Color Equivariant Convolutional Networks

Attila Lengyel **Ombretta Strafforello** **Robert-Jan Bruintjes**
Alexander Gielisse **Jan van Gemert**
Computer Vision Lab
Delft University of Technology
Delft, The Netherlands

Abstract

Color is a crucial visual cue readily exploited by Convolutional Neural Networks (CNNs) for object recognition. However, CNNs struggle if there is data imbalance between color variations introduced by accidental recording conditions. Color invariance addresses this issue but does so at the cost of removing all color information, which sacrifices discriminative power. In this paper, we propose Color Equivariant Convolutions (CEConvs), a novel deep learning building block that enables shape feature sharing across the color spectrum while retaining important color information. We extend the notion of equivariance from geometric to photometric transformations by incorporating parameter sharing over hue-shifts in a neural network. We demonstrate the benefits of CEConvs in terms of downstream performance to various tasks and improved robustness to color changes, including train-test distribution shifts. Our approach can be seamlessly integrated into existing architectures, such as ResNets, and offers a promising solution for addressing color-based domain shifts in CNNs.

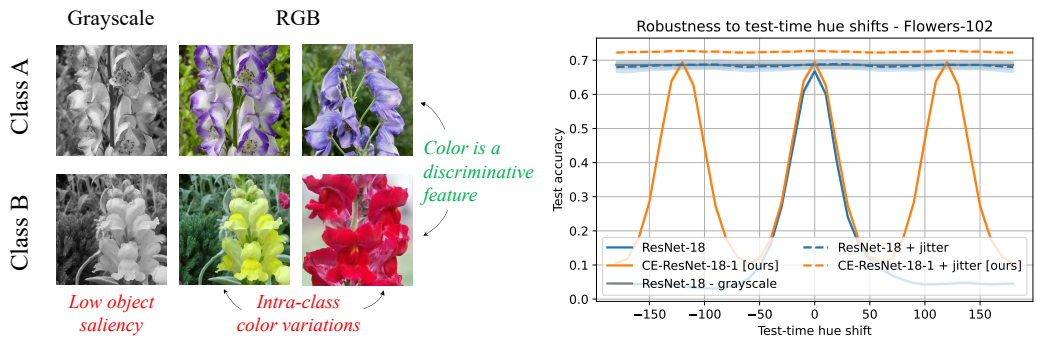
1 Introduction

Color is a powerful cue for visual object recognition. Trichromatic color vision in primates may have developed to aid the detection of ripe fruits against a background of green foliage [38, 45]. The benefit of color vision here is two-fold: not only does color information improve foreground-background segmentation by rendering foreground objects more salient, color also allows diagnostics, e.g. identifying the type (orange) and ripeness (green) where color is an intrinsic property facilitating recognition [3], as illustrated in Fig. 1a. Convolutional neural networks (CNNs) too exploit color information by learning color selective features that respond differently based on the presence or absence of a particular color in the input [42].

Unwanted color variations, however, can be introduced by accidental scene recording conditions such as illumination changes [29, 48], or by low color-diagnostic objects occurring in a variety of colors, making color no longer a discriminative feature but rather an undesired source of variation in the data. Given a sufficiently large training set that encompasses all possible color variations, a CNN learns to become robust by learning color invariant and equivariant features from the available data [36, 37]. However, due to the long tail of the real world it is almost impossible to collect balanced training data for all scenarios. This naturally leads to color distribution shifts between training and test time, and an imbalance in the training data where less frequently occurring colors are underrepresented. As CNNs often fail to generalize to out-of-distribution test samples, this can have significant impact on many real-world applications, e.g. a model trained mostly on red cars may struggle to recognize the exact same car in blue.

Color invariance addresses this issue through features that are by design invariant to color changes and therefore generalize better under appearance variations [14, 17]. However, color invariance comes at the loss of discriminative power as valuable color information is removed from the model's

37th Conference on Neural Information Processing Systems (NeurIPS 2023).



(a) (b)

Figure 1: Color plays a significant role in object recognition. (a) The absence of color makes flowers less distinct from their background and thus harder to classify. The characteristic purple-blue color of the Monkshood (Class A) enables a clear distinction from the Snapdragon (Class B) [35]. On the other hand, relying too much on colors might negatively impact recognition to color variations within the same flower class. (b) Image classification performance on the Flower-102 dataset [35] under a gradual variation of the image hue. Test-time hue shifts degrade the performance of CNNs (ResNet-18) drastically. Grayscale images and color augmentations result in invariance to hue variations, but fail to capture all the characteristic color features of flowers. Our color equivariant network (CE-ResNet-18-1) enables feature sharing across the color spectrum, which helps generalise to underrepresented colors in the dataset, while preserving discriminative color information, improving classification for unbalanced color variations.

internal feature representation [18]. We therefore propose to equip models with the less restrictive *color equivariance* property, where features are explicitly shared across different colors through a hue transformation on the learned filters. This allows the model to generalize across different colors, while at the same time also retaining important color information in the feature representation.

An RGB pixel can be decomposed into an orthogonal representation by the well-known hue-saturation-value (HSV) model, where hue represents the chromaticity of a color. In this work we extend the notion of equivariance from geometric to photometric transformations by hard-wiring parameter sharing over hue-shifts in a neural network. More specifically, we build upon the seminal work of Group Equivariant Convolutions [7] (GConvs), which implements equivariance to translations, flips and rotations of multiples of 90 degrees, and formulates equivariance using the mathematical framework of symmetry groups. We introduce Color Equivariant Convolutions (CEConvs) as a novel deep learning building block, which implements equivariance to the H_n symmetry group of discrete hue rotations. CEConvs share parameters across hue-transformed filters in the input layer and store color information in hue-equivariant feature maps.

CEConv feature maps contain an additional dimension compared to regular CNNs, and as a result, require larger filters and thus more parameters for the same number of channels. To evaluate equivariant architectures, it is common practice to reduce the width of the network to match the parameter count of the baseline model. However, this approach introduces a trade-off between equivariance and model capacity, where particularly in deeper layers the quadratic increase in parameter count of CEConv layers makes equivariance computationally expensive. We therefore investigate hybrid architectures, where early color invariance is introduced by pooling over the color dimension of the feature maps. Note that early color invariance is maintained throughout the rest of the network, despite the use of regular convolutional layers after the pooling operation. Limiting color equivariant filters to the early layers is in line with the findings that early layers tend to benefit the most from equivariance [5] and learn more color selective filters [37, 42].

We rigorously validate the properties of CEConvs empirically through precisely controlled synthetic experiments, and evaluate the performance of color invariant and equivariant ResNets on various more realistic classification benchmarks. Moreover, we investigate the combined effects of color equivariance and color augmentations. Our experiments show that CEConvs perform on par or better

than regular convolutions, while at the same time significantly improving the robustness to test-time color shifts, and is complementary to color augmentations.

The main contributions of this paper can be summarized as follows:

- We show that convolutional neural networks benefit from using color information, and at the same time are not robust to color-based domain shifts.
- We introduce Color Equivariant Convolutions (CEConvs), a novel deep learning building block that allows feature sharing between colors and can be readily integrated into existing architectures such as ResNets.
- We demonstrate that CEConvs improve robustness to train-test color shifts in the input.

All code and experiments are made publicly available on `https://github.com/Attila94/CEConv`.

2 Related work

Equivariant architectures Translation equivariance is a key property of convolutional neural networks (CNNs) [23, 28]: shifting the input to a convolution layer results in an equally shifted output feature map. This allows CNNs to share filter parameters over spatial locations, which improves both parameter and data efficiency as the model can generalize to new locations not covered by the training set. A variety of methods have extended equivariance in CNNs to other geometric transformations [44], including the seminal Group Equivariant Convolutions [7] for rotations and flips, and other works concerning rotations [2, 30, 52], scaling [50, 53] and arbitrary Lie groups [32]. Yet to date, equivariance to photometric transformations has remained largely unexplored. Offset equivariant networks [9] constrain the trainable parameters such that an additive bias to the RGB input channels results in an equal bias in the output logits. By applying a log transformation to the input the network becomes equivariant to global illumination changes according to the Von Kries model [13]. In this work we explore an alternative approach to photometric equivariance inspired by the seminal Group Equivariant Convolution [7] framework.

Color in CNNs Recent research has investigated the internal representation of color in Convolutional Neural Networks (CNNs), challenging the traditional view of CNNs as black boxes. For example, [41, 42] introduces the Neuron Feature visualization technique and characterizes neurons in trained CNNs based on their color selectivity, assessing whether a neuron activates in response to the presence of color in the input. The findings indicate that networks learn highly color-selective neurons across all layers, emphasizing the significance of color as a crucial visual cue. Additionally, [43] classifies neurons based on their class selectivity and observes that early layers contain more class-agnostic neurons, while later layers exhibited high class selectivity. A similar study has been performed in [12], further supporting these findings. [36, 37] investigate learned symmetries in an InceptionV1 model trained on ImageNet [10] and discover filters that demonstrated equivariance to rotations, scale, hue shifts, and combinations thereof. These results motivate color equivariance as a prior for CNNs, especially in the first layers. Moreover, in this study, we will employ the metrics introduced by [42] to provide an explanation for several of our own findings.

Color priors in deep learning Color is an important visual discriminator [15, 19, 51]. In classical computer vision, color invariants are used to extract features from an RGB image that are more consistent under illumination changes [14, 17, 18]. Recent studies have explored using color invariants as a preprocessing step to deep neural networks [1, 33] or incorporating them directly into the architecture itself [29], leading to improved robustness against time-of-day domain shifts and other illumination-based variations in the input. Capsule networks [22, 47], which use groups of neurons to represent object properties such as pose and appearance, have shown encouraging results in image colorization tasks [39]. Quaternion networks [16, 54] represent RGB color values using quaternion notation, and employ quaternion convolutional layers resulting in moderate improvements in image classification and inpainting tasks. Building upon these advancements, we contribute to the ongoing research on integrating color priors within deep neural architectures.

3 Color equivariant convolutions

3.1 Group Equivariant Convolutions

A CNN layer Φ is equivariant to a symmetry group G if for all transformations $g \in G$ on the input x the resulting feature mapping $\Phi(x)$ transforms similarly, i.e., first doing a transformation and then the mapping is similar to first doing the mapping and then the transformation. Formally, equivariance is defined as

$$\Phi(T_g x) = T'_g \Phi(x), \quad \forall g \in G, \tag{1}$$

where T_g and T'_g are the transformation operators of group action g on the input and feature space, respectively. Note that T_g and T'_g can be identical, as is the case for translation equivariance where shifting the input results in an equally shifted feature map, but do not necessarily need to be. A special case of equivariance is invariance, where T'_g is the identity mapping and the input transformation leaves the feature map unchanged:

$$\Phi(T_g x) = \Phi(x), \quad \forall g \in G. \tag{2}$$

We use the definition from [7] to denote the i-th output channel of a standard convolutional layer l in terms of the correlation operation $(\star)$ between a set of feature maps f and C^{l+1} filters ψ:

$$[f \star \psi^i](x) = \sum_{y \in \mathbb{Z}^2} \sum_{c=1}^{C^l} f_c(y) \psi_c^i(y - x). \tag{3}$$

Here $f : \mathbb{Z}^2 \to \mathbb{R}^{C^l}$ and $\psi^i : \mathbb{Z}^2 \to \mathbb{R}^{C^l}$ are functions that map pixel locations x to a C^l-dimensional vector. This definition can be extended to groups by replacing the translation x by a group action g:

$$[f \star \psi^i](g) = \sum_{y \in \mathbb{Z}^2} \sum_{c}^{C^l} f_c(y) \psi_c^i(g^{-1} y) \tag{4}$$

As the resulting feature map $f \star \psi^i$ is a function on G rather than $\mathbb{Z}^2$, the inputs and filters of all hidden layers should also be defined on G:

$$[f \star \psi^i](g) = \sum_{h \in G} \sum_{c}^{C^l} f_c(h) \psi_c^i(g^{-1} h) \tag{5}$$

Invariance to a subgroup can be achieved by applying a pooling operation over the corresponding cosets. For a more detailed introduction to group equivariant convolutions, please refer to [4, 7].

3.2 Color Equivariance

We define color equivariance as equivariance to hue shifts. The HSV color space encodes hue by an angular scalar value, and a hue shift is performed as a simple additive offset followed by a modulo operator. When projecting the HSV representation into three-dimensional RGB space, the same hue shift becomes a rotation along the $[1, 1, 1]$ diagonal vector.

We formulate hue equivariance in the framework of group theory by defining the group H_n of multiples of $360/n$-degree rotations about the $[1, 1, 1]$ diagonal vector in $\mathbb{R}^3$ space. H_n is a subgroup of the $SO(3)$ group of all rotations about the origin of three-dimensional Euclidean space. We can parameterize H in terms of integers k, n as

$$H_n(k) = \begin{bmatrix} \cos(\frac{2k\pi}{n}) + a & a - b & a + b \\ a + b & \cos(\frac{2k\pi}{n}) + a & a - b \\ a - b & a + b & \cos(\frac{2k\pi}{n}) + a \end{bmatrix} \tag{6}$$

with n the total number of discrete rotations in the group, k the rotation, $a = \frac{1}{3} - \frac{1}{3}\cos(\frac{2k\pi}{n})$ and $b = \sqrt{\frac{1}{3}} * \sin(\frac{2k\pi}{n})$. The group operation is matrix multiplication which acts on the continuous $\mathbb{R}^3$ space of RGB pixel values. The derivation of H_n is provided in Appendix A.

Color Equivariant Convolution (CEConv) Let us define the group $G = \mathbb{Z}^2 \times H_n$, which is a direct product of the $\mathbb{Z}^2$ group of discrete 2D translations and the H_n group of discrete hue shifts. We can then define the Color Equivariant Convolution (CEConv) in the input layer as:

$$[f \star \psi^i](x, k) = \sum_{y \in \mathbb{Z}^2} \sum_{c=1}^{C^l} f_c(y) \cdot H_n(k) \psi_c^i(y - x). \tag{7}$$

We furthermore introduce the operator $\mathcal{L}_g = \mathcal{L}_{(t,m)}$ including translation t and hue shift m acting on input f defined on the plane $\mathbb{Z}^2$:

$$[\mathcal{L}_g f](x) = [\mathcal{L}_{(t,m)} f](x) = H_n(m) f(x - t) \tag{8}$$

Since H_n is an orthogonal matrix, the dot product between a hue shifted input $H_n f$ and a filter ψ is equal to the dot product between the original input f and the inverse hue shifted filter $H_n^{-1}\psi$:

$$H_n f \cdot \psi = (H_n f)^T \psi = f^T H_n^T \psi = f \cdot H_n^T \psi = f \cdot H_n^{-1} \psi. \tag{9}$$

Then the equivariance of the CEConv layer can be derived as follows (using $C^l = 1$ for brevity):

$$\begin{aligned}
[[\mathcal{L}_{(t,m)} f] \star \psi^i](x, k) &= \sum_{y \in \mathbb{Z}^2} H_n(m) f(y - t) \cdot H_n(k) \psi^i(y - x) \\
&= \sum_{y \in \mathbb{Z}^2} f(y) \cdot H_n(m)^{-1} H_n(k) \psi^i(y - (x - t)) \\
&= \sum_{y \in \mathbb{Z}^2} f(y) \cdot H_n(k - m) \psi^i(y - (x - t)) \\
&= [f \star \psi^i](x - t, k - m) \\
&= [\mathcal{L'}_{(t,m)} [f \star \psi^i]](x, k)
\end{aligned} \tag{10}$$

Since input f and feature map $[f \star \psi]$ are functions on $\mathbb{Z}^2$ and G, respectively, $\mathcal{L}_{(t,k)}$ and $\mathcal{L'}_{(t,k)}$ represent two equivalent operators acting on their respective groups. For all subsequent hidden layers the input f and filters ψ^i are functions on G parameterized by x, k, and the hidden layer for CEConv is defined as:

$$[f \star \psi^i](x, k) = \sum_{y \in \mathbb{Z}^2} \sum_{r=1}^{n} \sum_{c=1}^{C^l} f_c(y, r) \cdot \psi_c^i(y - x, (r - k)\%n), \tag{11}$$

where n is the number of discrete rotations in the group and $\%$ is the modulo operator. In practice, applying a rotation to RGB pixels will cause some pixel values to fall outside of the RGB cube, which will then have to be reprojected within the cube. Due to this discrepancy, Eq. (9) only holds approximately, though in practice this has only limited consequences, as we empirical show in Appendix D.

3.3 Implementation

Tensor operations We implement CEConv similarly to GConv [7]. GConv represents the pose associated with the added spatial rotation group by extending the feature map tensor X with an extra dimension G^l to size $[C^l, G^l, H, W]$, denoting the number of channels, transformations that leave the origin invariant, and height and width of the feature map at layer l, respectively (batch dimension omitted). Similarly, a GConv filter $\tilde{F}$ with spatial extent k is of size $[C^{l+1}, G^{l+1}, C^l, G^l, k, k]$. The GConv is then defined in terms of tensor multiplication operations as:

$$X^{l+1}_{c',g',:,:} = \sum_{c}^{C^l} \sum_{g}^{G^l} \tilde{F}^l_{c',g',c,g,:,:} \star X^l_{c,g,:,:}, \tag{12}$$

where $(:)$ denotes tensor slices. Note that in the implementation, a GConv filter F only contains $[C^{l+1}, C^l, G^l, k, k]$ unique parameters - the extra G^{l+1} dimension is made up of transformed copies of F.

As the RGB input to the network is defined on $\mathbb{Z}^2$, we have $G^1 = 1$ and $\tilde{F}$ has size $[C^{l+1}, G^{l+1}, 3, 1, k, k]$. The transformed copies in G^{l+1} are computed by applying the rotation matrix from Eq. (6):

$$\tilde{F}^1_{c',g',:,1,u,v} = H_n(g')F^1_{c',:,1,u,v}. \tag{13}$$

In the hidden layers $\tilde{F}$ contains cyclically permuted copies of F:

$$\tilde{F}^l_{c',g',c,g,u,v} = F^l_{c',c,(g+g')\%n,u,v}. \tag{14}$$

Furthermore, to explicitly share the channel-wise spatial kernel over G^l [30], filter F is decomposed into a spatial component S and a pointwise component P as follows:

$$F^l_{c',c,g,u,v} = S_{c',c,1,u,v} \cdot P_{c',g',c,g,1,1} \tag{15}$$

F is precomputed in each forward step prior to the convolution operation in Eq. (12).

Input normalization is performed using a single value for the mean and standard deviations rather than per channel, as is commonly done for standard CNNs. Channel-wise means and standard deviations break the equivariance property of CECNN as a hue shift could no longer be defined as a rotation around the $[1, 1, 1]$ diagonal. Experiments have shown that using a single value for all channels instead of channel-wise normalization has no effect on the performance.

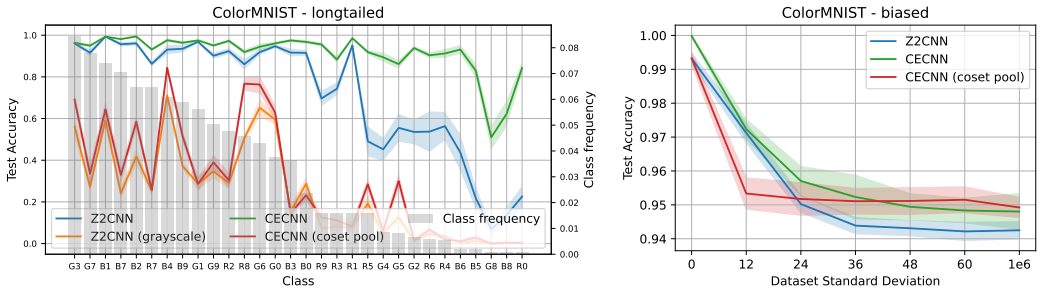
Compute efficiency CEConvs create a factor $|H_n|$ more feature maps in each layer. Due to the decomposition in Eq. (15), the number of multiply-accumulate (MAC) operations increase by only a factor $\frac{|H_n|^2}{k^2} + |H_n|$, and the number of parameters by a factor $\frac{|H_n|}{k^2} + 1$. See Appendix C.3 for an overview of parameter counts and MAC operations.

4 Experiments

4.1 When is color equivariance useful?

Color equivariant convolutions share shape information across different colors while preserving color information in the group dimension. To demonstrate when this property is useful we perform two controlled toy experiments on variations of the MNIST [11] dataset. We use the Z2CNN architecture from [7], and create a color equivariant version of the network called CECNN by replacing all convolutional layers by CEConvs with three rotations of 120°. The number of channels in CECNN is scaled such as to keep the number of parameters approximately equal to the Z2CNN. We also create a color invariant CECNN by applying coset max-pooling after the final CEConv layer, and a color invariant Z2CNN by converting the inputs to grayscale. All experiments are performed using the Adam [24] optimizer with a learning rate of 0.001 and the OneCycle learning rate scheduler. No data augmentations are used. We report the average performance over ten runs with different random initializations.

Color imbalance is simulated by *long-tailed ColorMNIST*, a 30-class classification problem where digits occur in three colors on a gray background, and need to be classified by both number (0-9) and color (red, green, blue). The number of samples per class is drawn from a power law distribution resulting in a long-tailed class imbalance. Sharing shape information across colors is beneficial as a certain digit may occur more frequently in one color than in another. The train set contains a total of 1,514 training samples and the test set is uniformly distributed with 250 samples per class. The training set is visualized in Appendix B.1. We train all four architectures on the dataset for 1000 epochs using the standard cross-entropy loss. The train set distribution and per-class test accuracies for all models are shown in Fig. 2a. With an average accuracy of $91.35 \pm 0.40\%$ the CECNN performs significantly better than the CNN with $71.59 \pm 0.61\%$. The performance increase is most significant for the classes with a low sample size, indicating that CEConvs are indeed more efficient in sharing shape information across different colors. The color invariant Z2CNN and CECNN networks, with an average accuracy of $24.19 \pm 0.53\%$ and $29.43 \pm 0.46\%$, respectively, are unable to discriminate between colors. CECNN with coset pooling is better able to discriminate between foreground and background and therefore performs slightly better. We repeated the experiment with a weighted loss and observed no significantly different results. We have also experimented with adding color jitter augmentations, which makes solving the classification problem prohibitive, as color is required. See Appendix B.2 for both detailed results on both experiments.



(a) (b)

Figure 2: Color equivariant convolutions efficiently share shape information across different colors. CECNN outperforms a vanilla network in both a long-tailed class imbalance setting (a), where MNIST digits are to be classified based on both shape and color, and a color biased setting (b), where the color of each class c is sampled according to $\theta_d \sim \mathcal{N}(\theta_c, \sigma)$.

Color variations are simulated by *biased ColorMNIST*, a 10-class classification problem where each class c has its own characteristic hue θ_c defined in degrees, distributed uniformly on the hue circle. The exact color of each digit x is sampled according to $\theta_x \sim \mathcal{N}(\theta_c, \sigma)$. We generate multiple datasets by varying σ between 0 and 10^6, where $\sigma = 0$ results in a completely deterministic color for each class and $\sigma = 10^6$ in an approximately uniform distribution for θ_x. For small σ, color is thus highly informative of the class, whereas for large σ the classification needs to be performed based on shape. The dataset is visualized in Appendix B.1. We train all models on the train set of 1.000 samples for 1500 epochs and evaluate on the test set of 10.000 samples. The test accuracies for different σ are shown in Fig. 2b. CECNN outperforms Z2CNN across all standard deviations, indicating CEConvs allow for a more efficient internal color representation. The color invariant CECNN network outperforms the equivariant CECNN model from $\sigma \geq 48$. Above this value color is no longer informative for the classification task and merely acts as noise unnecessarily consuming model capacity, which is effectively filtered out by the color invariant networks. The results of the grayscale Z2CNN are omitted as they are significantly worse, ranging between 89.89% ($\sigma = 0$) and 79.94 ($\sigma = 10^6$). Interestingly, CECNN with coset pooling outperforms the grayscale Z2CNN. This is due to the fact that a CECNN with coset pooling is still able to distinguish between small color changes and therefore can partially exploit color information. Networks trained with color jitter are unable to exploit color information for low σ; see Appendix B.2 for detailed results.

4.2 Image classification

Setup We evaluate our method for robustness to color variations on several natural image classification datasets, including CIFAR-10 and CIFAR-100 [27], Flowers-102 [35], STL-10 [6], Oxford-IIIT Pet [40], Caltech-101 [31], Stanford Cars [26] and ImageNet [10]. We train a baseline and color equivariant (CE-)ResNet [20] with 3 rotations and evaluate on a range of test sets where we gradually apply a hue shift between -180° and 180°. For high-resolution datasets (all except CIFAR) we train a ResNet-18 architecture and use default ImageNet data augmentations: we scale to 256 pixels, random crop to 224 pixels and apply random horizontal flips. For the CIFAR datasets we use the ResNet-44 architecture and augmentations from [7], including random horizontal flips and translations of up to 4 pixels. We train models both with and without color jitter augmentation to separately evaluate the effect of equivariance and augmentation. The CE-ResNets are downscaled in width to match the parameter count of the baseline ResNets. We have also included AugMix [21] and CIConv [29] as baselines for comparison. Training is performed for 200 epochs using the Adam [25] optimizer with a learning rate of 0.001 and the OneCycle learning rate scheduler. All our experiments use PyTorch and run on a single NVIDIA A40 GPU.

Hybrid networks In our toy experiments we enforce color equivariance throughout the network. For real world datasets however, we anticipate that the later layers of a CNN may not benefit from enforcing parameter sharing between colors, if the classes of the dataset are determined by color

specific features. We therefore evaluate hybrid versions of our color equivariance networks, denoted by an integer suffix for the number of ResNet stages, out of a possible four, that use CEConvs.

Original test set	**Caltech**	**C-10**	**C-100**	**Flowers**	**Ox-Pet**	**Cars**	**STL10**	**ImageNet**
Baseline	71.61	93.69	71.28	66.79	69.87	76.54	83.80	69.71
CIConv-W	**72.85**	75.26	38.81	**68.71**	61.53	**79.52**	80.71	65.81
CEConv	70.16	93.71	71.37	68.18	70.24	76.22	84.24	66.85
CEConv-2	71.50	**93.94**	**72.20**	68.38	**70.34**	77.06	**84.50**	**70.02**
Baseline + jitter	73.93	93.03	69.23	68.75	72.71	80.59	83.91	69.37
CIConv-W + jitter	**74.38**	77.49	42.27	**75.05**	64.23	**81.56**	81.88	65.95
CEConv + jitter	73.58	93.51	71.12	74.17	**73.29**	79.79	84.16	65.57
CEConv-2 + jitter	72.61	**93.86**	**71.35**	71.72	72.80	80.32	**84.46**	**69.42**
Baseline + AugMix	**71.92**	94.13	**72.64**	75.49	**76.02**	**82.32**	84.99	-
CEConv + AugMix	70.74	**94.22**	72.48	**78.10**	75.90	80.81	**85.46**	-
Hue-shifted test set								
Baseline	51.14	85.26	47.01	13.41	37.56	55.59	67.60	54.72
CIConv-W	**71.92**	74.88	37.09	**59.03**	**60.54**	**78.71**	**79.92**	**64.62**
CEConv	62.17	90.90	59.04	33.33	54.02	67.16	78.25	56.90
CEConv-2	64.51	**91.43**	**62.11**	33.32	51.14	68.17	77.80	62.26
Baseline + jitter	73.61	92.91	69.12	68.44	72.30	80.65	83.71	67.10
CIConv-W + jitter	**74.40**	77.28	42.30	**75.66**	63.93	**81.44**	81.54	65.03
CEConv + jitter	73.57	93.39	71.06	73.86	**72.94**	79.79	84.02	64.52
CEConv-2 + jitter	73.03	**93.80**	**71.33**	71.44	72.58	80.28	**84.31**	**68.74**
Baseline + AugMix	51.82	88.03	51.39	15.99	48.04	68.69	72.19	-
CEConv + AugMix	**62.29**	**91.68**	**60.75**	**41.43**	**62.27**	**73.59**	**80.17**	-

Table 1: Classification accuracy in % of vanilla vs. color equivariant (CE-)ResNets, evaluated both on the original and hue-shifted test sets. Color equivariant CNNs perform on par with vanilla CNNs on the original test sets, but are significantly more robust to test-time hue shifts.

Results We report both the performance on the original test set, as well as the average accuracy over all hue shifts in Table 1. For brevity we only show the fully equivariant and hybrid-2 networks, a complete overview of the performances of all hybrid network configurations and error standard deviations can be found in Appendix C.1. Between the full color equivariant and hybrid versions of our CE-ResNets, at least one variant outperforms vanilla ResNets on most datasets on the original test set. On most datasets the one- or two-stage hybrid versions are the optimal CE-ResNets, providing a good trade-off between color equivariance and leaving the network free to learn color specific features in later layers. CE-ResNets are also significantly more robust to test-time hue shifts, especially when trained without color jitter augmentation. Training the CE-ResNets with color jitter further improves robustness, indicating that train-time augmentations complement the already hard-coded inductive biases in the network. We show the detailed performance on Flowers-102 for all test-time hue shifts in Fig. 1b. The accuracy of the vanilla CNN quickly drops as a hue shift is applied, whereas the CE-CNN performance peaks at -120°, 0°and 120°. Applying train-time color jitter improves the CNN's robustness to the level of a CNN with grayscale inputs. The CE-CNN with color jitter outperforms all models for all hue shifts. Plots for other datasets are provided in Appendix C.2.

Color selectivity To explore what affects the success of color equivariance, we investigate the *color selectivity* of a subset of the studied datasets. We use the color selectivity measure from [42] and average across all neurons in the baseline model trained on each dataset. Fig. 3 shows that color selective datasets benefit from using color equivariance up to late stages, whereas less color selective datasets do not.

Feature representations of color equivariant CNNs We use the Neuron Feature [42] (NF) visualization method to investigate the internal feature representation of the CE-ResNet. NF computes a weighted average of the N highest activation input patches for each filter at a certain layer, as such representing the input patch that a specific neuron fires on. Fig. 4 shows the NF ($N = 50$) and top-3 input patches for filters at the final layers of stages 1-4 of a CE-ResNet18 trained on Flowers-102.

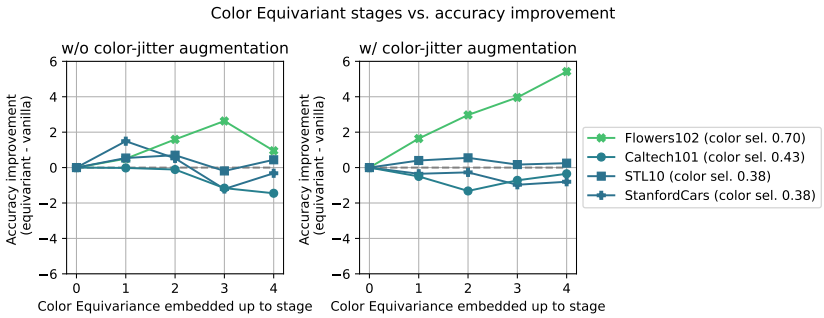


Figure 3: Color selective datasets benefit from using color equivariance up to late stages, whereas less color selective datasets do not. We compute average color selectivity [42] of all neurons in the baseline CNN trained on each dataset, and plot the accuracy improvement of using color equivariance in hybrid and full models, coloring each graphed dataset for color selectivity.

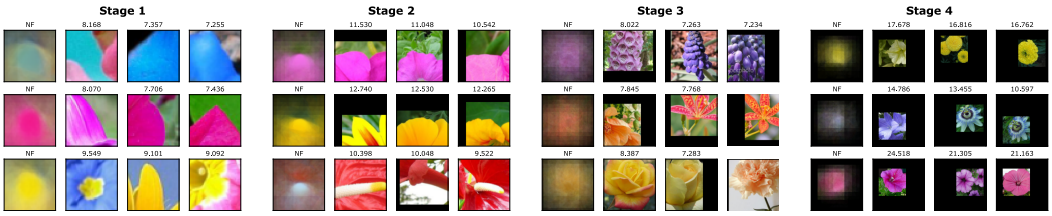


Figure 4: Neuron Feature [42] (NF) visualization with top-3 patches at different stages of a CE-ResNet18 trained on Flowers-102. Rows represent different rotations of the same filter. As expected, each row of a NF activates on the same shape in a different color.

Different rows represent different rotations of the same filter. As expected, each row of a NF activates on the same shape in a different color, demonstrating the color sharing capabilities of CEConvs. More detailed NF visualization are provided in Appendix C.4.

Ablation studies We perform ablations to investigate the effect of the number of rotations, the use of group coset pooling, and the strength of train-time color jitter augmentations. In short, we find that a) increasing the number of hue rotations increases robustness to test-time hue shifts at the cost of a slight reduction in network capacity, b) removing group coset pooling breaks hue invariance, and c) hue equivariant networks require lower intensity color jitter augmentations to achieve the same test-time hue shift robustness and accuracy. The full results can be found in Appendix D.

5 Conclusion

In this work, we propose Color Equivariant Convolutions (CEConvs) which enable feature sharing across colors in the data, while retaining discriminative power. Our toy experiments demonstrate benefits for datasets where the color distribution is long-tailed or biased. Our proposed fully equivariant CECNNs improve performance on datasets where features are color selective, while hybrid versions that selectively apply CEConvs only in early stages of a CNN benefit various classification tasks.

Limitations CEConvs are computationally more expensive than regular convolutions. For fair comparison, we have equalized the parameter cost of all models compared, at the cost of reducing the number of channels of CECNNs. In cases where color equivariance is not a useful prior, the reduced capacity hurts model performance, as reflected in our experimental results.

Pixel values near the borders of the RGB cube can fall outside the cube after rotation, and subsequently need to be reprojected. Due to this clipping effect the hue equivariance in Eq. (9) only holds approximately. As demonstrated empirically, this has only limited practical consequences, yet future work should investigate how this shortcoming could be mitigated.

Local vs. global equivariance The proposed CEConv implements local hue equivariance, i.e. it allows to model local color changes in different regions of an image separately. In contrast, global equivariance, e.g. by performing hue shifts on the full input image, then processing all inputs with the same CNN and combining representations at the final layer to get a hue-equivariant representation, encodes global equivariance to the entire image. While we have also considered such setup, initial experiments did not yield promising results. The theoretical benefit of local over global hue equivariance is that multiple objects in one image can be recognized equivariantly in any combination of hues - empirically this indeed proves to be a useful property.

Future work The group of hue shifts is but one of many possible transformations groups on images. CNNs naturally learn features that vary in both photometric and geometric transformations [5, 37]. Future work could combine hue shifts with geometric transformations such as roto-translation [7] and scaling [49]. Also, other photometric properties could be explored in an equivariance setting, such as saturation and brightness.

Our proposed method rotates the hue of the inputs by a predetermined angle as encoded in a rotation matrix. Making this rotation matrix learnable could yield an inexact but more flexible type of color equivariance, in line with recent works on learnable equivariance [34, 46]. An additional line of interesting future work is to incorporate more fine-grained equivariance to continuous hue shifts, which is currently intractable within the GConv-inspired framework as the number multiply-accumulate operations grow quadratically with the number of hue rotations.

Broader impact Improving performance on tasks where color is a discriminative feature could affect humans that are the target of discrimination based on the color of their skin. CEConvs ideally benefit datasets with long-tailed color distributions by increasing robustness to color changes, in theory reducing a CNN's reliance on skin tone as a discriminating factor. However, careful and rigorous evaluation is needed before such properties can be attributed to CECNNs with certainty.

Acknowledgements

This project is supported in part by NWO (project VI.Vidi.192.100).

References

[1] Naif Alshammari, Samet Akcay, and Toby P. Breckon. On the impact of illumination-invariant image pre-transformation for contemporary automotive semantic scene understanding. In *2018 IEEE Intelligent Vehicles Symposium (IV)*, pages 1027–1032, 2018.

[2] Erik J. Bekkers, Maxime W. Lafarge, Mitko Veta, Koen A. J. Eppenhof, Josien P. W. Pluim, and Remco Duits. Roto-translation covariant convolutional networks for medical image analysis. In *Medical Image Computing and Computer Assisted Intervention – MICCAI 2018*, pages 440–448. Springer International Publishing, 2018.

[3] Inês Bramão, Luís Faísca, Karl Magnus Petersson, and Alexandra Reis. The contribution of color to object recognition. In Ioannis Kypraios, editor, *Advances in Object Recognition Systems*, chapter 4. IntechOpen, Rijeka, 2012.

[4] Michael M. Bronstein, Joan Bruna, Taco Cohen, and Petar Velickovic. Geometric deep learning: Grids, groups, graphs, geodesics, and gauges. *CoRR*, abs/2104.13478, 2021.

[5] Robert-Jan Bruintjes, Tomasz Motyka, and Jan van Gemert. What affects learned equivariance in deep image recognition models? *arXiv preprint arXiv:2304.02628*, 2023.

[6] Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 215–223. JMLR Workshop and Conference Proceedings, 2011.

[7] Taco S. Cohen and Max Welling. Group equivariant convolutional networks. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, ICML'16, page 2990–2999. JMLR.org, 2016.

[8] Ian R. Cole. Modelling CPV. 6 2015.

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

[10] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009.

[11] Li Deng. The mnist database of handwritten digit images for machine learning research. *IEEE Signal Processing Magazine*, 29(6):141–142, 2012.

[12] Martin Engilberge, Edo Collins, and Sabine Süsstrunk. Color representation in deep neural networks. In *2017 IEEE International Conference on Image Processing (ICIP)*, pages 2786–2790, 2017.

[13] G.D. Finlayson, M.S. Drew, and B.V. Funt. Diagonal transforms suffice for color constancy. In *1993 (4th) International Conference on Computer Vision*, pages 164–171, 1993.

[14] G.D. Finlayson, S.D. Hordley, Cheng Lu, and M.S. Drew. On the removal of shadows from images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(1):59–68, 2006.

[15] B.V. Funt and G.D. Finlayson. Color constant color indexing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(5):522–529, 1995.

[16] Chase J. Gaudet and A. Maida. Deep quaternion networks. *2018 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, 2018.

[17] J. M. Geusebroek, R. van den Boomgaard, A. W. M. Smeulders, and H. Geerts. Color invariance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(12):1338–1350, 2001.

[18] T. Gevers, A. Gijsenij, J. van de Weijer, and J. M. Geusebroek. *Color in Computer Vision : Fundamentals and Applications*. Series in Imaging Science and Technology. The Wiley-IS&T, 2012.

[19] T. Gevers, A. Gijsenij, J. van de Weijer, and J. M. Geusebroek. *Color in Computer Vision : Fundamentals and Applications*. Series in Imaging Science and Technology. The Wiley-IS&T, 2012.

[20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In *Proceedings of 2016 IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778. IEEE, June 2016.

[21] Dan Hendrycks, Norman Mu, Ekin D. Cubuk, Barret Zoph, Justin Gilmer, and Balaji Lakshminarayanan. AugMix: A simple data processing method to improve robustness and uncertainty. *Proceedings of the International Conference on Learning Representations (ICLR)*, 2020.

[22] Geoffrey E Hinton, Sara Sabour, and Nicolas Frosst. Matrix capsules with EM routing. In *International Conference on Learning Representations*, 2018.

[23] Osman Semih Kayhan and Jan C van Gemert. On translation invariance in cnns: Convolutional layers can exploit absolute spatial location. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14274–14285, 2020.

[24] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *International Conference on Learning Representations*, 12 2014.

[25] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014.

[26] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *2013 IEEE International Conference on Computer Vision Workshops*, pages 554–561, 2013.

[27] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical Report 0, University of Toronto, Toronto, Ontario, 2009.

[28] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. In *Proceedings of the IEEE*, volume 86, pages 2278–2324, 1998.

[29] Attila Lengyel, Sourav Garg, Michael Milford, and Jan C. van Gemert. Zero-shot day-night domain adaptation with a physics prior. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4399–4409, October 2021.

[30] Attila Lengyel and Jan van Gemert. Exploiting learned symmetries in group equivariant convolutions. In *2021 IEEE International Conference on Image Processing (ICIP)*, pages 759–763, 2021.

[31] Fei-Fei Li, Marco Andreeto, Marc'Aurelio Ranzato, and Pietro Perona. Caltech 101, Apr 2022.

[32] Lachlan E. MacDonald, Sameera Ramasinghe, and Simon Lucey. Enabling equivariance for arbitrary lie groups. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8183–8192, June 2022.

[33] 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 *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1277–1285, 2019.

[34] Artem Moskalev, Anna Sepliarskaia, Ivan Sosnovik, and Arnold Smeulders. Liegg: Studying learned lie group generators. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 25212–25223. Curran Associates, Inc., 2022.

[35] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *Indian Conference on Computer Vision, Graphics and Image Processing*, Dec 2008.

[36] Chris Olah, Nick Cammarata, Ludwig Schubert, Gabriel Goh, Michael Petrov, and Shan Carter. An overview of early vision in inceptionv1. *Distill*, 2020. https://distill.pub/2020/circuits/early-vision.

[37] Chris Olah, Nick Cammarata, Chelsea Voss, Ludwig Schubert, and Gabriel Goh. Naturally occurring equivariance in neural networks. *Distill*, 2020. https://distill.pub/2020/circuits/equivariance.

[38] Daniel Osorio and Misha Vorobyev. Colour vision as an adaptation to frugivory in primates. *Proceedings. Biological sciences / The Royal Society*, 263:593–9, 06 1996.

[39] Gokhan Ozbulak. Image colorization by capsule networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2019.

[40] Omkar M. Parkhi, Andrea Vedaldi, Andrew Zisserman, and C. V. Jawahar. Cats and dogs. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2012.

[41] Ivet Rafegas and Maria Vanrell. Color representation in cnns: Parallelisms with biological vision. In *2017 IEEE International Conference on Computer Vision Workshops (ICCVW)*, pages 2697–2705, 2017.
[42] Ivet Rafegas and Maria Vanrell. Color encoding in biologically-inspired convolutional neural networks. *Vision Research*, 151:7–17, 2018. Color: cone opponency and beyond.
[43] Ivet Rafegas, Maria Vanrell, Luís A. Alexandre, and Guillem Arias. Understanding trained cnns by indexing neuron selectivity. *Pattern Recognition Letters*, 136:318–325, 2020.
[44] M. Rath and A. Condurache. Boosting deep neural networks with geometrical prior knowledge: A survey. *ArXiv*, abs/2006.16867, 2020.
[45] B Regan, C Julliot, Bruno Simmen, Françoise Viénot, P.C. Charles-Dominique, and John Mollon. Fruits, foliage and the evolution of primate colour vision. *Philosophical transactions of the Royal Society of London. Series B, Biological sciences*, 356:229–83, 03 2001.
[46] David W. Romero and Suhas Lohit. Learning partial equivariances from data. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 36466–36478. Curran Associates, Inc., 2022.
[47] Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton. Dynamic routing between capsules. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.
[48] Christos Sakaridis, Dengxin Dai, and Luc Van Gool. Guided curriculum model adaptation and uncertainty-aware evaluation for semantic nighttime image segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.
[49] Ivan Sosnovik, Artem Moskalev, and Arnold Smeulders. Disco: accurate discrete scale convolutions. In *Proceedings of the 32nd British Machine Vision Conference (BMVC)*, 2021.
[50] Ivan Sosnovik, Michał Szmaja, and Arnold Smeulders. Scale-equivariant steerable networks. In *International Conference on Learning Representations*, 2020.
[51] Jan C Van Gemert. Exploiting photographic style for category-level image classification by generalizing the spatial pyramid. In *Proceedings of the 1st ACM International Conference on Multimedia Retrieval*, pages 1–8, 2011.
[52] Maurice Weiler, Fred A. Hamprecht, and Martin Storath. Learning steerable filters for rotation equivariant cnns. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.
[53] Daniel Worrall and Max Welling. Deep scale-spaces: Equivariance over scale. In *Advances in Neural Information Processing Systems*, volume 32, pages 7366–7378. Curran Associates, Inc., 2019.
[54] Xuanyu Zhu, Yi Xu, Hongteng Xu, and Changjian Chen. Quaternion convolutional neural networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018.

A Derivation of H_n

Rotation around an arbitrary unit vector $\mathbf{u}$ by angle θ can be decomposed into five simple steps [8]:

1. rotating the vector such that it lies in one of the coordinate planes, e.g. xz using M_{xz};
2. rotating the vector such that it lies on one of the coordinate axes, e.g. x using M_x;
3. rotating the point around vector $\mathbf{u}$ on axis x using R_x;
4. reversing the rotation in step 2. using $M_x^{-1} = M_x^T$;
5. reversing the rotation in step 1. using $M_{xz}^{-1} = M_{xz}^T$.

These operations can be combined into a single matrix:

$$R_{\mathbf{u},\theta} = M_{xz}^T(M_x^T(R_{x,\theta}(M_{xz}(M_{xz})))) \tag{16}$$

$$= M_{xz}^T M_x^T R_{x,\theta} M_{xz} M_{xz} \tag{17}$$

$$= \begin{bmatrix} \cos\theta + u_x^2(1-\cos\theta) & u_x u_y(1-\cos\theta) - u_z\sin\theta & u_x u_z(1-\cos\theta) + u_y\sin\theta \\ u_y u_x(1-\cos\theta) + u_z\sin\theta & \cos\theta + u_y^2(1-\cos\theta) & u_y u_z(1-\cos\theta) - u_x\sin\theta \\ u_z u_x(1-\cos\theta) - u_y\sin\theta & u_z u_y(1-\cos\theta) + u_x\sin\theta & \cos\theta + u_z^2(1-\cos\theta) \end{bmatrix}. \tag{18}$$

Substituting $\mathbf{u} = [\frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}]$ yields

$$R_{\mathbf{u},\theta} = \begin{bmatrix} \cos\theta + \frac{1}{3}(1-\cos\theta) & \frac{1}{3}(1-\cos\theta) - \frac{1}{\sqrt{3}}\sin\theta & \frac{1}{3}(1-\cos\theta) + \frac{1}{\sqrt{3}}\sin\theta \\ \frac{1}{3}(1-\cos\theta) + \frac{1}{\sqrt{3}}\sin\theta & \cos\theta + \frac{1}{3}(1-\cos\theta) & \frac{1}{3}(1-\cos\theta) - \frac{1}{\sqrt{3}}\sin\theta \\ \frac{1}{3}(1-\cos\theta) - \frac{1}{\sqrt{3}}\sin\theta & \frac{1}{3}(1-\cos\theta) + \frac{1}{\sqrt{3}}\sin\theta & \cos\theta + \frac{1}{3}(1-\cos\theta) \end{bmatrix}, \tag{19}$$

and lastly, rearranging and substituting $\theta = \frac{2k\pi}{n}$ results in

$$H_n(k) = \begin{bmatrix} \cos(\frac{2k\pi}{n}) + a & a - b & a + b \\ a + b & \cos(\frac{2k\pi}{n}) + a & a - b \\ a - b & a + b & \cos(\frac{2k\pi}{n}) + a \end{bmatrix}. \tag{20}$$

with n the total number of discrete rotations in the group, k the rotation, $a = \frac{1}{3} - \frac{1}{3}\cos(\frac{2k\pi}{n})$ and $b = \sqrt{\frac{1}{3}} * \sin(\frac{2k\pi}{n})$.

B ColorMNIST

B.1 Dataset visualization

Long-tailed ColorMNIST dataset The training samples of the *Longtailed ColorMNIST* dataset are depicted in Fig. 5, clearly indicating a class imbalance.

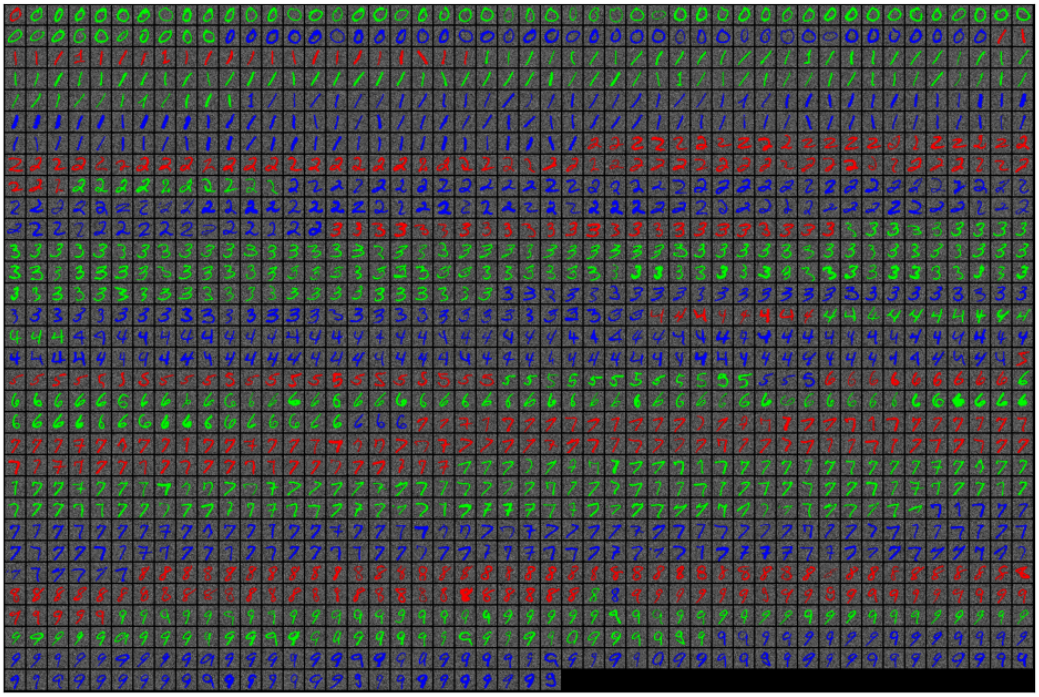
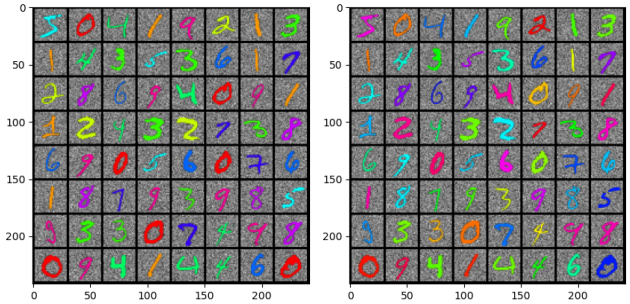


Figure 5: Long-tailed ColorMNIST. Note the strong class imbalance in the dataset. Best viewed in color.

Biased ColorMNIST dataset A small subset of the samples of Biased ColorMNIST is shown in Fig. 6 for $\sigma = 0$ (a) and $\sigma = 36$ (b), respectively. Note that the samples in (a) have a deterministic color, whereas in (b) exhibit some variation in hue.

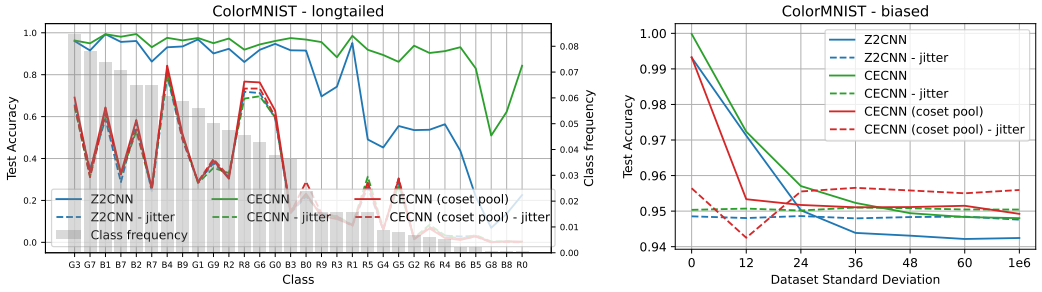


(a) (b)

Figure 6: Samples from Biased ColorMNIST for $\sigma = 0$ (a) and $\sigma = 36$ (b), respectively. Best viewed in color.

B.2 Additional experiments

Results with color jitter augmentation We performed both ColorMNIST experiments with color jitter augmentations. The results are shown in Fig. 7. (a) For long-tailed ColorMNIST, adding jitter makes solving the classification problem prohibitive, as color is required. Z2CNN and CECNN with jitter therefore perform no better than as the CECNN model with coset pooling. (b) For biased MNIST, performance decreases for small and improves for large σ, with CEConv still performing best.



(a) (b)

Figure 7: Color equivariant convolutions efficiently share shape information across different colors. CECNN outperforms a vanilla network in both a long-tailed class imbalance setting (a), where MNIST digits are to be classified based on both shape and color, and a color biased setting (b), where the color of each class c is sampled according to $\theta_d \sim \mathcal{N}(\theta_c, \sigma)$.

Long-tailed ColorMNIST with weighted loss We performed the longtailed ColorMNIST experiment both with a uniformly weighted loss and a loss where classes are weighted inversely to their frequency according to $w_i = \frac{N}{c * n_i}$, where w_i denotes the weight for class i, N the number of samples in the training set, c the number of classes, and n_i the number of samples for class i. The results are shown in Fig. 8. We observed no significant difference between the two setups, with the CECNN without coset pooling outperforming the other models by a large margin in both.

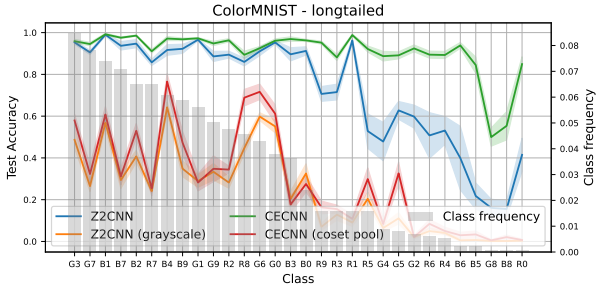


Figure 8: Per-class accuracy of various models trained with a loss function weighted by inverse class frequency. CECNN without coset pooling outperforms all other models, with no significant differences compared to an uniformly weighted loss function.

C Classification experiments

C.1 Overview of all CE-ResNet configurations

Table 2 shows an overview of the classification accuracies of all baselines and equivariant architectures. CEConv-x denotes the number of ResNet stages with CE convolutions with CEConv-4 (3 for CIFAR) being a fully equivariant ResNet. In nearly all cases, early equivariance is beneficial for improving classification accuracy on both the original as well as the hue shifted test sets. In case of the Flowers-

102 dataset late equivariance seems to have a significant advantage, whereas for Caltech-101 and Stanford Cars the color equivariance bias does not seem to have much added value.

Table 2 shows the classification results for all network architectures, trained with and without color jitter.

	Caltech-101	**CIFAR-10**	**CIFAR-100**	**Flowers-102**	**Oxford-IIIT Pet**	**Stanford Cars**	**STL-10**	**ImageNet**
Original test set								
Baseline	71.61 ± 0.87	93.69 ± 0.16	71.28 ± 0.20	66.79 ± 0.89	69.87 ± 0.57	76.54 ± 0.10	83.80 ± 0.36	69.71
CIConv-W	**72.85 ± 1.12**	75.26 ± 0.57	38.81 ± 0.66	68.71 ± 0.29	61.53 ± 0.53	**79.52 ± 0.42**	80.71 ± 0.27	65.81
CEConv-1	71.59 ± 0.64	**94.06 ± 0.09**	71.82 ± 0.36	67.29 ± 0.57	**70.47 ± 1.07**	78.03 ± 0.29	84.34 ± 0.38	**70.05**
CEConv-2	71.50 ± 0.29	93.94 ± 0.07	**72.20 ± 0.48**	68.38 ± 0.55	70.34 ± 0.67	77.06 ± 0.38	**84.50 ± 0.31**	70.02
CEConv-3	70.45 ± 0.41	93.71 ± 0.26	71.37 ± 0.24	**69.42 ± 0.58**	68.92 ± 0.46	75.33 ± 0.66	83.61 ± 0.35	69.35
CEConv-4	70.16 ± 1.05	-	-	68.18 ± 0.45	70.24 ± 0.79	76.22 ± 0.19	84.24 ± 0.49	66.85
Baseline + jitter	73.93 ± 0.73	93.03 ± 0.16	69.23 ± 0.44	68.75 ± 1.50	72.71 ± 0.67	80.59 ± 0.36	83.91 ± 0.38	69.37
CIConv-W + jitter	**74.38 ± 0.43**	77.49 ± 0.53	42.27 ± 0.56	**75.05 ± 0.39**	64.23 ± 0.51	**81.56 ± 0.32**	81.88 ± 0.24	65.95
CEConv-1 + jitter	73.43 ± 0.59	**93.93 ± 0.16**	71.08 ± 0.27	70.39 ± 0.81	72.44 ± 0.76	80.24 ± 0.51	84.31 ± 0.47	69.36
CEConv-2 + jitter	72.61 ± 0.95	93.86 ± 0.22	**71.35 ± 0.20**	71.72 ± 0.63	72.80 ± 0.87	80.32 ± 0.47	**84.46 ± 0.39**	**69.42**
CEConv-3 + jitter	73.21 ± 0.87	93.51 ± 0.10	71.12 ± 0.57	72.71 ± 0.23	72.55 ± 0.67	79.62 ± 0.54	84.08 ± 0.44	69.10
CEConv-4 + jitter	73.58 ± 0.68	-	-	74.17 ± 0.49	**73.28 ± 0.63**	79.79 ± 0.37	84.16 ± 0.10	65.57
Baseline + AugMix	**71.92 ± 0.95**	94.13 ± 0.22	**72.64 ± 0.27**	75.49 ± 0.24	**76.02 ± 0.51**	**82.32 ± 0.07**	84.99 ± 0.24	-
CEConv + AugMix	70.74 ± 1.12	**94.22 ± 0.16**	72.48 ± 0.18	**78.10 ± 0.50**	75.90 ± 0.22	80.81 ± 0.27	**85.46 ± 0.30**	-
Hue-shifted test set								
Baseline	51.14 ± 0.71	85.26 ± 0.56	47.01 ± 0.38	13.41 ± 0.34	37.56 ± 0.76	55.59 ± 0.74	67.60 ± 0.56	54.72
CIConv-W	**71.92 ± 1.11**	74.88 ± 0.54	37.09 ± 0.74	**59.03 ± 0.62**	**60.54 ± 0.46**	**78.71 ± 0.33**	**79.92 ± 0.25**	**64.62**
CEConv-1	65.60 ± 0.47	**91.93 ± 0.14**	**63.37 ± 0.17**	32.88 ± 0.83	52.97 ± 1.00	70.08 ± 0.21	78.83 ± 0.43	63.02
CEConv-2	64.51 ± 0.64	91.43 ± 0.18	62.11 ± 0.43	33.32 ± 0.55	51.14 ± 0.95	68.17 ± 0.86	77.80 ± 0.58	62.26
CEConv-3	62.22 ± 0.99	90.90 ± 0.25	59.04 ± 0.45	33.76 ± 0.38	49.45 ± 0.65	65.82 ± 1.34	76.23 ± 0.37	60.95
CEConv-4	62.17 ± 1.01	-	-	33.33 ± 0.38	54.02 ± 1.34	67.16 ± 0.58	78.25 ± 0.52	56.90
Baseline + jitter	73.61 ± 0.60	92.91 ± 0.17	69.12 ± 0.47	68.44 ± 1.60	72.31 ± 0.49	80.65 ± 0.36	83.71 ± 0.35	67.10
CIConv-W + jitter	**74.40 ± 0.55**	77.28 ± 0.54	42.30 ± 0.48	**75.66 ± 0.27**	63.93 ± 0.42	**81.44 ± 0.26**	81.54 ± 0.21	65.03
CEConv-1 + jitter	73.34 ± 0.96	**93.86 ± 0.20**	70.98 ± 0.22	69.98 ± 0.79	72.34 ± 0.58	80.18 ± 0.50	84.29 ± 0.50	**68.85**
CEConv-2 + jitter	73.03 ± 0.97	93.80 ± 0.14	**71.33 ± 0.19**	71.44 ± 0.57	72.58 ± 0.86	80.28 ± 0.52	**84.31 ± 0.34**	68.74
CEConv-3 + jitter	73.26 ± 0.74	93.39 ± 0.08	71.06 ± 0.53	72.47 ± 0.20	72.32 ± 0.64	79.62 ± 0.54	84.00 ± 0.33	68.03
CEConv-4 + jitter	73.57 ± 0.75	-	-	73.86 ± 0.39	**72.94 ± 0.56**	79.79 ± 0.34	84.02 ± 0.14	64.52
Baseline + AugMix	51.82 ± 0.60	88.03 ± 0.26	51.39 ± 0.19	15.99 ± 0.28	48.04 ± 0.74	68.69 ± 0.73	72.19 ± 0.45	-
CEConv + AugMix	**62.29 ± 0.97**	**91.68 ± 0.21**	**60.75 ± 0.24**	**41.43 ± 0.97**	**62.27 ± 0.81**	**73.59 ± 0.30**	**80.17 ± 0.15**	-

Table 2: Classification accuracy on various datasets. CEConv-s denotes a ResNet with s color equivariant stages. We report results for models trained with and without color jitter augmentation. (Hybrid) color equivariant networks improve performance over the baseline model on both the original as well as the hue-shifted test set.

C.2 Test-time hue shift plots

Fig. 9 shows the test accuracies under a test time hue shift on all datasets in the paper. Each figure includes a regular ResNet, a color equivariant ResNet-x (CE-ResNet-x) and a ResNet-x with color equivariant convolutions in the first ResNet stage (CE-ResNet-x-1), trained with and without color jitter augmentation. Finally, the plot shows the accuracy of a ResNet-x trained on grayscale inputs. CEConv improves robustness to test-time hue shifts on all datasets.

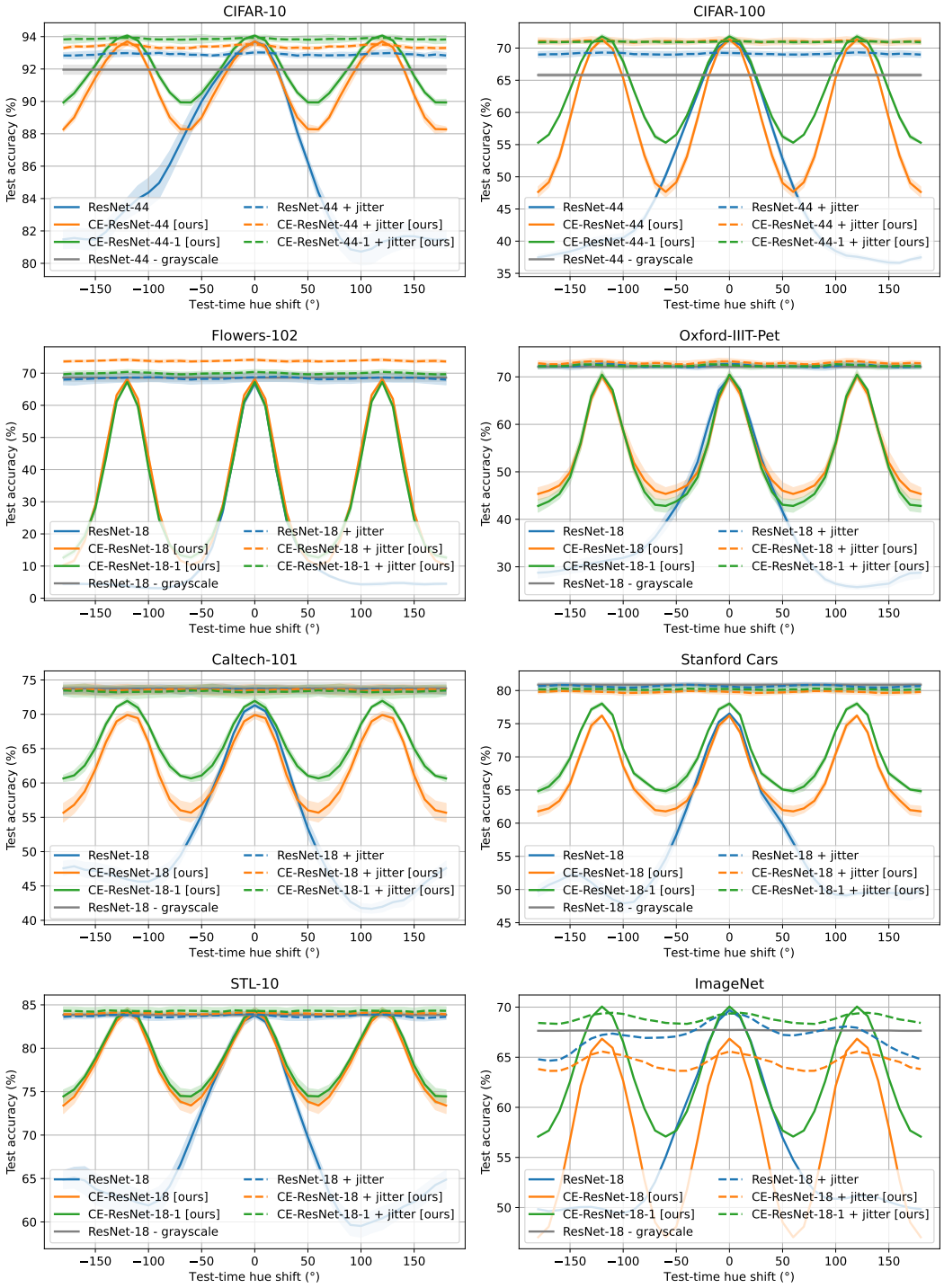


Figure 9: Test accuracy on various classification datasets under a test time hue shift.

C.3 CE-ResNet configurations

The configurations of the color equivariant ResNet with three hue rotations, as used in the classification experiment in Section 4.2, are shown in Table 3. CE stages 0 denotes a regular ResNet.

Model	CE stages	Width	Parameters (M)	MACs (G)
ResNet-18	0	64	11.69	3.59
	1	63	11.38	5.66
	2	63	11.57	7.37
	3	61	11.54	8.80
	4	55	11.79	10.32
ResNet-44	0	32	2.64	0.78
	1	31	2.51	1.23
	2	30	2.50	1.63
	3	27	2.60	1.83

Table 3: Color equivariant ResNet configurations.

C.4 Neuron Feature visualizations

Fig. 10 shows the Neuron Feature [42] (NF) visualization with top-3 patches of two neurons at different stages in a CE-ResNet18 trained on Stanford Cars. As expected, each row of a NF activates on the same shape in a different color. We show neurons that are insensitive to color (top row) and neurons that are sensitive to color (bottom row).

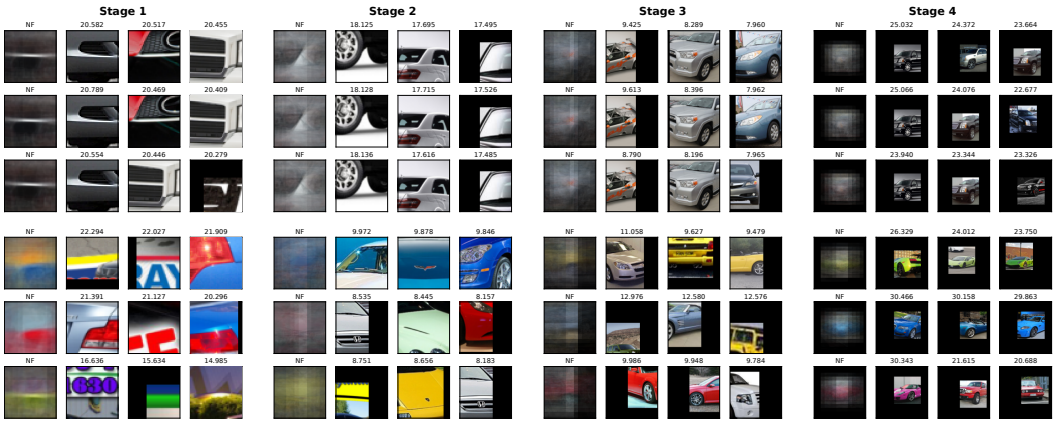
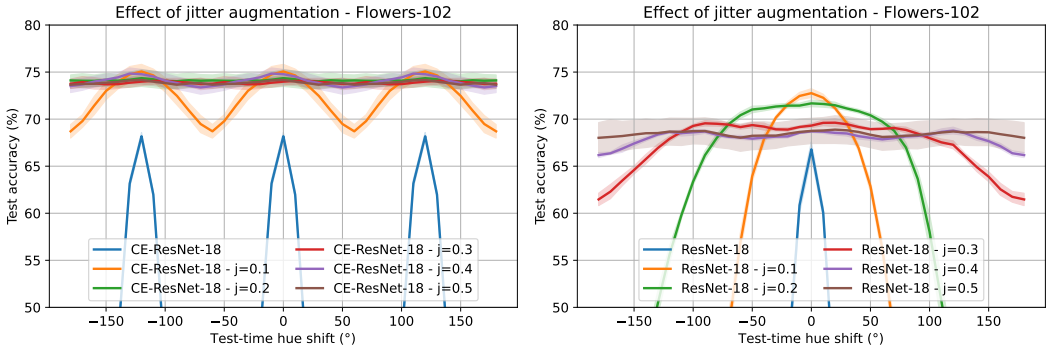


Figure 10: Neuron Feature [42] (NF) visualization with top-3 patches of two neurons at different stages in a CE-ResNet18 trained on Stanford Cars. Rows represent different rotations of the same filter.

D Ablation studies

Strength of color jitter augmentations Fig. 11 shows the effect of hue jitter augmentation during training on both a color equivariant ResNet-18 with 3 rotations (a) and a regular ResNet-18 (b) trained on Flowers-102. All runs have been repeated 3 times and the mean performance is reported. As expected, the color equivariant network (a) without jitter augmentation is equivariant to rotations of multiples of 120 degrees, but performance quickly degrades. Applying slight (0.1) hue jitter during training both helps in an absolute sense, increasing performance over all rotations, and makes the network more robust to hue changes as shown by the increasing width of the peaks. Further increasing the strength of the augmentation results in a uniform performance over all hue shifts, indicated by the flat lines. There appears to be no significant difference for jitter strength > 0.2. In comparison, the

regular ResNet (b) trained without hue augmentation shows a single peak around 0 degrees, which increases in width when applying more severe augmentation. Note that the increase in absolute performance is smaller compared to the color equivariant network. The reason for this is that the equivariant architecture only requires augmentation "between" the discrete rotations to which it is already robust, as opposed to the full scale of hue shifts for the baseline architecture. Augmentation and equivariance thus exhibit a remarkable synergistic interaction.



(a) (b)

Figure 11: Effect of hue jitter augmentation on a color equivariant (a) and a regular (b) ResNet-18.

Group coset pooling We have removed the group coset pooling operation by flattening the feature map group dimension into the channel dimension in the penultimate layer, before applying the final classification layer. As shown in Fig. 12, the model without pooling layer is no longer invariant to hue shifts and behaves identically to the baseline model.

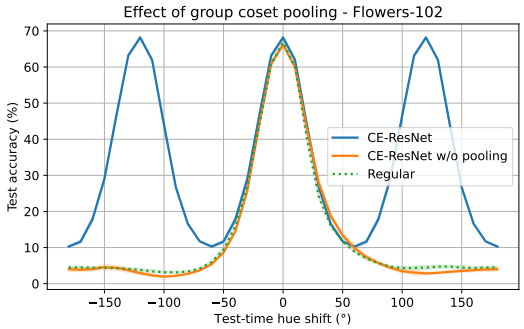
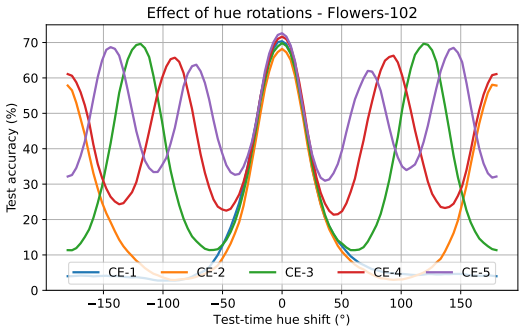


Figure 12: CE-ResNet without group coset pooling behaves similarly to a regular ResNet (average over 5 runs).

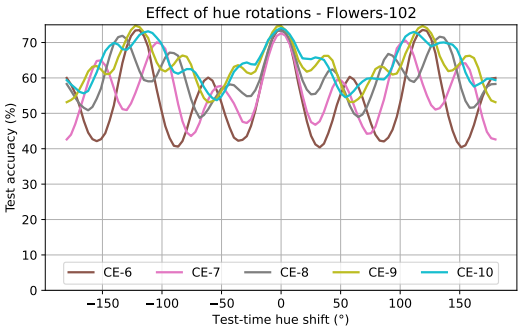
Number of color rotations We investigate the effect of the number of hue rotations in color equivariant convolutions by training CE-ResNets with 2-10 rotations on Flowers-102. Fig. 13 shows the test accuracies for rotations 1-5 (a) and 6-10 (b), respectively. Note that, for this particular dataset, more hue rotations not only lead to better robustness to test-time hue shifts, but also to better absolute performance. However, there is a trade-off between number of rotations and model capacity, as increasing the number of rotations increases the number of parameters in the model, and the model width needs to be scaled down to keep the number of parameters equal. Both the optimal number of color rotations and network width therefore depend on the amount of color vs. the complexity of the data, and therefore both need to be carefully calibrated per dataset.

As expected, the number of peaks increases with the number of hue rotations, though interestingly, the peaks do vary in height. This is an artifact due the way test-time hue shifts are applied to the input images. When RGB pixels are rotated about the [1,1,1] diagonal, values near the borders of the RGB cube tend to fall outside the cube and subsequently need to be reprojected. This reprojection is not modeled by the filter transformations in the CEConv layers, and subsequently

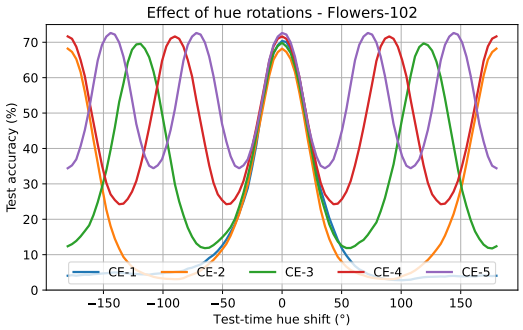
causes a discrepancy between the filter and the image transformations. Indeed, when the test-time hue shift is instead implemented through a rotation in RGB space without reprojecting into the cube, this artifact disappears and all peaks are of equal height, as shown in Fig. 13 (c-d). Note that rotations of multiples of 120 degrees always end up within the RGB cube, which is why this artifact does never occur at -120, 0 and 120 degrees. Future work should further investigate the extent to which this discrepancy is problematic in practice, and look into alternative solutions.



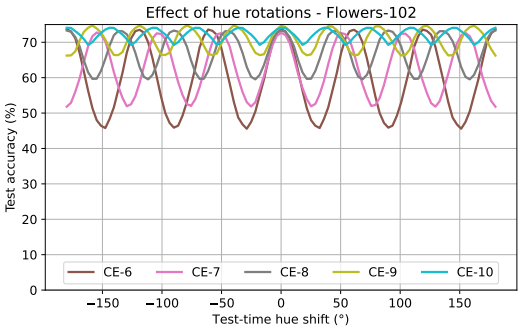
(a) Test-time hue shift with reprojection.



(b) Test-time hue shift with reprojection.



(c) Test-time hue shift without reprojection.



(d) Test-time hue shift without reprojection.

Figure 13: The effect of the number of hue rotations in color equivariant convolutions on downstream performance. More rotations increases robustness to test-time hue shifts. Note that in (a-b) the peaks are not of equal height due to clipping effects near the boundaries of the RGB cube. This artifact disappears when the test-time hue shift is also applied without reprojection, resulting in peaks of equal height (c-d).