Convolutional Neural Networks

on Randomly Colorized MNIST Data

Jocelyn Griser

griser.jocelyn@gmail.com

Abstract

Image classification is an ongoing challenge for Artificial Intelligence researchers. In this paper, black-and-white images from the MNIST dataset will be purposefully colorized to determine a convolutional neural network's robustness with handling irrelevant details. Multiple neural network architectures will be trained and tested, including ones based off of existing, published models.

Keywords

Image classification, convolutional neural networks, color images

Introduction

There is an ongoing challenge to create artificial neural networks that can perform image recognition as well as a human. One assumption made for both ANNs and human cognition is that more information is better. Humans however know when to ignore certain information, e.g., if a human is presented with images of a blue bus and a yellow bus, the human will know that they are both buses. In this work, an ANN will be developed and tested to try to ignore color information when it is irrelevant to categorization. By determining the difficulty of training ANNs to do this, image classification projects can better decide whether a semi-supervised learning approach using image pre-processing is more efficient than unsupervised learning.

Literature Review

Convolutional Neural Networks (CNNs) were originally developed with image classification in mind. The advent of CNNs was reported in a journal article (LeCun, Boser, et al. 1989) where the experiment performed was identifying handwritten zip codes provided by the U.S. Postal Service. The images of this dataset were black-and-white, meaning that each pixel of

the image could be expressed with just one number, i.e., it's value (also referred to as "lightness" or "brightness").

When moving into full-color images, each pixel became a set of three numbers, either expressed as Hue-Saturation-Value (HSV) or Red-Green-Blue (RGB). The HSV color space was developed specifically to be a digital alternative to the already-established RGB color space (Smith 1978). NN models have been experimented with both RGB (Rafegas and Vanrell 2018) (Li and Shui 2021) and HSV (Yang, et al. 2018). There have also been papers where a color space is specifically modified for the research, such as RGB-PCA (Red-Blue-Green Principal Component Analysis) (Flachot and Gegenfurtner 2021). The advancement of NNs to handle color information has been a great boon to image classification. For example, if trying to distinguish an apple from an orange, adding the dimensions of color helps the NN classify more accurately.

Data

The data being used is the "modified National Institute of Standards and Technology" database (MNIST) of handwritten digits (LeCun, Cortes and Burges, The MNIST database of handwritten digits n.d.). The MNIST database has been "used extensively in optical character recognition and machine learning research" (L. Deng 2012). It is a set of 70,000 greyscale images of handwritten digits expressed as 784 pixels (28 x 28), each pixel with a value ranging from 0 to 254.

For this paper, the dataset will be used as-is and will also be modified to transform the dataset into two colorized versions. In the colorized sets, the images will be expressed as 784 pixels with each pixel expressed as an RGB value in byte form. The RGB to byte transformation

will be done using Pillow (Clark 2021), a fork for the open-source Python Imaging Library (Lundh 2009).

In the first color version, each image in the dataset will be transformed to a monochromatic color image. First the value of each pixel (a lightness value (V_0) between 0 and 254) will be rescaled to be between 55 and 254 (V_1) (meaning the darkest pixel will not be black) and then to a percentage out of 255 (V'). A hue (H), expressed as a number between 0 and 359, will be randomly selected and then each pixel will be transformed from a Hue-Saturation-Value (HSV) format (using $S = 1$ for all pixels) into a RGB format (Figure 1).

In the second color version, a second randomized hue will be used at the background color. Both the foreground color and background color will be converted into RGB format, and then the individual pixels of the image will use their lightness value to determine where they are on the two-hue gradient (Figure 2).

$$C = S * (1 - |2 * V' - 1|)$$
$$X = C * \left(1 - \left|rem\left(\frac{H}{60}, 2\right) - 1\right|\right)$$
$$m = V' - \frac{C}{2}$$

$\longrightarrow$

	$H < 60$	$60 \leq H < 120$	$120 \leq H < 180$	$180 \leq H < 240$	$240 \leq H < 300$	$300 \leq H$
$R =$	C	X	0	0	X	C
$G =$	X	C	C	X	0	0
$B =$	X	X	0	C	C	0

Figure 1 - HSV to RGB conversion

Hue 1 $\rightarrow [R_1, G_1, B_1]$

Hue 2 $\rightarrow [R_2, G_2, B_2]$

$\rightarrow$

For each pixel, where V_p is the given lightness value:

$$x_p = x_1 + \frac{V_p}{255} * (x_2 - x_1), x \in [R, G, B]$$

Figure 2 – Two-color RGB equations

Examples of colorization:

Figure 3- First 49 images of MNIST

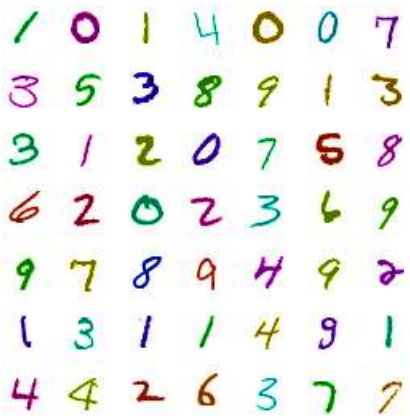

Figure 4 - First 49 digits of MNIST, randomly colorized with a single hue

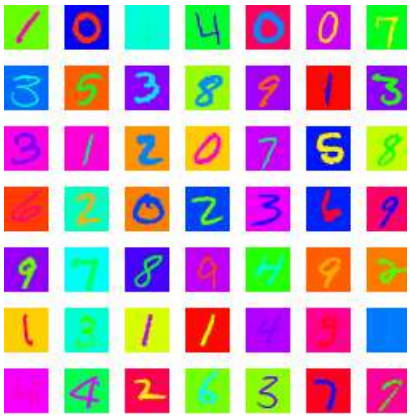

Figure 5 - First 49 digits of MNIST, randomly colorized with two hues

Methods

Two convolutional neural network models will be used to train on both the black-and-white dataset and the colorized datasets. The first is LeNet, named after the creator Yan LeCun, and introduced in the 1998 paper "Gradient-Based Learning Applied to Document Recognition". There have been variations and updates of this model (Sik-Ho 2018), the latest being LeCun-5 (LeCun, LeNet-5, convolutional neural networks 2010). This model was specifically developed using the MNIST dataset being used and modified for this paper. The model has 5 layers: 2 convolution layers (each followed by a pooling layer), and 3 dense layers.

The second model is AlexNet, named after Alex Krizhevsky and introduced in his co-authored 2012 paper "ImageNet Classification with Deep Convolutional Neural Networks". This model was developed for the 2012 ImageNet Large Scale Visual Recognition Challenge (ILSVRC-2012) using the using the ILSVRC-2010 dataset, which had 1.2 million, 50,000, and 150,000 images in the training, validation, and testing sets respectively (Krizhevsky, Sutskever and Hinton 2012). The images in the dataset are color photographs of various sizes spanning

1000 object categories (Figure 6). For the purposes of their model, Krizhevsky, Sutskever and Hinton cropped all images to 256 by 256 pixels. The AlexNet model contains 8 layers, all using ReLU as the activation function: 5 convolution layers (3 of which are followed by pooling layers) and 3 dense layers (2 of which are followed by dropout of p=0.5) (Krizhevsky, Sutskever and Hinton 2012). The model's architecture had to be modified slightly for this paper to fit the smaller size of the images (Figure 6).

Figure 6 – Two examples from the ImageNet database

By using model architectures developed for a simple dataset and then more detailed one, the first model might prove to be not robust enough for the color modifications made to the dataset, while the other model might prove too detailed for the task and lead to overfitting.

AlexNet – Original	AlexNet – Modified, B/W	AlexNet – Modified, RGB
Image Size: 224 x 224	Image Size: 32 x 32	Image Size: 32 x 32
1. Convolution – ReLU	**1. Convolution – Sigmoid**	**1. Convolution – Sigmoid**
11 x 11 Kernel +4 Stride	11 x 11 Kernel +2 Stride	5 x 5 Kernel +2 Stride
2. Pooling	**2. Pooling**	**2. Pooling**
3 x 3 Kernel +2 Stride	3 x 3 Kernel +1 Stride	3 x 3 Kernel +1 Stride
3. Convolution – ReLU	**3. Convolution – Sigmoid**	**3. Convolution – Sigmoid**
5 x 5 Kernel +2 Padding	5 x 5 Kernel +1 Padding	3 x 3 Kernel +1 Padding
4. Pooling	**4. Pooling**	**4. Pooling**
3 x 3 Kernel +2 Stride	3 x 3 Kernel +1 Stride	3 x 3 Kernel +1 Stride
5. Convolution – ReLU	**5. Convolution – ReLU**	**5. Convolution – ReLU**
3 x 3 Kernel +1 Padding	2 x 2 Kernel +1 Padding	2 x 2 Kernel +1 Padding
6. Convolution – ReLU	**6. Convolution – ReLU**	**6. Convolution – ReLU**
3 x 3 Kernel +1 Padding	2 x 2 Kernel +1 Padding	2 x 2 Kernel +1 Padding
7. Convolution – ReLU	**7. Pooling**	**7. Pooling**
3 x 3 Kernel +1 Padding	2 x 2 Kernel +2 Stride	2 x 2 Kernel +2 Stride
8. Pooling	**8. Flatten**	**8. Flatten**
3 x 3 Kernel +2 Stride	**9. Dense – ReLU, dropout p=0.5**	**9. Dense – ReLU, dropout p=0.5**
9. Flatten	288 fully connected neurons	800 fully connected neurons
10. Dense – ReLU, dropout p=0.5	**10. Dense – ReLU, dropout p=0.5**	**10. Dense – ReLU, dropout p=0.5**
4096 fully connected neurons	288 fully connected neurons	800 fully connected neurons
11. Dense – ReLU, dropout p=0.5	**11. Dense – ReLU**	**11. Dense – ReLU**
4096 fully connected neurons	100 fully connected neurons	100 fully connected neurons
12. Dense – ReLU	**12. Dense – ReLU**	**12. Dense – ReLU**
1000 fully connected neurons	10 fully connected neurons	10 fully connected neurons

Figure 7: AlexNet architecture: original and modified

Results

For the AlexNet model, various modifications were made to better suit the input images and the task; kernel size, number of strides, padding, number of layers, and number of neurons (for the dense layers) were experimented with. The two modified architectures listed in Figure 7 were ultimately the variations used. The LeNet model received no modification except for the input shape for the colorized models. The accuracy and loss statistics for the LeNet and final modified AlexNet models are listed in Table 1.

Dataset	***Original***		***Colorized, 1 Hue***		***Colorized, 2 Hue***	
Model	**LeNet**	**AlexNet**	**LeNet**	**AlexNet**	**LeNet**	**AlexNet**
Acc.	0.962	0.916	0.938	0.751	0.883	0.100
Val. Acc.	0.961	0.947	0.958	0.771	0.872	0.109
Loss	0.121	0.383	1.185	3.268	0.337	5.050
Val. Loss	0.121	0.234	0.129	3.166	0.331	4.962

Table 1: Statistics for final model results

The statistics for both types of models for the original dataset had standard adjustment curves for accuracy and loss, as did the LeNet model for the 2-Hue colorized set. The models for the 1-Hue colorized set both showed periods of over-fitting and then correction (Figures 8 – 11). The AlexNet model for the 2-Hue dataset showed no improvements over epochs (Figures 12, 13).

The first five models showed similar distributions for their classification mistakes (Figures 14 – 18) whereas the AlexNet model for the 2-Hue dataset classified all images as the same digit (Figure 19).

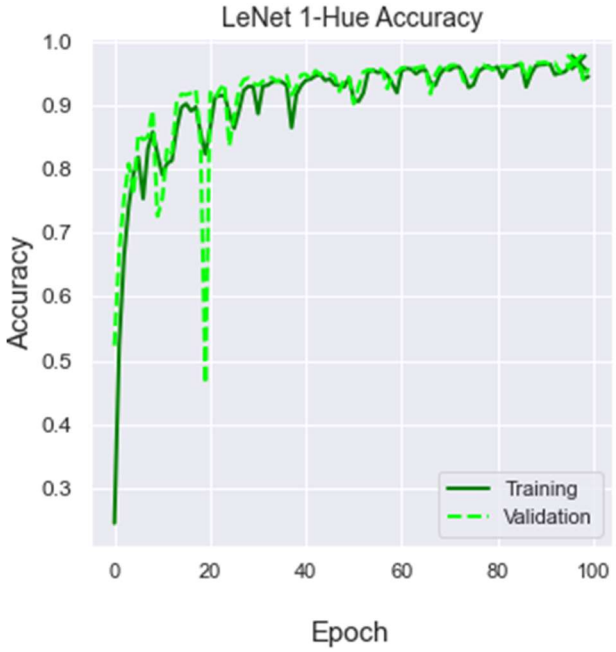

Figure 8 - LeNet 1-Hue Accuracy

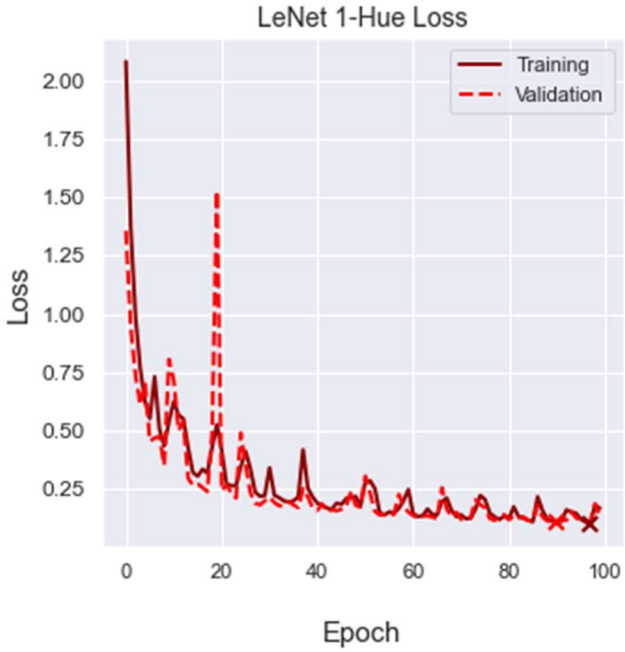

Figure 9 - LeNet 1-Hue Loss

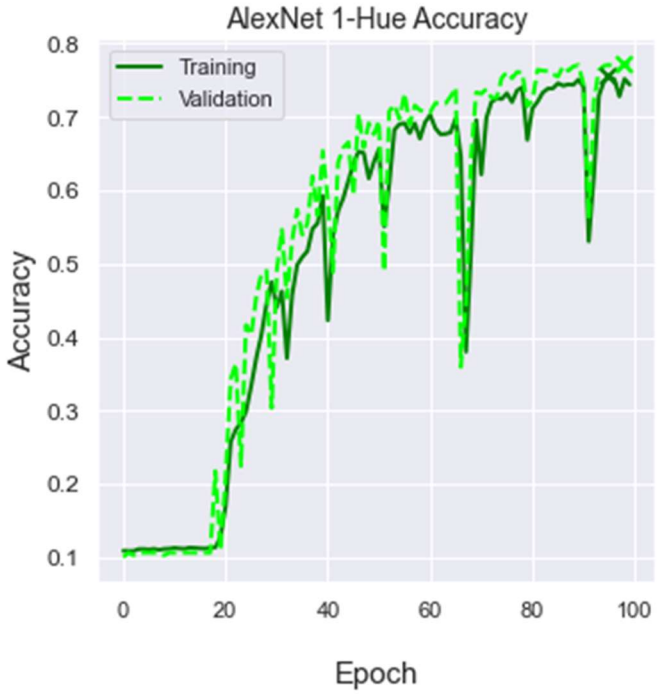

Figure 10 - AlexNet 1-Hue Accuracy

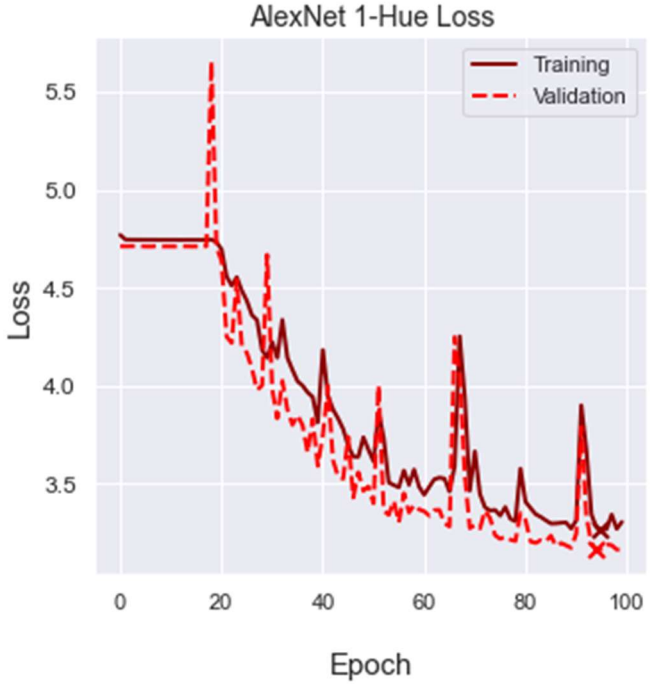

Figure 11 - AlexNet 1-Hue Loss

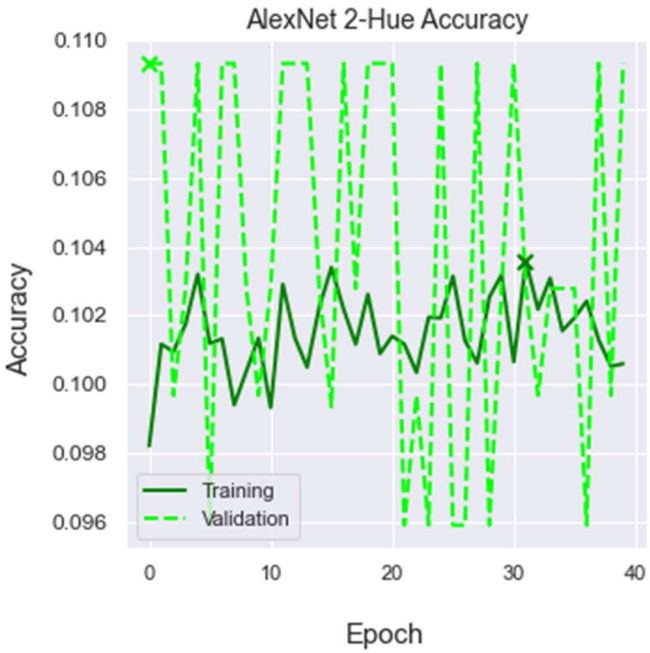

Figure 12 - AlexNet 2-Hue Accuracy

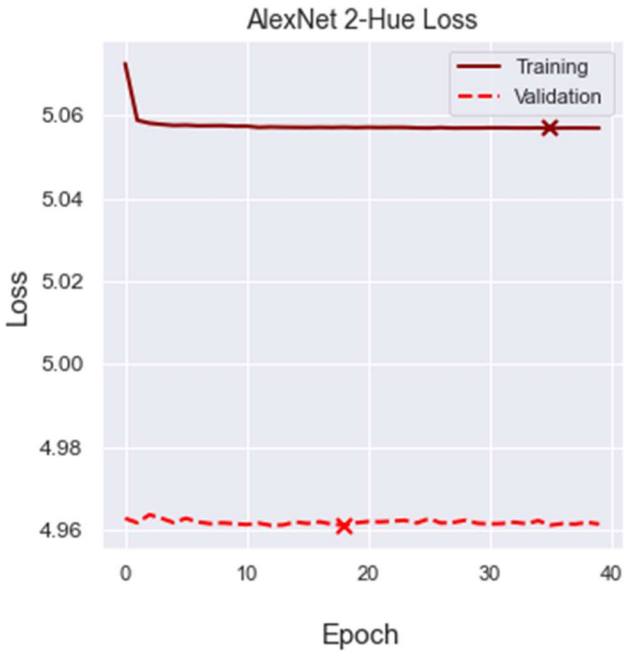

Figure 13 - AlexNet 2-Hue Loss

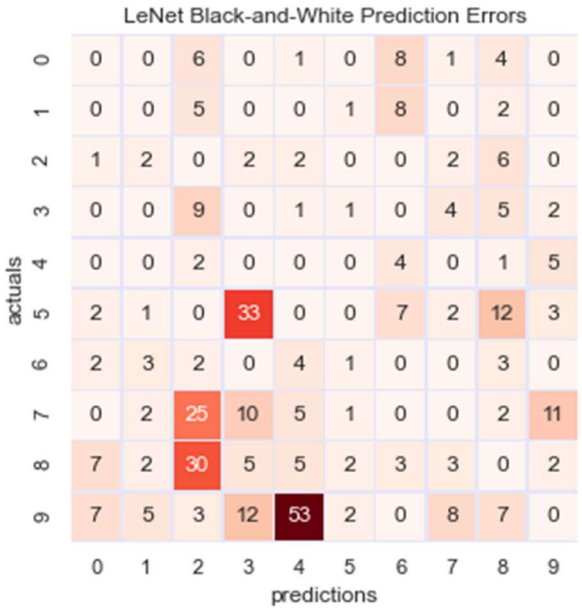

Figure 14 - LeNet Black and White Prediction Errors

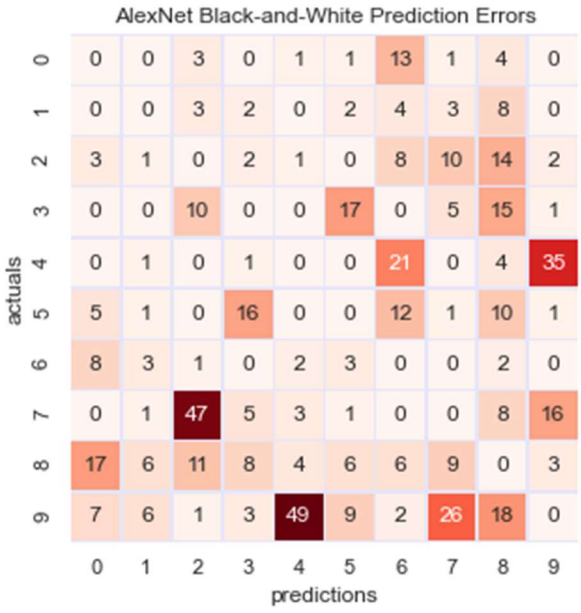

Figure 15 - AlexNet Black and White Prediction Errors

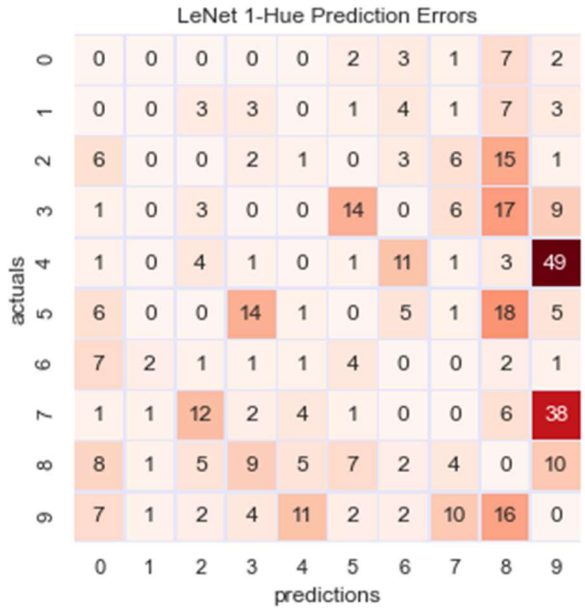

Figure 16 - LeNet 1-Hue Prediction Errors

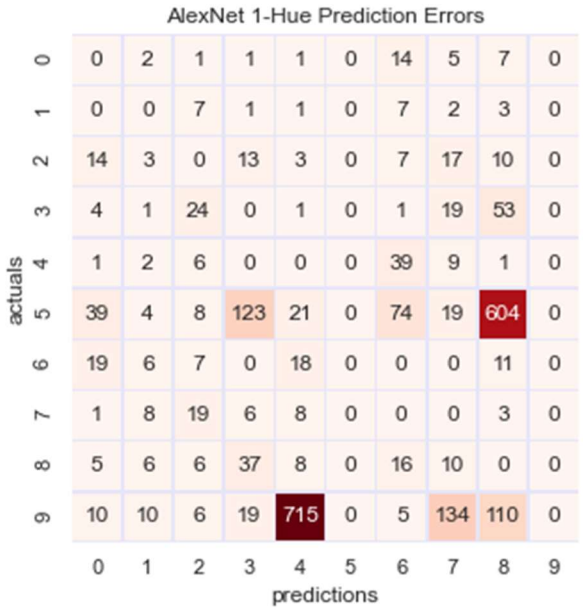

Figure 17 - AlexNet 1-Hue Prediction Errors

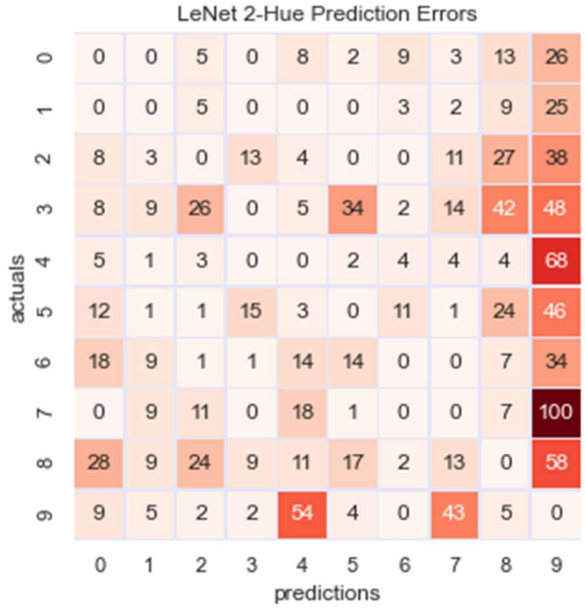

Figure 18 - LeNet 2-Hue Prediction Errors

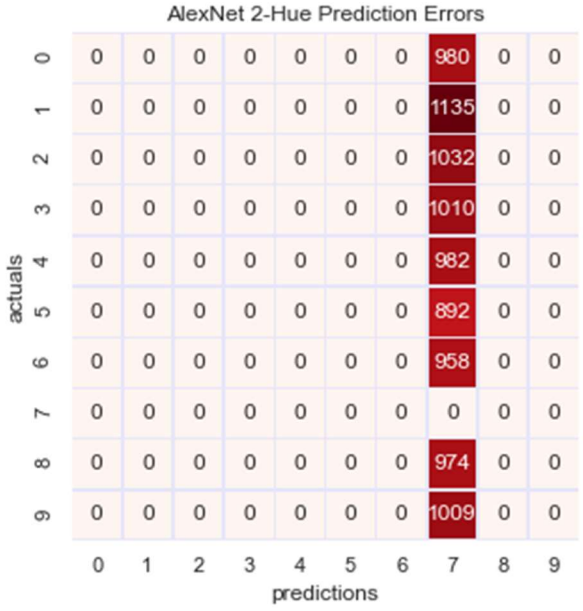

Figure 19 - AlexNet 2-Hue Prediction Errors

Analysis and Interpretation

With all three datasets, the LeNet model well outperformed the AlexNet model, regardless of the modifications made to the AlexNet architecture. While the modified AlexNet model performed much closer to the LeNet model with the original, non-colorized MNIST dataset, it fell well behind with the two colorized datasets.

In the context of what these models were originally developed for, the LeNet models' superior performance over the AlexNet models' does make sense. While the colorization of the dataset did add more information that the model needed to attune itself to, the AlexNet model was designed for not only larger but more complex images and into far more categories. The photographs in the ImageNet dataset have multitudes of colors; they are not two-toned (Figure 6).

The differences in accuracy between the 1-hue and the 2-hue datasets was significant for both models. The LeNet model was still able to distinguish the digits with some accuracy whereas the AlexNet model failed to make any differentiation between the digits. The changes here are likely because with the 1-hue model, there is still a guaranteed starkness between the "foreground" and the "background". Regardless what hue is randomly chosen, the RGB values will still be very distinct from white, ex. comparing (255, 0, 0) to (0, 0, 0). When two hues are randomly chosen, the difference between foreground and background might be much subtler, ex. comparing (255, 0, 0) to (255, 160, 0).

In the end, no modifications made to the AlexNet architecture could make it work with the 2-hue colorized dataset. As seen in Figure 19, the model always concluded with every instance being classified as the same digit. The ultimate model used for the 2-hue set was the same architecture used for the 1-Hue set, but the architecture used for the non-colorized set worked just as well. (A similar phenomenon was observed when trying to use the 1-hue architecture for the non-colorized set and vice versa.) This is likely due to the number of convolutions: if the instances distinction between foreground and background is very slight from the outset, then the multiple convolutions will just serve to blur the two hues further rather than making the distinction starker.

The singular class chosen in the AlexNet 2-hue model changed over iterations; the instances were all classified as ‘7’ in the final attempt, but previously ‘1’, ‘4’, and ‘9’ had also been selected as the sole class. These four digits were also common prediction errors for each other in the preceding models. It can therefore be inferred that once the AlexNet 2-hue model had decided these four digits were of the same class, that class was larger than all of the others and then all other instances were “fitted” to this singular class.

The LeNet model for the 2-hue dataset seemed to have a similar issue (Figure 18), but not to the extent the AlexNet model did. The predicted ‘9’ class was greedy, containing most of the prediction errors, although there were still some of the actual ‘9’ class that got misclassified, despite the size of its predicted class.

Conclusions

These results serve as proof to the simple lesson of picking the right tool for a job. “Newer” and “more intricate” do not always mean better. In this paper specifically, the results show that modifying a model to work with more information can work better than modifying a model to work with less. The LeNet model has been the launching point for many neural networks, including AlexNet (Krizhevsky, Sutskever and Hinton 2012), but each are developed to suit their own purpose. There will almost certainly never be a one-size-fits-all algorithm. As such, in order for scientists to determine what tools to use, they need to understand the nature of their task and the limits and abilities of the tools available to them. Sometimes the simplest tool is the best tool, and trying to fit the latest, most-intricate model to a less-demanding problem can result in failure.

Directions for Future Work

This paper was limited in scope to a dataset of small images of a set dimension with little detail. Further exploration could be done with the same concept (modifying colors of images to obfuscate their subject matter) with larger and/or more complex images, such as with photographs in the ImageNet dataset.

Some NNs for image classification have been developed to do edge detection, i.e., not by the whole of the image but by the significant curves in it, detected by "where grayscale or color takes abrupt change" (Li and Shui 2021). Edge detection via NNs has been explored with black and white images (Torre and Poggio 1986) and then color images (Hansen and Gegenfurtner 2017) (Li and Shui 2021). CNNs in particular have been found to be particularly adept at edge detection, with a number of models being created specifically for this task, such as DeepContour (Shen, et al. 2015), DeepEdge (Bertasius, Shi and Torresani 2015), and HED (Holistically-Nested Edge Detection) (Xie and Tu 2015). Edge detection could prove useful for classifying mis-colored images by focusing on the shape of the edges in the image rather than the color of those edges.

References

Bertasius, Gedas, Jianbo Shi, and Lorenzo Torresani. 2015. "DeepEdge: A Multi-Sclae Bifurcated Deep Network for Top-Down Contour Detection." *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR).* IEEE. 4380-4389. doi:10.1109/CVPR.2015.7299067.

Clark, Alex. 2021. *Pillow (PIL Fork) Release 8.2.0.* April 1. https://pillow.readthedocs.io/en/stable/.

Deng, Jia, Alex Berg, Sanjeev Satheesh, Hao Su, Aditya Khosla, and Fei-Fei Li. 2012. *ImageNet Large Scale Visual Recognition Challenge 2012 (ILSVRC2012).* Accessed February 27, 2022. https://image-net.org/challenges/LSVRC/2012/.

Deng, Li. 2012. "The MNIST Database of Handwritten Digit Images for Machine Learning Research [Best of Web]." *IEEE Signal Processing Magazine* (IEEE) 29 (6): 141-142. doi:10.1109/MSP.2012.2211477.

Flachot, Alban, and Karl R. Gegenfurtner. 2021. "Color for object recognition: Hue and chroma sensitivity in the deep features of convolutional neural networks." *Vision Research* 182: 89-100. doi:https://doi.org/10.1016/j.visres.2020.09.010.

Hansen, Thorsten, and Karl R. Gegenfurtner. 2017. "Color contributes to object-contour perception in natural scenes ." *Journal of Vision* 17 (3). doi:10.1167/17.3.14.

Kaziha, Omar, and Talal Bonny. 2019. "A Comparison of Quantized Convolutional and LSTM Recurrent Neural Network Models Using MNIST." *International Conference on Electrical and Computing Technologies and Applications (ICECTA).* IEEE. 1-5. doi: 10.1109/ICECTA48151.2019.8959793.

Krizhevsky, Alex, Ilya Sutskever, and Geoffrey H. Hinton. 2012. "ImageNet Classification with Deep Convolutional Neural Networks." *Advances in Neural Information Processing Systems.* 1097-1105. doi:10.1145/3065386.

LeCun, Yann. 2010. *LeNet-5, convolutional neural networks.* March 2. http://yann.lecun.com/exdb/lenet/.

LeCun, Yann, Bernhard E. Boser, John S. Denker, Donnie Henderson, Richard E. Howard, Wayne E. Hubbard, and Lawrence D. Jackel. 1989. "Backpropagation Applied to Handwritten Zip Code Recognition." *Neural Computation* 1 (4): 541-441. doi:10.1162/neco.1989.1.4.541.

LeCun, Yann, Corinna Cortes, and Christopher J.C. Burges. n.d. *The MNIST database of handwritten digits.* Accessed January 15, 2022. http://yann.lecun.com/exdb/mnist/.

LeCun, Yann, Léon Bottou, Yoshua Bengio, and Patrick Haffner. 1998. "Gradient-Based Learning Applied to Document Recognition." *Proceedings of the IEEE* 86 (11): 2278 - 2324. doi:10.1109/5.726791.

Li, Ou, and Peng-Lang Shui. 2021. "Color edge detection by learning classification network with anisotropic directional derivative matrices." *Pattern Recognition* 118. doi:10.1016/j.patcog.2021.108004.

Lundh, Fredrik. 2009. *Python Image Library 1.1.7.* November 15.

Rafegas, Ivet, and Maria Vanrell. 2018. "Color encoding in biologically-inspired convolutional neural networks." *Vision Research* 151: 7-17. doi:10.1016/j.visres.2018.03.010.

Russakovsky, Olga, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, et al. 2015. *ImageNet Large Scale Visual Recognition Challenge.* January 30. https://arxiv.org/abs/1409.0575.

Shen, Wei, Xinggang Wang, Yan Wang, Xiang Bai, and Zhijiang Zhang. 2015. "DeepContour: A deep convolutional feature learned by positive-sharing loss for contour detection." *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR).* IEEE. 3982-3991. doi:10.1109/CVPR.2015.7299024.

Sik-Ho, Tsang. 2018. *Review: LeNet-1, LeNet-4, LeNet-5, Boosted LeNet-4 (Image Classification).* August 8. https://sh-tsang.medium.com/paper-brief-review-of-lenet-1-lenet-4-lenet-5-boosted-lenet-4-image-classification-1f5f809dbf17.

Smith, Alvy Ray. 1978. "Color gamut transform pairs." *Computer Graphics* 12 (3): 12-19. doi:10.1145/965139.807361.

Torre, Vincent, and Tomaso A. Poggio. 1986. "On Edge Detection." *IEEE Transcations on Pattern Analysis and Machine Intelligence* PAMI-8 (2): 147-163. doi:10.1109/TPAMI.1986.4767769.

Xie, Saining, and Zhuowen Tu. 2015. "Holistically-Nested Edge Detection." *2015 IEEE International Conference on Computer Vision (ICCV).* Santiago, Chile: IEEE. doi:10.1109/ICCV.2015.164.

Yang, Zhen, Weilan Shi, Zhiyi Huang, Zhijin Yin, Fan Yang, and Meichan Wang. 2018. "Combining Gaussian Mixture Model and HSV Model with Deep Convolution Neural Network for Detecting Smoke in Videos." *2018 IEEE 18th International Conference on Communication Technology (ICCT).* Chongqing, China: IEEE. doi:10.1109/ICCT.2018.8599905.