Rethinking the Camera Pipeline for Computer Vision

Mark Buckler
Cornell University
mab598@cornell.edu

Suren Jayasuriya
Carnegie Mellon University
sjayasur@andrew.cmu.edu

Adrian Sampson
Cornell University
asampson@cs.cornell.edu

Abstract

Computer vision is undergoing a revolution that is enabling new categories of visual applications for consumers, but it still incurs costs that prevent energy-strapped mobile devices from deploying these new capabilities. Part of the problem is the camera system itself: smartphone cameras and their associated signal processing hardware are designed to capture high-quality images for human consumption, not to efficiently feed a computer vision algorithm.

In ongoing work, we propose to redesign the imaging systems in mobile devices from the ground up for computer vision. The key finding is that much of the work that makes up a traditional camera pipeline is unnecessary when feeding a vision algorithm. We propose a programmable design for a camera sensor and the associated image signal processor (ISP) chip that can switch into an efficient vision mode, which delivers raw signal data instead of high-quality photographs. Using experiments with both classical vision algorithms and deep convolutional neural networks, we find significant energy savings are possible with only minor degradation in vision accuracy: where humans see low-quality photographs, computer vision algorithms see usable data.

1. Overview

The deep learning revolution has unlocked a new set of capabilities in computer vision. To help bring the energy cost of these new techniques within the battery budget of a smartphone, a torrent of recent research has explored hardware acceleration for machine learning inference [3–5, 9, 13, 17]. That work, however, only addresses part of the whole cost: real-time vision involves the entire imaging pipeline, from photons to classification. Figure 1a depicts the complete pipeline, which includes the camera itself and its associated image signal processor (ISP) chip. These components can consume a significant portion of a smartphone's energy budget [2].

We argue that today's cameras and ISPs waste work when they are used for computer vision. They spend time and energy to capture a high-quality image for human viewers, but vision algorithms are sensitive to different image properties compared to human eyes. We envision a second, approximate mode for the imaging pipeline that produces low-quality, raw sensor signals in exchange for energy savings. Figure 1b depicts the proposed pipeline and its approximate mode, where the camera adjusts to produce lower-resolution, lower-precision data, and the ISP is disabled entirely.

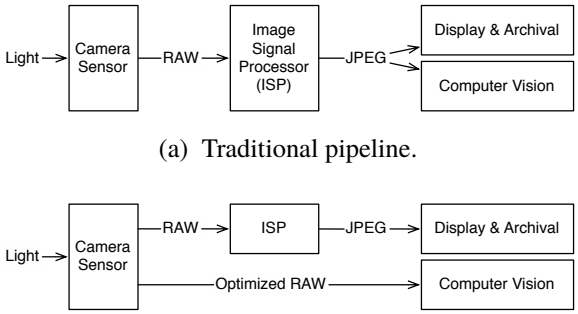

Figure 1. In current systems, the camera pipeline produces a high-quality photograph regardless of whether it is destined for human viewing or for computer vision. We propose a second mode that uses low-quality settings in the camera sensor and bypasses the ISP altogether to cheaply produce the inputs to vision algorithms.

In this position paper, we argue empirically that vision applications are highly tolerant to approximate image capture. We measure minimal degradation in accuracy when disabling most stages in a simulated ISP pipeline. Similarly, vision algorithms are only minimally impacted when the camera drastically reduces the number of pixels and the number of bits per pixel that it captures. Together, these tolerances highlight the need for a low-power, approximate, vision-oriented mode for camera pipelines.

2. Measuring Tolerance in Vision Algorithms

To examine the tolerance that vision algorithms have to approximate image capture, we use a suite of benchmarks and an array of tools for simulating our proposed camera pipeline. Table 1 lists the applications we measured, which include both deep convolutional neural networks and "classical" algorithms with hand-tuned features.

For learning-based algorithms, a central challenge is *retraining* the model to work on the raw, unprocessed output

Algorithm	CNN	Vision Task
3 Deep LeNet [12]	Yes	Object Classification
20 Deep ResNet [6]	Yes	Object Classification
44 Deep ResNet	Yes	Object Classification
FasterRCNN [18]	Yes	Object Detection
OpenFace [1]	Yes	Face Identification
OpenCV [8] Farneback	No	Optical Flow
OpenCV SGBM	No	Stereo Matching
OpenMVG [16] SfM	No	Structure from Motion

Table 1. Vision applications used in our experiments.

(a) Original. (b) Approx. RAW. (c) Detail.

Figure 2. An image from the COCO data set and its approximate RAW version produced by our conversion tool.

signal from our proposed hardware instead of conventional RGB images. Deep neural networks require large training sets, but no labeled training data exists in the form of raw camera signals. We need a tool that can convert popular image data sets such as COCO [14], LFW [7], and CIFAR [11] from plain images to raw sensor signals.

We have developed a pipeline that can perform these conversions by simulating a traditional camera pipeline in reverse. The tool is based on work in the imaging community on inferring RAW sensor data based on a particular camera profile [10]. To evaluate a specific camera pipeline proposal, the tool can simulate arbitrary ISP stages and sensor settings. Figure 2 shows the result of running the tool on an image from the COCO [14] data set to approximate the RAW data produced by an image sensor. We use the conversion tool to create training data for learning-based vision algorithms and to evaluate accuracy under different pipeline configurations.

3. Skipping the ISP

Modern mobile devices couple the camera with a specialized image signal processor (ISP), which transforms the raw camera signal into a final image. ISPs compose a series of stages: demosaicing to account for sensor device layout, denoising, white balance correction, and so on [15].

Only two of the standard stages have a large impact on the accuracy of any of the benchmarks in Table 1: demosaicing and gamma compression. With only these two stages enabled, the benchmarks all show only small quality degradation with respect to a full ISP: LeNet's [12] classification error increases by 1.3%, for example, and the FasterRCNN face detection network [18] loses 1.2% accuracy.

Accuracy is worse when these two stages are disabled. Demosaicing is critical because it compensates for the green-dominated Bayer pattern produced by the camera sensor (see Figure 2c) to a plain RGB image; and gamma compression normalizes the data so it has a wider dynamic range. Instead of enabling these two ISP stages, we are exploring a camera sensor design that can approximate both of them without signal processing. The first technique downsamples the image to read out a lower-resolution RGB image, eliminating the need for costly demosaicing. The second adjusts the quantization in the camera's analog-to-digital converters (ADC) to produce pre-normalized raw data in most cases. With these two features enabled in the camera itself, vision algorithms can achieve acceptable accuracy while skipping the ISP entirely.

4. Resolution and Quantization in the Camera

To trade off quality for energy in the camera sensor itself, we propose to adjust two parameters that are traditionally fixed at design time: image resolution and analog-to-digital converter (ADC) quantization.

Many vision algorithms use low-resolution inputs when compared to the multi-megapixel outputs from modern cameras. Current systems capture a high-resolution image and then scale it down for efficient vision processing. Instead, our proposed camera can save energy by reading out a lower-resolution image *a priori*. The camera powers off subsets of the CMOS photodiodes and ADCs that make up the pixel sensors. Reducing resolution saves an amount of energy linear in the pixel count.

The ADC in each sensor cell is responsible for quantizing an analog luminance signal to a digital set of levels. An ADC's time and energy cost is exponential in its number of bits. Although most cameras configure their ADCs to produce 12 or more bits, we find that most vision applications can achieve high accuracy on data with only 5 bits per pixel. Furthermore, by adjusting the ADCs' quantization levels to match the distribution of light intensities found in real images, we observe acceptable accuracy with as few as 4 bits.

5. Next Steps

This position paper explores simple changes to a traditional camera pipeline to make it better suited for computer vision. There are many more opportunities for co-designing camera hardware more deeply with computer vision algorithms. For example, we are exploring the creation of a dynamic feedback loop between the vision application and the sensor. In this design, the application uses inference results from frame n to determine how to best capture frame $n + 1$. The vision algorithm will predict which portions of the next image will be irrelevant, for example, or whether a higher dynamic range is necessary to capture more detail in an object of interest.

References

[1] Brandon Amos, Bartosz Ludwiczuk, and Mahadev Satyanarayanan. OpenFace: A general-purpose face recognition library with mobile applications. Technical Report CMU-CS-16-118, CMU School of Computer Science, 2016.

[2] Xiang Chen, Yiran Chen, Zhan Ma, and Felix C. A. Fernandes. How is energy consumed in smartphone display applications? In *HotMobile*, 2013.

[3] Yu-Hsin Chen, Tushar Krishna, Joel Emer, and Vivienne Sze. Eyeriss: An energy-efficient reconfigurable accelerator for deep convolutional neural networks. In *IEEE International Solid-State Circuits Conference (ISSCC)*, 2016.

[4] Zidong Du, Robert Fasthuber, Tianshi Chen, Paolo Ienne, Ling Li, Tao Luo, Xiaobing Feng, Yunji Chen, and Olivier Temam. ShiDianNao: Shifting vision processing closer to the sensor. In *International Symposium on Computer Architecture (ISCA)*, 2015.

[5] Song Han, Xingyu Liu, Huizi Mao, Jing Pu, Ardavan Pedram, Mark A. Horowitz, and William J. Dally. EIE: Efficient inference engine on compressed deep neural network. In *International Symposium on Computer Architecture (ISCA)*, 2016.

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015.

[7] Gary B. Huang, Manu Ramesh, Tamara Berg, and Erik Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical Report 07-49, University of Massachusetts, Amherst, October 2007.

[8] Itseez. OpenCV. `http://opencv.org`.

[9] Norm Jouppi. Google supercharges machine learning tasks with TPU custom chip. `https://cloudplatform.googleblog.com/2016/05/Google-supercharges-machine-learning-tasks-with-custom-chip.html`.

[10] S. J. Kim, H. T. Lin, Z. Lu, S. Süsstrunk, S. Lin, and M. S. Brown. A new in-camera imaging model for color computer vision and its application. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34 (12), 2012.

[11] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

[12] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86 (11): 2278–2324, 1998.

[13] Robert LiKamWa, Yunhui Hou, Julian Gao, Mia Polansky, and Lin Zhong. RedEye: Analog ConvNet image sensor architecture for continuous mobile vision. In *International Symposium on Computer Architecture (ISCA)*, 2016.

[14] Tsung-Yi Lin, Michael Maire, Serge J. Belongie, Lubomir D. Bourdev, Ross B. Girshick, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. `http://arxiv.org/abs/1405.0312`.

[15] Z. Liu, T. Park, H. S. Park, and N. S. Kim. Ultra-low-power image signal processor for smart camera applications. *Electronics Letters*, 51 (22): 1778–1780, 2015.

[16] Pierre Moulon, Pascal Monasse, Renaud Marlet, and Others. OpenMVG: An open multiple view geometry library. `https://github.com/openMVG/openMVG`.

[17] B. Reagen, P. Whatmough, R. Adolf, S. Rama, H. Lee, S. K. Lee, J. M. Hernández-Lobato, G. Y. Wei, and D. Brooks. Minerva: Enabling low-power, highly-accurate deep neural network accelerators. In *International Symposium on Computer Architecture (ISCA)*, 2016.

[18] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems (NIPS)*, 2015.