Co-Optimization of Optics, Architecture, and Vision Algorithms

Samuel Triest
Department of Computer Science
University of Rochester
striest@u.rochester.edu

Daniel Nikolov
Institute of Optics
University of Rochester
daniel.k.nikolov@gmail.com

Jannick Rolland
Institute of Optics
University of Rochester
rolland@optics.rochester.edu

Yuhao Zhu
Department of Computer Science
University of Rochester
yzhu@rochester.edu

Abstract

Traditionally, cameras are solely imaging devices designed to capture images for human consumption. With the prevalence of computer vision as a major technique in many modern application domains, such as robotics, unmanned aerial vehicles, and video surveillance, it is no longer the case that cameras necessarily capture data for human use. Instead, there has been a recent trend toward intelligent cameras that extract visual insights. Although demand for these systems is high, there are many challenges in designing and implementing intelligent cameras. In many application domains, application-specific cameras need to operate within energy and latency constraints, which may not be feasible with a generic camera. However, the cost of hand-designing an application-specific camera is prohibitively expensive. Furthermore, in many scenarios an intelligent camera must be able to process a variety of different tasks. For instance, a camera on a mobile robot must be able to supply useful image data for depth estimation and object detection tasks. As such, while optimizations for a single task are desirable, they must be balanced with some level of flexibility. As such, this research is aims to answer the question: "How can we design intelligent camera systems that effectively balance efficiency and flexibility?" We answer this question by proposing the Self-Optimizing Intelligent Camera system, which enables efficiency through automatic tuning of its design parameters, and flexibility though dynamic adjustment of its design parameters for different computer vision tasks.

1 Camera Design as an End-to-End Optimization

A key insight in designing a custom camera system for a particular task is to consider the camera system as a whole. In general, a camera system is comprised of three parts: optical elements, a vision algorithm, and hardware architecture. Prior work tends to focus on optimizing a single component of this system. However, co-optimization of these components is critical because they interact with each other in ways that have non-trivial effects on latency, energy efficiency, and accuracy. Thus, by co-optimizing optics, algorithms and hardware, more efficient systems can be designed. We are thus able to formulate the design of intelligent camera systems as a co-optimization problem that tunes three groups of design parameters: 1) optical parameters, such as lens curvatures, distances between lenses and between the lens and sensor plane, and lens aperture, 2) algorithmic parameters, such as CNN weights and sparsity, and 3) hardware parameters, such as the number of MAC units and on-chip buffer sizes. This optimization framework can optimize for metrics of interest (e.g. accuracy), while meeting application constraints such as energy budgets and latency requirements.

WAX'19, June 22, 2019, Phoenix, AZ, USA
2019. ACM ISBN …$15.00
https://doi.org/

2 Modeling the Camera Pipeline

2.1 Modeling Optical Parameters

When considering models of optical parameters in the co-optimization pipeline, it is important to balance model simplicity with accuracy. Also, since we use a gradient descent-based approach to optimize optical parameters, it is critical that the model also be differentiable with respect to the optical parameters. We use point-spread functions (PSFs) to model the optical system because such a model has the desirable property of being differentiable with respect to optical parameters [1] Additionally, we simplify the model by assuming light hits the lens as a plane wave. According to Sitzmann et al. [1], approximating light as a plane wave introduces a small amount of error into the system, but the amount of error decreases as objects are farther away from the optical system. Even at close distances, the level of error is usually tolerable. We follow the formulation of optics presented by Sitzmann [1] for differentiable optical elements.

2.2 Modeling Sensor Noise

Since the sensor plane may not be completely accurate, it is necessary to model noise from the imaging process. Since this sensor noise must be incorporated into the pipeline, we choose to model sensor noise as a Gaussian to retain differentiability.

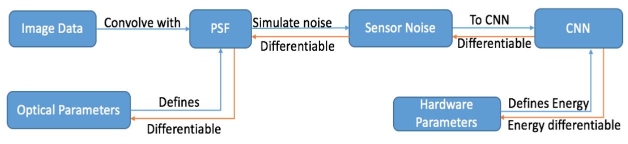

Figure 1. Diagram of the co-optimization pipeline

2.3 Modeling Hardware Architecture

Modeling hardware architecture allows for optimization of hardware parameters, such as on-chip buffer size or number of MAC units. When modeling hardware, it is important to know how hardware parameters affect the overall latency and energy-efficiency of the system. Previous work by Yang et al. shows that for a systolic array-based architecture, overall energy consumption of CNN inference can be modeled as a sum of the energy used for computation (through use of MAC units) and the energy used for data movement (moving intermediate values between DRAM, SRAM and buffer). In cases where CNN inference is being executed on other architectures, such as GPUs or CPUs, energy consumption can still be predicted and optimized using a combination of the Alternating Direction Method of Multipliers (ADMM) framework and gradient-based learning algorithms [2, 3]. As such, we are able to include energy consumption by hardware into the camera model, and optimize hardware parameters within a given energy budget.

2.4 Modeling the CNN

Since the CNN is in software, no additional work is needed to model the CNN. In addition, optimization of CNN parameters using gradient descent is a well-researched topic. As such, the models of the other components are integrated into the CNN optimization process.

2.5 Co-optimization Pipeline

The above sections have outlined a forward pass through the camera system. In the training process, we will be able to calculate domain-specific loss from the pipeline. Since each element of the pipeline is differentiable with respect to its parameters, we can employ gradient descent in order to tune the camera system's parameters based on the loss for a given example. Figure 1 presents a block diagram of this co-optimization pipeline, where given image data, we are able to complete a forward pass to the CNN through optics and sensor. We are then able to backpropagate to CNN, optics, and hardware parameters.

3 Self-Optimizing Camera

We propose a camera system that leverages the above co-optimization process to reconfigure itself for a given computer vision task. When given a different task, the system can re-execute the optimization process for the given task.

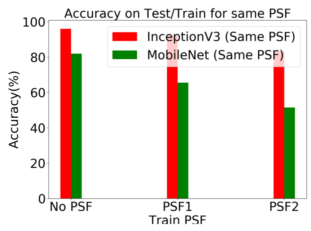

Figure 2. Accuracy comparison of CNNs that are trained/tested on the same PSF.

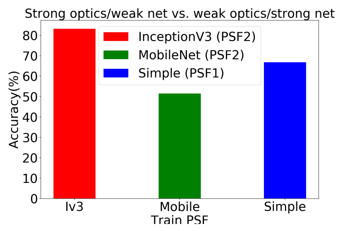

Figure 3. Comparison of stronger networks and weaker PSFs, and a weaker network and stronger PSF.

By doing so, it is able to determine and adjust to the optimal configuration within the system constraints. Key to this optimization process it the fact that many of the optimization parameters are dynamically adjustable. Some of these dynamically adjustable parameters include: the focal length and aperture size of the optical system; sparsity and quantization of the vision algorithm; and the on-chip buffer partitioning of the hardware architecture. By dynamically tuning these system parameters according to the application-specific co-optimization procedure, we are able to build a system that is able to optimize itself for its current task.

4 Prototype and Preliminary Results

To motivate the need for a co-optimization procedure that utilizes optical parameters, we ran an experiment that modeled various lens configurations, and forward-modeled image data as seen through the lens for an image classification task. We found that introducing a small amount of blur through the PSF of a potentially unoptimized lens can reduce accuracy by up to 20%. We find that much of the accuracy can be recovered through training of the CNN with image data convolved with the PSF. Figure 2 compares the performance of Inception V3 and MobileNet on an image classification task on a subset of Imagenet dataset. They are tested on several hand-picked optical front-ends of differing quality (PSF0, PSF1, PSF2, in descending order of quality). We can see that the accuracy of the CNN is significantly impacted by the optical front-end. In Figure 3, a simpler CNN is used to illustrate how having stronger optics can lead to higher accuracy even for eaker CNNs. We are in the process of manufacturing a prototype of the self-optimizing camera.

References

[1] Vincent Sitzmann, Steven Diamond, Yifan Peng, Xiong Dun, Stephen Boyd, Wolfgang Heidrich, Felix Heide, and Gordon Wetzstein. 2018. End-to-end optimization of optics and image processing for achromatic extended depth of field and super-resolution imaging. *ACM Transactions on Graphics (TOG)* 37, 4 (2018), 114.

[2] Haichuan Yang, Yuhao Zhu, and Ji Liu. 2019. ECC: Energy-Constrained Deep Neural Network Compression via a Bilinear Regression Model. *International Conference on Computer Vision and Pattern Recognition (CVPR)* (2019).

[3] Haichuan Yang, Yuhao Zhu, and Ji Liu. 2019. Energy-Constrained Compression for Deep Neural Networks via Weighted Sparse Projection and Layer Input Masking. *International Conference on Learning Representations (ICLR)* (2019).