# ASP Vision: Optically Computing the First Layer of Convolutional Neural Networks using Angle Sensitive Pixels

Huaijin G. Chen* Rice University
Suren Jayasuriya* Cornell University
Jiyue Yang Cornell University
Judy Stephen Cornell University
Sriram Sivaramakrishnan Cornell University
Ashok Veeraraghavan Rice University
Alyosha Molnar Cornell University

## Abstract

*Deep learning using convolutional neural networks (CNNs) is quickly becoming the state-of-the-art for challenging computer vision applications. However, deep learning's power consumption and bandwidth requirements currently limit its application in embedded and mobile systems with tight energy budgets. In this paper, we explore the energy savings of optically computing the first layer of CNNs. To do so, we utilize bio-inspired Angle Sensitive Pixels (ASPs), custom CMOS diffractive image sensors which act similar to Gabor filter banks in the V1 layer of the human visual cortex. ASPs replace both image sensing and the first layer of a conventional CNN by directly performing optical edge filtering, saving sensing energy, data bandwidth, and CNN FLOPS to compute. Our experimental results (both on synthetic data and a hardware prototype) for a variety of vision tasks such as digit recognition, object recognition, and face identification demonstrate up to 90% reduction in image sensor power consumption and 90% reduction in data bandwidth from sensor to CPU, while achieving similar performance compared to traditional deep learning pipelines.*

## 1. Introduction

State-of-the-art visual recognition algorithms utilize convolutional neural networks (CNNs) which use hierarchical layers of feature computation to discriminate visual stimuli. Early CNNs from LeCun et al. [26] showed promising results in digit recognition [27]. The advent of GPU computing has allowed CNN training on large, public online data sets, and triggered an explosion of current research. CNNs have started to perform on par with or even surpass humans on some image recognition challenges such as ImageNet [23, 43]. CNNs have been universally applied to different vision tasks including object detection and localization [50], pedestrian detection [49], face recognition [44], and even synthesizing new objects [8]. However, many applications in embedded vision such as vision for mobile platforms, autonomous vehicles/robots, and wireless sensor networks have stringent constraints on power and bandwidth, limiting the deployment of CNNs in these contexts.

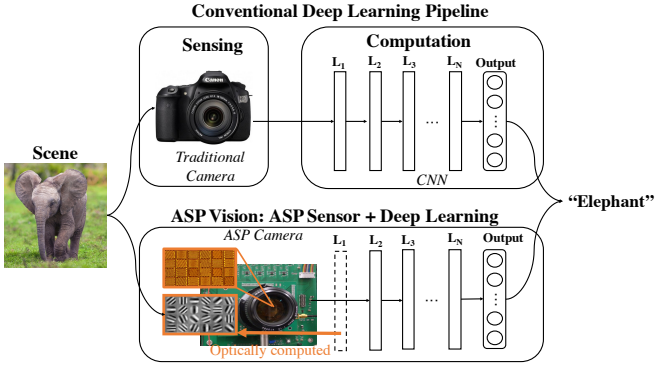



Figure 1: **A diagram of our proposed ASP Vision**. ASP Vision, our proposed system, is compared with a conventional deep learning pipeline. ASP Vision system saves 90% in energy and transmission bandwidth in the sensing stage, compared to a traditional camera.

### 1.1. Motivation and Challenges

Porting deep learning vision systems to embedded and battery-operated applications necessitates overcoming the following challenges:

- **Sensor power:** Image sensors are notoriously power-hungry, sometimes accounting for more than 50% of the power consumption in many embedded vision applications [30]. In addition, current image sensors are not optimized to significantly save power for such computer vision tasks [30]. Several researchers, most recently [29], have argued that always-on, battery-operated, embedded

*Authors contributed equally to this paper

vision systems necessitate a complete redesign of the image sensor to maximize energy-efficiency.

- **Computing power:** CNNs, while providing enormous performance benefits, also suffer from significantly increased computational complexity. GPUs and multi-core processors are power hungry, and the number of FLOPS (floating point operations) for CNNs can easily be on the order of billions.
- **Data bandwidth:** Data bandwidth requirements place strict design constraints on traditional vision architectures. Moderate image resolution of 1 megapixel at 30 fps (frames per second) results in a bandwidth requirement of over 0.5 Gbps (Giga-bits per second). This can bottleneck I/O buses that transfer images off the sensor to the CPU and increases the power requirements, computational complexity, and memory for the system.

### 1.2. Our Proposed Solution

To solve the challenges described above, we explore novel image sensors that can save energy in an embedded vision pipeline. In particular, we use existing Angle Sensitive Pixels (ASPs) [53], bio-inspired CMOS image sensors that have Gabor wavelet impulse responses similar to those in the human visual cortex, to perform optical convolution for the CNN first layer. We call this combination of ASP sensor with CNN backend *ASP Vision*. This system addresses embedded deep learning challenges by:

- **Reducing sensor power** by replacing traditional image sensors with energy-efficient ASPs that only digitize edges in natural scenes, giving orders of magnitude reduction in power consumption.
- **Reducing computing power** by optically computing the first convolutional layer using ASPs, thus leaving subsequent network layers with reduced FLOPS to compute.
- **Reducing bandwidth** by relying on the inherent reduced bandwidth of ASP sensors encoding only edge responses.

### 1.3. Contributions

In this paper, we will describe in detail our system for optically computing the first layer of CNNs. Note that we are neither introducing ASPs for the first time nor claiming a new CNN architecture. Instead, we are deploying ASPs to increase energy efficiency in an embedded vision pipeline while maintaining high accuracy. In particular, our main contributions in this paper include:

- Showing the optical response of Angle Sensitive Pixels emulates the first layer of CNNs
- Analysis of the energy and bandwidth efficiency of this optical computation
- Evaluation of system performance on multiple datasets: MNIST [27], CIFAR-10/100 [22], and PF-83 [1].
- An operational prototype of the ASP Vision system and real experimental results on digit recognition and face recognition using our prototype.

### 1.4. Limitations

Our proposed approach is also limited by some practical factors. While there are significant potential FLOPS savings from optical computation, our current prototype achieves a modest fraction of these savings due to the prefabricated sensor's design choices. In addition, ASPs themselves are challenged with low light efficiency and reduced resolution that we address in detail in Section 3.4.2. Finally, our current hardware prototype has limited fidelity since it was not fabricated in an industrial CMOS image sensor process. We discuss this in Section 5. We caution readers from placing too much expectation on the visual quality of a research prototype camera, but hope the ideas presented in the paper inspire further research in novel cameras for computer vision systems.

## 2. Related Work

In this section, we survey the literature with particular focus on energy-efficient deep learning, computational cameras, and hardware-based embedded vision systems.

**Convolutional Neural Networks** are currently the subject of extensive research. A high level overview of CNNs is given by LeCun et al. [25]. Since this paper does not improve CNN accuracy or propose new networks, we highlight recent work on real-time performance and resource efficiency. Ren et al. use faster R-CNN [42] to achieve millisecond execution time, enabling video frame rates for object detection. In addition, researchers have explored reducing floating point multiplications [32], quantization of weights in CNNs [11, 14], network compression [5], and trading off accuracy for FLOPs [44].

On the sensor side, **computational cameras** have emerged to expand the toolset of modern imaging systems. Cameras have been augmented to capture light fields [18, 35, 52], polarization [13], high dynamic range [45], and depth [39]. Similar to ASPs, cameras that compute features include on-chip image filtering [12, 36] or detect events [28].

**Embedded vision** has been spurred by advances in imaging technology and digital processing. For convolutional neural networks, analog ASICs [3], FPGAs [9], and neuromorphic chips [40] implement low power calculations with dedicated hardware accelerators. LiKamWa et al. [29] propose a new analog-to-digital converter for image sensors that performs CNN image classification directly to avoid the I/O bottleneck of sending high resolution images to the processor. Micro-vision sensors [21] perform optical edge filtering for computer vision on tight energy budgets. Similar to this paper, inference/learning on coded sensor measure-

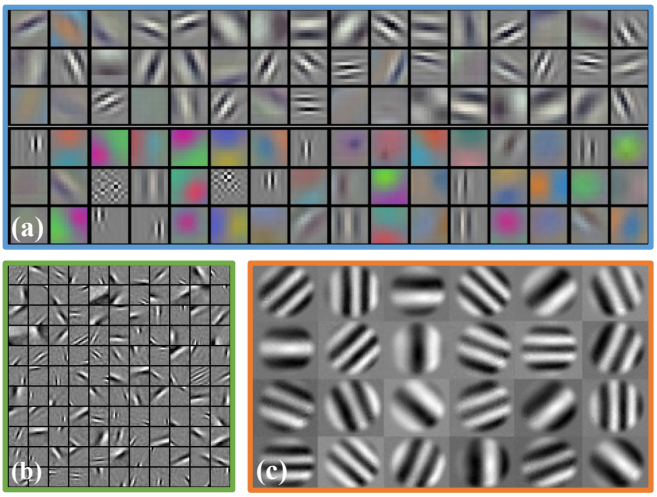



Figure 2: **Comparison of first layer weights for three different systems:** (a) Traditional deep learning architecture AlexNet trained on ImageNet [23], (b) Set of weights given by sparse coding constraints similar to the receptive fields of simple cells in the V1 [37], and (c) ASP optical impulse responses for 2D incidence angles [18].

ments from compressive sensing imaging has saved bandwidth/computation [19, 24, 33]. Dynamic Vision Sensors (DVS) have been used for face recognition while saving energy compared to conventional image sensors [34]. All this research forecasts higher levels of integration between deep learning and embedded vision in the future.

# 3. ASP Vision

A diagram of our proposed ASP Vision system is presented in Figure 1. The custom image sensor is composed of Angle Sensitive Pixels which optically computes the first layer of the CNN used for visual recognition tasks. In the following subsections, we describe how hardcoding the first layer is application-independent, ASP design, and how ASPs perform optical convolution with energy and bandwidth savings. Finally, we discuss current limitations with ASP design and imaging for embedded vision.

## 3.1. Hardcoding the First Layer of CNNs

In partitioning a deep learning pipeline, a central question is what layers of the CNN should be implemented in hardware versus software. Hardcoded layers generally lead to significant energy savings provided a suitably efficient hardware implementation is used. However, maintaining the last layers of CNNs in software allows flexibility for network reconfiguration, transfer learning [38], and fine-tuning [2].

In our system, we are interested in optically computing the first layer of CNNs in hardware. We note recent research that shows the first layers of CNNs are application-independent and transferable [55]. Fine-tuning the CNN by retraining only the last few layers on a new application-domain leads to high accuracy. In particular, the first layer learned by most CNN architectures consists of oriented edge filters, color blobs, and color edges (as visualized AlexNet's [23] first layer in Figure 2(a)). These edge filters are not a surprise and are also found in the receptive fields of simple cells in the V1 layer of the human visual system. Olhausen and Field characterized these filters as Gabor wavelets, visualized in Figure 2(b), and showed how they perform sparse coding on natural image statistics [37].

Therefore hardcoding this first layer should be independent of application and roughly converges to the same set of Gabor filters for most networks. Our main idea is to use Angle Sensitive Pixels (ASPs) in our image sensor front end to compute this convolutional layer in the optical domain at low electronic power consumption.

## 3.2. Angle Sensitive Pixels

### 3.2.1 Background

ASPs are photodiodes, typically implemented in a CMOS fabrication process, with integrated diffraction gratings that image the Talbot diffraction pattern of incoming light [53]. These diffraction gratings give a sinusoidal response to incident angle of light given by the following equation [18]:

$$i(x,y) = 1 + m\cos\left(\beta\left(\cos\left(\gamma\right)\theta_x + \sin\left(\gamma\right)\theta_y\right) + \alpha\right), \quad (1)$$

where $\theta_x, \theta_y$ are 2D incidence angles, $\alpha, \beta, \gamma$ are parameters of the ASP pixel corresponding to phase, angular frequency, and grating orientation, and $m$ is the amplitude of the response. A tile of ASPs contain a diversity of angle responses, and are repeated periodically over the entire image sensor to obtain multiple measurements of the local light field. ASPs have been shown to capture 4D light fields [18] and polarization information [20]. An advantage of these sensors is that they are CMOS-compatible and thus can be manufactured in a low-cost industry fabrication process.

### 3.2.2 Optical Convolution

In particular, ASP responses to incidence angle allow optical convolution and edge filtering. Using two differential pixels of phase $\alpha$ and phase $\alpha+\pi$ (pixels A and B of Figure 3), we subtract their responses, $i_\alpha - i_{\alpha+\pi}$, to obtain the sinusoidal term of Equation 1 which depends solely on angle without the fixed DC offset. Figure 3 shows these measured differential pixel's impulse responses across an ASP tile. They resemble several different Gabor wavelets of different frequency, orientation, and phase which tile 2D frequency space. These impulse responses are convolved optically with objects in the scene during the capture process. The

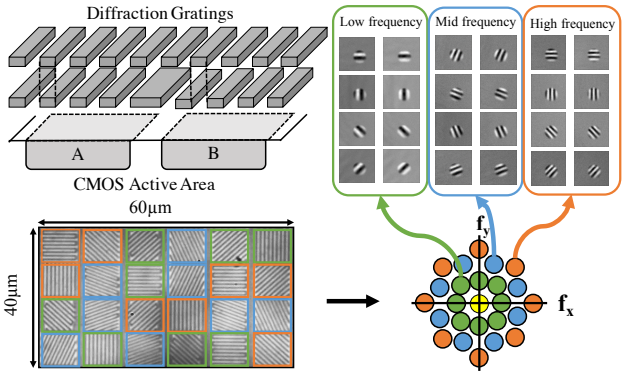



Figure 3: **ASP Pixel Designs:** ASP differential pixel design using diffraction gratings is shown. A 4 × 6 tile contains 10$\mu$m pixels whose optical responses are Gabor filters with different frequency, orientation, and phase. These filters act as bandpass filters in 2D frequency space [54].

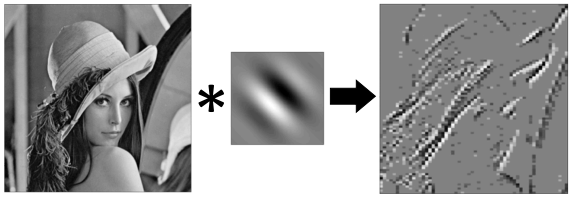


Figure 4: **ASP Differential Output:** Optical convolution of a scene with a differential ASP impulse response results in an edge filtered image (real images from prototype camera in [54]).

resulting ASP output correspond to edge filtered images as displayed in Figure 4. We use this optical convolution with Gabor wavelets to compute the first convolutional layer of a CNN.

Analogously, the V1 layer of the visual cortex contains Gabor wavelet responses for the receptive fields of simple cells, and Olhausen and Field showed that this representation is maximally efficient in terms of information transfer and sparse coding [37]. This is partly why we claim this system is bio-inspired: we are taking advantage of ASP's Gabor-like response to naturally compress the statistics of natural scenes to edges. This edge representation has direct implications for the low power consumption of ASPs.

#### 3.2.3 Energy and Bandwidth Efficiency

Prior work has designed ASP readout circuitry to leverage the sparseness of edge filtered images, enhancing energy efficiency [54]. Circuit readout that involves a differential amplifier can read out differential pixels, subtract their responses, and feed it to an analog-to-digital (ADC) converter [54]. This ADC is optimized to only convert pixels when there is sufficient edge information, leading to low power image sensing and digitization as compared to a traditional image sensor.

A comparison of an ASP-based image sensor [54] to a modern Sony mobile image sensor [48] is shown in Table 1. All numbers reported are from actual sensor measurements, but we caution the readers that these comparisons are approximate and do not take into account process technology and other second order effects. We report energy normalized for pixels and frame rate:

$$E = \frac{P}{f \cdot N}, \tag{2}$$

where normalized energy $E$ has units $[J/(frame \cdot pixel)]$, $P$ is power, $f$ is frame rate, and $N$ is number of pixels. We calculate the amount of normalized energy using ASPs to be 33 pJ/frame/pixel as opposed to the Sony sensor with 340 pJ/frame/pixel. This yields **a 90% reduction in image sensing energy**. This savings is due to the low power consumption of the ADC (100uW) that only converts pixel data when there is an edge present.

Since edge data is significantly smaller to transmit, ASPs can also save on the bandwidth of image data sent off the image sensor, thus alleviating an I/O bottleneck from image sensor to CPU. Prior work has shown that ASPs obtain a **bandwidth reduction of 10:1 or 90%** for images by only storing non-zero coefficients of edges and using run-length encoding [54]. For a traditional image sensor, 1.2 Mbits is needed to digitize 150Kpixels (384 × 384) at 8 bit resolution while ASPs only require 120Kbits. We refer readers to [54] for more details about these circuit designs and their energy and bandwidth efficiency for ASP imaging.

#### 3.2.4 Limitations of ASPs for Visual Recognition

Some limitations with using ASPs for visual recognition include reduced image sensor resolution, low light efficiency, and depth-dependent edge filtering behavior. We outline these challenges and recent research to alleviate these issues.

Since a tile of ASPs is required to obtain different edge filters, image sensor resolution is reduced by the tile resolution. It is not clear how small ASP pixels can be fabricated, especially since a few periods of diffraction gratings are needed for adequate signal-to-noise ratio (SNR) and to avoid edge effects. However, recent research in interleaved photodiode design has increased pixel density by 2× [46, 47]. Reduced resolution may have an adverse effect on vision tasks [7], although no critical minimum resolution/spatial frequency threshold has been suggested for image sensors to capture.

| | Sony (ISSCC 2015) | ASP Image Sensor |
|---|---|---|
| Resolution | 5256 x 3934 (20M) | 384 x 384 (effective ASP tile resolution: 96 x 64) |
| Energy consumption | Total power: 428 mW<br>No breakdown of power reported | Total Power: 1.8 mW<br>Pixel Array: 300 µW<br>Amplifiers: 900 µW<br>Timing/Addressing: 500 µW<br>ADCs: 100 µW |
| | **340 pJ/frame/pixel** | **33 pJ/frame/pixel** |
| | **90.2% Energy saving** | |
| Transmission bandwidth | Transmitting the entire image | Transmitting only edges |
| | 1.2 Mbits/frame @ 384 ×384×8bits | 120 Kbits/frame @ 384×384×8bits |
| | **10:1 Compression ratio** | |
| Capabilities | 2D image and video capture | 2D images and video, edge filtered images, light field information |

Table 1: **Comparison of ASP image sensor [54] and modern smartphone image sensor** [48].

ASP pixels can suffer loss of light through the diffraction gratings as low as 10% relative quantum efficiency, which yields decreased SNR for differential edge responses. This in part explains the noisy visual artifacts present in the hardware prototype, and the need for large amounts of light in the scene. However, recent work in phase gratings [46, 47] have increased light efficiency up to 50% relative quantum efficiency.

Finally, the optical edge-filtering behavior of ASPs is depth-dependent since the optical responses only work away from the focal plane with a large aperture camera [53]. This depth-dependence limits the application of ASPs to wide aperture systems with shallow depth-of-field, but also enables the potential for depth and light field information to be utilized as scene priors (which we do not explore in the scope of this paper).

# 4. Analysis

To analyze our proposed design and its tradeoffs, we developed a simulation framework to model both ASP image formation and CNNs. We simulate ASP image capture, and then propogate the resulting ASP edge images through the rest of the CNN. Typically this output data has dimensions W $\times$ H $\times$ D where there are D ASP filtered images, each of size W $\times$ H. We use the same input image resolution for both ASPs and baselines since we already accounted for image resolution in our normalized energy savings in Table 1.

For all our simulations, we use the ASP tile design of Figure 3 which matches the existing hardware prototype of [54]. We use 12 out of 24 of the ASP filters with cosine responses ($\alpha = 0$) and low, medium, and high angular frequencies. The other 12 filters have sine responses ($\alpha = \pi/2$) which did not yield suitably different convolution outputs, and thus these matching input channels caused gradient exploding and convergence issues. Finally, since our prototype ASP system does not have color pixels, we report all baselines with respect to grayscale for performance. All our dataset results are summarized in Figure 5 and discussed in the following subsection.

We use MatConvNet [51] to perform deep learning experiments and train on a NVIDIA GeForce GTX TITAN Black GPU.

## 4.1. Performance on Visual Recognition Tasks

We first analyze the performance of ASP Vision across several visual recognition tasks to show the broad applicability of this system. The datasets we benchmark include MNIST [27] for digit recognition, CIFAR-10/100 [22] for object recognition, and PF-83 [1] for face identification.

For all experiments, we benchmark baselines with their original first layer number of filters (D) and also with $D = 12$ for a more fair comparison with ASP Vision when we analyze FLOPS in the next subsection.

**MNIST:** Our first simulation involved digit recognition on MNIST, 60,000 training and 10,000 test images of size 28 $\times$ 28. For a baseline, we use LeNet [6] which is a five layer CNN with both 20 and 12 first-layer filters to achieve 99.12%and 99.14% percent respectively. Using LeNet, ASP Vision achieved 99.04% performance.

**CIFAR-10/100:** Our second simulation involved the CIFAR-10/100 data sets [22] for object recognition with 50,000 training and 10,000 test images of size 32 $\times$ 32 (the 10/100 corresponds to the number of classes). Our baseline algorithm for these datasets was the Network in Network (NiN) structure [31] that uses CNNs with fully connected networks acting as inner layers. The baseline used both 92 and 12 first-layer filters to achieve respectively 86.40% and 84.90% percent on CIFAR-10, and 57.50% and 55.60% on CIFAR-100. Note again that these percentages are for grayscale images. ASP Vision achieved 81.8% and 50.9% respectively on CIFAR-10/100.

**PF-83:** Our final simulation on PF-83 [1] is an example of fine-grained classification to show that ASP features are transferable even for a difficult task like face identification (not to be confused with face verification or detection). The data consists of 13,002 images with size 256 $\times$ 256 with 83 classes of faces. Our baseline VGG-M-128 algorithm [4] achieved 65.67% and 69.78% percent on this data set with 192 and 12 first-layer filters respectively. Using ASP Vision, we achieved 66.8% percent on PF-83.

Across all datasets, ASP Vision was within 0.1-5.6% of the baseline accuracies. Note that this comparable-to-slight

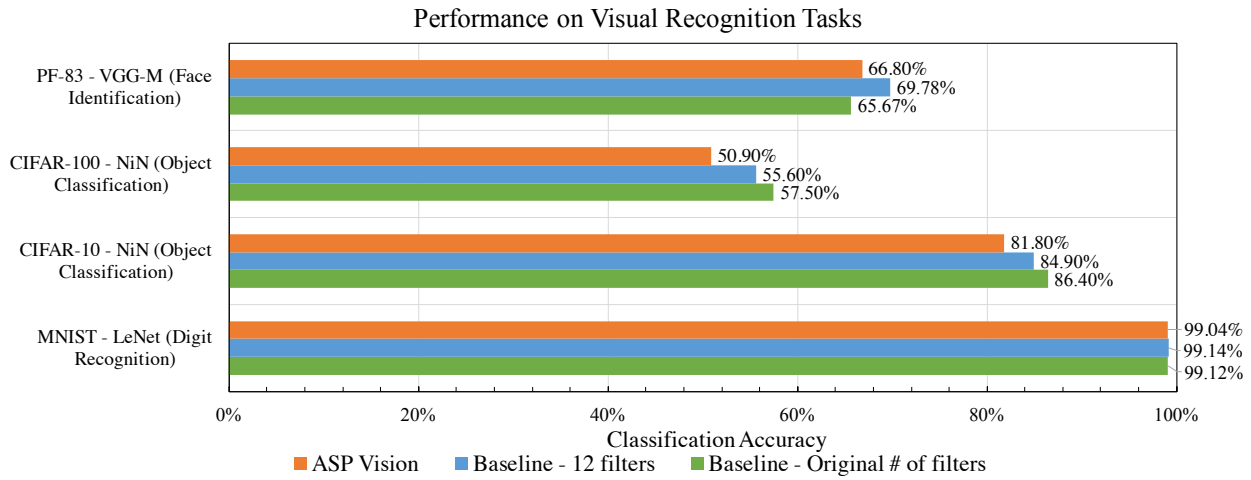



Figure 5: **ASP Vision Performance:** ASP Vision's performance on various visual recognition tasks, evaluated using three networks, LeNet [6], NiN [31] and VGG-M-128 [4], and over four different datasets: MNIST [27], CIFAR-10 [22], CIFAR-100 [22], and PF-83 [1].

| | **VGG-M** | | **NiN** | | **LeNet** | |
|---|---|---|---|---|---|---|
| # of Conv. Layers | 8 | | 9 | | 4 | |
| Input Image Size | $224 \times 224 \times 3$ | | $32 \times 32 \times 3$ | | $28 \times 28 \times 1$ | |
| # of First Layer Filters | 96 (Original) | 12 (Prototype) | 192 (Original) | 12 (Prototype) | 20 (Original) | 12 (Prototype) |
| First Layer Conv. Kernel | $7 \times 7 \times 96$ | $7 \times 7 \times 12$ | $5 \times 5 \times 192$ | $5 \times 5 \times 12$ | $5 \times 5 \times 20$ | $5 \times 5 \times 12$ |
| FLOPS of First Layer | 708.0M | 88.5 M | 14.75M | 921.6K | 392 K | 235 K |
| Total FLOPS | 6.02G | 3.83 G | 200.3M | 157 M | 10.4 M | 8.8 M |
| First Layer FLOPS Saving | 11.76% | 2.3% | 7.36% | 0.6% | 3.77% | 2.67% |

Table 2: **Network Structure and FLOPS:** Common CNN architectures such as VGG-M-128 [4], NiN [31], LeNet [6] are compared for the FLOPS savings from optically computing the first layer of these networks. The actual FLOPS savings for the working prototype ASP Vision system are also included.

degradation in performance comes with the energy savings of 90% in image sensing and 90% in transmission bandwidth by using ASPs.

### 4.2. FLOPS savings

The FLOPS saved by ASP Vision is dependent on both the network architecture and the size of the input images.

We first look at different CNN architectures and their potential savings from optically computing the first layer shown in Table 2. Additionally, since we simulate only our hardware prototype of 12 filters, we compare the FLOPS of our prototype ASP Vision system with those of modified CNNs with a 12-filter first layer. This comparison results in lower FLOPS savings, but yields higher visual recognition performance. Using an ASP with more numbers of filters would allow more FLOPS savings when compared to CNNs with the equivalent number of first-layer filters.

Secondly, FLOPS are input image size dependent as larger input image sizes will yield proportionally more FLOPS savings for an ASP Vision system. Even for a relatively deep network, the first layer still contributes a considerable amount of FLOPS if the input image is large. For example, the FLOPS of the first layer of GoogLeNet [50] is about 2.5% of the total FLOPS.

### 4.3. Noise analysis

In Figure 6, we simulate the effects of additive white noise during image sensing for MNIST images. We compare ASP Vision versus the baseline LeNet with SNR varying from 9dB to 28dB. Note that at low SNRs, ASP Vi-

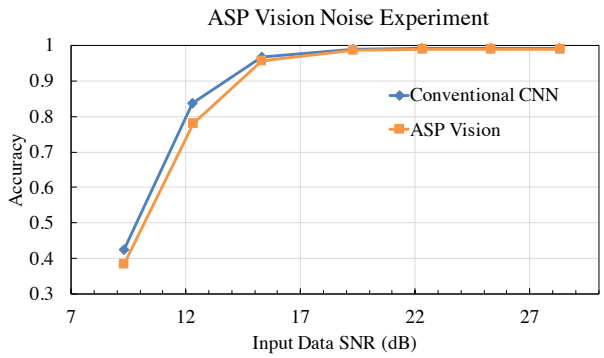



Figure 6: **ASP Vision Noise Analysis:** To explore the impact of noise to the performance of ASP Vision, we vary SNR from 9 dB to 28 dB and compared ASP Vision with baseline LeNet performance on MNIST.

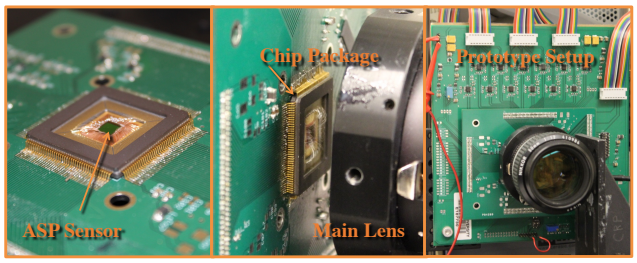



Figure 7: **ASP Camera Setup:** Working prototype with 5 mm x 5 mm CMOS ASP image sensor, F1.2 Nikon lens, and associated readout printed circuit board [18, 54].

sion suffers more from accuracy degradation (9dB - 38.6%, 12dB - 77.9%) as compared to the baseline (9dB - 42.6%, 12dB - 83.6%). However, above 15dB SNR, both methods have high accuracy and are comparable.

### 4.4. ASP parameter design space

We finally explore how choice of ASP parameters affects performance with the salient parameters being angular frequency $\beta$ and grating orientation $\chi$. We performed a coarse sweep of $\beta \in [5, 50]$, $\chi \in [-\frac{\pi}{2}, \frac{\pi}{2})$ for one filter on MNIST, and found no strong dependence on parameters and performance.

We also ran sensitivity analysis on the parameter set by running 100 simulations using 6 randomized ASP filters each time on the MNIST dataset. We obtained a mean of 1.13% error with a standard deviation of 0.13%, which suggests there is no strong dependence of ASP parameters. This might be partly because the CNN learns to work with the filters it is given in the first layer.

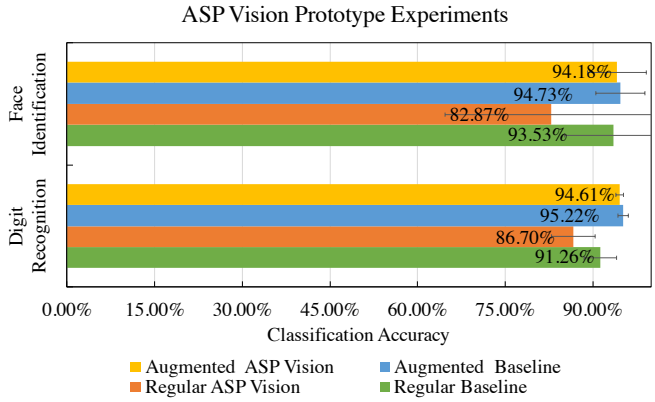



Figure 8: **ASP Vision Prototype Experiments:** Real-world digit recognition and face identification tasks were performed on ASP Vision prototype system. Accuracy and standard deviation for 20 trials are shown.

## 5. Hardware Prototype and Experiments

Finally, to completely validate our system design, we show results of classification on an existing camera prototype for digit and face recognition. We report mean validation accuracy and standard deviation for 20 trials with a random split of 85% for training and 15% validation.

The prototype camera system is the same setup as used in [18, 54]. A 5 mm $\times$ 5 mm CMOS image sensor was fabricated in a 180nm process, using a tile size of 4 $\times$ 6 ASPs with 10um pixels for a 64 $\times$ 96 resolution sensor. This sensor is placed behind a Nikon F1.2 lens for imaging small objects on an optical bench. See Figure 7 for picture of our prototype camera.

In general, our prototype camera suffers from high noise even after a fixed pattern noise subtraction. This may be due to noise issues from the readout circuits or even from external amplifiers on the printed circuit board. This limits the aesthetics of the ASP edge images, but we still achieved high accuracy in visual recognition. Further circuit design such as correlated double sampling and fabrication in an industrial CMOS image sensor process could help alleviate these noise issues in the future.

**Digit Recognition:** Using a display with appropriate brightness approximately one meter away, we show images of the MNIST dataset, and capture ASP responses as shown in Figure 9. We captured over 300 pictures of real digits to be used in our learning experiment. We also used linear shifts and rotations to augment the size of our dataset to 2946 images. For real data, the baseline LeNet algorithm performed 91.26% with $\sigma = 2.77\%$ on the regular dataset, and 95.22% with $\sigma = 0.87\%$ on the augmented dataset. ASP Vision achieved 86.7% with $\sigma = 3.75\%$ on the regular dataset, and 94.61% with $\sigma = 0.68\%$ on the augmented

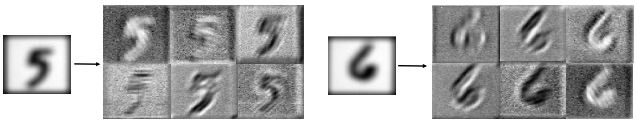


Figure 9: **Digit Recognition:** Digits are captured by the ASP image sensor, 6 of 12 sample edge responses from the tile are shown. ASP Vision achieved >90% accuracy in digit recognition on this dataset.

dataset.

**Face Identification:** To test face identification, we took 200 pictures of 6 subjects approximately 2.5 meters away in the lab, and the edge responses and example results and errors are visualized in Figure 10. We used dataset augmentation again to increase the dataset to 7200 pictures. For the baseline NiN, we achieved 93.53% with $\sigma = 8.37\%$ on the regular dataset, and 94.73% with $\sigma = 4.2\%$ on the augmented dataset. ASP Vision achieved 82.87% with $\sigma = 18.12\%$ on the regular dataset, and 94.18% with $\sigma = 5.04\%$ on the augmented dataset.

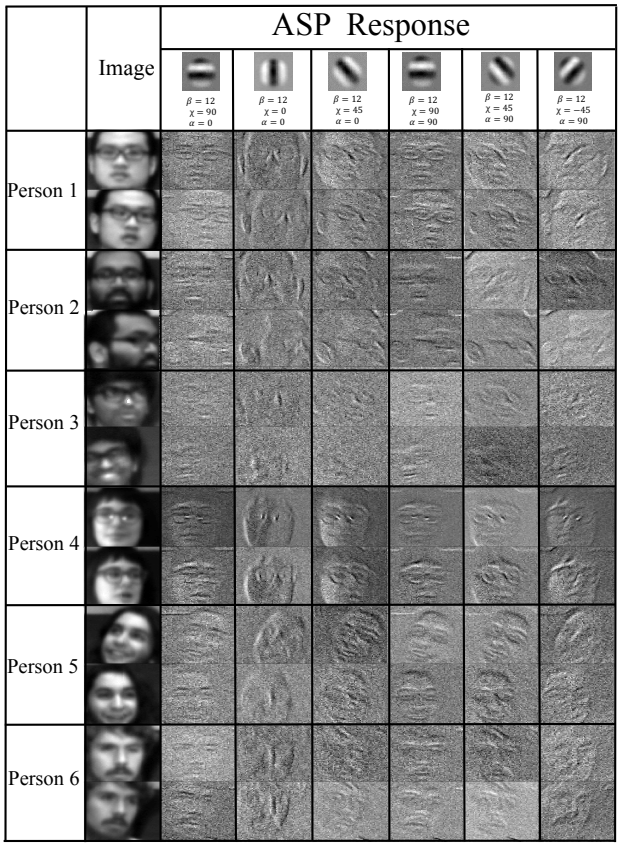
ASP Vision performs about 5-10% worse than baseline with regular data. After introducing linear shifts and rotations to augment the data, ASP Vision performs on par with conventional CNNs. These datasets may not generalizable and may exhibit underlying trends/bias due to the custom data acquisition. However, these results clearly show the feasibility of ASP Vision on a real working camera prototype.

## 6. Discussion

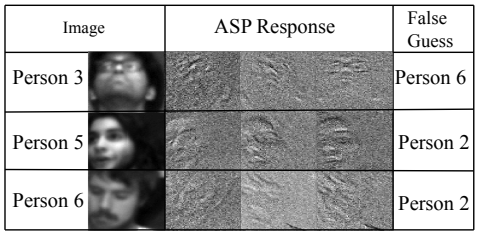
Optically computing the first layers of CNNs is a technique that is not solely limited to ASPs. Sensors such as the DVS can compute edge features at low power/bandwidth [28], or using cameras with more general optical computation [56] could capture convolutional features. In addition, it is not possible to hardcode additional convolutional layers optically in ASPs beyond the first layer, limiting the potential energy savings. Fully optical systems for artificial neural networks using holography [10, 17, 41] or light waves in fiber [16] may achieve better energy savings.

We have presented an energy-efficient imaging system that uses custom Angle Sensitive Pixels for deep learning. We leverage up to 90% energy savings and bandwidth reduction in ASPs while achieving good visual recognition performance on synthetic and real hardware data sets. We hope our work inspires sensor+ deep learning co-design for embedded vision tasks in the future.

**Acknowledgements:** We would like to thank Albert Wang for help designing the existing ASP prototype, and Dalu Yang and Dr. Arvind Rao for lending us the GPU. We also especially want to thank Dr. Eric Fossum for pointing out a correction with regard to our energy comparison. We also received valuable feedback from Cornell's Graphics & Vision group, Prof. David Field, Achuta Kadambi, Robert LiKamWa, Tan Nguyen, Dr. Ankit Patel, Kuanchuan Peng, and Hang Zhao. This work was supported by NSF CCF-1527501, THECB-NHARP 13308, and NSF CAREER-1150329. S.J. was supported by a NSF Graduate




(a)




(b)

Figure 10: **Face Recognition:** 200 images of 6 subjects were captured in the lab. Edge responses for (a) correct and (b) misidentified identification is showed. ASP Vision achieved >90% accuracy for face identification.

Research Fellowship and a Qualcomm Innovation Fellowship. H.G.C was partially supported by a Texas Instruments Graduate Fellowship.

## References


[1] B. C. Becker and E. G. Ortiz. Evaluating open-universe face identification on the web. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2013 IEEE Conference on*, pages 904–911. IEEE, 2013. 2, 5, 6

[2] Y. Bengio. Deep learning of representations for unsupervised and transfer learning. *Unsupervised and Transfer Learning Challenges in Machine Learning*, 7:19, 2012. 3

[3] B. E. Boser, E. Sackinger, J. Bromley, Y. Le Cun, and L. D. Jackel. An analog neural network processor with programmable topology. *Solid-State Circuits, IEEE Journal of*, 26(12):2017–2025, 1991. 2

[4] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman. Return of the devil in the details: Delving deep into convolutional nets. In *British Machine Vision Conference*, 2014. 5, 6

[5] W. Chen, J. T. Wilson, S. Tyree, K. Q. Weinberger, and Y. Chen. Compressing convolutional neural networks. *CoRR*, abs/1506.04449, 2015. 2

[6] Y. L. Cun, B. Boser, J. S. Denker, R. E. Howard, W. Habbard, L. D. Jackel, and D. Henderson. Advances in neural information processing systems 2. chapter Handwritten Digit Recognition with a Back-propagation Network, pages 396–404. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1990. 5, 6

[7] D. Dai, Y. Wang, Y. Chen, and L. Van Gool. How useful is image super-resolution to other vision tasks? 2016. 4

[8] A. Dosovitskiy, J. Tobias Springenberg, and T. Brox. Learning to generate chairs with convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1538–1546, 2015. 1

[9] C. Farabet, C. Poulet, and Y. LeCun. An fpga-based stream processor for embedded real-time vision with convolutional networks. In *Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on*, pages 878–885. IEEE, 2009. 2

[10] N. H. Farhat, D. Psaltis, A. Prata, and E. Paek. Optical implementation of the hopfield model. *Applied Optics*, 24(10):1469–1475, 1985. 8

[11] Y. Gong, L. Liu, M. Yang, and L. Bourdev. Compressing deep convolutional networks using vector quantization. *arXiv preprint arXiv:1412.6115*, 2014. 2

[12] V. Gruev and R. Etienne-Cummings. Implementation of steerable spatiotemporal image filters on the focal plane. *Circuits and Systems II: Analog and Digital Signal Processing, IEEE Transactions on*, 49(4):233–244, 2002. 2

[13] V. Gruev, R. Perkins, and T. York. Ccd polarization imaging sensor with aluminum nanowire optical filters. *Optics express*, 18(18):19087–19094, 2010. 2

[14] S. Han, H. Mao, and W. J. Dally. Deep compression: Compressing deep neural network with pruning, trained quantization and huffman coding. *CoRR*, abs/1510.00149, 2015. 2

[15] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015.

[16] M. Hermans and T. Van Vaerenbergh. Towards trainable media: Using waves for neural network-style training. *arXiv preprint arXiv:1510.03776*, 2015. 8

[17] W. F. Herrington. Micro-optic elements for a compact optoelectronic integrated neural coprocessor. *Thesis*. 8

[18] M. Hirsch, S. Sivaramakrishnan, S. Jayasuriya, A. Wang, A. Molnar, R. Raskar, and G. Wetzstein. A switchable light field camera architecture with angle sensitive pixels and dictionary-based sparse coding. In *Computational Photography (ICCP), 2014 IEEE International Conference on*, pages 1–10. IEEE, 2014. 2, 3, 7

[19] M. Iliadis, L. Spinoulas, and A. K. Katsaggelos. Deep fully-connected networks for video compressive sensing. *arXiv preprint arXiv:1603.04930*, 2016. 3

[20] S. Jayasuriya, S. Sivaramakrishnan, E. Chuang, D. Guruaribam, A. Wang, and A. Molnar. Dual light field and polarization imaging using cmos diffractive image sensors. *Optics Letters*, 40(10):2433–2436, 2015. 3

[21] S. J. Koppal, I. Gkioulekas, T. Young, H. Park, K. B. Crozier, G. L. Barrows, and T. Zickler. Toward wide-angle microvision sensors. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(12):2982–2996, 2013. 2

[22] A. Krizhevsky and G. Hinton. Learning multiple layers of features from tiny images, 2009. 2, 5, 6

[23] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012. 1, 3

[24] K. Kulkarni and P. Turaga. Reconstruction-free action inference from compressive imagers. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 38(4):772–784, 2016. 3

[25] Y. LeCun, Y. Bengio, and G. Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015. 2

[26] 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. 1

[27] Y. LeCun, C. Cortes, and C. J. Burges. The mnist database of handwritten digits, 1998. 1, 2, 5, 6

[28] P. Lichtsteiner, C. Posch, and T. Delbruck. A 128× 128 120 db 15 $\mu$s latency asynchronous temporal contrast vision sensor. *Solid-State Circuits, IEEE Journal of*, 43(2):566–576, 2008. 2, 8

[29] R. LiKamWa, Y. Hou, Y. Gao, M. Polansky, and B. . P. T. . R. Y. . . Zhong, Lin. 1, 2

[30] R. LiKamWa, B. Priyantha, M. Philipose, L. Zhong, and P. Bahl. Energy characterization and optimization of image sensing toward continuous mobile vision. In *Proceeding of the 11th annual international conference on Mobile systems, applications, and services*, pages 69–82. ACM, 2013. 1

[31] M. Lin, Q. Chen, and S. Yan. Network in network. *CoRR*, abs/1312.4400, 2013. 5, 6

[32] Z. Lin, M. Courbariaux, R. Memisevic, and Y. Bengio. Neural networks with few multiplications. *CoRR*, abs/1510.03009, 2015. 2

[33] S. Lohit, K. Kulkarni, P. Turaga, J. Wang, and A. Sankaranarayanan. Reconstruction-free inference on compressive measurements. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 16–24, 2015. 3

[34] Y. Miyatani, S. Barua, and A. Veeraraghavan. Direct face detection and video reconstruction from event cameras. In *Applications of Computer Vision (WACV), 2016 IEEE Winter Conference on*. IEEE, 2016. 3

[35] R. Ng, M. Levoy, M. Brédif, G. Duval, M. Horowitz, and P. Hanrahan. Light field photography with a hand-held plenoptic camera. *Computer Science Technical Report CSTR*, 2(11), 2005. 2

[36] A. Nilchi, J. Aziz, and R. Genov. Focal-plane algorithmically-multiplying cmos computational image sensor. *Solid-State Circuits, IEEE Journal of*, 44(6):1829–1839, 2009. 2

[37] B. A. Olshausen and D. J. Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381(6583):607–609, 1996. 3, 4

[38] S. J. Pan and Q. Yang. A survey on transfer learning. *Knowledge and Data Engineering, IEEE Transactions on*, 22(10):1345–1359, 2010. 3

[39] A. Payne. A 512x424 CMOS 3D Time-of-Flight Image Sensor with Multi-Frequency Photo-Demodulation up to 130MHz and 2GS/s ADC. *IEEE International Solid-State Circuits Conference*, pages 134–136, 2014. 2

[40] P.-H. Pham, D. Jelaca, C. Farabet, B. Martini, Y. LeCun, and E. Culurciello. Neuflow: Dataflow vision processing system-on-a-chip. In *Circuits and Systems (MWSCAS), 2012 IEEE 55th International Midwest Symposium on*, pages 1044–1047. IEEE, 2012. 2

[41] D. Psaltis, D. Brady, X.-G. Gu, and S. Lin. Holography in artificial neural networks. *Nature*, 343(6256):325–330, 1990. 8

[42] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *arXiv preprint arXiv:1506.01497*, 2015. 2

[43] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, pages 1–42, 2014. 1

[44] F. Schroff, D. Kalenichenko, and J. Philbin. Facenet: A unified embedding for face recognition and clustering. *CoRR*, abs/1503.03832, 2015. 1, 2

[45] A. Serrano, F. Heide, D. Gutierrez, G. Wetzstein, and B. Masia. Convolutional sparse coding for high dynamic range imaging. *Computer Graphics Forum*, 35(2), 2016. 2

[46] S. Sivaramakrishnan, A. Wang, P. Gill, and A. Molnar. Design and characterization of enhanced angle sensitive pixels. *Transactions on Electron Devices*, 63(1), 2016. 4, 5

[47] S. Sivaramakrishnan, A. Wang, P. R. Gill, and A. Molnar. Enhanced angle sensitive pixels for light field imaging. In *2011 International Electron Devices Meeting*. 2011. 4, 5

[48] A. Suzuki, N. Shimamura, T. Kainuma, N. Kawazu, C. Okada, T. Oka, K. Koiso, A. Masagaki, Y. Yagasaki, S. Gonoi, et al. A 1/1.7-inch 20mpixel back-illuminated stacked cmos image sensor for new imaging applications. In *Solid-State Circuits Conference-(ISSCC), 2015 IEEE International*, pages 1–3. IEEE, 2015. 4, 5

[49] M. Szarvas, A. Yoshizawa, M. Yamamoto, and J. Ogata. Pedestrian detection with convolutional neural networks. In *Intelligent Vehicles Symposium, 2005. Proceedings. IEEE*, pages 224–229. IEEE, 2005. 1

[50] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. *arXiv preprint arXiv:1409.4842*, 2014. 1, 6

[51] A. Vedaldi and K. Lenc. Matconvnet – convolutional neural networks for matlab. 2015. 5

[52] A. Veeraraghavan, R. Raskar, A. Agrawal, A. Mohan, and J. Tumblin. Dappled photography: Mask enhanced cameras for heterodyned light fields and coded aperture refocusing. *ACM Trans. Graph. (SIGGRAPH)*, 26(3):69, 2007. 2

[53] A. Wang and A. Molnar. A light-field image sensor in 180 nm cmos. *Solid-State Circuits, IEEE Journal of*, 47(1):257–271, 2012. 2, 3, 5

[54] A. Wang, S. Sivaramakrishnan, and A. Molnar. A 180nm cmos image sensor with on-chip optoelectronic image compression. In *Proc. IEEE Custom Integrated Circuits Conference (CICC)*, pages 1–4, 2012. 4, 5, 7

[55] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson. How transferable are features in deep neural networks? In *Advances in Neural Information Processing Systems*, pages 3320–3328, 2014. 3

[56] A. Zomet and S. K. Nayar. Lensless imaging with a controllable aperture. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 1, pages 339–346. IEEE, 2006. 8