IEEE Open Journal of
Signal Processing

Received 20 October 2022; revised 5 January 2023; accepted 6 January 2023. Date of publication 25 January 2023;
date of current version 2 February 2023. The review of this article was arranged by Associate Editor Esteban Vera.

Digital Object Identifier 10.1109/OJSP.2023.3239819

ISP Distillation

ELI SCHWARTZ [1], ALEX M. BRONSTEIN [2] (Fellow, IEEE), AND RAJA GIRYES [1] (Senior Member, IEEE)

[1]Tel Aviv University, Tel Aviv 6997801, Israel
[2]Technion – Israel Institute of Technology, Haifa 3200003, Haifa, Israel

CORRESPONDING AUTHOR: ELI SCHWARTZ (e-mail: me@eli-schwartz.com)

This work was supported in part by the European Research Council under Grant ERC-StG 757497, PI Giryes, in part by the European Union's Horizon 2020 research and innovation programme under Grant 863839, PI Bronstein, and in part by the Israel Council For Higher Education - Planning & Budgeting Committee.

ABSTRACT Nowadays, many of the images captured are 'observed' by machines only and not by humans, e.g., in autonomous systems. High-level machine vision models, such as object recognition or semantic segmentation, assume images are transformed into some canonical image space by the camera Image Signal Processor (ISP). However, the camera ISP is optimized for producing visually pleasing images for human observers and not for machines. Therefore, one may spare the ISP compute time and apply vision models directly to RAW images. Yet, it has been shown that training such models directly on RAW images results in a performance drop. To mitigate this drop, we use a RAW and RGB image pairs dataset, which can be easily acquired with no human labeling. We then train a model that is applied directly to the RAW data by using knowledge distillation such that the model predictions for RAW images will be aligned with the predictions of an off-the-shelf pre-trained model for processed RGB images. Our experiments show that our performance on RAW images for object classification and semantic segmentation is significantly better than models trained on labeled RAW images. It also reasonably matches the predictions of a pre-trained model on processed RGB images, while saving the ISP compute overhead.

INDEX TERMS RAW image, object recognition, ISP, knowledge distillation, computational photography.

I. INTRODUCTION

Traditionally, Image Signal Processors (ISPs) are designed to optimize human perception of photographs. The core task is, given a raw measurement from an array of sensors (with different sensitivity to different light frequencies), to produce an image that looks natural to the human observer. To this end, there is a need to estimate the right colors and tones, and also compensate for acquisition process artifacts, such as noise.

However, in many domains, scenes are captured for machine consumption only. Examples include robot cameras, autonomous driving, and security cameras that are automatically monitored. In these domains, the objective is not to produce visually pleasing images, but rather to achieve high performance in a given downstream task, e.g. object recognition. Thus, discarding the ISP and training the model directly on the raw data is tempting in these cases. An ISP is typically a dedicated hardware system designed and optimized for low latency. Discarding it can reduce design costs, silicon area, and the power used. Unfortunately, simply discarding the ISP has been shown in the literature to cause performance degradation for high-level computer vision tasks such as object-recognition [5], [9], [13]. Note though that for lower-level tasks such as optical-flow it might be beneficial [33]). This happens because the ISP serves as a 'normalization' of the data, transforming it into a canonical space, which is independent (or less dependent) of the camera used or the capturing environment. Solutions discussed in the literature usually involve adjustments of the ISP for the vision task, either manually [31] or via learning [30]. Unlike alternative methods that focus on designing a minimal ISP for vision, in this paper we focus on completely discarding the ISP. We explore how the drop in performance can be mitigated.

Training a model on RAW images requires annotating the data. Human labeling of RAW images is impossible, as they are sometimes almost unrecognizable to humans. The alternative is to use RGB images for labeling and then transfer the labels to RAW. The transfer to RAW can be done by having an inverse model of the ISP (ISP_{inv}). So given a dataset of labeled

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

RGB images $\{(RGB_i, l_i)\}$, we can generate a matching labeled RAW dataset $\{(RAW_i, l_i)\} = \{(ISP_{inv}(RGB_i), l_i)\}$ relying on pixel alignment between RAW and RGB to get the labels for the RAW [5], [9], [13].

In this work, instead of using an ISP inverse model for transferring labels from RGB to RAW, we employ a dataset of RAW-RGB pairs. Building such a dataset is relatively easy since it does not require human labeling and by labeling the RGB images we immediately get labels for the RAW. Also, we wish to reduce the labeling cost. Thus, instead of manually labeling the RGB images we use a pre-trained model to label them.

With the above-mentioned dataset we can train our model on RAW images with the transferred labels as ground truth. Yet, we still suffer from a big drop in performance compared to working with RGB inputs. To mitigate that, we suggest using Knowledge Distillation (KD) [16] to make the RAW predictions fit the RGB predictions. KD is a technique that is known to work quite well for compressing deep models, i.e., making a smaller model behave similarly to a larger model. Here we use KD for compressing both the heuristically designed ISP *and* a higher-level deep vision model, e.g. classifier, into a new deep model for classification but with the size of only the latter part. We show the advantage of our approach also when training the network to have similar predictions for short-exposure RAW images. In that case, in addition to the ISP and classification, the model also compresses the 'knowledge' of an ideal (non-existent) denoiser that maps the short-exposure RAW to a longer-exposure RAW image. We show that this technique works well for both classification and semantic segmentation.

II. RELATED WORK

ISP for Vision: The ISP consists of a set of algorithms, usually applied sequentially, intended for transforming a RAW image into a visually appealing RGB image. The different steps either fix some degradation in the acquisition process (e.g., noise) or just transform the image to better fit human perception (e.g., tone mapping). Recently, some works suggested replacing some or all of these operations with a learned model [6], [26]. However, these designed or learned ISPs are optimized for visual appearance and not the vision tasks.

Simply dropping the ISP does not work well. Several works used simulated RAW images to train a classifier and observed a substantial gap in accuracy [9]. Hansen et al. [13] report a larger gap for smaller models, ∼16% drop for MobileNet, attributed to the failure of compact models to compensate for the lack of ISP. Buckler et al. [5] identified the lack of demosaicing and gamma correction to be a critical cause of performance degradation. They suggested modifying the imaging sensor such that demosaicing and gamma correction are no longer necessary, which limits the effect of the lack of ISP.

Some methods suggest optimizing the ISP for downstream vision tasks. [21], [28], [31] suggested tuning the parameters of a traditional (not learned) ISP to improve downstream tasks. Sharma et al. [27] add a component that takes an RGB image processed by the ISP and further enhances it for the downstream task. Diamond et al. [9] suggested jointly learning a low-level processing module, which performs denoising and deblurring, with the classifier. They train with simulated RAW images. Wu et al. [30] suggested VisionISP, a trainable ISP, which is trained to optimize object detection in an autonomous driving setting. Wang et al. [29] suggested a totally different approach for performing classification on low-quality images (e.g. fog or low-contrast).

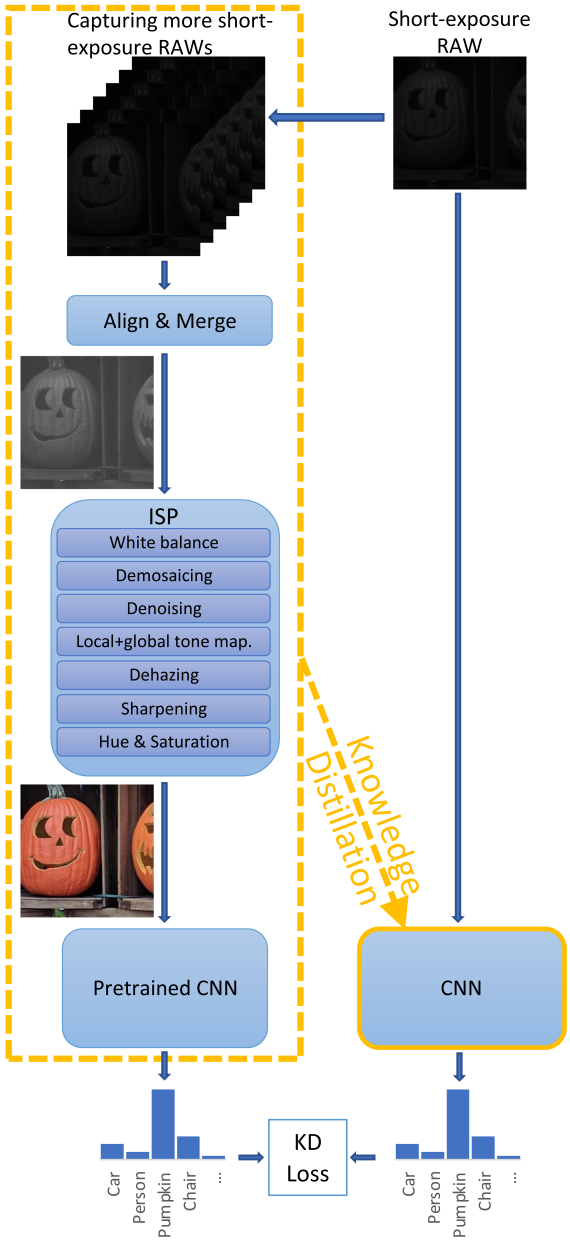


FIGURE 1. ISP Distillation: Distilling the knowledge of the ISP algorithm together with the pretrained CNN (e.g. classifier or semantic segmentation) that follows it improves the performance of a CNN applied directly to RAW images. Optionally, the CNN can operate on short-exposure RAW images, effectively distilling the information in the physical process of acquiring a signal with better SNR.

They use a model pre-trained on high-quality images and learned a mapping between the deep representations of the low-quality images and the high-quality images. Essentially, unlike other methods, they perform domain adaptation by mapping the outputs of the classifier rather than the inputs. Other methods for classifying low-quality images include [3], [18], [32]. In this work, instead of designing an image processing module, i.e., modifying the input image for a certain task, we focus on applying the vision model directly to RAW data.

Knowledge Distillation: Compressing larger models into smaller ones was first suggested by Bucilua et al. [4]. Application of this technique to deep neural networks, known as Knowledge Distillation, was suggested by Hinton et al. [16]. The key idea behind KD is that the soft label output (or soft probabilities) of a classifier contains much more information about the data point than the hard label. KD has been widely used for numerous different applications, e.g. [2], [19], [22]. It has also been shown to work for distilling knowledge of non-neural network machine-learning models [11]. In [8], [12] KD was used to transfer knowledge across different types of sensors for either image reconstruction or high-level tasks, and in [17] it was theoretically analyzed. In this work, we use KD to distill not just a deep model, but also the non-learned (manually engineered) algorithms in the ISP and the information gained in the physical process of acquiring a better signal (that has a better SNR).

III. METHOD

We are interested in training a classification model to operate on images from one modality, when no semantic labels are available. What we have is a function that maps the images to another modality for which we have a pre-trained model. Alternatively, we might have a dataset of image pairs from two different modalities with pixel alignment, even if a function that maps these modalities does not exist (or we do not have access to it). In our case, the two modalities are a *RAW* and a processed image $RGB = ISP(RAW)$. We are also interested in operating on a short exposure image, RAW_{short}, where the reference processed image is based on a longer exposure, $RGB_{long} = ISP(RAW_{long})$, or multiple short exposure RAW images, $RGB_{long} = ISP(\{RAW^i_{short}\})$ (clearly, there is no deterministic mapping f such that $RGB_{long} = f(RAW_{short})$).

Manually labeling RAW images is hard as they are almost impossible to be recognized by humans. The alternative is to use RGB images for labeling and transfer the labels to RAW. The transferring to RAW can be done by having an inverse model of the ISP (ISP_{inv}). So given a dataset of labeled RGB images $\{(RGB_i, l_i)\}$, we can generate a matching labeled RAW dataset $\{(RAW_i, l_i)\} = \{(ISP_{inv}(RGB_i), l_i)\}$ relying on pixel alignment between RAW and RGB to get the labels for the RAW. But the ISP is not invertible and thus the inverse model is just an estimation and due to the one-to-many mapping should be stochastic. Also, designing the inverse model is not trivial. In this work, instead of having an inverse model of the ISP for transferring the labels from RGB to RAW, we use a dataset of RAW-RGB pairs. Moreover, to reduce the labeling cost, instead of manually labeling the RGB images, we use a pretrained model, M_{RGB}, to label them.

We argue that training on the dataset $\{(RAW_i, l_i)\}$, where l_i are the hard labels (e.g., an integer value or a one-hot encoding), with the cross-entropy (CE) loss is not the best option. Instead of using l_i, we use the soft probability distribution vector over the classes, p_i, predicted by a pretrained model. Following many works that have shown its benefits, we use the KD loss.

Given the probability vectors $p = M_{RGB}(RGB)$ and $q = M_{RAW}(RAW)$, which are the outputs of a softmax layer with temperature T. A softamx with a temperature T for a logit vector x is defined as

$$p_i = \frac{e^{x_i/T}}{\sum_{k=1}^{K} e^{x_k/T}}. \tag{1}$$

The temperature T controls the "softness" of the predicted distribution. For smaller T we get a low-entropy vector (an approximation of one-hot), while for higher T we get a high-entropy vector (the probability is distributed more evenly between all classes). The KD loss is given by

$$L_{KD} = -\sum p_i \cdot \log(q_i). \tag{2}$$

This loss simultaneously distills the information from the heuristically designed ISP and the CNN model (classifier) pretrained on RGB images, M_{RGB}.

An extension of the KD loss that can be beneficial is adding an ℓ_2 loss on the intermediate representation vectors [23]. We found this extension to be useful in our case too. We use the extended form of the loss

$$L_{KD} = \ell_2(f_{RGB}, f_{RAW}) - \sum p_i \cdot \log(q_i), \tag{3}$$

where f_{RGB} and f_{RAW} are the intermediate representations of the RGB image from M_{RGB} and the RAW image from M_{RAW}. Unlike [23] where the representation is captured in the middle of the network, we use the representation from the last layer before the classifier. This is more likely to produce aligned probability predictions, even without balancing the CE loss and the features loss magnitudes.

In the case of short-exposure RAW, we use

$$p = M_{RGB}\left(RGB_{long}\right) \tag{4}$$

$$q = M_{RAW}\left(RAW_{short}\right), \tag{5}$$

and we also implicitly learn to classify low-dynamic-range and extremely noisy images. As commonly done, we use a linear combination of the Cross-Entropy (CE) loss and KD loss

$$L = \alpha L_{CE} + (1-\alpha) L_{KD}, \tag{6}$$

Where the CE loss is defined as

$$L_{CE} = -\sum p_i \log(y_i) \tag{7}$$

(y is the ground-truth one-hot probability vector). In our experiments, we chose $T = 4$ (similar to [16]) and we heuristically chose $\alpha = 0.9$ to balance the two loss terms (same order of magnitude).

We also extend the method to the semantic segmentation task. Where the class label (or class probabilities) is predicted at each pixel or spatial location. In this case, the loss is averaged over all spatial locations at the output layer (which might be different from the input resolution). In the case of spatial elements equal to N and $L_{CE}(i), L_{KD}(i)$ the loss functions at location i, the final loss is defined as:

$$L = \frac{1}{N}\sum_{i=1}^{N} \alpha L_{CE}(i) + (1-\alpha)L_{KD}(i). \tag{8}$$

It is common practice to normalize the inputs so each channel has zero-mean and unit STD (calculated on the relevant training set). For the RAW images, we do the same, where the mean and STD are calculated separately for R, G, and B pixels in the Bayer pattern.

In practice, for faster training, we initialize the RAW classifier, M_{RAW}, with the weights of the pre-trained RGB model M_{RGB}. Using this initialization, we can resort to a short training (4 epochs in our experiments). Since RAW images have a single channel and not 3 RGB channels, we need to make some adaptations to be able to use off-the-shelf classifiers with them (especially when initializing with pre-trained models). We do so in a very simple way by transforming the RAW images to RGB and filling in the missing values using bilinear interpolation, similar to what is done in [26]. There is no loss of information in bilinear interpolation. The original pixel values are unchanged, and the interpolation only fills in the missing values introduced by the conversion from the 1-channel Bayer pattern to RGB.

IV. EXPERIMENTS

To validate our approach we used two test cases. In the first, we test performance when operating on noisy and mosaiced images, i.e., discarding the denoising and demosaicing preprocessing. In the second, we test performance when the full ISP is discarded.

A. DISCARDING DENOISING AND DEMOSAICING

We first test our method on synthetically generated RAW images, since for these images we can compare to the classification performance of training with ground truth labels. In this experiment, we limit the simulated ISP (we want to discard) to include only denoising and demosaicing. Thus, the RAW images are the RGB images sub-sampled according to a Bayer pattern with added Gaussian noise. We use the ImageNet dataset for this experiment (ILSVRC2017 [24]).

Table 1 presents ResNet18 [15] and MobileNetV2 [25] results for the mosaiced and noisy ImageNet validation set, with noise STD equals to 0.1 (pixel values in [0,1]). Under such a distortion, the performance of ResNet18 drops from 69.76% top-1 accuracy to 29.23%. Training the model on distorted images (RAW) using either ground truth labels or hard labels produced by a pretrained ResNet18 (based on the clean RGB images) improves performance to ∼57%. Using the proposed ISP Distillation we improve the performance by more than 4%. Similar trends are observed for MobileNetV2 too. Fig. 3 shows the improvement is consistent across noise levels, including for $STD = 0$ where it is just demosaicing.

TABLE 1. Classification Performance on Noisy and Mosaiced Images. We Use the ImageNet Images in a Color Filter Array Format With Added Synthetic Gaussian Noise With STD = 0.1

	ResNet18		MobileNetV2	
	top-1	top-5	top-1	top-5
Clean images (upper bound)	69.76	89.08	71.88	90.29
Baselines:				
Pretrained model	29.23	52.53	25.65	48.73
Trained w/ GT labels	57.21	80.76	56.31	80.06
Trained w/ pred. labels	56.59	79.70	56.73	80.15
ISP Distillation (ours)	**62.46**	**84.41**	**62.92**	**84.84**

B. DISCARDING THE FULL ISP

To test the effect of discarding the full ISP, we use real data based on images from the HDR+ dataset [14]. This dataset includes 3640 bursts (containing 28461 images in total). Each burst has between 2 to 10 short-exposure raw photos, where each is generally 12-13 Mpixels, depending on the type of camera used for the capturing. The images in a burst are generally captured with the same exposure time and gain. The dataset also provides for each burst a merged RAW image that is generated by aligning the short-exposure RAW images and combining them to produce a single high-dynamic-range RAW. This merged RAW is then processed by their ISP to produce the final RGB image. See [14] for more details.

We performed two kinds of experiments. In the first, we want to distill the ISP and a pre-trained classification model. Thus, we use the merged RAW as input to our model, trying to mimic the predictions on the final RGB. In the second experiment, we choose a single short-exposure RAW and train our model to mimic the predictions made by the RGB model on the final RGB, *produced from the merged RAW*. We always choose the short-exposure RAW that is pixel-aligned with the merged RAW (information provided in the dataset).

For evaluation we used both a larger pseudo-label dataset and a smaller manually labeled dataset. For the pseudo-labeled dataset the top-1 and top-5 accuracies are measured as the agreement of our model with the predictions of the pre-trained classification model on the final RGB. The bursts are randomly split into 80% training set and 20% test set. The images in the original HDR+ dataset are of very high resolution, but popular classifier architectures expect images to be in the range of $200 - 300$ pixels. While we could down-sample the images, it would have removed the effect of the Bayer pattern (and potentially the effect of the noise too), and we are

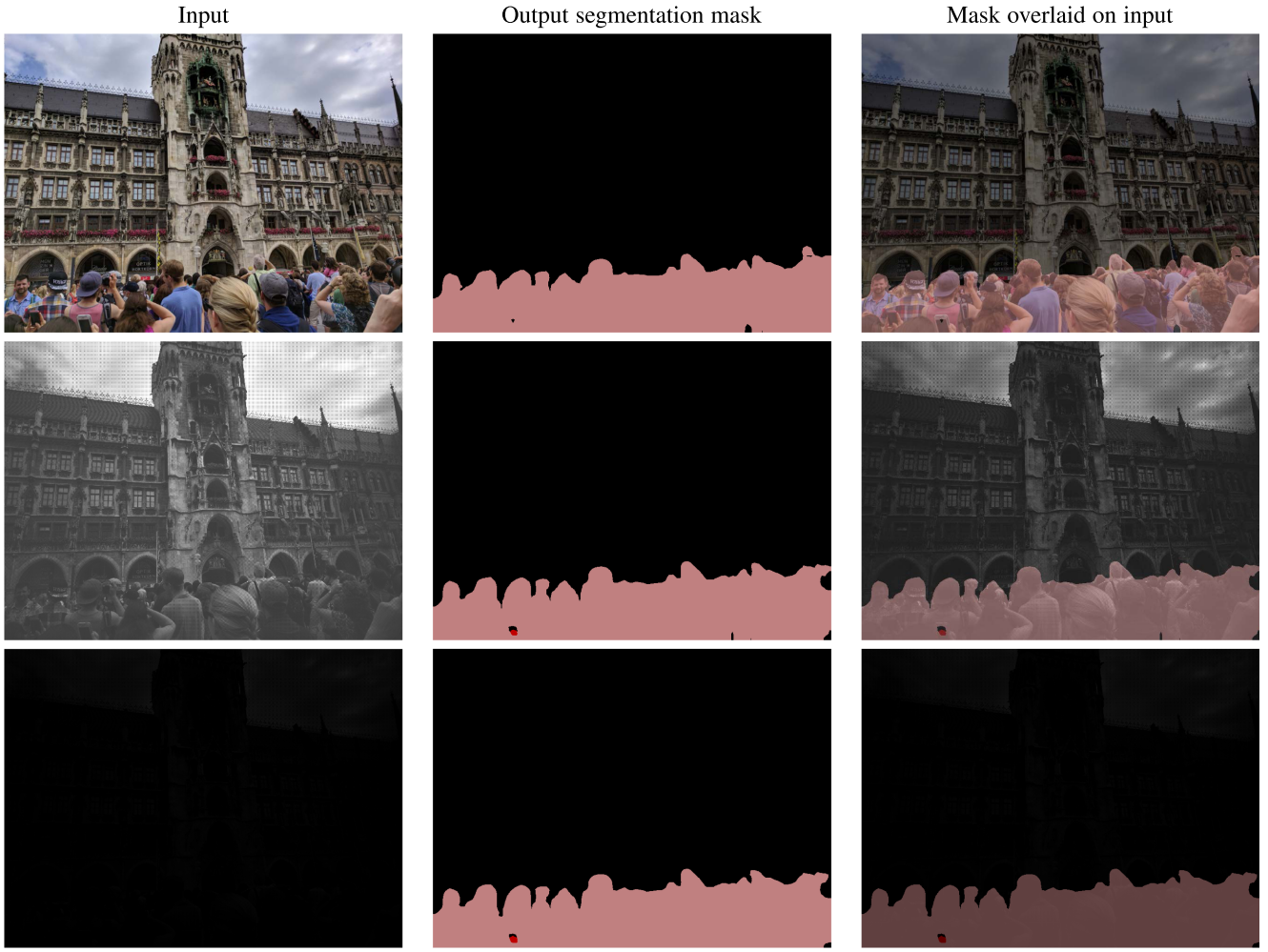


FIGURE 2. **Semantic segmentation result example: Top row is the ground truth RGB results; middle row is our results for RAW input images; bottom row is our results for short-exposure RAW images.**

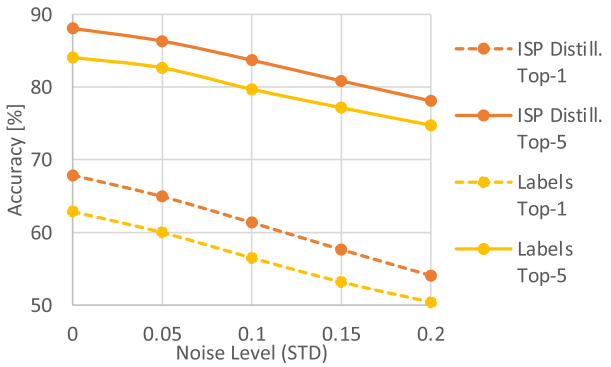


FIGURE 3. **Performance vs. noise level for noisy mosaiced ImageNet with ResNet18.**

interested in understating the ability of the model to overcome these artifacts. Therefore, we chose to split the images into smaller 256×256 non-overlapping patches. For training, we use all the patches originating from the bursts in the training set. For the test, since many of the patches do not contain an object, we only tested those for which the pre-trained RGB classifier predicted an object with probability $p > 0.8$. The manually labeled test set contains 211 images out of the pseudo-labeled dataset test set described above.

Table 2 compares the performance of training the RAW model with predicted labels vs. applying ISP Distillation. Our approach shows a substantial improvement of $+9\%$ top-1 accuracy for ResNet18 and $+6\%$ for MobileNet V2. It also exhibits a similar advantage for the model trained on short-exposure RAW, where the improvement is $+8\%$ and $+11\%$ for ResNet18 and MobileNet V2, respectively. A noticeable advantage exists in all experiments for top-5 accuracy too, bringing the accuracy to $97 - 98\%$. This suggests that the RAW model predicted probability distribution highly matches the one from the RGB model. Table 3 compares the performance on the small manually labeled test set. While the results are a bit lower compared to the pseudo-labeled test set, we observe similar trends.

For the short-exposure RAW experiment, we also compared to a sequential combination of a network that performs the

TABLE 2. Performance When Discarding the Full ISP and Training the Classifier on RAW Images (Or Short-Exposure RAW). Measuring the Agreement Between the RAW Model and Pre-Trained RGB Model Predictions on an Automatically Collected Test Set. ISP Distillation Consistently Outperforms Training With Labels (Labels Predicted on RGB). Note That ISP Distillation Achieves Competitive Results to the Computationally Demanding Solution of Using an ISP network [26] Prior to the Classifier

	ResNet18		MobileNet V2	
	top-1	top-5	top-1	top-5
Classifying RAW images:				
Training with labels	84.73	95.40	87.16	97.37
ISP Distillation (ours)	**93.83**	**98.12**	**93.32**	**98.08**
Classifying *short-exposure* RAW images:				
Training with labels	83.73	95.31	80.62	94.48
ISP Distillation (ours)	**91.87**	**96.98**	**91.57**	**96.99**
With ISP (DeepISP [26])	91.86	97.67	92.30	97.71

TABLE 3. Performance When Discarding the Full ISP and Training the Classifier on RAW Images (Or Short-Exposure RAW) Measured on a Manually Labeled Test Set (211 Images). ISP Distillation Consistently Outperforms Training With Labels (Labels Predicted on RGB)

	ResNet18		MobileNet V2	
	top-1	top-5	top-1	top-5
Classifying RAW images:				
Training with labels	82.93	93.83	83.88	94.78
ISP Distillation (ours)	**90.99**	**94.78**	**90.99**	**96.20**
Classifying *short-exposure* RAW images:				
Training with labels	81.99	92.89	80.09	91.46
ISP Distillation (ours)	**90.04**	**94.78**	**89.09**	**95.26**

TABLE 4. Accuracy Vs. FLOPs. Our Classifier-Only Model Reach Comparable Results to Training an ISP and a Classifier While Having Much Lower Computation Cost (FLOPs)

	FLOPs	top-1	top-5
DeepISP + ResNet18	4.5G	91.86	97.67
ResNet18	2G	91.87	96.98
DeepISP + MobileNetV2	2.8G	92.30	97.71
MobileNetV2	0.3G	91.57	96.99

ISP part (DeepISP [26]) before the classifier, where both are trained end-to-end to optimize classification performance. The computational costs vs. accuracy are summarized in Table 4. DeepISP flop count is ∼2.5 G (compared to ResNet18's 2 *G* and MobileNetV2's 0.3 *G*). Note that the classifier alone, trained with ISP Distillation, performs almost as well as the combination of models, which is more computationally demanding. The expressive power of the joint ISP and classifier is probably limited by the constraint of having a 3-channel tensor at their interface. When training just the classifier, using our approach, both models are mostly compressed into the classifier alone.

TABLE 5. Semantic Segmentation Performance. MIOU is Computed as Agreement With the Pretrained Model Predictions for the Corresponding RGB Image

	mIOU
RAW images:	
Training with labels	92.47%
ISP Distillation (ours)	**94.98%**
***short-exposure* RAW images:**	
Training with labels	91.88%
ISP Distillation (ours)	**94.51%**

TABLE 6. Finetuning Only the First Layers. Most of the Gain is Due to Finetuning the First and Second Quarters of the Model. Tested With ResNet18 on Short-Exposure RAW

	top-1
Training the first quarter of the model	90.61
Training the first half of the model	91.66
Training Full model	**91.87**

C. SEMANTIC SEGMENTATION

We tested the extension to semantic segmentation, where the KD loss is applied at each spatial location, as mentioned in Section III. For this experiment we used the DeepLabV3 model [7] with ResNet101 backbone [15]. The model used for the RGB images was trained on Pascal VOC dataset [10] and a subset of the COCO dataset [20] with images containing objects from the 20 classes in the VOC dataset. As in the classification case, the same pretrained weights were used to initialize the model operating on RAW images.

Figs. 2 and 4 provide representative examples of the semantic segmentation results. To quantitatively assess the model performance we compute the Mean Intersection Over Union (mIOU) between the pretrained RGB model outputs and our RAW model (i.e. the agreement between the RGB outputs and the RAW outputs). The quantitative results are summarized in Table 5. We observed improvement over the training with labels baseline of ∼ 2.5% mIOU for both RAW images and short-exposure RAW images. This is a significant improvement, but not as big as the improvement for classification. We suspect that the per-pixel output adds higher-level information providing better guidance to the student and thus the effect of having the soft-labels rather than the hard-labels is smaller.

D. ABLATION STUDIES

Partial model finetuning: Since we initialize our RAW model with ImageNet pre-trained weights we want to test how many of the layers need to be adapted to accommodate the RAW input. Is it just about local artifacts and finetuning only the first layers will be enough or there are global distortions that require higher-level features from deeper layers to adapt too? Table 6 shows that indeed training just the first layers is beneficial, most of the gains are thanks to the finetuning of the first

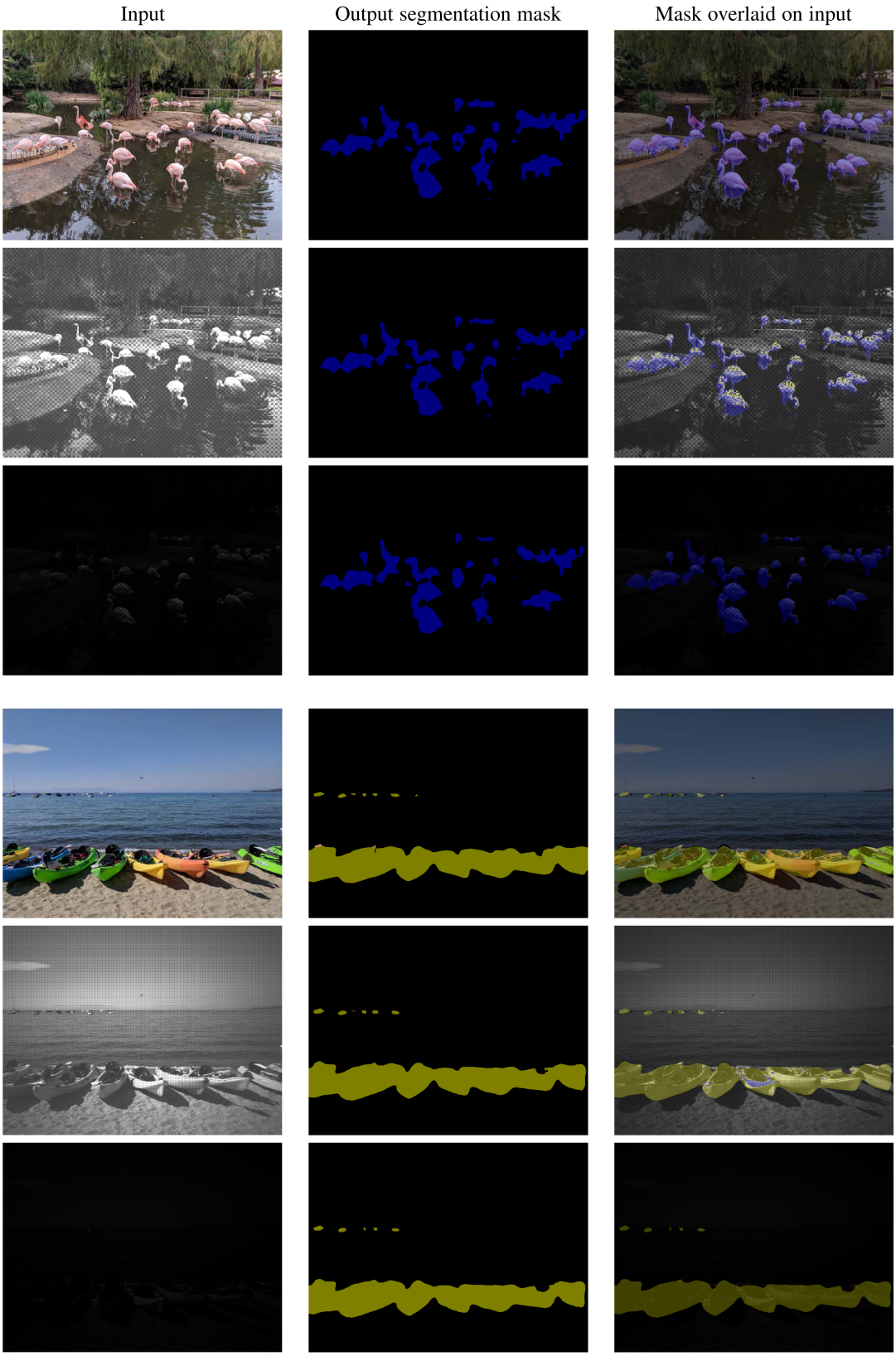


FIGURE 4. **Additional semantic segmentation result examples.** For each example the top row is the ground truth RGB results; the middle row is our results for RAW input images; the bottom row is our results for short-exposure RAW images.

TABLE 7. Ablations. Performed on the ImageNet Experiment. Removing the Feature Loss Results in a Drop in Performance. Using Stronger Teacher (ResNet50) Improve Performance

	ResNet18		MobileNetV2	
	top-1	top-5	top-1	top-5
ISP Dist. (Ours)	**62.46**	**84.41**	**62.92**	**84.84**
ISP Dist.-feat. loss	61.35	83.70	61.43	83.95
ISP Dist.-feat. loss+ResNet50	62.43	84.18	62.62	84.14

and second quarters of the network. Training the first half of ResNet18 is almost as good as training the full model.

Features loss and stronger teacher: Knowledge Distillation was originally used for distilling knowledge from stronger (more parameters, deeper) models to weaker ones. We verify performance can be further improved by using a stronger teacher in our case. We test the case where the teacher architecture is ResNet50 and the student is either ResNet18 or MobileNet V2. The teacher and student models' feature dimension is different, and even if it was the same since they are pretrained independently points in the two unaligned embedding spaces cannot be compared. For this reason, we drop the $\ell 2$ loss on the features in this case. Table 7 shows that dropping the features loss results in a drop in about 1% drop in performance. Adding the stronger teacher improves the performance so it is comparable to the effect of features loss. In all our experiments we use the features loss. It might be possible to get even higher performance by combining the features loss with a stronger teacher (by having the same features dimension and enforcing embedding space alignment).

E. THINGS THAT DID NOT WORK

For completeness, we also report ideas we hypothesized should improve performance but empirically did not.

Gradual blending: We are trying to solve the domain adaptation problem from RGB to RAW. Unlike the usual domain adaptation task, in this case, we have pixel-aligned pairs of images from both domains. Since we have pixel alignment, we can generate infinitely many intermediate domains by interpolating between the RAW and RGB images, i.e. we can define a set of new domains D_α such that the images in D_α are $\{\alpha RGB_i + (1-\alpha)RAW_i\}$. This approach was used by [1] for the case of shifting from the classification of images where the background is masked out to the classification of the same images where the background is not masked. We expected that gradually (linearly) shifting between the domains, i.e. first training the model to classify from domains with small α and finishing with $\alpha = 1$, would help the model to learn better. But, in our experiments, we found the model finally converged to a similar performance as directly training for $\alpha = 1$.

Localized distillation: We observed in the case of semantic segmentation that even training with hard labels yields very strong results. We hypothesized that the localized semantic signal helps the student learn better by providing more information. To test it we performed the experiment of distilling both the ISP and a classifier (similarly to Section IV-B) but requiring spatial consistency between the teacher and student. To keep the spatial information we removed the global pooling layer and replaced the fully connected layer with a 1×1 convolution layer (with the same weights). We then applied the KD loss at each spatial location. However, in our experiments, this didn't improve the results beyond what we got with the 'global' KD loss.

V. CONCLUSION

We have shown that it is possible to distill the knowledge of not just pre-trained models but also heuristically designed ISP, to improve the performance of classification and semantic segmentation models for RAW images. We have also shown improvement for short-exposure RAW, distilling the information of the physical process of acquiring a better signal. Our proposed ISP distillation is a step towards reaching similar performance on RAW images compared to RGB in high-level vision tasks. It can advance the deployment of vision models on RAW images in domains where the images are consumed by machines and not humans. This will save the computing cost of the ISP.

Possible future directions include:

- Combining ISP-Distillation with ISPs designed for vision tasks, i.e. distilling an ISP that was optimized for a vision task into the high-level model.
- Exploring hybrid solutions where some blocks of the ISP are distilled into the high-level model and removed from the pre-processing pipeline while others (low compute) are kept.
- Extending ISP Distillation to other vision tasks (e.g. detection).
- Exploring using this method for other sensors that might better fit the needs of a machine vision system (e.g. event cameras).

REFERENCES

[1] M. Arar, N. Fish, D. Daniel, E. Tenetov, A. Shamir, and A. Bermano, "Focus-and-expand: Training guidance through gradual manipulation of input features," 2020, *arXiv:2007.07723*.

[2] H. Bagherinezhad, M. Horton, M. Rastegari, and A. Farhadi, "Label refinery: Improving imagenet classification through label progression," 2018, *arXiv:1805.02641*.

[3] T. S. Borkar and L. J. Karam, "DeepCorrect: Correcting DNN models against image distortions," *IEEE Trans. Image Process.*, vol. 28, no. 12, pp. 6022–6034, Dec. 2019.

[4] C. Bucilua, R. Caruana, and A. Niculescu-Mizil, "Model compression," in *Proc. 12th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2006, pp. 535–541.

[5] M. Buckler, S. Jayasuriya, and A. Sampson, "Reconfiguring the imaging pipeline for computer vision," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 975–984.

[6] C. Chen, Q. Chen, J. Xu, and V. Koltun, "Learning to see in the dark," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 3291–3300.

[7] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking Atrous convolution for semantic image segmentation," 2017, *arXiv:1706.05587*.

[8] Y. Chi, A. Gnanasambandam, V. Koltun, and S. H. Chan, "Dynamic low-light imaging with quanta image sensors," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 122–138.

[9] S. Diamond, V. Sitzmann, S. Boyd, G. Wetzstein, and F. Heide, "Dirty pixels: Towards end-to-end image processing and perception," *ACM Transactions on Graphics*, vol. 40, no. 3, 2017, pp. 1–15.
[10] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (VOC) challenge," *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303–338, Jun. 2010.
[11] S. Fukui, J. Yu, and M. Hashimoto, "Distilling knowledge for non-neural networks," in *Proc. Asia-Pacific Signal Inf. Process. Assoc. Annu. Summit Conf.*, 2019, pp. 1411–1416.
[12] A. Gnanasambandam and S. H. Chan, "Image classification in the dark using quanta image sensors," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 484–501.
[13] P. Hansen et al., "ISP4ML: The role of image signal processing in efficient deep learning vision systems," in *Proc. IEEE 25th Int. Conf. Pattern Recognit.*, 2021, pp. 2438–2445.
[14] S. W. Hasinoff et al., "Burst photography for high dynamic range and low-light imaging on mobile cameras," *ACM Trans. Graph.*, vol. 35 no. 6, pp. 1–12, 2016.
[15] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.
[16] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," 2015, *arXiv:1503.02531*.
[17] G. Hong, Z. Mao, X. Lin, and S. H. Chan, "Student-teacher learning from clean inputs to noisy inputs," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 12075–12084.
[18] L. J. Karam, T. Borkar, Y. Cao, and J. Chae, "Generative sensing: Transforming unreliable sensor data for reliable recognition," in *Proc. IEEE Conf. Multimedia Inf. Process. Retrieval*, pp. 100–105, 2018.
[19] Z. Li and D. Hoiem, "Learning without forgetting," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 12, pp. 2935–2947, Dec. 2018.
[20] T.-Y. Lin et al., "Microsoft COCO: Common objects in context," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 740–755.
[21] A. Mosleh, A. Sharma, E. Onzon, F. Mannan, N. Robidoux, and F. Heide, "Hardware-in-the-loop end-to-end optimization of camera image processing pipelines," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 7529–7538.
[22] Z. Peng, Z. Li, J. Zhang, Y. Li, G.-J. Qi, and J. Tang, "Few-shot image recognition with knowledge transfer," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 441–449.
[23] A. Romero, N. Ballas, S. E. Kahou, A. Chassang, C. Gatta, and Y. Bengio, "FitNets: Hints for thin deep nets," 2014, *arXiv:1412.6550*.
[24] O. Russakovsky et al., "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, 2015.
[25] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 4510–4520.
[26] E. Schwartz, R. Giryes, and A. M. Bronstein, "DeepiSP: Toward learning an end-to-end image processing pipeline," *IEEE Trans. Image Process.*, vol. 28, no. 2, pp. 912–923, Feb. 2019.
[27] V. Sharma, A. Diba, D. Neven, M. S. Brown, L. Van Gool, and R. Stiefelhagen, "Classification-driven dynamic image enhancement," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 4033–4041.
[28] E. Tseng et al., "Hyperparameter optimization in black-box image processing using differentiable proxies," *ACM Trans. Graph.*, vol. 38, no. 4, pp. 27–1, 2019.
[29] Y. Wang, Y. Cao, Z.-J. Zha, J. Zhang, and Z. Xiong, "Deep degradation prior for low-quality image classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 11049–11058.
[30] C.-T. Wu et al., "VisionISP: Repurposing the image signal processor for computer vision applications," in *Proc. IEEE Int. Conf. Image Process.*, 2019, pp. 4624–4628.
[31] L. Yahiaoui, J. Horgan, B. Deegan, S. Yogamani, C. Hughes, and P. Denny, "Overview and empirical analysis of ISP parameter tuning for visual perception in autonomous driving," *J. Imag.*, vol. 5, no. 10, 2019, Art. no. 78.
[32] S. Zheng, Y. Song, T. Leung, and I. Goodfellow, "Improving the robustness of deep neural networks via stability training," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 4480–4488.
[33] Y. Zheng, M. Zhang, and F. Lu, "Optical flow in the dark," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 6749–6757.



ELI SCHWARTZ received the B.Sc. degree in electrical engineering from the Technion - Israel Institute of Technology, Haifa, Israel, in 2013, and the M.Sc. degree in electrical engineering in 2018 from Tel Aviv University, Tel Aviv, Israel, where he is currently working toward the Ph.D. degree in electrical engineering. He has authored 17 academic papers with publications in top journals and conferences and was a Program Chair of the Learning with Limitted Labels Workshop with CVPR. He was the recipient of the IBM Fellowship in 2020. He is currently a Research Scientist working for IBM Research. He co-founded and was the CTO of a robotics startup - Inka Robotics that developed the world's first tattooing robot during 2015–2017. Before that, he was with Microsoft, developing algorithms for the HoloLens AR headset during 2013–2016. He was also with the chip design industry for Qualcomm during 2011–2013 and IBM 2008–2011.



ALEX M. BRONSTEIN (Fellow, IEEE) is currently a Professor of computer science with the Technion-Israel Institute of Technology, Haifa, Israel, and a Principal Engineer with Intel Corporation. His research interests include numerical geometry, computer vision, and machine learning. He has authored more than 100 publications in leading journals and conferences, more than 30 patents and patent applications, the research monograph Numerical Geometry of Non-Rigid Shapes, and edited several books. Highlights of his research were featured in CNN, SIAM News, and Wired. In addition to his academic activity, he co-founded and was the Vice President of technology with the Silicon Valley start-up company Novafora, from 2005 to 2009. He was a Co-Founder and one of the main inventors and developers of the 3D sensing technology with the Israeli startup Invision, subsequently acquired by Intel, in 2012. His technology is currently the core of the Intel RealSense 3D camera integrated into a variety of consumer electronic products. He is also a Co-Founder of the Israeli video search startup Videocites and the London-based startup Sibylla, where he is the Chief Scientist. He is a Fellow of the IEEE for his contribution to 3D imaging and geometry processing.



RAJA GIRYES (Senior Member, IEEE) is currently an Associate Professor with the school of electrical engineering with Tel Aviv University, Tel Aviv, Israel. His research interests include intersection between signal and image processing and machine learning, and in particular, in deep learning, inverse problems, sparse representations, computational photography, and signal and image modeling. He was the recipient of the EURASIP Best Ph.D. Award, the ERC-StG grant, Maof prize for excellent Young Faculty during 2016–2019, VATAT scholarship for excellent Postdoctoral Fellows during 2014–2015, Intel Research and Excellence Award in 2005 and 2013, respectively, the Excellence in Signal Processing Award (ESPA) from Texas Instruments in 2008, and was part of the Azrieli Fellows program during 2010–2013. He is an Associate Editor of the IEEE TRANSACTIONS ON IMAGE PROCESSING and *Elsevier Pattern Recognition* and has organized workshops and tutorials on deep learning theory in various conferences including ICML, CVPR, ECCV and ICCV. He is also a Co-Organizer of the Israel computer vision day. He is a Member of the Israeli Young Academy since 2022.