversarial learning examples. (a) Nullspace learning (b) Trained system learning (c) Tiled trained system learning for object detection. These learned degradations exhibit a clear utility/privacy tradeoff for face identification and object detection.

to the adversarial task, such that this loss is minimized when the target task performance is maximized and the adversarial task performance is minimized.

$$loss = L_{primary} + L_{adversarial} \tag{9}$$

For a frozen ML system, the weights of the network cannot be modified. Intuitively, we know that in order to retain matching performance, the convolutional responses of the network must be preserved. Therefore, our primary loss ($L_{primary}$) is defined as the L_2 difference of network $\mathcal{N}$ at layer k. The idea here is that if the convolutional responses at layer k are preserved, the downstream network responses will also be preserved.

$$L_{primary} = ||\mathcal{N}_k(I) - \mathcal{N}_k(I_m)||_2 \tag{10}$$

The adversarial loss ($L_{adversarial}$) is the compliment of the SSIM function, since our goal is to minimize the SSIM towards zero.

$$L_{adversarial} = 1 - SSIM(I, I_m) \tag{11}$$

The middle-left set of images in Figure 7 illustrate an example of a nullspace learned image transformation for the face identification task, where the primary task loss was applied at the "conv5" layer of the network. The reference (i.e., unmodified) images are in the upper left. Note that the transformed images are minimally degraded; they appear darker and with some high frequency noise. This transformation is not privacy preserving. Metrics from this experiment are a Rank-1 classification accuracy of 0.921 and an SSIM value of 0.065. These metrics are also listed in the "Null Space" row of Table 1 and the training output can be viewed in Figure 9. We emphasize that for this mask model, we were not able to learn a transformation with a lower SSIM (human perception loss) that achieved reasonable performance at the identification task.

A.6.2 Trained System Adversarial Learning

The previous experiment provides evidence that there *does not* exist an optical transformation that can degrade an input image, while preserving performance of a pre-trained ML system. The logical next question is to pose the question asking if such an optical transformation exists if we also jointly minimize a ML task loss (as in §4, *Trained System*). If we

can accomplish this joint learning task, a key-net would also be unnecessary, as a fine-tuning exercise would be sufficient. In this section, we report on such an experiment.

Out joint trained system retains the same mask model as the previous experiment. That is, the image transformation is linear (8) without a bias parameter. As with the nullspace learning experiment, we utilize an adversarial loss function, which is now regulated by an β parameter and switches between values of 0 and 1. When $\beta = 0$, the network weights are updated for the primary task and when $\beta = 0$, the transformation weights are updated for the adversarial task. This approach was considered to ensure degradation of the image would not be skewed toward background content. Note that for the primary task, the switching point occurred when the cross-entropy loss on the training set decreased below a value of 0.1 (i.e., saturated). For the adversarial task, the switching point occurred when the Rank-1 classification rate on the validation data decreased below 86%.

$$loss = \alpha L_{primary} + (1-\beta) L_{adversarial} \tag{12}$$

Since in this experiment the weights of the ML system are being learned, we must adjust the $L_{primary}$ loss term accordingly. For the face identification task, we set this loss term to be the cross-entropy loss function. The adversarial loss term, $L_{adversarial}$, retains unchanged (11). The bottom-left set of images in Figure 7 illustrate an example of a transformation learned from this approach (the reference images are in the upper left). Here, we see a considerable increase in the magnitude of the transformation. Arguably, it may be possible to infer that the example images are of faces, but it is very difficult to deduce the identity, even when provided the reference image data. However, the trade-off for this transformation is a slightly reduced identification rate. In this experiment, we achieved a rank-1 accuracy of 0.892, before the adversarial loss saturated with an SSIM value of 0.007. This data is also reported in the "Trained System" row of Table 1 and the training output can be viewed in Figure 9. Note that the Rank-1 accuracy from training only slightly decreases as a function of training time. This behavior is due to the cyclical nature of the joint optimization.

A.6.3 Geometric Trained System

In the previous two experiments, the image transformation directly modified the value(s) of the image data. This is not the only mechanism for generating an image transformation. As described in §3.3, we can also create a permutation matrix to "shuffle" the image data, which can destroy visual cues for human identification. Intuitively, we expect that permuting an image would cause a pretrained ML algorithm to fail at its task on this type of data. In this section, we explore whether it is possible to finetune a pretrained network to perform its primary task, except with permuted image data.

In this experiment, the actual image transformation is not learned. Instead, the transformation function was carefully crafted to minimize convolutional responses specifically from the VGG-16 network. The transformation function used permutes blocks of neighboring pixels globally an locally. The global transformation is a constant translation for each block. The local transformation is a localized permutation within the image block. This approach also allows us to regulate (or parameterize) the "amount" of shuffling that is applied. Optically, this mask model is a surrogate for a custom optical element utilizing fiber bundles. The top-right and middle-right set of images in Figure 7 illustrate examples of permuted images using a low-shuffling (top-right) and a high-shuffling (middle-right) approach. Note that the low-shuffling approach appears similar to a blurring function. These images are still

Table 1: Summary of baseline and optical transformation performance for the face identification task.

Experiment	Rank-1 Accuracy	SSIM
None (Baseline)	0.945	1.0
Null Space	0.921	0.065
Trained System	0.892	0.007
Geometric (low)	0.940	0.339
Geometric (high)	0.830	0.057
Combined	0.891	0.004

very human recognizable. The high-shuffling approach however generates images that are not human recognizable. As with the trained system adversarial learning results, it may be possible to deduce that these (permuted) images are of faces, but it is not possible to infer identity. We observed that after finetuning the VGG-16 network, the validation accuracy saturated at approximately 94% and 83% for the low-shuffling and high-shuffling approaches, respectively. These metrics are also reported in Table 1 in the "Geometric (low)" and "Geometric (high)" rows, respectively. The training output can be viewed in Figure 9. Note that for the "high" geometric transformation, the initial identification performance decreases to zero, but is quickly recovered (up to a point). Again, we find there is a privacy trade-off between the achieved identification performance and the reconstruction loss. We hypothesize that this loss in identification performance is due to violating locality of feature data and introducing edges from each local permutation block.

A.6.4 Combined Trained System

In this experiment, we combine the trained system approaches in §A.6.2 and §A.6.3. Here, the finetuned network for a "low" geometric transformation is also trained to learn a degraded image (8). We do not perform this experiment with the "high" geometric transformation because the identification performance is too low (83%) and learning the image degradation would only further reduce performance. The bottom-right row of images in Figure 7 illustrates examples of this transformation. This combination of transformations achieved a negligible difference in rank-1 identification accuracy (0.891 vs. 0.892) and SSIM value (0.004 vs 0.007) compared to the trained system approach without geometric permuting of the image data. These metrics are reported in Table 1 within the "Combined" row and the training output can be viewed in Figure 9.

A.6.5 Object Detection

The previous experiments demonstrate that for the face identification task, it is possible to (digitally) apply an optical transformation to an image that strongly reduces human perception (via SSIM) with a small loss in identification performance. Assuming the optical transformations are physically realizable, the image data is not representative of an end-to-end image acquisition to classification task. In an end-to-end task, the face data must be detected from a raw, full-scene image, prior to classification. As such, we performed an experiment to evaluate whether faces could be detected on transformed images.

In our preliminary face detection experiment, we applied the trained system optical transform (§A.6.2) to images in the VGG-Face-1 validation set and executed a face detector. The

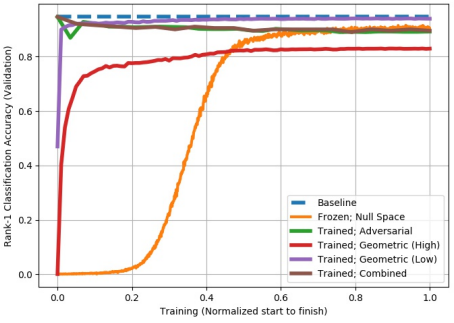


Figure 9: Summary of Rank-1 classification performance achieved for each face identification learning experiment vs. training run-time. A privacy preserving ML system *must* achieve similar performance to the unmodified baseline (dashed). Note that *all* of these approaches exhibit some amount of privacy-performance trade-off.

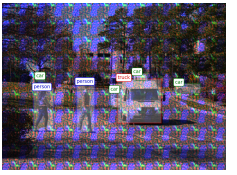
face detector was based on the Faster-R-CNN convolutional network and trained to detect faces in natural images. We observed a 0.0% detection rate on the transformed images, which suggests that the detector must also be trained for an actual end-to-end system.

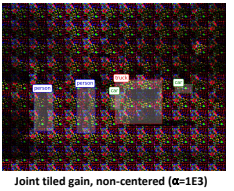
Next, we conducted an experiment to extend the trained system adversarial learning to a detection and classification task. Here, The object detection system is the Faster R-CNN convolutional network, trained on the MS-COCO object dataset. The primary machine learning task is localization and classification of 80 object classes (e.g. people, vehicles) and the adversarial machine learning task is structural similarity (SSIM) to degrade the image to reduce human perceptibility. In our experiment, we considered eight total configurations. Each configuration is listed in Figure 10 (left). Where denoted, "tiled gain" refers to the tiling of pretrained optical transforms from the face detection task. "Joint gain" denotes learning of an optical transform using the full scene image. Evaluation metrics are: AP=average precision, AP for small or large objects only, Relative performance=AP ratio relative to baseline showing performance loss between experimental configurations and baseline, SSIM=structural similarity index. Examples of applied image transforms are illustrated in Figure 10 (right). Results show that there is a strong trade-off in detection performance relative to the baseline. Similar to the null-space experiments for face identification, as it was not possible to learn a full-scene optical transform that did not exhibit considerable performance loss. These results continue to suggest evidence that alternative training strategies are not sufficient for privacy preservation and our key-net approach is required.

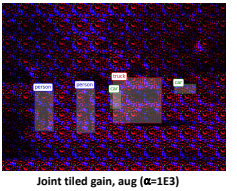
A.6.6 Summary

In this section, we performed experiments that justify the necessity of a keynet architecture for a privacy preserving vision sensor (§4). These experiments demonstrate that traditional conv-net architectures cannot be refactored to be privacy preserving with indirect (e.g., nullspace learning) or direct (e.g., joint adversarial learning, geometric data permutations) training of the ML algorithm. In each example there is a clear limit and tradeoff on the extent of human perception loss and performance of the primary ML task. This is evidenced in Table 1, which reports the achieved Rank-1 accuracy (primary task metric) and SSIM (human perception loss metric) for the face identification task. In contrast, a key-net does not inherit


Baseline


Joint tiled gain, aug (α=1E2)


Joint tiled gain, non-centered (α=1E3)


Joint tiled gain, aug (α=1E3)

Experiment	AP	AP (small)	AP (large)	Relative AP (%)	SSIM
Baseline	0.3653	0.2086	0.4773	100%	1.0
Joint gain (α=1E5)	0.2300	0.0883	0.3455	63.0%	0.14
Joint gain (α=1E6)	0.1413	0.0408	0.2490	38.7%	0.06
Joint tiled gain	0.1639	0.0396	0.2844	44.9%	0.06
Joint tiled gain, non-centered (α=1E3)	0.2178	0.0640	0.3515	59.6%	0.14
Joint tiled gain, non-centered (α=1E1)	0.3454	0.1885	0.4594	94.6%	0.42
Joint tiled gain, non-centered (α=1E2)	0.3015	0.1391	0.4243	82.5%	0.16
Joint tiled gain, aug (α=1E2)	0.2965	0.1352	0.4241	81.2%	0.21
Joint tiled gain, aug (α=1E3)	0.2120	0.0625	0.3451	58.0%	0.07

Figure 10: Object detection training study. Results with photometric optical transformation under different tiling and hyperparameter assumptions. See Section A.6.5 for details.

this privacy-performance tradeoff as its design is fully homomorphic (§3.2).

A.7 Privacy Analysis

In this section, we discuss keynet privacy. First, we connect the problem of recovering source conv-net weights to the problem of non-negative matrix factorization. Next, we show that the form of encryption we pose is an example of the Hill cipher, a classic cryptosystem based on linear algebra. Finally, we discuss the primary concern on semantic security, and introduce a challenge problem for the community to analyze it.

A.7.1 Non-negative Matrix Factorization

Non-negative matrix factorization (NMF) [35] is defined as follows. Given a matrix $V = WH$, factor V into terms (W,H) subject to the constraint $(W,H) \geq 0$, such that elements of the factors are non-negative. Non-negative matrix factorization in general is NP-hard, with special polynomial time factorizations where V is known low rank.

Let AWA^{-1} be grouped as $A(WA^{-1})$. In general, for positive definite matrix A with non-negative entries, the inverse A^{-1} will not be non-negative. Let $B = (WA^{-1})$, then B can be decomposed elementwise into the sum of non-negative terms as $B = B_p - B_n$ where B_{p_i}=0 if $B_i < 0$ else B_i (B_{n_i}=0 if $B_i > 0$ else $-B_i$, resp.). Then,

$$\hat{W} = A(B_p - B_n) \tag{13}$$

$$\hat{W} = AB_p - AB_n \tag{14}$$

which transforms the matrix $\hat{W}$ into the sum of products of non-negative matrices. The elements of A are non-negative by assumption, and the elements of B_n and B_p are non-negative by construction, so then factorization of AB_p or AB_n reduces to non-negative matrix factorization to recover the desired non-negative factor A, which can be used to recover A^{-1} and W. An efficient solution to this factorization requires a polynomial time solution to non-negative matrix factorization, for which exact NMF is NP-hard for full rank matrices [3][61]. Finally, in the case where exhaustive search is possible for "small" matrices V,

NMF in general is non-unique unless further constrained [28]. So, even if NMF is feasible, the matrix decomposition to recover exactly A is still infeasible.

A.7.2 Hill Cipher

The form of optical transformation described is known in the cryptographic literature as a Hill cipher [27]. The Hill cipher is a classic cryptosystem based on a linear transformation matrix as secret key. Transformed images (Ax) are robust to cryptanalysis and can be safely made public, as long as the key A is kept secret. Furthermore, as described in section A.7.1, the product AWA^{-1} is also secure to known ciphertext attacks, due to the hardness of non-negative matrix factorization. This enables public disclosure of both optical transformed images and key-nets, while ensuring security of raw images and source network weights.

However, the Hill cipher exhibits a two known weaknesses in the form of *chosen plaintext* and *chosen ciphertext* attacks. In a chosen plaintext attack scenario, the unknown A can be recovered through least squares regression with at least N tuples (x,Ax), for A with known sparsity $|A|_0 = N$. However, this requires that the attacker has physical access to the sensor, and in this scenario, privacy has already been compromised. The sensor can be assumed to be locked in a private space such as the home, with physical access restricted to authorized users, so tuples (x,Ax) cannot be collected by policy.

The Hill cipher also exhibits a weakness to *chosen ciphertext attack*. In this attack scenario, the adversary is provided decryptions $A^{-1}y$ of a chosen ciphertext y. Like the chosen plaintext attack, the unknown A^{-1} can be estimated using least squares regression. However, the key-nets will not be used in this scenario by design, as the image does not require decryption and the output inferences can be public. So, while the Hill cipher does have a weakness as a general cryptosystem, we believe it is an appropriate and practical assumption for a privacy preserving vision sensor.

Finally, the most challenging requirement is proving *semantic security*. Semantic security is the problem of exposing information about the plaintext given only the ciphertext. For example, in a key-net consider the case where the optical transformation function is the identity matrix. The resulting key-net is exactly the source network, and the encrypted images are identical to the raw images. Clearly, this provides no security. A more subtle challenge for semantic security is when the optical transformation is a diagonal matrix or a permutation matrix. In section A.7.3, we discuss that these transformations exhibit a semantic security weakness, which exposes the structure of $\hat{W}$ to attack. We discuss that using the generalized stochastic matrix with privacy parameter $\alpha > 1$ shows promise to defend against this attack.

A.7.3 Semantic Security

Semantic security is the problem of exposing information about the plaintext given only the ciphertext. A subtle challenge with semantic security considers the case where the degradation is either a scaling or a permutation rather than a generalized stochastic matrix. In the case of a scaling, the weakness leverages natural image statistics for recovery, such that gradients are sparse for neighboring pixels (e.g. images are smooth almost everywhere). For example, blind deconvolution techniques with Total Variation (TV) regularization can be used in some cases to jointly recover the unknown degradation kernel and the original image. For key-net attacks, the concept is to leverage the distribution of sparse gradients in natural images, which can be used to regularize this ill-posed problem to recover the unknown image

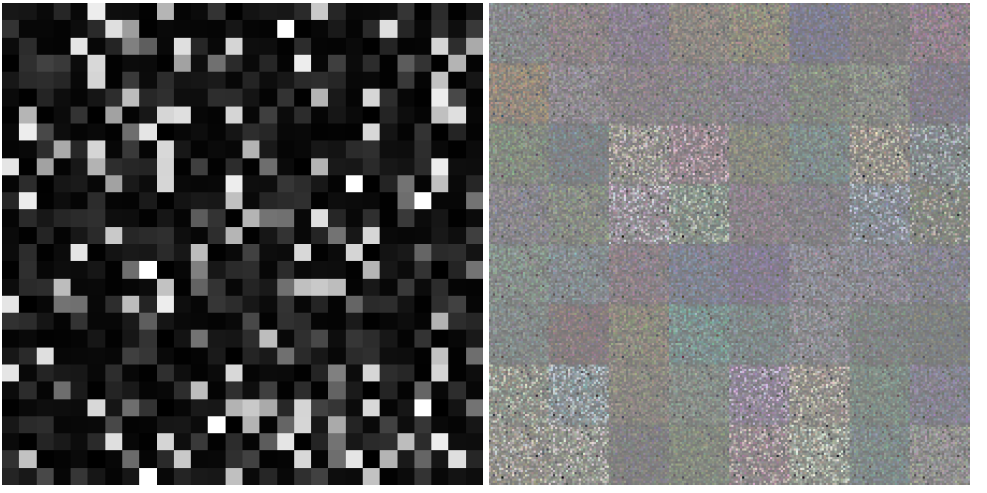


Figure 11: Keynet challenge problem. These images contain a secret message. We will release these images along with their paired keynet to challenge the research community to discover a weakness in semantic security of our proposed approach.

mask and raw image. Future work will consider these different optimization strategies to determine conditions for which image reconstruction using this strategy is feasible.

A second subtle challenge for semantic security is when the optical transformation is a permutation matrix. In this case, the neighborhood structure of a convolution is present in the non-zero structure of the Toeplitz matrices in the key-net. The keyed layers of the key-net are public information, so the sparsity structure of the weights can be inspected and used by an attacker. For example, there exists a greedy optimization based on graph embedding to recover the structure of a permuted image with known neighbors simply by minimizing the pairwise embedding distance of pixels. This is analogous to "puzzle solving", with the simplification that puzzle piece neighbors are observable in the sparsity pattern in the Toeplitz matrices which implement keyed convolutions. This is not a risk if the key-net is kept private, but if the key-net is public, then $\hat{W}$ exposes private information about Ax. Introducing the privacy parameter α can mitigate this attack by making the neighborhood structure ambiguous by increasing the sparsity of W by a user specified privacy factor that is independent of the true neighborhood structure. This introduces a tradeoff between inference runtime/memory and privacy that mitigates this attack. Furthermore, combining the permutation with an analog scaling and bias results in limiting the attack due to natural image statistics. Future work will investigate the feasibility of this style of attack for key-net images as a function of α.

Finally, the conditions listed in Section 3.1 are sufficient, but not necessary. Future work will explore alternative selections of the image key A_0 that are positive semi-definite. In this case, the sensor observation cannot be inverted to recover the image, even under a plaintext attack, since the least squares optimization is under determined. In the key-net framework, we would set $A_0^{-1} = I$ and continue the key-net encoding as currently described. This would further protect against semantic security attacks, but would likely introduce a utility/privacy tradeoff which would degrade the trained ML task performance as A_0 becomes increasingly rank deficient.

A.7.4 Challenge Problem

Finally, we plan on publicly releasing the challenge images in figure 11 and associated public key-nets for a lenet and vgg-16 topology. These challenge images contain a secret message that can only be discovered by exploting a weakness in semantic security. We would like to encourage the community to collaborate to discover such weaknesses in our approach by sponsoring a prize challenge. These images and public keynets are available for analysis at https://visym.github.io/keynet.