

Exploring Physical Parameters for Incoherent Imaging

Testing Blur Kernels, Illumination Matrices, and RGB Channel Weights for Classifying Datasets of Varying Complexity

Connor Davis
Biomedical Engineering
Duke University
Durham, NC USA
connor.davis@duke.edu

ABSTRACT

As some of the focus in machine learning shifts to designing optimal physical parameters for imaging systems, we must examine how some of the most basic of these parameters behave for datasets of varying complexity. To that end, in this paper, we look at three relatively simple physical parameters, a blur kernel (blurring filter), illumination matrix (image lighting), and RGB channel weights (another form of image lighting), and how their incorporation into a machine learning framework affects classification performance of a simple CNN on four different datasets. The datasets used for this analysis are MNIST, Fashion MNIST, CIFAR10, and HAM10000 (“Skin Cancer MNIST”). The validation accuracy results show that for simple sets of images, such as the MNIST and Fashion MNIST, incorporating the physical parameters does not much affect classification performance. This result is unsurprising given that even simple CNNs will achieve quite high validation accuracies for these data, so incorporating pre-processing layers is unlikely to have much of an effect. Additionally, these physical layers appear to have only a slight affect on classification of the CIFAR 10 data, if there is any real affect at all. Most interestingly, however, different physical parameters have varying and significant effects on classification of the HAM10000 data. This result may stem from certain aspects of these images, and, given more tests to prove the validity of these outcomes, could reveal aspects of skin lesion images that are critical to their being properly labeled.

INTRODUCTION

Historically, machine learning has been used to analyze pre-existing sets of images for tasks such as segmentation, object identification, and classification; however, recently some of the focus in machine learning has shifted toward using these networks to develop optimal parameters for the physical imaging systems that produce those datasets [1]. In this way, images may be created such that analyzing them through neural networks is easier, faster, and/or more accurate.

The goal of this paper is to explore some of these physical layers for incoherent imaging systems with data of varying complexity. In this case, an incoherent imaging system is one in which we treat light only as an intensity, not as a wave [2], and the

datasets used for analysis were MNIST (handwritten digits 0 – 9), Fashion MNIST (images of 10 different clothing articles), CIFAR10 (color images of 10 different objects), and HAM10000. This last dataset, also known as the Skin Cancer MNIST, is a set of 10,015 color images showing skin lesions of seven distinct types [3]. In order to compare the results of learning various physical parameters for each of these datasets, the color images in CIFAR10 and HAM10000 were converted to grayscale, so as to be of the same type as MNIST and Fashion MNIST. An example image from each dataset is shown in the figure below.



Figure 1. Sample of MNIST, Fashion MNIST, CIFAR10, and HAM10000 Data, respectively

Three to four different physical layers were examined in conjunction with simple CNNs for each set of images. First, an illumination matrix was imposed and learned with the CNNs. The idea of such a matrix is that by multiplying it element-wise with each image to be examined, certain important features could be illuminated (or highlighted) more than others. Hopefully, such a physical feature as applying different levels of illumination to each pixel could produce images that would give more accurate results in a CNN. An example of a learned illumination matrix is shown below – note that the illumination values differ most in the pixels that are likely to contain information on the MNIST digit.

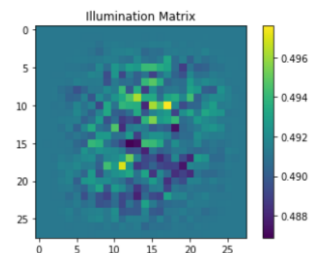


Figure 2. Sample of Learned Illumination Matrix for an MNIST Classifier

As an extension of this physical layer, different matrices of weights were learned for each of the red, green, and blue channel data of the CIFAR10 and HAM10000 color images. Physically speaking, this application is similar to applying variable RGB filters over a camera lens or illuminating a sample with varying intensities of RGB light. (These weights are considered the fourth physical parameter in the paper and are not applicable for MNIST or Fashion MNIST.)

The second and third physical parameters considered here are both types of blur kernels, one of which we refer to as a “static” kernel and the other as a “dynamic” kernel. In a physical sense, these kernels act as filters at the aperture level. The mathematical idea for them is that as they are convolved with an image, they will blur certain parts of it. Because the static kernel does not change as it slides over an image, it should blur everything in the same manner, as any averaging, low-pass filter would. The images in the figure below show how a static blur kernel would affect an image.

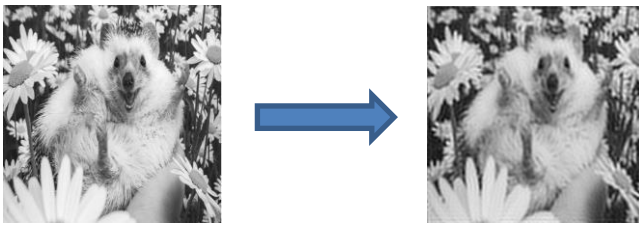


Figure 3. Convoluting an Image with a Static Kernels Blurs It Uniformly

The dynamic kernel, however, is allowed to change as it shifts over the image; this property allows it to blur certain parts of an image more than others. The hope with implementing such a blur kernel is that it could obscure parts of an image that contain little information, thereby bringing important parts to the foreground.

In order to create this dynamic kernel in Python, some mathematical manipulations had to be performed, and these manipulations depend on the fact that a convolution may be written as the multiplication between a matrix and the flattened image input [4]. For images of size 28x28 pixels, such as for MNIST and Fashion MNIST, the inputs were padded with zeros of depth 2 and then flattened. Two matrices of size (32*30)-by-(32²) were then initialized with zeros. The first of these matrices was input as a TensorFlow variable and contained the weights to be learned; these weights, all of the same random value, were initialized along the appropriate diagonals of the matrix such that it imitated a convolution. (These diagonals are more easily seen than described – refer to the image at the end of this paragraph.) The second matrix was populated by the same method as the first, but it was treated as a constant and only the value of 1.0 was input along the diagonals. Before every step of gradient descent during training, the Hadamard product of these matrices was computed before matrix multiplying with the flattened image input. In this way, only those convolutional kernel values along the diagonal would be learned in TensorFlow and not any of the initialized zeros. For the 32x32 images (CIFAR10 and HAM10000), the Toeplitz matrices were of

size (36*34)-by-(36²). Refer to the commented code for a more complete description of how these computations were performed.

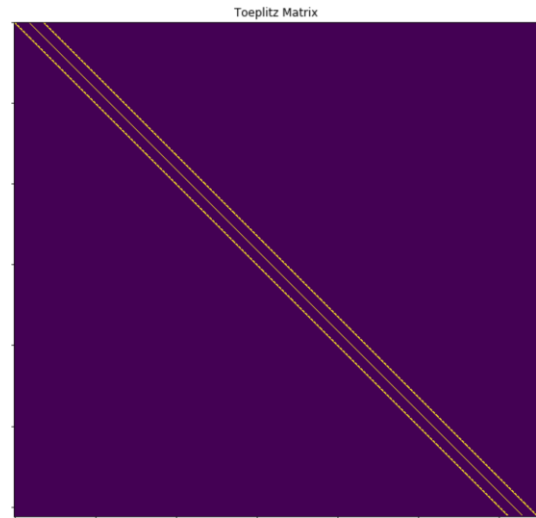


Figure 4. A Toeplitz Matrix to Simulate Convolution with a Dynamic Kernel

RELATED WORK

The prior work regarding these methods and datasets is vast, although somewhat scattered. For example, a number of CNNs have been developed for classification of skin lesions using the HAM10000 data, especially considering this task was posted as a Kaggle challenge [3][6][7]. And some of these networks perform quite well, achieving accuracies of above 90% [7]. However, these analyses are not necessarily performed with an eye towards optimizing physical parameters of imaging systems that would capture these skin lesions.

Other papers examine how different image alterations, such as blurring by the static kernel, affect classification by neural networks. One such study finds that while these distortions tend to have little or no affect on classification of simple image sets, such as MNIST, they can significantly and adversely affect classification of more complex sets such as CIFAR10 [8]. Still, these papers do not also consider dynamic blur kernels in their models, or how these blur kernels may interact with other physical parameters such as variable illumination.

Although only a small portion of related work is described here, there are still many factors to be explored in this field of imaging.

METHODS

The MNIST, Fashion MNIST, and CIFAR10 datasets were all downloaded from the TensorFlow Keras API, and the HAM10000 images were acquired from the public Harvard Dataverse. For reasons of computer memory, the sake of simplicity, and the ease of comparing results to those for the other three image sets, the skin lesion images were converted to 32x32x3 images using the Python `resizeimage` module. Additionally, as mentioned

in the introduction, the CIFAR10 and HAM10000 color images were converted to grayscale. This conversion was achieved either by learning the RGB weights or, in the cases where the weights were not trained, using the luminance method as described in the following equation [5].

$$\text{Grayscale} = 0.21 * R + 0.72 * G + 0.07 * B$$

Each of these four datasets was trained using similar physical parameter variables and simple CNNs. Every model was trained for 2500 iterations and a learning rate of 0.01 or 0.001 was used, depending on the value for which training loss converged to a non-infinite number. Using a greater number of iterations would have been unproductive, as the training loss had reached a minimum at or before 2500 steps. Pseudocode for the CNNs used are shown in the table below.

MNIST/Fashion MNIST CNN	CIFAR10/HAM10000 CNN
2 x 2D Conv – 32 Filters	2 x 2D Conv – 32 Filters
1 x 2D Max Pooling	1 x 2D Max Pooling
2 x 2D Conv – 64 Filters	2 x 2D Conv – 64 Filters
1 x 2D Max Pooling	1 x 2D Max Pooling
1 x 1D Dense – 512 units	2 x 2D Conv – 256 Filters
1 x 1D Dense – 10 units	2 x 1D Dense – 512 units
Softmax Cross Entropy Loss	1 x 1D Dense – 10 or 7 units (Above depends on data)
	Softmax Cross Entropy Loss

Table 1. CNN Psuedocode

Note that the CNN for CIFAR10 and HAM10000 datasets was only changed minimally to keep them somewhat comparable to the CNN used for MNIST and Fashion MNIST.

The following physical imaging parameters were trained separately with the CNN for each dataset: illumination matrix, static 3x3 blur kernel, dynamic blur kernel, and RGB channel weights (for CIFAR10 and HAM10000 only). Additionally, in a separate run, the illumination matrix and dynamic blur kernel were trained together with the CNN for each dataset, and in another run, the illumination matrix, static blur, and RGB weights (if applicable) were trained together with the CNN. These last two runs simulated the effects of combining multiple physical parameters into a single system. For each run, the physical parameters were all initialized with their own single random constant float between 0 and 1. In this way, every pixel in a physical layer would start with the same value; initializing these layers with varying randomized values was attempted but yielded no informative results.

RESULTS

Visualizations for the learned parameters in each run were recorded along with the validation accuracies. Note that certain images and values are returned in the code provided, but do not appear in this results section, such as sample images from before and after training and training loss plots. These omissions have been performed because no significant conclusions could be pulled from these data. Comparisons of the training loss plots across models showed no discernible pattern, and the sample images only appear to change slightly when modified with different physical layers (likely due to their small sizes of 28x28 or 32x32).

The first results table below shows the learned illumination matrices and static blur kernels learned separately (i.e. apart from each other) on each dataset. It also provides the

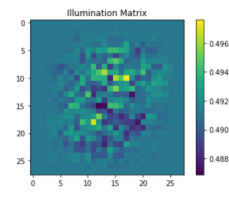
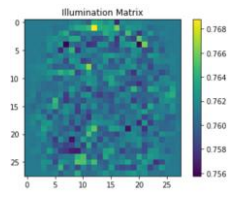
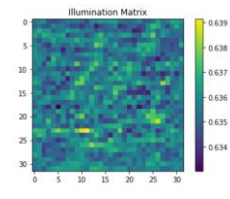
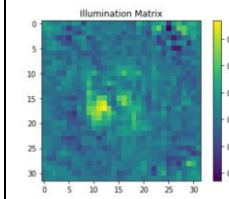
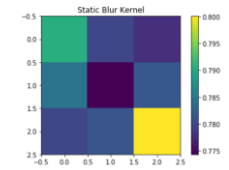
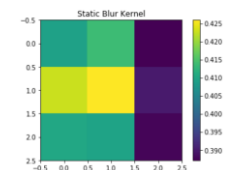
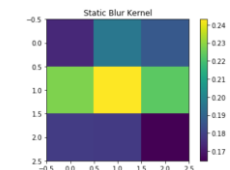
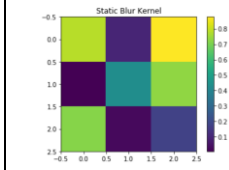
Dataset	MNIST	Fashion MNIST	CIFAR10	HAM10000
No Alterations Validation Accuracy	96.5%	86.3%	48.9%	31.7%
Illumination Only				
Validation Accuracy	96.4%	86.5%	45.1%	66.4%
Static Blur Kernel				
Validation Accuracy	96.6%	84.9%	43.1%	26.3%

Table 2. Initial Validation Accuracy and Illumination, Static Blur Kernel, and Their Individual Validation Accuracies

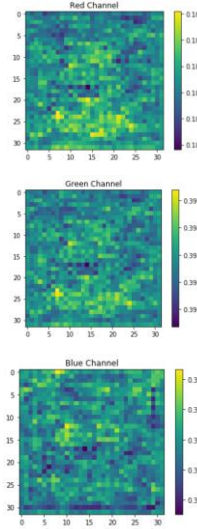
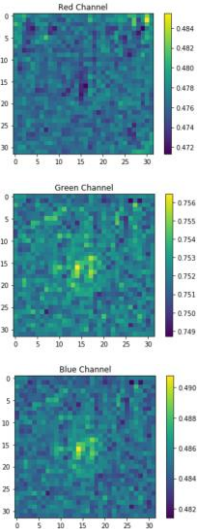
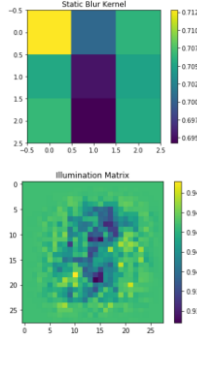
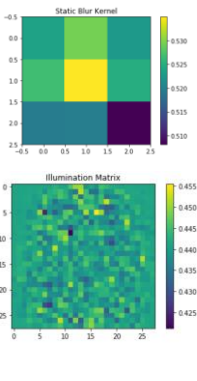
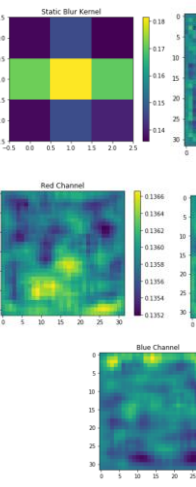
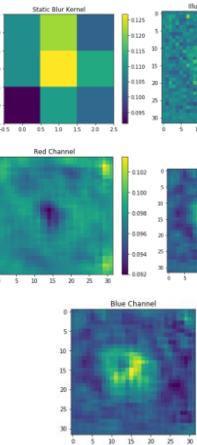
Dataset	MNIST	Fashion MNIST	CIFAR10	HAM10000
RGB Weights	N/A	N/A		
Validation Accuracy	N/A	N/A	47.8%	47.8%
Combined Physical Parameters				
Validation Accuracy	95.6%	86.2%	40.8%	25.5%

Table 3. Learned RGB Weights and Combined Physical Layers with Validation Accuracies

reference validation accuracies for the CNN models in which no physical parameters were trained (labeled “no alterations”).

In the second results table, Table 3, the learned RGB weight matrices are shown when trained on their own with the CNN. This table also contains the resulting physical parameters and validation accuracies for when the static kernel, illumination matrix, and RGB weights are all learned simultaneously with the CNN.

The final results table, shown on the next page, illustrates how the dynamic blur kernel affects the classification outcomes. For this table, the dynamic kernel matrices are shown as a

difference between the learned matrix and the starting matrix (i.e. from before training); this choice helps to more clearly demonstrate the behavior of this kernel, although the differences in values are still quite hard to see. The results for training the dynamic kernel with and without simultaneous training of an illumination matrix are shown. The two most interesting learned dynamic kernel matrices, those for the HAM10000 data, are expanded in the figure below this table.

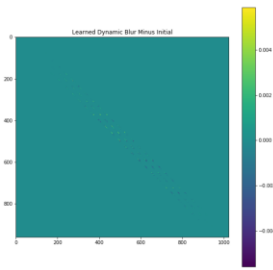
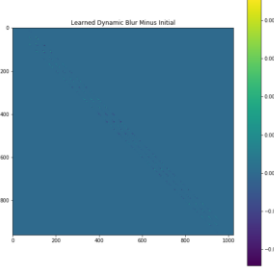
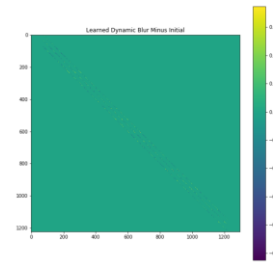
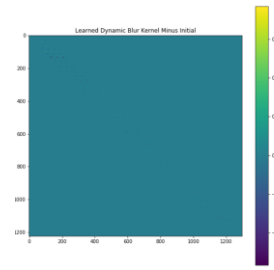
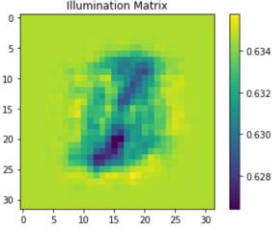
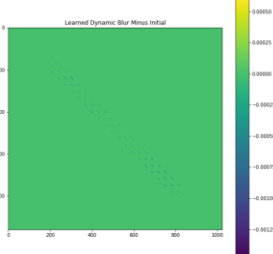
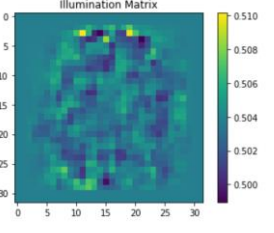
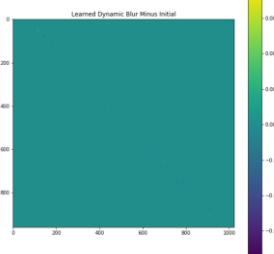
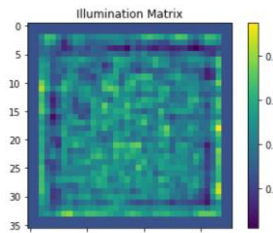
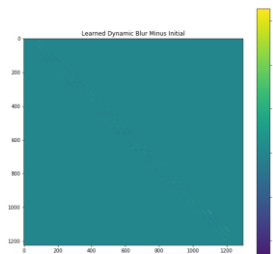
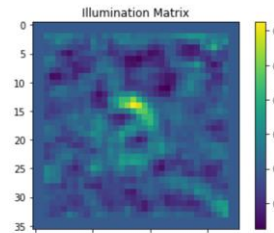
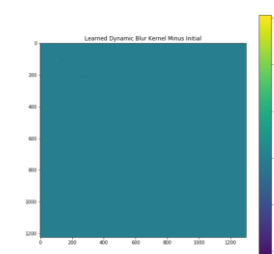
Dataset	MNIST	Fashion MNIST	CIFAR10	HAM10000
Dynamic Kernel w/o Illumination				
Validation Accuracy	96.9%	84.1%	48.0%	66.9%
Dynamic Kernel w/ Illumination	 	 	 	 
Validation Accuracy	96.9%	86.5%	48.9%	64.5%

Table 4. Learned Dynamic Kernels With and Without Learned Illumination

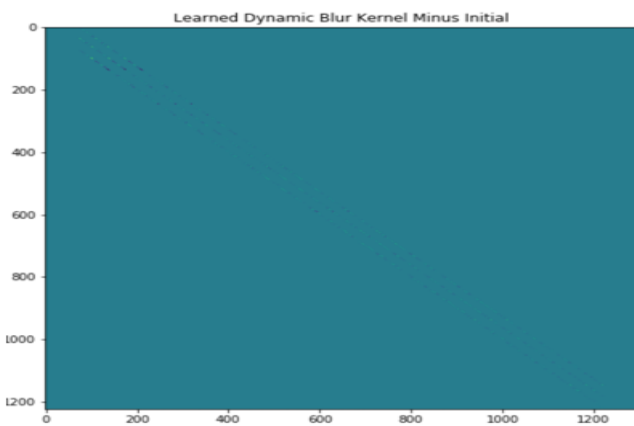


Figure 5. Left – Learned Dynamic Blur Kernel without Illumination for HAM10000

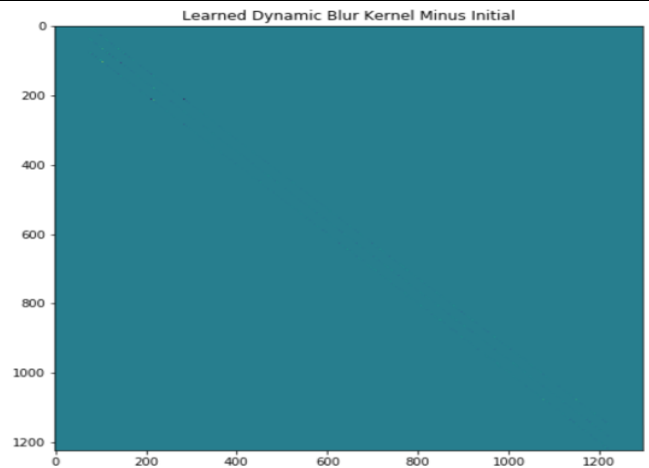


Figure 5. Right – Learned Dynamic Blur Kernel with Illumination for HAM10000

DISCUSSION

Most of the results obtained in these models do make quite a bit of sense. For a simple example, looking at the learned illumination matrices for the four datasets shows that unique illumination values tend to be learned in the regions that contain most of the information for the data. The matrix for MNIST shows the most variability in the pixels that are most likely to contain part of a number; the domain of this variability increases for Fashion MNIST and again for CIFAR10, as both of these data have important identifying features spread throughout the image. Finally, for the HAM10000 data, the illumination matrix has its highest values in the region of the image likely to contain the lesion, highlighting that spot and shadowing out the background. The most interesting part of this result is the fact that incorporating this illumination matrix as the sole physical layer results in a jump from 31.7% to 66.4% validation accuracy for the same CNN with HAM10000, suggesting that light contrast between the lesion and background are important for identification.

The learned RGB weights for both CIFAR10 and HAM10000 agree with the results discussed above, likely because these weights are just a variation on the illumination matrix. Instead of having a single illumination matrix, the RGB weights give a different illumination matrix to each of the three color channels. So, just as the illumination matrix didn't have much of an effect on CIFAR10 classification, the RGB weights similarly don't markedly change validation accuracy. And similar to the illumination matrix for the HAM10000 data, the learned RGB weights significantly bring the validation accuracy up to 47.8%. Another interesting result for this second dataset is that the red channel appears to move toward lower weights, while the green and blue channels in the region of the lesion move toward higher weights, suggesting that the blue and green tones help better classify the skin marking. Thinking about the skin tones in this dataset, the result again makes sense; red tones are more likely to appear in the background, healthy skin, while the darker green and blue tones would appear in the lesion.

The results for the learned static blur kernel agree with those results obtained by other researchers [8], in that blurring the images does not have much of an effect on classification performance for simple datasets like the MNIST or Fashion MNIST, but it does have a significant, negative affect on classification performance for the more complicated datasets. This blurring results in a drop of 5% or more for classification accuracy of CIFAR10 and HAM10000 images.

When combining the above three physical layers with a single CNN, we did not expect to see much of a change in validation accuracy for the MNIST, Fashion MNIST, or CIFAR10 datasets. Our main goal was to see if the gain in performance achieved by the illumination matrices for the HAM10000 dataset would outweigh the loss of performance imposed by the blur kernel. We observed that this tradeoff did not occur, and the affect of the blur kernel appeared to dominate the result. This outcome could suggest that the edges of the lesion are critical to its classification. Even with the illumination and RGB matrices

highlighting the pixels containing lesion information, the blur kernel, which would have the greatest affect on the lesion edges, still prevented an improvement in validation accuracy.

Given the effect of the static blur kernel on classification performance, we were unsure what impact the dynamic kernel would have. Again, with this changing kernel, the results for MNIST, Fashion MNIST, and CIFAR10 are as expected; there is no major effect on validation accuracy. However, the kernel has an unexpected, hugely positive affect on the HAM10000 data. Incorporation of the dynamic kernel as the only physical parameter resulted in a validation accuracy of 66.9%, comparable to that achieved with the illumination matrix. When illumination and the dynamic kernel are trained together, this improved performance is unaffected, and the validation accuracy comes to 64.5%. So it does not appear that these two important physical parameters have an additive effect on classification accuracy. The reason for this may be explained by the images in Figure 5. The left image shows how the Toeplitz matrix containing the dynamic kernel behaves when illumination is not trained with it, while the right image shows how it behaves when illumination is trained with it. There is significantly more variability in the left image than the right, and many of the dynamic kernel weights in the right image appear to be unchanged from their original value (recall that these matrices show final minus initial value and are largely zero). This result suggests that the illumination matrix and the dynamic kernel could have similar physical effects on the image data, if introducing an illumination matrix prevents learning of dynamic kernel weights without decreasing the validation accuracy. Additionally, if the illumination matrix is learned such that it highlights the lesion and dims the background, as stated previously, and if the dynamic kernel acts to blur the background but not the critical, information-containing region, then these two physical parameters should indeed have a similar effect.

To continue this work, we would first have to run these models many times to ensure that the results obtained are valid. And in the future, it would be interesting to see how more complicated CNNs, such as those that achieve approximately 90% accuracy on the HAM10000 data, would interact with the physical parameters examined here.

REFERENCES

- [1] D. Cardinal (2018). "Stanford Researchers Build AI Directly Into Camera Optics." <https://www.extremetech.com>
- [2] Joseph W. Goodman. 1996. *Introduction to Fourier Optics* (2nd ed.). McGraw-Hill Companies, Inc, New York, Chapter 6.
- [3] P. Tschandl, C. Rosendahl, and H. Kittler (2018). "The HAM10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions." *Nature Scientific Data*. 14(5).
- [4] A. Vasudevan, A. Anderson, and D. Gregg (2017). "Parallel Channel Convolution using General Matrix Multiplication."
- [5] R. M. H. Nguyen and M. S. Brown (2017). "Why You Should Forget Luminance Conversion and Do Something Better." 2017 IEEE Conference on Computer Vision and Pattern Recognition.
- [6] S. Nasiri, M. Jung, J. Helsper, and M. Fathi (2018). "Deep-CLASS at ISIC Machine Learning Challenge 2018."
- [7] Y. Ge, B. Li, Y. Zhao, E. Guan, and W. Yan (2018). "Melanoma Segmentation and Classification in Clinical Images Using Deep Learning." *Proceedings of the 2018 10th International Conference on Machine Learning and Computing*.
- [8] Y. Zhou, S. Song, and N. M. Cheung (2017). "On Classification of Distorted Images with Deep Convolutional Neural Networks."