Logarithmic Lenses: Exploring Log RGB Data for Image Classification

Bruce A. Maxwell, Sumegha Singhania, Avnish Patel, Rahul Kumar, Heather Fryling,
Sihan Li, Haonan Sun, Ping He, Zewen Li
Northeastern University, Boston, USA
{b.maxwell, singhania.s, patel.avni, kumar.rahul4, fryling.h, }@northeastern.edu
{li.siha, sun.haon, hi.pin, li.zewen }@northeastern.edu

Abstract

The design of deep network architectures and training methods in computer vision has been well-explored. However, in almost all cases the images have been used as provided, with little exploration of pre-processing steps beyond normalization and data augmentation. Virtually all images posted on the web or captured by devices are processed for viewing by humans. Is the pipeline used for humans also best for use by computers and deep networks?

The human visual system uses logarithmic sensors; differences and sums correspond to ratios and products. Features in log space will be invariant to intensity changes and robust to color balance changes. Log RGB space also reveals structure that is corrupted by typical pre-processing.

We explore using linear and log RGB data for training standard backbone architectures on an image classification task using data derived directly from RAW images to guarantee its integrity. We found that networks trained on log RGB data exhibit improved performance on an unmodified test set and invariance to intensity and color balance modifications without additional training or data augmentation.

Furthermore, we found that the gains from using high quality log data could also be partially or fully realized from data in 8-bit sRGB-JPG format by inverting the sRGB transform and taking the log. These results imply existing databases may benefit from this type of pre-processing.

While working with log data, we found it was critical to retain the integrity of the log relationships and that networks using log data train best with meta-parameters different than those used for sRGB or linear data.

Finally, we introduce a new 10-category 10k RAW image data set (RAW10) for image classification and other purposes to enable further the exploration of log RGB as an input format for deep networks in computer vision.

1. Introduction

Image classification is a fundamental task in computer vision. The ImageNet database enabled the development of large scale deep networks which have been the foundation of the machine learning revolution in computer vision and related areas [6]. The task has been well studied and spawned numerous foundational architectures that have found utility in multiple applications [21][41][14][42][16][38].

While many variations in network design, training methods, meta-parameters, and data augmentation have been studied in-depth, very little effort has been directed towards evaluating different image data spaces. This is likely, in part, because most large scale databases like ImageNet, COCO, Pascal VOC, Faces in the Wild, and Intrinsic Images in the Wild are built from images collected from the web [6][22][8][17][1]. Therefore, the pre-processing steps are unknown, and it is difficult to revert the data to true linear RGB. It may also be due to the fact that people have no problem understanding these images.

However, people do not learn to see based on web images. Our visual systems learn from our own sensors, which means they have access to the physical rules that govern the interactions of light and matter. Furthermore, the human visual system uses a logarithmic sensor, which is part of what gives us the capability of seeing across a wide dynamic range [37]. It also means that basic features–those generated by adding and subtracting signals–are computing products and ratios rather than differences.

In this work we evaluate the hypothesis that using log of linear RGB, can provide improvements in performance for convolutional networks orthogonal to prior work on network design, training methods, or meta-parameter adjustment. Furthermore, we demonstrate that networks trained using log RGB provide natural invariance to intensity and color balance variation.

There is some prior work on using RAW imagery for object detection, and color constancy databases are linear

RGB as white balancing occurs before other pre-processing [44][33][24][13][4][7]. One prior work has shown that log RGB can provide improved performance on a small data set using a simple five layer convolutional neural network [30].

Our work is the first to show improved overall performance and invariance to intensity and color balance on an image classification task using standard backbone architectures: ResNet18, and DenseNet 121 [14][16]. The models were trained from scratch on a 10-category 10k image data base of RAW images [RAW10]. The RAW10 database of RAW images, and processed sRGB-JPG, linear, and log RGB versions of them, will be available after CVPR 2024.

For our primary experiment we trained a ResNet18, DenseNet121, and a custom CNN architecture on three versions of the RAW10 data set: sRGB, linear RGB, and log RGB. In addition to evaluating their performance on an unmodified test set, we also evaluated their performance on the test set with random modifications to intensity and color balance. For all cases, the log RGB data provided gains in performance across all three networks.

We further explored whether it is useful to take 8-bit sRGB JPG images, execute an inverse sRGB transform, and take the log to get pseudo-log RGB data. We evaluated pseudo-log data on a log-trained network and log data on a pseudo-log trained network. In both cases, the pseudo-log data exhibited most or all of the gains achieved by using true log RGB data, suggesting that existing image databases may also benefit from this transformation, and that true log RGB data sets may be expanded with pseudo-log data.

As RAW10 contains some objects where color is a feature and some objects where it is not, we also analyzed the results and found that log-trained networks actually showed more balanced results between the color-relevant and color-irrelevant objects than the other two.

While training and evaluating log data trained networks, we also explored different pre-processing steps and meta-parameters. We found that retaining the integrity of the logarithmic nature of the data was critical, and that log-trained networks work best with somewhat different meta-parameters than linear- or sRGB-trained networks. A simple substitution of log RGB for sRGB in an existing pipeline will likely not work. With attention to detail and some simple guidelines, however, using log RGB is not difficult.

Finally, we discuss some potential explanations for why log RGB provides these improvements, including visualizations of log RGB histograms and basic features in log space.

Our novel contributions include the following.

1. We show that using log of linear RGB data improves performance on a substantial image classification task.
2. We show that using log of linear RGB data provides invariance to intensity and color balance variation.
3. We show that these improvements occur using multiple standard backbone architectures.
4. We further show that these improvements may be realized for data that has undergone an sRGB transformation and reduction to 8-bits per color channel, suggesting that it may be helpful on existing databases.
5. We introduce a new image database, RAW10, suitable for image classification and other computer vision tasks.

2. Related Work

As noted above, image classification is a foundational and well-studied task. There are many standard deep network architectures, or backbones, that have been trained on ImageNet and used for a wide variety of vision tasks. Examples include AlexNet, VGG-16/19, GoogLeNet, ResNet, MobileNet, and DenseNet, and experimentation continues [21][41][14][42][38][16]. For our purposes, we chose ResNet18 and DenseNet121 as high-performance architectures with smaller parameter counts, given that we needed to train them from scratch on our database.

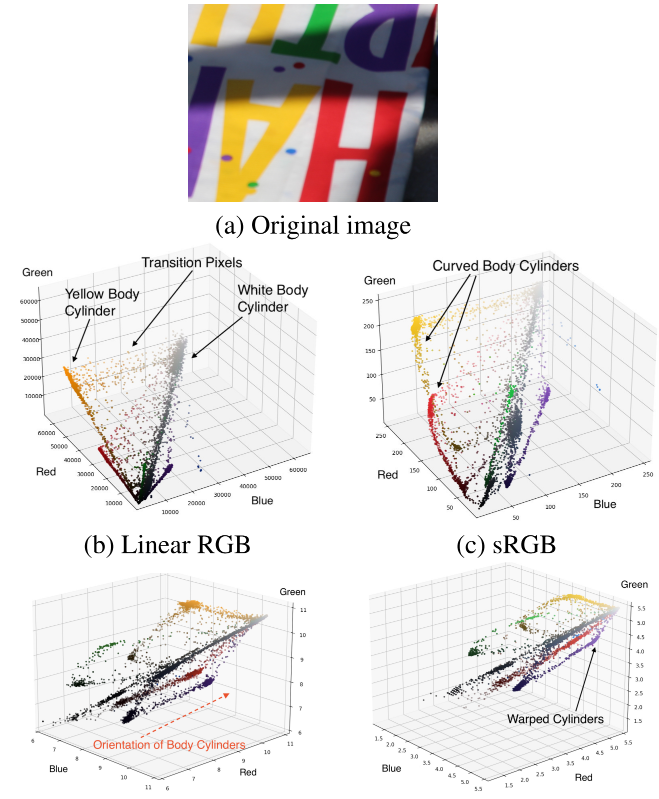
We collected a new database, because there are few image sets available in RAW or linear format outside of color constancy, as noted above. The PascalRAW and LOD data sets support object detection or instance segmentation [33][15][3], however, we wanted to use a larger and more balanced set of categories than PascalRAW, and LOD is intended for special illumination conditions. The ROD database built by [44] provides 25k high dynamic range linear imagery for object detection using six categories in driving scenes, but it was not released until after we had collected the data for RAW10, and the HDR nature of the data would need to be tested for linearity.

Prior work on using RAW data outside of color constancy is minimal, with the work of [44] being the most recent. There is also some work using PascalRAW, but reported results are limited [33][24].

Log RGB has been explored in some applications, such as for computing illumination invariant HOG features in [43] and computing material priors in [35] . Marchant and Onyango [26] and Finlayson *et al*. [9][10] separately derived a 1-D log of chromaticity space, which is invariant to Planckian illuminants. This analysis and methodology has been used for illumination invariant skin-lesion detection [34][25] and for work on color constancy [40].

Maxwell *et al*. (2008) showed there was regular structure in log RGB that could be exploited to create an illumination invariant 2-D log chromaticity space [28]. This has been used for both intrinsic image decomposition [23] and shadow removal on road surfaces [29].

Cotogni and Cusano proposed equivariant networks, which contain a modification to standard convolution enabling them to be invariant to offsets in the input data [5]. They converted both CIFAR [20] and ImageNet into pseudo-log RGB using an inverse sRGB transform and demonstrated that the networks maintained performance de-



(a) Original image

(b) Linear RGB

(c) sRGB

(d) Log of RGB

(e) Log of sRGB

Figure 1. 3-D Histograms showing the structure of body cylinders for linear RGB, sRGB, log RGB, and log of sRGB spaces

spite synthetically varying the color balance of the images. They did the same for color constancy on the NUS data set [4]. However, more recent work suggests that log RGB provides invariance to intensity change and robustness to color variation for convolutional networks without modification.

Funt and Zhu, for example, showed that using Laplacian of log RGB as input to a ResNet50 network also provides invariance to intensity and color variation on standard databases, however, their performance was not better than sRGB on the unmodified data [11]. The method is equivalent to placing a fixed 1-filter layer in front of the network and using log RGB as input. Our approach gives the network the flexibility to learn the first layer filters, and on the RAW10 DB using log RGB provided the best performance.

Maxwell *et al.* (2023) also showed invariance to intensity and color balance for log RGB, and showed log RGB was the best performer for a 2-class image classification task with a small 5-layer network [30]. Herein we use standard backbone deep network architectures and the much larger RAW10 database, replicating the performance gains and robustness to intensity and color balance on a larger scale.

Most other prior work on data quality examines noise, blur, and compression, such as [2] and [12].

3. Background and Methods

Most of the photons measured by a camera sensor come from reflected light, which is the result of physical interactions between incident illumination and the reflecting material. The reflective properties of inhomogeneous dielectric materials–paint, plastic, wood, skin, leaves, cloth, and many others–are well modeled by the dichromatic reflection model, which includes two reflection mechanisms: body reflection and surface reflection [39][19]. Body reflection is what we generally consider to be the color of an object, while surface reflection produces specularities or highlights and reflective or mirror-like effects.

The Lambertian model is most commonly used for body reflection; in linear RGB the reflected light is the product of the material reflectance and the incident illumination. The intensity of the incident illumination can vary due to both geometry and cast shadows.

The Bi-Illuminant Dichromatic Reflection (BIDR) Model separates the incident illumination into two components: the direct illuminant, and the ambient illumination. The direct illuminant is the illuminant that causes most shadows and shading, while the ambient illumination is all the remaining indirect light falling on a surface [28].

The body reflection components of the BIDR model follow a well-defined relationship as the amount of direct illumination changes, given by (1). The image value I is the result of the product of the and the body reflection R_B and the ambient illumination L_A plus the product of the and the body reflection and the direct illumination L_D modified by γ, which represents both geometric shading and cast shadows that modify the intensity of the direct illuminant.

$$I = L_A R_B + \gamma L_D R_B = R_B(L_A + \gamma L_D) \qquad (1)$$

In linear RGB, the body reflection for each material falls along the line, or cylinder, defined by (1) that starts at the color of a fully shadowed pixel ($\gamma = 0$) and ends at the color of a fully lit pixel ($\gamma = 1$). If the ambient and direct illumination are different colors–almost all scenes fall in this category–then the cylinders do not intersect at a single point and do not go through the origin $(R, G, B) = (0, 0, 0)$. In linear RGB space both the length and orientation of the body cylinders are dependent on both the direct illumination color and the material body color. This can be seen in Figure 1, which shows the 3-D histograms of an image of a shadow crossing a multi-colored surface. Each body color on the surface appears in Figure 1(b) as a line in linear RGB space. Brighter materials have longer lines because R_B has a larger magnitude. Note how the non-linear sRGB transform stretches and bends the body cylinders in Figure 1(c).

A log transform of the body reflection model separates the confounded body reflection and illumination terms (2).

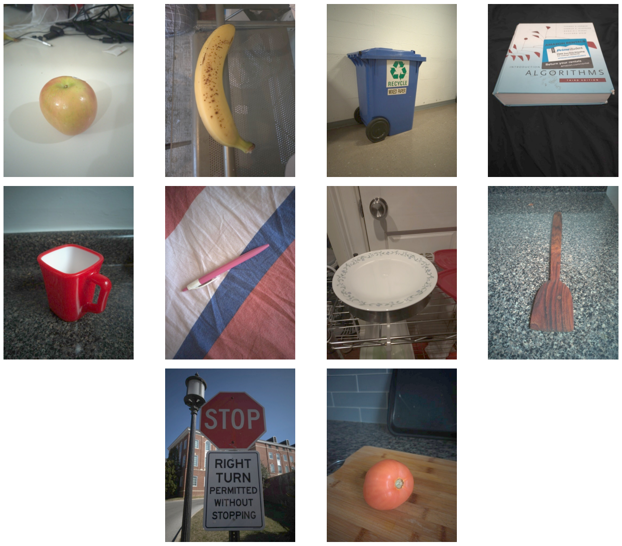


Figure 2. Example images of each class in the RAW10 database

$$\begin{aligned}\log I &= \log R_B(L_A + \gamma L_D)\\ &= \log R_B + \log(L_A + \gamma L_D)\end{aligned} \tag{2}$$

The body reflection cylinders (approximate) for each material are now defined solely by the illumination colors and intensities, with their location defined by the material's body color. As illumination is a slow-changing signal, this means that the body reflection cylinders for different materials are all oriented in roughly the same direction and are all roughly the same length. Figure 1(d) shows the log RGB histogram of the multi-colored surface. In the log RGB histogram, the body cylinders in log space are all pointing up and right. The more horizontal strip of yellow pixels are actually transition pixels from the yellow to the white material. Note how in log space the sRGB transformation warps the structure of the body cylinders by bending them at the bright end as in Figure 1(e).

The hypothesis we are testing in this work is whether using linear RGB or log RGB as the input data for deep networks produces better results–broadly defined–for computer vision tasks. If we give deep networks access to data that follows physical rules, does it improve their ability to learn the task?

4. Experiments and Results

4.1. RAW10 Database

Given the lack of publicly available databases of high quality RAW imagery, we collected a new data set of roughly 10,000 images, 1000 each of ten different categories: apples, tomatoes, bananas, recycle bins, stop-signs, plates, books, spatulas, mugs, and pens. We intentionally picked five objects where color is an informative feature and five where color is not inherent to the object.

The images were collected by roughly one hundred different individuals with different cameras and objects. All images were collected in a RAW format (e.g. CR2, DNG, NEF). The total data set contains over 10,000 images. We used 1000 images–100 per category–for the validation set, and 2000 images–200 per category–for the test set, leaving over 7,000 for the training set. Figure 2 shows a sample image from each category. For a subset of the images, we also have camera JPEG versions provided by the capture device.

From the RAW images we built three databases to use for training: JPG-sRGB, TIFF-Linear, and EXR-Log (referred to hereafter as sRGB, linear, and log). To build the databases we first processed the RAW images using the rawPy package to de-Bayer, white balance, and scale the exposure levels [36]. We used the camera white balance and specified that fewer than 0.001% of the pixels in the image should be saturated. We then used the OpenCV resize function with the INTER_AREA method to scale the images to make the short edge 480 pixels. At this point in the process the data was represented using 16-bits per color channel with a range of [0, 65535].

To create the sRGB data set, we converted the data to float32, scaled the data to [0, 1], applied the sRGB transformation [18], scaled the values to [0, 255], and converted the image to 8-bit uint type before saving it as a JPG using the OpenCV imwrite method with default settings. To create the linear data set, we stored the 16-bit data in the TIFF format using the imageio imwrite method. To create the log data set, we converted the data to float32 and took the natural log of the non-zero pixel values to get log RGB with a range of [0, 11.1]. We then saved the data in a float32 EXR file using the imageio imwrite method with default settings. Note the EXR package is not enabled by default in OpenCV. We used the rescaled images in the sRGB, linear, and log data sets for all subsequent experiments.

We also created three alternative versions of the test set with different transformations applied: random color balance, random intensity variation, and both random intensity and color balance variation using a protocol identical to [30] and based on the same model as [11]. We applied the transformations to the linear data before converting it to JPG-sRGB, linear, and log representations. We evaluated the different networks on identically modified test sets.

4.2. Core Experiments

For the core experiments we trained three versions of three different network architectures and evaluated each one using four different test sets. The three architectures we evaluated were ResNet18 (11.7M), DenseNet121 (8M), and an 11-layer custom CNN (3M). Given the size of the database, we chose lower parameter models to minimize overfitting.

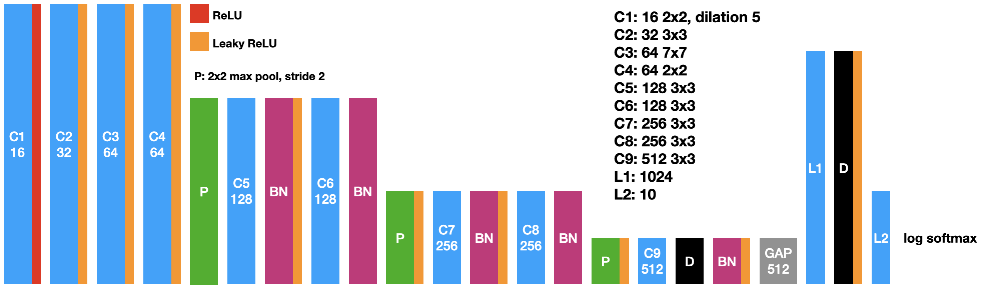


Figure 3. Custom 11-layer CNN with ∼3M parameters.

DenseNet121	sRGB	Lin RGB	Log RGB
Original Test Set (2k)	91.4	91.6	**91.9**
Random CB	84.7	84.7	**88.0**
Random Intensity	91.0	91.4	**91.8**
Both	81.0	82.0	**86.9**
Validation set (1k)	88.30	89.50	**90.17**
Training set (7k)	**99.79**	99.54	98.92
ResNet18	sRGB	Lin RGB	Log RGB
Original Test Set (2k)	86.9	88.9	**90.6**
Random CB	72.6	75.7	**86.9**
Random Intensity	72.8	76.3	**87.2**
Both	67.3	71.9	**83.9**
Validation set (1k)	87.60	87.40	**89.08**
Training set (7k)	99.22	96.49	**99.61**

Table 1. Results on the RAW10 data set for ResNet18 and DenseNet121 network architectures, three data types, and four variations of the training set.

We used the ResNet18 model from torchvision, but added a 0.5 dropout layer and a Batch Normalization layer after each ResNet block. We added one additional Batch Normalization layer after the first convolution layer. These updates enabled the network to train with less overfitting.

For the DenseNet121 architecture, we added a single 0.5 dropout layer right before the final fully connected layer to avoid overfitting. Figure 3 shows the custom network design. While the custom design did not perform as well as the two standard backbones, it is a third example of log data performing better than sRGB or linear using the same network architecture. Our goal was to find a smaller architecture that could still learn the problem.

We trained three versions of each network architecture, randomly initialized, on the three different input types: sRGB, linear, and log. For each data type we experimented with meta-parameters to obtain the best-performing model. The sRGB and linear data produced their best results using the same meta-parameters. However, the log network required different meta-parameters in order to train well. Section 4.4 discusses the experiments in more detail.

When presenting the linear and log data to the network, we did not follow the usual ImageNet type normalization for linear and log inputs. Instead, the linear data was rescaled to [0, 1] and the log data was left unchanged with the range [0, 11.1]. We experimented with scaling the log data to [0, 1] and scaling to [0, 256] before taking the log, but neither method trained as well. Likewise, normalizing the log RGB data to [0, 1]–which is roughly the same as taking the 11th root of the data–did not work. Subtracting 5.05 from the log data to center the data on zero also did not work. Note that it is also important not to resize the log RGB data, as interpolation in log space does not produce the same image as interpolation in linear space. All resizing, intensity, and color balance operations should be executed in linear RGB to retain the integrity of the data.

Table 1 shows the accuracy results for ResNet18 and DenseNet121. The networks trained on the log RGB data performed the best on all versions of the test set: original, random color balance, random intensity, and both augmentations. The log trained network exhibited minimal impact of intensity variation across the three input types. The sRGB-trained and linear-trained networks performed similarly, and neither generalized as well to the modified test sets as the log trained network, although the DenseNet architecture exhibited better generalization than ResNet18.

Table 2 shows the results on the custom network. Somewhat surprisingly, the sRGB data did better in some cases when trained on the original data set. We also trained the network on an augmented, but fixed training set as well as a dynamically augmented training set, in both cases including images with no augmentation, random color balance, random intensity variation, and both types of variation with equal probability. In the latter two cases, the custom network trained on log data did better across the board and learned the task better than on the original data set. While the sRGB and linear networks also improved their performance, they did not improve as much as the log network. Note also that the losses on the test set indicate the log network was more confident in its answers than the sRGB or Linear networks, even when its accuracy was slightly lower.

Original	sRGB	Lin RGB	Log RGB
Original Test Set	**72.05**/1.7	64.32/3.1	70.42/1.4
Random CB	**59.75**/2.7	52.89/3.7	58.93/2.2
Random Intensity	**72.05**/1.7	62.50/3.0	71.88/1.4
Both	54.71/3.2	48.27/3.7	**56.18**/2.5
Validation set (1k)	71.50/2.0	73.10/1.7	**72.90**/1.4
Fixed	sRGB	Lin RGB	Log RGB
Original Test Set	72.05/1.8	73.52/1.9	**76.92**/1.4
Random CB	74.04/1.7	72.58/1.8	**76.45**/1.4
Random Intensity	74.28/1.7	73.63/1.8	**76.39**/1.4
Both	73.52/1.7	72.23/1.9	**76.39**/1.4
Validation set (1k)	71.10/2.0	71.30/2.0	**73.20**/1.7
Dynamic	sRGB	Lin RGB	Log RGB
Original Test Set	72.17/1.6	73.87/1.4	**75.40**/1.4
Random CB	73.11/1.5	72.29/1.5	**75.92**/1.4
Random Intensity	73.29/1.5	73.05/1.4	**75.63**/1.4
Both	73.93/1.5	72.29/1.5	**75.04**/1.4
Validation set (1k)	72.60/1.8	71.30/1.7	**73.10**/1.3

Table 2. Accuracy/Loss on the custom 11-layer CNN for the original training set, a fixed augmented training set, and a dynamically augmented training set.

True log train Test Data (log)	ResNet True	ResNet Pseudo	DenseNet True	DenseNet Pseudo
Original	90.64	89.90	91.94	91.70
CB	86.90	86.40	87.95	87.85
Intensity	87.22	89.90	91.75	91.90
Both	83.87	83.95	86.85	86.65
Pseudo-log train Test Data (log)	ResNet True	ResNet Pseudo	DenseNet True	DenseNet Pseudo
Original	90.29	89.85	91.74	91.85
CB	86.60	86.95	87.70	87.60
Intensity	90.20	90.25	91.70	91.60
Both	83.40	83.35	84.05	83.80

Table 3. Results of training ResNet18 and DenseNet121 on true-log data or pseudo-log data and testing on both true-log and pseudo-log data. All combinations are better than comparable results training and testing on sRGB (column 1 of figure 1).

4.3. Pseudo-log Experiments and Results

We captured RAW images to guarantee linear data that preserves the physics of reflection and illumination. However, most available data sets exist as 8-bit per color channel sRGB data, with other potential modifications such as color saturation, sharpening, or contrast enhancement. Images from the web may also have been resized or edited and resaved in JPEG format, further degrading the integrity of the data. Nevertheless, we explored whether it would be worthwhile to invert the sRGB transformation and then take the log to get data that approximates log of linear RGB.

We introduce the terms *pseudo-linear* and *pseudo-log* to designate data that was originally 8-bit sRGB and has had an inverse sRGB transformation applied (pseudo-linear) followed by a log transform (pseudo-log). True linear and true log data are derived from images in RAW format using a process, such as in section 4.1, that retains the integrity and numerical precision of the original sensor data.

	ResNet			DenseNet		
Par	sRGB	Lin	Log	sRGB	Lin	Log
Optim.	SGD	SGD	Adam	SGD	SGD	Adam
LR	.0005	.0045	.00736	.005	.01	.005
mom	0.9	0.9	n/a	0	0	n/a
eps	n/a	n/a	0.1	n/a	n/a	0.1
Sched.	Step	Step	Poly	Poly	Poly	Poly
gamma	0.8	0.8	n/a	n/a	n/a	n/a
step	5	5	n/a	n/a	n/a	n/a
decay	0.01	0.01	0.01	0.01	0.01	0.03
train	99.22	96.49	**99.61**	**99.79**	99.54	98.92
val	87.60	87.40	**89.08**	88.30	89.50	**90.17**
test	86.92	88.91	**90.64**	91.35	91.60	**91.94**

Table 4. Optimizer (Adam or SGD), scheduler (Poly = PolynomialLR or Step = StepLR), meta-parameters, train set, validation set, and unmodified test set accuracy for ResNet18 and DenseNet121. The Polynomial scheduler order was one (linear), with the number of iterations set to 100.

To create the pseudo-log version of the RAW10 data set, we used the sRGB JPG files as the source data, executed an inverse sRGB transform, then a log transform. Numerical issues with this process include quantization to 8-bits, the inverse sRGB transform mapping multiple values to a single value for darker pixels while creating skips between values for brighter pixels, and JPG compression artifacts.

We executed four experiments using true-log and pseudo-log data. First, we evaluated the performance of the true-log and pseudo-log test sets on true-log trained ResNet18 and DenseNet121 networks. Second, we trained both networks using the pseudo-log data and evaluated their performance on both the true-log and pseudo-log test sets. The results are in table 3. In both cases, using the pseudo-log data provided a benefit compared to a network trained on the original sRGB data, particularly for the modified test sets. Conversely, the differences between training on pseudo-log data or true-log data were small, though training on true-log data showed a slight benefit in most cases.

This finding is significant because it means that the millions of images in existing annotated data sets may be useful for training log of RGB networks, even if those networks are then used with true log data.

4.4. Meta-parameter Exploration

In order to test our hypothesis that log images are better inputs for this task, we needed to identify the best performing networks for each input data type. Given the differences in ranges and data meanings, we evaluated two optimizers–AdamW and SGD–and two schedulers–a Polynomial scheduler and a Step Scheduler. We trained all three networks using all four combinations as well as exploring different learning rates, momentum coefficients, and weight decays. All network combinations were trained for 100 epochs. All of the networks attained over 99% performance on the training set regardless of the optimizer and scheduler except the DenseNet log network and ResNet18 linear network, which attained 98.92% and 96.49%, respectively, with their optimal validation set parameters. Table 4 shows the optimal parameters and accuracies for the ResNet and DenseNet networks and data types.

The custom network showed less sensitivity to meta-parameters. All three versions of the custom network used a learning rate of 0.003, and the Adam optimizer with an eps of 0.1. The sRGB and Linear networks used a dropout percentage of 0 (no dropout), while the log network used a dropout percentage of 0.5.

4.5. A Bias Towards Color?

We selected five categories of the RAW10 data base to be objects where color may be a helpful feature: apples, bananas, tomatoes, recycle bins, and stop-signs. The other five categories we selected to be color agnostic: plates, mugs, pens, books, and spatulas (broadly defined). We wanted to test the hypothesis that networks trained on log RGB might perform better on objects where color was a defining feature. Table 5 shows the accuracies per class for the ResNet18 network trained on sRGB, linear, and log data.

Some objects stand out in their performance. Of note, the stop-sign proved to be an easy category regardless of input space, likely because it was the only object captured solely outdoors, meaning that context provided useful information. The sRGB network had the most difficulty with the banana, pen, and spatula categories in the modified test sets.

The average accuracies for color-relevant versus color-irrelevant categories exhibit much less difference for the log network, both in absolute terms and as a percentage, than the linear and sRGB networks. The gains in performance over the linear and sRGB networks were larger in the color-irrelevant categories. This suggests that ratios may be useful features not only within an object, but also between objects. For two pixels within an object, their ratios should stay consistent across illumination changes. Ratios between objects, however, are less likely to exhibit a consistent pattern, especially if the objects are not at the same depth.

The results of this analysis do not support the hypothesis that a log-trained network will perform better only on objects where color is relevant, but instead show improved performance for both color-relevant and color-irrelevant objects. In fact, the relative gain in performance was greater for color-irrelevant objects.

5. Discussion

The experiments support the hypothesis that using log of linear RGB is a more effective pre-processing pipeline than the pipeline typically used for images to be viewed by humans. In addition, the results support the hypothesis that even images stored in an sRGB 8-bit jpg-compressed can provide improved performance by undoing the sRGB transformation and converting them to pseudo-log images.

An important question is to ask why log images are better inputs. The mathematical derivations and visualizations in section 3 provide a basis for building hypotheses. In particular, computing differences in log space means calculating ratios in linear space, and ratios between nearby pixels under similar illumination are invariant to illumination intensity and color balance. Ratios have also been found to be useful for object recognition and differentiating surfaces that are continuous from ones that are not [31], [32], [27].

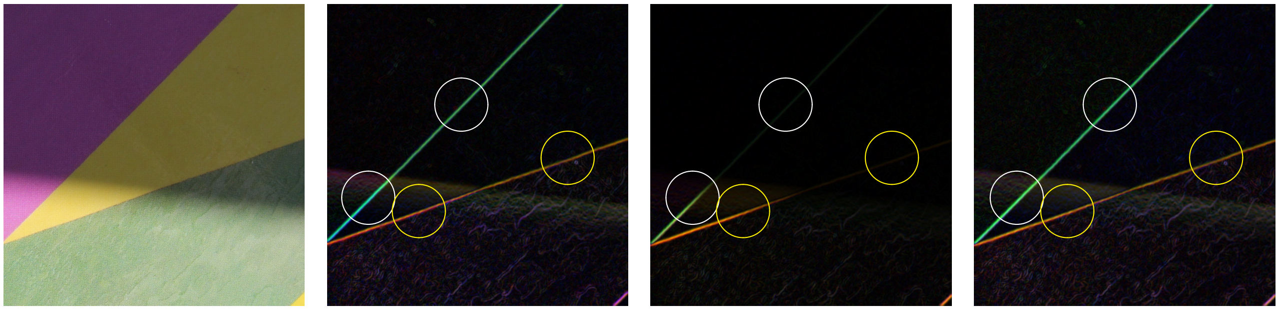
For example, figure 4 shows the gradient magnitude of an image with edges between differently colored regions that cross shadow boundaries. These are the type of features that would be computed by a CNN filter. In sRGB the gradient features change color and magnitude as the illumination changes. In linear RGB the color of the gradient features is consistent, but their magnitude still changes with the illumination. In log of linear RGB, the gradient features not only retain their color, but also their magnitude. The features of a multi-colored object become stable across illumination change in log RGB, whereas in sRGB the features change due to both material color and illumination conditions. A network trained on sRGB data has to either be sloppy with respect to color or learn different rules for different colors and different illumination conditions.

These visualizations provide a potential explanation for the improvements in performance and invariance to intensity and color balance provided by log-trained networks. The performance results by themselves challenge the idea of using images processed for human consumption. The improvement gains of using log of linear RGB inputs are orthogonal to network design, meaning that most network-based CV solutions could improve their performance simply by changing the preprocessing of the inputs. The gains also do not appear to be solely for categories where color is an identifying property of the object.

The results of the experiments with pseudo-log data, also suggest that there is benefit in re-processing existing images and retraining networks with pseudo-log data. Such a finding means that, while it is beneficial to train on high quality log data, networks can benefit from training on re-processed existing data. This has broad implications across many vision applications. Every computer vision researcher and developer should be asking themselves why they are training on sRGB images, especially if they are working on edge devices or have access to RAW data in their vision pipeline.

sRGB	apple	banana	bin	stop-sign	tomato	book	mug	pen	plate	spatula	μ_c	μ_m	$\%\Delta$
original	82.5	85.5	89.0	99.5	95.0	92.0	90.0	73.0	87.5	75.0	90.3	83.5	8.1
color balance	66.5	46.0	82.0	96.0	89.0	90.5	84.5	47.0	81.0	43.0	75.9	69.2	9.7
intensity	67.0	48.5	80.5	96.5	88.0	92.0	84.0	46.5	81.5	43.0	76.1	69.4	9.7
both	59.0	43.5	78.0	87.5	82.0	86.0	82.5	41.5	75.5	37.0	70.0	64.5	8.5
Linear	apple	banana	bin	stop-sign	tomato	book	mug	pen	plate	spatula	μ_c	μ_m	$\%\Delta$
original	90.0	92.0	86.5	99.0	89.0	83.0	90.0	85.5	89.5	84.5	91.3	86.5	5.5
color balance	88.0	64.0	79.0	93.0	72.5	72.5	75.5	71.0	75.5	65.5	79.3	72.0	10.1
intensity	88.5	64.0	80.0	94.0	73.0	74.0	76.0	71.0	76.5	65.5	79.9	72.6	10.1
both	88.5	56.5	77.5	87.5	62.0	69.5	73.5	74.5	70.6	59.0	74.4	69.4	7.2
Log	apple	banana	bin	stop-sign	tomato	book	mug	pen	plate	spatula	μ_c	μ_m	$\%\Delta$
original	90.0	91.5	87.5	98.0	94.0	90.0	89.0	85.5	91.0	89.0	**92.2**	**88.9**	**3.7**
color balance	87.0	81.0	88.0	98.5	87.5	87.0	89.0	81.5	87.5	82.0	**88.4**	**85.4**	**3.5**
intensity	87.5	84.0	88.5	97.5	87.5	85.6	89.5	82.0	88.5	81.0	**89.0**	**85.3**	**4.3**
both	85.0	72.5	86.5	97.0	80.5	85.5	85.5	80.5	89.0	76.9	**84.3**	**83.5**	**1.0**

Table 5. Accuracies per class on RAW10 for ResNet18, average accuracy of color-relevant objects μ_c and color-insensitive objects μ_m, and % difference between the two $\%\Delta$, given for the original test set, color balance variation, intensity variation, and both.



(a) Original image in sRGB

(b) Gradients of sRGB.

(c) Gradients of linear RGB.

(d) Gradients of log of linear RGB

Figure 4. Gradient magnitudes of the three data types. The circles–white pair and yellow pair–show shadow and lit areas of the same edge. Gradients in sRGB change color in the shadow, in linear RGB change magnitude, and in log RGB are consistent in both intensity and color.

The finding that using log data improves a network's performance does not, however, mean that performance cannot be improved by network structures designed specifically for log data. For example, there may be benefits to modifying the designs of the first few network layers. Other more radical network modifications may be worth exploring. Log space is not as intuitive as linear space, and there is lots of room for more exploration of both architectures and the training process to identify stable and effective meta-parameters and training methods.

6. Summary

Using log RGB data improves the performance of two standard deep network architectures compared to training on sRGB or linear data for a substantial image classification task. Furthermore, the benefit appears to hold for these networks even when converting from 8-bit sRGB JPG files into a pseudo-log RGB space using an inverse sRGB transformation. This implies that existing data sets may be useful in training networks to use log RGB images as inputs. These gains appear to be orthogonal to network architecture and training set augmentation.

The interaction of light and matter produces structure in the appearance of objects, and that structure may be made easier to learn by a log transformation. Furthermore, basic features such as edges are more consistent in log RGB than in sRGB or linear RGB space, giving log-trained networks a natural robustness to intensity and color balance variation.

There is still much to explore. The natural separation of illumination and reflectance that occurs in log space suggests it may be more suitable for tasks such as color constancy, shadow removal, and intrinsic imaging. Future exploration should also include building custom architectures that take advantage of the structure found in log RGB data.

Acknowledgements: The authors would like to recognize the contributions of the students in the Spring 2023 MS CS Computer Vision class at Northeastern University, who took almost all of the images in the RAW10 database.

References

[1] Sean Bell, Kavita Bala, and Noah Snavely. Intrinsic images in the wild. *ACM Trans. Graphics (SIGGRAPH)*, 33 (4), 2014. 1

[2] Christoph Borel-Donohue and S. Susan Young. Image quality and super resolution effects on object recognition using deep neural networks. In *Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications*, page 110061M. International Society for Optics and Photonics, SPIE, 2019. 3

[3] Linwei Chen, Ying Fu, Kaixuan Wei, Dezhi Zheng, and Felix Heide. Instance segmentation in the dark. *International Journal of Computer Vision*, 131, 2023. 2

[4] D. Cheng, D.K. Prasad, and M.S. Brown. Illuminant estimation for color constancy: why spatial-domain methods work and the role of the color distribution. *Journal of the Optical Society of America A*, 31(5):1049–1058, 2014. 2, 3

[5] Marco Cotogni and Claudio Cusano. Offset equivariant networks and their applications. *Neurocomputing*, 502:110–119, 2022. 2

[6] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *IEEE conference on computer vision and pattern recognition*. IEEE, 2009. 1

[7] Egor Ershov, Alexey Savchik, Illya Semenkov, Nikola Banić, Alexander Belokopytov, Daria Senshina, Karlo Koščević, Marko Subašić, and Sven Lončarić. The cube++ illumination estimation dataset. *IEEE Access*, 8, 2020. 2

[8] Mark Everingham, Luc Van Gool, Christopher K. I. Williams, John Winn, and Andrew Zisserman. The Pascal Visual Object Classes (VOC) Challenge. *International Journal of Computer Vision*, 88(2):303–338, 2010. 1

[9] G. D. Finlayson, S. D. Hordley, and M. S. Drew. Removing shadows from images. In *Proc. of European Conf. on Computer Vision*, pages 823–836, London, UK, 2002. Springer-Verlag. 2

[10] G. D. Finlayson, M. S. Drew, and L. Cheng. Intrinsic images by entropy minimization. In *Proc. of European Conf. on Computer Vision*, pages 582–595, 2004. 2

[11] Brian Funt and Ligeng Zhu. Laplacian of logarithm as illumination-invariant input space. In *Proc. of 30th Color and Imaging Conference*, 2022. 3, 4

[12] Tomasz Gandor and Jakub Nalepa. First gradually, then suddenly: Understanding the impact of image compression on object detection using deep learning. *Sensors*, 22, 2022. 3

[13] P.V. Gehler, C. Rother, A. Blake, T. Minka, and T. Sharp. Bayesian color constancy revisited. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2008. 2

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2015. 1, 2

[15] Yang Hong, Kaixuan Wei, Linwei Chen, and Ying Fu. Crafting object detection in very low light. In *BMVC*, 2021. 2

[16] Gao Huang, Zhuang Liu, and Laurens van der Maaten. Densely connected convolutional networks. In *CVPR*, 2017. 1, 2

[17] Gary B. Huang, Manu Ramesh, Tamara Berg, and Erik Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical Report 07-49, University of Massachusetts, Amherst, 2007. 1

[18] IEC. 61966-2-1:1999 multimedia systems and equipment - colour measurement and management - part 2-1: Colour management - default rgb colour space - srgb: Amendment 1. Technical report, International Electrotechnical Commission, 2003. 4

[19] Gudrun J. Klinker, Steven A. Shafer, and Takeo Kanade. A physical approach to image understanding. *Int'l J. of Computer Vision*, 4(1):7–38, 1990. 3

[20] A. Krizhevsky and G. Hinton. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009. 2

[21] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *NeuroIPS*, 2012. 1, 2

[22] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft COCO: Common Objects in Context, 2015. arXiv:1405.0312 [cs]. 1

[23] Yuanliu Liu, Ang Li, Zejian Yuan, Badong Chen, and Nanning Zheng. Consistency-aware shading orders selective fusion for intrinsic image decomposition. *ArXiv*, abs/1810.09706, 2018. 2

[24] William Ljungbergh, Joakim Johnander, Christoffer Petersson, and Michael Felsberg. Raw or cooked? object detection on raw images. In *Image Analysis*, pages 374–385, Cham, 2023. Springer Nature Switzerland. 2

[25] Ali Madooei, Mark S. Drew, Maryam Sadeghi, and M. Stella Atkins. Intrinsic melanin and hemoglobin colour components for skin lesion malignancy detection. *Med Image Comput Comput Assist Interv*, 15, 2012. 2

[26] John A. Marchant and Christine M. Onyango. Shadow-invariant classification for scenes illuminated by daylight. *J. of the Optical Society of America A*, 17(11), 2000. 2

[27] B. A. Maxwell and S. A. Shafer. Physics-based segmentation of complex objects using multiple hypotheses of image formation. *Computer Vision and Image Understanding*, 65 (2):269–295, 1997. 7

[28] Bruce A Maxwell, Richard M Friedhoff, and Casey A Smith. A bi-illuminant dichromatic reflection model for understanding images. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008. 2, 3

[29] Bruce A. Maxwell, Casey A. Smith, Maan Qraitem, Ross Messing, Spencer Whitt, Nicolas Thien, and Richard M. Friedhoff. Real-time physics-based removal of shadows and shading from road surfaces. In *CVPR Workshop on Autonomous Driving*. IEEE / CVF, 2019. 2

[30] Bruce A. Maxwell, Sumegha Singhania, Heather Fryling, and Haonan Sun. Log rgb images provide invariance to intensity and color balance variation for convolutional networks. In *British Machine Vision Conference*, 2023. 2, 3, 4

[31] S. K. Nayar and R. M. Bolle. Computing reflectance ratios from an image. *Pattern Recognition*, 26(10):1529–1542, 1993. 7

[32] S. K. Nayar and R. M. Bolle. Reflectance based object recognition. *Int'l J. of Computer Vision*, 17(3):219–240, 1996. 7

[33] Alex Omid-Zohoor, David Ta, and Boris Murmann. Pascalraw: Raw image database for object detection, 2015. 2

[34] Luisa F. Polanía, Raja Bala, Ankur Purwar, Paul Matts, and Martin Maltz. Skin chromophore estimation from mobile selfie images using constrained independent component analysis. *Electronic Imaging*, 14, 2020. 2

[35] Jerome Put, Nick Michiels, and Philippe Bekaert. Material-specific chromaticity priors. In *British Machine Vision Conference*, 2016. 2

[36] Mike Reichert. rawpy, 2023. 4

[37] Albert Rose. The sensitivity performance of the human eye on an absolute scale. *Journal of the Optical Society of America*, 28(2):196–208, 1948. 1

[38] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *CVPR*, 2018. 1, 2

[39] Steven A. Shafer. Using color to separate reflection components. *Color Research Applications*, 10:210–218, 1985. 3

[40] Wu Shi, Chen Change Loy, and Xiaoou Tang. Deep specialized network for illuminant estimation. In *European Conf. on Computer Vision*, 2016. 2

[41] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015. 1, 2

[42] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2015. 1, 2

[43] X. Wang, G. Doretto, T. Sebastian, J. Rittscher, and P. Tu. Shape and appearance context modeling. In *International Conference on Computer Vision*, 2007. 2

[44] Ruikang Xu, Chang Chen, Jingyang Peng, Cheng Li, Yibin Huang, Fenglong Song, Youliang Yan, and Zhiwei Xiong. Toward raw object detection: A new benchmark and a new model. In *CVPR*, 2023. 2