Color-Dependent Prediction Stability of Popular CNN Image Classification Architectures

Laurent Mertens[1,2,3](✉), Elahe' Yargholi[4], Jan Van den Stock[5,6], Hans Op de Beeck[4], and Joost Vennekens[1,2,3]

[1] Department of Computer Science, KU Leuven, De Nayer Campus, J.-P. De Nayerlaan 5, 2860 Sint-Katelijne-Waver, Belgium

[2] Leuven.AI - KU Leuven Institute for AI, 3000 Leuven, Belgium
`laurent.mertens@kuleuven.be`

[3] Flanders Make@KU Leuven, Leuven, Belgium

[4] Department of Brain and Cognition, Faculty of Psychology and Educational Sciences, Leuven Brain Institute, KU Leuven, 3000 Leuven, Belgium

[5] Neuropsychiatry, Leuven Brain Institute, KU Leuven, 3000 Leuven, Belgium

[6] Geriatric Psychiatry, University Psychiatric Center KU Leuven, 3000 Leuven, Belgium

Abstract. The ImageNet-1k dataset has been a major contributor to the development of novel CNN-based image classification architectures over the past 10 years. This has led to the advent of a number of models, pre-trained on this dataset, that form a popular basis for creating custom image classifiers by means of transfer learning. A corollary of this process is that whatever weaknesses and biases the original model possesses, the derived model will also have. Some of these have already been extensively covered, but color sensitivity has so far been understudied. This paper explores the prediction stability of several popular CNN architectures when input images are subjected to hue or saturation shifts. We show that even small shifts in image hue can alter a model's initial prediction, with larger shifts introducing changes up to 60% and 40% of the time for AlexNet and VGG16 respectively. For all models considered, saturation changes have less impact. To illustrate the issue being inherited by models obtained through transfer learning, we confirm that EmoNet, a model derived from AlexNet, exhibits similar behavior. By further comparing a same architecture trained separately on ImageNet-1k, Places365 and Stylized ImageNet, we confirm that the issue is shared across datasets. Finally, we propose a new preprocessing data augmentation to alleviate this problem.

Keywords: Convolutional Neural Networks · Image Recognition · Reliable Machine Learning · Robust Machine Learning · Trustworthy AI

L. Iliadis et al. (Eds.): ICANN 2023, LNCS 14254, pp. 491–503, 2023.
https://doi.org/10.1007/978-3-031-44207-0_41

1 Introduction

ImageNet[1] [2] is a large, publicly available, image dataset (14M+ images). Its images are organized according to the WordNet hierarchy, making it especially useful for image classification tasks, as target labels are readily available. In 2010, the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [12] introduced a particular subset of images from 1000 different categories known as the ImageNet-1k dataset, with an accompanying image classification challenge.

In 2012, a Convolutional Neural Network (CNN) now widely known as AlexNet [9] convincingly won this competition. This result led to quick and widespreak adoption of CNNs to solve image classification and recognition tasks; while AlexNet was the only CNN submitted in 2012, by the next year the majority of submissions were CNN-based. Other popular architectures that were either submitted to ILSVRC or trained on the ImageNet-1k dataset, and that will be evaluated in this paper, are VGG16 [13], ResNet18 and ResNet50 [5], and DenseNet161 [7].

Despite their popularity and successes, these architectures also have weaknesses, both ethical [1,14,16] and technical. The most famous in this latter category are arguably adversarial attacks [15]: tiny alterations to an original image, imperceptible to the human eye, that fool the network into misclassifying the image. Also visible alterations such as blurring, pixelation, addition of several types of noise, etc., severly impact model performance [6]. In summary, these networks tend to perform very well on the type of data they are trained on, but fail to generalize beyond that.

Within this context, this paper focuses on a type of alteration that seems understudied, namely color changes. Moreover, existing work, such as [3,11], focuses on comparison between the human vision system and CNNs. Both studies use models trained on ImageNet-1k—the former VGG-M and the latter AlexNet, VGG16 and VGG19—to investigate the color sensitivity and selectivity of unique CNN filters and layers. Their interest lies in decyphering how these CNNs encode color, and to what extent this overlaps with biological systems. The results obtained in [3] state that overall, the models they studied are more sensitive to changes in hue than in saturation, and that both affect model accuracy. Both results will be discussed and compared to our results below. Our focus lies solely on how color affects model robustness and performance. Although complete color invariance is not desireable, neither should a useful CNN model alter its predictions when small color shifts, that would not fool humans, are applied to images. An example of undesireable behavior is depicted in Fig. 1, which shows AlexNet misclassifying an originally correctly classified image when it is subjected to modest hue shifts. In this context, it is interesting to note that the original AlexNet paper [9] describes a data augmentation scheme that (last paragraph Sect. 5.1):

[1] https://www.image-net.org/.

> [...] approximately captures an important property of natural images, namely, that object identity is invariant to changes in the intensity and color of the illumination.

For some reason, this specific augmentation disappeared from later implementations, e.g., the PyTorch implementation we use.

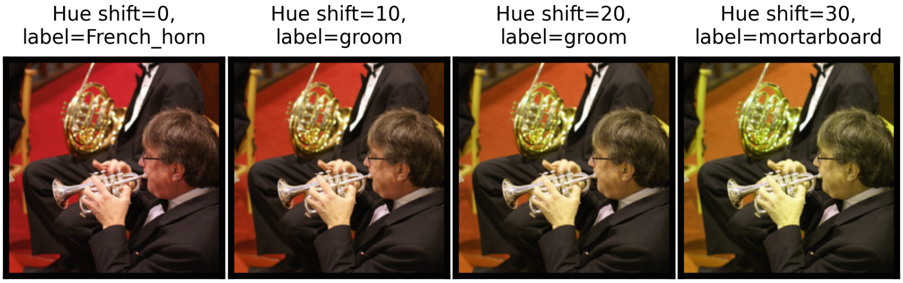

Fig. 1. Example of AlexNet sensitivity to hue shifts, expressed in degrees.

We start by investigating the effect of applying hue and saturation shifts to ImageNet-1k images on ImageNet-1k trained models, both in terms of prediction robustness—i.e., does a prediction for an altered image differ from that of the original image, regardless of the correctness of that original prediction?—and accuracy. Next, we turn our attention to EmoNet [8], an image classification model obtained by taking AlexNet trained on ImageNet-1k, replacing its last layer with a 20 node linear layer and training only this new layer on a custom dataset of 137k images annotated with one of 20 emotion labels representing the emotion elicited by the images in an observer. The question we want to answer is to what extent this model obtained by means of transfer learning inherits its parent's properties. EmoNet forms an interesting case, because elicited emotions form a dimension that can also reasonably be assumed to be independent of moderate color changes; a few degrees of hue shift shall not make a puppy less cute. Following this, we look at some of the earlier mentioned CNNs, but trained from scratch on different large datasets. In particular, we consider Stylized ImageNet [4], a dataset derived from ImageNet-1k by means of style transfer, and Places365 [17], a dataset of millions of images annotated with one of 365 scene classes. By comparing the effect of color-related changes on a same architecture trained on different datasets, we determine if this effect is an inherent property of the architecture or a consequence of the training data. Stylized ImageNet is of particular interest, as its authors specifically constructed the dataset to obtain models that use more global ("style") rather than local ("texture") features.

Finally, we propose two image preprocessing steps, one related to hue, the other to saturation, to augment a model's robustness with regards to alterations in these dimensions. To demonstrate the effectiveness of these preprocessors, we focus on ImageNet-1k and show that one can simply continue training a pretrained model using these additional preprocessors to achieve the desired effect; there is no need to train a model from scratch. All our code and models are available through our GitLab page [10].

The remainder of this paper is organized as follows: in Sect. 2 we explain the methodology used to test model robustness to hue and saturation changes, followed by a discussion of obtained results in Sect. 3. Section 4 deals with retraining pretrained models using additional preprocessing steps in order to increase model robustness to hue and saturation changes. The paper concludes with Sect. 5.

2 Exploring Color Robustness: Implementation

We perform experiments with a number of models that were trained by others on specific training sets. In our experiments, we test performance using a number of existing validation sets. Table 1 shows an overview of the model-training-validation combinations we consider. The ImageNet-1k train and validation sets consist of 1,281,167 and 50,000 images respectively. For Places365, the models were trained using 8,000,000 images, with the corresponding validation set containing 36,500 images. Our code is Python-based, using PyTorch[2] as deep learning framework and Pillow[3], often referred to as PIL, as image processing package. All ImageNet-1k models are standard PyTorch implementations. The SIN and Places365 models were obtained through their respective public Git repositories. EmoNet is officially released as a MatLab model, and was ported by one of the current authors to PyTorch[4].

Table 1. Overview of training and validation data per model. "$\langle ModelName \rangle$" is a placeholder for a valid architecture, "IN-1k" = ImageNet-1k, "SIN" = Stylized ImageNet, "train" = train data, "val" = validation data.

Model	Trained on	Validated on
AlexNet, VGG16, ResNet18/50, DenseNet161	IN-1k train	IN-1k val
$\langle ModelName \rangle$-SIN	SIN	IN-1k val
$\langle ModelName \rangle$-P365	Places365 train	Places365 val
EmoNet	IN-1k train + EmoNet	IN-1k val

2.1 Applying Hue Changes

For a given pre-trained model M and corresponding validation data V, we apply hue shifts with degrees $d \in [0, 10, 20, \ldots, 350]$ to obtain shifted data sets V_d. Note that $V = V_{d=0}$.

[2] https://pytorch.org/.

[3] https://pillow.readthedocs.io/en/stable/.

[4] This port is available at https://gitlab.com/EAVISE/lme/emonet.

To apply the hue shifts, we first load the images as PIL images, then transform them to tensors using PyTorch. These tensors, which encode RGB information, are then converted to HSV[5]. Following this, the H-dimension is shifted by the required amount of degrees, and the image converted back to RGB.

2.2 Applying Saturation Changes

To change the saturation level of an image, we use the `enhance(`g`)` method of the `PIL.ImageEnhance.Color` class, with $g \in [0, +\infty[$. Using color gain $g = 1$ returns the original image, $g = 0$ returns a black-and-white copy, values $0 < g < 1$ produce desaturated images and $g > 1$ saturates the image. Starting again from validation data V, we produce data sets V_g using the described approach with $g \in [0.00, 0.05, \ldots, 1.95, 2.00]$, where $V = V_{g=1}$. The upper limit value of 2 was chosen heuristically by visual inspection.

2.3 Assessing Model Robustness

For a given model M, we determine its reference predictions, defined as its predictions for $V = V_{d=0} = V_{g=1}$. We then let M process all other data sets $V_{d \neq 0}$ and $V_{g \neq 1}$, and check what percentage of predictions remain unchanged. For each data set, we also compute the accuracy and look at what percentage of originally correct and wrong predictions were left unchanged. In other words, this tells us whether the internal model representation of correctly classified images is more stable than that of wrongly classified images.

3 Exploring Color Robustness: Results

A graphical representation of the evolution of model performance with hue and saturation shifts for AlexNet-based models is depicted in Fig. 2. Due to space limitations, we do not include plots for the other models, but instead make those available on our GitLab page [10]. Just as the symmetricity of the hue shift plot can be explained by the hue shift being controlled by a 360-degree parameter, the non-symmetricity of the saturation shift plot follows from the g parameter being only lower bound, and non-symmetric around 1. Statistics for all models are shown in Table 2. Besides the familiar Top1 accuracy, this table also includes the following metrics:

- Equal predictions (*Equal*): for a given hue shift $d \neq 0$ or saturation shift $g \neq 1$, this represents the fraction of images for which the predicted label remains the same as the original prediction ($d = 0$, $g = 1$), regardless of the correctness of the original prediction.
- OverLap+ (*OL+*): the fraction of originally correctly classified images whose predicted label did not change.

[5] We use the code available at https://github.com/limacv/RGB_HSV_HSL for this.

- OverLap− ($OL-$): the fraction of originally wrongly classified images whose predicted label did not change.
- Original Position ($O.P.$): the position of the label predicted for the shifted image in the list $[l_0, l_1, \ldots, l_n]$ of labels ordered by their likelihood as predicted for the original image. That is, if O.P. $= 0$, the shift did not change the prediction, but if O.P. is, e.g., 2, this means that the label predicted for the shifted image was originally the third most likely label. The result tables show averages that were computed taking only non-zero values into account.

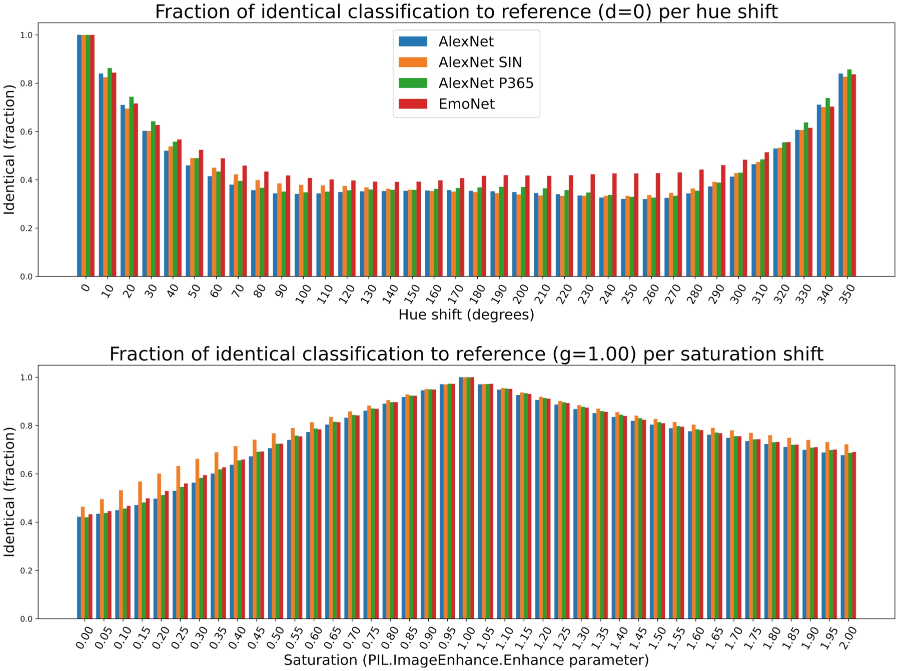

Fig. 2. Fraction of identically classified images compared to the reference prediction ($d = 0$, $g = 1$) for increasing hue and saturation shifts for AlexNet-based models.

For the Equal metric, we observe very similar results for the same architectures trained on different datasets. EmoNet does appear to perform slightly better than other AlexNet-based models wrt. larger hue shifts, but given that it only has 20 output nodes compared to 365 and 1000 for the other models, suggesting that larger perturbations are needed to switch output nodes, the overall similarities are remarkable. For saturation shifts, the differences are negligible. The slightly better AlexNet-SIN performance compared to AlexNet for saturation shifts is puzzling, given that both VGG16 and ResNet50 show the opposite behavior. Overall, the fact that SIN-trained models appear to be less robust wrt. both hue and saturation shifts than the ImageNet-1k models is intruiging, given that the aim of the SIN dataset is to create models that focus more on "global" than "local" features. Since hue and saturation shifts are global transformations, one would have expected the opposite. Our results confirm and expand on the findings of [3] that hue sensitivity is higher than saturation sensitivity[6], apparent from the much lower values for Equal d_{all} than for Equal g_{all}.

Turning to the OverLap metrics, it is noteworthy that images that are originally correctly classified consistently have a lesser probability of being misclassified after applying hue/saturation changes. This suggests that the internal model representations for these images are inherently more robust, although it is not clear at first sight why this is the case. The magnitude of the gaps between the OL+ and OL− metrics is striking. Even more so is the fact that, despite all models being less sensitive to saturation changes, the corresponding OL+/OL− gap lies considerably higher than for hue changes.

Finally, the O.P. results are in line with the previous results. As the number of output nodes diminishes, so does the O.P. Furthermore, for hue changes, the O.P. is higher than for saturation changes. For smaller perturbations ($|d| \leq 30$, $g \in \,]0.5, 1.5[_{\setminus\{1\}}$), the O.P. is markedly smaller than when considering d_{all} or g_{all}. As the size of the perturbation increases, so does the erraticness of the change in predicted label. This is specifically apparent in the very large gap in standard deviations between both regimes.

Concerning overall model performance, the top panel in Fig. 2 suggests that, for AlexNet and EmoNet, this more or less linearly decreases until it plateaus at around an 80° hue shift in either direction. Similar behavior can be observed for the other models, with the exception of ResNet50, for which the performance shows a slight bump around the 170°–180° region. In their paper, [3] report an average drop in performance of 31.6% over hue shifts, averaged over VGG16, VGG19 and AlexNet performance, with 42% for AlexNet alone. This matches our 41.5% for AlexNet[7]. In a non-reported experiment, we obtained 28.9% for VGG19, which combined with the 22.9% for VGG16 derivable from Table 2 amounts to a 30.9% average, closely matching their result. The slight differences can be explained by the useage of different pretrained models, namely CAFFE vs. PyTorch implementations. Turning to grayscale (corresponding to $g = 0$) vs. original images, they report average drops of 25% across all three networks, and 33% for AlexNet, compared to 25.5% and 40.2% for us respectively, 18.8% for VGG16. Although the average across networks matches, we can only speculate as to the larger implied individual differences indicated by the AlexNet mismatch.

[6] Note that [3] use "chroma" instead of "saturation", but given the similarity between both, our conclusion still stands.

[7] Divide "Top1 d_{all}" by "Top1 d_0, g_1" to compute this number.

Table 2. Pretrained model statistics wrt. hue and saturation shifts applied to input images. “Top1” = Top1 accuracy, “OL+/−” = overlap between predicted labels for $d_0 = 0$ vs. $d \neq 0$, and $g_1 = 1$ vs. $g \neq 1$, for originally correctly (+) and wrongly (−) predicted samples, “O.P.” = Original Position of differing winning prediction for $d \neq 0$ or $g \neq 1$, d_{all} and g_{all} refer to all non-default (d_0, g_1) degree and color gain values, $|d| \leq 30$ represents $d \in [-30, -20, -10, 10, 20, 30]$. Except for the “Top1 d_0, g_1” values, all normal case values represent averages and all superscript values represent standard deviations over the relevant parameter range.

	AlexNet	AlexNet SIN	AlexNet P365	EmoNet	VGG16	VGG16 SIN	ResNet18	ResNet18 P365	ResNet50	ResNet50 SIN	DenseNet161	DenseNet161 P365
Equal d_{all}	$.431^{.15}$	$.441^{.14}$	$.447^{.15}$	$.489^{.12}$	$.659^{.10}$	$.581^{.12}$	$.659^{.10}$	$.614^{.11}$	$.800^{.06}$	$.632^{.10}$	$.759^{.07}$	$.662^{.09}$
Equal $\|d\| \leq 30$	$.718^{.10}$	$.709^{.09}$	$.747^{.09}$	$.724^{.09}$	$.861^{.05}$	$.803^{.06}$	$.847^{.05}$	$.830^{.06}$	$.910^{.03}$	$.826^{.06}$	$.897^{.04}$	$.847^{.05}$
Equal g_{all}	$.746^{.15}$	$.786^{.13}$	$.756^{.15}$	$.758^{.15}$	$.883^{.08}$	$.857^{.09}$	$.874^{.09}$	$.842^{.10}$	$.950^{.03}$	$.885^{.08}$	$.946^{.05}$	$.863^{.09}$
Equal $g \in]0.5, 1.5[\setminus\{1\}$	$.863^{.07}$	$.883^{.06}$	$.872^{.07}$	$.870^{.07}$	$.943^{.03}$	$.924^{.04}$	$.937^{.03}$	$.919^{.04}$	$.972^{.01}$	$.940^{.03}$	$.977^{.01}$	$.929^{.04}$
Top1 d_0, g_1	.566	.400	-	-	.716	.522	.697	-	.803	.602	.771	-
Top1 d_{all}	$.331^{.09}$	$.263^{.06}$	-	-	$.552^{.07}$	$.409^{.05}$	$.548^{.06}$	-	$.713^{.04}$	$.488^{.05}$	$.661^{.05}$	-
OL+ d_{all}	$.528^{.15}$	$.579^{.14}$	-	-	$.732^{.10}$	$.713^{.10}$	$.743^{.09}$	-	$.857^{.05}$	$.752^{.09}$	$.823^{.06}$	-
OL− d_{all}	$.305^{.14}$	$.349^{.14}$	-	-	$.475^{.12}$	$.437^{.13}$	$.465^{.12}$	-	$.568^{.09}$	$.451^{.12}$	$.541^{.10}$	-
Top1 g_{all}	$.505^{.06}$	$.382^{.02}$	-	-	$.688^{.04}$	$.511^{.02}$	$.668^{.04}$	-	$.798^{.01}$	$.591^{.02}$	$.762^{.02}$	-
OL+ g_{all}	$.855^{.14}$	$.846^{.14}$	-	-	$.917^{.15}$	$.874^{.14}$	$.915^{.15}$	-	$.938^{.15}$	$.897^{.15}$	$.934^{.15}$	-
OL− g_{all}	$.253^{.17}$	$.345^{.15}$	-	-	$.379^{.16}$	$.361^{.11}$	$.359^{.14}$	-	$.451^{.09}$	$.362^{.10}$	$.399^{.10}$	-
Top1 $\|d\| \leq 30$	$.501^{.04}$	$.367^{.02}$	-	-	$.681^{.02}$	$.500^{.01}$	$.662^{.02}$	-	$.782^{.01}$	$.580^{.01}$	$.747^{.02}$	-
OL+ $\|d\| \leq 30$	$.829^{.08}$	$.844^{.06}$	-	-	$.920^{.04}$	$.904^{.04}$	$.914^{.04}$	-	$.951^{.02}$	$.917^{.04}$	$.944^{.03}$	-
OL− $\|d\| \leq 30$	$.575^{.12}$	$.620^{.11}$	-	-	$.713^{.09}$	$.693^{.09}$	$.692^{.09}$	-	$.742^{.07}$	$.688^{.09}$	$.740^{.08}$	-
Top1 $g \in]0.5, 1.5[\setminus\{1\}$	$.550^{.01}$	$.397^{.00}$	-	-	$.711^{.01}$	$.520^{.00}$	$.691^{.01}$	-	$.802^{.00}$	$.601^{.00}$	$.770^{.00}$	-
OL+ $g \in]0.5, 1.5[\setminus\{1\}$	$.883^{.02}$	$.863^{.02}$	-	-	$.943^{.01}$	$.894^{.01}$	$.941^{.01}$	-	$.962^{.00}$	$.919^{.01}$	$.956^{.00}$	-
OL− $g \in]0.5, 1.5[\setminus\{1\}$	$.197^{.04}$	$.310^{.04}$	-	-	$.337^{.04}$	$.344^{.03}$	$.319^{.03}$	-	$.444^{.02}$	$.346^{.02}$	$.381^{.01}$	-
O.P. d_{all}	58.5^{122}	40.6^{96}	29.7^{53}	3.7^{3}	28.0^{76}	26.5^{73}	25.3^{71}	13.0^{29}	31.6^{119}	24.5^{70}	19.1^{61}	9.9^{24}
O.P. $\|d\| \leq 30$	10.9^{37}	6.7^{23}	4.9^{13}	2.1^{2}	5.2^{18}	4.4^{16}	5.0^{18}	2.8^{6}	6.4^{45}	4.2^{14}	4.0^{15}	2.6^{6}
O.P. g_{all}	11.9^{40}	19.8^{55}	20.0^{44}	2.3^{2}	5.3^{20}	4.3^{17}	4.9^{18}	3.8^{10}	5.5^{23}	11.9^{36}	2.9^{10}	2.9^{8}
O.P. $g \in]0.5, 1.5[\setminus\{1\}$	2.5^{5}	18.9^{52}	19.4^{42}	1.4^{1}	1.6^{2}	1.5^{2}	1.6^{2}	1.5^{1}	5.5^{22}	11.9^{36}	1.2^{1}	1.4^{1}

4 Increasing Color Robustness by Adding Extra Preprocessing Steps

4.1 Training of Models

For this experiment, we focused on the AlexNet, VGG16 and ResNet18 ImageNet-1k models. To increase their robustness to changes in hue and saturation, we apply random hue and saturation changes to input images during the training phase, on top of the standard ImageNet-1k preprocessing steps. The image processing is done as explained in Sects.2.1 and 2.2. The difference is that this time, the magnitude of the change is chosen randomly whenever an image is loaded. The number of degrees of the hue shift is sampled from a normal distribution $\mathcal{N}(\mu = 0, \sigma = 30)$, while the gain factor for the saturation shift is sampled from a normal distribution $\mathcal{N}(\mu = 1, \sigma = 0.5)$. The choice for these particular distributions is heuristic. For hue changes, $\sigma = 30$ was chosen as this range coincides with a steep descent in model performance and comprises hue changes that, as illustrated in Fig. 1, are not too extreme. For saturation changes, given the reduced model sensitivity, we opted to have 2σ span the entire covered spectrum.

The hue and saturation changes are applied right before normalizing the image. Model validation is performed on the original validation set.

As a starting point, we take the pretrained PyTorch implementations of the aformentioned models, available through the `torchvision` library. We then continue training these models using the ImageNet-1k train data, CrossEntropyLoss, dropout = 0.25, Adam optimizer with `weight decay` $= 10^{-6}$, batch size 64 for VGG16 and 256 for Alexnet and ResNet18, and the learning update rule:

$$\mathrm{lr}_e = \frac{\mathrm{lr}_0}{\sqrt{(e//2)+1}}, \tag{1}$$

with lr_e the learning rate at epoch e and the initial learning rate $\mathrm{lr}_0 = 10^{-5}$. By virtue of the floor division (//), this means we update the learning rate once every 2 epochs. Training stops when either the best loss or the best weighted F1 score on the validation set lies 6 epochs behind the current epoch, with the model corresponding to this best epoch put forward as the final trained model.

Models were trained using hue (*+h*), and hue + saturation (*+hs*) preprocessing. Given the increased model sensitivity to hue changes, we opted not to train models using only saturation preprocessing. To check the effect of only retraining a CNN's classifier (*class.*; i.e., the final linear layers following the convolution layers) instead of the entire model, we also retrained the AlexNet classifier, consisting of the final 3 linear layers including the output layer, while keeping the convolution layers fixed.

4.2 Results

Metrics for our retrained models are depicted in Table 3. Plots depicting model performance are made available through out GitLab page [10]. Noteworthy is the fact that our retrained models retain the Top1 performance of the original models, but manifest clearly improved robustness to hue and saturation alterations. This means that separate sets of CNN filters achieve the same accuracy on the same dataset, but nonetheless show vastly different behavior when performing a specific transformation on the input images. Although the "AlexNet class." models already show a significant improvement in robustness compared to AlexNet, the fact that the full retrained models perform even better confirms the intuition that CNN filters are the crucial ingredient in obtaining robust models, rather than the linear classification layers. For all models, the additional preprocessing does not seem to alter the gap between OL+ and OL− for hue changes, i.e., they are both affected similarly, but additional saturation preprocessing has a clear positive effect for its corresponding gap. More striking is the large decrease in O.P., specifically for hue. For saturation, the effect is less pronounced[8], arguably in part because there is less room for improvement to begin with. Moreover, additional hue preprocessing tends to negatively influence O.P. for saturation changes, but using both hue and saturation preprocessing benefits the O.P. for both types of changes. All this suggests that these preprocessing steps contribute to creating more robust internal model representations.

5 Conclusion

This paper explores the prediction stability of the popular CNN architectures AlexNet, VGG16, ResNet18 and 50, and DenseNet161. We show that all models alter their predictions when input images have their hue shifted, with larger shifts increasing alteration frequency. Averaged over all hue shifts, relative model performance experiences a drop of 41.5%, 22.9%, 21.4%, 11.3% and 14.3% respectively for the aforementioned models, resulting in an average drop of 22.28% over all models; larger models show less sensitivity. The largest drops are observed within up to 30° shifts from reference, with performance stabilizing around the 80° mark. Moreover, models trained on ImageNet-1k, Stylized ImageNet and Places365 are compared, showing the training data has little to no effect on this issue. EmoNet, a model derived from AlexNet, is shown to inherit essentially the same behavior as its parent. Saturation shifts elicit similar but more restrained behavior, with an average performance drop of only 4.0% over all models. Importantly, for both hue and saturation alterations, the prediction for images originally correctly predicted tends to be more robust than for images originally wrongly predicted. We propose to include two additional preprocessing steps in the training process, namely random hue shifts and saturation changes, which, when used to retrain existing models, are shown to improve average prediction stability for hue shifts on ImageNet-1k with 19%, 13% and 12% for AlexNet, VGG16 and ResNet18 respectively. For saturation changes, 11%, 6% and 6% improvements are obtained, in the last two cases lifting stability up to 94% and 93%. Interestingly, these retrained models retain the original model's ImageNet-1k performance, leading to the question: How exactly can several sets

[8] We compare ⟨ModelName⟩ to ⟨ModelName⟩ +hs.

Table 3. Retrained model statistics wrt. hue and saturation shifts applied to input images. The original models are also included for easy comparison. "+h" = random hue shift added to train image preprocessing, "+hs" = random hue and saturation shift added to train image preprocessing, "class." means only the classifier was retrained instead of the entire model, "Top1" = Top1 accuracy, "OL+/−" = overlap between predicted labels for $d_0 = 0$ vs. $d \neq 0$, and $g_1 = 1$ vs. $g \neq 1$, for originally correctly (+) and wrongly (−) predicted samples, "O.P." = Original Position of differing winning prediction for $d \neq 0$ or $g \neq 1$, d_{all} and g_{all} refer to all non-default (d_0, g_1) degree and color gain values, $|d| \leq 30$ represents $d \in [-30, -20, -10, 10, 20, 30]$. Except for the "Top1 d_0, g_1" values, all normal case values represent averages and all superscript values represent standard deviations over the relevant parameter range.

	AlexNet	AlexNet class.+h	AlexNet class.+hs	Alexnet +h	AlexNet +hs	VGG16	VGG16 +h	VGG16 +hs	ResNet18	ResNet18 +h	ResNet18 +hs
Equal d_{all}	$.431^{.15}$	$.528^{.14}$	$.576^{.14}$	$.587^{.16}$	$.623^{.17}$	$.659^{.10}$	$.778^{.11}$	$.793^{.11}$	$.659^{.10}$	$.769^{.10}$	$.779^{.10}$
Equal $\|d\| \leq 30$	$.718^{.10}$	$.788^{.07}$	$.817^{.06}$	$.854^{.05}$	$.877^{.04}$	$.861^{.05}$	$.938^{.02}$	$.944^{.02}$	$.847^{.05}$	$.927^{.03}$	$.932^{.02}$
Equal g_{all}	$.746^{.15}$	$.751^{.15}$	$.820^{.11}$	$.748^{.15}$	$.857^{.09}$	$.883^{.08}$	$.880^{.08}$	$.942^{.04}$	$.874^{.09}$	$.873^{.08}$	$.931^{.05}$
Equal $g \in]0.5, 1.5[\setminus\{1\}$	$.863^{.07}$	$.865^{.07}$	$.902^{.05}$	$.861^{.07}$	$.926^{.04}$	$.943^{.03}$	$.938^{.03}$	$.972^{.01}$	$.937^{.03}$	$.934^{.04}$	$.967^{.02}$
Top1 d_0, g_1	.566	.565	.564	.564	.565	.716	.714	.714	.697	.703	.702
Top1 d_{all}	$.331^{.09}$	$.400^{.08}$	$.432^{.07}$	$.428^{.08}$	$.451^{.08}$	$.552^{.07}$	$.631^{.06}$	$.644^{.06}$	$.548^{.06}$	$.623^{.06}$	$.628^{.06}$
OL+ d_{all}	$.528^{.15}$	$.645^{.14}$	$.701^{.13}$	$.700^{.16}$	$.739^{.15}$	$.732^{.10}$	$.847^{.09}$	$.864^{.09}$	$.743^{.09}$	$.847^{.09}$	$.856^{.09}$
OL− d_{all}	$.305^{.14}$	$.377^{.14}$	$.414^{.15}$	$.441^{.17}$	$.473^{.19}$	$.475^{.12}$	$.605^{.15}$	$.617^{.16}$	$.465^{.12}$	$.585^{.14}$	$.599^{.15}$
Top1 g_{all}	$.505^{.06}$	$.508^{.06}$	$.538^{.03}$	$.506^{.06}$	$.548^{.02}$	$.688^{.04}$	$.685^{.03}$	$.709^{.01}$	$.668^{.04}$	$.674^{.03}$	$.693^{.01}$
OL+ g_{all}	$.855^{.14}$	$.861^{.14}$	$.884^{.14}$	$.860^{.14}$	$.887^{.14}$	$.917^{.15}$	$.919^{.15}$	$.928^{.15}$	$.915^{.15}$	$.914^{.15}$	$.927^{.15}$
OL− g_{all}	$.253^{.17}$	$.263^{.18}$	$.352^{.15}$	$.278^{.18}$	$.356^{.13}$	$.379^{.16}$	$.413^{.17}$	$.474^{.10}$	$.359^{.14}$	$.381^{.15}$	$.453^{.11}$
Top1 $\|d\| \leq 30$	$.501^{.04}$	$.534^{.02}$	$.545^{.01}$	$.550^{.01}$	$.556^{.00}$	$.681^{.02}$	$.709^{.00}$	$.711^{.00}$	$.662^{.02}$	$.697^{.00}$	$.697^{.00}$
OL+ $\|d\| \leq 30$	$.829^{.08}$	$.893^{.04}$	$.916^{.03}$	$.934^{.03}$	$.949^{.02}$	$.920^{.04}$	$.974^{.01}$	$.977^{.01}$	$.914^{.04}$	$.970^{.01}$	$.972^{.01}$
OL− $\|d\| \leq 30$	$.575^{.12}$	$.652^{.10}$	$.690^{.09}$	$.750^{.08}$	$.785^{.07}$	$.713^{.09}$	$.848^{.05}$	$.862^{.04}$	$.692^{.09}$	$.828^{.06}$	$.838^{.05}$
Top1 $g \in]0.5, 1.5[\setminus\{1\}$	$.550^{.01}$	$.550^{.01}$	$.558^{.01}$	$.548^{.01}$	$.561^{.00}$	$.711^{.01}$	$.708^{.01}$	$.713^{.00}$	$.691^{.01}$	$.696^{.01}$	$.700^{.00}$
OL+ $g \in]0.5, 1.5[\setminus\{1\}$	$.883^{.02}$	$.888^{.02}$	$.906^{.01}$	$.891^{.02}$	$.908^{.01}$	$.943^{.01}$	$.946^{.01}$	$.950^{.00}$	$.941^{.01}$	$.941^{.01}$	$.949^{.00}$
OL− $g \in]0.5, 1.5[\setminus\{1\}$	$.197^{.04}$	$.208^{.04}$	$.312^{.04}$	$.222^{.04}$	$.329^{.03}$	$.337^{.04}$	$.371^{.04}$	$.466^{.02}$	$.319^{.03}$	$.341^{.03}$	$.440^{.02}$
O.P. d_{all}	58.5^{122}	34.4^{84}	25.8^{68}	27.6^{72}	22.4^{61}	28.0^{76}	12.8^{41}	10.9^{36}	25.3^{71}	11.0^{36}	10.1^{34}
O.P. $\|d\| \leq 30$	10.9^{37}	4.3^{13}	3.1^{7}	2.4^{5}	1.9^{3}	5.2^{18}	1.7^{4}	1.5^{2}	5.0^{18}	1.6^{2}	1.6^{2}
O.P. g_{all}	11.9^{40}	38.6^{101}	10.6^{28}	36.0^{96}	9.6^{26}	5.3^{20}	5.0^{18}	4.7^{14}	4.9^{18}	9.0^{30}	2.2^{5}
O.P. $g \in]0.5, 1.5[\setminus\{1\}$	2.5^{5}	37.5^{97}	10.4^{27}	34.3^{91}	9.3^{25}	1.6^{2}	1.7^{2}	4.8^{14}	1.6^{2}	8.9^{30}	1.2^{1}

of convolution filters result in the same ImageNet-1k accuracy, yet show markedly different behavior when subjected to particular image transformations? We hope to address this question in future work.

References

1. Crawford, K., Paglen, T.: Excavating AI: the politics of images in machine learning training sets. https://excavating.ai/. Accessed 8 Mar 2023
2. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: ImageNet: a large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition, pp. 248–255. IEEE (2009)
3. Flachot, A., Gegenfurtner, K.R.: Color for object recognition: hue and chroma sensitivity in the deep features of convolutional neural networks. Vision. Res. **182**, 89–100 (2021). https://doi.org/10.1016/j.visres.2020.09.010
4. Geirhos, R., Rubisch, P., Michaelis, C., Bethge, M., Wichmann, F.A., Brendel, W.: ImageNet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness. https://doi.org/10.48550/ARXIV.1811.12231 (2018)
5. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 770–778 (2016). https://doi.org/10.1109/CVPR.2016.90
6. Hendrycks, D., Dietterich, T.G.: Benchmarking neural network robustness to common corruptions and perturbations. CoRR abs/1903.12261 (2019), https://arxiv.org/abs/1903.12261
7. Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q.: Densely connected convolutional networks. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2261–2269 (2017). https://doi.org/10.1109/CVPR.2017.243
8. Kragel, P.A., Reddan, M.C., LaBar, K.S., Wager, T.D.: Emotion schemas are embedded in the human visual system. Sci. Adv. **5**(7), eaaw4358 (2019). https://doi.org/10.1126/sciadv.aaw4358
9. Krizhevsky, A., Sutskever, I., Hinton, G.E.: ImageNet classification with deep convolutional neural networks. Commun. ACM **60**(6), 84–90 (2017). https://doi.org/10.1145/3065386
10. Mertens, L.: GitLab repository containing the code and additional material for this paper. https://gitlab.com/EAVISE/lme/nncolorstabilityanalysis-paper
11. Rafegas, I., Vanrell, M.: Color encoding in biologically-inspired convolutional neural networks. Vision Res. **151**, 7–17 (2018). https://doi.org/10.1016/j.visres.2018.03.010
12. Russakovsky, O., et al.: ImageNet large scale visual recognition challenge. Int. J. Comput. Vision **115**(3), 211–252 (2015). https://doi.org/10.1007/s11263-015-0816-y
13. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: Bengio, Y., LeCun, Y. (eds.) 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7–9, 2015, Conference Track Proceedings (2015)
14. Steed, R., Caliskan, A.: Image representations learned with unsupervised pre-training contain human-like biases. In: Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency, pp. 701–713. FAccT 2021, Association for Computing Machinery, New York, NY, USA (2021). https://doi.org/10.1145/3442188.3445932

15. Szegedy, C., et al.: Intriguing properties of neural networks. In: 2nd International Conference on Learning Representations, Conference date: 14–04-2014 Through 16–04-2014. ICLR (2014)
16. Yang, K., Qinami, K., Fei-Fei, L., Deng, J., Russakovsky, O.: Towards fairer datasets: filtering and balancing the distribution of the people subtree in the ImageNet hierarchy. In: Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency, pp. 547–558. FAT* 2020, Association for Computing Machinery, New York, NY, USA (2020). https://doi.org/10.1145/3351095.3375709
17. Zhou, B., Lapedriza, A., Khosla, A., Oliva, A., Torralba, A.: Places: a 10 million image database for scene recognition. IEEE Trans. Pattern Anal. Mach. Intell. **40**, 1452–1464 (2017)