ISP4ML: Understanding the Role of Image Signal Processing in Efficient Deep Learning Vision Systems

Patrick Hansen*  
Arm Research

Alexey Vilkin†  
Twitter

Yury Khrustalev*  
Arm

James Imber‡  
Imagination Technologies

David Hanwell*  
Arm

Matthew Mattina*  
Arm Research

Paul N. Whatmough*  
Arm Research

Abstract

Convolutional neural networks (CNNs) are now predominant components in a variety of computer vision (CV) systems. These systems typically include an image signal processor (ISP), even though the ISP is traditionally designed to produce images that look appealing to humans. In CV systems, it is not clear what the role of the ISP is, or if it is even required at all for accurate prediction. In this work, we investigate the efficacy of the ISP in CNN classification tasks, and outline the system-level trade-offs between prediction accuracy and computational cost. To do so, we build software models of a configurable ISP and an imaging sensor in order to train CNNs on ImageNet with a range of different ISP settings and functionality. Results on ImageNet show that an ISP improves accuracy by 4.6%-12.2% on MobileNet architectures of different widths. Results using ResNets demonstrate that these trends also generalize to deeper networks. An ablation study of the various processing stages in a typical ISP reveals that the tone mapper is the most significant stage when operating on high dynamic range (HDR) images, by providing 5.8% average accuracy improvement alone. Overall, the ISP benefits system efficiency because the memory and computational costs of the ISP is minimal compared to the cost of using a larger CNN to achieve the same accuracy.

1. Introduction

In recent years, deep convolutional neural networks (CNNs) have surpassed traditional algorithms on many computer vision (CV) tasks. In common CV applications such as advanced driver assistance systems (ADAS), the system captures images intended to be consumed solely by vision processing algorithms and generally not intended to be viewed by humans. Such systems present an interesting design opportunity since they are no longer constrained to generate images optimized for human viewing. This is in contrast to scenarios such as photography, where the design goal is to generate images that are subjectively aesthetically pleasing to the human eye, as well as requiring an 8-bit, gamma-corrected, RGB format to be properly output on a typical display. Without the need to conform to these traditional constraints, there is scope to improve efficiency and algorithmic performance of the system by relaxing or optimizing the image representation.

CNNs are typically trained on image datasets (such as ImageNet) whose images have been processed using an image signal processor (ISP) and stored in an RGB image representation. An ISP is a hardware component consisting of several pipelined processing stages designed to process raw Bayer color filter array (CFA) sensor data into RGB output images. Figure 1 depicts how an ISP is typically used in embedded CV. A camera lens focuses light onto a CFA image sensor, which produces a single plane of digital raw pixel values (Figure 1a). These raw images are then processed by an ISP to generate RGB images (Figure 1b), which are consumed either by a display for human viewing or by a CV algorithm for inference.

While optimization of the display path in Figure 1 is

\[ \text{first.last}@arm.com \]
\[ \text{previously with Arm. alexey.vilkin1@gmail.com} \]
\[ \text{previously with Arm. james.imerb@imgtec.com} \]

\[ \text{Perhaps with the exception of debugging and development purposes.} \]
a well studied problem, the role and (co-)optimization of the ISP for CNN inference is relatively unexplored. Traditionally, ISP algorithms are designed to optimize for metrics such as peak signal-to-noise-ratio (PSNR), modulation transfer function (MTF), and subjective human rating scores. However, these metrics clearly may not be optimal for CNN prediction accuracy. If we are able to reduce the complexity of the ISP by removing unneeded functions, that would lead to lower latency and lower energy consumption. Even though CNNs are exceptional in their ability to adapt to various input representations, co-optimization of the ISP and network architecture may also enable reduction in the size of the CNN model at the same prediction accuracy, again reducing latency and energy. Any complexity savings that can be made to the CNN would be especially potent, as CNNs are known for their very high computational cost, even when implemented on highly-efficient and high-throughput neural processing units (NPUs) \cite{17, 21, 3}.

In this paper, we seek to understand the impact of the ISP on CNN classification problems. However, this is not straightforward, because standard labeled datasets commonly used for CNN experiments consist of RGB images, which have already been processed from raw sensor data with fixed ISP settings. Collecting and labeling a new dataset of raw images for training is prohibitively expensive for a useful number of images. Our approach circumvents this problem by implementing a capture model, which simulates RAW images from RGB by modelling an imaging sensor, similar to the approach of Buckler et al. \cite{1}. We process simulated raw images using a configurable, industry-grade, high dynamic range (HDR) ISP to generate RGB images. This enables us to generate datasets with a variety of ISP configurations and settings, which we then use to train state-of-the-art CNN models – from compact width-scalable MobileNets \cite{12}, which are suited to mobile devices, to larger depth-scalable ResNets \cite{10}, which are employed in applications such as automotive – to formulate hypotheses on the impact of ISP design on CNN classification accuracy.

Our results confirm that there is indeed a benefit, especially marked for compact models, to incorporating an ISP in CNN inference. Furthermore, we determine what functions of the ISP are most significant to achieve accurate predictions. Finally, we present a system-level analysis of the cost trade-offs of different ISP configurations, and find that all considered use cases benefit from an ISP.

In summary, this paper makes the following contributions:

- **Raw vs RGB.** Using a capture model and an industry-grade ISP model, we process the ImageNet dataset to train and test CNN classification models with fully configurable ISP settings. Results show that full ISP-processed RGB provides an average 7.0% accuracy improvement over raw images for MobileNet models trained on ImageNet.

- **Impact of Model Size.** The accuracy improvement provided by ISP processing is most significant for smaller model sizes, suggesting larger models are more readily able to learn the functionality of the ISP. Experiments with much larger models, ResNet-50 and ResNet-101, further support this claim.

- **ISP Stage Ablation.** An ablation study on the individual stages of the ISP reveals that the tone mapper has the greatest impact on performance for models trained on HDR images, providing an average of 5.8% accuracy improvement alone. Results indicate this is due to the ineffectiveness of CNNs learning from heavily uneven pixel distributions.

- **Compute and Memory Analysis.** A computation-cost analysis shows that utilizing an ISP improves system efficiency compared to a baseline with no image processing. This result can be attributed to the negligible cost of the ISP in comparison to CNNs.

The remainder of the paper is organized as follows: Section 2 describes related work, Section 3 gives background and methodology. Results are presented in Section 4 and Section 5 investigates the impact of tone mapping. Section 6 evaluates trade-offs in accuracy and compute. Finally, Section 7 concludes.
2. Related Work

**ISP Optimization for ML Systems.** Recent research and commercial products have begun exploring ISP specialization for vision systems. For example, the Arm Mali-C71 [18] is an ISP designed specifically for use in smart automotive systems, with different functional modes for display and computer vision. Liu et al. [23] proposed an ISP that can selectively disable stages based on application needs. Lubana et al. [19] proposed a minimal in-sensor accelerator to enable inference directly on sensor outputs without CNN retraining. Buckler et al. [1] investigated the impact of ISP stages on different CV algorithms, and proposed a sensor design with adjustable precision and subsampling. Our methodology is inspired by Buckler et al. [1], involving an imaging sensor model to simulate raw images and a model of a commercial-grade, high dynamic range (HDR) ISP. In contrast to Buckler et al. [1], we validated our imaging sensor model using a dataset of raw images that we captured specifically for our experiments. To the best of our knowledge, we are the first to understand the impact of the ISP on training CNNs for ImageNet-size problems, using both compact MobileNet models and deep ResNet models.

**Neural ISP Algorithms.** CNN models have been developed for denoising [30], demosaicing [9, 26], and end-to-end imaging solutions [24, 33]. These works demonstrate that CNNs are capable of replicating ISP functions, and even exceeding them in terms of PSNR. However, this improved performance comes at a significant compute cost, on the order of 1M operations/pixel [24], compared to around 1K ops/pixel for a traditional ISP. Diamond et al. [7] investigates the impact of image distortions on classification accuracy, and propose a network architecture for joint deblurring, denoising, and classification. Our work differs from previous works in that we do not attempt to emulate ISP functions with neural algorithms, but rather explore the need for ISP functions in the first place.

**Efficient CNN Inference.** CNNs are computationally expensive, so it is helpful to start with efficient hardware and an efficient network architecture before attempting to optimize the ISP for system efficiency. Recent research has focused on developing dedicated hardware for efficient deep learning inference [17, 21, 3]. Additionally, much effort has gone into developing efficient CNN architectures [12, 11, 28, 13, 32] and algorithms for automatically generating efficient CNN architectures [2, 8, 16, 27]. In this work, we focus on MobileNets [12], a family of CNNs designed for energy-efficient inference. These models employ depthwise-separable convolutions [4] to reduce the number of parameters and operations in each layer. State-of-the-art models in both the energy-efficient [11] and unconstrained [23] domains iterate upon the basic MobileNet structure.

![Figure 3: Key components of the ISP and capture models.](image)

3. Background and Methodology

To evaluate the impact of the ISP on CV systems, we train ML models on images that have been processed by a variety of ISP configurations (including no ISP at all). This approach enables us to study the impact of each stage of the ISP on prediction accuracy on a large dataset (like ImageNet). This section describes the ISP we model, our approach in generating images for training/evaluation, and the ML benchmarks we use for evaluation.

3.1. ISP Overview

For our experiments, we model the ISP stages shown in Figure 3a: denoise, black level subtraction, white balance, tone-mapping, demosaic, color correction, and gamma correction. We implement each of these stages in a C++ software model, using production-quality algorithms. The ISP software model allows stages to be optionally enabled or disabled. This allows us to test a variety of ISP configurations in the ISP design space in relation to CNN configurations.

---

2 Limited to valid combinations of ISP stages to ensure compatible data types and image representations.
3.2. Simulated Raw Data

In order to evaluate the impact of ISP design decisions on CNN model performance, we require a large quantity of labelled data. However, datasets relevant to CNN tasks are nearly exclusively captured in 8-bit RGB format. In other words, these images have already been processed by an ISP, without storing the original raw image data. Even worse, these images are typically captured using a range of different (unknown) sensors and ISPs. Although some raw image datasets do exist, they are typically intended for research on ISP demosaicing and denoising algorithms and are not suitable for training modern CNNs, as they are too small and unlabeled.

As such, we simulate raw HDR captures by processing standard RGB image datasets using a software model which we refer to as the capture model. This model can be thought of as a “reverse ISP” in the sense that it generates raw images from RGB images; however the goal of the capture model is to generate images with statistics representative of raw images, rather than to perfectly recreate an original raw image. The capture model seeks to “undo” the ISP processing that has already been applied to these images when they were originally captured. It accomplishes this by performing the inverse (or approximate inverse) of each stage in the ISP pipeline in reverse order. Figure 3 depicts the approach. Characteristic parameters for colour confusion, colour imbalance, black level and noise are all calibrated on actual sensor measurements from a Sony IMX290.

We simulate HDR captures by applying 3 different gains to an RGB image (Figure 4) after linearization to simulate different exposure times, resulting in 3 different RGB images of the same scene. The resulting images are independently processed via the capture model, and then merged to simulate a single simulated HDR raw image (Figure 5). Images generated by the capture model can be subsequently processed by any ISP configuration to generate an RGB image dataset (Figure 6) to use for training CNNs. This approach enables us to efficiently explore the impact of the ISP design space on CNN classification performance, without the huge expense of manually capturing and labelling a sufficiently large image dataset. We validate this approach by testing resultant CNN models on true raw images and observing consistent trends in prediction accuracy.

3.3. Benchmarks

We use MobileNet [12] and ResNet [10] CNN architectures trained on ImageNet [6] to evaluate the impact of the ISP. MobileNet is designed to be a parameter-efficient, scalable CNN architecture for image classification on mobile devices, with hyperparameters to scale input resolution and the width of each layer. Sweeping these parameters produces a family of models, which are collectively known as MobileNets. MobileNets are designed for compute-constrained devices, however derivative network architectures have demonstrated SOTA results in other application domains as well [25, 28]. We use this model architecture for our investigations because MobileNet-like models are highly relevant in the field today, and because its scalability enables us to test on a range of model sizes. To demonstrate the generalization of our results to deeper models, we also train ResNets [10] on ImageNet.

In our experiments, we train each network with the same training parameters as used in their original publication [12, 10]. The only exceptions to this are: (1) batch normalization decay, which we set to 0.99 for models trained on certain image sets, and (2) data augmentation, which we limit to mean and variance normalization for fair comparison between different image representations. The original training recipes for these networks are carefully tuned for RGB images, so we made sure to tune our training recipe for raw images, as described in the Appendix.

In addition to the training recipes being optimized for RGB images, we acknowledge that the model architectures themselves are tuned for RGB images. It is possible that

---

3Not all ISP stages are perfectly invertible, viz. denoise, white balance, and tone mapping. The original sensor noise cannot be perfectly recreated from a denoised image. Similarly, white balance and tone mapping use original image statistics that cannot be recovered after-the-fact.

4We train MobileNets on 224x224 images, and sweep the width multiplier across [0.25, 0.50, 0.75, 1.00].

Random cropping and flipping cannot be applied to raw images because they do not preserve the CFA mosaic pattern.
there are architectures better suited for RAW images, which is a non-trivial question we leave as future work.

4. Evaluation of ISP Stages

In this section, we discuss the results of CNN training experiments using images generated by different ISP configurations. We consider ISP configurations with all stages enabled, no stages enabled, and stages selectively enabled/disabled.

4.1. Raw vs. RGB Experiment

We first evaluate whether the ISP has any impact on CNN classification accuracy. To accomplish this, we compare the performance of models trained on raw images (no ISP processing) against those trained on RGB images (full ISP processing). We train MobileNets on two processed versions of ImageNet: the first contains simulated raw images generated by processing ImageNet with the capture model (Section 3.2). The second contains RGB images generated by applying our ISP software model (Section 3.1) to the simulated raw images. Figure 4 provides the results of this experiment, and compares the accuracy of these models to the published MobileNet results [12]. This figure demonstrates that the models trained on RGB images outperform those trained on raw images, by an average of 7.0%.

Note that the accuracies of our Simulated RGB models are lower than published results (Original RGB). This is expected because the capture model cannot perfectly reproduce the original raw images that produced the ImageNet images – the original color and noise profiles of the original raw images are unknown, and the capture model throws away data by remosaicing and adds noise. Additionally, as noted previously, data augmentation techniques such as random cropping and flipping cannot be readily applied to raw data, so we forego these in order to make a fair comparison between raw and RGB training experiments (Section 3.3).

We notice that the accuracy difference is larger for compact models, which may indicate that models trained on raw images are more capacity-bound than those trained on RGB. Since even the largest MobileNet width is a relatively small model, we also repeated this experiment on two sizes of ResNets [10], to understand the impact of raw vs RGB on larger and deeper models. Similar to MobileNet, ResNet results (Table 1) show a benefit to training models on RGB images, however the gap in accuracy is smaller than was seen for MobileNet (c.f. Figure 5). This suggests that larger (or deeper) models, such as ResNets, are more readily able to learn from image formats other than RGB.

4.2. Ablation Study

Having demonstrated a benefit to using an ISP, we aim to understand which components of the ISP contribute most to improved classification accuracy. To this end, we train MobileNets on images generated with different stages of the ISP selectively enabled. Because of the prevalence of research into the impact of noise on neural networks, the following ablation study is separated into (1) a study of only denoise and (2) a study of remaining ISP components.

The denoising stage of an ISP is designed to reduce the measurement noise that exists in a raw image due to physical characteristics of the sensor. Recent research has demonstrated the susceptibility of CNNs to mis-classifying images with imperceptible adversarial perturbations applied to the input [22]. Further, Diamond et al. [7] demonstrate poor compatibility between traditional denoising algorithms and CNNs inference. Our results (Figure 6a) agree with this conclusion; we observe an average accuracy improvement provided by denoising of 0.27% and 0.09% (within the bounds of model variance) for raw images and RGB images, respectively. In several cases, models trained on denoised images even performed worse than their noisy counterparts. A potential explanation for this result is that the denoiser, while removing noise, also tends to blur fine detail which may have otherwise been useful to the CNN. These results indicate negligible impact of traditional denoising on classification accuracy and questionable value for including a denoiser in a CV system.

| Model       | Params | Image type | Top-1 acc. | Top-5 acc. |
|-------------|--------|------------|------------|------------|
| ResNet-50   | 25.6M  | RAW        | 67.45      | 88.04      |
|             |        | RGB        | 71.01      | 91.82      |
| ResNet-101  | 44.5M  | RAW        | 70.07      | 89.90      |
|             |        | RGB        | 72.34      | 92.93      |

Table 1: Raw vs. RGB (both simulated by capture model) results for large models: ResNet-50 and ResNet-101.
To evaluate the remaining ISP stages, we compare the performance of ISP configurations with incrementally more stages added until arriving at a full ISP pipeline\(^6\). We chose two configurations to evaluate and compare classification performance: (1) black level subtraction + tone mapping and (2) black level subtraction + white balance + tone mapping. Figure 6 shows the result of models trained on data processed with these ISP configurations, along with baselines (models trained on raw images and full RGB images). For each stage added, we see some improvement in accuracy across all models. This indicates that each of these ISP stages contributes to improved CNN performance. The most significant improvement in accuracy (5.8% improvement) is between models trained on raw images and models trained on images processed using black level subtraction and tone mapping. This improvement in accuracy should be attributed to tone mapping because black level subtraction merely shifts the input pixel values by a scalar, whereas tone mapping has a significant impact on the statistics of input images. We explore this phenomenon in greater detail in Section 5.

4.3. Capture Model Validation

Lubana et al.\(^19\) have criticized the use of simulated raw data for training CNN models. To validate generalization of our results to true raw images, we use real raw image captures to test CNN models previously trained on simulated raw images. Since the capture model cannot perfectly reverse the ISP processing, it cannot be validated by simply comparing a raw image to the result of running that image through our ISP model and then the capture model. Instead, we want to show that the capture model does not introduce any artifacts that negatively impact CNN prediction accuracy. By demonstrating similar trends in accuracy between testing on real and simulated raw data, we hope to dispel uncertainty in the generalization of our results. We created a dataset of real raw images, which we used to validate the results we achieved with simulated raw images. This dataset contains 4000 raw images of 50 objects, taken with a variety of lighting conditions and exposure settings. The images are captured using an IMX290 sensor\(^25\). Running inference on our trained models using this lab captured validation set provides results that broadly follow the trends of Figures 4 and 6 supporting our conclusions. These results are discussed in further detail in the Appendix.

5. Input Distribution Investigation

Having identified tone mapping as a significant component for achieving high CNN accuracy, we next investigate the effect the tone mapper has on the statistics of input images, and how that influences classification performance.

5.1. Histogram Analysis

The tone mapper we implement in our software ISP model performs localized histogram equalization\(^7\) to pixel values. Therefore, tone mapping tends to result in images that better occupy the full dynamic range, as seen in Figure 7.

---

\(^6\)Informed by expert knowledge, we restrict the space of possible ISP configurations. For example, tone mapping should only be performed after black level subtraction so that the black level offset is not scaled with light intensity.

\(^7\)Our tone mapper compensates for gamma correction later in the ISP pipeline, resulting in a higher concentration of values toward zero.
Figure 7: Pixel value distributions (by color channel) at three points along the ISP pipeline. Sampled data is comprised of 25 random images from each class in the ImageNet training set.

| Distribution     | Skew  | Kurtosis |
|------------------|-------|----------|
| ImageNet-raw     | 7.85  | 95.80    |
| ImageNet-bl-wb-tm| 0.99  | 0.25     |
| ImageNet-rgb     | 0.64  | -0.54    |

Table 2: Skew and kurtosis of ImageNet dataset with different processing.

which illustrates the histograms of ImageNet data at three points in the image processing pipeline.

Figure 7a shows the distribution of simulated raw pixel values; this distribution vastly differs from that of RGB images (Figure 7c). The raw distribution is heavily concentrated near the black level value of the sensor, and trails off in density very rapidly towards larger values. The non-uniformity of the raw distribution is exacerbated due to the HDR simulation performed in the capture model, which increases the overall contrast in the image. In comparison, the RGB distribution is relatively uniform along its full range [0,1]. More formally, we consider the skew and kurtosis of these distributions. Skew describes a distribution’s lack of symmetry. Kurtosis is the degree of peakedness (or flatness) of a distribution. The raw distribution has much higher skew and kurtosis than RGB images (Table 2). Tone mapping does much of the work in transforming the distribution of raw pixel values to that of RGB, as can be seen in Figure 7b. The skew and kurtosis of this distribution are quite close to that of RGB images when compared against raw images.

5.2. Skewed Data Experiment

We hypothesize that the difference in image distributions we observe between raw and RGB images (attributed to the tone mapper) results in improved test accuracy for models trained with RGB data (lower skew and kurtosis). Lubana et al. [19] showed that a difference in image pixel distributions between training and testing may cause a drop in test accuracy. However, in this work, we consider the impact the pixel distributions constant between training and test time, but different across experiments. To explore this, we trained CNN models using image data with a range of skew and kurtosis values. We produce such distributions by applying a pixel-wise transform of \( f(x) = x^n \) to input images, where each pixel of an image, \( x \), is normalized to the range \([0,1]\), and \( n \) controls the degree of skew/kurtosis. This transformation concentrates data toward low values and creates a distribution resembling that of raw images.

We trained ResNet-18 [10] on CIFAR-10 [14] using this transformation on both the training and test images. This combination of model and dataset enables us to reasonably train many models to build confidence in our results. We sweep \( n \) across the range \([1,10]\) with increment 0.5 and correspondingly generate data with which to train ResNet-18.
The results of this experiment (Figure 8) suggest that the distribution of the input image can considerably impact accuracy. Image distributions with high skew result in trained model accuracy with considerable accuracy reduction relative to baseline \((n = 1)\). At \(n = 10\), there is an average accuracy degradation of 3.70%. Note that no information is being destroyed via this transform, and the original image can be perfectly reconstructed using the transform \(f^{-1}(x) = \sqrt{x}\). Therefore, it is only the representation of the input data that seemingly causes performance degradation. Even with such a drastic change to the input distribution, the skew and kurtosis of the input data in this experiment (skew = 3.67, kurtosis 13.28) are much lower than that of our simulated raw ImageNet data (skew = 7.85, kurtosis = 95.80). Therefore, it is not surprising that using tone mapping to normalize the input data distribution results in models with an average of 5.77% higher accuracy than those trained on raw data.

### 6. System Cost Analysis

The results in Section 4 demonstrate that most components of an ISP have a beneficial impact on CNN performance to a greater or lesser extent. However, from a systems perspective, it is important to understand how the compute complexity of these components compares with the accuracy improvement. Hence, we analyzed the trade-off between accuracy and computational cost. We measure this cost in terms of both arithmetic operations per inference and total size of memory accessed per inference. We calculate the number of operations based on our software implementation of each ISP component and by the number of multiply and accumulate operations required by MobileNets.

For memory, we assume the worst case scenario for the ISP— that the ISP must read raw images from DRAM rather than streaming raw images directly from the image sensor.

Figure 9 illustrates the trade-offs between cost and accuracy for different ISP configurations and MobileNet widths. The relative improvement in accuracy provided by a minimally-viable ISP (BL + WB + TM) far outweighs the additional algorithmic complexity associated with that ISP. Adding more stages to the ISP pipeline provides both marginal increases in accuracy and marginal computational costs (ISP cost independent of the CNN model cost). However, the cost of even a full ISP is minuscule in comparison to any relevant CNN architecture, so the improvements in accuracy prove worthwhile. At large compute budgets, it is clear that a full ISP pipeline provides enough benefit to warrant its computational cost. By utilizing an ISP, even systems with low memory budgets benefit require fewer operations per inference (thus, lower latency and higher throughput). Overall, these results show clear evidence that including an ISP significantly improves the cost/accuracy pareto trade-off.

### 7. Conclusion

In this paper, we empirically studied the impact of image signal processing (ISP) on CNN prediction accuracy. This is performed using a software model of an ISP and a model of an imaging sensor to enable the study of relevant application domains. We validate this approach by comparing training results from simulated raw images against raw images captured in-lab. We found that processing images with an ISP improves accuracy by an average of 7.0% for a chosen set of MobileNets. Our results indicate that the ISP has a more significant impact on smaller CNN models, and our results on ResNet-50 and ResNet-101 are consistent with this trend. Each component of the ISP pipeline provides accuracy gains across all models, except for denoise, which is found to have questionable benefit to CNN performance. Tone mapping, which equalizes pixel value distributions in our implementation, provides a dominant 5.8% accuracy improvement. We also show empirical evidence that uneven pixel distributions result in degraded classification performance. Finally, the ISP benefits system efficiency because the algorithmic cost is significantly lower than the cost of using a larger CNN to achieve the same accuracy.

---

8 Ignoring minimal numerical loss due to floating-point arithmetic.

9 Operations are counted as multiply, add, and simple transcendental functions.
References

[1] Mark Buckler, Suren Jayasuriya, and Adrian Sampson. Reconfiguring the imaging pipeline for computer vision. In IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017, pages 975–984, 2017.

[2] Han Cai, Ligeng Zhu, and Song Han. Proxylessnas: Direct neural architecture search on target task and hardware. In 7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019.

[3] Yu-Hsin Chen, Tien-Ju Yang, Joel S. Emer, and Vivienne Sze. Eyriiss v2: A flexible accelerator for emerging deep neural networks on mobile devices. IEEE J. Emerg. Sel. Topics Circuits Syst., 9(2):292–308, 2019.

[4] François Chollet. Xception: Deep learning with depthwise separable convolutions. In 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017, pages 1800–1807, 2017.

[5] Duc-Tien Dang-Nguyen, Cecilia Pasquini, Valentina Conotter, and Giulia Boato. Raise: A raw images dataset for digital image forensics. In Proceedings of the 6th ACM Multimedia Systems Conference, MM Sys ’15, pages 219–224, New York, NY, USA, 2015. ACM.

[6] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In CVPR09, 2009.

[7] Steven Diamond, Vincent Sitzmann, Stephen P. Boyd, Gordon Wetzstein, and Felix Heide. Dirty pixels: Optimizing image classification architectures for raw sensor data. CoRR, abs/1701.06487, 2017.

[8] Igor Fedorov, Ryan P. Adams, Matthew Mattina, and Paul N. Whatmough. Sparse: Sparse architecture search for cnns on resource-constrained microcontrollers. CoRR, abs/1905.12107, 2019.

[9] Michaël Gharbi, Gaurav Chaurasia, Sylvain Paris, and Frédéric Durand. Deep joint demosaicing and denoising. ACM Trans. Graph., 35(6):191:1–191:12, Nov. 2016.

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016, pages 770–778, 2016.

[11] Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, Quoc V. Le, and Hartwig Adam. Searching for mobilenetv3. CoRR, abs/1905.02244, 2019.

[12] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Wey, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. CoRR, abs/1704.04861, 2017.

[13] Forrest N. Iandola, Matthew W. Moskewicz, Khalid Ashraf, Song Han, William J. Dally, and Kurt Keutzer. SqueezeNet: Alexnet-level accuracy with 50x fewer parameters and <1mb model size. CoRR, abs/1602.07360, 2016.

[14] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009.

[15] Robert LiKamWa, Yunhui Hou, Yuan Gao, Mia Polansky, and Lin Zhong. Redeye: Analog convnet image sensor architecture for continuous mobile vision. In 43rd ACM/IEEE Annual International Symposium on Computer Architecture, ISCA 2016, Seoul, South Korea, June 18-22, 2016, pages 255–266, 2016.

[16] Hanxiao Liu, Karen Simonyan, and Yiming Yang. DARTS: differentiable architecture search. In 7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019.

[17] Arm Ltd. Arm Machine Learning Processor. https://developer.arm.com/products/processors/machine-learning/arm-ml-processor Accessed: 2019-08-30.

[18] Arm Ltd. Mali-c71. https://www.arm.com/products/silicon-ip-multimedia/image-signal-processor/mali-c71 Accessed: 2019-08-30.

[19] Ekedubbana, Robert Dick, Vinayak Aggarwal, and Pyari Mohan Pradhan. Minimalistic image signal processing for deep learning applications. pages 4165–4169, 09 2019.

[20] Nicola Massari, Massimo Gottardi, Lorenzo Gonzo, David Stoppa, and Andrea Simoni. A CMOS image sensor with programmable pixel-level analog processing. IEEE Trans. Neural Networks, 16(6):1673–1684, 2005.

[21] NV DLA. Nvidia Deep Learning Accelerator (NVDLA).

[22] Nicolas Papernot, Patrick D. McDaniel, Xi Wu, Somesh Jha, and Ananthram Swami. Distillation as a defense to adversarial perturbations against deep neural networks. In IEEE Symposium on Security and Privacy, SP 2016, San Jose, CA, USA, May 22-26, 2016, pages 582–597, 2016.

[23] T Park, Z Liu, N.S. Kim, and Hyun-Sang Park. Ultra-low-power image signal processor for smart camera applications. Electronics Letters, 51, 10 2015.

[24] Eli Schwartz, Raja Giryes, and Alexander M. Bronstein. Deeisp: Toward learning an end-to-end image processing pipeline. IEEE Trans. Image Processing, 28(2):912–923, 2019.

[25] Sony. IM X290LQR-C: Diagonal 6.46 mm (Type 1/2.8) CMOS Solid-state Image Sensor with Square Pixel for Color Cameras.

[26] Nai-Sheng Syu, Yu-Sheng Chen, and Yung-Yu Chuang. Learning deep convolutional networks for demosaicing. CoRR, abs/1802.03769, 2018.

[27] Mingxing Tan, Bo Chen, Ruoming Pang, Vijay Vasudevan, Mark Sandler, Andrew Howard, and Quoc V. Le. Mnasnet: Platform-aware neural architecture search for mobile. In IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019, pages 2820–2828, 2019.

[28] Mingxing Tan and Quoc V. Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA, pages 6105–6114, 2019.
[29] Paul N. Whatmough, Chuteng Zhou, Patrick Hansen, Shreyas K. Venkataramanaiah, Jae-sun Seo, and Matthew Mattina. Fixynn: Efficient hardware for mobile computer vision via transfer learning. In Proceedings of the 2nd SysML Conference 2019, 31 March-2 April, Stanford, California, USA, 2019.

[30] Kai Zhang, Wangmeng Zuo, Yunjin Chen, Deyu Meng, and Lei Zhang. Beyond a gaussian denoiser: Residual learning of deep CNN for image denoising. IEEE Trans. Image Processing, 26(7):3142–3155, 2017.

[31] Lei Zhang, Xiaolin Wu, Antoni Buades, and Xin Li. Color demosaicking by local directional interpolation and nonlocal adaptive thresholding. J. Electronic Imaging, 20(2):023016, 2011.

[32] Xiangyu Zhang, Xinyu Zhou, Mengxiao Lin, and Jian Sun. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In 2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018, pages 6848–6856, 2018.

[33] Ruofan Zhou, Radhakrishna Achanta, and Sabine Süsstrunk. Deep residual network for joint demosaicing and super-resolution. CoRR, abs/1802.06573, 2018.
A. Raw Image Representation

The mapping of mosaic images to inputs of traditional CNNs is not obvious because mosaic images have shape $[H, W, 1]$ whereas the expected input to a CNN has shape $[H, W, 3]$. We considered three different representations of mosaic images to be used for training CNNs:

1. Keep pixels on a single plane (shape = $[H, W, 1]$)

2. Stretch pixels to 3 channels based on color, and insert zeros for missing color information (shape = $[H, W, 3]$)

3. Stretch pixels to 4 channels based on Bayer pattern (R, Gr, Gb, B), and do not insert zeros (shape = $[H/2, W/2, 4]$)

Each of these representations can be used as inputs to MobileNets and ResNets with little or no modification to the first convolutional layer. Representation 1 requires the first convolution layer to have 1 input feature map. Representation 3 requires the first convolution layer to have 4 input feature maps and a stride of 1 (so that the output activations have the same shape as the baseline model). Representations 1 and 2 have information of the Bayer pattern embedded in 2x2 spatial grids, but luckily the first layer in each network has a stride of 2 so each weight corresponds to an individual color in the Bayer pattern.

We trained 3 copies of MobileNet on raw data using each of the representations, and found negligible difference in performance amongst the models. We chose to publish results using Representation 2 because it requires no modifications to the CNNs, which we felt was the most fair approach when comparing to models trained on demosaiced images.

B. Training on Raw Images

Figure 10 depicts the training and validation top-1 accuracy for several MobileNets during training. These models were all trained with the default hyperparameters from the original publication of MobileNets. We noticed that accuracy curves from models trained on raw images (Figures 10c and 10d) had different shape than those from models trained on RGB images (Figures 10a and 10b). The accuracy curves associated with raw images have much more variation between each iteration, indicating instability of the optimizer. We recognize that the training recipe we used was tuned assuming RGB input images, and the performance clearly does not translate well to raw images. We intuited that the cause for the poor accuracy on raw images was batch normalization, which behaved poorly when used with input images with pixel distributions seen in raw images. Upon investigation of batch normalization hyperparameters, we determined that changing batch normalization decay from 0.9997 to 0.99 removed instability from the raw training curves. This change effectively enables batch normalization statistics to update more quickly, which we believe is important for raw data due to its highly non-linear pixel distribution. The described change to the raw training recipe successfully removed the variation in accuracy seen in Figures 10c and 10d. We explored other changes to the raw training recipe, but we found that none of our tests yielded better results.

C. Lab-Captured Validation Set Results

Section 4.3 discusses validation tests designed to provide confidence in our experimental methodology (specifically, to ensure that the capture model does not introduce any strange behavior when used for CNN training). Tables 3 and 4 provide the accuracy of testing the models trained on data simulated using the capture model. We see similar trends in these test results as we do when testing on processed versions of the ImageNet validation set (Figures 4 and 6).

The overall test accuracy on our lab-captured validation set is lower than on the processed ImageNet validation set. We believe that the difference in accuracy is due to a higher difficulty to classify the images in our lab-captured dataset. The levels of noise in our dataset are much higher, and lighting conditions are on average worse than what is found in ImageNet. These differences can be seen in Figure 11, which displays randomly sampled images from the ImageNet validation set and our lab-captured dataset.
Figure 10: Training and validation accuracy curves for 3 random initializations of MobileNet using default training hyperparameters.

Table 3: Top-1 test accuracy on real data for MobileNets trained on simulated data (MN = MobileNet).

| Image processing | MN-0.25 acc. (%) | MN-0.50 acc. (%) | MN-0.75 acc. (%) | MN-1.00 acc. (%) |
|------------------|------------------|------------------|------------------|------------------|
| None             | 1.70             | 6.70             | 9.50             | 7.65             |
| Denoise          | 1.54             | 7.20             | 8.00             | 8.43             |
| BL + TM          | 2.65             | 8.67             | 7.54             | 11.33            |
| BL + WB + TM     | 1.95             | 8.77             | 8.02             | 10.96            |
| ISP w/o denoise  | 2.45             | 8.50             | 8.35             | 14.00            |
| Full ISP         | 3.15             | 9.90             | 8.65             | 14.95            |

Table 4: Top-5 test accuracy on real data for MobileNets trained on simulated data (MN = MobileNet).

| Image processing | MN-0.25 acc. (%) | MN-0.50 acc. (%) | MN-0.75 acc. (%) | MN-1.00 acc. (%) |
|------------------|------------------|------------------|------------------|------------------|
| None             | 5.00             | 19.15            | 21.10            | 21.35            |
| Denoise          | 3.35             | 17.85            | 20.95            | 22.80            |
| BL + TM          | 9.23             | 20.10            | 21.90            | 26.05            |
| BL + WB + TM     | 10.54            | 24.01            | 22.50            | 27.20            |
| ISP w/o denoise  | 10.40            | 23.60            | 20.65            | 26.15            |
| Full ISP         | 13.10            | 27.60            | 22.35            | 29.20            |
Figure 11: Selection of images from ImageNet and from our lab-captured dataset (sampled randomly). The images shown from ImageNet have been processed using our capture model and ISP model, whereas the images shown from our lab-captured dataset have only been processed using our ISP model.