Programmable Sensors for Task-Specific Imaging

Orr Zohar

Abstract—Real-world scenes have a much higher dynamic range than today's imaging sensors, leading to frequent over/under exposure of different image portions. Object detection under such extreme lighting conditions is easily confounded, which challenges existing object detection pipelines [1]. Emerging programmable focal-plane sensor-processors offer unprecedented in-pixel processing capabilities; however, their utilization as a possible solution to many computational imaging problems remains largely unexplored. Herein, we present a general approach for utilizing such sensors for *task-specific* objectives. By developing a differentiable model for both the focal-plane processing capabilities and the analog to digital conversion, we can model the entire data processing pipeline - allowing us to perform task-specific optimization (in this case, object detection). Utilizing this approach, we can conjointly learn the optimal hardware and software configuration for a given task. We illustrate this approach in the context of object detection in real-world scenes.

Index Terms—Computational Photography, End-to-End Optimization, Machine Learning, Object Detection, High Dynamic Range, Focal-Plane Sensor-Processors

1 Introduction

In recent years, we have witnessed the rapid acceleration of Machine Learning, producing endless novel applications. One of the more significant advancements has been object detection/segmentation, where current state-of-the-art (SOTA) neural algorithms reach unprecedented accuracy, even surpassing their human counterparts. Such advancements are significant for a bevy of exciting applications such as autonomous driving, personal robotics, and even in medicine [22]. Algorithms such as YOLO [7], Mask R-CNN [9], RetinaNet [11], and others leverage deep neural architectures with sophisticated layers and losses in order to achieve SOTA results. A key component of their success is owed to the quality of the data that these models are trained on. Most of them rely on large datasets, such as MS-COCO [8](which contains over 1.5 million annotated object instances). Unfortunately, these datasets are composed of LDR images, which cannot fully depict real-world scenes.

Most modern digital cameras utilize the same optical design as their analog predecessors - where optics are used to create an in-focus image of the desired scene on the photosensor array within, and the array is exposed for a fixed (and constant) time interval (using either a global or rolling shutter). The sensors integrate the incoming luminescence during the exposure, ultimately reporting the accumulated intensity measured at each pixel. However, due to the limited range of intensity these sensors are sensitive to (i.e., their *dynamic range*), only a range of incoming luminescence can be accurately reported, with some values getting saturated in bright regions while others measure values below the sensor SNR (under-exposure). Real-world scenes can have a dynamic range of up to 280 dB [1], while typical cameras have a dynamic range of 50-70 dB. Over/under exposure in such images is detrimental to the aforementioned SOTA object detection algorithms and their subsequent deployment in real-world applications.

Different computational photography approaches have managed to capture High Dynamic Range (HDR) images - images whose dynamic range more closely approximates that of the real-world scenes - however, each time, there is an inherent trade-off (see related works) [12], [13], [14], [15]. Futhermore, these methods are optimized explicitly for visual perception, rather than any single downstream task, such as object detection. It is fair to assume that, for object detection and segmentation, different details affect the accuracy of the approach. For example, edges are particularly important for object detection and segmentation, and therefore it would make sense to preserve this information more accurately.

A new trend in Computation imaging attempts to integrate hardware and software into a singular ”neural network” utilizing differential system modeling and end-to-end optimization [4], [5], [6]. In this approach, the imaging pipeline can be conceptualized as a neural auto-encoder, with the hardware essentially ”encoding” the scene, passing that information through a ”bottleneck” (which describes all the integrations performed by the camera and the clipping/quantization at the analog to digital conversion). After the bottleneck, a neural network can ”decode” the encoded measurements. Herein, we purpose to utilize task-specific end-to-end optimization for the development of an image acquisition pipeline optimized specifically for object detection. Here, the ”camera” does not necessarily need to capture images in the traditional sense, but rather measurements containing the most pertinent information for object detection.

2 Related Work

High dynamic range imaging (HDRI) attempts to extend the conventional camera's limited dynamic range through

- *O. Zohar affiliated with the Department of Electrical Engineering, Stanford University, Stanford,CA.*
  E-mail: orrzohar@stanford.edu

the use of a variety of approaches. For example, several low dynamic range images can be captured in quick succession before being combined to create a single HDR image [12], [13], [14], [15]. However, such approaches tend to quickly degrade in dynamic scenes, where the motion of objects causes ”ghosting artifacts”. Even more recently, several single-capture methods have been proposed [4], [16]. Such approaches use various methods (ND filter, SLMs, and more) to create a spatially-distributed exposure time - where the effective exposure time is dependent on the pixel position. Therefore, a post-capture HDR image reconstruction step is required for such approaches, and there is an inherent dynamic range-resolution trade-off. Furthermore, such systems tend to be expensive and require the sensor to be permanently altered, further hindering development and prototyping. Finally, some methods attempt to compress the dynamic range of the signal at the time of acquisition [2], [3] [3]. Common among these approaches is the effective ”encoding” of the high dynamic range information during the image acquisition, which requires more complex post-processing algorithms to ”decode” the captured image into an HDR image acquisition. However, these methods are rarely stable and require extensive reconstruction schemes, limiting their frame rate and practicality.

Auto-exposure algorithms attempt to (in real-time) estimate the optimal exposure for a given scene. Such methods could work well in cases where most of the scene has a relatively low dynamic range, and there are only some relatively bright/dark areas. However, even for a highly optimized algorithm, if a scene has two objects, one in a well-lit area and the other in a low-lit area, it would be impossible to select an exposure time where both would be visible. For example, Onzon et al. [1] implemented a neural network for automatic exposure selection. The proposed network, whose input is both the multi-scale histogram of the raw image and semantic feedback from a ResNet feature extractor, attempts to estimate the optimal exposure time for a given scene. By back-propagating the object detection loss through the (neural representation) of the ISP, end-to-end optimization was performed - where both the ISP and auto-exposure prediction network were simultaneously refined. The resulting object detection mean-average precision (mAP) at 0.5 IOU was 34 on a custom dataset. Such methods are an alternative to the use of HDRI in the deployment of vision applications in real-world applications.

Tone mapping operators (TMOs) take HDR images and map them into visually-Representative LDR images. Tonemapping methods fall into two categories: global tone mappers and local tone mappers. Global tone mappers apply the same compression function to all pixels in the image (e.g., gamma correction). Meanwhile, local tone mappers apply a per-pixel tone mapping based on the pixel neighborhood. Although global tone mappers are computationally more efficient, they do not preserve as much contrast, often resulting in washed-out-looking images. On the other hand, local tone mappers can preserve contrast ratios while also adhering to regional details. An example of such local TMOs is the Reinhard tone mapper [10]. Such algorithms can be used with HDRI to use existing trained object detection pipelines with minimal/no transfer learning needed.

End-to-end optimization of camera hardware and software attempts to optimize both the camera hardware and post-processing simultaneously. While the co-design of hardware and software lies at the heart of computational imaging, only recently have there been attempts to conjointly learn both the software and hardware [4], [5], [6]. New tools, such as automatic differentiation software and differential modeling of hardware, have enabled the conceptualization of the camera as a neural autoencoder (see figure 1). Here, the hardware essentially ”encodes” the incident scene before passing the measurements to the neural decoder through the bottleneck. This work aims to append this traditional framework with a frozen pre-trained object detection (YOLOv5s) network and use it to generate the ”object detection” loss. We then utilize this loss during the optimization process allowing for the improvement of high-order objectives.

Task-specific imaging is a rapidly growing field inside computational imaging that attempts to create specialized cameras that are optimized for a particular task. By optimizing the entire imaging pipeline for a single task, rather than opting for a more general approach, these methods have seen impressive improvements in their objective relative to their conventional counterparts. Great examples include hybrid electro-optic CNNs [**?**], deep optics [6]. Chang et al. [**?**] recently showed a fascinating example that performed such end-to-end optimization for image classification. Their work found that, by conjointly learning optics, a 2x the classification accuracy for the same power or ½ the power consumption for the same classification accuracy can be achieved.

Focal Plane Sensor-Processors - or focal-plane sensor-processors - co-locate both sensing and processing electronics in their pixels [24], [25]. While recent trends tend to focus on developing highly specialized, high frame-rate sensors [26], some sensors such as SCAMP-5 [28] have advanced the sensor′s programmability - and are therefore extremely adaptable and can be used for the research and implementation of a bevy of applications [4], [27]. Here, we introduce the use of such sensors for the task-specific conjoint learning of hardware software for object detection in real-world scenes. Our work demonstrates the use of this new class of sensors for task-specific optical sensing and imaging.

3 Proposed Method

One can conceptualize the image formation model (from the actual scene to the compressed and digitized image) in various ways. However, for end-to-end optimization, perhaps the most practical way of viewing a camera is as an encoder-bottleneck-decoder (neural auto-encoder). When addressing dynamic range issues in current imaging pipelines, the camera hardware - from the incoming HDR scene, all the way to the Analog to Digital (A2D) conversion - acts as a ”hardware encoder”. Before A2D, the analog signal effectively has a (noisy) unbounded image representation. At the A2D, the signal is clipped and quantized, usually to 8-12 bits, before being post-processed into the low dynamic representation we are used to seeing. In this regard, the A2D can be considered an information bottleneck, which restricts the amount of (dynamic range) information that can be transmitted and

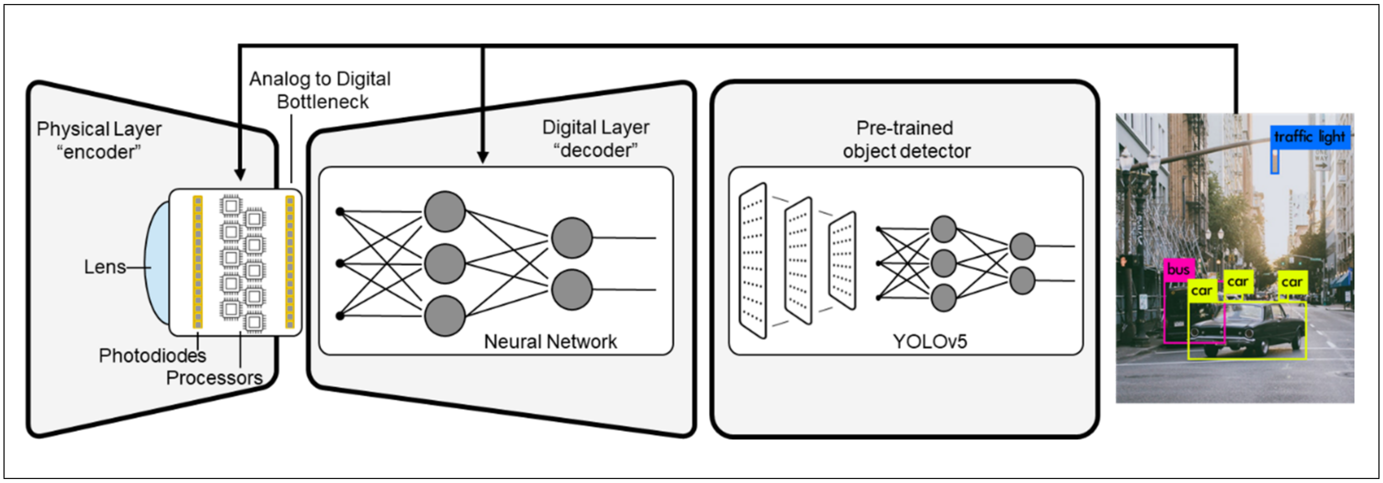

Fig. 1. Block diagram of the proposed method. An incident scene is focused on the focal-plane sensor-processors, where it is encoded ”physical encoding” before being passed through the bottleneck to the digital decoder (NN). Finally, the decoded image is passed to a frozen, pre-trained NN and the mAP loss is back-propagated to the relevant modules.

subsequently stored. The final part of the imaging signal-processing is the post-capture image processing, where the raw measurements are demosaiced, denoised, and tone mapped before compression (post-processing).

In our approach, we model the processing capabilities of the focal-plane sensor-processors, where the input to this model is a raw HDR image, and its output is the signal to be sent to the A2D. We then use Surrogate Gradients [23] in order to perform clipping and quantization of this signal before using a conventional CNN as a decoder. Finally, we use a frozen, pre-trained object detection network to perform object detection and produce the object-detection loss needed for back-propagation. This loss is back-propagated through the entire network and is used to train the hardware encoder and CNN decoder (see figure 1).

We tested a variety of different encoder-decoder pairs (see table 1), and compared their performance. Each time, we first train our entire model to try and reconstruct the corresponding Reinhard tone-mapped HDR image (with no clipping/quantization). Once converged, we used the trained weights for initialization, appended our model with the YOLOv5s network, and refined our encoder-decoder pair for object detection. We then can compare both the mAP values before/after Object Detection (OD) training and L1 losses before/after OD training.

TABLE 1
Sub-modules Architecture and Weight Initialization, bold=learned

Name	Encoder	Decoder
Raw Camera	Identity	Identity
Conventional Camera	Identity	sRGB
Optimal Camera	Identity	Rein()
Log Camera	Log	Rein(exp())
Gradient Camera	Grad(log())	Rein(exp(pois()))
Learned Log Camera	Log	**CNN**
Learned Gradient Camera	Grad(log())	**CNN**
Camera that CNNs	**1L-CNN**	**CNN**
log-CNN camera	**1L-CNN(log())**	**CNN**
programmable sensors	**Sensor model**	**CNN**

3.1 Hardware Encoding

The SCAMP-5 programmable focal-plane sensor-processors can implement a variety of hardware encoders [28]. Cameras with logarithmic response curves have been shown to have significantly improved dynamic range [18] [19]. Using SCAMP-5, it is possible to implement a logarithmic response curve, and we can therefore use this as a baseline comparison for more complicated encoders. As the HDR images were pre-normalized to [0,1], we utilized a gain*log(x+1) response for the hardware encoding, where the gain is a trainable parameter. We use this baseline to set a baseline mAP for our approach and dataset while evaluating different neural decoder network architectures.

The second hardware encoder implementation we test is a single CNN layer at the focal plane. Here, we used the same CNN architecture previously shown by Bose et al. [20] on a SCAMP-5 camera. Here, 16 5x5 filters were used, and subsequently, the 16 channels were weighted-averaged before quantization. No nonlinearity was used at the focal plane as it is not feasible to implement both the convolutions and a nonlinearity.

3.2 CNN Decoder

Neural tone-mapping is notoriously difficult as CNNs are ill-suited for this task. This structural misalignment stems from the wide pixel intensity distribution that such images have and the fact that the distribution of intensities can be very different image-to-image [17]. Furthermore, we would like the filter to be adaptive relative to both the global intensity distribution and a particular pixel's neighborhood. We initially opted to use the same CNN structure reported by Yang et al. [17], which we termed ”Deep-CNN” (DCNN) - see table 2. However, after some experimentation, we concluded that ”wide and shallow” CNNs (WCNN - see table 2) are better suited to learn local tone mapping operations, while ”deep and narrow” CNNs are better suited for global tone mapping operations. We believe this is because wider CNNs can sample more intensity values while deeper CNNs have a wider perceptual field and greater capability to model one-dimensional functions. We compared the relative

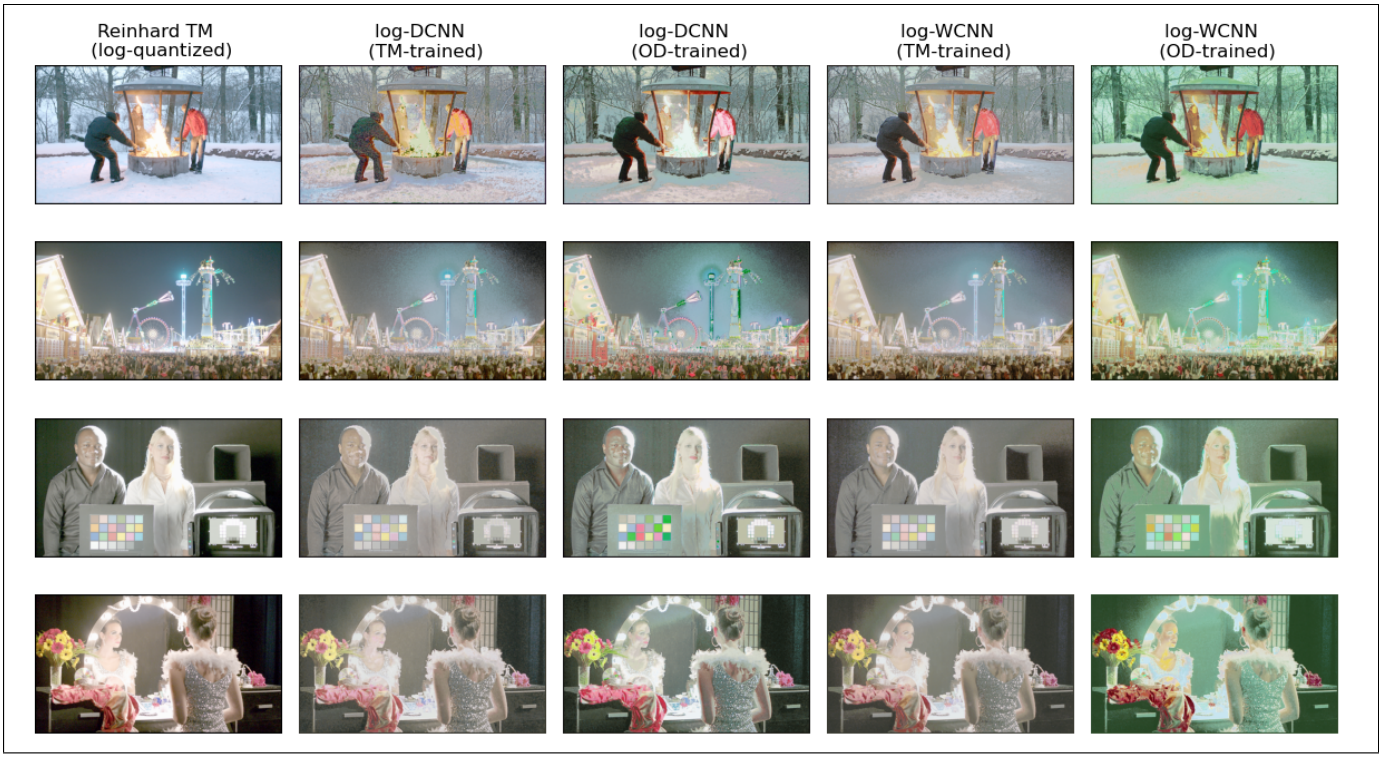

Fig. 2. Representative results for the log-camera implementation - where a logarithmic encoder was used before quantization and the decoder. Reinhard TM - a Reinhard tone mapper is used as the decoder. log-DCNN - a 10-layer "deep" CNN was used as the decoder. WCNN - a 5-layer "wide" CNN was used as the decoder. TM-trained is Tonemapped trained, i.e., after training the pipeline to reconstruct the Reinhard tone mapping operator (see left). OD trained - Object Detection trained, i.e., after performing refinement of the encoder-decoder pair with the YOLO network on the object detection loss.

performance of both CNNs with the logarithmic encoder experiment, including its capability to reproduce Reinhard and for Object Detection.

TABLE 2
Two CNN Architectures

Module	Layers	Input Size	Ker. Size	Ker. Num.
DCNN	input	W × H × 3		
	conv1	W × H × 3	3×3	32
	ReLU1	W × H × 32	-	-
	conv2	W × H × 32	3×3	32
	ReLU2	W × H × 32	-	-
	...	...	...	...
	conv9	W × H × 32	3×3	32
	ReLU9	W × H × 32	-	-
	conv10	W × H × 32	3×3	3
	sigmoid	W × H × 3	-	-
	output	W × H × 3		
WCNN	input	W × H × 3		
	conv1	W × H × 3	3×3	64
	ReLU1	W × H × 64	-	-
	conv2	W × H × 64	3×3	64
	ReLU2	W × H × 64	-	-
	...	...	...	...
	conv4	W × H × 64	3×3	64
	ReLU4	W × H × 64	-	-
	conv5	W × H × 64	3×3	3
	sigmoid	W × H × 3	-	-
	output	W × H × 3		

3.3 Object Detection Network

We performed end-to-end optimization of our encoder-decoder pairs by appending them with the YOLOv5s [7] object detection network. YOLOv5s is a state-of-the-art object detection model that takes in LDR images and outputs bounding boxes predictions and object classification probabilities. It is trained to maximize the probabilities of the ground truth objects in the image and predict the correct, tight bounding boxes. We use mean average precision (mAP), which is the average over multiple Intersection over Union (IoU) values for correct bounding box predictions as the loss for training. We froze the YOLOv5s network's weights during training, thus only updating our model (encoder-decoder pair). Ultimately, our model generates an LDR image given an HDR image, optimized especially for object detection. In contrast to most previously reported works, we do not attempt to perform tone-mapping to produce visually coherent images, *but instead, produce an image optimized to serve as input for object detection*. Our end-to-end approach can be seen in figure 1.

4 Experimental Results

All the experimental results we provide were created using the HDR4RTT [21] database. This database contains 4080 class-labeled HDR images with corresponding bounding boxes and segmentation maps, and we performed a 0.7/0.15/0.15 train/val/test split. Each HDR image was converted to linear raw if needed, and the top and bottom 1% pixels of each image were clipped to imitate some exposure control. The images were then re-scaled to [0,1].

4.1 Comparison to classical approach

There are several alternatives to our proposed method. Among them is the use of HDR imaging with tone mapping, utilization of auto exposure as described in related works, and logarithmic camera's with out-of-the-box tone mappers. As our dataset did not include HDR videos or images with motion artifacts, we could not create a fair comparison to the first two competing methods, as both would mostly be troubled by dynamic scenes.

However, the third benchmark can be tested. Here, we used an out-of-the-box (Reinhard) tone-mapper [10]. Then a camera with a log(x+1) responce was simulated, followed by A2D clipping & quantization. Post-processing entailed returning the image to the linear domain (using an exponential function) before using a Reinard(2.4,0,0.8,0.8) tone-mapping operator on the image. Using this approach had a mean average precision (mAP) of 26.4/11.7 @ 0.5/0.95, respectively (see table 3). When studying figure 2, we can qualitatively see that the output images are visually appealing, with little clipping/quantization visible. The mAP values reached using this method are actually the highest we recorded, showing that perhaps our learned decoders are not deep/wide enough to compete with this classical tonemapper.

4.2 Learned-Log Camera CNN Decoder Comparison

When attempting to train the encoder-decoder pairs for object detection directly, it was clear that loss propagation is problematic (and very slow), resulting in low mAP values and convergence to non-optimal local minimus. Therefore, we opted first to train our encoder-decoder pairs to reconstruct the Reinhard tone mapper using an L1 loss. This led us to use neural architectures reported to perform well in tone mapping tasks for our decoder network.

The CNN decoder is a critical part of our proposed method (see figure 1). We began by implementing the same CNN structure reported by Yang et al. [17] (see table 2), as this model has shown (some) capability of performing HDR image tone mapping. Briefly, a 10-layer CNN with 32 3x3 filters was used with batch normalization and ReLUs between all layers beside the output layer, which had a sigmoid nonlinearity to [0,1] bound our output image. We compare this ″deep″ CNN (DCNN) model to a ″wide″ CNN model (WCNN), where we used a 5-layer CNN with 64 filters/layer (see table 2).

Qualitatively, we found that the WCNN seemed to be more capable of representing a large number of images from the database compared to the DCNN (see representative results in figure 2). Quantitatively, the average L1 loss on the validation dataset was comparable (see table 3). After performing object detection refinement, it was evident that the WCNN outperformed the DCNN model. Qualitatively, images from the WCNN seemed to have fewer saturation artifacts (see figure 2, 2nd row, DCNN OD trained vs. WCNN OD trained). Quantitatively, we observed a 7-10 point improvement in mAP after training both networks (see table 3). As the WCNN decoder outperformed the DCNN decoder, we opted to use this architecture in future experiments. Visually, it is interesting to compare the output images before & after object detection training. We found that, in both the log WCNN & DCNN models, the color contrast is improved after OD training (although it was different from the Reinhard tone mapped image seen in the left of figure 2). The improved color contrast is especially evident when studying the color calibration curves located in the third row of figure 2.

4.3 CNN-Camera

Recently, Bose et al. [20] reported on the implementation of a single convolutional layer at the focal plane of a SCAMP-5 camera. Therefore, we opted to try and use the same 1-layer CNN they reported as our encoder. Briefly, a single convolutional layer with 16 5x5 filters was used, followed by weighted averaging the resulting 16 channels back to 3 channels before the clipping and quantization. Like the authors, we did not implement a nonlinearity on the focal-plane sensor-processors as it was not feasible to implement both a 1-layer CNN and a nonlinearity simultaneously. Like before, we first trained the encoder-decoder pair to reconstruct the Reinhard tone mapper, reaching an L1 loss of 0.087 (see table 3). This loss is greater than that of the log camera baselines of ˜ 0.08, which may be due to convergence to a less optimal solution. Visually, we can see much more quantization artifacts, and what seems like three distinct intensity resolution levels (see figure 3). When studying the encoder output (compared the the Reinhard TM image), it is quite evident that there is (green) discoloring - see section 6 (discussion) for the analysis of this phenomenon. Meanwhile, at output of the decoder, the image returns to a more ″natural″ coloring, however we can see three distinct intensity rings. We believe this results from how the intensity is encoded, where each color channel is used for a different range of intensities.

TABLE 3
Comparison of L1 and Object Detection loss before and after Object Detection training

Name	Avg. L1 loss		mAP @0.5/0.95	
Log camera	0.0987		26.4/11.7	
After training for:	TMO	OD	TMO	OD
Log-DCNN	0.074	0.16	9.2/4.1	19.2/8.5
Log-WCNN	0.084	0.13	13.2/6.5	20.8/9.5
Camera that CNNs	0.087	0.13	12.1/5.4	18.8/6.7

5 Discussion

The logarithmic hardware encoder outperformed the CNN encoder with all our evaluation metrics, both qualitatively and quantitatively. We believe this is due to how the logarithmic function compresses the incident scene's dynamic range. Like gamma correction, it effectively ″brightens″ the image before quantization, resulting in a more ″flat″ distribution of pixel intensities, resulting in better contrast (similar to histogram equalization). As can be seen in figure 4, the log-image histogram is flatter than the linear image histogram (keep in mind that y is in log-scale, so slight differences between the distributions are quite significant). Meanwhile, the CNN could only perform spatial operations and could not affect this intensity distribution before quantization as much. We tried multiple pre-training procedures

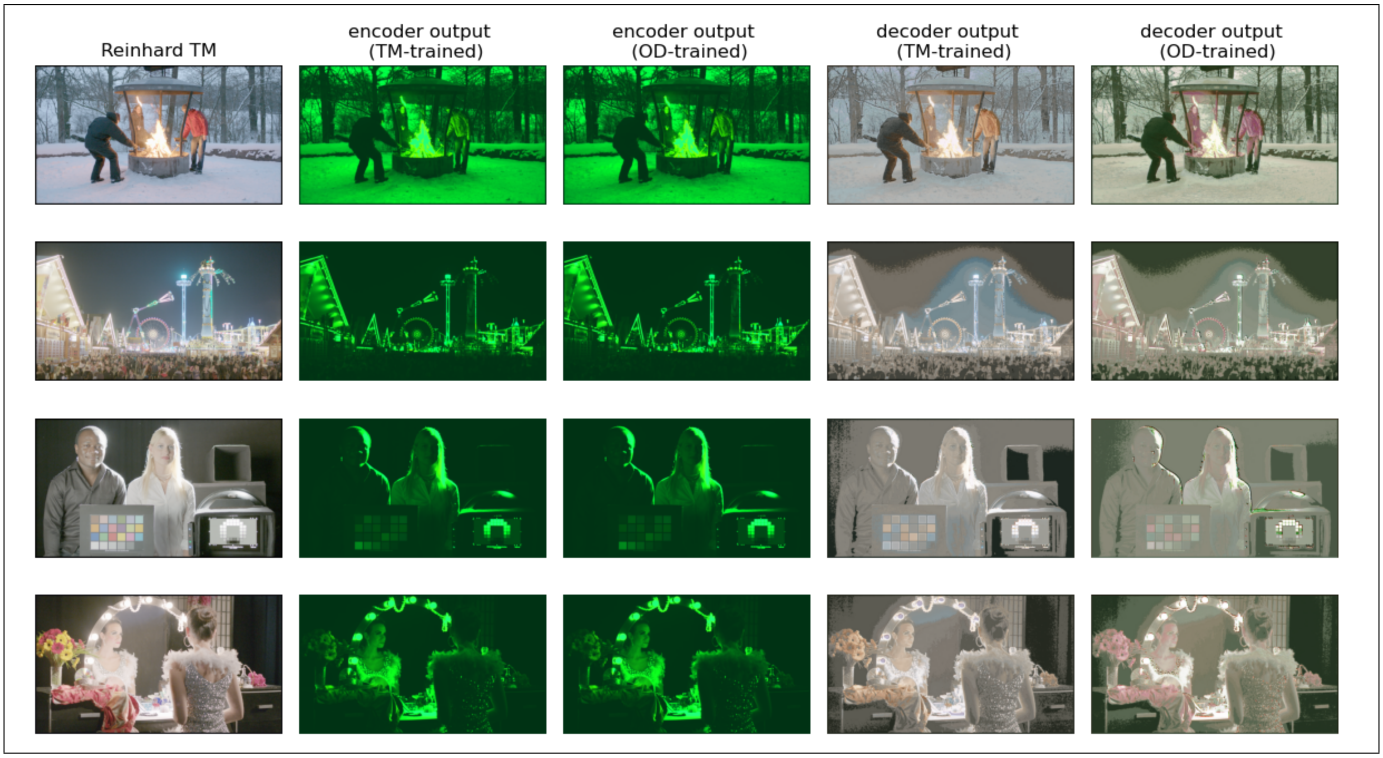
Reinhard TM
encoder output (TM-trained)
encoder output (OD-trained)
decoder output (TM-trained)
decoder output (OD-trained)

Fig. 3. CNN camera results.

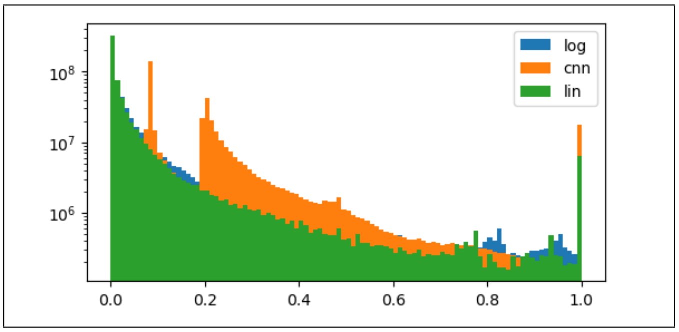

Fig. 4. Average picture histogram

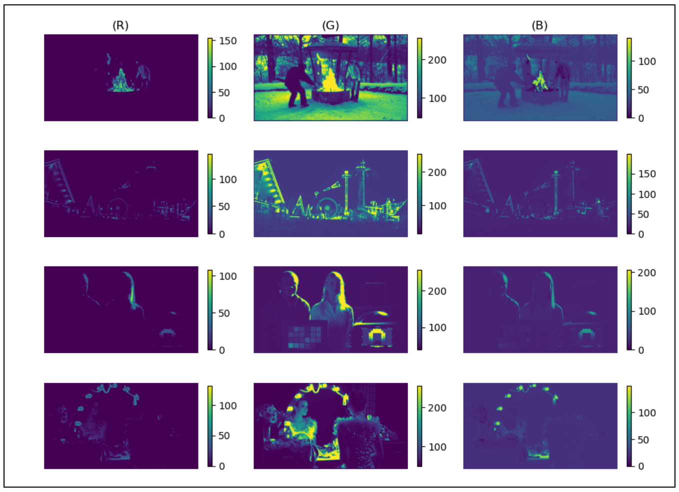

Fig. 5. CNN encoder output RGB channels

to push the CNN encoder to different solutions, such as pre-training it to perform image smoothing/sharpening, identity, HPF, and even Reinhard reconstruction. However, the best solution was reached after random initialization. When studying how the CNN encoder functions, it becomes clear that it utilizes different color channels to sample different average intensities. As can be seen in figure 5, it can be seen that each color channel is shifted relative to the other two, allowing for the improved sampling of the incoming scene. Now, when we study the results shown in figure **??** However, it is unclear whether this solution is convergent or simply how bias was initialized.

When comparing the two decoder CNN networks, it was apparent that, while the deeper model had more non-linearities allowing for the expression of more complex functions and a larger perceptual field, the wider CNN decoder outperformed the DCNN decoder by 1-3 points. We believe that this has to do with the fact that, as there are more filters in the ”wide” model, it can ”sample” more intensities than its deeper counterpart. Therefore, we concluded that it would be optimal to utilize the WCNN decoder going forward. In the future, additional architectures should be investigated, such as pyramid CNNs (CNNs that the number of filters/layer grows/shrinks with along the layer dimension), as well as U-nets. Finally, as discussed in section 4, the ”classical” method of using an empirically-derived out-of-the-box tone mapper outperformed all our encoder-decoder pairs. We believe that, while our methods had lower mAP, our method has significant potential when considering encoder models that we have not tested here. Better encoder models could result in more information preservation that could only be decoded using a neural architecture. It is also possible that our limitations on the

CNN depth/width were also too strict, and in the future we should relax them.

6 Future Work

Our model did not converge on many well-known encoders that one could test. For example the modulo [3] and the gradient camera [2] encoding. It would be interesting to pretrain our encoder to generate these encoders and then perform end-to-end optimization, as we would likely converge to different solutions to what we have seen here (the CNN camera should be able to closely approximate at least the gradient camera). See table 4.

An additional and exciting direction would be to develop a scene-aware exposure algorithm at the focal plane. As the processing and sensing are co-located at the focal plane, it is possible to introduce ways to influence each other. For example, one could add a threshold on the inpixel integration time that is dependent on the spatial features measured during the exposure (see [27]). This would increase the resulting adaptability of the entire framework and could be especially suited for HDR scenes. See table 4, ”Programmable Sensors” . Here, we could learn parameters like what convolutional filters are optimal and the optimal threshold parameter (which we could implement using the surrogate gradient method) and find the optimal encoding for object detection.

TABLE 4
Future work, bold=learned

Name	Encoder	Decoder
Optimal Camera	Identity	Rein()
Gradient Camera	Grad(Log())	Rein(Exp(Pois()))
Learned Gradient Camera	Grad(Log())	**CNN**
Mod Camera	Modulo	Rein(Unwrapping())
Learned Mod Camera	Modulo	**CNN**
Log-CNN Camera	**1L-CNN(log())**	**CNN**
Programmable Sensors	**Sensor model**	**CNN**

7 Conclusion

We presented a method and system for the rapid development and deployment of task-specific imaging with learned focal-plane processing. We believe that end-to-end learning of both hardware and software for a given task has the capability of bridging many of the inherent limitations of current imaging pipelines. Rather than developing the hardware and software separately, ultimately creating a ”one-size-fits-all” solution, task-specific imaging aims to create cameras that are learned end-to-end for a particular task. Cameras generated using this approach are not necessarily a camera in the traditional sense (as they do not necessarily report images) but more of a sort of visual optical sensing. For example, for object detection pipelines, the pipeline’s output is the object bounding boxes and classification rather than images. In this context, it is helpful to conceptualize these imaging pipelines as an autoencoder. Here, the ”camera” hardware physically encodes an incoming scene into the raw digital measurements; all the aspects of information loss (such as integrations, quantization, clipping, etc.) form the ”information bottleneck”, which is subsequently decoded by the digital post-processing. When combined with a high-order objective, like object detection, segmentation, posture estimation, and more - such pipelines need not produce *any* human-interpretable information (images).

There are many limitations to this framework, chief among them is the difficulty in optimization and generalization. In order to converge to optimal solutions, we often had to perform multiple pre-training procedures, where we separately trained different parts of our pipeline, trained our pipeline on different objectives, and more. However, a clear establishment of training protocols could help with such shortcomings. An additional limitation of this framework is its adaptability. Even though the object detection framework was learned on diverse data, it may behave unpredictably when encountering previously unseen data.

Acknowledgments

The authors would like to thank Suyeon Choi and Cindy M. Nguyen for providing the script for the surrogate gradients and Ultralytics for their YOLOv5s object detection network. We would also like to thank Prof. Wetzstein for his guidance and use of the SCI servers.

References

[1] E. Onzon, F. Mannan and F. Heide, ”Neural Auto-Exposure for High-Dynamic Range Object Detection,” 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2021, pp. 7706-7716, doi: 10.1109/CVPR46437.2021.00762.

[2] J. Tumblin, A. Agrawal and R. Raskar, ”Why I want a gradient camera,” 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’05), 2005, pp. 103-110 vol. 1, doi: 10.1109/CVPR.2005.374

[3] H. Zhao, B. Shi, C. Fernandez-Cull, S. -K. Yeung and R. Raskar, ”Unbounded High Dynamic Range Photography Using a Modulo Camera,” 2015 IEEE International Conference on Computational Photography (ICCP), 2015, pp. 1-10, doi: 10.1109/ICCPHOT.2015.7168378.

[4] J. N. P. Martel, L. K. Müller, S. J. Carey, P. Dudek and G. Wetzstein, ”Neural Sensors: Learning Pixel Exposures for HDR Imaging and Video Compressive Sensing With Programmable Sensors,” in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 42, no. 7, pp. 1642-1653, 1 July 2020, doi: 10.1109/TPAMI.2020.2986944.

[5] Chang, J., Sitzmann, V., Dun, X. et al. Hybrid optical-electronic convolutional neural networks with optimized diffractive optics for image classification. Sci Rep 8, 12324 (2018).
https://doi.org/10.1038/s41598-018-30619-y

[6] Metzler, C., Ikoma, H., Peng, Y., Wetzstein, G., Deep Optics for Single-shot High-dynamic-range Imaging, CVPR 2020.

[7] Redmon, Joseph, Santosh Divvala, Ross Girshick, and Ali Farhadi. ”You only look once: Unified, real-time object detection.” In Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 779-788. 2016.

[8] Lin, Tsung-Yi, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. ”Microsoft coco: Common objects in context.” In European conference on computer vision, pp. 740-755. Springer, Cham, 2014.

[9] He, Kaiming, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. ”Mask r-cnn.” In Proceedings of the IEEE international conference on computer vision, pp. 2961-2969. 2017.

[10] E. Reinhard and K. Devlin, ”Dynamic range reduction inspired by photoreceptor physiology,” in IEEE Transactions on Visualization and Computer Graphics, vol. 11, no. 1, pp. 13-24, Jan.-Feb. 2005, doi: 10.1109/TVCG.2005.9.

[11] Lin, Tsung-Yi, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. ”Focal loss for dense object detection.” In Proceedings of the IEEE international conference on computer vision, pp. 2980-2988. 2017.

[12] P. E. Debevec and J. Malik, ”Recovering high dynamic range radiance maps from photographs,” in Proceedings of the 24th annual conference on Computer graphics and interactive techniques. ACM Press/Addison-Wesley Publishing Co., 1997, pp. 369–378.

[13] S. W. Hasinoff, D. Sharlet, R. Geiss, A. Adams, J. T. Barron, F. Kainz, J. Chen, and M. Levoy, ”Burst photography for high dynamic range and low-light imaging on mobile cameras,” ACM Transactions on Graphics (TOG), vol. 35, no. 6, p. 192, 2016.

[14] S. W. Hasinoff and K. N. Kutulakos, ”Multiple-aperture photography for high dynamic range and post-capture refocusing,” IEEETrans. Pattern Anal. Mach. Intell., vol. 1, no. 1, pp. 1–17, Jan. 2009.

[15] M. D. Tocci, C. Kiser, N. Tocci, and P. Sen, “A versatile HDR video production system,” ACM Trans. Graph., vol. 30, no. 4, 2011, Art. no. 41.

[16] S. K. Nayar and T. Mitsunaga, ”High dynamic range imaging: Spatially varying pixel exposures,” in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), vol. 1, 2000, pp. 472–479.

[17] Yang2021DeepRL,Jie Yang and Ziyi Liu and Mengchen Lin and Svetlana N. Yanushkevich and Orly Yadid-Pecht, ”Deep Reformulated Laplacian Tone Mapping”, in ArXiv, 2021, abs/2102.00348

[18] M. Loose, K. Meier and J. Schemmel, ”A self-calibrating single-chip CMOS camera with logarithmic response,” in IEEE Journal of Solid-State Circuits, vol. 36, no. 4, pp. 586-596, April 2001, doi: 10.1109/4.913736.

[19] I. Hong, G. Kim, Y. Kim, D. Kim, B. -G. Nam and H. -J. Yoo, ”A 27 mW Reconfigurable Marker-Less Logarithmic Camera Pose Estimation Engine for Mobile Augmented Reality Processor,” in IEEE Journal of Solid-State Circuits, vol. 50, no. 11, pp. 2513-2523, Nov. 2015, doi: 10.1109/JSSC.2015.2463074.

[20] Bose, L., Chen, J., Carey, S. J., Dudek, P., Mayol-Cuevas, W. 2019. A Camera That CNNs: Towards Embedded Neural Networks on Pixel Processor Arrays. arXiv e-prints.

[21] https://hdr4rtt.inesctec.pt/

[22] Shinde, P. P., Shah, S. (2018). A Review of Machine Learning and Deep Learning Applications. 2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA). doi:10.1109/iccubea.2018.8697857

[23] SChristos Louizos, Matthias Reisser, Tijmen Blankevoort, Efstratios Gavves, Max Welling: Relaxed Quantization for Discretized Neural Networks. ICLR 2019

[24] A. Zarandy, Focal-Plane Sensor-Processor Chips. Berlin, Germany: Springer, 2011.]. While initially conceptualized for low-level image processing, they now provide some level of programmability [F. Paillet, D. Mercier, and T. M. Bernard, “Second generation programmable artificial retina,” in Proc. IEEE Int. ASIC/SOC Conf., 1999, pp. 304–309.

[25] W. Zhang, Q. Fu, and N.-J. Wu, ”A programmable vision chip based on multiple levels of parallel processors,” IEEE J. Solid-State Circuits, vol. 46, no. 9, pp. 2132–2147, Sep. 2011.

[26] L. Millet et al., ”A 5500FPS 85GOPS/W 3D stacked BSI vision chip based on parallel in-focal-plane acquisition and processing,” in Proc. IEEE Symp. VLSI Circuits, 2018, pp. 245–246.

[27] J. N. P. Martel, L. K. Müller, S. J. Carey and P. Dudek, ”Parallel HDR tone mapping and auto-focus on a cellular processor array vision chip,” 2016 IEEE International Symposium on Circuits and Systems (ISCAS), 2016, pp. 1430-1433, doi: 10.1109/ISCAS.2016.7527519.

[28] S. J. Carey, A. Lopich, D. R. Barr, B. Wang, and P. Dudek, ”A 100,000 fps vision sensor with embedded 535GOPS/W 256 256 SIMD processor array,” in Proc. Symp. VLSI Circuits, 2013, pp. C182–C183.