SACoD: Sensor Algorithm Co-Design Towards Efficient CNN-powered Intelligent PhlatCam

Yonggan Fu[1], Yang Zhang[2], Yue Wang[1], Zhihan Lu[1], Vivek Boominathan[1],
Ashok Veeraraghavan[1], Yingyan Lin[1]
[1]Rice University [2]MIT-IBM Watson AI Lab

Abstract

There has been a booming demand for integrating Convolutional Neural Networks (CNNs) powered functionalities into Internet-of-Thing (IoT) devices to enable ubiquitous intelligent "IoT cameras". However, more extensive applications of such IoT systems are still limited by two challenges. First, some applications, especially medicine- and wearable-related ones, impose stringent requirements on the camera form factor. Second, powerful CNNs often require considerable storage and energy cost, whereas IoT devices often suffer from limited resources. PhlatCam, with its form factor potentially reduced by orders of magnitude, has emerged as a promising solution to the first aforementioned challenge, while the second one remains a bottleneck. Existing compression techniques, which can potentially tackle the second challenge, are far from realizing the full potential in storage and energy reduction, because they mostly focus on the CNN algorithm itself. To this end, this work proposes SACoD, a ***S****ensor* ***A****lgorithm* ***Co-D****esign framework to develop more efficient CNN-powered PhlatCam. In particular, the mask coded in the PhlatCam sensor and the backend CNN model are jointly optimized in terms of both model parameters and architectures via differential neural architecture search. Extensive experiments including both simulation and physical measurement on manufactured masks show that the proposed SACoD framework achieves aggressive model compression and energy savings while maintaining or even boosting the task accuracy, when benchmarking over two state-of-the-art (SOTA) designs with six datasets across four different vision tasks including classification, segmentation, image translation, and face recognition. Our codes are available at: https://github.com/RICE-EIC/SACoD.*

1. Introduction

Recent CNN breakthroughs trigger a growing demand for intelligent IoT devices, such as wearables and biology devices (e.g., swallowed endoscopes). However, two major challenges are hampering more extensive applications of CNN-powered IoT devices. First, some applications, especially medicine- and biology-related ones, impose strict requirements on the form factor, especially the thickness, which are often too stringent for existing lens-based imaging systems. Second, powerful CNNs often come at a considerable cost, whereas IoT devices are subject to limited resources [**?**, **?**, **?**, **?**, **?**].

For the first challenge, lensless imaging systems [**?**, **?**, **?**, **?**, **?**] have emerged as a promising rescue. For example, PhlatCam [**?**] replaces the focal lenses with a set of phase masks, which encodes the incoming light instead of directly focusing it. The encoded information can be either computationally decoded to reconstruct the images or processed specifically for different applications. Such lensless imaging systems can be made much smaller and thinner, because the phase masks are smaller than the focal lens, and they can be placed much closer to the sensors and fabricated with much lower costs. For the second challenge, many recent works focus on designing CNNs with improved hardware efficiency, i.e., by applying generic neural architecture search (NAS) to find efficient CNNs.

As such, a naive way to address the two aforementioned challenges simultaneously is to introduce lensless cameras as the signal acquisition frontend and then apply NAS to optimize the backend CNN. However, such approaches would result in disjoint optimization that can be far from optimal. A generic NAS would treat the camera as given, and only optimize the CNN. Likewise, existing phase mask designs for lensless cameras treat the CNNs as given, and only optimize the masks. Such disjoint optimization fails to (1) take advantage of the masks' potential computational capacity, with which the NAS optimization can be fundamentally improved, and (2) perform an end-to-end optimization.

[**?**] shows that, under some assumptions, the phase masks in PhlatCam essentially perform 2D convolutions on the incoming lights, and the convolution kernel is encoded in the masks. Moreover, unlike other convolutional layers,

the phase masks' convolutions are almost free (i.e., do not consume additional energy, computation power, or storage), **regardless of** what value each mask takes. Therefore, we aim to incorporate the phase mask design into NAS to enable an end-to-end optimization of the sensing-processing pipeline, while exempting a portion of the pipeline from the efficiency penalties. Such co-designs are expected to achieve better accuracy and efficiency tradeoffs.

To this end, we propose a **S**ensor **A**lgorithm **Co-D**esign (**SACoD**) framework to enable more energy-efficient CNN-powered IoT devices. While we develop and evaluate SACoD in the context of PhlatCam [**?**] based imaging systems, it is generally applicable to different sensing and intelligent processing systems. The successful proposal, design, and validation of SACoD is expected to positively impact many real-world applications by enabling CNNs to be more extensively deployed into IoT devices equipped with intelligent sensors. Our main contributions are:

- We propose SACoD, a novel co-design framework that jointly optimizes the sensor and neural networks to enable more energy-efficient CNN-powered IoT devices. To our best knowledge, SACoD is **the first** to propose sensor algorithm co-design for CNN inferences.
- We develop an effective design of the optical layer to (1) exploit its potential computation capability and (2) enable co-search of the optical layer and backend algorithm. We then characterize the trade-off between accuracy and the required area of the corresponding imaging systems to demonstrate its effectiveness under practical size constraints.
- Extensive experiments and ablation studies validate that SACoD consistently achieves reduced hardware costs/area while offering a comparable or even better task accuracy, when evaluated over **two SOTA lensless imaging systems** on **four vision tasks** (classification, segmentation, image translation, and face recognition) and **six datasets**. Experiments with fabricated masks are also provided to validate SACoD's advantages under the physical measurements.

2. Related works

Neural architecture search. Recently NAS [**?**, **?**] has attracted increasing attention. It eliminates the handcrafting process and automatically searches for neural architectures. Existing NAS techniques can be divided into three categories, evolution-based, reinforcement-learning (RL)-based, and one-shot NAS. As the computational overheads of evolution- or RL-based approaches can be unacceptably high, many techniques [**?**, **?**, **?**, **?**, **?**, **?**] have been proposed to reduce the searching cost, among which differentiable architecture search (DARTS) has gained intensive interests. While being conceptually general, SACoD in this paper adopts the DARTS method, where a super-network is optimized during search and the strongest sub-network is preserved and then retrained. The readers are referred to [**?**] for more details about NAS.

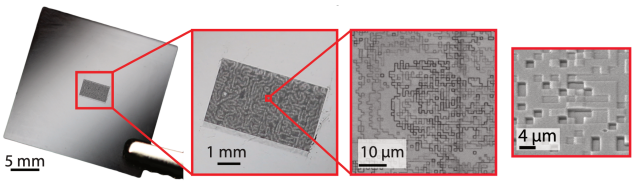


Figure 1: A fabricated phase mask used in the PhlatCam lensless imaging system [**?**].

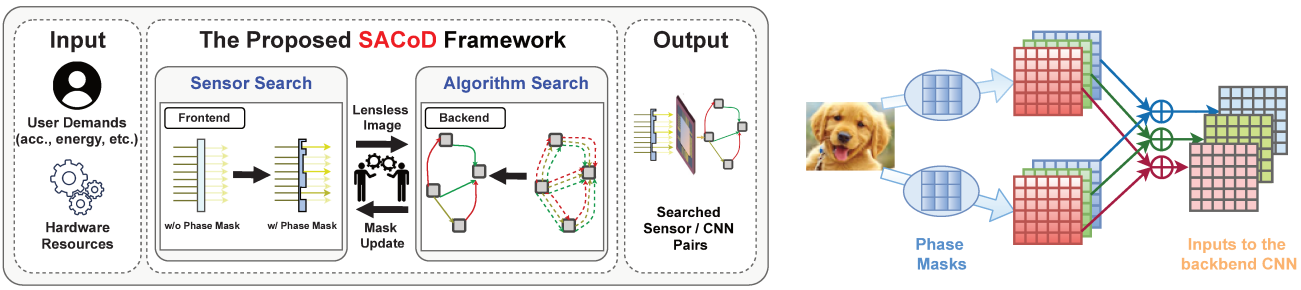
Lensless imaging systems. To eliminate the size or thickness burden caused by the lens, various lensless imaging systems have been developed. While lensless imaging systems have been widely used for capturing X-ray and gamma-ray [**?**, **?**], it is still in an exploring stage for visible spectrum uses [**?**, **?**, **?**, **?**]. In general, lensless imaging systems capture the scene either directly on the sensor or after being modulated by a mask element.

In this paper, we focus on a specific lensless imaging system based on phase masks called PhlatCam [**?**], which is a general-purpose framework to create phase masks that can achieve desired sharp point-spread-functions (PSFs). A phase mask modulates the phase of incident lights, and allows most of the light to pass through, providing a high signal-to-noise ratio. Hence, they are desirable for low light scenarios and photon-limited imaging. Fig. 1 shows a fabricated phase mask, which is essentially a transparent material with different thicknesses at different locations. Based on this lensless imaging system, we develop and validate our SACoD framework, aiming to explore and demonstrate the feasibility and advantages of sensor-algorithm co-design for enabling more efficient CNN-powered IoT solutions.

Sensor-algorithm co-training. There have recently been some attempts that try to jointly optimize the sensor parameters and the neural network backend. For lens-based image systems, novel lens designs are introduced and trained concurrently with the neural network backend to jointly optimize for image reconstruction [**?**], depth estimation [**?**], and high-dynamic-range imaging [**?**]. Similar approaches have also been applied to other imaging systems, including cameras with color multiplexing [**?**], PhaseCam3D [**?**], and Single Photon Avalanche Photodiodes cameras [**?**]. Yet these methods still consider the neural network architecture as fixed, and do not explore the potential of sensor-algorithm co-design.

3. The proposed SACoD framework

This section presents our SACoD framework. We first outline the framework and introduce the optical sensing frontend, and then describe how we implement SACoD's



(a) (b)

Figure 2: (a) An overview of the proposed SACoD framework, and (b) the proposed optical design as an interface between the frontend and backend of the SACoD pipeline, in which we take a two-channel mask as an example.

optimization algorithm. Finally, we provide discussions on the specificity and generality of SACoD generated masks.

3.1. SACoD: Framework setup

Overview. The SACoD framework shown in Fig. 2 (a) consists of two modules, an optical sensing frontend and a neural network backend. The coded masks of PhlatCam in the sensor are jointly optimized with the backend using a SOTA differential NAS algorithm [**?**], where the coded masks, together with the neural network weights, are regarded as network parameters.

Framework formulation. Specifically, the first module, i.e., the optical sensing frontend, is denoted as $O(\cdot;\boldsymbol{m})$, where $\boldsymbol{m} = \boldsymbol{m}(x,y)$ denotes the phase mask values. The optical layer is based on the PhlatCam system [**?**]. It receives the light signal from the object in front of the camera, processes the signal using the phase masks, and generates the sensor output. The second module, i.e. the neural network backend, is denoted as $F(\cdot;\boldsymbol{w},\boldsymbol{\alpha})$, where $\boldsymbol{w}$ represents weights of the neural network, and $\boldsymbol{\alpha}$ parameterizes the architecture. The neural network backend receives the sensor signal and produces an output for the intended applications.

Formally, we denote the light signal as $\boldsymbol{I}(x,y) \in \mathbb{R}^{H\times W\times 3}$, where x and y are the coordinate indices and H and W represent the height and width of the range of light that the camera can receive, respectively. The light signal contains RGB channel, and hence the last dimension is 3. Denoting the signal received at the sensor as $\boldsymbol{Z}(x,y) \in \mathbb{R}^{H'\times W'\times N}$, where H' and W' represent the height and width, respectively, N as the number of channels, and Y as the final output of the neural network backend, we have:

$$\boldsymbol{Z} = O(\boldsymbol{I};\boldsymbol{m}), \quad Y = F(\boldsymbol{Z};\boldsymbol{w},\boldsymbol{\alpha}). \tag{1}$$

The following subsections will introduce the form of $O(\cdot;\boldsymbol{m})$ and how to determine $\boldsymbol{m}$, $\boldsymbol{w}$, and $\boldsymbol{\alpha}$.

3.2. SACoD: The optical sensing frontend

Frontend formulation. Assuming that the light signal $\boldsymbol{I}(x,y)$ comes from an object whose distance to the camera is d, and that the depth of the object is relatively small, $O(\cdot;\boldsymbol{m})$ takes the following convolutional form [**?**]:

$$\boldsymbol{Z}(x,y) = O(\boldsymbol{I};\boldsymbol{m}) = \boldsymbol{p}(x,y;\boldsymbol{m},d) * \boldsymbol{I}(x,y), \tag{2}$$

where $*$ denotes 2D convolution, $\boldsymbol{p}(x,y;\boldsymbol{m},z)$ is called the *point spread function* (PSF) of the phase mask, which is determined by the phase mask $\boldsymbol{m}(x,y)$ and the distance d.

Once we optimize the PSF, the phase masks are designed for the PSF and a chosen d. The fabricated mask then produces the PSF at the given d. For the fabricated system shown in Sec. 4.5, d is set to be 2 mm for making our system much thinner than conventional cameras (thickness ranges between 7-20 mm). The mask is fixed at distance d to the sensor during operation, and thus the convolution property will continue to hold. According to Eq. (2), the optical layer can be regarded as a special convolutional layer. Note that one phase mask can only perform a single-channel convolution with a *positive kernel*, thus it takes two phase masks to implement a single-channel convolution with a real-valued kernel, where one implements the positive part of the kernel and the other implements the negative part. For example, in order to construct a three-channel convolutional layer with real-valued kernels, we need six masks in the imaging system. In addition, the input light has three color channels (R, G, and B), and each phase mask operates on all the color channels. Therefore, a three-channel convolution will produce a total of nine feature maps (FMs).

Optical layer design. To reorganize the rendered FMs as the input for the CNN backend, we propose the optical layer design in Fig. 2 (b) which takes a two-channel mask as an example. Specifically, it accumulates the FMs across the same color and outputs a 3-channel FM, which is still in an RGB-like shape. We adopt this design since it applies independent transformations on the RGB channels to maintain the original channel-wise discriminative information.

3.3. SACoD: The formulation and algorithm

SACoD formulation. Here we introduce the formulation and optimization of SACoD which aims to simultaneously optimize the phase mask $\boldsymbol{m}$, and the neural network's

architecture $\boldsymbol{\alpha}$, and the neural network's weights $\boldsymbol{w}$. Formally, SACoD aims to solve:

$$\min_{\boldsymbol{\alpha}} \mathcal{L}_{val}\left(\boldsymbol{m}^*(\boldsymbol{\alpha}), \boldsymbol{w}^*(\boldsymbol{\alpha}), \boldsymbol{\alpha}\right) + \lambda \mathcal{L}_e(\boldsymbol{\alpha}), \tag{3}$$

$$\boldsymbol{m}^*(\boldsymbol{\alpha}), \boldsymbol{w}^*(\boldsymbol{\alpha}) = \underset{\{\boldsymbol{m}, \boldsymbol{w}\}}{\operatorname{argmin}} \mathcal{L}_{tr}(\boldsymbol{m}, \boldsymbol{w}, \boldsymbol{\alpha}). \tag{4}$$

$\mathcal{L}_{tr}$ and $\mathcal{L}_{val}$ are task-specific performance losses evaluated on the training and validation set, respectively, $\mathcal{L}_e$ is the efficiency loss (*e.g.* model size, computational cost, or energy consumption), and λ is the tuning parameter trading-off the accuracy and efficiency. Following the same parameterization scheme in DARTS [**?**], $\boldsymbol{\alpha}$ denotes the weights of different candidate operations.

Modifications over DARTS. SACoD integrates two major modifications as compared to the original DARTS [**?**] framework. The first difference is that the efficiency loss $\mathcal{L}_e$, measured by the sum of each layer's computational cost weighted by the network parameter $\boldsymbol{\alpha}$, is introduced. More importantly, the second and major difference is that the phase mask $\boldsymbol{m}$ is optimized jointly in the framework. It is worth pointing out that although mathematically similar, $\boldsymbol{m}^*$ and $\boldsymbol{w}^*$ have different degrees of dependencies on $\boldsymbol{\alpha}$. Specifically, $\boldsymbol{w}^*$ is directly impacted by $\boldsymbol{\alpha}$ because $\boldsymbol{\alpha}$ governs which subset of the $\boldsymbol{w}$ is ultimately used, while $\boldsymbol{m}^*$ is only indirectly influenced by $\boldsymbol{\alpha}$. Therefore, incorporating $\boldsymbol{m}$ will largely improve the tradeoff between the model performance and model complexity. Note that SACoD is naturally compatible with other NAS methods. We adopt differential NAS for the fast generation of the optical mask and network. When using other NAS methods, e.g., RL-based NAS [**?**], we still observe similar system performance (within 0.3% accuracy on CIFAR-100), but the search time increases to 8 GPU-days from 0.5 GPU-days.

Two-stage workflow. The whole co-design process can be divided into two stages: a searching stage and a training stage. In the searching stage, we apply the alternate gradient descent of Eq. (3) and Eq. (4) to search for the optimal network architecture $\boldsymbol{\alpha}^*$. In the training stage, the optimal mask and weights are determined by optimizing Eq. (4) conditioning on the optimal network architecture $\boldsymbol{\alpha}^*$.

3.4. SACoD: Discussions

Specificity of SACoD generated masks. As formulated in Eq. (3), α controls the searched network structure, which favors different distributions of phase masks m^*. To validate the influence of α on m^*, we fabricate the physical masks under various settings and observe that the optimal masks for different searched networks are quite different, which are visualized in the Appendix. In addition, we evaluate SACoD generated masks against the transferred masks from other tasks in Sec. 4.6 to show the necessity of specifically customizing the masks for each target task.

Generality of SACoD generated masks. Considering (1) the captured features of the first several layers in CNNs are general and can be transferred among tasks [**?**], and (2) the masks are jointly optimized with both the network structure and the network weights in SACoD, it can be expected that SACoD's generated masks are able to learn to adapt to the general features of CNNs and thus can achieve better generality and transferability among vision tasks, compared with the masks based on fixed filters like Gabor-mask [**?**]. This advantage of SACoD is validated in Sec. 4.6.

Generality vs. specificity. There always exists a tradeoff between the achieved performance and the manufacture cost in practical uses of intelligent sensors, i.e., the benefits of higher accuracy and lower energy of specifically designed masks for the target task versus their higher manufacture cost compared with one-for-all fixed mask (such as Gabor-mask [**?**]). Fortunately, one key highlight of SACoD is that it achieves such high specificity at extremely low manufacture costs, as each mask costs one order of magnitude lower than lens-based cameras [**?**] in addition to PhlatCam's advantageous thin feature, indicating SACoD's general applicability on IoT applications.

4. Experiments results

This section presents evaluation results of SACoD applied on PhlatCam. We first describe the experiment settings in Sec. 4.1, and then benchmark SACoD over SOTA lensless imaging systems on classification tasks, IoT applications, and other vision tasks in Sec. 4.2, 4.3, 4.4, respectively. We next show the effectiveness of the physically fabricated masks generated by SACoD in Sec. 4.5 and provide various ablation studies of SACoD in Sec. 4.6.

4.1. Experiment setup

Optical layer constraints. As mentioned, the optical layer first performs convolutional operations on the input scene optically, the outputs of which are then processed by the backend neural network. The physical device construction imposes design constraints on the optical layer design. Specifically, since the phase mask is placed closer to the sensor, the optically achievable kernel size cannot be arbitrarily small [**?**]. Here, we adopt kernel sizes that are not smaller than 7x7. Additionally, since all the designed masks are sharing the same sensor area, the number of masks cannot be large due to the limited sensor area. Here, we constrain the number of masks to be no more than six. We adopt simulated masks in Sec. 4.2~ Sec.4.4 and evaluate on physically fabricated masks in Sec. 4.5.

Algorithm setting. Datasets: we evaluate SACoD on a total of **four vision tasks with six datasets**: two classification datasets CIFAR-10/100, two IoT datasets including FlatCam Face [**?**] and Head Pose [**?**], one segmentation dataset Cityscapes [**?**], and one unpaired image translation dataset horse2zebra [**?**]. The same and standard data augmentation (e.g., random crop and normalization) is adopted

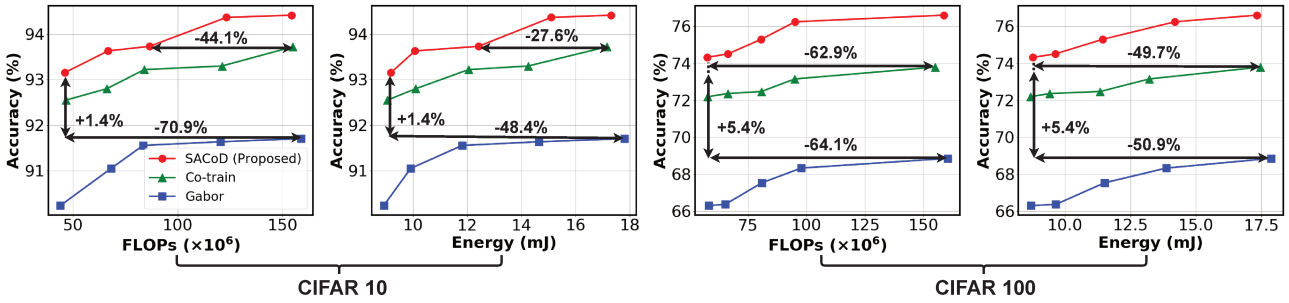


Figure 3: Accuracy vs. FLOPs/energy trade-offs of SACoD and the baselines on CIFAR-10/100.

for both SACoD and the baselines. Baselines: we evaluate SACoD against two SOTA lensless imaging systems:

- Gabor-mask System: we fix the optical layer to be the Gabor-mask [**?**] and search for networks using the same NAS method as SACoD.
- Co-train System: we fix the backend network to be a SOTA CNN (e.g., MobileNetV2 [**?**] for the classification task) and jointly train it with the optical layer.

Efficiency metrics: we consider both FLOPs (Floating Point Operations) and energy cost based on **real-device** measurements as the efficiency metrics. Specifically, we adopt the NVIDIA JETSON TX2 [**?**], a popular IoT GPU, as the target platform, which is connected to a laptop with the real-time energy cost being obtained via the sysfs [**?**] of the embedded INA3221 [**?**] power rails monitor.

4.2. SACoD over SOTA imaging systems on classification tasks

Settings. In this set of experiments, we search for neural networks on CIFAR-10/100 for both the SACoD and Gabor-mask systems, and quantize all the operations to 8-bit using a SOTA quantization training method [**?**], which is a common practice considering the constrained sources on IoT devices. We adopt the search space and training settings in [**?**] with minor changes, which are detailed in the Appendix. Here the model adopted by the Co-train baseline is MobileNetV2 [**?**]. To benchmark SACoD over SOTA imaging systems, we fix the number of masks to be six among all the settings, and then study their accuracy under different FLOPs and energy costs. We control the FLOPs of the SACoD and Gabor-mask systems by controlling λ in Eq. (3) and that of the Co-train system by changing the width multiplier [**?**].

Results analysis. Fig. 3 shows the trade-off between the accuracy and required hardware costs in terms of both FLOPs and energy cost for the SACoD and the two baseline lensless imaging systems on CIFAR-10/100. We can see that SACoD consistently requires reduced FLOPs and energy cost while achieving a comparable or higher accuracy over the baselines. On CIFAR-10, SACoD achieves a 44.1% and 70.9% reduction in FLOPs, and a 27.6% and 48.4% reduction in energy, while offering a +0.01% and +1.45% higher accuracy, compared with the Co-train and Gabor-mask baselines, respectively; On CIFAR-100, SACoD reduces the FLOPs by 62.9% and 64.1%, and energy cost by 49.7% and 50.9%, while achieving a +0.71% and +5.46% higher accuracy, compared to the Co-train and Gabor-mask baselines, respectively. This set of experiments validates that the end-to-end optimization engine in SACoD indeed can lead to superior performance in both task performance and hardware efficiency.

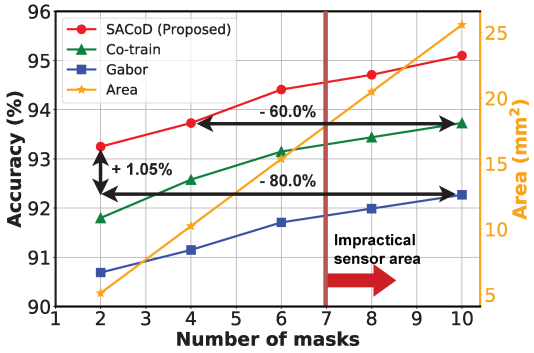


Figure 4: Achieved accuracy and sensor/mask area under various number of masks on CIFAR-10.

Considering that the form factor or area is another influential design factor in lensless IoT imaging systems, we evaluate SACoD over the baselines in terms of the trade-off between accuracy and area by controlling the number of masks in the optical layer, and summarize the results in Fig. 4. We can see that the proposed SACoD achieves the best accuracy-area tradeoffs among all the designs under the same number of masks (and thus area) and the same model size. In particular, SACoD achieves a 60.0% and 80.0% reduction in area while offering a +0.01% and +1.05% higher accuracy, compared with the Co-train and Gabor-mask baselines, respectively. As the sensor area becomes impractical with more masks, we constrain the number of masks to be no more than six in other experiments.

4.3. SACoD over SOTA imaging systems on IoT applications

Here we benchmark SACoD over the SOTA baselines on two IoT applications (including FlatCam Face recog-

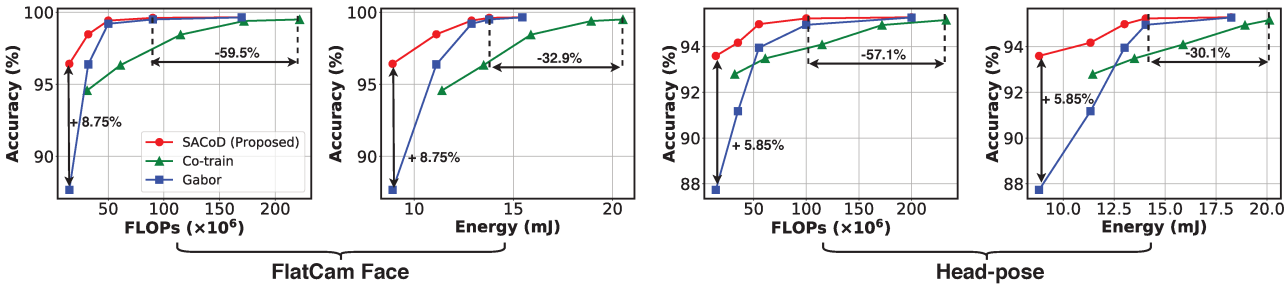


Figure 5: Accuracy vs. FLOPs/energy of SACoD over the baselines on two IoT tasks.

nition [?] and Head Pose detection [?]) to evaluate its effectiveness on real-world IoT tasks. Here we adopt the same search space as in Sec. 4.2 and further constrain the FLOPs of the derived backend CNNs to see if SACoD is still applicable to extremely energy-constrained scenarios. As shown in Fig. 5, we can see that again SACoD consistently outperforms the baselines under all settings in terms of accuracy-cost tradeoffs. Specifically, compared with the Co-train baseline, SACoD achieves a 59.5% and 57.1% reduction in FLOPs, a 32.9% and 30.1% reduction in energy cost with a +0.11% and +0.07% higher accuracy, on the FlatCam Face and Head Pose datasets, respectively. Meanwhile, compared with the Gabor-mask baseline, SACoD shows a better scalability to more energy-constrained scenarios: when the FLOPs or energy constraint is extremely low, SACoD achieves a +8.75% and +5.85% higher accuracy, under the same FLOPs/energy cost on the FlatCam Face and Head Pose datasets, respectively, indicating its superiority in more real-world IoT applications.

Table 1: SACoD over SOTA baselines on a segmentation task with the Cityscapes dataset.

Method	2 masks		4 masks		6 masks	
	mIOU	GFLOPs	mIOU	GFLOPs	mIOU	GFLOPs
Co-train	69.0	435.0	69.6	435.0	68.8	435.0
Gabor-mask	65.8	45.64	66.1	38.32	67.3	36.34
SACoD	**69.8**	**36.17**	**70.4**	**33.56**	**71.6**	**29.51**

Table 2: SACoD over SOTA baselines on unpaired image translation tasks. Row 2-4: zebra2horse dataset; Row 5-7: horse2zebra dataset. Lower FID indicates better results.

Method	2 masks		4 masks		6 masks	
	FID	GFLOPs	FID	GFLOPs	FID	GFLOPs
Co-train	147.03	54.17	140.70	54.17	139.83	54.17
Gabor-mask	137.79	6.89	141.11	5.04	145.87	7.15
SACoD	**136.35**	**5.93**	**136.41**	**3.89**	**138.23**	**3.57**
Co-train	66.82	54.17	61.21	54.17	68.26	54.17
Gabor-mask	91.87	5.87	106.27	4.34	88.36	4.72
SACoD	**89.80**	**3.70**	**86.00**	**3.82**	**87.10**	**4.03**

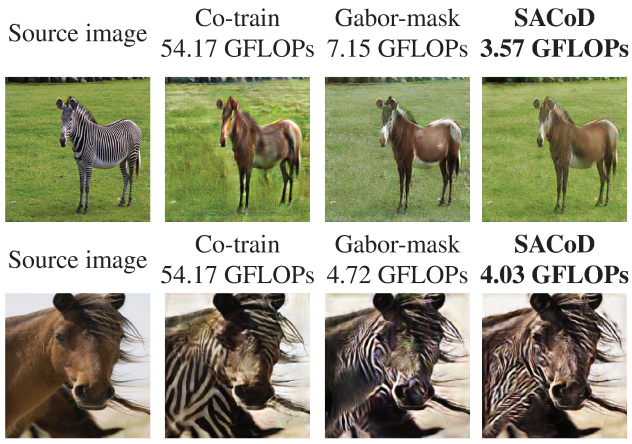


Figure 6: Visualizations of the translation results on the zebra2horse (row 1) and horse2zebra (row 2) tasks under six masks. The resulting FLOPs of each method are annotated.

4.4. SACoD over SOTA imaging systems on other vision tasks

Considering the diverse applications of IoT devices, we also evaluate SACoD on other vision tasks including one segmentation dataset (Cityscapes [?]) and one unpaired image translation dataset (zebra2horse and horse2zebra [?]), which require a more challenging trade-off on CNN-powered intelligent IoT devices.

Settings. We adopt the SOTA search spaces and settings in [?] for the segmentation task and [?] for the unpaired image translation task. The models adopted for the Co-train baseline are DeepLabV3 [?] with a ResNet-50 [?] backbone and CycleGAN [?] for the segmentation and image translation tasks, respectively. More details can be found in the Appendix.

Results on the segmentation task. Tab. 1 shows that SACoD achieves the highest mean Intersection Over Union (mIOU) under all the mask constraints, while requiring the smallest FLOPs. Specifically, SACoD achieves a 0.8%~4.3% higher mIOU and 12.4%~93.2% reduction in FLOPs over the Co-train and Gabor-mask baselines, respectively, under all the mask settings.

Results on the image translation task. We show both the quantitative results in Tab. 2 and the visualization ef-

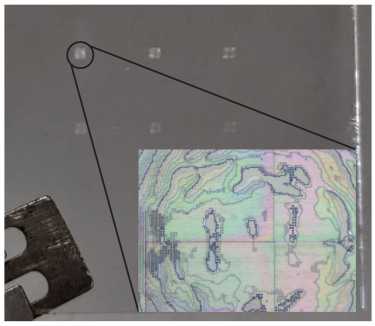


Figure 7: Fabricated masks.

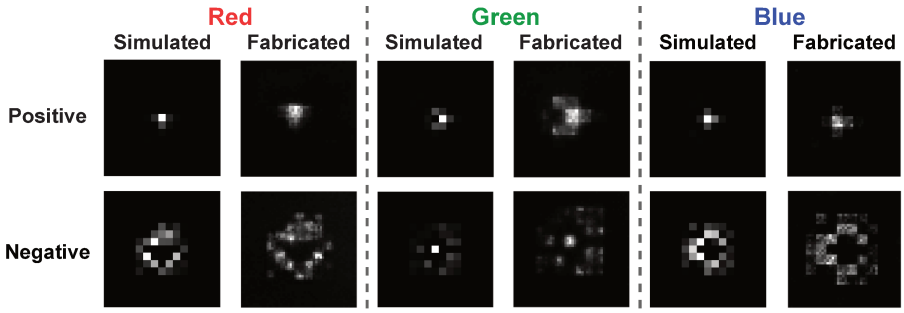


Figure 8: Visualizing the searched PSFs and the corresponding fabricated ones.

fects in Fig. 6 as the former cannot always capture the image quality. Tab. 2 shows that SACoD requires the smallest FLOPs under all the six cases, while Fig. 6 shows that SACoD provides notably the best visualization effect among all the methods. In particular, compared with the Gabor-mask baseline, SACoD achieves a $12.0\% \sim 50.1\%$ reduction in FLOPs with a $1.26 \sim 20.27$ better FID (the lower, the better), while providing notably better visualization effects; compared with the Co-train baseline, SACoD reduces the FLOPs by $92.56\% \sim 93.4\%$ and offers better visualization effects with more fine-grained textures.

The evaluation results on the above two vision tasks consistently validate SACoD's superiority over the baseline systems and indicate its general applicability in a wide range of IoT applications driven by intelligent sensors.

Table 3: Accuracy comparison of SACoD and Gabor-mask using the simulated and fabricated masks based on a real-world PhlatCam imaging system with CIFAR-10.

Method	Simulated (%)	Fabricated (%)	Gap (%)
Gabor-mask	91.71	87.17	4.54
SACoD	**94.41**	**90.02**	**4.39**
Improvement	**+ 2.70**	**+ 2.85**	**- 0.15**

4.5. SACoD with physically fabricated masks

Settings. To evaluate the performance of SACoD in real-world prototyped PhlatCam imaging system, we further fabricate the physical masks based on the PSF of the searched optical layer by SACoD. We then capture the real measurements of the CIFAR-10 dataset by displaying images on a monitor and capturing them using our prototyped PhlatCam imaging system with fabricated masks. The CMOS sensor in our prototype has a Bayer RGB filter array, so the sensor measurements after the mask can be split to different raw RGB color channels. Hence, our raw measurements have RGB channels as shown in Fig. 2 (b). All the backend models are under similar FLOPs (the rightmost points in Fig. 3).

Fabricated masks: Each phase mask is of size $600\mu\text{m}\times600\mu\text{m}$. At a time, 6 phase masks corresponding to 6 small filters are fabricated onto the same glass substrate in Fig. 7, which evenly fill the space of the sensor. Particularly, the 6 phase masks are fabricated in a 2×3 array with an even spacing of 4.4 mm.

Visualizing fabricated masks. Fig. 7 shows the microscope image of the six fabricated masks, under which SACoD achieves an accuracy of 94.43% on CIFAR-10, and Fig. 8 compares the visualization of the simulated and fabricated PSFs, in which the top/bottom row shows the positive/negative masks and the columns from the left to the right represent the three RGB-channels respectively. From Fig. 8, we can observe that the fabricated PSFs generally keep the original shape as compared to the simulated ones, while slightly shift in the brightness of some pixels.

Real measured accuracy. We compare the accuracy of the SACoD and Gabor-mask systems with simulated and fabricated masks in Tab. 3 and observe that (1) our SACoD still outperforms the Gabor-mask system with a +2.85% higher accuracy under fabricated-mask measurements, indicating the consistency of SACoD's superiority in both simulated and fabricated systems, and (2) both systems suffer from a 4% accuracy drop after fabrication and SACoD shows a slightly less accuracy drop (0.15%). We would like to clarify that the large accuracy drop could be attributed to non-idealities in in-house fabrication and other experimental errors such as mask-sensor alignments, which have been observed before, e.g., [**?**] shows that only 88% of the correctly classified images by the optimal model can be still correctly classified on MNIST when using real-fabricated masks. It can be expected that with industry-standard fabrication and manufacturing quality, the resulting accuracy drops after fabrication can be alleviated.

Real-world images captured by the fabricated masks. We visualize the images captured by fabricated masks generated by SACoD on CIFAR-10 and Flatcam Face in Fig. 9. Since the PSFs of different color channels are different according to Fig. 8, the captured images show a color shift over the original RGB images while still maintain good visual quality for recognition.

Figure 9: Visualizing the captured images by physically fabricated masks on CIFAR-10/Flatcam Face.

Table 4: Accuracy when using Gabor-mask, SACoD's generated masks transferred from those dedicated for the FlatCam Face dataset, and SACoD's generated masks customized for the target tasks on the CIFAR-10/100 dataset.

Method	CIFAR-10 Acc (%)	CIFAR-100 Acc (%)
Gabor-mask	91.71	68.85
SACoD (from FlatCam Face)	93.10	72.50
SACoD (customized)	**94.41**	**76.67**

4.6. Ablation studies of SACoD

Generality vs. specificity. To evaluate the generality and specificity of SACoD, we benchmark SACoD transferred from the FlatCam Face dataset against (1) SACoD customized for each target task and (2) the Gabor-mask baseline which is a general mask based on fixed filters, on the CIFAR-10/100 dataset. All the backend models have similar FLOPs (those corresponding to the rightmost points in Fig. 3). As shown in Tab. 4, SACoD with masks transferred from those dedicated for the FlatCam Face dataset achieves a +1.39% and +3.65% higher accuracy on CIFAR-10/100, respectively, over that of Gabor-mask, while suffering from a -1.31%/4.17% accuracy drop on CIFAR-10/100, as compared to SACoD customized for the target task. This validates the assumption in Sec. 3.4 that SACoD's generated masks show a better generality and transferability over masks based on fixed filters like Gabor-mask, while specificity, i.e., customization for each target task, of SACoD masks can further improve the achieved accuracy. One key highlight is that SACoD achieves specificity at extremely low manufacture costs, as each mask costs one order of magnitude lower than lens-based cameras [**?**].

Feature extraction of SACoD. To further explore the reason behind SACoD's success, we compare the discriminative power of the features captured by the optical layers of SACoD and the Gabor-mask baseline. Specifically, following [**?**], we average the optical layer's activations over the output channels to obtain a vector and use the corresponding softmax value as the feature distribution for each input image. We then calculate the KL divergence between the feature distribution from different classes to see how discriminative the features are. Fig. 10 visualizes the average KL divergence (over 100 randomly selected images) between every two classes on the test dataset of CIFAR-10. We can see that the feature distribution difference of SACoD between different classes is notably and consistently larger

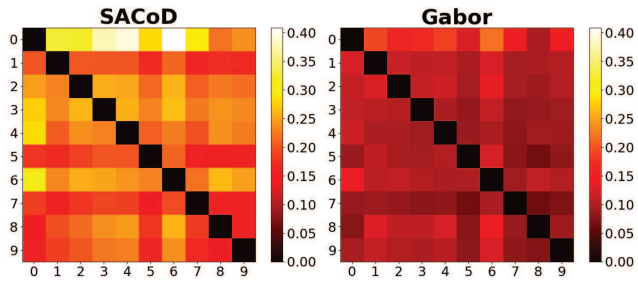


Figure 10: KL divergence of the output distribution between different classes captured by the searched optical layer of SACoD and Gabor-mask on CIFAR-10, where the x-axis and y-axis are the class id, and the heatmap value denotes the magnitude of KL divergence.

than that of the Gabor-mask baseline, further verifying that the optical layer of SACoD can more effectively extract the discriminative information from the input and thus reduce the required computations of the backend CNN.

SACoD vs. lens-based systems. To fairly benchmark against lens-based systems, we remove the optical layer and its associated constraints, and search for the optimal network within the same the search space [**?**]. We find that under a slightly reduced FLOPs (154M FLOPs vs. 158M FLOPs), SACoD achieves a 0.39% and 0.62% lower accuracy on CIFAR-10 and CIFAR-100, respectively, while reducing the thickness of the imaging systems by 10× which makes it possible to be integrated into more IoT applications. This set of experiments shows that our proposed SACoD can offer similar task performance and hardware efficiency as compared to lens-based systems, while being able to shrink the thickness of the system by one order.

5. Conclusion

We propose SACoD, a sensor algorithm co-design framework, to enable more energy-efficient and robust CNN-powered IoT systems, and validate it in the context of PhlatCam. A novel end-to-end co-search algorithm is presented to jointly optimize the coded mask of PhlatCam in the sensor and the backend CNN. Extensive experiments and ablation studies validate the superiority of SACoD in terms of both task performance and hardware efficiency as well as the its general applicability, when evaluated over SOTA lensless imaging systems on various tasks and datasets. The success demonstration of the sensor algorithm co-design principle in SACoD can positively impact many real-world IoT applications demanding intelligent sensors.

Acknowledgements

The work is supported by the NSF EPCN program (Award number: 1934767), the NSF RTML program (Award number: 1937592), the ONR funding (Award number: N00014-19-1-2440), the NSF CAREER program

(Award number: IIS-1652633), the NSF PATHS-UP program (Award number: EEC-1648451), and the NIH Rockefeller program (Award number: 1RF1NS110501).