



Delft University of Technology

On Color and Symmetries for Data Efficient Deep Learning

Lengyel, A.

DOI

[10.4233/uuid:079a4a73-1445-44f9-ac87-a0bc312b71ad](https://doi.org/10.4233/uuid:079a4a73-1445-44f9-ac87-a0bc312b71ad)

Publication date

2024

Document Version

Final published version

Citation (APA)

Lengyel, A. (2024). *On Color and Symmetries for Data Efficient Deep Learning*. [Dissertation (TU Delft), Delft University of Technology]. <https://doi.org/10.4233/uuid:079a4a73-1445-44f9-ac87-a0bc312b71ad>

Important note

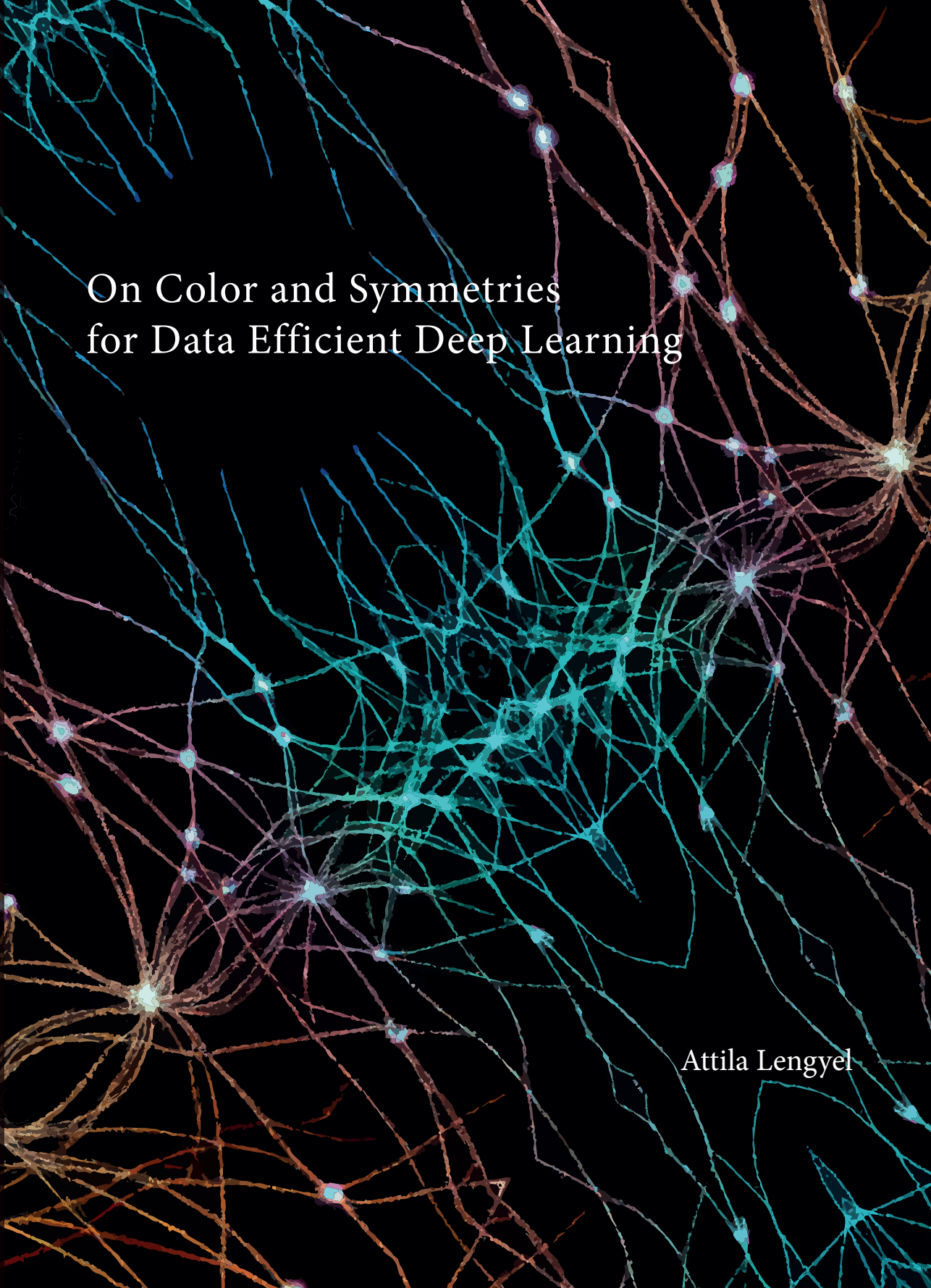
To cite this publication, please use the final published version (if applicable).
Please check the document version above.

Copyright

Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

Takedown policy

Please contact us and provide details if you believe this document breaches copyrights.
We will remove access to the work immediately and investigate your claim.



On Color and Symmetries
for Data Efficient Deep Learning

Attila Lengyel

**ON COLOR AND SYMMETRIES
FOR DATA EFFICIENT DEEP LEARNING**

ON COLOR AND SYMMETRIES FOR DATA EFFICIENT DEEP LEARNING

Dissertation

for the purpose of obtaining the degree of doctor
at Delft University of Technology
by the authority of the Rector Magnificus, prof. dr. ir. T.H.J.J. van der Hagen,
chair of the Board for Doctorates
to be defended publicly on
Friday, the 26th of April 2024 at 12:30

by

Attila LENGYEL

Master of Science in Electrical Engineering,
Delft University of Technology, The Netherlands,
born in Nové Zámky, Slovak Republic.

This dissertation has been approved by the promotors.

Composition of the doctoral committee:

Rector Magnificus,	chairperson
Prof. dr. M.J.T. Reinders,	Delft University of Technology, <i>promotor</i>
Dr. J.C. van Gemert,	Delft University of Technology, <i>promotor</i>

Independent members:

Prof. dr. S.C. Pont,	Delft University of Technology
Prof. dr. T. Gevers,	University of Amsterdam
Dr. J. van de Weijer,	Universitat Autònoma de Barcelona
Dr. E. Trulls,	Google
Prof. dr. G.C.H.E. de Croon,	Delft University of Technology, <i>reserve member</i>



Keywords: computer vision, visual inductive priors, data efficiency, equiv-
ariance, invariance, color

Printed by: ProefschriftMaken.nl

Cover by: Ghiline van Furth, Hannah Vear

Copyright © 2024 by A. Lengyel

ISBN 978-94-6366-854-5

An electronic copy of this dissertation is available at
<https://repository.tudelft.nl/>.

CONTENTS

Summary	vii
Samenvatting	ix
Összefoglalás	xi
1 Introduction	1
1.1 What is machine learning?	3
1.2 Color in computer vision	3
1.3 Deep learning	7
1.4 Equivariance in neural networks	8
1.5 Data augmentation	11
1.6 Organization of this thesis	12
References	16
2 Zero-Shot Day-Night Domain Adaptation with a Physics Prior	19
2.1 Introduction	20
2.2 Related work	22
2.3 Method	23
2.4 Experiments	26
2.5 Discussion	35
References	37
Appendices	45
3 Color Equivariant Convolutional Networks	53
3.1 Introduction	54
3.2 Related work	56
3.3 Color equivariant convolutions	58
3.4 Experiments	62
3.5 Conclusion	68
References	70
Appendices	75

4	Exploiting Learned Symmetries in Group Equivariant CNNs	89
4.1	Introduction	90
4.2	Related Work	90
4.3	Method	92
4.4	Experiments	96
4.5	Discussion	98
	References	99
5	Using and Abusing Equivariance	101
5.1	Introduction	102
5.2	Related Work	104
5.3	How subsampling breaks equivariance	106
5.4	Experiments	110
5.5	Conclusion	118
5.6	Limitations and Future Work	119
	References	120
	Appendices	125
6	Discussion	129
6.1	Data and compute efficiency	129
6.2	Approximate knowledge priors	130
6.3	Future outlook	132
6.4	Final words	134
	References	135
	Acknowledgments	137
	List of Publications	141
	Curriculum Vitæ	143

SUMMARY

Computer vision algorithms are getting more advanced by the day and slowly approach human-like capabilities, such as detecting objects in cluttered scenes and recognizing facial expressions. Yet, computers learn to perform these tasks very differently from humans. Where humans can generalize between different lighting conditions or geometric orientations with ease, computers require vast amounts of training data to adapt from day to night images, or even to recognize a cat hanging upside-down. This requires additional data, annotations and compute power, increasing the development costs of useful computer vision models. This thesis is therefore concerned with reducing the data and compute hunger of computer vision algorithms by incorporating *prior knowledge* into the model architecture. Knowledge that is built in no longer needs to be learned from data.

This thesis considers various knowledge priors. To improve the robustness of deep learning models to changes in illumination, we make use of color invariant representations derived from physics-based reflection models. We find that a color invariant input layer effectively normalizes the feature map activations throughout the entire network, thereby reducing the distribution shift that normally occurs between day and night images.

Equivariance has proven to be a useful network property for improving data efficiency. We introduce the color equivariant convolution, where spatial features are explicitly shared between different colors. This improves generalization to out-of-distribution colors, and therefore reduces the amount of required training data.

We subsequently investigate Group Equivariant Convolutions (GConvs). First, we discover that GConv filters learn redundant symmetries, which can be hard-coded using separable convolutions. This preserves equivariance to rotation and mirroring, and improves data and compute efficiency. We also explore the notion of approximate equivariance in GConvs. Subsampling is known to introduce equivariance errors in regular convolutional layers, and we find that it similarly breaks exact equivariance for rotation and mirroring. This turns out to be a double-edged sword: while it improves performance on in-distribution data, at the same time it negatively affects out-of-distribution generalization. Finally, we show that exact equivariance can be restored by

choosing an appropriate input size.

This thesis aims to provide a step forward in the adoption of invariant and equivariant architectures to improve data and compute efficiency in deep learning.

SAMENVATTING

Beeldherkenningsalgoritmen worden met de dag geavanceerder en beginnen langzamerhand mensachtige vaardigheden te vertonen, zoals het detecteren van objecten in complexe scènes of het herkennen van gezichtsuitdrukkingen. Echter leren computers deze vaardigheden op een heel andere manier dan mensen. Waar wij in staat om ons met gemak aan te passen op verschillende lichtomstandigheden of om grotere objecten te herkennen, hebben computers enorme hoeveelheden trainingsdata nodig om te generaliseren tussen dag- en nachtbeelden en voorwerpen in verschillende geometrische oriëntaties. Dit vereist extra trainingsdata, annotaties en rekenkracht, met alsmaar stijgende ontwikkelingskosten voor AI-modellen als gevolg. Computer vision-algoritmen zijn dorstig naar data en rekenkracht, en dit proefschrift tracht deze dorst te lessen door *voorkennis* in de modelarchitectuur op te nemen. Kennis die is ingebouwd, hoeft niet langer uit data te worden geleerd.

Dit proefschrift verkent verschillende vormen van voorkennis. Om deep learning-modellen robuuster te maken voor veranderende lichtomstandigheden maken we gebruik van kleurinvariante representaties, welke zijn afgeleid van fysieke reflectiemodellen. Een kleurinvariante invoerlaag blijkt zeer effectief in het normaliseren van de feature map-activaties in het gehele netwerk, wat op zijn beurt de distributievervalsing die normaliter optreedt tussen overdag- en nachtbeelden vermindert.

Equivariantie is een nuttige eigenschap voor het efficiënter maken van neurale netwerken. We introduceren de kleur-equivariante convolutie, waarmee spatiële features expliciet worden gedeeld tussen verschillende kleuren. Dit verbetert de generalisatie naar kleuren buiten de trainingsdistributie, en vermindert daardoor de hoeveelheid vereiste trainingsdata.

Vervolgens onderzoeken we Groep Equivariante Convoluties (GConvs). Allereerst ontdekken we dat geleerde GConv-filters overtollige symmetrieën bevatten, die wij direct in de architectuur inbakken met behulp van separerbare convoluties. Dit zorgt voor een verbeterde data- en rekenefficiëntie, waarbij de equivariantie-eigenschappen van het netwerk behouden blijven. We onderzoeken tevens onder welke voorwaarden GConvs exact equivariant zijn. Zoals reeds bekend, introduceren subsampling-lagen equivariantiefouten in reguliere convoluties. We constateren dat het op vergelijkbare wijze

de exacte equivariantie voor rotatie en spiegeling verbreekt. Dit blijkt een tweesnijdend zwaard te zijn: het verbetert enerzijds de prestaties op data binnen de trainingsdistributie, maar heeft tegelijkertijd een negatief effect op de generalisatie daarbuiten. Tenslotte laten we zien dat exacte equivariantie eenvoudig hersteld kan worden door een geschikte invoergrootte te kiezen.

Dit proefschrift heeft als doel om de adoptie van in- en equivariante architecturen en stap dichterbij te brengen, en om daarmee de data- en rekenefficiëntie van deep learning-modellen te verbeteren.

ÖSSZEFOGLALÁS

A számítógépes látástechnológia napról napra fejlődik, és lassan emberhez hasonló képességeket közelít meg, például tárgyak észlelését komplex környezetekben, vagy arckifejezések felismerését. A számítógép ennek ellenére az embertől nagyon eltérő módon tanulja meg e feladatok elvégzését. Míg az ember könnyedén képes általánosítani különböző fényviszonyok vagy geometriai orientációk között, a számítógépnek hatalmas mennyiségű betanító adatra van szüksége ahhoz, hogy nappali és éjszakai viszonyok között alkalmazkodjon, vagy például felismerjen egy fejjel lefelé ábrázolt macskát. Mindehhez további adatra, annotációra és számítási teljesítményre van szükség, ami következőképpen növeli a számítógépes látásmodellek fejlesztési költségét. E disszertáció célja a számítógépes látással kapcsolatos algoritmusok adat- és számításbeli igényének csökkentése az elsődleges tudás beépítésének segítségével – a modellarchitektúrába beépített tudást elvéve nem szükséges újra adatból megtanulni.

A disszertáció különböző típusú elsődleges tudást vizsgál. Elsősorban a mélytanulási modellek fényviszonyváltozásokkal szembeni robusztusságát szánjuk növelni. Ez érdekében fizikai alapú reflexiós modellekből származó színinvariáns reprezentációkat használunk. Ennek alapján úgy találjuk, hogy egy színinvariáns bemeneti réteg hatékonyan normalizálja a jellemzőtérkép-aktiválásokat az egész hálózaton keresztül, ezáltal csökkentve a nappali és éjszakai képek között általában előforduló eloszláseltolódást.

Az ekvivariancia hasznos hálózati tulajdonságnak bizonyult adathatékony-ság növelésében. E disszertáció bemutatja a Szín-Ekvivariáns Konvolúciót, ami lehetővé teszi a térbeli jellemzők megosztását különböző színek között. Ez javítja az új, eddig nem észlelt színekre való általánosítást, és ezáltal csökkenti a szükséges betanítási adat mennyiségét.

Ezt követően a Csoport Ekvivariáns Konvolúció (CKonv) számos tulajdonságát vizsgáljuk. Először is bemutatjuk, hogy a betanított CKonv szűrők redundáns szimmetriákat tartalmaznak, amelyek szeparálható konvolúciókkal keményen kódolhatók. Ez megőrzi a háló ekvivarianciáját a forgatással és tükrözéssel szemben, valamint javítja az adat- és számítási hatékonyságot. Továbbá megvizsgáljuk a CKonv ekvivariánciai pontosságát is. Az almintavételről ismert, hogy szabályos konvolúciós rétegekben is az ekvivariancia

hibákat okoz. Úgy találjuk, hogy az almintavétel hasonlóan megtöri a pontos ekvivarianciát a forgatás és tükrözés esetén is. Ez valójában ambivalensnek bizonyul: miközben javítja a teljesítményt az eloszláson belüli adatokon, ugyanakkor negatívan befolyásolja az eloszláson kívüli általánosítást. Végül megmutatjuk, hogy a pontos ekvivariancia helyreállítható a megfelelő bemeneti méret beállításával.

E dolgozat célja, hogy előrelépést biztosítson a mélytanulásban az adat- és számítási hatékonyság javítására szolgáló in- és ekvivariáns architektúrák alkalmazása terén.

1

INTRODUCTION

The tantalizing promise of artificial intelligence has captivated our minds for many decades. We have witnessed several breakthrough moments where AI appeared to have surpassed human capabilities - take for example the historic match in 1997 where IBM's Deep Blue beat chess grandmaster Garry Kasparov, or in 2016, when DeepMind's AlphaGo triumphed over the champion go player Lee Sedol. Yet, though these are impressive feats, progress was mostly limited in being highly domain specific and often relying on brute-forced search algorithms rather than artificial human-like intelligence. Until 2022, when the world witnessed the next AI breakthrough moment with the emergence of generative AI tools like ChatGPT [1] and DALL·E [2]. Generative AI exhibits remarkable capabilities and offers a natural way of interacting, which is why it quickly became adopted into the everyday routines of the general public. Based on a simple query like "What is the impact of air pollution on global warming?", a model is now able to provide us a detailed response accompanied by photorealistic visualizations, as depicted in Figure 1.

Ironically, generative AI models consume vast amounts of electricity and thereby contribute significantly to our carbon footprint. Emissions from training GPT3.5, the large language model powering ChatGPT, are estimated at a staggering 552 metric tonnes of CO2 equivalents [3]. Similar figures hold for text-to-image generation models like DALL·E and Stable Diffusion [4, 5]. As it turns out, progress in deep learning, which is the foundation of modern AI and its subfields *computer vision* and *natural language processing*, is fueled by big data and big compute. Both vision and language models require vast amounts of training data to be able to produce accurate predictions. Collecting and annotating these datasets, often consisting of millions of labeled examples, is costly and difficult. Additionally, due to the trend of ever growing number of parameters in neural networks [6], increasingly large distributed computing systems with high-performance GPUs and TPUs are needed for training and

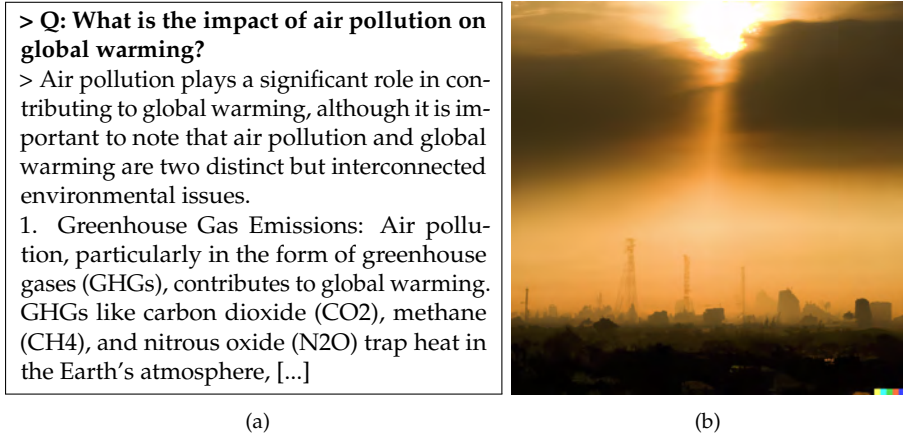


Figure 1.1: Generative AI on the impact of air pollution: (a) ChatGPT is able to generate an elaborate explanation on the various aspects of the topic (text shortened), while (b) DALL-E can generate a relevant photo-realistic image.

deploying models. This is not only problematic from an environmental perspective, it also makes the field less accessible to researchers and organizations with limited resources.

To make deep learning less data-hungry we draw inspiration from techniques before the advent of neural networks, commonly known as classical computer vision. Computer vision is the field of computer science concerned with extracting meaningful information from images and videos. The first step in computer vision algorithms is generally the extraction of distinctive characteristics from input data called *features*, that capture relevant visual or semantic information. Where deep learning utilizes large models to learn custom features directly from the training data, classic computer vision methods rely on manually handcrafted feature descriptors such as SIFT [7], SURF [8] and ORB [9] to obtain a description of the input image. This description incorporates some degree of robustness to input transformations like rotations and scaling, which is desirable to handle changes in camera pose and other real-world variations in data. Similarly, handcrafted color invariants [10, 11] offer a robust image representation to variations in illumination, shadows and shading. These methods were developed based on prior knowledge derived from physics and image processing principles, and, contrary to deep learning, allow general purpose features to be extracted from images with relatively low compute power and without the need for extensive training data. Deep

learning based methods on the other hand are able to extract specialized features that better fit the data distribution, and as a result have outperformed classical computer vision methods by a significant margin.

To combine the best of both worlds, this thesis therefore investigates how to pre-wire deep neural networks with generic visual innate knowledge structures, which allows to incorporate hard won existing knowledge from physics and mathematics, such as light reflection models. Rather than re-learning all knowledge from data, we aim to reduce the data dependency of deep learning models by employing well-proven color manipulation methods from classical computer vision, including color invariants and color transformations. Furthermore, we investigate the properties of Group Equivariant Convolutions on data and compute efficiency.

1.1 WHAT IS MACHINE LEARNING?

Machine learning is the field of computer science that is concerned with learning from data. There are many things a computer could learn from data; for one, computer vision practitioners seem to find great pleasure in teaching them to distinguish between images of cats and dogs. Applications that are often considered more useful include classifying tumorous and healthy cells in medical images, or distinguishing drivable areas from obstacles such as pedestrians and cars in the context of autonomous driving, based on manually labeled data. More formally, in supervised machine learning we have a collection of n data-label pairs called the training set, defined as $D_{\text{train}} = \{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$ and our goal is to learn a function $\hat{y} = f(x)$ that maps an input sample x to the corresponding target value y . Here, x is generally a d -dimensional vector $x \in \mathbb{R}^d$, y represents a discrete class label when the task is classification or a continuous value for regression tasks, and \hat{y} is the prediction by the function. Assuming that the samples are representative of the underlying data distribution $P(X, Y)$, the function can then be applied to new, unseen data samples from the test set D_{test} drawn from the same distribution.

1.2 COLOR IN COMPUTER VISION

Unfortunately, there are many external variables that can cause a distribution shift between the train and test set, which in turn can affect the accuracy of the predictions that a model makes on new, unseen data. One such variable are lighting conditions, which directly influence the pixel measurements obtained

1

by the camera sensor and can thus introduce undesired variations in the recordings. To get a better intuitive understanding of this process we will briefly describe the process of color image formation, i.e. how a camera renders a real-world scene into a digital image. We then explain how a color invariant representation can be obtained that is robust to one or more factors that influence the lighting conditions, most importantly: scene geometry, Fresnel reflections, illumination intensity, and illumination color.

Color image formation Color image formation consists of three stages: *illumination*, *material reflection* and *detection*. The process starts with a light source casting light upon the environment. The light is characterized by its power spectral density $e(\lambda, \mathbf{x})$, which is a function of the wavelength λ and spatial position \mathbf{x} . Colored light has a non-uniform spectral density, while the spectrum of white light is uniform and simplifies to $e(\lambda, \mathbf{x}) = e(\mathbf{x})$. Moreover, we distinguish between an isotropic light source which radiates in all directions with the same power spectral density, and a directed light source where the power spectrum also depends on the relative location \mathbf{x} .

Light interacts with an object by partly being reflected and partly being absorbed. A material property called the surface albedo determines which part of the spectrum of the light is absorbed and consequently defines the object color, e.g. a red object absorbs light rays of all wavelengths except those corresponding to the red color. When the intensity of the reflected light is independent of the viewing angle, i.e. the reflection is isotropic, we speak of Lambertian reflection. This is the case for matte materials such as fabric, unfinished wood and paper. In addition to Lambertian reflection, glossy materials also exhibit interface reflections, which introduce highlights on the object. In case of interface reflections, the incident light is reflected directly from the object surface without interacting with the albedo and therefore the spectral density of the reflected light beam is not affected. Various physics models have been developed to simulate material reflections - in this thesis we explore the photometric reflectance model based on the Kubelka-Munk theory [12], which models the power of the reflected light $E(\lambda, \mathbf{x})$ as:

$$E(\lambda, \mathbf{x}) = e(\lambda, \mathbf{x}) \left(\underbrace{(1 - \rho_f(\mathbf{x}))^2 R_{\infty}(\lambda, \mathbf{x})}_{\text{material reflection}} + \underbrace{\rho_f(\mathbf{x})}_{\text{interface reflection}} \right). \quad (1.1)$$

Here, \mathbf{x} denotes the spatial location on the image plane, λ the wavelength of

the light, $e(\lambda, \mathbf{x})$ the spectrum of the light source, R_∞ the material reflectivity and ρ_f the Fresnel reflectance coefficient.

Finally, the reflected light is captured by the camera sensors by integrating the energy of the photons over a certain bandwidth ω , spatial area and period of time. According to the trichromatic theory [13] three independent detectors, each tuned to a specific wavelength, are required to record the full color space observed by humans. In a camera, each pixel is recorded by capturing the intensity of the incident light at three wavelengths using three separate sensors, corresponding to the three cones in the retina of an eye. The RGB pixel intensities in a camera are therefore given by

$$f^c(\mathbf{x}) = \int_{\omega} E(\lambda, \mathbf{x}) \rho^c(\lambda) d\lambda, \quad (1.2)$$

with $c \in R, G, B$ representing the red, green or blue color channel with corresponding sensor spectral sensitivity $\rho^c(\lambda)$, and $E(\lambda, \mathbf{x})$ the spectrum of reflected light as modeled by a reflection model such as Eq. (1.1). The photometric image formation process according to the Kubelka-Munk model is illustrated in Fig. 1.2.

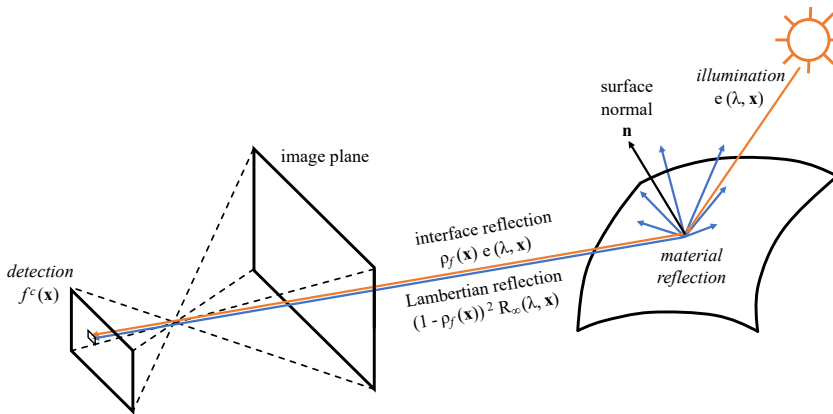


Figure 1.2: The photometric image formation process according to the Kubelka-Munk model for material reflections. Incident light from a light source e is reflected from an object through interface and/or material reflection and is detected by the camera sensor. Source: adapted from [14].

Color invariants From Eq. (1.2) it is clear that the resulting pixel values are dependent on lighting, and thus the trick is to derive a representation where one or more illumination variables no longer play a role. Indeed, a representation that is independent of illumination variables cannot be influenced by accidental lighting conditions. Such representation is called a *color invariant*. Geusebroek et al. [10] derived several color invariant representations by making some simplifying assumptions in the Kubelka-Munk reflection model from Eq. (1.1). For example, assuming that the light source $e(\lambda, \mathbf{x})$ is spectrally and spatially uniform, it can be represented by a constant e . Moreover, assuming only matte surfaces, i.e. $\rho_f(\mathbf{x}) = 0$, Eq. (1.1) reduces to

$$E(\lambda, \mathbf{x}) = eR_\infty(\lambda, \mathbf{x}). \quad (1.3)$$

Denoting the partial derivative $\partial E/\partial \mathbf{x}$ by E_x (omitting (λ, \mathbf{x})), the ratio $W_x = E_x/E$ is then independent of the illuminant e :

$$W_x = \frac{E_x}{E} = \frac{1}{R_\infty(\lambda, \mathbf{x})} \frac{\partial R_\infty(\lambda, \mathbf{x})}{\partial \mathbf{x}} \quad (1.4)$$

The same holds for the ratios

$$W_{\lambda x} = \frac{E_{\lambda x}}{E}, \quad W_{\lambda \lambda x} = \frac{E_{\lambda \lambda x}}{E},$$

where

$$E_{\lambda x} = \frac{\partial^2 E}{(\partial \lambda \partial \mathbf{x})} \quad \text{and} \quad E_{\lambda \lambda x} = \frac{\partial^3 E}{(\partial^2 \lambda \partial \mathbf{x})}.$$

This results in the color invariant W , which is defined as

$$W = \sqrt{W_x^2 + W_{\lambda x}^2 + W_{\lambda \lambda x}^2 + W_y^2 + W_{\lambda y}^2 + W_{\lambda \lambda y}^2}. \quad (1.5)$$

W is thus invariant to *illumination intensity*, as the intensity term e is canceled out. For the derivation of other Kubelka-Munk based color invariants we refer the interested reader to [10] or the supplementary material of [15] in Section 2.A.

Relation to this thesis Color invariants are a well-proven method [10, 11, 14] in classical computer vision for obtaining feature descriptors that remain stable under illumination changes. This thesis will investigate the use of color invariants derived from the Kubelka-Munk model for material reflections to improve the illumination robustness of deep learning architectures.

1.3 DEEP LEARNING

Deep learning is a subset of machine learning, where the mapping function f between an input x and an output y is modeled by an artificial neural network, consisting of multiple interconnected layers of neurons. In its most basic form, an artificial neural network (ANN) consists of an input layer, several hidden layers, and an output layer, where the neurons of each neighboring layer are linked through weighted connections. The input layer takes the input data, which in the case of computer vision constitutes of pixel values, and the number of input neurons is equal to the dimensionality of the input data. The number of hidden layers and neurons are design choices, while the number of output neurons is set as required by the task. Inside each neuron, a weighted sum of all incoming values from the previous layer is calculated and is offset with a bias term. Thus, the activation of neuron j in a specific layer is computed as

$$a_j = \sigma \left(\sum_{i=1}^n x_i w_{ij} + b_j \right). \quad (1.6)$$

Here, n denotes the number of neurons in the previous layer, x_i the activation of neuron i in the previous layer, w and b represent the trainable weight and bias parameters, respectively, and σ is an activation function, which allows the neural network to represent non-linear functions. Popular activation functions for hidden layers include the Rectified Linear Unit (ReLU) [16] and its variations [17, 18], while softmax is used to normalize the outputs of a neural network to represent a probability distribution.

Training a neural network involves finding the optimal set of parameters that minimize a task-specific loss averaged over the training set D_{train} . Typical loss functions include the cross-entropy loss for classification and the mean-squared-error loss for regression. Training is performed by computing the gradient of the loss with respect to all weights and biases, and updating the parameters in the direction of the negative gradient. This process is known as backpropagation [19], as the error in the output is propagated back through the network. During a single iteration of training, the loss and gradients are averaged over a random subset of D_{train} called a mini-batch. Samples are drawn without replacement and having drawn all samples concludes one epoch, after which the next epoch is started. This process is repeated for multiple rounds until the loss no longer decreases, i.e. the network has converged.

Convolutional Neural Networks The layers discussed so far are fully-connected, meaning that all neurons in layer l have connections to all neurons in layer $l + 1$. As such, each pixel location is treated as a separate input feature. Consequently, when a network is trained to recognize an object in the top left corner of an image, it will fail to recognize the same object when it appears in the bottom right corner of a test image. In other words, the network is unable to generalize over locations in an image. This is often referred to as *overfitting* to the training set. Deep learning often involves the skillful art of finding and incorporating an inductive bias into the learning model that correctly fits the data and reduces overfitting.

Convolutional Neural Networks [20] (CNNs) incorporate a spatial inductive bias into the network architecture by means of parameter sharing over image locations. Instead of treating each pixel as a separate feature, CNNs compute the dot product between an input patch and a spatial kernel with learnable weights that is shifted over all image locations. This operation is called convolution (although cross-correlation is more technically correct), and the resulting output is a matrix of feature activations which is referred to as a feature map.

1.4 EQUIVARIANCE IN NEURAL NETWORKS

A key property of CNNs is that they are, up to border effects, translation *equivariant*: shifting the input image to a convolutional layer results in an equally translated feature map [21]. The information in a feature map can be pooled to a single scalar value by computing the mean or max over the spatial locations. This results in a translation *invariant* representation: translating the input image does not change its feature representation. Through this property CNNs are able to generalize over image locations, even if not all locations are equally well represented in the training data.

CNNs are equivariant to translations, but not to other input transformations such as rotation and scaling. Motivated by the promise of improved data and computational efficiency, in recent years a relatively novel line of research has emerged focusing on investigating and extending equivariance in CNNs to additional transformations. The seminal work of Cohen et al. [22] introduced the elegant theoretical framework of Group Equivariant Convolutions (GConvs), which allows the incorporation of equivariance to other input transformations in CNNs. While the original work only considered translations, discrete rotations of multiples of 90 degrees, and horizontal and vertical flips, the framework can be used to implement equivariance to any *transformation*

group. In this section we first formally define equivariance. As GConvs are based on mathematical foundations in Group theory, we will then provide a minimal introduction within the context of symmetry groups and CNNs.

Formal definition A CNN layer Φ is equivariant to a transformation T if transforming the input x by T results in an equally transformed feature map. In other words, first performing a transformation and then the mapping is equivalent to first performing the mapping and then the transformation. Formally, equivariance is defined as

$$\Phi(T(x)) = T'(\Phi(x)). \quad (1.7)$$

T and T' can be identical transformations, 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. If $T \neq T'$, we formally speak of *covariance*, but in practice the term equivariance is used more broadly. A special case of equivariance is *invariance*, where T' is the identity mapping and the input transformation leaves the feature map unchanged:

$$\Phi(T(x)) = \Phi(x). \quad (1.8)$$

Such transformation T is also called a *symmetry transformation*. Note that there are actually two scenarios in which Eq. (1.8) holds. The symmetry can either denote a geometrical symmetry in the input image such that applying the transformation does not change the image, i.e. $T(x) = x$. This is the case when, for example, rotating an image of a circle. On the other hand, Eq. (1.8) can also be satisfied through a property of Φ that maps both inputs x and $T(x)$ to the same output. When speaking of equivariant CNNs we refer to the latter, e.g. for classification the semantic class y of an input image x does not change under transformation T , even if $T(x) \neq x$. Fig. 1.3 illustrates the concept of shift equivariance and invariance in a convolutional neural network.

Symmetry groups Rather than considering individual symmetry transformations T , it is more convenient to think of a collection of similar transformations as a whole. For instance, a classification model that is invariant to a single pixel translation $T = (0, 1)$ is not very useful, while invariance to the set of all 2D integer translations is very much so! This is where group theory comes into play.

A set of symmetries, together with a binary operation $\circ : G \times G \rightarrow G$ is called a group (G, \circ) if it satisfies the following conditions called the *group axioms*:

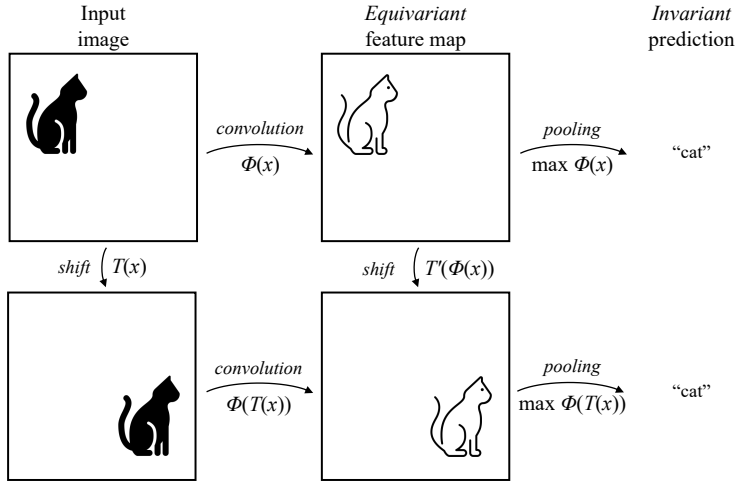


Figure 1.3: Equivariance and invariance in a convolutional neural network. Convolutions are translation *equivariant*: a shift in the input (top left to bottom left) results in an equally shifted feature map (top center to bottom center), regardless of the order in which the shift and convolution operations are applied. Applying a pooling operation results in a shift *invariant* prediction (right).

1. *Closure*: any two group elements combined through the group operator \circ produce a third group element, i.e. $g \circ h \in G \forall g, h \in G$;
2. *Associativity*: $(g \circ h) \circ l = g \circ (h \circ l) \forall g, h, l \in G$;
3. *Identity*: there exists a unique identity element $e \in G$ satisfying $e \circ g = g \circ e = g \forall g \in G$;
4. *Inverse*: for each $g \in G$ there is a unique inverse $g^{-1} \in G$ such that $g \circ g^{-1} = g^{-1} \circ g = e$.

Often the group operator is defined as matrix multiplication and is omitted from notation. There are many symmetry groups that are relevant in the context of equivariant CNNs. Specifically in this thesis we will consider:

- the $(\mathbb{Z}^2, +)$ group of 2D integer translations;
- the $p4$ group of 2D integer translations and discrete rotations of multiples of 90 degrees;

- the $p4m$ group including the transformations in $p4$ and, additionally, horizontal and vertical flips;
- the $SO(3)$ group of all rotations around the origin of 3D Euclidean space.

Lastly, let us introduce the notion of subgroups. Let (G, \circ) and (H, \circ) be two groups. If $H \subset G$ and H satisfies all group axioms, then H is called a *subgroup* of G . For example, $p4$ is a subgroup of $p4m$.

Relation to this thesis These principles lay the foundations for Group Equivariant Convolutions [22], which a large part of this thesis is concerned with. A further introduction is provided in Chapter 3, where we propose Color Equivariant Convolutions. The subsequent chapters further investigate computational aspects and edge cases in GConvs that break exact equivariance.

1.5 DATA AUGMENTATION

Good training data is diverse and represents all possible real-world variations. Unfortunately, we often have to deal with imperfect training data containing various appearance biases [23–25] to which our model can and will overfit. As discussed previously, incorporating inductive biases in the model architecture is an effective way to generalize beyond our training set. An alternative approach to this is offered by *data augmentation*, where the diversity of training data is artificially increased by generating new samples from existing samples. Augmentation involves applying (a combination of) various geometric and photometric transformations, such as rotations, scaling, cropping, shearing, mirroring, and color jittering to training samples before feeding them into the model. Importantly, data augmentation is based on the prior knowledge that applying the transformation to an image does not affect its semantic meaning, e.g. the image of a cat that is flipped horizontally still represents a cat and therefore should map to the same class label. Examples of some popular augmentations are shown in Fig. 1.4.

Relation to this thesis The interplay between data augmentation and equivariant architectures exhibits interesting properties. For example, one would expect that a CNN would not benefit from applying translation augmentations to the input as the network is already equivariant to translations. Yet, this is in fact common practice and empirical results do show improvements in model

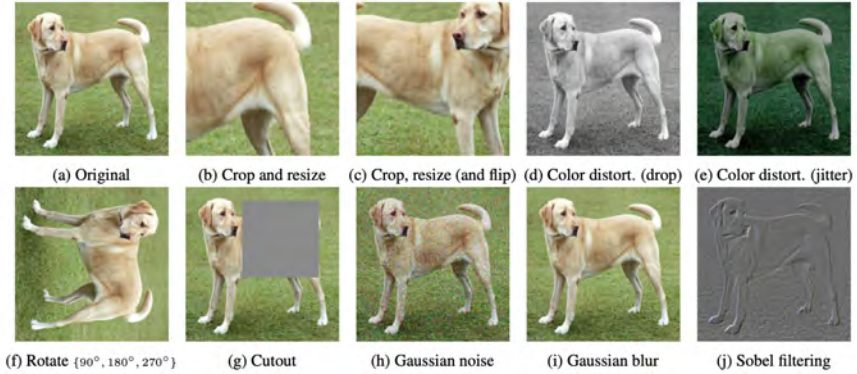


Figure 1.4: Examples of popular data augmentation transformations. Source: [26].

performance. It has been demonstrated that certain network characteristics, including border effects [21] and subsampling [27] break exact equivariance and introduce an equivariance error in the feature representation. Data augmentation has a regularizing effect on approximately equivariant networks and helps to reduce the equivariance error. A part of this thesis further investigates the relationship between equivariance and augmentation.

1.6 ORGANIZATION OF THIS THESIS

The remainder of this thesis is composed of the following original contributions:

Chapter 2

- *Based on:* A. Lengyel, S. Garg, M. Milford, and J. 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)*. Oct. 2021, pp. 4399–4409
- *Contribution of authors:*
 - A. Lengyel: all aspects
 - S. Garg: technical implementation
 - M. Milford: supervision and insights

- J. C. van Gemert: supervision and insights

This chapter investigates the use of physics-based color invariants in a deep learning setting for performing zero-shot day-to-night domain adaptation. Test-time variations in data can introduce unwanted domain shifts, hurting model performance. The popular domain adaptation setting is to train on one domain and adapt to the target domain by exploiting unlabeled data samples from the test set. As gathering relevant test data is expensive and sometimes even impossible, we remove any reliance on test data imagery and instead exploit a visual inductive prior derived from physics-based reflection models for domain adaptation. We cast a number of color invariant edge detectors as trainable layers in a convolutional neural network and evaluate their robustness to illumination changes. We show that the color invariant layer reduces the day-night distribution shift in feature map activations throughout the network. We demonstrate improved performance for zero-shot day to night domain adaptation on both synthetic as well as natural datasets in various tasks, including classification, segmentation and place recognition.

Chapter 3

- *Based on:* A. Lengyel, O. Strafforello, R.-J. Bruintjes, A. Gielisse, and J. van Gemert. “Color Equivariant Convolutional Networks”. In: *Advances in Neural Information Processing Systems*. Vol. 36. 2023, pp. 29831–29850
- *Contribution of authors:*
 - A. Lengyel: all aspects
 - O. Strafforello: technical implementation and insights
 - R. Bruintjes: technical implementation and insights
 - A. Gielisse: technical implementation and insights
 - J. C. van Gemert: supervision and insights

While color invariance improves robustness to color variations, it does so at the cost of removing color information, which sacrifices discriminative power. Therefore, in this chapter we introduce Color Equivariant Convolutions (CEConvs), a novel deep learning building block that enables sharing shape features 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 using the framework of Group Equivariant Convolutions. 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.

Chapter 4

- *Based on:* A. Lengyel and J. C. van Gemert. “Exploiting Learned Symmetries in Group Equivariant Convolutions”. In: *2021 IEEE International Conference on Image Processing (ICIP)*. 2021, pp. 759–763. DOI: 10.1109/ICIP42928.2021.9506362
- *Contribution of authors:*
 - A. Lengyel: all aspects
 - J. C. van Gemert: supervision and insights

This chapter studies Group Equivariant Convolutions (GConvs) from a compute efficiency perspective. GConvs enable convolutional neural networks to be equivariant to various transformation groups, but at the cost of using additional parameters and compute. We investigate redundancies in the filter parameters learned by GConvs and show that they can be efficiently decomposed into depthwise separable convolutions while preserving equivariance properties, and demonstrate improved performance and data efficiency.

Chapter 5

- *Based on:* T. Edixhoven, A. Lengyel, and J. C. van Gemert. “Using and Abusing Equivariance”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*. Oct. 2023, pp. 119–128
- *Contribution of authors:*
 - T. Edixhoven: all aspects
 - A. Lengyel: technical implementation, supervision and insights
 - J. C. van Gemert: supervision and insights

Lastly, we investigate the practical implications of the subsampling operation in Group Equivariant Convolutional Neural Networks, and derive conditions under which theoretical equivariance no longer holds. We show that subsampling not only breaks equivariance, but that Group Equivariant Convolutions actively exploit inexact equivariance by becoming less equivariant during

training. We find that this is a double edged sword: on one hand, approximate equivariance results in worse generalization to unseen transformations compared to exact equivariance, while on the other hand, approximate equivariance allows the network to relax equivariance constraints when beneficial, improving performance on datasets without symmetries.

Other publications Additional papers published during the research that are not integral to this thesis can be found in the List of publications on 141.

REFERENCES

- [1] J. Schulman, B. Zoph, C. Kim, J. Hilton, J. Menick, J. Weng, J. F. C. Uribe, L. Fedus, L. Metz, M. Pokorny, R. G. Lopes, S. Zhao, A. Vijayvergiya, E. Sigler, A. Perelman, C. Voss, and M. Heato. Nov. 2022. URL: <https://openai.com/blog/chatgpt>.
- [2] A. Ramesh, M. Pavlov, G. Goh, S. Gray, C. Voss, A. Radford, M. Chen, and I. Sutskever. “Zero-Shot Text-to-Image Generation”. In: *Proceedings of the 38th International Conference on Machine Learning*. Ed. by M. Meila and T. Zhang. Vol. 139. Proceedings of Machine Learning Research. PMLR, July 2021, pp. 8821–8831.
- [3] D. A. Patterson, J. Gonzalez, Q. V. Le, C. Liang, L. Munguia, D. Rothchild, D. R. So, M. Texier, and J. Dean. “Carbon Emissions and Large Neural Network Training”. In: *CoRR* abs/2104.10350 (2021). arXiv: 2104.10350. URL: <https://arxiv.org/abs/2104.10350>.
- [4] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer. “High-Resolution Image Synthesis With Latent Diffusion Models”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2022, pp. 10684–10695.
- [5] R. Rombach, P. Esser, and D. Ha. *Stable Diffusion v2 Model Card*. URL: <https://huggingface.co/stabilityai/stable-diffusion-2>.
- [6] J. Sevilla and P. Villalobos. *Parameter counts in machine learning - ai alignment forum*. June 2021. URL: <https://www.alignmentforum.org/posts/GzoWcYibWYwJva8aL/parameter-counts-in-machine-learning>.
- [7] D. G. Lowe. “Distinctive Image Features from Scale-Invariant Keypoints”. In: *Int. J. Comput. Vision* 60.2 (Nov. 2004), pp. 91–110. ISSN: 0920-5691. DOI: 10.1023/B:VISI.0000029664.99615.94. URL: <http://dx.doi.org/10.1023/B:VISI.0000029664.99615.94>.
- [8] H. Bay, T. Tuytelaars, and L. Van Gool. “SURF: Speeded Up Robust Features”. In: *Computer Vision – ECCV 2006*. Ed. by A. Leonardis, H. Bischof, and A. Pinz. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 404–417. ISBN: 978-3-540-33833-8.
- [9] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski. “ORB: An efficient alternative to SIFT or SURF”. In: *2011 International Conference on Computer Vision*. 2011, pp. 2564–2571. DOI: 10.1109/ICCV.2011.6126544.
- [10] J. M. Geusebroek, R. van den Boomgaard, A. W. M. Smeulders, and H. Geerts. “Color Invariance”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 23.12 (2001), pp. 1338–1350.
- [11] T. Gevers and A. W. Smeulders. “Color-based object recognition”. In: *Pattern recognition* 32.3 (1999), pp. 453–464.

- [12] P. Kubelka and F. Munk. "Ein beitrage zur optik der farbanstriche". In: *Zeitung fur Technische Physik*. Vol. 12. 1999, p. 593.
- [13] T. Young. "[Bakerian Lecture] On the theory of light and colours". In: *The Royal Society Archives, London*. Nov. 1801, L&P/11/172. URL: https://making.science.royalsociety.org/items/l-and-p_11_172/paper-on-the-theory-of-light-and-colours-bakerian-lecture-by-thomas-young (visited on 07/06/2023).
- [14] 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. URL: <https://ivi.fnwi.uva.nl/isis/publications/2012/GeversSIST2012>.
- [15] A. Lengyel, S. Garg, M. Milford, and J. 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)*. Oct. 2021, pp. 4399–4409.
- [16] V. Nair and G. E. Hinton. "Rectified linear units improve restricted boltzmann machines". In: *ICML 2010*. 2010, pp. 807–814.
- [17] A. Maas, A. Hannun, and A. Ng. "Rectifier Nonlinearities Improve Neural Network Acoustic Models". In: *Proceedings of the International Conference on Machine Learning*. Atlanta, Georgia, 2013.
- [18] D. Hendrycks and K. Gimpel. "Gaussian Error Linear Units (GELUs)". In: *arXiv e-prints*, arXiv:1606.08415 (June 2016), arXiv:1606.08415. DOI: 10.48550/arXiv.1606.08415. arXiv: 1606.08415 [cs.LG].
- [19] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. <http://www.deeplearningbook.org>. MIT Press, 2016.
- [20] Y. Lecun, Y. Bengio, and G. Hinton. "Deep learning". English (US). In: *Nature* 521.7553 (May 2015), pp. 436–444. ISSN: 0028-0836. DOI: 10.1038/nature14539.
- [21] O. Kayhan and J. C. van Gemert. "On Translation Invariance in CNNs: Convolutional Layers can Exploit Absolute Spatial Location". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020.
- [22] T. S. Cohen and M. Welling. "Group Equivariant Convolutional Networks". In: *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48. ICML'16*. New York, NY, USA: JMLR.org, 2016, pp. 2990–2999.
- [23] M. Afifi and M. S. Brown. "What Else Can Fool Deep Learning? Addressing Color Constancy Errors on Deep Neural Network Performance". In: *2019 IEEE/CVF International Conference on Computer Vision (ICCV) (2019)*, pp. 243–252.
- [24] D. Dai and L. V. Gool. "Dark Model Adaptation: Semantic Image Segmentation from Daytime to Nighttime". In: *ITSC*. Nov. 2018, pp. 3819–3824. DOI: 10.1109/ITSC.2018.8569387.

- [25] M. Wulfmeier, A. Bewley, and I. Posner. “Addressing appearance change in outdoor robotics with adversarial domain adaptation”. In: *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2017, pp. 1551–1558. DOI: 10.1109/IROS.2017.8205961.
- [26] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton. “A Simple Framework for Contrastive Learning of Visual Representations”. In: *ICML’20*. JMLR.org, 2020.
- [27] J. Xu, H. Kim, T. Rainforth, and Y. W. Teh. “Group Equivariant Subsampling”. In: *Advances in Neural Information Processing Systems*. Ed. by A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan. 2021. URL: <https://openreview.net/forum?id=CtaDl9L0bIQ>.
- [28] A. Lengyel, O. Strafforello, R.-J. Brintjes, A. Gielisse, and J. van Gemert. “Color Equivariant Convolutional Networks”. In: *Advances in Neural Information Processing Systems*. Vol. 36. 2023, pp. 29831–29850.
- [29] A. Lengyel and J. C. van Gemert. “Exploiting Learned Symmetries in Group Equivariant Convolutions”. In: *2021 IEEE International Conference on Image Processing (ICIP)*. 2021, pp. 759–763. DOI: 10.1109/ICIP42928.2021.9506362.
- [30] T. Edixhoven, A. Lengyel, and J. C. van Gemert. “Using and Abusing Equivariance”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*. Oct. 2023, pp. 119–128.

2

ZERO-SHOT DAY-NIGHT DOMAIN ADAPTATION WITH A PHYSICS PRIOR

We explore the zero-shot setting for day-night domain adaptation. The traditional domain adaptation setting is to train on one domain and adapt to the target domain by exploiting unlabeled data samples from the test set. As gathering relevant test data is expensive and sometimes even impossible, we remove any reliance on test data imagery and instead exploit a visual inductive prior derived from physics-based reflection models for domain adaptation. We cast a number of color invariant edge detectors as trainable layers in a convolutional neural network and evaluate their robustness to illumination changes. We show that the color invariant layer reduces the day-night distribution shift in feature map activations throughout the network. We demonstrate improved performance for zero-shot day to night domain adaptation on both synthetic as well as natural datasets in various tasks, including classification, segmentation and place recognition.

This chapter has been published as:

A. Lengyel, S. Garg, M. Milford, and J. 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)*, Oct. 2021, pp. 4399–4409.

Code available at:

<https://github.com/Attila94/CICConv>

2.1 INTRODUCTION

Deep image recognition methods are sensitive to illumination shifts caused by accidental recording conditions such as camera viewpoint, light color, and illumination changes caused by time of day or weather [1–3], as for example a model trained with daylight data will not generalize to nighttime. Robustness to such recording conditions is essential for autonomous driving and other safety-critical computer vision applications. An illumination shift between train and test data is typically addressed by unsupervised domain adaptation [4–6] where the labeled training set is from one domain and the test set is from a different domain. The main assumption is that the test data is readily available and the challenge is how to make use of the unlabeled test data in an unsupervised setting to address the domain shift. However, adding test data is often non-trivial as it may be expensive and time consuming to obtain, and due to the long tail of the real world impossible to collect for all possible scenarios in advance.

Instead of adding more data, prior knowledge can be built in as a visual inductive bias. The champion of such a bias is the convolution operator added to a deep network, which yields a Convolutional Neural Network (CNN). The CNN is translation invariant, and thus saves a massive amount of data as the deep network no longer needs training samples at all possible locations. Here, we replace data by an inductive photometric bias. We introduce a novel zero-shot domain adaptation method for addressing day-night domain shifts, exploiting learnable photometric invariant features as a physics-based visual inductive prior. In contrast to unsupervised domain adaptation, our zero-shot method reduces the data dependency by removing any reliance on the availability of test data.

Illumination changes to the source domain induce a distribution shift of feature map activations throughout all layers of a CNN. This is shown as the baseline in the top row of Fig. 2.1, where the activations of a CNN trained on daytime data are shown for a ‘Normal’ (source) and ‘Darker’ (target) test set. Such a distribution shift, in turn, has a severe detrimental effect on the accuracy of the CNN [7]. Because the distribution shift is between the training data and unavailable test data, this shift cannot be addressed in a data-driven manner using, for example, variants of Batch Normalization [7, 8]. Instead, we normalize feature map activations in a data-free setting by exploiting photometric invariant features which are explicitly designed to tackle distribution shifts caused by illumination changes.

Photometric invariant features, or color invariants, represent object properties irrespective of the accidental recording conditions [9, 10], including 1)

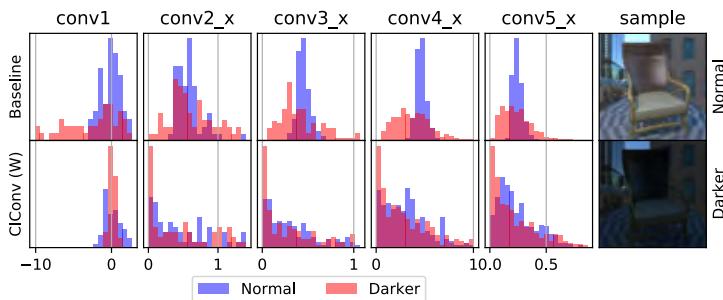


Figure 2.1: Feature map activations in various layers of a baseline ResNet-18 and a color invariant W -ResNet-18, averaged over all samples in a ‘Normal’ and ‘Darker’ test set (samples on right). The intensity change between the test sets causes an internal distribution shift throughout all layers of the baseline model. W normalizes the input, resulting in more domain invariant features.

scene geometry, which affects the formation of shadows and shading, the 2) color and 3) intensity of the light source, which changes the overall tint and brightness of the scene, and 4) Fresnel reflections occurring on shiny materials where the incoming light is directly reflected from the surface without interacting with the material color. Thanks to their robustness to these lighting changes, color invariants have been widely used in classical computer vision applications [11, 12], yet their use in a deep learning setting has remained largely unexplored. We implement the color invariant edge detectors from [9] as a trainable Color Invariant Convolution (CIConv) layer which can be used as the input layer to any CNN to transform the input to a domain invariant representation. Fig. 2.1, bottom row, shows that CIConv reduces the distribution shift between the source and target test set in all network layers, improving target domain performance.

We have the following contributions: (i) we introduce CIConv, a learnable color invariant CNN layer that reduces the activation distribution shift in a CNN under an illumination-based domain shift; (ii) we evaluate several color invariants in the day-night domain adaptation setting on our two carefully curated classification datasets; and (iii) we demonstrate performance improvements on tasks related to autonomous driving, including classification, segmentation and place recognition.

2.2 RELATED WORK

Domain Adaptation The aim of domain adaptation [6] is to train a model on a source domain dataset such that it performs well on a different but similar target domain dataset. This alleviates the burden of annotating datasets for applications in new domains where insufficient training data is available. Popular approaches rely on generative adversarial networks (GANs) to generate synthetic target domain samples [13], or aim to minimize the feature divergence between the two domains through an adversarial term [14, 15] or a discrepancy metric [16, 17] in the loss function. The day-night domain adaptation setting is particularly important due to the promise of self-driving cars and thus includes much work for semantic segmentation [2–5, 18–23], and for place recognition [24–26]. However, all aforementioned methods (except [18]) require either training data from the target domain or additional modalities, whereas our approach uses only source domain image data. Our approach requires no extra information sources and thus preempts expensive data gathering costs.

Zero-shot Domain Adaptation Research on zero-shot learning [27–32] has been readily extended from unseen classes to unseen domains, where domain adaptation is performed without having access to the target domain. However, current zero-shot domain adaptation methods require additional information in the form of: (i) extra task-irrelevant source and target domain data pairs to adapt to the task-relevant target domain [33, 34]; (ii) a parametrization of the domain shift by an attribute, where the attribute probability distribution for the unseen target domain is required to be known [35]; (iii) additional data from domains besides the source and target domain to learn a domain-invariant subspace projection [36]; or (iv) extra data in a partially labeled target domain [37]. These four types of information are generally not known for day-night domain shifts and are therefore not directly applicable. AdaBN [7] argues that domain-specific knowledge is stored in the batch normalization (BN) [8] layers of a model and performs domain adaptation by resampling BN statistics from the target domain. This again requires access to the target domain dataset. AdaBN [7] can be considered zero-shot if only the statistics of the current batch are used. However, this makes the method reliant on large batch sizes where classes are evenly represented. In contrast, our method does not require any information from the target domain other than the task agnostic physics-based illumination prior given by color invariants which are readily available from literature.

Physics-Guided Neural Networks Adding prior knowledge from physical models in a neural network has the potential to improve performance without additional training data. The canonical example is adding translation equivariance through a convolutional prior [38, 39] where recent work shows benefits from adding prior knowledge, for example in line detection [40], spectral leakage [41] and anti-aliasing in CNNs [42]. In the case of physical image formation models, recent examples include intrinsic image decomposition [43], underwater image enhancement [44], or rain image restoration [45]. Here, we add a physical image formation prior to compensate for the lack of data in zero-shot domain adaptation. We investigate a relatively unexplored direction combining deep learning with physical color and reflection invariants.

Color invariants The use of physics-based reflection models to improve invariance to illumination changes is a well-researched topic in classical computer vision [10, 46–51]. Early work includes invariants derived from the Kubelka-Munk (KM) reflection model [9, 52]. Based on the image formation model introduced in [53] various methods have been proposed for shadow removal or intrinsic image decomposition [54, 55] with applications in place recognition [12, 56], road detection [11, 57–59] and street image segmentation [60]. Recent works have shown improved segmentation performance by applying a color invariant transformation as a preprocessing step [61–63] or using the ground truth albedo as input on a synthetic dataset [64]. [1] demonstrates the sensitivity of CNNs to changes in white balance (WB) settings and shows how robustness can be improved using an auto-WB preprocessing step. Our work further explores the use of classical color invariants as a trainable deep network layer.

2.3 METHOD

Our color invariant layers make use of the invariant edge detectors from [9]. The edge detectors are derived from the image formation model based on the Kubelka-Munk theory [52] for material reflections, which describes the spectrum of light E reflected from an object in the viewing direction as

$$E(\lambda, \mathbf{x}) = e(\lambda, \mathbf{x}) \left((1 - \rho_f(\mathbf{x}))^2 R_\infty(\lambda, \mathbf{x}) + \rho_f(\mathbf{x}) \right) \quad (2.1)$$

where \mathbf{x} denotes the spatial location on the image plane, λ the wavelength of the light, $e(\lambda, \mathbf{x})$ the spectrum of the light source, R_∞ the material reflectivity and ρ_f the Fresnel reflectance coefficient. Partial derivatives of E with respect

to x and λ are denoted by subscripts E_x and E_λ , respectively.

A color invariant representation does not rely on accidental scene properties such as lighting and viewing direction, and depends only on the material property R_∞ . By exploring simplifying assumptions in Eq. (2.1), we can derive various invariant representations, as summarized in Table 2.1. The derived invariants E , W , C , N and H represent edge detectors that are invariant to various combinations of illumination changes, including scene geometry (i.e. does not detect shadow and shading edges), Fresnel reflections, and the intensity and color of the illuminant. For the complete derivations of the color invariants in Table 2.1, we refer to Section 2.A.

Invariant	Definition	SG	FR	II	IC
E	$E = \sqrt{E_x^2 + E_\lambda^2 + E_{\lambda\lambda x}^2 + E_y^2 + E_{\lambda y}^2 + E_{\lambda\lambda y}^2}$	×	×	×	×
W	$W = \sqrt{W_x^2 + W_\lambda^2 + W_{\lambda\lambda x}^2 + W_y^2 + W_{\lambda y}^2 + W_{\lambda\lambda y}^2}$ $W_x = \frac{E_x}{E}, W_\lambda = \frac{E_\lambda}{E}, W_{\lambda\lambda x} = \frac{E_{\lambda\lambda x}}{E}$	×	×	✓	×
C	$C = \sqrt{C_{\lambda x}^2 + C_{\lambda\lambda x}^2 + C_{\lambda y}^2 + C_{\lambda\lambda y}^2}$ $C_{\lambda x} = \frac{E_{\lambda x}E - E_\lambda E_x}{E^2}, C_{\lambda\lambda x} = \frac{E_{\lambda\lambda x}E - E_{\lambda\lambda}E_x}{E^2}$	✓	×	✓	×
N	$N = \sqrt{N_{\lambda x}^2 + N_{\lambda\lambda x}^2 + N_{\lambda y}^2 + N_{\lambda\lambda y}^2}$ $N_{\lambda x} = \frac{E_{\lambda x}E - E_\lambda E_x}{E^2}, N_{\lambda\lambda x} = \frac{E_{\lambda\lambda x}E^2 - E_{\lambda\lambda}E_xE - 2E_{\lambda x}E_\lambda E + 2E_\lambda^2 E_x}{E^3}$	✓	×	✓	✓
H	$H = \sqrt{H_x^2 + H_y^2}, H_x = \frac{E_{\lambda\lambda}E_{\lambda x} - E_\lambda E_{\lambda\lambda x}}{E_\lambda^2 + E_{\lambda\lambda}^2}$	✓	✓	✓	×

Table 2.1: Overview of color invariant edge detectors [9] and their invariance properties to Scene Geometry, Fresnel Reflections, Illumination Intensity, and Illumination Color. E is a baseline intensity edge detector and is not invariant to any changes. Subscripts denote partial derivatives, where λ is the spectral derivative and x the spatial derivative of Eq. (2.1). Spatial derivatives for the y direction follow directly from the ones given for the x direction.

The Gaussian color model [9] is used to estimate E , E_λ and $E_{\lambda\lambda}$ from the RGB camera responses as

$$\begin{bmatrix} E(x, y) \\ E_\lambda(x, y) \\ E_{\lambda\lambda}(x, y) \end{bmatrix} = \begin{bmatrix} 0.06 & 0.63 & 0.27 \\ 0.3 & 0.04 & -0.35 \\ 0.34 & -0.6 & 0.17 \end{bmatrix} \begin{bmatrix} R(x, y) \\ G(x, y) \\ B(x, y) \end{bmatrix} \quad (2.2)$$

where x, y are pixel locations in the image. Spatial derivatives E_x and E_y are

calculated by convolving E with a Gaussian derivative kernel g with standard deviation σ , i.e.

$$E_x(x, y, \sigma) = \sum_{i, j \in \mathbb{Z}} E(i, j) \frac{\partial g(x-i, y-j, \sigma)}{\partial x} \quad (2.3)$$

and similarly for $E_y, E_{\lambda x}, E_{\lambda \lambda x}, E_{\lambda y}$ and $E_{\lambda \lambda y}$. Finally, the color invariant edge map is defined as the gradient magnitude of all relevant spatial derivatives as shown in Table 2.1.

The σ parameter in Eq. (2.3) determines the scale at which the image is convolved with the Gaussian derivative filters and as such the amount of detail preserved in the color invariant representation of an image. A small σ results in a detailed edge map but is more sensitive to noise, whereas a large σ is more robust but may omit important details. A visualization is given in Fig. 2.2 for color invariant W .

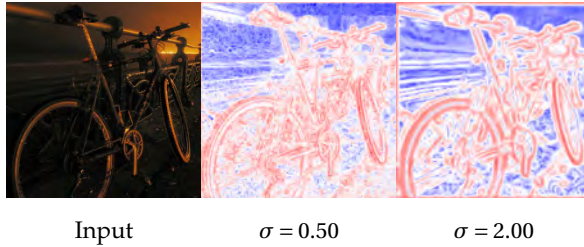


Figure 2.2: Color invariant representation W of the input image for two different values of σ . Note the trade-off between detail (small σ) and noise robustness (large σ).

Rather than fixing σ a-priori we implement the edge detector as a trainable layer to learn the task-specific optimal scale. The resulting Color Invariant Convolution (CICConv) is used as the input layer of the CNN and outputs a single-channel representation onto which subsequent convolutional layers can be stacked. For computational simplicity we omit the square root from the gradient magnitude of the color invariants, and apply a log transformation and sample-wise normalization such that the distribution of the edge maps is close to standard normal. Furthermore, instead of directly optimizing σ , we train a scale parameter s such that $\sigma = 2^s$. This stabilizes training by reducing the backpropagation gradient for small values of s and ensures that σ is always

positive. CConv is thus defined as

$$\text{CConv}(x, y) = \frac{\log(\text{CI}^2(x, y, \sigma = 2^s) + \epsilon) - \mu_{\mathcal{S}}}{\sigma_{\mathcal{S}}}, \quad (2.4)$$

with CI the color invariant of choice from Table 2.1, $\mu_{\mathcal{S}}$ and $\sigma_{\mathcal{S}}$ the sample mean and standard deviation over $\log(\text{CI}^2 + \epsilon)$, and ϵ a small term added for numerical stability.

2.4 EXPERIMENTS

2.4.1 ILLUMINATION ROBUSTNESS OF CNNs

We investigate to what degree CConv improves a CNN’s robustness to accidental recording conditions by performing a classification experiment on a synthetic image dataset where we have accurate control over the illumination of the scene. The images are rendered from a subset of the ShapeNet [65] dataset using the physically based renderer Mitsuba [66]. The scene is illuminated by a point light modeled as a black-body radiator with temperatures ranging between $[1900, 20000]K$ and an ambient light source. The training set contains 1,000 samples for each of the 10 object classes recorded under “normal” lighting conditions ($T = 6500K$). Multiple test sets with 300 samples per class are rendered for a variety of light source intensities and colors. Fig. 2.3 shows an overview of the illumination conditions represented in the test set.

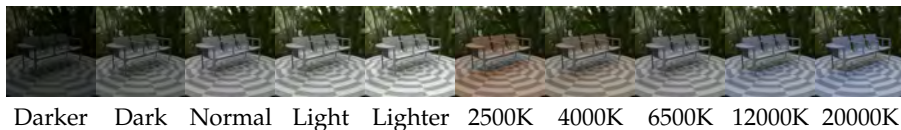


Figure 2.3: Sample from the synthetic classification dataset rendered from ShapeNet [65], shown in all illumination conditions represented in the test set. The five leftmost samples correspond to a varying light source intensity, whereas in the five rightmost samples a range of light source temperatures is shown. “Normal” and “6500K” are the same.

CConv improves illumination robustness We train a baseline ResNet-18 [67] and five models with the CConv layer with invariants E , W , C , N and

H , respectively. Training is done for 175 epochs with a batch size of 64 using SGD with momentum 0.9, weight decay $1e-4$ and an initial learning rate of 0.05 with stepwise reduction by factor 0.1, step size 50. Data augmentation is performed in the form of random horizontal flips, random cropping and random rotations. The models are evaluated on both test sets and the average classification accuracy over three runs is shown in Fig. 2.4. The accuracy of the baseline RGB model quickly drops as lighting conditions start to diverge from the training set. The performance of the color invariant networks remains more stable with W consistently outperforming all others.

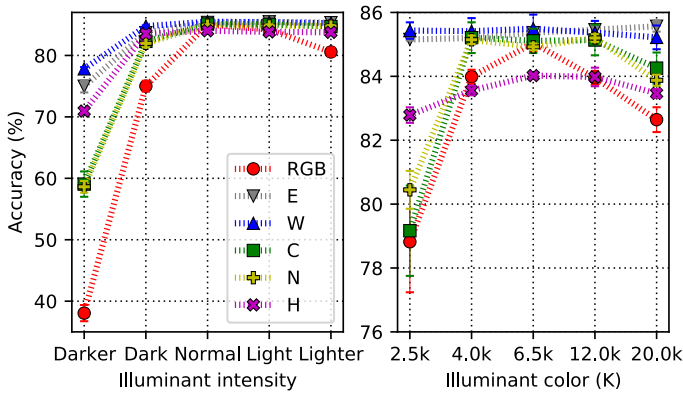


Figure 2.4: Classification accuracy of ResNet-18 with various color invariants on the synthetic ShapeNet dataset. RGB (not invariant) performance degrades when illumination conditions differ between train and test set, while color invariants remain more stable. W performs best overall.

CICov reduces feature map distribution shift The robustness of the color invariant networks compared to the baseline can be explained by analyzing the feature map activations of the networks. We calculate the mean feature map activation in different layers of the networks, averaged over all samples in the Normal and Dark test sets. The histograms in Fig. 2.1 show that the intensity change between the normal and low-light test sets caused a clear distribution shift throughout all network layers of the baseline model. In contrast, the CICov layer with invariant W produces a domain invariant feature representation and consequently the distributions in the network are more aligned between the two domains. We quantify the distribution shift as the L2 distance between feature maps for the two domains, where again

W yields the smallest distance. The L2 distances as well as histograms of the distributions of feature map activations for other color invariants are provided in Section 2.B.

2

2.4.2 DAY-NIGHT NATURAL IMAGE CLASSIFICATION

To verify that the properties of the color invariants also generalize to natural images we perform a classification experiment on a novel day-to-night dataset. We present the Common Objects Day and Night (CODaN) dataset, consisting of images from 10 common object classes recorded in both day and nighttime. It contains a daytime training set of 1,000 samples per class, a daytime validation set of 50 samples per class, and separate day and night test sets of 300 samples per class. CODaN is composed from the ImageNet [68], COCO [69] and ExDark [70] datasets. Samples of the day and night test sets are shown in Fig. 2.5.

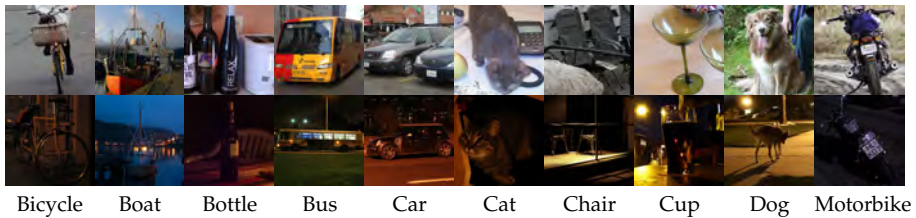


Figure 2.5: Samples from the day (source domain) and night (target domain) test sets of the CODaN dataset.

Performance on natural images We trained color invariant versions of ResNet-18 on CODaN using the same settings as in Section 2.4.1, but without random cropping and with random brightness, contrast, hue and saturation augmentations. Table 2.2 shows the accuracy of the baseline and the color invariant networks, averaged over three runs. Additionally, other color invariants (luminance, normalized RGB, comprehensive normalization [71] and others [11, 12]) are evaluated, which are implemented as a preprocessing step. We also consider a slightly adjusted version of AdaBN as a possible zero-shot domain adaptation method, which provides a significant performance increase by sampling the batch statistics for the Batch Normalization layers during test time for each individual batch. This is opposed to the original AdaBN method, where the batch statistics are calculated from the target domain dataset a

priori. W outperforms all other models on the nighttime test set by a large margin. The luminance baseline performs surprisingly well, whereas the other non-trainable color invariants even result in a performance drop.

Method	Day	Night
Baseline	80.39 ± 0.38	48.31 ± 1.33
E	79.79 ± 0.40	49.95 ± 1.60
W	81.49 ± 0.49	59.67 ± 0.93
C	78.04 ± 1.08	53.44 ± 1.28
N	77.44 ± 0.00	52.03 ± 0.27
H	75.20 ± 0.56	50.52 ± 1.34
Luminance	80.67 ± 0.32	51.37 ± 0.58
Normalized RGB	63.44 ± 1.52	41.66 ± 1.56
Comprehensive norm. [71]	70.52 ± 1.10	44.34 ± 1.57
Alvarez and Lopez [11]	64.41 ± 0.74	30.06 ± 0.57
Maddern et al. [12]	60.83 ± 0.98	33.04 ± 1.28
AdaBN [7]	79.72 ± 0.59	55.55 ± 1.07
Ablations	Day	Night
Baseline + norm.	63.43 ± 1.32	42.15 ± 0.98
Baseline + log + norm.	63.49 ± 0.55	41.90 ± 0.69
Baseline w/o color aug.	78.99 ± 0.59	36.00 ± 0.59
W w/o color aug.	79.71 ± 0.57	53.62 ± 0.88

Table 2.2: CODaN classification accuracy of a ResNet-18 architecture with various color invariants (top). W performs best. Ablation studies (bottom) show the individual effect of normalization, log scaling and photometric augmentations.

Color invariant transformations on natural images We visualize the E , W , C , N and H color invariant transformations of a day and night test sample (RGB) in Fig. 2.6. E being a non-invariant edge detector has low edge strengths in low intensity parts of the dark image. W on the other hand normalizes for intensity, yielding a more constant edge map. C , N and H are invariant to changes in scene geometry and therefore do not detect edges with low color saturation, resulting in significant information loss. In addition, these invariants seem to be more amplifying the noise in low intensity parts of the image. Overall, W is able to 1) detect low intensity and low saturation edges and 2) suppress noise in low-intensity parts of the image, and therefore produces the most robust and informative edge map.

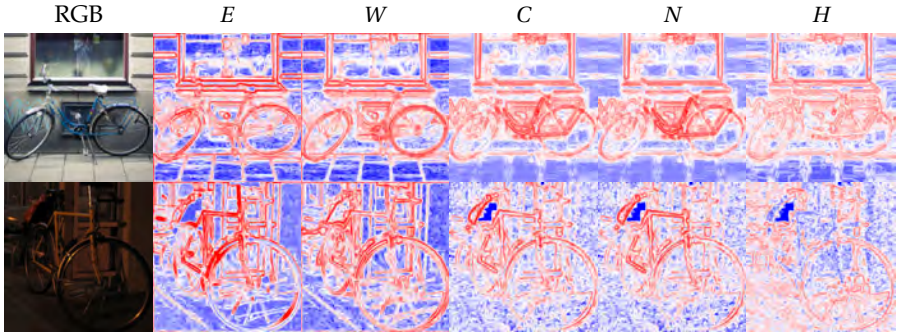


Figure 2.6: Color invariant visualizations of day and night samples from CODaN (red: positive; blue: negative values). E does not detect low intensity edges, whereas C , N and H do not detect edges that have low color saturation. W produces the most robust and informative edge map.

Learned vs. fixed scale We verify that CIconv learns the optimal scale by training the model with a range of fixed σ values, using invariant W . Fig. 2.7 shows the average accuracy over five runs. We observe that selecting the wrong scale σ has a detrimental effect on accuracy. When the scale is learnable, it converges to the optimal value for the daytime dataset, as indicated by the red cross in the figure. This value proves also optimal for the nighttime domain.

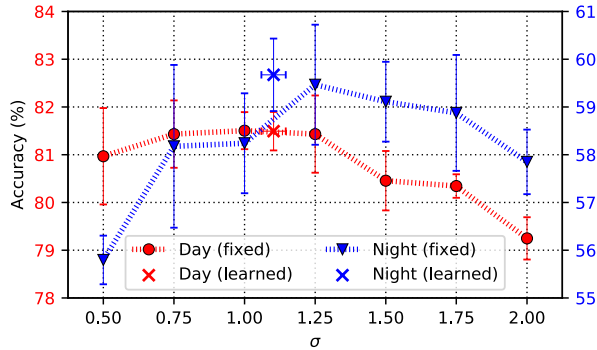


Figure 2.7: Performance on CODaN day (left y-axis) and night (right y-axis) test sets for various fixed values of σ . Learned σ and corresponding accuracies are indicated by crosses. CIconv learns the optimal value.

Ablation studies We evaluate whether simple log scaling and sample-wise normalization of RGB images, without applying a color invariant transformation, can achieve the same improved performance on the nighttime test set. Furthermore, we investigate how the baseline and W networks perform when trained without brightness, contrast, hue and saturation augmentations. The results are shown in the bottom part of Table 2.2. Normalization, both with and without log scaling, does not yield better performance for the baseline model. This indicates that addressing the distribution shift between the source and target domain observed in the feature map activations of a network requires more than simple intensity normalization of the input sample. Moreover, photometric augmentations mostly seem to benefit the baseline network, whereas the model with color invariant W is inherently more robust to illumination changes. Both results underscore the importance and effectiveness of the color invariant transformation.

2.4.3 SEMANTIC SEGMENTATION

We perform a semantic segmentation experiment using the RefineNet [72] architecture with ResNet-101 and W -ResNet-101 feature extractors pre-trained on the ImageNet [68] dataset. The segmentation model is trained on the training set of the CityScapes [73] dataset containing 2,975 densely annotated daytime street images and evaluated on the 50 coarsely annotated street images from Nighttime Driving [2] and the 151 densely annotated images from the Dark Zurich [20] test set. We perform training using SGD with momentum 0.9, weight decay $1e-4$ and an initial learning rate of 0.1 which is step-wise reduced by a factor 0.1 after every 30 epochs. All input images are resized to 1024×512 pixels and randomly cropped to 768×384 pixels, allowing a batch size of 6 on 2 GeForce GTX 1080 Ti GPUs. Data augmentation is applied by random scaling, brightness, contrast and hue shifting, and horizontal flipping. Inference is done on 1024×512 samples without cropping.

Results are shown in Table 2.3 as the mean Intersection-over-Union (mIoU). Results for other methods are taken from their corresponding papers. The color invariant W -RefineNet significantly outperforms the vanilla RefineNet and RefineNet-AdaBN models, which are also trained only on source domain data, and has competitive performance compared to methods trained on both source and target domain data. Qualitative segmentation results are shown in Fig. 2.8. Detailed per-class scores are included in Section 2.D.

Method	Nighttime Driving	Dark Zurich
Trained on source data only		
RefineNet [72]	34.1	30.6
<i>W</i> -RefineNet [ours]	41.6	34.5
RefineNet-AdaBN [7]	36.3	31.3
Trained on source and target data		
ADVENT [74]	34.7	29.7
BDL [75]	34.7	30.8
AdaptSegNet [76]	34.5	30.4
DMAda [2]	41.6	32.1
Day2Night [21]	45.1	-
GCMA [20]	45.6	42.0
MGCDa [5]	49.4	42.5

Table 2.3: Segmentation performance on Nighttime Driving [2] and Dark Zurich [20], reported as mIoU scores. *W*-RefineNet outperforms other methods trained only on daytime data and has competitive performance to methods also using nighttime images.

2.4.4 VISUAL PLACE RECOGNITION (VPR)

We present results for VPR task in two phases: first, we compare against a similar work for place recognition based on a learnable normalisation of images [25], and then we benchmark place representations based on color-invariant trained CNNs on an additional dataset, evaluation metric, and descriptor type to show broader applicability within VPR.

Learnable normalisation We use the Tokyo 24/7 [77] day-night place recognition dataset for this purpose, and follow the evaluation procedure described in [25]. To obtain place representations, the VGG Generalized Mean Pooling (GeM) [78] network is prepended with our CConv layer (*W*-VGG GeM) and trained on the Retrieval-SfM dataset as described in [78]. The train dataset contains query images as well as both positive and negative target images of places photographed in daytime conditions. The results are reported as the mean Average Precision (mAP) in Table 2.4. Results of competing methods are borrowed from Tables 1 and 2 in [25]. It can be observed that our method outperforms all models trained on daytime data only and achieves competitive

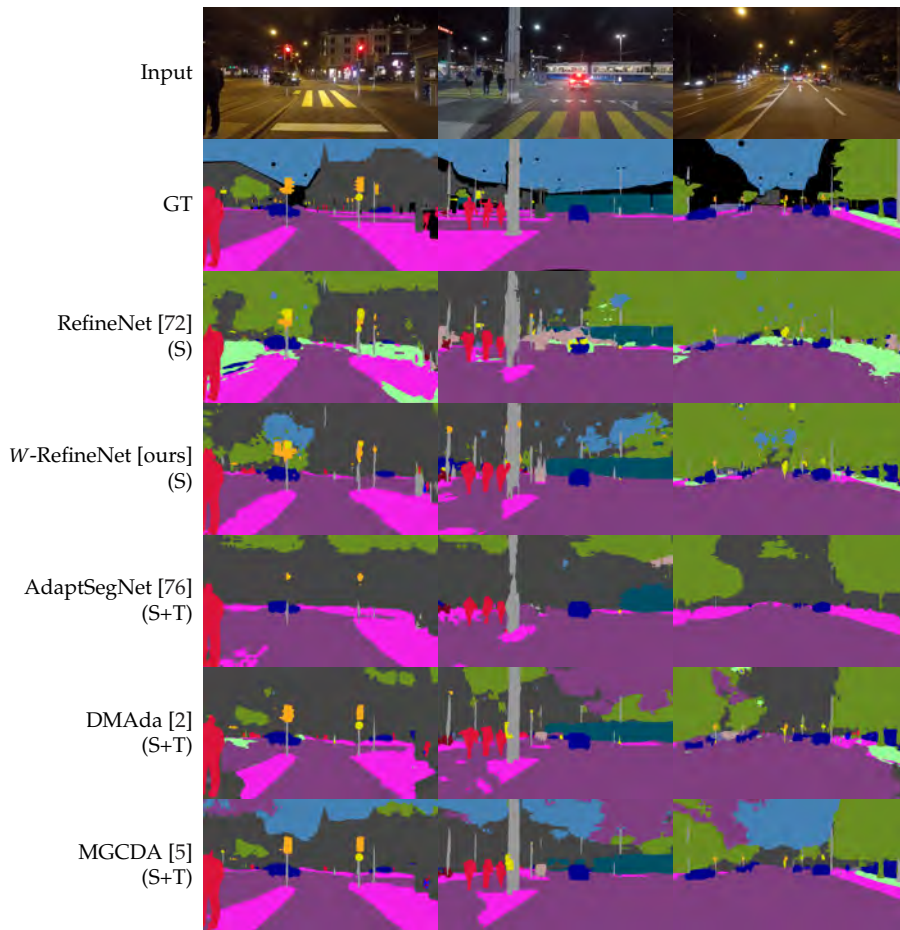


Figure 2.8: Qualitative semantic segmentation results on the Dark Zurich [20] dataset. S and T indicate whether the model was trained on the source or target domain, respectively.

results to the current state-of-the-art, which is an ensemble of two models trained on both daytime and nighttime data.

Broader VPR applicability Here, we use the two outdoor day-night datasets from VPRBench [81]: Gardens Point and Tokyo 24/7, where latter’s evaluation

Method	Tokyo 24/7 (mAP)
Trained on source data only	
VGG GeM [78]	79.4
<i>W</i> -VGG GeM [ours]	83.3
ResNet101 GeM [78]	85.0
<i>W</i> -ResNet101 GeM [ours]	88.3
EdgeMAC [79]	75.9
U-Net jointly [25]	79.8
CLAHE [80]	84.1
EdgeMAC + VGG GeM [25]	85.4
Trained on source and target data	
VGG GeM [78]	79.8
U-Net jointly [25]	86.5
CLAHE [80]	87.0
EdgeMAC + CLAHE [25]	90.5
EdgeMAC + U-Net jointly [25]	90.0

Table 2.4: Place recognition results on the Tokyo 24/7 dataset [77]. VGG GeM with our CIConv layer outperforms all other methods trained on daytime data. + denotes an ensemble of different models.

is similar to the previous experiment but using Recall@1 as the evaluation metric in this case for both the datasets. For the Gardens Point dataset, we consider two settings: Appearance only (A), with only day-night variations, and the more challenging Appearance + Viewpoint (A+V), where viewpoint is also laterally shifted. We consider three descriptor pooling types here using an ImageNet-trained ResNet-101 (R101) as the backbone network: Maximum Activations of Convolutions (MAC) [82], flattened tensor (Flat) [83] and GeM, where only GeM is further trained on image retrieval task as described in the previous subsection. For all three descriptor types, we compute results for training with and without the prepended color invariant layer. Additionally, we compare against state-of-the-art VPR methods: DenseVLAD [84] and AP-GeM [85].

In Table 2.4, it can be observed that *W*-R101 GeM achieves state-of-the-art results for all datasets. Furthermore, all methods based on color invariant perform better than their vanilla counterparts, including the Flat and MAC descriptors. This shows that color invariant networks provide robust place representation for different pooling types even without VPR-specific training.

Method	GP:A+V	GP:A	Tokyo 24/7
AP-GeM [85]	0.87	0.92	0.91
DenseVLAD [84]	0.81	0.89	0.89
R101 MAC [82]	0.51	0.56	0.20
R101 Flat [83]	0.56	0.68	0.84
R101 GeM [78]	0.90	0.96	0.91
W-R101 MAC [ours]	0.53	0.70	0.20
W-R101 Flat [ours]	0.61	0.91	0.85
W-R101 GeM [ours]	0.94	0.97	0.93

Table 2.5: Recall@1 for VPR using different feature pooling types on Gardens Point (GP) and Tokyo 24/7 dataset. Color-invariant layer (W) based networks outperform their vanilla counterparts with W-R101-GeM achieving state-of-the-art results.

2.5 DISCUSSION

The image formation model that lies at the foundation of the color invariants used in the CIconv layer is based on certain simplifying assumptions, such as purely matte reflections, non-transparent materials and a single, spatially uniform light source. Although most natural scenes do not satisfy these strict conditions, our results show that CNNs nevertheless do benefit from prior information derived from such approximate models. Moreover, current publicly available datasets, including the ones used in our experiments, are not appropriate for physics-based vision due to various artifacts introduced in post-processing steps (see Discussion in [63]). CIconv and other physics based methods can therefore only reach their full potential when sufficient attention is paid to preserving the physical correctness of the data during image capturing.

The robustness of color invariants to illumination changes comes at the loss of some discriminative power [9]. The CIconv layer transforms the input image into an edge map representation that is no longer sensitive to the intensity and color of the light source, but as a side effect also removes valuable color information. We found that naively concatenating color invariants with the RGB input degrades performance, see Section 2.C. Future research should therefore focus on implementing an adaptive mechanism for optimally combining color information and color invariant edge information.

Zero-shot domain adaptation is a promising method for reducing the data

dependency and the corresponding data collection and annotation costs in computer vision. We therefore hope that this paper inspires future research on integrating physics priors into neural networks.

REFERENCES

- [1] M. Afifi and M. S. Brown. “What Else Can Fool Deep Learning? Addressing Color Constancy Errors on Deep Neural Network Performance”. In: *2019 IEEE/CVF International Conference on Computer Vision (ICCV)* (2019), pp. 243–252.
- [2] D. Dai and L. V. Gool. “Dark Model Adaptation: Semantic Image Segmentation from Daytime to Nighttime”. In: *ITSC*. Nov. 2018, pp. 3819–3824. DOI: 10.1109/ITSC.2018.8569387.
- [3] M. Wulfmeier, A. Bewley, and I. Posner. “Addressing appearance change in outdoor robotics with adversarial domain adaptation”. In: *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. 2017, pp. 1551–1558. DOI: 10.1109/IROS.2017.8205961.
- [4] E. Romera, L. M. Bergasa, K. Yang, J. M. Álvarez, and R. Barea. “Bridging the Day and Night Domain Gap for Semantic Segmentation”. In: *2019 IEEE Intelligent Vehicles Symposium (IV)* (2019), pp. 1312–1318.
- [5] C. Sakaridis, D. Dai, and L. Van Gool. “Map-Guided Curriculum Domain Adaptation and Uncertainty-Aware Evaluation for Semantic Nighttime Image Segmentation”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2020). DOI: 10.1109/TPAMI.2020.3045882.
- [6] M. Wang and W. Deng. “Deep Visual Domain Adaptation: A Survey”. In: *Neurocomputing* 312 (2018), pp. 135–153.
- [7] Y. Li, N. Wang, J. Shi, J. Liu, and X. Hou. “Revisiting Batch Normalization For Practical Domain Adaptation”. In: *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Workshop Track Proceedings*. OpenReview.net, 2017. URL: <https://openreview.net/forum?id=Hk6dkJQFx>.
- [8] S. Ioffe and C. Szegedy. “Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift”. In: *CoRR* abs/1502.03167 (2015). arXiv: 1502.03167. URL: <http://arxiv.org/abs/1502.03167>.
- [9] J. M. Geusebroek, R. van den Boomgaard, A. W. M. Smeulders, and H. Geerts. “Color Invariance”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 23.12 (2001), pp. 1338–1350.
- [10] T. Gevers and A. W. Smeulders. “Color-based object recognition”. In: *Pattern recognition* 32.3 (1999), pp. 453–464.
- [11] J. M. A. Alvarez and A. M. Lopez. “Road Detection Based on Illuminant Invariance”. In: *IEEE Transactions on Intelligent Transportation Systems* 12.1 (Mar. 2011), pp. 184–193. ISSN: 1524-9050. DOI: 10.1109/TITS.2010.2076349.

- [12] W. Maddern, A. Stewart, C. McManus, B. Upcroft, W. Churchill, and P. Newman. "Illumination Invariant Imaging: Applications in Robust Vision-based Localisation, Mapping and Classification for Autonomous Vehicles". In: *Proceedings of the Visual Place Recognition in Changing Environments Workshop, IEEE International Conference on Robotics and Automation (ICRA)*. Hong Kong, China, May 2014.
- [13] J. Hoffman, E. Tzeng, T. Park, J.-Y. Zhu, P. Isola, K. Saenko, A. Efros, and T. Darrell. "CyCADA: Cycle-Consistent Adversarial Domain Adaptation". In: *Proceedings of the 35th International Conference on Machine Learning*. Ed. by J. Dy and A. Krause. Vol. 80. Proceedings of Machine Learning Research. Stockholmsmässan, Stockholm Sweden: PMLR, July 2018, pp. 1989–1998.
- [14] J. Hoffman, D. Wang, F. Yu, and T. Darrell. "FCNs in the Wild: Pixel-level Adversarial and Constraint-based Adaptation". In: *CoRR abs/1612.02649* (2016). arXiv: 1612.02649. URL: <http://arxiv.org/abs/1612.02649>.
- [15] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell. "Adversarial Discriminative Domain Adaptation". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. July 2017.
- [16] E. Tzeng, J. Hoffman, N. Zhang, K. Saenko, and T. Darrell. "Deep Domain Confusion: Maximizing for Domain Invariance". In: *CoRR abs/1412.3474* (2014). arXiv: 1412.3474. URL: <http://arxiv.org/abs/1412.3474>.
- [17] M. Long, Y. Cao, J. Wang, and M. Jordan. "Learning Transferable Features with Deep Adaptation Networks". In: *Proceedings of the 32nd International Conference on Machine Learning*. Ed. by F. Bach and D. Blei. Vol. 37. Proceedings of Machine Learning Research. Lille, France: PMLR, July 2015, pp. 97–105. URL: <https://proceedings.mlr.press/v37/long15.html>.
- [18] S. W. Cho, N. R. Baek, J. H. Koo, M. Arsalan, and K. R. Park. "Semantic Segmentation With Low Light Images by Modified CycleGAN-Based Image Enhancement". In: *IEEE Access* 8 (2020), pp. 93561–93585. DOI: 10.1109/ACCESS.2020.2994969.
- [19] S. Di, Q. Feng, C. Li, M. Zhang, H. Zhang, S. Elezovikj, C. C. Tan, and H. Ling. "Rainy Night Scene Understanding With Near Scene Semantic Adaptation". In: *IEEE Transactions on Intelligent Transportation Systems* (2020), pp. 1–9. DOI: 10.1109/TITS.2020.2972912.
- [20] C. Sakaridis, D. Dai, and L. V. 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)*. Oct. 2019.
- [21] L. Sun, K. Wang, K. Yang, and K. Xiang. "See clearer at night: towards robust nighttime semantic segmentation through day-night image conversion". In: *Artificial Intelligence and Machine Learning in Defense Applications*. Ed. by J. Dijk. Strasbourg, France: SPIE, Sept. 2019, p. 8. DOI: 10.1117/12.2532477. (Visited on 11/15/2019).

- [22] A. Valada, J. Vertens, A. Dhall, and W. Burgard. "AdapNet: Adaptive semantic segmentation in adverse environmental conditions". In: *2017 IEEE International Conference on Robotics and Automation (ICRA)*. 2017, pp. 4644–4651. DOI: 10.1109/ICRA.2017.7989540.
- [23] J. Vertens, J. Zürn, and W. Burgard. "Heatnet: Bridging the day-night domain gap in semantic segmentation with thermal images". In: *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE. 2020, pp. 8461–8468.
- [24] A. Anoosheh, T. Sattler, R. Timofte, M. Pollefeys, and L. V. Gool. "Night-to-Day Image Translation for Retrieval-based Localization". In: *2019 International Conference on Robotics and Automation (ICRA)* (2019), pp. 5958–5964.
- [25] T. Jeníček and O. Chum. "No Fear of the Dark: Image Retrieval Under Varying Illumination Conditions". In: *2019 IEEE/CVF International Conference on Computer Vision (ICCV)* (2019), pp. 9695–9703.
- [26] H. Porav, W. Maddern, and P. Newman. "Adversarial Training for Adverse Conditions: Robust Metric Localisation Using Appearance Transfer". In: *2018 IEEE International Conference on Robotics and Automation (ICRA)*. 2018, pp. 1011–1018. DOI: 10.1109/ICRA.2018.8462894.
- [27] Z. Akata, F. Perronnin, Z. Harchaoui, and C. Schmid. "Label-embedding for image classification". In: *IEEE transactions on pattern analysis and machine intelligence* 38.7 (2015), pp. 1425–1438.
- [28] C. H. Lampert, H. Nickisch, and S. Harmeling. "Attribute-based classification for zero-shot visual object categorization". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 36.3 (2013), pp. 453–465.
- [29] T. Mensink, E. Gavves, and C. G. Snoek. "Costa: Co-occurrence statistics for zero-shot classification". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2014, pp. 2441–2448.
- [30] M. Norouzi, T. Mikolov, S. Bengio, Y. Singer, J. Shlens, A. Frome, G. Corrado, and J. Dean. "Zero-shot learning by convex combination of semantic embeddings". English (US). In: *2nd International Conference on Learning Representations, ICLR 2014 ; Conference date: 14-04-2014 Through 16-04-2014*. Jan. 2014.
- [31] Y. Xian, C. H. Lampert, B. Schiele, and Z. Akata. "Zero-shot learning—a comprehensive evaluation of the good, the bad and the ugly". In: *IEEE transactions on pattern analysis and machine intelligence* (2018).
- [32] Z. Zhang and V. Saligrama. "Zero-Shot Learning via Joint Latent Similarity Embedding". In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2016.
- [33] K.-C. Peng, Z. Wu, and J. Ernst. "Zero-Shot Deep Domain Adaptation". In: *ECCV*. 2017.
- [34] J. Wang and J. Jiang. "Conditional Coupled Generative Adversarial Networks for Zero-Shot Domain Adaptation". In: *ICCV*. Oct. 2019.

- [35] M. Ishii, T. Takenouchi, and M. Sugiyama. “Zero-shot Domain Adaptation Based on Attribute Information”. In: *ACML*. 2019.
- [36] Y. Yang and T. M. Hospedales. “Zero-Shot Domain Adaptation via Kernel Regression on the Grassmannian”. In: *CoRR* abs/1507.07830 (2015). arXiv: 1507.07830. URL: <http://arxiv.org/abs/1507.07830>.
- [37] Q. Wang, P. Bu, and T. P. Breckon. “Unifying Unsupervised Domain Adaptation and Zero-Shot Visual Recognition”. In: *CoRR* abs/1903.10601 (2019). arXiv: 1903.10601. URL: <http://arxiv.org/abs/1903.10601>.
- [38] O. Kayhan and J. C. van Gemert. “On Translation Invariance in CNNs: Convolutional Layers can Exploit Absolute Spatial Location”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020.
- [39] G. Urban, K. J. Geras, S. E. Kahou, O. Aslan, S. Wang, A. Mohamed, M. Philipose, M. Richardson, and R. Caruana. “Do Deep Convolutional Nets Really Need to be Deep and Convolutional?” In: *ICLR*. 2016.
- [40] Y. Lin, S. L. Pintea, and J. C. van Gemert. “Deep Hough-Transform Line Priors”. In: *European Conference on Computer Vision*. Springer. 2020, pp. 323–340.
- [41] N. Tomen and J. C. van Gemert. “Spectral Leakage and Rethinking the Kernel Size in CNNs”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*. Oct. 2021, pp. 5138–5147.
- [42] R. Zhang. “Making Convolutional Networks Shift-Invariant Again”. In: *Proceedings of the 36th International Conference on Machine Learning, ICML*. Vol. 97. 2019, pp. 7324–7334. URL: <http://proceedings.mlr.press/v97/zhang19a.html>.
- [43] A. S. Baslamisli, H.-A. Le, and T. Gevers. “CNN Based Learning Using Reflection and Retinex Models for Intrinsic Image Decomposition”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2018.
- [44] Y. Zhou and K. Yan. “Domain Adaptive Adversarial Learning Based on Physics Model Feedback for Underwater Image Enhancement”. In: *ArXiv* abs/2002.09315 (2020).
- [45] R. Li, L.-F. Cheong, and R. T. Tan. “Heavy Rain Image Restoration: Integrating Physics Model and Conditional Adversarial Learning”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2019.
- [46] K. Barnard, G. Finlayson, and B. Funt. “Color constancy for scenes with varying illumination”. In: *Computer vision and image understanding* 65.2 (1997), pp. 311–321.
- [47] G. J. Burghouts and J.-M. Geusebroek. “Performance evaluation of local colour invariants”. In: *Computer Vision and Image Understanding* 113.1 (2009), pp. 48–62.
- [48] B. V. Funt and G. D. Finlayson. “Color constant color indexing”. In: *IEEE transactions on Pattern analysis and Machine Intelligence* 17.5 (1995), pp. 522–529.

- [49] K. Van De Sande, T. Gevers, and C. Snoek. "Evaluating color descriptors for object and scene recognition". In: *IEEE transactions on pattern analysis and machine intelligence* 32.9 (2009), pp. 1582–1596.
- [50] J. Van de Weijer, T. Gevers, and A. D. Bagdanov. "Boosting color saliency in image feature detection". In: *IEEE transactions on pattern analysis and machine intelligence* 28.1 (2005), pp. 150–156.
- [51] J. Van de Weijer, T. Gevers, and J.-M. Geusebroek. "Edge and corner detection by photometric quasi-invariants". In: *IEEE transactions on pattern analysis and machine intelligence* 27.4 (2005), pp. 625–630.
- [52] P. Kubelka and F. Munk. "Ein beitrag zur optik der farbanstriche". In: *Zeitung fur Technische Physik*. Vol. 12. 1999, p. 593.
- [53] G. D. Finlayson and S. D. Hordley. "Color constancy at a pixel." In: *Journal of the Optical Society of America. A, Optics, image science, and vision* 18 2 (2001), pp. 253–64.
- [54] G. D. Finlayson, M. S. Drew, and C. Lu. "Entropy Minimization for Shadow Removal". In: *International Journal of Computer Vision* 85 (2009), pp. 35–57.
- [55] G. D. Finlayson, S. D. Hordley, C. Lu, and M. S. Drew. "On the removal of shadows from images". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 28 (2006), pp. 59–68.
- [56] P. I. Corke, R. Paul, W. Churchill, and P. Newman. "Dealing with shadows: Capturing intrinsic scene appearance for image-based outdoor localisation". In: *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems* (2013), pp. 2085–2092.
- [57] J. A. E. Alvarez, A. Lopez, and R. Baldrich. "Illuminant-invariant model-based road segmentation". In: *2008 IEEE Intelligent Vehicles Symposium* (2008), pp. 1175–1180.
- [58] T. Kim, Y.-W. Tai, and S.-e. Yoon. "PCA Based Computation of Illumination-Invariant Space for Road Detection". In: *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)* (2017), pp. 632–640.
- [59] T. Krajník, J. Blažíček, and J. M. Santos. "Visual road following using intrinsic images". In: *2015 European Conference on Mobile Robots (ECMR)*. 2015, pp. 1–6. DOI: 10.1109/ECMR.2015.7324212.
- [60] B. Upcroft, C. McManus, W. Churchill, W. P. Maddern, and P. Newman. "Lighting invariant urban street classification". In: *2014 IEEE International Conference on Robotics and Automation (ICRA)* (2014), pp. 1712–1718.
- [61] N. Alshammari, S. Akcay, and T. P. Breckon. "On the Impact of Illumination-Invariant Image Pre-transformation for Contemporary Automotive Semantic Scene Understanding". In: *2018 IEEE Intelligent Vehicles Symposium (IV)*. 2018, pp. 1027–1032.

- [62] N. Alshammari, S. Akçay, and T. Breckon. “Multi-Task Learning for Automotive Foggy Scene Understanding via Domain Adaptation to an Illumination-Invariant Representation”. In: *ArXiv abs/1909.07697* (2019).
- [63] B. A. Maxwell, C. A. Smith, M. Qraitem, R. Messing, S. Whitt, N. Thien, and R. 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)* (2019), pp. 1277–1285.
- [64] A. S. Baslamisli, T. T. Groenestege, P. Das, H. A. Le, S. Karaoglu, and T. Gevers. “Joint Learning of Intrinsic Images and Semantic Segmentation”. In: *European Conference on Computer Vision*. 2018. URL: <https://ivi.fnwi.uva.nl/isis/publications/2018/BaslamisliECCV2018>.
- [65] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, J. Xiao, L. Yi, and F. Yu. *ShapeNet: An Information-Rich 3D Model Repository*. Tech. rep. arXiv:1512.03012 [cs.GR]. Stanford University — Princeton University — Toyota Technological Institute at Chicago, 2015.
- [66] W. Jakob. *Mitsuba renderer*. <http://www.mitsuba-renderer.org>. 2010.
- [67] K. He, X. Zhang, S. Ren, and J. Sun. “Deep Residual Learning for Image Recognition”. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 770–778. DOI: 10.1109/CVPR.2016.90.
- [68] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. “ImageNet: A large-scale hierarchical image database”. In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*. 2009, pp. 248–255. DOI: 10.1109/CVPR.2009.5206848.
- [69] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. “Microsoft COCO: Common Objects in Context”. In: *Computer Vision – ECCV 2014*. Ed. by D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars. Cham: Springer International Publishing, 2014, pp. 740–755. ISBN: 978-3-319-10602-1.
- [70] Y. P. Loh and C. S. Chan. “Getting to Know Low-light Images with The Exclusively Dark Dataset”. In: *Computer Vision and Image Understanding* 178 (2019), pp. 30–42. DOI: <https://doi.org/10.1016/j.cviu.2018.10.010>.
- [71] G. D. Finlayson, B. Schiele, and J. L. Crowley. “Comprehensive colour image normalization”. In: *Computer Vision — ECCV’98*. Ed. by H. Burkhardt and B. Neumann. Berlin, Heidelberg: Springer Berlin Heidelberg, 1998, pp. 475–490. ISBN: 978-3-540-69354-3.
- [72] G. Lin, A. Milan, C. Shen, and I. Reid. “RefineNet: Multi-Path Refinement Networks for High-Resolution Semantic Segmentation”. In: *CVPR*. July 2017.
- [73] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. “The Cityscapes Dataset for Semantic Urban Scene Understanding”. In: *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016.

- [74] T.-H. Vu, H. Jain, M. Bucher, M. Cord, and P. Pérez. “ADVENT: Adversarial Entropy Minimization for Domain Adaptation in Semantic Segmentation”. In: *CVPR*. 2019.
- [75] Y. Li, L. Yuan, and N. Vasconcelos. “Bidirectional Learning for Domain Adaptation of Semantic Segmentation”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2019.
- [76] Y. Tsai, W. Hung, S. Schuler, K. Sohn, M. Yang, and M. Chandraker. “Learning to Adapt Structured Output Space for Semantic Segmentation”. In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2018, pp. 7472–7481. DOI: 10.1109/CVPR.2018.00780.
- [77] A. Torii, R. Arandjelović, J. Sivic, M. Okutomi, and T. Pajdla. “24/7 place recognition by view synthesis”. In: *CVPR*. 2015.
- [78] F. Radenović, G. Tolias, and O. Chum. “Fine-Tuning CNN Image Retrieval with No Human Annotation”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 41 (2017), pp. 1655–1668.
- [79] F. Radenović, G. Tolias, and O. Chum. “Deep Shape Matching”. In: *ECCV (2018)*.
- [80] K. Zuiderveld. “Contrast Limited Adaptive Histogram Equalization”. In: *Graphics Gems IV*. USA: Academic Press Professional, Inc., 1994, pp. 474–485. ISBN: 0123361559.
- [81] M. Zaffar, S. Garg, M. Milford, J. Kooij, D. Flynn, K. McDonald-Maier, and S. Ehsan. “Vpr-bench: An open-source visual place recognition evaluation framework with quantifiable viewpoint and appearance change”. In: *International Journal of Computer Vision* (2021), pp. 1–39.
- [82] G. Tolias, R. Sivic, and H. Jégou. “Particular object retrieval with integral max-pooling of CNN activations”. In: *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*. Ed. by Y. Bengio and Y. LeCun. 2016. URL: <http://arxiv.org/abs/1511.05879>.
- [83] N. Sünderhauf, S. Shirazi, F. Dayoub, B. Upcroft, and M. Milford. “On the performance of convnet features for place recognition”. In: *2015 IEEE/RSJ international conference on intelligent robots and systems (IROS)*. IEEE. 2015, pp. 4297–4304.
- [84] A. Torii, R. Arandjelovic, J. Sivic, M. Okutomi, and T. Pajdla. “24/7 place recognition by view synthesis”. In: 2015, pp. 1808–1817.
- [85] J. Revaud, J. Almazán, R. S. Rezende, and C. R. d. Souza. “Learning with average precision: Training image retrieval with a listwise loss”. In: 2019, pp. 5107–5116.

APPENDICES

2.A DERIVATION OF COLOR INVARIANTS

This section summarizes the derivation of the Kubelka-Munk [1] based color invariants by Geusebroek et al. [2].

The Kubelka-Munk model for material reflections describes the spectrum of light E reflected from an object in the viewing direction as

$$E(\lambda, \mathbf{x}) = e(\lambda, \mathbf{x}) \left((1 - \rho_f(\mathbf{x}))^2 R_\infty(\lambda, \mathbf{x}) + \rho_f(\mathbf{x}) \right), \quad (2.5)$$

where \mathbf{x} denotes the spatial location on the image plane, λ the wavelength of the light, e the spectrum of the light source, R_∞ the material reflectivity and ρ_f the Fresnel reflectance coefficient. Partial derivatives of E with respect to x and λ are denoted by subscripts E_x and E_λ , respectively.

By exploring certain simplifying assumptions in Eq. (2.5) we can derive representations that are invariant to one or more of the following conditions: 1) *scene geometry*, i.e. shadows and shading; 2) *Fresnel reflections* from shiny surfaces; 3) *illumination intensity*; and 4) *illumination color*.

E

E is a non-invariant baseline edge detector and therefore no simplifying assumptions are made on Eq. (2.5). Color invariant E is simply defined as:

$$E = \sqrt{E_x^2 + E_{\lambda x}^2 + E_{\lambda \lambda x}^2 + E_y^2 + E_{\lambda y}^2 + E_{\lambda \lambda y}^2}. \quad (2.6)$$

W

Assuming spectrally and spatially uniform illumination, $e(\lambda, \mathbf{x})$ can be represented by a constant i . Moreover, assuming only matte surfaces, i.e. $\rho_f(\mathbf{x}) = 0$, Eq. (2.5) reduces to

$$E(\lambda, \mathbf{x}) = i R_\infty(\lambda, \mathbf{x}). \quad (2.7)$$

The ratio $W_x = \frac{E_x}{E}$ is then independent of the illuminant i :

$$W_x = \frac{E_x}{E} = \frac{1}{R_\infty(\lambda, \mathbf{x})} \frac{\partial R_\infty(\lambda, \mathbf{x})}{\partial \mathbf{x}}. \quad (2.8)$$

The same holds for the ratios $W_{\lambda x} = \frac{E_{\lambda x}}{E}$ and $W_{\lambda\lambda x} = \frac{E_{\lambda\lambda x}}{E}$, and consequently the invariant W can be defined as

$$W = \sqrt{W_x^2 + W_{\lambda x}^2 + W_{\lambda\lambda x}^2 + W_y^2 + W_{\lambda y}^2 + W_{\lambda\lambda y}^2}. \quad (2.9)$$

W is invariant to *illumination intensity*.

C

We assume a spectrally uniform illuminant represented as $i(\mathbf{x})$ and matte surfaces, i.e. $\rho_f(\mathbf{x}) = 0$. Eq. (2.5) then reduces to

$$E(\lambda, \mathbf{x}) = i(\mathbf{x}) R_\infty(\lambda, \mathbf{x}). \quad (2.10)$$

The ratio $C_\lambda = \frac{E_\lambda}{E}$ is then independent of the illuminant i :

$$C_\lambda = \frac{E_\lambda}{E} = \frac{1}{R_\infty(\lambda, \mathbf{x})} \frac{\partial R_\infty(\lambda, \mathbf{x})}{\partial \lambda}. \quad (2.11)$$

The same holds for the ratios $C_{\lambda\lambda} = \frac{E_{\lambda\lambda}}{E}$, $C_{\lambda x} = \frac{E_{\lambda x} E - E_\lambda E_x}{E^2}$ and $C_{\lambda\lambda x} = \frac{E_{\lambda\lambda x} E - E_{\lambda\lambda} E_x}{E^2}$. The color invariant C is defined as

$$C = \sqrt{C_{\lambda x}^2 + C_{\lambda\lambda x}^2 + C_{\lambda y}^2 + C_{\lambda\lambda y}^2}. \quad (2.12)$$

C is invariant to *scene geometry* and *illumination intensity*.

N

We assume a colored illuminant where the power spectrum remains constant over the scene and only varies in intensity, such that the illuminant can be decomposed into a separate spectral and spatial term as $e(\lambda, \mathbf{x}) = e(\lambda) i(\mathbf{x})$. Furthermore, we again assume matte surfaces, i.e. $\rho_f(\mathbf{x}) = 0$. Eq. (2.5) is then defined as

$$E(\lambda, \mathbf{x}) = e(\lambda) i(\mathbf{x}) R_\infty(\lambda, \mathbf{x}). \quad (2.13)$$

Differentiating Eq. (2.13) with respect to λ yields

$$E_\lambda = i(\mathbf{x})R_\infty(\lambda, \mathbf{x}) \frac{\partial e(\lambda)}{\partial \lambda} + e(\lambda)i(\mathbf{x}) \frac{\partial R_\infty(\lambda, \mathbf{x})}{\partial \lambda}. \quad (2.14)$$

Dividing Eq. (2.14) by Eq. (2.13) results in a representation that is invariant to the spatial illuminant term i :

$$N_\lambda = \frac{E_\lambda}{E} = \frac{1}{e(\lambda)} \frac{\partial e(\lambda)}{\partial \lambda} + \frac{1}{R_\infty(\lambda, \mathbf{x})} \frac{\partial R_\infty(\lambda, \mathbf{x})}{\partial \lambda}. \quad (2.15)$$

Additionally differentiating with respect to \mathbf{x} results in the left term dropping out, yielding the color invariant $N_{\lambda x}$ which only depends on the material property R_∞ :

$$N_{\lambda x} = \frac{\partial}{\partial \mathbf{x}} \left\{ \frac{E_\lambda}{E} \right\} = \frac{\partial}{\partial \mathbf{x}} \left\{ \frac{1}{R_\infty(\lambda, \mathbf{x})} \frac{\partial R_\infty(\lambda, \mathbf{x})}{\partial \lambda} \right\}, \quad (2.16)$$

$$= \frac{E_{\lambda x}E - E_\lambda E_x}{E^2}. \quad (2.17)$$

The same holds for higher order derivatives, e.g.

$$N_{\lambda \lambda x} = \frac{E_{\lambda \lambda x}E^2 - E_{\lambda \lambda}E_xE - 2E_{\lambda x}E_\lambda E + 2E_\lambda^2 E_x}{E^3}. \quad (2.18)$$

The color invariant N is defined as

$$N = \sqrt{N_{\lambda x}^2 + N_{\lambda \lambda x}^2 + N_{\lambda y}^2 + N_{\lambda \lambda y}^2} \quad (2.19)$$

and is invariant to *scene geometry*, *illumination intensity* and *illumination color*.

H

We again assume an illuminant with uniform power spectrum such that $e(\lambda, \mathbf{x}) = i(\mathbf{x})$. Eq. (2.5), including Fresnel reflections, then simplifies to

$$E(\lambda, \mathbf{x}) = i(\mathbf{x}) \left((1 - \rho_f(\mathbf{x}))^2 R_\infty(\lambda, \mathbf{x}) + \rho_f(\mathbf{x}) \right). \quad (2.20)$$

The first and second order derivatives with respect to λ are defined as

$$E_\lambda = i(x) (1 - \rho_f(\mathbf{x}))^2 \frac{\partial R_\infty(\lambda, \mathbf{x})}{\partial \lambda}, \quad (2.21)$$

$$E_{\lambda\lambda} = i(x) (1 - \rho_f(\mathbf{x}))^2 \frac{\partial^2 R_\infty(\lambda, \mathbf{x})}{\partial \lambda^2}. \quad (2.22)$$

The ratio $H = \frac{E_\lambda}{E_{\lambda\lambda}}$ then only depends on the material property R_∞ and is thus an invariant to *scene geometry, illumination intensity* and *Fresnel reflections*. Since the spatial derivative $H_x = \frac{\partial}{\partial x} \frac{E_\lambda}{E_{\lambda\lambda}}$ is ill-defined for $E_{\lambda\lambda} = 0$, H is instead defined as $H = \arctan \frac{E_\lambda}{E_{\lambda\lambda}}$, for which the spatial derivative is

$$H_x = \frac{1}{1 + \left(\frac{E_\lambda}{E_{\lambda\lambda}}\right)^2} \frac{E_{\lambda\lambda} E_{\lambda x} - E_\lambda E_{\lambda\lambda x}}{E_{\lambda\lambda}^2} \quad (2.23)$$

$$= \frac{E_{\lambda\lambda} E_{\lambda x} - E_\lambda E_{\lambda\lambda x}}{E_\lambda^2 + E_{\lambda\lambda}^2}. \quad (2.24)$$

Color invariant H is defined as

$$H = \sqrt{H_x^2 + H_y^2}. \quad (2.25)$$

2.B DISTRIBUTION ALIGNMENT BY CICONV

Fig. 2.B.1 shows the feature map activations of a baseline ResNet-18 model and each of the different color invariant models, as described in Section 2.4.1. The intensity change between the "Normal" (daytime) and "Darker" (nighttime) test set causes a clear distribution shift throughout all network layers of the baseline model. In contrast, the CIconv layer produces a domain invariant feature representation and consequently the distributions in the color invariant networks are more aligned between the two domains. This is the case for each of the color invariants, although the "Normal" and "Darker" distributions in the final layer appear to be most aligned for W , which may explain its generally better performance compared to the other invariants.

To quantify the distribution shift we computed the L2 distance between the feature map activations for the "Normal" and "Darker" test sets. As shown in Table 2.B.1, W has indeed the most constant feature map activations.

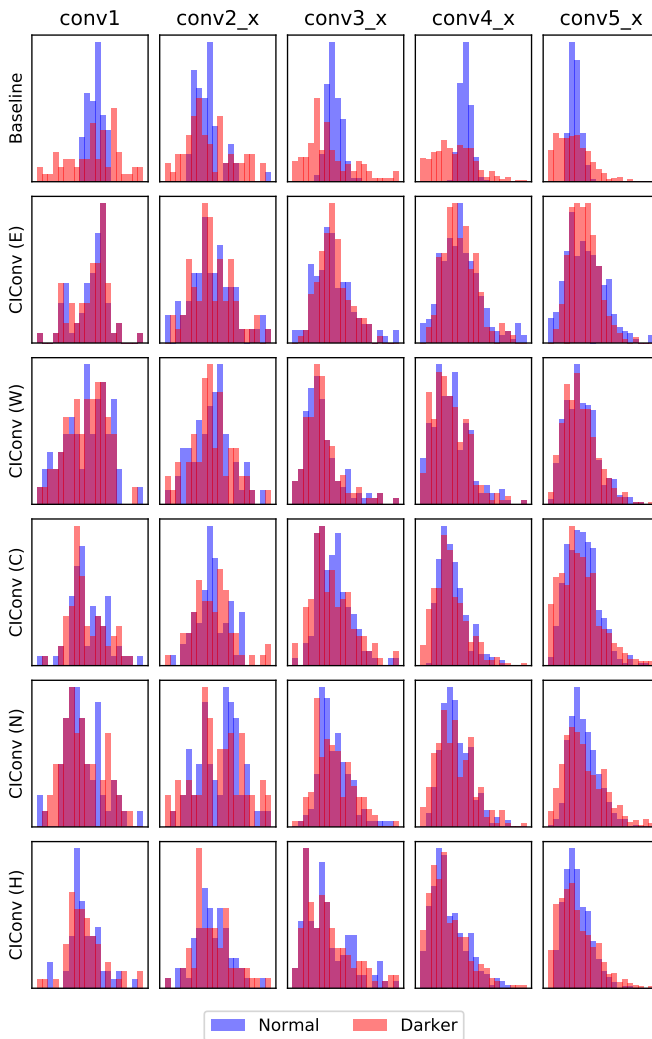


Figure 2.B.1: Histogram of ResNet-18 feature map activations for "Normal" (daytime) and "Darker" (nighttime) test sets of synthetic dataset. Baseline network shows clear distribution shift between test sets, which is greatly reduced in color invariant networks.

	conv1	conv2	conv3	conv4	conv5
Baseline	25.77	2.25	2.32	2.96	2.71
E	0.02	0.43	0.5	0.58	0.58
W	0.02	0.36	0.38	0.55	0.46
C	0.02	0.95	0.91	1.33	1.14
N	0.02	1.1	1.06	1.33	1.14
H	0.01	0.8	0.88	0.98	1.19

Table 2.B.1: Feature map activation similarities of ResNet-18 feature maps for "Normal" and "Darker" test sets of synthetic dataset, measured by L2 distance. W has most constant feature maps.

2.C COMBINING COLOR INVARIANTS

We investigated the use of multiple input modalities by concatenating the output of W with either RGB, E , C , N or H in the input layer. Results on the CODaN classification dataset in Table 2.C.1 show that performance deteriorates compared to only W (None), likely due to overfitting on a combination of input modalities rather than using them in a complementary fashion. This again shows the need for developing an adaptive fusion mechanism as mentioned in the Discussion.

$W+$	None	RGB	E	C	N	H
Day	81.49	66.08	69.72	66.00	66.48	68.56
Night	59.67	43.52	46.65	46.44	45.19	47.65

Table 2.C.1: Classification accuracy (%) on CODaN. Combining W with other input modalities does not improve performance.

2.D SEMANTIC SEGMENTATION PER-CLASS SCORES

The per-class Intersection-over-Union (IoU) scores of the semantic segmentation experiment are shown in Table 2.D.1 for Nighttime Driving [3] and Table 2.D.2 for Dark Zurich [4]. Our W -RefineNet improves segmentation over the baseline performance across nearly all classes.

Method	road	sidew.	build.	wall	fence	pole	light	sign	veget.	terrain	sky	person	rider	car	truck	bus	train	motorc.	bicycle	mIoU
Trained on source data																				
RefineNet [5]	83.2	32.9	82.0	18.6	0.0	35.5	22.5	39.4	45.8	0.0	29.0	53.0	0.0	57.7	0.0	67.7	63.1	0.0	16.5	34.1
W-RefineNet [ours]	89.6	52.7	82.7	16.2	0.0	39.6	52.2	60.6	43.9	0.0	38.6	55.1	24.3	72.0	0.0	73.2	66.8	0.0	23.6	41.6
RefineNet-AdaBN [6]	88.9	58.2	75.5	22.5	0.0	39.0	21.3	50.9	36.4	0.0	25.7	53.4	0.0	68.0	0.0	63.3	62.7	0.0	24.4	36.3

Table 2.D.1: Per-class semantic segmentation results on the Nighttime Driving [3] dataset, reported as IoU.

Method	road	sidew.	build.	wall	fence	pole	light	sign	veget.	terrain	sky	person	rider	car	truck	bus	train	motorc.	bicycle	mIoU
Trained on source data																				
RefineNet [5]	86.2	34.8	62.0	26.0	12.8	30.9	14.4	27.7	38.4	10.0	3.1	38.3	34.5	49.1	6.0	0.0	55.4	31.1	20.4	30.6
W-RefineNet [ours]	90.3	48.3	57.8	29.3	11.1	36.3	24.4	30.2	45.8	7.6	8.0	37.6	40.1	69.7	10.1	0.0	55.0	37.4	16.0	34.5
RefineNet-AdaBN [6]	87.0	51.8	53.1	28.4	14.7	32.8	11.3	31.9	33.8	18.4	2.4	32.4	39.6	59.7	10.5	0.0	32.9	34.2	20.0	31.3
Trained on source and target data																				
AdaptSegNet [7]	86.1	44.2	55.1	22.2	4.8	21.1	5.6	16.7	37.2	8.4	1.2	35.9	26.7	68.2	45.1	0.0	50.1	33.9	15.6	30.4
ADVENT [8]	85.8	37.9	55.5	27.7	14.5	23.1	14.0	21.1	32.1	8.7	2.0	39.9	16.6	64.0	13.8	0.0	58.8	28.5	20.7	29.7
BDL [9]	85.3	41.1	61.9	32.7	17.4	20.6	11.4	21.3	29.4	8.9	1.1	37.4	22.1	63.2	28.2	0.0	47.7	39.4	15.7	30.8
DMAda [3]	75.5	29.1	48.6	21.3	14.3	34.3	36.8	29.9	49.4	13.8	0.4	43.3	50.2	69.4	18.4	0.0	27.6	34.9	11.9	32.1
GCMA [10]	81.7	46.9	58.8	22.0	20.0	41.2	40.5	41.6	64.8	31.0	32.1	53.5	47.5	75.5	39.2	0.0	49.6	30.7	21.0	42.0
MGCDA [4]	80.3	49.3	66.2	7.8	11.0	41.4	38.9	39.0	64.1	18.0	55.8	52.1	53.5	74.7	66.0	0.0	37.5	29.1	22.7	42.5

Table 2.D.2: Per-class semantic segmentation results on the Dark Zurich [10] dataset, reported as IoU. Results of methods trained on source and target data taken from [4].

REFERENCES

- [1] P. Kubelka and F. Munk. "Ein beitrage zur optik der farbanstriche". In: *Zeitung fur Technische Physik*. Vol. 12. 1999, p. 593.
- [2] J. M. Geusebroek, R. van den Boomgaard, A. W. M. Smeulders, and H. Geerts. "Color Invariance". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 23.12 (2001), pp. 1338–1350.
- [3] D. Dai and L. V. Gool. "Dark Model Adaptation: Semantic Image Segmentation from Daytime to Nighttime". In: *ITSC*. Nov. 2018, pp. 3819–3824. DOI: 10.1109/ITSC.2018.8569387.
- [4] C. Sakaridis, D. Dai, and L. Van Gool. "Map-Guided Curriculum Domain Adaptation and Uncertainty-Aware Evaluation for Semantic Nighttime Image Segmentation". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2020). DOI: 10.1109/TPAMI.2020.3045882.
- [5] G. Lin, A. Milan, C. Shen, and I. Reid. "RefineNet: Multi-Path Refinement Networks for High-Resolution Semantic Segmentation". In: *CVPR*. July 2017.
- [6] Y. Li, N. Wang, J. Shi, J. Liu, and X. Hou. "Revisiting Batch Normalization For Practical Domain Adaptation". In: *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Workshop Track Proceedings*. OpenReview.net, 2017. URL: <https://openreview.net/forum?id=Hk6dkJQFx>.
- [7] Y. Tsai, W. Hung, S. Schulter, K. Sohn, M. Yang, and M. Chandraker. "Learning to Adapt Structured Output Space for Semantic Segmentation". In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2018, pp. 7472–7481. DOI: 10.1109/CVPR.2018.00780.
- [8] T.-H. Vu, H. Jain, M. Bucher, M. Cord, and P. Pérez. "ADVENT: Adversarial Entropy Minimization for Domain Adaptation in Semantic Segmentation". In: *CVPR*. 2019.
- [9] Y. Li, L. Yuan, and N. Vasconcelos. "Bidirectional Learning for Domain Adaptation of Semantic Segmentation". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2019.
- [10] C. Sakaridis, D. Dai, and L. V. 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)*. Oct. 2019.

3

COLOR EQUIVARIANT CONVOLUTIONAL NETWORKS

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 (CEConv), 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 CEConv 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.

This chapter has been published as:

A. Lengyel, O. Strafforello, R. Brintjes, A. Gielisse, and J. van Gemert. “Color Equivariant Convolutional Networks”. In: *Advances in Neural Information Processing Systems (NeurIPS)*, Dec. 2023, pp. 29831–29850.

Code available at:

<https://github.com/Attila94/CEConv>

3.1 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 [1, 2]. 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 type and ripeness of a fruit, where color is an intrinsic property facilitating recognition [3]. This is illustrated in Fig. 3.1. 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 [4].

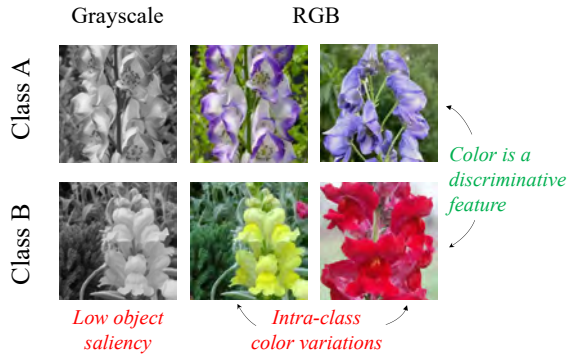


Figure 3.1: Color plays a crucial role in object recognition. The absence of color makes flowers less distinct from their background and thus harder to classify. The characteristic purple-blue color of the Monkshood (Class A) enables a clear distinction from the Snapdragon (Class B) [5]. On the other hand, relying too much on colors might negatively impact recognition to color variations within the same flower class.

However, unwanted color variations can be introduced by accidental scene recording conditions such as illumination changes [6, 7], 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 [8, 9]. Yet, 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 [10, 11]. However, color invariance comes at the loss of discriminative power as valuable color information is removed from the model’s internal feature representation [12]. 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 retain 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 [13] (GConvs), enabling 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 [14] and learn more color selective filters [4, 9].

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.

3.2 RELATED WORK

Equivariant architectures Translation equivariance is a key property of convolutional neural networks (CNNs) [15, 16]: 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 [17], including the seminal Group Equivariant Convolutions [13] for rotations and flips, and other works concerning rotations [18–20], scaling [21, 22] and arbitrary Lie groups [23]. Yet to date, equivariance to photometric transformations has remained largely unexplored. Offset equivariant networks [24] 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 [25]. In this work we explore an alternative approach to photometric

equivariance inspired by the seminal Group Equivariant Convolution [13] 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, [4, 26] 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, [27] 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 [28], further supporting these findings. [8, 9] investigate learned symmetries in an InceptionV1 model trained on ImageNet [29] and discover filters that show 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 [4] to provide an explanation for several of our own findings.

Color priors in deep learning Color is an important visual discriminator [30–32]. In classical computer vision, color invariants are used to extract features from an RGB image that are more consistent under illumination changes [10–12]. Recent studies have explored using color invariants as a preprocessing step to deep neural networks [33, 34] or incorporating them directly into the architecture itself [6], leading to improved robustness against time-of-day domain shifts and other illumination-based variations in the input. Capsule networks [35, 36], which use groups of neurons to represent object properties such as pose and appearance, have shown encouraging results in image colorization tasks [37]. Quaternion networks [38, 39] 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.

3.3 COLOR EQUIVARIANT CONVOLUTIONS

3.3.1 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 equivalently, 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

$$\Phi(T_g x) = T'_g \Phi(x), \quad \forall g \in G, \quad (3.1)$$

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:

$$\Phi(T_g x) = \Phi(x), \quad \forall g \in G. \quad (3.2)$$

We use the definition from [13] 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 :

$$[f \star \psi^i](x) = \sum_{y \in \mathbb{Z}^2} \sum_{c=1}^{C^l} f_c(y) \cdot \psi_c^i(y-x). \quad (3.3)$$

Here $f: \mathbb{Z}^2 \rightarrow \mathbb{R}^{C^l}$ and $\psi^i: \mathbb{Z}^2 \rightarrow \mathbb{R}^{C^l}$ 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 :

$$[f \star \psi^i](g) = \sum_{y \in \mathbb{Z}^2} \sum_c^{C^l} f_c(y) \cdot \psi_c^i(g^{-1}y) \quad (3.4)$$

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 :

$$[f \star \psi^i](g) = \sum_{h \in G} \sum_c^{C^l} f_c(h) \cdot \psi_c^i(g^{-1}h) \quad (3.5)$$

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 [13, 40].

3.3.2 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

$$H_n(k) = \begin{bmatrix} \cos(\frac{2k\pi}{n}) + a & a - b & a + b \\ a + b & \cos(\frac{2k\pi}{n}) + a & a - b \\ a - b & a + b & \cos(\frac{2k\pi}{n}) + a \end{bmatrix} \quad (3.6)$$

with n the total number of discrete rotations in the group, k the rotation index, $a = \frac{1}{3} - \frac{1}{3} \cos(\frac{2k\pi}{n})$ 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 Section 3.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:

$$[f \star \psi^i](x, k) = \sum_{y \in \mathbb{Z}^2} \sum_{c=1}^{C^l} f_c(y) \cdot H_n(k) \psi_c^i(y - x). \quad (3.7)$$

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 :

$$[\mathcal{L}_g f](x) = [\mathcal{L}_{(t,m)} f](x) = H_n(m) f(x - t) \quad (3.8)$$

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 $H_n^{-1} \psi$:

$$H_n f \cdot \psi = (H_n f)^T \psi = f^T H_n^T \psi = f \cdot H_n^T \psi = f \cdot H_n^{-1} \psi. \quad (3.9)$$

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

$$\begin{aligned} [[\mathcal{L}_{(t,m)} f] \star \psi^i](x, k) &= \sum_{y \in \mathbb{Z}^2} H_n(m) f(y - t) \cdot H_n(k) \psi^i(y - x) \\ &= \sum_{y \in \mathbb{Z}^2} f(y) \cdot H_n(m)^{-1} H_n(k) \psi^i(y - (x - t)) \\ &= \sum_{y \in \mathbb{Z}^2} f(y) \cdot H_n(k - m) \psi^i(y - (x - t)) \\ &= [f \star \psi^i](x - t, k - m) \\ &= [\mathcal{L}'_{(t,m)} [f \star \psi^i]](x, k) \end{aligned} \quad (3.10)$$

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:

$$[f \star \psi^i](x, k) = \sum_{y \in \mathbb{Z}^2} \sum_{r=1}^n \sum_{c=1}^{C^l} f_c(y, r) \cdot \psi_c^i(y - x, (r - k) \% n), \quad (3.11)$$

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 the RGB cube. This causes a subtle difference between applying hue shifts through rotation in RGB space versus a transformation in HSV space, as in the latter pixels are reprojected within the cube. Due to this discrepancy, Eq. (3.9) only holds approximately when input images are transformed in HSV

space, though in practice this has only limited consequences, as we empirical show in Section 3.D.

3.3.3 IMPLEMENTATION

Tensor operations We implement CEConv similarly to GConv [13]. 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, the number of transformations that leave the origin invariant, and the 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:

$$X_{c',g',:,,:}^{l+1} = \sum_c \sum_g^{G^l} \tilde{F}_{c',g',c,g',:,,:}^l \star X_{c,g',:,,:}^l \quad (3.12)$$

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. (3.6):

$$\tilde{F}_{c',g',:,1,u,v}^1 = H_n(g') F_{c',:,1,u,v}^1 \quad (3.13)$$

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

$$\tilde{F}_{c',g',c,g,u,v}^l = F_{c',c,(g+g')\%n,u,v}^l \quad (3.14)$$

Furthermore, to explicitly share the channel-wise spatial kernel over G^l [19], filter F is decomposed into a spatial component S and a pointwise component P as follows:

$$F_{c',c,g,u,v}^l = S_{c',c,1,u,v} \cdot P_{c',g',c,g,1,1} \quad (3.15)$$

\tilde{F} is precomputed in each forward step prior to the convolution operation in Eq. (3.12).

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.

3

Compute efficiency CEConvs create a factor $|H_n|$ more feature maps in each layer. Due to the decomposition in Eq. (3.15), 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 Section 3.C.3 for an overview of parameter counts and MAC operations.

3.4 EXPERIMENTS

3.4.1 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 [41] dataset. We use the Z2CNN architecture from [13], 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 [42] 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 Section 3.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. 3.2a. With an average accuracy of $91.35 \pm 0.40\%$ the CECNN performs significantly better than the Z2CNN 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 Section 3.B.2 for both detailed results on both experiments.

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 Section 3.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. 3.2b. CECNN outperforms Z2CNN across all standard deviations, indicating that CEConvs allow for a more efficient internal color representation. The color invariant CECNN network outperforms the equivariant CECNN model for $\sigma \geq 48$. 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 Section 3.B.2 for detailed results.

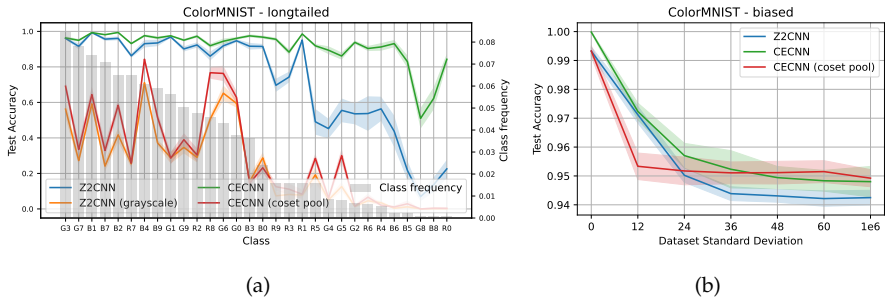


Figure 3.2: Color equivariant convolutions efficiently share shape information across different colors. CECNN outperforms a vanilla network in both a long-tailed class imbalance setting (a), where MNIST digits are to be classified based on both shape and color, and a color biased setting (b), where the color of each class c is sampled according to $\theta_d \sim \mathcal{N}(\theta_c, \sigma)$.

3.4.2 IMAGE CLASSIFICATION

Setup We evaluate our method for robustness to color variations on several natural image classification datasets, including CIFAR-10 and CIFAR-100 [43], Flowers-102 [5], STL-10 [44], Oxford-IIIT Pet [45], Caltech-101 [46], Stanford Cars [47] and ImageNet [29]. We train a baseline and color equivariant (CE) ResNet [48] 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 [13], 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 [49] and CIConv [6] as baselines for comparison. Training is performed for 200 epochs using the Adam [50] 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 equivariant networks, denoted by an integer suffix for the number of ResNet stages, out of a possible four, that use CEConv.

<i>Original test set</i>	Caltech	C-10	C-100	Flowers	Ox-Pet	Cars	STL10	ImNet
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	-
<i>Hue-shifted test set</i>								
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 3.1: 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 3.1. 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 Section 3.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. 3.3. 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 Section 3.C.2.

Color selectivity To explore what affects the effectiveness of color equivariance, we investigate the *color selectivity* of a subset of the studied datasets. We use the color selectivity measure from [4] and average across all neurons in the baseline model trained on each dataset. Fig. 3.4 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 [4] (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. 3.5 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. 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 Section 3.C.4.

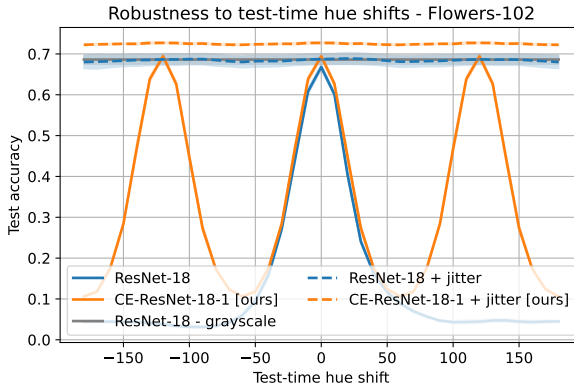


Figure 3.3: Classification accuracy on the Flowers-102 dataset [5] under a gradual variation of the image hue. test time hue shifts degrade the baseline performance (ResNet-18) drastically. Grayscale images and color augmentations result in invariance to hue variations, but fail to capture characteristic color features. Our color equivariant network (CE-ResNet-18-1) enables feature sharing across colors, which improves generalization while preserving discriminative color information.

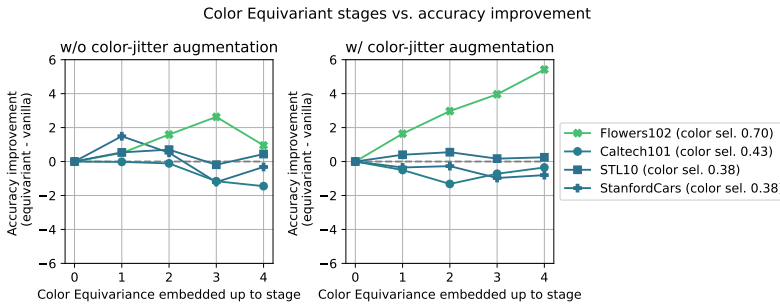


Figure 3.4: Color selective datasets benefit from using color equivariance up to late stages, whereas less color selective datasets do not. We compute average color selectivity [4] of all neurons in the baseline CNN trained on each dataset, and plot the accuracy improvement of using color equivariance in hybrid and full models, coloring each graphed dataset for color selectivity.

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

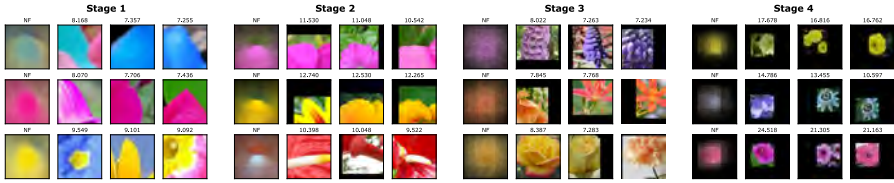


Figure 3.5: Neuron Feature [4] (NF) visualization with top-3 patches at different stages of a CE-ResNet18 trained on Flowers-102. Rows represent different rotations of the same filter. As expected, each row of a NF activates on the same shape in a different color.

3

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 Section 3.D.

3.5 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. (3.9) 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 transformation groups on images. CNNs naturally learn features that vary in both photometric and geometric transformations [9, 14]. Future work could combine hue shifts with geometric transformations such as roto-translation [13] and scaling [51]. 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 [52, 53]. 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.

REFERENCES

- [1] D. Osorio and M. Vorobyev. “Colour Vision as an Adaptation to Frugivory in Primates”. In: *Proceedings. Biological sciences / The Royal Society* 263 (June 1996), pp. 593–9. DOI: 10.1098/rspb.1996.0089.
- [2] B. Regan, C. Julliot, B. Simmen, F. Viénot, P. Charles-Dominique, and J. Mollon. “Fruits, foliage and the evolution of primate colour vision”. In: *Philosophical transactions of the Royal Society of London. Series B, Biological sciences* 356 (Mar. 2001), pp. 229–83. DOI: 10.1098/rstb.2000.0773.
- [3] I. Bramão, L. Faísca, K. M. Petersson, and A. Reis. “The Contribution of Color to Object Recognition”. In: *Advances in Object Recognition Systems*. Ed. by I. Kypraios. Rijeka: IntechOpen, 2012. Chap. 4. DOI: 10.5772/34821. URL: <https://doi.org/10.5772/34821>.
- [4] I. Rafegas and M. Vanrell. “Color encoding in biologically-inspired convolutional neural networks”. In: *Vision Research* 151 (2018). Color: cone opponency and beyond, pp. 7–17. ISSN: 0042-6989. DOI: <https://doi.org/10.1016/j.visres.2018.03.010>. URL: <https://www.sciencedirect.com/science/article/pii/S0042698918300592>.
- [5] M.-E. Nilsback and A. Zisserman. “Automated Flower Classification over a Large Number of Classes”. In: *Indian Conference on Computer Vision, Graphics and Image Processing*. Oct. 2008.
- [6] A. Lengyel, S. Garg, M. Milford, and J. 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)*. Oct. 2021, pp. 4399–4409.
- [7] C. Sakaridis, D. Dai, and L. V. 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)*. Oct. 2019.
- [8] C. Olah, N. Cammarata, L. Schubert, G. Goh, M. Petrov, and S. Carter. “An Overview of Early Vision in InceptionV1”. In: *Distill* (2020). DOI: 10.23915/distill.00024.002. URL: <https://distill.pub/2020/circuits/early-vision>.
- [9] C. Olah, N. Cammarata, C. Voss, L. Schubert, and G. Goh. “Naturally Occurring Equivariance in Neural Networks”. In: *Distill* (2020). DOI: 10.23915/distill.00024.004. URL: <https://distill.pub/2020/circuits/equivariance>.
- [10] G. Finlayson, S. Hordley, C. Lu, and M. Drew. “On the removal of shadows from images”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 28.1 (2006), pp. 59–68. DOI: 10.1109/TPAMI.2006.18.
- [11] J. M. Geusebroek, R. van den Boomgaard, A. W. M. Smeulders, and H. Geerts. “Color Invariance”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 23.12 (2001), pp. 1338–1350.

- [12] 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. URL: <https://ivi.fnwi.uva.nl/isis/publications/2012/GeversSIST2012>.
- [13] T. S. Cohen and M. Welling. “Group Equivariant Convolutional Networks”. In: *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*. ICML’16. New York, NY, USA: JMLR.org, 2016, pp. 2990–2999.
- [14] R.-J. Bruintjes, T. Motyka, and J. van Gemert. “What Affects Learned Equivariance in Deep Image Recognition Models?” In: *arXiv preprint arXiv:2304.02628* (2023).
- [15] O. S. Kayhan and J. C. v. 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*. 2020, pp. 14274–14285.
- [16] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. “Gradient-Based Learning Applied to Document Recognition”. In: *Proceedings of the IEEE*. Vol. 86. 1998, pp. 2278–2324.
- [17] M. Rath and A. Condurache. “Boosting Deep Neural Networks with Geometrical Prior Knowledge: A Survey”. In: *ArXiv abs/2006.16867* (2020).
- [18] E. J. Bekkers, M. W. Lafarge, M. Veta, K. A. J. Eppenhof, J. P. W. Pluim, and R. Duits. “Roto-Translation Covariant Convolutional Networks for Medical Image Analysis”. In: *Medical Image Computing and Computer Assisted Intervention – MICCAI 2018*. Springer International Publishing, 2018, pp. 440–448. ISBN: 978-3-030-00928-1.
- [19] A. Lengyel and J. C. van Gemert. “Exploiting Learned Symmetries in Group Equivariant Convolutions”. In: *2021 IEEE International Conference on Image Processing (ICIP)*. 2021, pp. 759–763. DOI: 10.1109/ICIP42928.2021.9506362.
- [20] M. Weiler, F. A. Hamprecht, and M. Storath. “Learning Steerable Filters for Rotation Equivariant CNNs”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2018.
- [21] I. Sosnovik, M. Szmaja, and A. Smeulders. “Scale-Equivariant Steerable Networks”. In: *International Conference on Learning Representations*. 2020. URL: <https://openreview.net/forum?id=HJgpugrKPS>.
- [22] D. Worrall and M. Welling. “Deep Scale-spaces: Equivariance Over Scale”. In: *Advances in Neural Information Processing Systems*. Vol. 32. Curran Associates, Inc., 2019, pp. 7366–7378.
- [23] L. E. MacDonald, S. Ramasinghe, and S. Lucey. “Enabling Equivariance for Arbitrary Lie Groups”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2022, pp. 8183–8192.

- [24] M. Cotogni and C. Cusano. “Offset equivariant networks and their applications”. In: *Neurocomputing* 502 (2022), pp. 110–119. ISSN: 0925-2312. DOI: <https://doi.org/10.1016/j.neucom.2022.06.118>. URL: <https://www.sciencedirect.com/science/article/pii/S0925231222008499>.
- [25] G. Finlayson, M. Drew, and B. Funt. “Diagonal transforms suffice for color constancy”. In: *1993 (4th) International Conference on Computer Vision*. 1993, pp. 164–171. DOI: 10.1109/ICCV.1993.378223.
- [26] I. Rafegas and M. Vanrell. “Color Representation in CNNs: Parallelisms with Biological Vision”. In: *2017 IEEE International Conference on Computer Vision Workshops (ICCVW)*. 2017, pp. 2697–2705. DOI: 10.1109/ICCVW.2017.318.
- [27] I. Rafegas, M. Vanrell, L. A. Alexandre, and G. Arias. “Understanding trained CNNs by indexing neuron selectivity”. In: *Pattern Recognition Letters* 136 (2020), pp. 318–325. ISSN: 0167-8655. DOI: <https://doi.org/10.1016/j.patrec.2019.10.013>. URL: <https://www.sciencedirect.com/science/article/pii/S0167865519302909>.
- [28] M. Engilberge, E. Collins, and S. Süsstrunk. “Color representation in deep neural networks”. In: *2017 IEEE International Conference on Image Processing (ICIP)*. 2017, pp. 2786–2790. DOI: 10.1109/ICIP.2017.8296790.
- [29] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. “ImageNet: A large-scale hierarchical image database”. In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*. 2009, pp. 248–255. DOI: 10.1109/CVPR.2009.5206848.
- [30] B. V. Funt and G. D. Finlayson. “Color constant color indexing”. In: *IEEE transactions on Pattern analysis and Machine Intelligence* 17.5 (1995), pp. 522–529.
- [31] 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. URL: <https://ivi.fnwi.uva.nl/isis/publications/2012/GeversSIST2012>.
- [32] J. 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*. 2011, pp. 1–8.
- [33] N. Alshammari, S. Akcay, and T. P. Breckon. “On the Impact of Illumination-Invariant Image Pre-transformation for Contemporary Automotive Semantic Scene Understanding”. In: *2018 IEEE Intelligent Vehicles Symposium (IV)*. 2018, pp. 1027–1032. DOI: 10.1109/IVS.2018.8500664.
- [34] B. A. Maxwell, C. A. Smith, M. Qraitem, R. Messing, S. Whitt, N. Thien, and R. 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)*. 2019, pp. 1277–1285. DOI: 10.1109/CVPRW.2019.00167.

- [35] G. E. Hinton, S. Sabour, and N. Frosst. “Matrix capsules with EM routing”. In: *International Conference on Learning Representations*. 2018. URL: <https://openreview.net/forum?id=HJWLfGWRb>.
- [36] S. Sabour, N. Frosst, and G. E. Hinton. “Dynamic Routing Between Capsules”. In: *Advances in Neural Information Processing Systems*. Ed. by I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett. Vol. 30. Curran Associates, Inc., 2017.
- [37] G. Ozbulak. “Image Colorization by Capsule Networks”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*. June 2019.
- [38] C. J. Gaudet and A. Maida. “Deep Quaternion Networks”. In: *2018 International Joint Conference on Neural Networks (IJCNN)* (2018), pp. 1–8.
- [39] X. Zhu, Y. Xu, H. Xu, and C. Chen. “Quaternion Convolutional Neural Networks”. In: *Proceedings of the European Conference on Computer Vision (ECCV)*. June 2018.
- [40] M. M. Bronstein, J. Bruna, T. Cohen, and P. Velickovic. “Geometric Deep Learning: Grids, Groups, Graphs, Geodesics, and Gauges”. In: *CoRR* abs/2104.13478 (2021). arXiv: 2104.13478. URL: <https://arxiv.org/abs/2104.13478>.
- [41] L. Deng. “The mnist database of handwritten digit images for machine learning research”. In: *IEEE Signal Processing Magazine* 29.6 (2012), pp. 141–142.
- [42] D. Kingma and J. Ba. “Adam: A Method for Stochastic Optimization”. In: *International Conference on Learning Representations* (Dec. 2014).
- [43] A. Krizhevsky and G. Hinton. *Learning multiple layers of features from tiny images*. Tech. rep. 0. Toronto, Ontario: University of Toronto, 2009. URL: <https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf>.
- [44] A. Coates, A. Ng, and H. Lee. “An analysis of single-layer networks in unsupervised feature learning”. In: *Proceedings of the fourteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings. 2011, pp. 215–223.
- [45] O. M. Parkhi, A. Vedaldi, A. Zisserman, and C. V. Jawahar. “Cats and Dogs”. In: *IEEE Conference on Computer Vision and Pattern Recognition*. 2012.
- [46] F.-F. Li, M. Andreeto, M. Ranzato, and P. Perona. *Caltech 101*. Apr. 2022. DOI: 10.22002/D1.20086.
- [47] J. Krause, M. Stark, J. Deng, and L. Fei-Fei. “3D Object Representations for Fine-Grained Categorization”. In: *2013 IEEE International Conference on Computer Vision Workshops*. 2013, pp. 554–561. DOI: 10.1109/ICCVW.2013.77.
- [48] K. He, X. Zhang, S. Ren, and J. Sun. “Deep Residual Learning for Image Recognition”. In: *Proceedings of 2016 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, June 2016, pp. 770–778.

- [49] D. Hendrycks, N. Mu, E. D. Cubuk, B. Zoph, J. Gilmer, and B. Lakshminarayanan. “AugMix: A Simple Data Processing Method to Improve Robustness and Uncertainty”. In: *Proceedings of the International Conference on Learning Representations (ICLR)* (2020).
- [50] D. P. Kingma and J. Ba. “Adam: A Method for Stochastic Optimization”. In: *CoRR* abs/1412.6980 (2014).
- [51] I. Sosnovik, A. Moskalev, and A. Smeulders. “DISCO: accurate discrete scale convolutions”. In: *Proceedings of the 32nd British Machine Vision Conference (BMVC)*. 2021.
- [52] A. Moskalev, A. Sepiarskaia, I. Sosnovik, and A. Smeulders. “LieGG: Studying Learned Lie Group Generators”. In: *Advances in Neural Information Processing Systems*. Ed. by S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh. Vol. 35. Curran Associates, Inc., 2022, pp. 25212–25223.
- [53] D. W. Romero and S. Lohit. “Learning Partial Equivariances From Data”. In: *Advances in Neural Information Processing Systems*. Ed. by S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh. Vol. 35. Curran Associates, Inc., 2022, pp. 36466–36478.

APPENDICES

3.A DERIVATION OF H_n

Rotation around an arbitrary unit vector \mathbf{u} by angle θ can be decomposed into five simple steps [1]:

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:

$$R_{\mathbf{u},\theta} = M_{xz}^T (M_x^T (R_{x,\theta} (M_x (M_{xz})))) \quad (3.16)$$

$$= M_{xz}^T M_x^T R_{x,\theta} M_x M_{xz} \quad (3.17)$$

$$= \begin{bmatrix} \cos\theta + u_x^2(1 - \cos\theta) & u_x u_y(1 - \cos\theta) - u_z \sin\theta & u_x u_z(1 - \cos\theta) + u_y \sin\theta \\ u_y u_x(1 - \cos\theta) + u_z \sin\theta & \cos\theta + u_y^2(1 - \cos\theta) & u_y u_z(1 - \cos\theta) - u_x \sin\theta \\ u_z u_x(1 - \cos\theta) - u_y \sin\theta & u_z u_y(1 - \cos\theta) + u_x \sin\theta & \cos\theta + u_z^2(1 - \cos\theta) \end{bmatrix}. \quad (3.18)$$

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

$$R_{\mathbf{u},\theta} = \begin{bmatrix} \cos\theta + \frac{1}{3}(1 - \cos\theta) & \frac{1}{3}(1 - \cos\theta) - \frac{1}{\sqrt{3}}\sin\theta & \frac{1}{3}(1 - \cos\theta) + \frac{1}{\sqrt{3}}\sin\theta \\ \frac{1}{3}(1 - \cos\theta) + \frac{1}{\sqrt{3}}\sin\theta & \cos\theta + \frac{1}{3}(1 - \cos\theta) & \frac{1}{3}(1 - \cos\theta) - \frac{1}{\sqrt{3}}\sin\theta \\ \frac{1}{3}(1 - \cos\theta) - \frac{1}{\sqrt{3}}\sin\theta & \frac{1}{3}(1 - \cos\theta) + \frac{1}{\sqrt{3}}\sin\theta & \cos\theta + \frac{1}{3}(1 - \cos\theta) \end{bmatrix}, \quad (3.19)$$

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

$$H_n(k) = \begin{bmatrix} \cos(\frac{2k\pi}{n}) + a & a - b & a + b \\ a + b & \cos(\frac{2k\pi}{n}) + a & a - b \\ a - b & a + b & \cos(\frac{2k\pi}{n}) + a \end{bmatrix}, \quad (3.20)$$

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 $b = \sqrt{\frac{1}{3}} * \sin(\frac{2k\pi}{n})$.

3

3.B COLORMNIST

3.B.1 DATASET VISUALIZATION

Long-tailed ColorMNIST dataset The training samples of the *Longtailed ColorMNIST* dataset are depicted in Fig. 3.B.1, clearly indicating a class imbalance.

Biased ColorMNIST dataset A small subset of the samples of Biased ColorMNIST is shown in Fig. 3.B.2 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.

3.B.2 ADDITIONAL EXPERIMENTS

Results with color jitter augmentation We performed both ColorMNIST experiments with color jitter augmentations. The results are shown in Fig. 3.B.3.

(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 the CECNN model with coset pooling.

(b) For biased MNIST, performance decreases for small σ and improves for large σ , with CEConv still performing best.

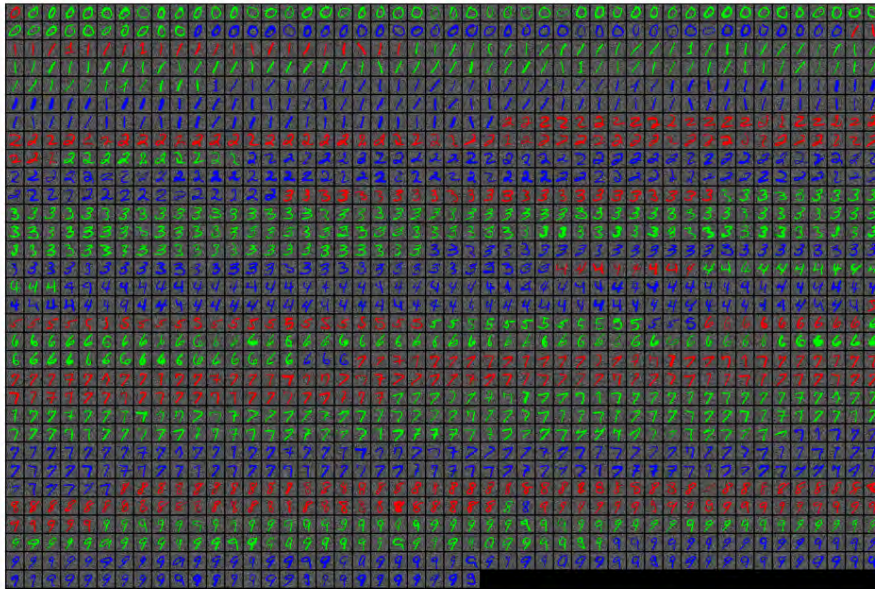


Figure 3.B.1: Long-tailed ColorMNIST. Note the strong class imbalance in the dataset. Best viewed in color.

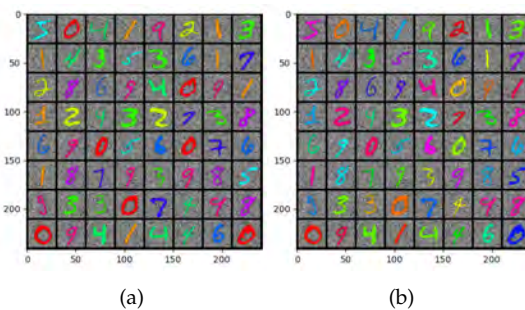


Figure 3.B.2: Samples from Biased ColorMNIST for $\sigma = 0$ (a) and $\sigma = 36$ (b), respectively. Best viewed in color.

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

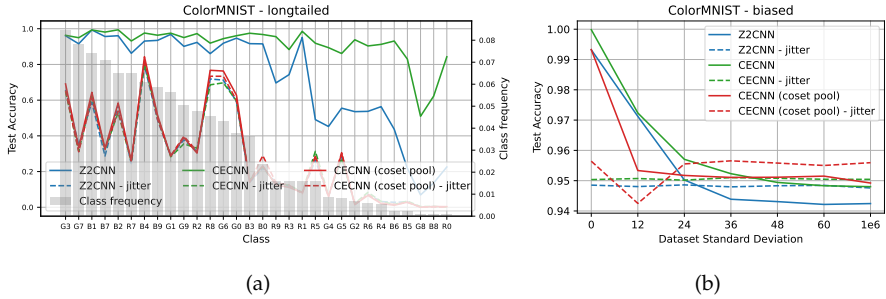


Figure 3.B.3: Classification results on long-tailed and biased ColorMNIST, including models trained with color jitter augmentations. Color jitter makes the models invariant to color.

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. 3.B.4. 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.

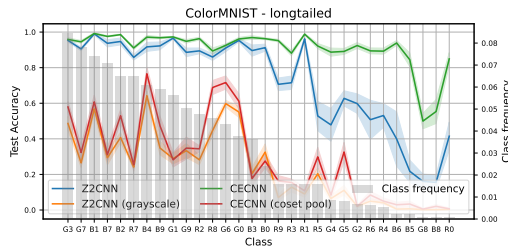


Figure 3.B.4: Per-class accuracy of various models trained with a loss function weighted by inverse class frequency. CECNN without coset pooling outperforms all other models, with no significant differences compared to a uniformly weighted loss function.

3.C CLASSIFICATION EXPERIMENTS

3.C.1 PERFORMANCE OF ALL CE-RESNET CONFIGURATIONS

Table 3.C.1 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 the case of the Flowers-102 dataset, late equivariance provides a significant advantage, whereas for Caltech-101 and Stanford Cars the color equivariance bias does not seem to have much added value.

3.C.2 TEST TIME HUE SHIFT PLOTS

Fig. 3.C.1 shows the test accuracies under a test time hue shift on all datasets in the paper. Each figure includes a regular ResNet, a fully 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.

3.C.3 CE-RESNET CONFIGURATIONS

The configurations of the color equivariant ResNet with three hue rotations, as used in the classification experiment in Section 3.4.2, are shown in Table 3.C.2. CE stages 0 denotes a regular ResNet.

3.C.4 NEURON FEATURE VISUALIZATIONS

Fig. 3.C.2 shows the Neuron Feature [2] (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).

	Caltech-101	CIFAR-10	CIFAR-100	Flowers-102	Oxford Pet	Stanford Cars	STL-10	ImageNet
<i>Original test set</i>								
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 + J	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 + J	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 + J	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 + J	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 + J	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 + J	73.58 ± 0.68	-	-	74.17 ± 0.49	73.28 ± 0.63	79.79 ± 0.37	84.16 ± 0.10	65.57
Baseline + AM	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 + AM	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	-

Table 3.C.1: (continues on next page) Classification accuracy on various datasets. CEConv- s denotes a ResNet with s color equivariant stages, J denotes color jitter and AM denotes AugMix. 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.

	Caltech-101	CIFAR-10	CIFAR-100	Flowers-102	Oxford Pet	Stanford Cars	STL-10	ImageNet
<i>Hue-shifted test set</i>								
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
CEConv-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 + J	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
CEConv-W + J	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 + J	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 + J	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 + J	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 + J	73.57 ± 0.75	-	-	73.86 ± 0.39	72.94 ± 0.56	79.79 ± 0.34	84.02 ± 0.14	64.52
Baseline + AM	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 + AM	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 3.C.1: (continued from previous page) Classification accuracy on various datasets with gradual hue shifts applied at test time. (Hybrid) color equivariant networks improve performance over the baseline model on both the original as well as the hue-shifted test set.

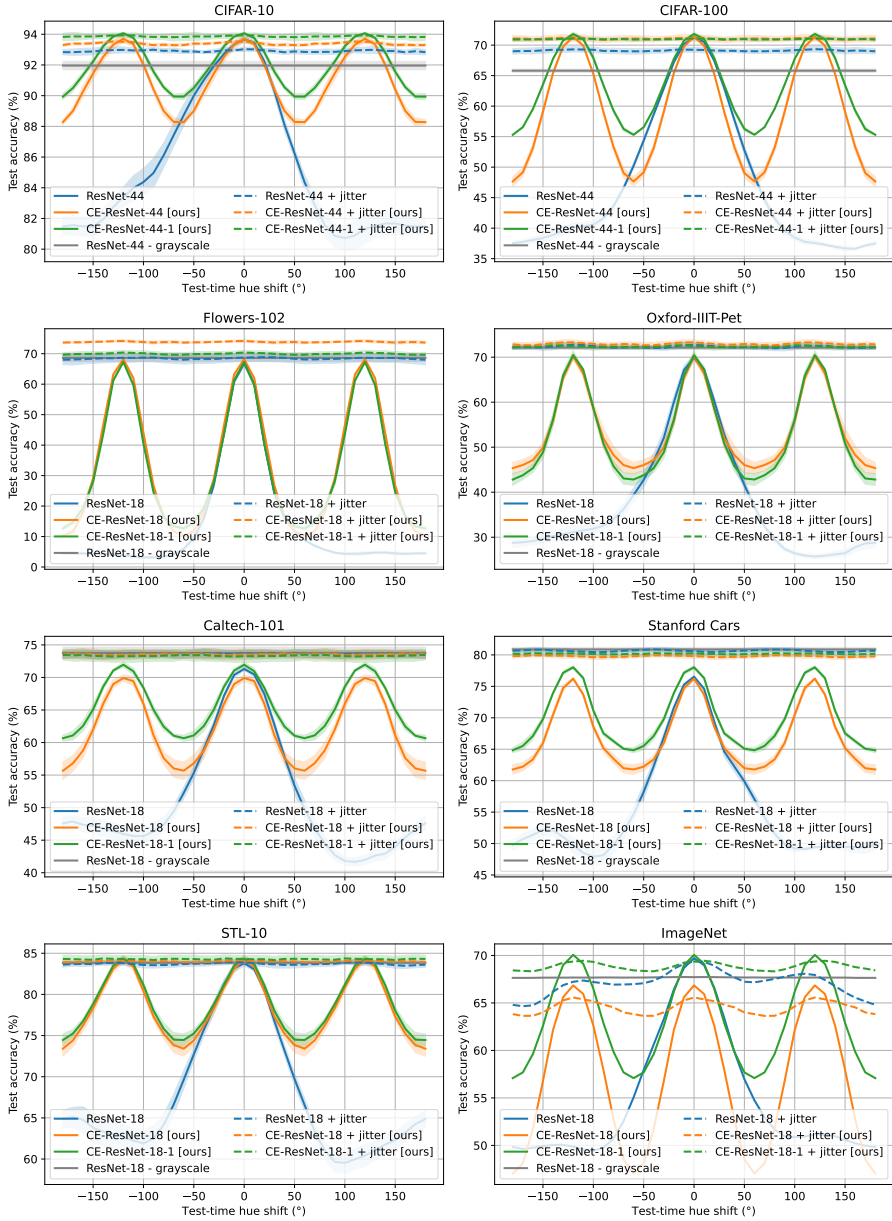


Figure 3.C.1: Test accuracy on various classification datasets under a test time hue shift.

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

3

Table 3.C.2: Color equivariant ResNet configurations. CE=0 denotes a regular ResNet.

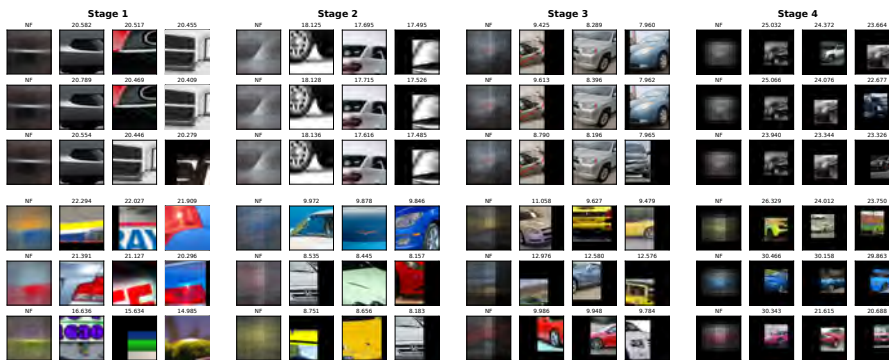


Figure 3.C.2: Neuron Feature [2] (NF) visualization with top-3 patches of two neurons at different stages in a CE-ResNet18 trained on Stanford Cars. Rows represent different rotations of the same filter.

3.D ABLATION STUDIES

Strength of color jitter augmentations Fig. 3.D.1 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.

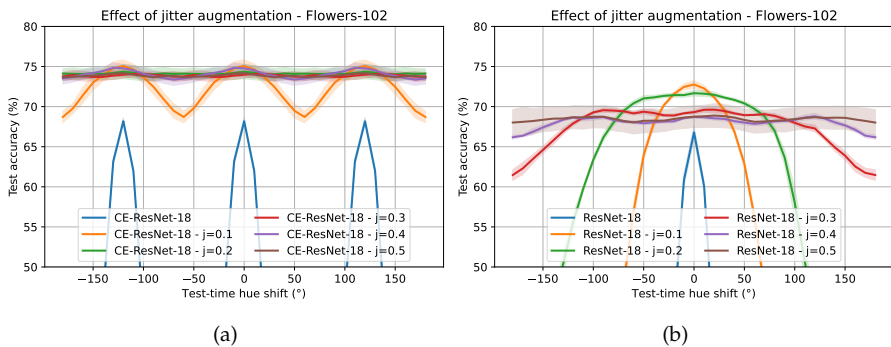


Figure 3.D.1: Effect of hue jitter augmentation on a color equivariant (a) and a regular (b) ResNet-18.

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. 3.D.2, the model without pooling layer is no longer invariant to hue shifts and behaves identically to the baseline model.

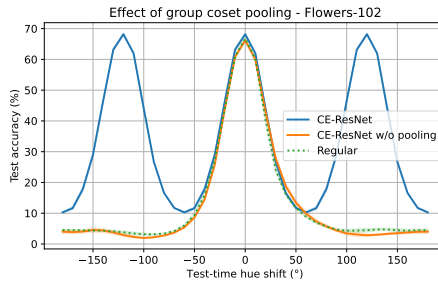


Figure 3.D.2: CE-ResNet without group coset pooling behaves similarly to a regular ResNet (average over 5 runs).

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. 3.D.3 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 the 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. 3.D.3 (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.

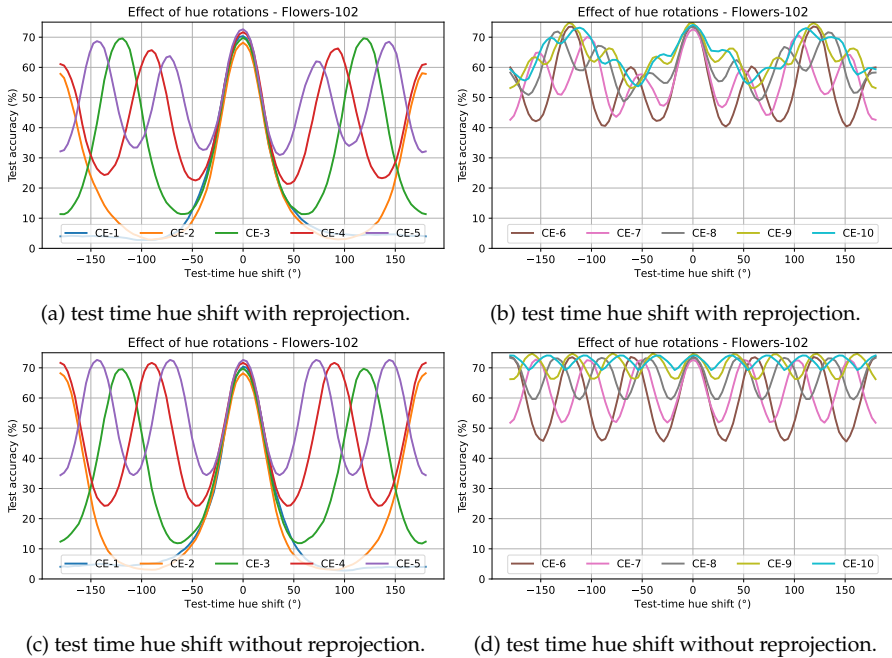


Figure 3.D.3: The effect of the number of hue rotations in color equivariant convolutions on downstream performance. More rotations increases robustness to test time hue shifts. Note that in (a-b) the peaks are not of equal height due to clipping effects near the boundaries of the RGB cube. This artifact disappears when the test time hue shift is also applied without projection, resulting in peaks of equal height (c-d).

REFERENCES

- [1] I. R. Cole. "Modelling CPV". In: (June 2015). URL: https://repository.lboro.ac.uk/articles/thesis/Modelling_CPV/9523520.
- [2] I. Rafegas and M. Vanrell. "Color encoding in biologically-inspired convolutional neural networks". In: *Vision Research* 151 (2018). Color: cone opponency and beyond, pp. 7–17. ISSN: 0042-6989. DOI: <https://doi.org/10.1016/j.visres.2018.03.010>. URL: <https://www.sciencedirect.com/science/article/pii/S0042698918300592>.

4

EXPLOITING LEARNED SYMMETRIES IN GROUP EQUIVARIANT CNNs

Group Equivariant Convolutions (GConvs) enable convolutional neural networks to be equivariant to various transformation groups, but at an additional parameter and compute cost. We investigate the filter parameters learned by GConvs and find certain conditions under which they become highly redundant. We show that GConvs can be efficiently decomposed into depthwise separable convolutions while preserving equivariance properties and demonstrate improved performance and data efficiency on two datasets.

This chapter has been published as:

A. Lengyel and J. C. van Gemert. "Exploiting Learned Symmetries in Group Equivariant Convolutions". In: *2021 IEEE International Conference on Image Processing (ICIP)*, pp. 759–763. DOI: 10.1109/ICIP42928.2021.9506362.

Code available at:

<https://github.com/Attila94/SepGroupPy>

4.1 INTRODUCTION

Adding convolution to neural networks (CNNs) yields translation equivariance [1]: first translating an image x and then convolving is the same as first convolving x and then translating. Group Equivariant Convolutions [2] (GConvs) enable equivariance to a larger group of transformations G , including translations, rotations of multiples of 90 degrees ($p4$ group), and horizontal and vertical flips ($p4m$ group). Equivariance to a group of transformations G is guaranteed by sharing parameters between filter copies for each transformation in the group G . Adding such geometric symmetries as prior knowledge offers a hard generalization guarantee to all transformations in the group, reducing the need for large annotated datasets and extensive data augmentation.

In practice, however, GConvs occasionally learn filters that are near-invariant to transformations in G . An invariant filter is independent of the transformation and will for GConvs yield identical copies of the transformed filters in the consecutive layer, as shown in Fig. 4.1. This implies parameter redundancy, as these filters could be represented by a single spatial kernel. We propose an equivariant pointwise and a depthwise decomposition of GConvs with increased parameter sharing and thus improved data efficiency. Motivated by the observed inter-channel correlations in learned filters in [3] we explore additionally sharing the same spatial kernel over all input channels of a GConv filter bank. Our contributions are: (i) we show that near-invariant filters in GConvs yield highly correlated spatial filters; (ii) we derive two decomposed GConv variants; and (iii) improve accuracy compared to GConvs on RotMNIST and CIFAR10.

4.2 RELATED WORK

Equivariance in deep learning Equivariance is a promising research direction for improving data efficiency [4]. A variety of methods have extended the Group Equivariant Convolution for the $p4$ and $p4m$ groups introduced in [2] to larger symmetry groups including translations and discrete 2D rotations [5, 6], 3D rotations [7–9], and scale [10, 11]. Here, we investigate learned invariances in the initial GConv framework [2] for the $p4$ and $p4m$ groups, yet our analysis extends to other groups where invariant filters exist.

Depthwise separable decomposition [12] These decompose a multi-channel convolution into spatial convolutions applied on each individual input channel

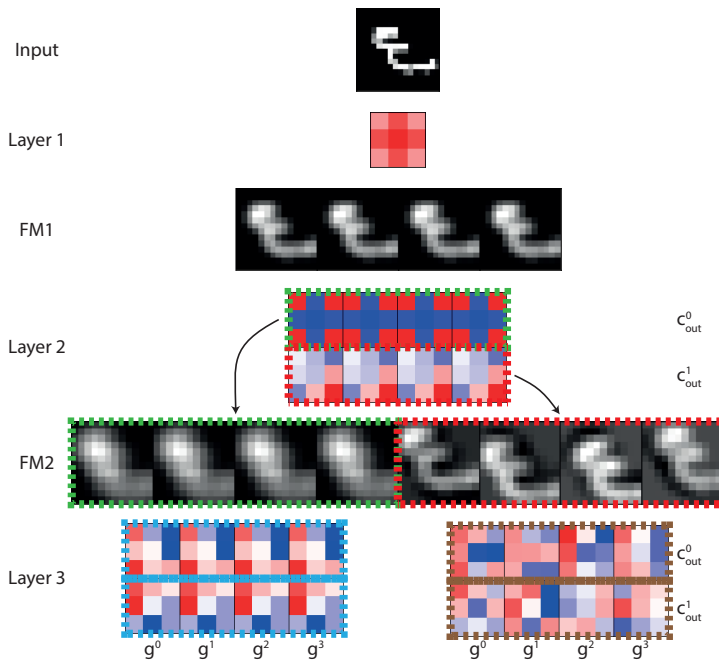


Figure 4.1: Filters and feature maps of a GConv architecture trained on Rotated MNIST. Rotation invariant filters in Layer 2 result in identical feature maps FM2 (green) and cause Layer 3 to learn identical weights along the group dimension g (blue). In contrast, non-symmetric filters in Layer 2 (red) result in non-identical filters in Layer 3 (brown).

separately, followed by a pointwise (1x1) convolution. Depthwise separable convolutions significantly reduce parameter count and computation cost at the expense of a slight loss in representation power and therefore generally form the basis of network architectures optimized for efficiency [13–15]. The effectiveness of depthwise separable convolutions is motivated [3] by the observed inter-channel correlations occurring in the learned filter banks of a CNN, which is quantified using a PCA decomposition. We do a similar analysis to motivate and derive our separable implementation of GConvs. While such implementation is briefly mentioned in [15], it lacks any explanation or analysis provided in our work.

4.3 METHOD

4.3.1 GROUP EQUIVARIANT CONVOLUTIONS

Equivariance to a group of transformations G is defined as

$$\Phi(T_g x) = T'_g \Phi(x), \quad \forall g \in G, \quad (4.1)$$

where Φ denotes a network layer and T_g and T'_g a transformation g on the input and feature map, respectively. Note that in the case of translation equivariance T and T' are the same, but in general not need to be. To simplify the explanation, we focus on the $p4$ group of translations and 90-degree rotations. Let us denote a regular convolution as

$$X_{n,:,:,}^{l+1} = \sum_c^{C^l} F_{n,c,:,:,}^l * X_{c,:,:,}^l, \quad (4.2)$$

with X the input and output tensors of size $[C^l, H, W]$, where C^l is the number of channels in layer l , H is height and W is width, and F the filter bank of size $[C^{l+1}, C^l, k, k]$, with k the spatial extent of the filter. In addition to spatial location, GConvs encode the added transformation group G in an extra tensor dimension such that X becomes of size $[C^l, G^l, H, W]$, where G^l denotes the size of the transformation group G at layer l , i.e. 4 for the $p4$ group. Likewise, GConv filters acting on these feature maps contain an additional group dimension, yielding a filter bank F^l of size $[C^{l+1}, C^l, G^l, k, k]$. As such, filter banks in GConvs contain G^l times more trainable parameters compared to regular convolutions. A GConv is then performed by convolving over both the input channel and input group dimensions C^l and G^l and summing up the outputs:

$$X_{n,h,:,:,}^{l+1} = \sum_c^{C^l} \sum_g^{G^l} \tilde{F}_{n,h,c,g,:,:,}^l * X_{c,g,:,:,}^l, \quad (4.3)$$

Here \tilde{F}^l denotes the full GConv filter of size $[C^{l+1}, G^{l+1}, C^l, G^l, k, k]$ containing an additional dimension for the output group G^{l+1} . \tilde{F}^l is constructed from F^l during each forward pass, where G^{l+1} contains rotated and cyclically permuted versions of F^l (see [2] for details). Note that input images do not have a group dimension, so the input layer has $G^l=1$ and $X_{c,g,:,:,}^1$ reduces to $X_{c,:,:,}^1$, whereas for all following layers $G^l=4$ for the $p4$ group (and $G^l=8$ for $p4m$).

4.3.2 FILTER REDUNDANCIES IN GCONVS

A rotational symmetric filter is invariant to the relative orientation between the filter and its input. Thus, if the filter kernels in the group dimension of a $p4$ GConv filter bank F^l are rotational symmetric and identical, the resulting feature maps will also be identical along the group dimension due to the rotation and cyclic permutation performed in constructing the full filter bank \tilde{F}^l . As a result, the filters in the subsequent layer acting on these feature maps receive identical gradients and, given same initialization, learn identical filters. This is illustrated in Fig. 4.1, where a $p4$ equivariant CNN is trained on Rotated MNIST. The first layer contains a single fixed rotation invariant filter. All layers have equal initialization along the group dimension and linear activation functions. The filters in layer 2 converge to be identical along the group dimension. Furthermore, the filter kernels in the second layer belonging to the first output channel (green) are also rotational symmetric, resulting in identical feature maps in FM2 (green) and consequently the filters learned in the first input channel of layer 3 (blue) become highly similar. This is in contrast to the non-symmetric filters in layer 2 (red), resulting in non-identical filters in layer 3 (brown).

Even non-rotational symmetric filters can induce filter correlations in the subsequent layer. For instance, an edge detector will result in inverse feature maps along the group dimension, i.e. $g^0 \approx -g^2$ and $g^1 \approx -g^3$ and the filters acting on these feature maps will receive inverse gradients and consequently converge to be inversely correlated. Inversely correlated filters can be decomposed into the same spatial kernel multiplied by a positive and negative scalar.

Upon visual inspection of the learned filter parameters of a regular $p4$ equivariant CNN we observe that, even without any fixed symmetries or initialization and with ReLU activation functions, the filter kernels tend to be correlated along the group axis. To quantify this correlation we perform a PCA decomposition similar as in [3]. We reshape the filter bank F to size $[C^{l+1} \times C^l, G^l, k^2]$ and perform PCA on each set of filters $F_{n,:}$ for all $n \in [1, C^{l+1} \times C^l]$, where for each n we have G^l features with k^2 samples. This results in G^l principal components of size k^2 , with PC1 being the filter kernel explaining the most variance within the decomposed set. We perform this decomposition for all layers in a $p4$ equivariant network. Fig. 4.2 shows the ratio of the variance explained by PC1 for each layer (after the input layer), before and after training. In many cases a substantial part of the variance is explained by a single component, demonstrating a significant redundancy in filter parameters.

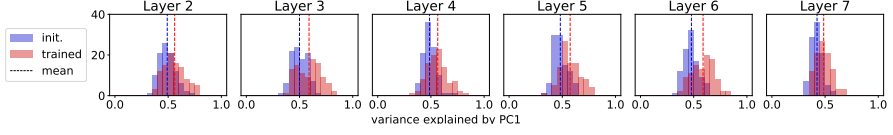


Figure 4.2: Ratio of variance explained by the first principal component when decomposing a filter kernel along the group dimension, before (blue) and after (red) training on Rotated MNIST. Redundancy in filter parameters increases as the network converges.

4

4.3.3 SEPARABLE GROUP EQUIVARIANT CONVOLUTIONS

To exploit the correlations in GConvs we decompose the filter bank F^l into a 2D kernel K that is shared along the group dimension, and a pointwise component w which encodes the inter-group correlations:

$$F_{n,c,g,:}^l = K_{n,c,:}^l \cdot w_{n,c,g}^l \quad (4.4)$$

The full GConv filter bank is then constructed as

$$\tilde{F}_{n,h,c,g,:}^l = T_h(K_{n,c,:}^l) \cdot \tilde{w}_{n,h,c,g}^l \quad (4.5)$$

where T_h denotes the 2D transformation corresponding to output group channel h and \tilde{w}^l contains copies of w^l that are cyclically permuted along the input group dimension. A naive implementation would be to precompute \tilde{F} and perform a regular GConv as in Eq. (4.3). Alternatively, for better computational efficiency we can substitute the filter decomposition in Eq. (4.5) into the GConv in Eq. (4.3) and rearrange as follows:

$$X_{n,h,:}^{l+1} = \sum_c^{C^l} \sum_g^{G^l} X_{c,g,:}^l * \left(T_h(K_{n,c,:}^l) \cdot \tilde{w}_{n,h,c,g}^l \right) \quad (4.6)$$

$$= \sum_c^{C^l} \sum_g^{G^l} \left(X_{c,g,:}^l \cdot \tilde{w}_{n,h,c,g}^l \right) * T_h(K_{n,c,:}^l) \quad (4.7)$$

$$= \sum_c^{C^l} \tilde{X}_{n,h,c,:}^l * T_h(K_{n,c,:}^l) \quad (4.8)$$

with

$$\tilde{X}_{n,h,c,:}^l = \sum_g^{G^l} \left(X_{c,g,:}^l \cdot \tilde{w}_{n,h,c,g}^l \right). \quad (4.9)$$

Expanding the dimensions of \tilde{w}^l to $[C^{l+1}, G^{l+1}, C^l, G^l, 1, 1]$ we can implement Eq. (4.9) as a grouped 1×1 convolution with C^l groups, followed by a grouped spatial convolution with $C^{l+1} \times G^{l+1}$ groups, as given in Eq. (4.8). We refer to this separable GConv variant as g -GConv, denoting the summation variable in Eq. (4.9).

Alternatively, we share the spatial kernel K along both the group and input channel dimension by decomposing F^l as:

$$F_{n,c,g,:}^l = K_{n,:}^l \cdot w_{n,c,g}^l, \quad (4.10)$$

$$\tilde{F}_{n,h,c,g,:}^l = T_h(K_{n,:}^l) \cdot \tilde{w}_{n,h,c,g}^l. \quad (4.11)$$

Substituting \tilde{F}^l in Eq. (4.3) and rearranging yields

$$X_{n,h,:}^{l+1} = \sum_c^{C^l} \sum_g^{G^l} X_{c,g,:}^l * \left(T_h(K_{n,:}^l) \cdot \tilde{w}_{n,h,c,g}^l \right) \quad (4.12)$$

$$= \sum_c^{C^l} \sum_g^{G^l} \left(X_{c,g,:}^l \cdot \tilde{w}_{n,h,c,g}^l \right) * T_h(K_{n,:}^l) \quad (4.13)$$

$$= \tilde{X}_{n,h,:}^l * T_h(K_{n,:}^l) \quad (4.14)$$

with

$$\tilde{X}_{n,h,:}^l = \sum_c^{C^l} \sum_g^{G^l} \left(X_{c,g,:}^l \cdot \tilde{w}_{n,h,c,g}^l \right). \quad (4.15)$$

This way the GConv essentially reduces to an inverse depthwise separable convolution with Eq. (4.15) being the pointwise and Eq. (4.14) being the depthwise component. This variant is named gc -GConv after the summation variables in Eq. (4.15).

While the g and gc decompositions may impose too stringent restrictions on the hypothesis space of the model, the improved parameter efficiency, as detailed in Section 4.3.4, allows us to increase the network width given the same parameter budget resulting in better overall performance.

4.3.4 COMPUTATION EFFICIENCY

The decomposition of GConvs allows for a theoretically more efficient implementation, both in terms of the number of stored parameters and multiply-accumulate operations (MACs). As opposed to the $[C^{l+1} \times C^l \times G^l \times k^2]$ parameters in a GConv filter bank, g - and gc -GConvs require only $[C^{l+1} \times C^l \times (G^l + k^2)]$ and $[C^{l+1} \times (C^l \times G^l + k^2)]$, respectively. Similarly, a regular GConv layer performs $[C^{l+1} \times G^{l+1} \times C^l \times G^l \times k^2 \times H \times W]$ MACs, whereas g - and gc -GConvs do only $[C^{l+1} \times G^{l+1} \times C^l \times (G^l + k^2) \times H \times W]$ and $[C^{l+1} \times G^{l+1} \times (C^l \times G^l + k^2) \times H \times W]$, assuming ‘same’ padding. This translates to a reduction by a factor of $\frac{1}{k^2} + \frac{1}{G^l}$ and $\frac{1}{k^2} + \frac{1}{C^l \times G^l}$, both in terms of parameters and MAC operations. The decrease in MACs comes at the cost of a larger GPU memory footprint due to the need of storing intermediate feature maps, as is generally the case for separable convolutions. Separable GConvs are therefore especially suitable for applications where the available processing power is the bottleneck as opposed to memory.

4

4.4 EXPERIMENTS

4.4.1 ROTATED MNIST

We construct a g -separable (Eqs. 4.8-4.9) and gc -separable (Eqs. 4.14-4.15) version of the P4CNN architecture [2] and evaluate on Rotated MNIST [16]. Rotated MNIST has 10 classes of randomly rotated handwritten digits with 12k train and 60k test samples. We set the width w of the g -P4CNN and gc -P4CNN networks such that the number of parameters are as close as possible to our Z2CNN and P4CNN baselines of 20 and 10 channels, respectively. We follow the training procedure of [2] and successfully reproduced the results.

Table 4.1 shows the test error averaged over 5 runs. Both g - and gc -P4CNN significantly outperform the regular P4CNN architecture and perform comparably or better than other architectures with a similar parameter count. Both g - and gc -P4CNN also outperform a depthwise separable version of Z2CNN (c -Z2CNN), validating that GConvs are more efficiently decomposable than regular convolutions. Additionally, we evaluate data-efficiency in a reduced data setting. As Fig. 4.3a shows, both g - and gc -P4CNN consistently outperform P4CNN. Sharing the same 2D kernel in a GConv filter bank is thus a strong inductive bias and improves the model’s sample efficiency. The test error as a function of number of parameters is also shown in Fig. 4.3b. Separable

GConvs do better for all model capacities.

Network	Test error	w	Param.	MACs
Z2CNN [2]	5.20 ± 0.110	20	25.21 k	2.98 M
c -Z2CNN	4.64 ± 0.126	57	25.60 k	4.14 M
P4CNN [2]	2.23 ± 0.061	10	24.81 k	11.67 M
g -P4CNN [ours]	2.60 ± 0.098	10	8.91 k	4.37 M
gc -P4CNN[ours]	2.88 ± 0.169	10	3.42 k	1.80 M
g -P4CNN [ours]	1.97 ± 0.044	17	25.26 k	12.34 M
gc -P4CNN [ours]	1.74 ± 0.070	30	24.64 k	13.01 M
SFCNN [6]	0.71 ± 0.022	-	-	-
DREN [17]	1.56	-	25 k	-
H-Net [18]	1.69	-	33 k	-
α -P4CNN [19]	1.70 ± 0.021	10	73.13 k	-
a -P4CNN [20]	2.06 ± 0.043	-	20.76 k	-

Table 4.1: Test error on Rotated MNIST - comparison with $z2$ baseline and other $p4$ -equivariant methods. w denotes network width. Separable GConv architectures perform better compared to regular GConvs (top part) and comparable to other equivariant methods (bottom part).

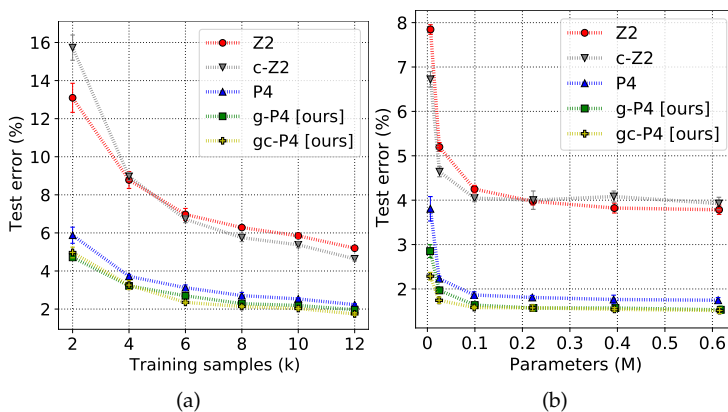


Figure 4.3: Test error on Rotated MNIST for varying training set (a) and model sizes (b). Architectures with separable GConvs perform consistently better.

4.4.2 CIFAR 10

Similarly, we perform a benchmark on the CIFAR10 dataset [21] using a $p4m$ equivariant version of ResNet44 as detailed in [2]. CIFAR 10+ denotes moderate data augmentation including random horizontal flips and random translations of up to 4 pixels. Our $gc-p4m$ -ResNet44 outperforms all other methods using fewer parameters, as shown in Table 4.2. Also in a low data regime using only 20% of the training samples our $gc-p4m$ architecture outperforms the regular $p4m$ network with an error rate of 13.43% vs. 14.20%.

Network	CIFAR10	CIFAR10+	Param.
ResNet44 [†] [2]	13.10	7.66	2.64M
$p4m$ -ResNet44 [‡] [2]	8.06	5.78	2.62M
α_F - $p4m$ -ResNet44 [19]	10.82	10.12	2.70M
a - $p4m$ -ResNet44 [20]	9.12	-	2.63M
g - $p4m$ -ResNet44 [ours]	7.60	6.09	1.78M
gc - $p4m$ -ResNet44 [ours]	6.72	5.43	1.88M

^{†‡} Unable to reproduce results from [2]: 9.45 / 5.61[†], 6.46 / 4.94[‡].

Table 4.2: Test error on CIFAR10 - comparison with other $p4m$ -equivariant methods. gc - $p4m$ -ResNet44 performs best.

4.5 DISCUSSION

Our method exploits naturally occurring symmetries in GConvs by explicit sharing of the same filter kernel along the group and input channel dimension using a pointwise and depthwise decomposition. Experiments show that imposing such restriction on the architecture only causes a minor performance drop while allowing to significantly reduce the network parameters. This in turn (i) improves data efficiency and (ii) allows to increase the network width for the same parameter budget resulting in better overall performance. Sharing the spatial kernel over only the group dimension (g) proves less effective than additionally sharing over input channels (gc) as the latter also efficiently exploits inter-channel correlations in the network. This allows to further increase the network width and thereby its representation power.

REFERENCES

- [1] O. Kayhan and J. C. van Gemert. “On Translation Invariance in CNNs: Convolutional Layers can Exploit Absolute Spatial Location”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020.
- [2] T. S. Cohen and M. Welling. “Group Equivariant Convolutional Networks”. In: *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*. ICML’16. New York, NY, USA: JMLR.org, 2016, pp. 2990–2999.
- [3] D. Haase and M. Amthor. “Rethinking Depthwise Separable Convolutions: How Intra-Kernel Correlations Lead to Improved MobileNets”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2020.
- [4] M. Rath and A. Condurache. “Boosting Deep Neural Networks with Geometrical Prior Knowledge: A Survey”. In: *ArXiv abs/2006.16867* (2020).
- [5] E. J. Bekkers, M. W. Lafarge, M. Veta, K. A. J. Eppenhof, J. P. W. Pluim, and R. Duits. “Roto-Translation Covariant Convolutional Networks for Medical Image Analysis”. In: *Medical Image Computing and Computer Assisted Intervention – MICCAI 2018*. Springer International Publishing, 2018, pp. 440–448. ISBN: 978-3-030-00928-1.
- [6] M. Weiler, F. A. Hamprecht, and M. Storath. “Learning Steerable Filters for Rotation Equivariant CNNs”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2018.
- [7] M. Winkels and T. S. Cohen. “Pulmonary nodule detection in CT scans with equivariant CNNs”. In: *Medical Image Analysis* 55 (2019), pp. 15–26. ISSN: 1361-8415. DOI: <https://doi.org/10.1016/j.media.2019.03.010>. URL: <http://www.sciencedirect.com/science/article/pii/S136184151830608X>.
- [8] D. E. Worrall and G. J. Brostow. “CubeNet: Equivariance to 3D Rotation and Translation”. In: *Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part V*. 2018, pp. 585–602. DOI: 10.1007/978-3-030-01228-1_35.
- [9] M. Weiler, M. Geiger, M. Welling, W. Boomsma, and T. S. Cohen. “3D Steerable CNNs: Learning Rotationally Equivariant Features in Volumetric Data”. In: *Advances in Neural Information Processing Systems*. Vol. 31. Curran Associates, Inc., 2018, pp. 10381–10392.
- [10] D. Worrall and M. Welling. “Deep Scale-spaces: Equivariance Over Scale”. In: *Advances in Neural Information Processing Systems*. Vol. 32. Curran Associates, Inc., 2019, pp. 7366–7378.

- [11] I. Sosnovik, M. Szmaja, and A. Smeulders. “Scale-Equivariant Steerable Networks”. In: *International Conference on Learning Representations*. 2020. URL: <https://openreview.net/forum?id=HJgpugrKPS>.
- [12] L. Sifre and P. S. Mallat. *Ecole Polytechnique, CMAP PhD thesis Rigid-Motion Scattering For Image Classification* Author: 2014.
- [13] A. Howard, M. Sandler, G. Chu, L.-C. Chen, B. Chen, M. Tan, W. Wang, Y. Zhu, R. Pang, V. Vasudevan, Q. V. Le, and H. Adam. “Searching for MobileNetV3”. In: *2019 IEEE/CVF International Conference on Computer Vision (ICCV) (2019)*, pp. 1314–1324.
- [14] M. Tan and Q. Le. “EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks”. In: *Proceedings of the 36th International Conference on Machine Learning*. Vol. 97. Proceedings of Machine Learning Research. PMLR, 2019, pp. 6105–6114.
- [15] M. Mohamed, G. Cesa, T. S. Cohen, and M. Welling. *A Data and Compute Efficient Design for Limited-Resources Deep Learning*. 2020. arXiv: 2004.09691 [cs.LG].
- [16] H. Larochelle, D. Erhan, A. Courville, J. Bergstra, and Y. Bengio. “An Empirical Evaluation of Deep Architectures on Problems with Many Factors of Variation”. In: *Proceedings of the 24th International Conference on Machine Learning*. ICML '07. Corvallis, Oregon, USA: Association for Computing Machinery, 2007, pp. 473–480. ISBN: 9781595937933. DOI: 10.1145/1273496.1273556. URL: <https://doi.org/10.1145/1273496.1273556>.
- [17] J. Li, Z. Yang, H. Liu, and D. Cai. “Deep rotation equivariant network”. In: *Neurocomputing* 290 (2018), pp. 26–33.
- [18] D. E. Worrall, S. J. Garbin, D. Turmukhambetov, and G. J. Brostow. “Harmonic Networks: Deep Translation and Rotation Equivariance”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. July 2017.
- [19] D. Romero, E. Bekkers, J. Tomczak, and M. Hoogendoorn. “Attentive Group Equivariant Convolutional Networks”. In: *Proceedings of the 37th International Conference on Machine Learning*. Vol. 119. Proceedings of Machine Learning Research. PMLR, July 2020, pp. 8188–8199. URL: <http://proceedings.mlr.press/v119/romero20a.html>.
- [20] D. W. Romero and M. Hoogendoorn. “Co-Attentive Equivariant Neural Networks: Focusing Equivariance On Transformations Co-Occurring in Data”. In: *International Conference on Learning Representations*. 2020. URL: <https://openreview.net/forum?id=r1g6ogrtDr>.
- [21] A. Krizhevsky. “Learning Multiple Layers of Features from Tiny Images”. In: (2009), pp. 32–33. URL: <https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf>.

5

USING AND ABUSING EQUIVARIANCE

This study explores how Group Equivariant Convolutional Neural Networks (GCNNs) leverage subsampling to learn and break equivariance to 2D rotation and reflection symmetries. We show that changing the input size by as little as a single pixel can cause GCNN architectures to shift from exact to approximate equivariance, and vice versa. We furthermore assess the effects of exact versus approximate equivariance on downstream performance. The findings show that approximately equivariant architectures have poorer generalization capabilities to transformations not seen during training compared to networks that are exactly equivariant. However, when the data does not reflect the symmetries embedded in the equivariant architecture, these approximately equivariant networks have the ability to relax their constraints on equivariance. This flexibility allows them to match or even surpass the performance of exactly equivariant networks, as we demonstrate on common classification benchmarks.

This chapter has been published as:

T. Edixhoven, A. Lengyel, and J. C. van Gemert. "Using and Abusing Equivariance". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, Oct. 2023, pp. 119–128.

Code available at:

https://github.com/TFedixhoven/using_equivariance

5.1 INTRODUCTION

Nature contains lots of symmetries [1], and neural networks used in computer vision have been shown to benefit greatly from prior knowledge of these symmetries. Most notably, the introduction of the convolution operator resulted in the creation of *Convolutional Neural Networks* [2] (CNNs), which form the backbone of many computer vision applications. Convolutions are *equivariant* to the translation symmetry, meaning that if an object in the input image is shifted, the output of the convolution is shifted equally. Due to translation equivariance, networks no longer have to explicitly learn to recognize objects at all possible locations, as the knowledge that location plays no role is embedded into the network.

However, images often contain other relevant symmetries for which CNNs are not equivariant. Take for example the field of histopathology, which entails the microscopic examination of organic tissue. In histopathology, the rotational orientation of the tissue is arbitrary [3]. A network that varies its output when the input is rotated is therefore a cause for uncertainty. More formally, the output of the network should be *invariant* to rotation, meaning that the output should not change when the input is rotated.

A major innovation in equivariance for computer vision was the introduction of *Group Equivariant Convolutions* [4] (GECs), which made it possible for CNNs to guarantee equivariance or invariance to a finite group of discrete transformations, also referred to as a *symmetry group*. Using GECs instead of standard convolutions to create a network yields a *Group Equivariant Convolutional Neural Network* (GCNN). Due to the group equivariant properties of GECs, GCNNs guarantee that the network output does not change when the input is rotated.

In this paper, we explore subsampling layers in GCNNs that allow the networks to break their guarantee of equivariance. Consider the *MaxPool* subsampling layer in Fig. 5.1. The feature map resulting from first rotating and then subsampling contains different numerical values than the result of first subsampling and then rotating, and as such the *MaxPool* layer is not equivariant to rotations. Whether a subsampling layer breaks equivariance is dependent on the width and height of the input, also referred to as *input dimension*. Including a subsampling layer that breaks equivariance in a GCNN will void the entire GCNN's guarantee of equivariance. However, subsampling layers are deemed almost essential for computer vision models and are used in nearly all GCNNs and modern CNNs. Typically, no distinction is made between GCNNs that do or do not contain subsampling layers that break equivariance. In this work, we show why a distinction should be made. We

refer to networks in which subsampling layers break the guarantee of equivariance as *approximately equivariant*, while networks in which the guarantee is not broken are referred to as *exactly equivariant*.

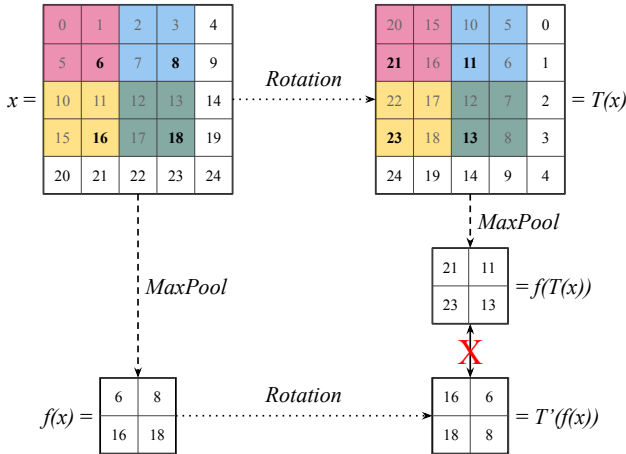


Figure 5.1: Example of how subsampling can break equivariance. Dotted arrows indicate a *Rotation* and the dashed arrows indicate a *MaxPool* subsampling layer with a kernel size and stride of 2. The locations where pooling is applied are colored. One can see that $f(T(x))$ and $T'(f(x))$ contain different numerical values, breaking equivariance.

We offer the following contributions: (i) We give a formal definition of exact equivariance under subsampling and analyze when equivariance is broken; (ii) we show that approximately equivariant networks learn to become less equivariant and as a result generalize significantly worse to unseen symmetries compared to their exact counterparts; and (iii) we show that slightly changing the input dimensions is often enough to make a network exactly equivariant rather than approximately equivariant.

5.2 RELATED WORK

5.2.1 EQUIVARIANCE IN DEEP LEARNING

CNNs can learn to become equivariant from data [5, 6]. However, this provides no general equivariance guarantees and results in redundant filters in the network. For example, the network can learn separate filters for detecting horizontal and vertical lines, rather than learning a single filter to detect lines of any rotation. Excellent works have investigated how to efficiently learn equivariance to relevant symmetries during training, either by separating symmetry weights from filter weights [7–9], using contrastive learning [10, 11] or using marginal likelihood [12, 13]. However, while these methods significantly increase a network’s ability to become equivariant, they do not guarantee it, as each method relies on the network learning the equivariance from the training data. However, the training data seldom guarantees a full and uniformly distributed representation of the relevant symmetries. These possible biases in the training data can then propagate into biases in the network. This can be cause for concern, as biased networks make systematic errors due to faulty assumptions about the data.

5

If the symmetry group for which a network needs to be equivariant is known, a common solution is to encode the symmetries into the network as prior knowledge using Group Equivariant Convolutions [4] (GECs). GECs are equivariant to a finite set of discrete transformations defined in a symmetry group. Filter weights are then shared within the GECs according to the transformations. Because the GECs include all transformations from their symmetry group, GECs guarantee equivariance to the symmetries, regardless of biases in the training data.

The introduction of GECs kick-started much follow-up work, including extensions from 2D planes to 3D manifolds [14–16] and the generalization from discrete to continuous transformations using Lie algebra [17] or other means [18]. In this work, we focus on using GECs that are equivariant to the 2D roto-translation group, as the group has been proven to be useful in the field of histopathology [3, 19, 20] and processing satellite data [21, 22]. The 2D roto-translation group consists of all rotations and translations in a 2-dimensional space. However, as GECs are equivariant to discrete transformations we limit ourselves to the $p4$ group consisting of all compositions of translations and multiples of 90° degree rotations.

5.2.2 BREAKING EQUIVARIANCE

While CNNs are generally regarded to be translation equivariant, a plethora of work has shown that this is not always the case. Convolutions and pooling with a stride larger than 1 have been shown to break translation equivariance [23–25]. CNNs have also been shown to be able to learn absolute location, which equally voids translation equivariance [26]. This is important to note, as Group Equivariant Convolutions assume that standard convolutions are translation equivariant to prove their equivariance to other transformations. Preventing networks from breaking their roto-translation equivariance has been investigated for reconstruction learning by introducing a group equivariant subsampling layer [25]. This method however requires additional compute and the effects on classification performance have not been investigated, where invariance is often more desirable than equivariance. In this work, we extend the current literature by investigating the influence of subsampling on rotation equivariance for classification.

The general proof for equivariance in GCNNs holds when the convolution operation spans the entire input. However, networks not always satisfy this assumption. Pooling and strided convolutions are used to aggregate local information and increase the receptive field of a network [27]. The combination of stride, input size and kernel size in subsampling layers can result in different indices being sampled from the input feature map, in turn resulting in approximate equivariance rather than exact equivariance [28]. While this might seem like a minute detail, we find that it negatively affects a GCNN’s downstream performance. Other examples of rotation equivariant GCNNs exhibiting unexpected behavior can be found in [29, 30]. In this work, we show that equivariance in GCNNs can be guaranteed by introducing a relatively simple restriction on the combination of input size, kernel size and stride.

5.2.3 RELAXING EQUIVARIANT CONSTRAINTS

Recent work has shown the possible benefits of relaxing equivariance constraints, demonstrating improved performance by equipping networks with less strict forms of equivariance [31, 32]. Similarly, approximate equivariance also enables networks to relax their equivariance constraints. As we will show, setting the appropriate input size to the network allows the computer vision practitioner to make a conscious decision on whether to relax the equivariance constraints on a network.

5.3 HOW SUBSAMPLING BREAKS EQUIVARIANCE

In this section we provide a more formal introduction to equivariance and Group Equivariant Convolutions (GConvs). We will then show a simple network configuration including a subsampling operation that breaks the equivariance property of GConvs. Subsequently we introduce a constraint on the network configuration that does guarantee exact equivariance under subsampling, and provide a proof for rotations and mirroring.

5.3.1 GROUP EQUIVARIANT CONVOLUTIONS

A network f is equivariant to transformation T , when the output of f on input x changes predictably when x is transformed by T . More formally, there exists a transformation T' for which the following equality holds:

$$f(T(x)) = T'(f(x)). \quad (5.1)$$

GCNNs are equivariant to a set of transformations defined in a symmetry group G , where in practice the transformations are stored in an additional group dimension in the feature maps. In the case of the $p4$ -group, T' consists of a rotation in the spatial dimensions and permutation of the group dimension on the feature map. Invariance to T is achieved by applying a coset pooling operation on the group dimension, such that the final representation satisfies

$$f(T(x)) = f(x). \quad (5.2)$$

As such, invariance is considered a special case of equivariance.

5.3.2 EXACT AND INEXACT EQUIVARIANCE

From Eq. (5.1) and the cyclic permutation defined by T' it follows that the feature maps of an exactly equivariant GCNN should contain the same numerical values regardless of the applied rotation T to the input. However, layers that perform subsampling on the input can introduce numerical differences depending on the input transformation, resulting in the network architecture no longer being exactly equivariant, as we will now demonstrate.

Let x be a rectangular input of dimension $i = 5$, f a network consisting of a single *MaxPool* layer with kernel size $k = 2$ and stride $s = 2$, and T a clockwise rotation of 90° . As the reference point of the 2D subsampling operation is al-

ways defined at the $(0,0)$ index of the input, applying T results in the sampling indices being shifted by a single pixel from the perspective of the *MaxPool* layer. Subsequently, $T'(f(x))$ and $f(T(x))$ contain different numerical values, as shown in Fig. 5.1. To guarantee exact equivariance we therefore need to ensure that the same indices are sampled, irrespective of the order in which the sampling operation and rotation are applied. Our proposed solution is simple as it does not require any modifications to the network architecture and relies purely on setting appropriate input dimensions to the network. For comprehensibility, we focus on the case of square inputs $\in \mathbb{R}^{i \times i}$ and an arbitrary kernel size k and stride s , but the proof can be readily extended to rectangular inputs $\in \mathbb{R}^{j \times i}$.

A GCNN is exactly equivariant to rotations of multiples of 90° if the following equation holds for all layers in the network:

$$(i - k) \pmod s = 0. \quad (5.3)$$

5

We prove that Eq. (5.3) is a necessary condition for exact equivariance to 90° rotations by asserting that the sampled indices for a given output index remain the same under rotation. We define a new function called *index*, that returns the indices of the input values used by a convolutional or pooling layer to calculate the value located at index (x, y) in the output:

$$\text{index} \left(\begin{bmatrix} x \\ y \end{bmatrix} \right) = \left[\begin{bmatrix} sx \\ sy \end{bmatrix}, \begin{bmatrix} sx + k - 1 \\ sy + k - 1 \end{bmatrix} \right]. \quad (5.4)$$

Here s is the stride used for subsampling and k represents the kernel size. The output of the function is a square patch, denoted as $[\vec{u}, \vec{v}]$, where \vec{u} and \vec{v} represent the indices of the top left and bottom right corner, respectively. The sampled indices include all integer tuples within this patch. We also introduce the function R , which takes an index (x, y) as input and returns the indices rotated 90° counterclockwise:

$$R_n \left(\begin{bmatrix} x \\ y \end{bmatrix} \right) = \begin{bmatrix} y \\ n - 1 - x \end{bmatrix}, \quad (5.5)$$

where n indicates the width and height of the feature map in which the index (x, y) is located. We further generalize Eq. (5.5) to an input patch $[\vec{u}, \vec{v}]$ rather

than a single coordinate, resulting in Eq. (5.6):

$$\begin{aligned} R_n([\vec{u}, \vec{v}]) &= R_n\left(\left[\begin{bmatrix} x_1 \\ y_1 \end{bmatrix}, \begin{bmatrix} x_2 \\ y_2 \end{bmatrix}\right]\right) \\ &= \left[\begin{bmatrix} y_1 \\ n-1-x_2 \end{bmatrix}, \begin{bmatrix} y_2 \\ n-1-x_1 \end{bmatrix}\right] \end{aligned} \quad (5.6)$$

In the resulting output coordinates x_1 and x_2 get interchanged due to the counterclockwise rotation of the patch: the top left corner becomes the bottom left corner, while the bottom right corner becomes the top right corner.

Given that our layer takes a feature map of width and height i as input, we can write the width and height of the output feature map as

$$o = \lfloor \frac{i-k}{s} \rfloor + 1. \quad (5.7)$$

5

For a layer to be exactly equivariant, determining the sampled indices and then rotating should return the same result as rotating first and then determining the sampled indices, which we can formally denote as

$$\text{index}\left(R_o\left(\begin{bmatrix} x \\ y \end{bmatrix}\right)\right) = R_i\left(\text{index}\left(\begin{bmatrix} x \\ y \end{bmatrix}\right)\right). \quad (5.8)$$

To solve the left-hand side, we substitute Eq. (5.5) into Eq. (5.4), yielding

$$\begin{aligned} \text{index}\left(R_o\left(\begin{bmatrix} x \\ y \end{bmatrix}\right)\right) &= \text{index}\left(\begin{bmatrix} y \\ \lfloor \frac{i-k}{s} \rfloor - x \end{bmatrix}\right) \\ &= \left[\begin{bmatrix} sy \\ s\lfloor \frac{i-k}{s} \rfloor - sx \end{bmatrix}, \begin{bmatrix} sy+k-1 \\ s\lfloor \frac{i-k}{s} \rfloor - sx+k-1 \end{bmatrix}\right]. \end{aligned} \quad (5.9)$$

The same can be done for the right-hand side, by substituting Eq. (5.4) into Eq. (5.6), resulting in

$$\begin{aligned} R_i\left(\text{index}\left(\begin{bmatrix} x \\ y \end{bmatrix}\right)\right) &= R_i\left(\left[\begin{bmatrix} sx \\ sy \end{bmatrix}, \begin{bmatrix} sx+k-1 \\ sy+k-1 \end{bmatrix}\right]\right) \\ &= \left[\begin{bmatrix} sy \\ i-k-sx \end{bmatrix}, \begin{bmatrix} sy+k-1 \\ i-1-sx \end{bmatrix}\right]. \end{aligned} \quad (5.10)$$

Substituting Eqs. (5.9) and (5.10) into Eq. (5.8), we find two equations

$$s \lfloor \frac{i-k}{s} \rfloor - sx = i - k - sx, \quad (5.11)$$

$$s \lfloor \frac{i-k}{s} \rfloor - sx + k - 1 = i - 1 - sx. \quad (5.12)$$

Removing duplicate terms yields a single equation

$$s \lfloor \frac{i-k}{s} \rfloor = i - k, \quad (5.13)$$

which can be simplified to Eq. (5.3). As Eq. (5.3) holds for a rotation of 90° , it automatically holds for rotations of 180° and 270° , as these can be composed using multiple rotations of 90° . We provide a similar proof for mirroring in Appendix 5.A.

5.3.3 MEASURING EQUIVARIANCE AND INVARIANCE

We evaluate the exactness of the equivariance property of GConvs both in terms of their measured invariance and equivariance.

Measuring equivariance Since in GConvs both T and T' are known transformations, feature maps $f(T(x))$ and $T'(f(x))$ can be computed independently. We can thus define the equivariance error in terms of the Mean Squared Error (MSE) between the two feature maps:

$$\begin{aligned} \epsilon &= \text{MSE}(f(T(x)), T'(f(x))) \\ &= \frac{1}{N} \sum_{i,j,k} (f(T(x))_{ijk} - T'(f(x))_{ijk})^2, \end{aligned} \quad (5.14)$$

where i and j sum over the spatial dimensions of the feature map, k sums over the group dimension, and N is the total number of summed values. The equivariance error can be evaluated at any GConv layer in the network.

Measuring invariance To measure the invariance of the network output after coset pooling we apply a range of rotations between $[0^\circ, 360^\circ)$ to the test set and report the test accuracy for each set separately. However, rotating an image by degrees other than multiples of 90° introduces artifacts at the

corners of the image, as shown in Fig. 5.2 (left). These artifacts may have an additional detrimental effect on the network’s performance. To ensure we only measure the performance drop due to rotation, we apply a *CircleCrop*, which sets all values whose coordinates are not inside the largest possible inscribed circle to 0, as visualized in Fig. 5.2 (right). To prevent any domain shift between the train and test set, we apply *CircleCrop* during both training and evaluation. *Nearest Neighbor Interpolation* also affected model performance and so all rotations are performed using *Bilinear Interpolation*.



Figure 5.2: **Left:** Rotated input without *CircleCrop*. **Right:** Rotated input with *CircleCrop*. Without *CircleCrop* rotation introduces artifacts near the corners.

5

5.4 EXPERIMENTS

5.4.1 BREAKING EQUIVARIANCE

In this subsection, we show how networks can learn to break their equivariance to improve their performance, and demonstrate on commonly used classification datasets that they in practice indeed do so.

Can GCNNs break equivariance? If a GCNN is truly invariant, it should be unable to distinguish between an input x and $T(x)$, where T is a 90° rotation. We challenge this assumption by explicitly training a simple GCNN to differentiate between the two input samples x and $T(x)$. We construct a network consisting of (i) a GConv layer with $k = 3$, $s = 2$, 1 output channel and a padding of 1; (ii) a global average pooling layer over the spatial dimensions; (iii) a coset max pooling layer over the group dimension to obtain a rotation invariant representation; and (iv) a fully connected layer with two output features. We define an input $x_1 \in \mathbb{R}^{32 \times 32}$ as shown in the top left of Fig. 5.3 and train the network on x_1 and $T(x_1)$. We find that the network can perfectly distinguish between the two samples as shown by the feature maps in the

right column in Fig. 5.3. This demonstrates that the network is not invariant, despite the pooling operation on the group dimension. We furthermore define a second input $x_2 \in \mathbb{R}^{33 \times 33}$ and repeat the experiment. Now the network is not able to distinguish between x_2 and $T(x_2)$, showing that the same network architecture is exactly invariant for inputs in $\mathbb{R}^{33 \times 33}$, while not being invariant for inputs in $\mathbb{R}^{32 \times 32}$. This is in line with our findings in Section 5.3.2, as for $k = 3$, $s = 2$ and $i = 33$:

$$(i - k) \bmod s = (33 - 3) \bmod 2 = 0$$

holds, whereas for $i = 32$

$$(i - k) \bmod s = (32 - 3) \bmod 2 = 1 \neq 0$$

does not. Thus, GCNNs can break equivariance.

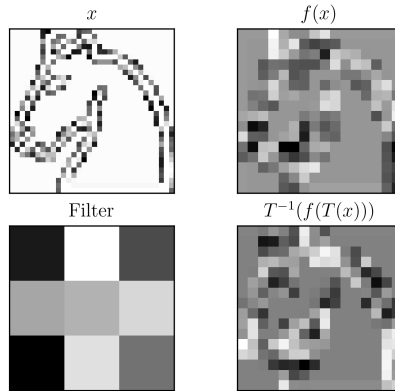


Figure 5.3: A subsampling Group Equivariant net f that is equivariant to the 90° rotation transformation T can learn a filter that returns almost inverted values for $f(x)$ and $T^{-1}(f(T(x)))$, while these outputs should be identical in theory. Because the outputs are dissimilar, network f can perfectly distinguish between x and $T(x)$.

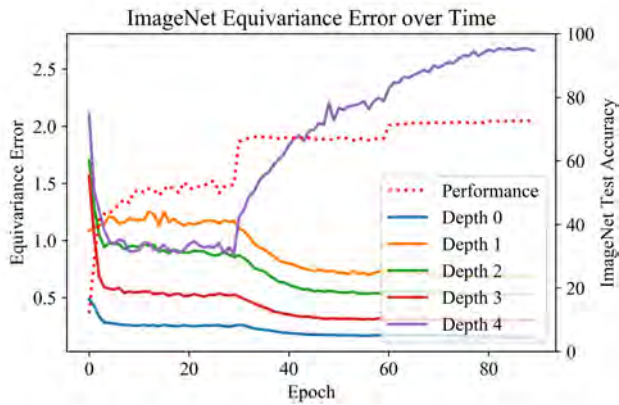
Breaking equivariance on common benchmarks. We have shown that when the objective of a GCNN is to break its equivariance, it will do so if possible. However, the question remains whether a network will also learn to do so when breaking equivariance is not explicitly the objective.

We first investigate the ImageNet [33] classification problem. We create a rotation equivariant ResNet18 [34] by substituting standard convolutions with $p4$ convolutions. The network width is divided by $\sqrt{4}$ to keep the number of parameters approximately equal to a standard ResNet18 and the input images are kept at their original 224×224 size. Training is performed for 90 epochs using the default training settings, i.e. SGD with momentum 0.9 and learning rate 0.1, which is step-wise reduced by a factor 0.1 every 30 epochs. The network is trained to classify the standard ImageNet classes, so there is no explicit objective to distinguish between rotations. Throughout training we monitor the equivariance error as defined in Eq. (5.14) after the first layer and each of the four ResNet stages. The measured equivariance errors are shown in Fig. 5.4a. We observe that initially all equivariance errors drop to a more or less constant value, and upon decreasing the learning rate at epoch 30 the equivariance error further drops in most stages in the network. However, the error at the last stage of the network increases rapidly, finally plateauing at a value higher than after random initialization.

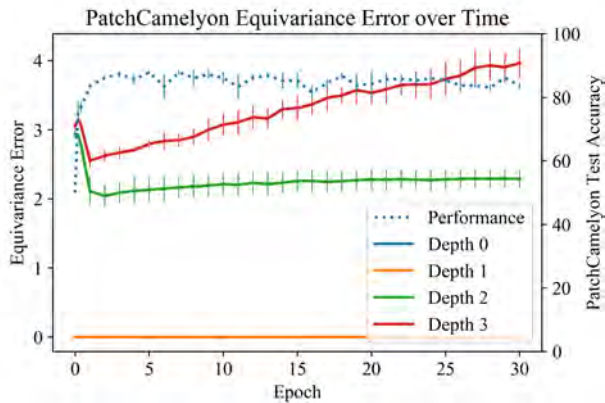
5

Secondly, we look at the PatchCamelyon dataset [20]. This dataset is interesting because it is pathology data, which, unlike ImageNet images, should not contain any dominant rotation bias. We use a setup similar to our previous ImageNet experiment, but we replace the ResNet18 with a ResNet44. The network width is decreased to obtain an approximately equal number of parameters as in the architecture used in [20] when evaluating on PatchCamelyon. The results on PatchCamelyon can be found in Fig. 5.4b. As the first layer and the first stage both have a stride of 1, the equivariance error is a constant 0 at the first two depth measurements. The other two stages show a similar behavior as in the ImageNet experiment, with a rapid decrease in the initial part of the training and a gradual increase in the final stage throughout the remainder of the training run. Interestingly, even for a rotation invariant problem the network still learns to break its equivariance.

We can thus conclude that a network that is equivariant to rotations can learn to abuse its approximate equivariance to become less equivariant. This occurs both when the network is trained on ImageNet, a dataset which contains only limited rotations and a clear upright orientation bias, as well as the PatchCamelyon dataset which should be rotation invariant by definition. A network learning differences between rotations in a rotation invariant setting is cause for concern, as the rotations are arbitrary and therefore should not contain any relevant information.



(a)



(b)

Figure 5.4: The measured equivariance error at different depths in a $p4$ -ResNet, trained on ImageNet (a) and PatchCamelyon (b), respectively. The classification accuracy is indicated as a dotted line. The equivariance error in the final layer increases throughout training, indicating that the network is learning to become less equivariant, even when trained on the rotation invariant PatchCamelyon dataset.

5.4.2 IMPACT OF EXACT EQUIVARIANCE

Performance on unseen rotations Due to the discrete nature of GCNNs, it is impossible to include all continuous rotations in the discrete group. Thus, it is important to generalize well to rotations that are not part of the group dimension. To compare how well approximately and exactly equivariant networks generalize to unseen rotations, we perform a controlled experiment on the MNIST [35] dataset of handwritten digits. Since MNIST contains limited rotations due to slanted handwriting, we are able to control what rotations are included during training and testing by transforming the data.

For this experiment, we use the Z2CNN and P4CNN architectures from [4]. The Z2CNN consists of 6 layers of 3×3 convolutions, followed by a single 4×4 convolutional layer, each layer consisting of 20 channels. Each layer is followed by a ReLU activation and batch normalization layer. A dropout layer with $p = 0.3$ is added after layers 1 through 5, and a max-pooling layer with a stride of 2 after the second layer. The convolutional part is followed by a global spatial average-pooling layer, and lastly, a fully connected layer. The P4CNN architecture is created by substituting standard convolutions with $p4$ convolutions and introducing a group coset max-pooling layer before the fully connected layer. To keep the number of parameters of Z2CNN and P4CNN approximately equal, the number of channels in P4CNN is divided by $\sqrt{4}$. We use the default input size of 28×28 for exact equivariance, and input sizes 27×27 and 29×29 for approximate equivariance. The results are averaged over 10 runs with different random seeds. The models are trained for 50 epochs using Adam [36] and an initial learning rate of 0.01, which is halved every 10 epochs.

The results in Fig. 5.5 show the performance of a model trained on MNIST and evaluated on RotMNIST, a rotated version of MNIST. The exactly equivariant network significantly outperforms its approximate counterparts on rotated samples. All the $p4$ equivariant networks still outperform the Z2CNN baseline. We also observe a much higher standard deviation in the performance of the approximately equivariant networks. The performance increase of Z2CNN at 180° can be attributed to the rotational symmetries in the MNIST dataset. The 0, 1 and 8 classes stay roughly identical when rotated 180° .

To further evaluate network generalizability to unseen rotations, we create two new versions of RotMNIST with biased rotation transformations. The rotation of each training digit is sampled from a normal distribution. Both datasets use a mean rotation of 45° , one has a standard deviation of 20° and the other of 40° . We then train the networks on these biased training sets and evaluate them on a test set with uniform rotations. The results in Fig. 5.6

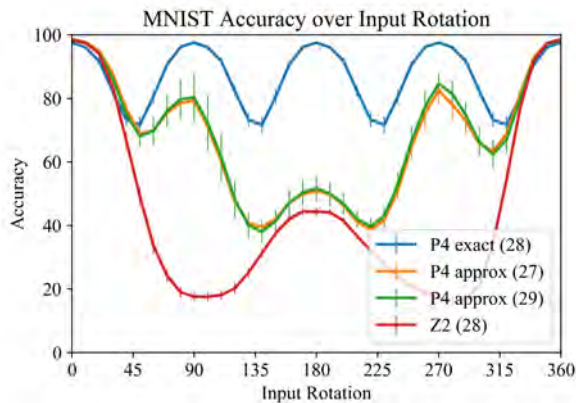
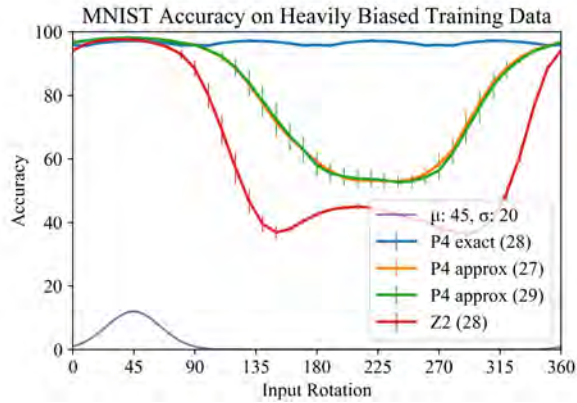


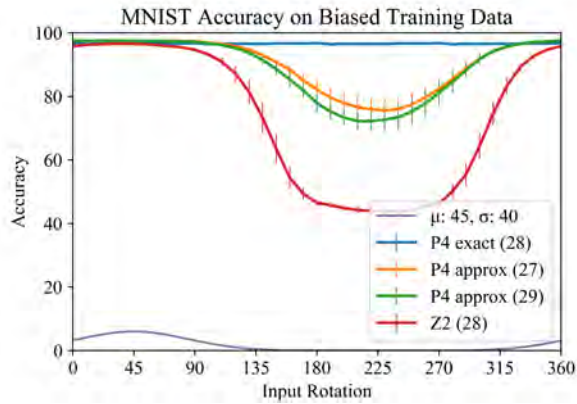
Figure 5.5: The equivariance of a network can be evaluated by explicitly applying the transformation on the test set. When the training contains no data augmentations, an exactly equivariant network generalizes significantly better than its approximate counterpart.

show that, similarly to training on non-rotated data, the exactly equivariant network generalizes noticeably better than the others. The exactly equivariant network almost becomes invariant to rotations in general, while all other networks exhibit a significant drop in performance on rotations that are not in the training data. Since the transformation distributions in the training data are often unknown, it is important to generalize to instances of the transformation that are not in the training data.

We further evaluate the impact of exact and approximate equivariance on common benchmark datasets, where we differentiate between datasets with and without rotational symmetries. Datasets with rotations include *Flowers-102* [37], where many classes have a rotationally symmetric shape, and *PatchCamelyon* [20], which is completely invariant to rotation. Datasets without rotational symmetries include *CIFAR-10*, *CIFAR-100* [38] and *ImageNet* [33]. We evaluate for unseen rotations by rotating the test set by multiples of 90° and averaging the performance over the four rotations. To achieve exact and approximate $p4$ -equivariance, we change the input size of the network, such that equation 3 holds for all layers in the network. The results are shown in the top part of Table 5.1. The approximately equivariant networks are outperformed by the exactly equivariant networks in all cases. The difference is most significant for datasets that contain limited rotations, i.e. *CIFAR-10*,



(a)



(b)

Figure 5.6: An exactly equivariant network generalizes significantly better to unseen rotations in case of a rotation bias in the training data. The rotation angles of the training set are sampled from a normal distribution $\mathcal{N}(\mu, \sigma)$, visualized at the bottom of each plot. (a) and (b) show the performance on a biased distribution with $\sigma = 20$ and $\sigma = 40$, respectively, where the performance drop in the latter is less severe but still significant.

CIFAR-100 and *ImageNet*, as here the model is not able to learn the symmetries from data. Both the exact and inexact networks outperform the baseline CNN

by a large margin, showing that in general equivariance is a beneficial property.

Dataset	Model	Test accuracy		
		Approximate $p4$	Exact $p4$	Baseline CNN
Unseen rotations				
Flowers-102	RN-18	83.32 ± 1.21	86.65 ± 1.41	68.78 ± 0.29
PCam	RN-44	86.66 ± 0.72	87.40 ± 0.71	82.43 ± 1.71
CIFAR10	RN-44	79.19 ± 0.47	93.41 ± 0.09	52.68 ± 0.11
CIFAR100	RN-44	58.82 ± 0.36	72.28 ± 0.40	38.62 ± 0.16
ImageNet	RN-18	60.01	72.55	48.10
Seen rotations				
Flowers-102	RN-18	86.28 ± 1.32	86.65 ± 1.41	82.18 ± 0.53
PCam	RN-44	87.52 ± 1.20	87.40 ± 0.71	85.35 ± 1.04
CIFAR10	RN-44	94.80 ± 0.21	93.41 ± 0.09	93.20 ± 0.11
CIFAR100	RN-44	75.00 ± 0.52	72.28 ± 0.40	70.09 ± 0.28
ImageNet	RN-18	72.48	72.55	70.00

Table 5.1: Test accuracies of ResNet models on seen and unseen rotations of common classification benchmarks. Exactly equivariant networks perform significantly better on unseen rotations, especially on datasets containing no rotation symmetries, i.e. CIFAR and ImageNet. Approximate networks are able to relax equivariance constraints and perform better on seen rotations. All $p4$ equivariant networks outperform the baseline CNN.

Performance on seen rotations For the performance on rotations that are included in the training data, also referred to as seen rotations, we first evaluate MNIST and RotMNIST using P4CNN and Z2CNN as in Section 5.4.2. We report the mean and standard deviations of the test accuracies over 100 run with different random seeds in Table 5.2. On MNIST the exactly equivariant network exhibits a performance drop between 0.65% and 0.91% compared to its approximately equivariant counterparts, which is confirmed to be statistically significant, as shown in Appendix 5.B. On RotMNIST the exact network performs identically to the approximate networks, as the approximate networks are able to learn to become invariant from the transformations found in the training data.

To evaluate for seen rotations on common classification benchmarks we compute the model accuracy on the default test set. The benchmark results can be found in the bottom part of Table 5.1. The exactly equivariant networks are generally matched or outperformed by their approximately equivariant

Model	Equivariance	MNIST	RotMNIST
Z2CNN	- (28)	98.47 ± 0.17	91.60 ± 1.25
P4CNN	Approximate (27)	98.52 ± 0.26	96.92 ± 0.27
P4CNN	Exact (28)	97.69 ± 0.17	96.89 ± 0.21
P4CNN	Approximate (29)	98.42 ± 0.25	96.87 ± 0.25

Table 5.2: Network accuracy denoted as mean \pm standard deviation on MNIST and RotMNIST test sets. The standard deviation is computed from 100 runs with different seeds. The equivariance column indicates whether the network is exactly or approximately equivariant and contains the network input size in parentheses.

counterparts, even on datasets containing rotational symmetries. This seems to indicate that there lies value in relaxing the equivariant constraints of networks. Furthermore, both $p4$ equivariant networks outperform their $z2$ equivariant counterparts, even when the dataset is not known for containing rotational symmetries. This could indicate that the improvements from the group equivariant architecture might not be solely from equivariance, but could also originate from other traits of GCNNs. We speculate that other possible explanations may be related to the increase in computations or the amount of gradients a GCNN uses compared to a standard CNN.

5

5.5 CONCLUSION

In this work, we show that Group Equivariant Convolutions [4] can and do learn to break their equivariance towards 2D rotations in common use cases. We prove theoretically and empirically that changing the input size of the network is sufficient to prevent a network from breaking its rotation equivariance. We find that exactly equivariant networks generalize significantly better to unseen rotations than their approximately equivariant counterparts, but that when the training data contains all relevant rotations there is no significant difference. As the broken rotation equivariance is essentially a consequence of pooling layers breaking translation equivariance, an interesting future research direction would be to investigate the effects of methods that improve translation equivariance, such as [39], on equivariance to other symmetries.

Interestingly, we also find results that suggest equivariant networks offer performance increases to datasets that do not contain the relevant transformations, suggesting that using GCNNs might offer benefits other than equivariance to

certain symmetries. Furthermore, we find that relaxing equivariant constraints can be beneficial for network performance. However, relaxing equivariant constraints also allows networks to become biased towards the distribution of transformations in the training data.

5.6 LIMITATIONS AND FUTURE WORK

The symmetries our method applies to are limited to rotations and reflections. Although relevant, an interesting line of future work is to investigate additional symmetry groups or more fine-grained rotations.

We found experimentally that padding has a large influence on how well a GCNN generalizes to unseen rotations, similar to CNNs [26]. While we found no conclusive explanation, we do believe it is worth further investigating.

Section 5.4.1 suggests that equivariant layers are more desirable at some depths than others, since the equivariance error drops at early layers and increases at later ones. An interesting future work would be to conduct a robust analysis of the effectiveness of equivariance at different depths in a network.

Finally, we welcome further investigations of our results on PatchCamelyon, where we found that the approximately equivariant network learned to break its equivariance to increase performance, even on a problem supposedly invariant to rotation. With the rise of relaxed equivariant constraints [31, 32], an interesting question to ask would be whether we are actually achieving better performance or simply exploiting unknown biases in the data or in the network.

REFERENCES

- [1] I. G. Johnston, K. Dingle, S. F. Greenbury, C. Q. Camargo, J. P. K. Doye, S. E. Ahnert, and A. A. Louis. “Symmetry and simplicity spontaneously emerge from the algorithmic nature of evolution”. In: *Proceedings of the National Academy of Sciences* 119.11 (2022), e2113883119. DOI: 10.1073/pnas.2113883119. eprint: <https://www.pnas.org/doi/pdf/10.1073/pnas.2113883119>. URL: <https://www.pnas.org/doi/abs/10.1073/pnas.2113883119>.
- [2] O. S. Kayhan and J. C. v. 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*. 2020, pp. 14274–14285.
- [3] M. W. Lafarge, E. J. Bekkers, J. P. Pluim, R. Duits, and M. Veta. “Roto-translation equivariant convolutional networks: Application to histopathology image analysis”. In: *Medical Image Analysis* 68 (2021), p. 101849. ISSN: 1361-8415. DOI: <https://doi.org/10.1016/j.media.2020.101849>. URL: <https://www.sciencedirect.com/science/article/pii/S1361841520302139>.
- [4] T. S. Cohen and M. Welling. “Group Equivariant Convolutional Networks”. In: *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48. ICML’16*. New York, NY, USA: JMLR.org, 2016, pp. 2990–2999.
- [5] C. Olah, N. Cammarata, C. Voss, L. Schubert, and G. Goh. “Naturally Occurring Equivariance in Neural Networks”. In: *Distill* (2020). DOI: 10.23915/distill.00024.004. URL: <https://distill.pub/2020/circuits/equivariance>.
- [6] J. Geiping, M. Goldblum, G. Somepalli, R. Shwartz-Ziv, T. Goldstein, and A. G. Wilson. “How Much Data Are Augmentations Worth? An Investigation into Scaling Laws, Invariance, and Implicit Regularization”. In: *The Eleventh International Conference on Learning Representations*. 2023. URL: <https://openreview.net/forum?id=3aQs3MCsexD>.
- [7] A. Zhou, T. Knowles, and C. Finn. “Meta-learning Symmetries by Reparameterization”. In: *International Conference on Learning Representations*. 2021. URL: <https://openreview.net/forum?id=-QxT4mJdijq>.
- [8] M. Jaderberg, K. Simonyan, A. Zisserman, and k. kavukcuoglu koray. “Spatial Transformer Networks”. In: *Advances in Neural Information Processing Systems*. Ed. by C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett. Vol. 28. Curran Associates, Inc., 2015.
- [9] D. Kuzminykh, D. Polykovskiy, and A. Zhebrak. “Extracting Invariant Features From Images Using An Equivariant Autoencoder”. In: *Proceedings of The 10th Asian Conference on Machine Learning*. Ed. by J. Zhu and I. Takeuchi. Vol. 95. Proceedings of Machine Learning Research. PMLR, Nov. 2018, pp. 438–453. URL: <https://proceedings.mlr.press/v95/kuzminykh18a.html>.

- [10] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton. “A Simple Framework for Contrastive Learning of Visual Representations”. In: ICML’20. JMLR.org, 2020.
- [11] A. DEVILLERS and M. Lefort. “EquiMod: An Equivariance Module to Improve Visual Instance Discrimination”. In: *The Eleventh International Conference on Learning Representations*. 2023. URL: <https://openreview.net/forum?id=eDLwjKmtYFt>.
- [12] T. F. van der Ouderaa and M. van der Wilk. “Learning invariant weights in neural networks”. In: *Proceedings of the Thirty-Eighth Conference on Uncertainty in Artificial Intelligence*. Ed. by J. Cussens and K. Zhang. Vol. 180. Proceedings of Machine Learning Research. PMLR, Aug. 2022, pp. 1992–2001. URL: <https://proceedings.mlr.press/v180/ouderaa22a.html>.
- [13] M. van der Wilk, M. Bauer, S. John, and J. Hensman. “Learning Invariances using the Marginal Likelihood”. In: *Advances in Neural Information Processing Systems*. Ed. by S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett. Vol. 31. Curran Associates, Inc., 2018.
- [14] T. Cohen, M. Weiler, B. Kicanaoglu, and M. Welling. “Gauge Equivariant Convolutional Networks and the Icosahedral CNN”. In: *Proceedings of the 36th International Conference on Machine Learning*. Ed. by K. Chaudhuri and R. Salakhutdinov. Vol. 97. Proceedings of Machine Learning Research. PMLR, June 2019, pp. 1321–1330. URL: <https://proceedings.mlr.press/v97/cohen19d.html>.
- [15] P. D. Haan, M. Weiler, T. Cohen, and M. Welling. “Gauge Equivariant Mesh {CNN}s: Anisotropic convolutions on geometric graphs”. In: *International Conference on Learning Representations*. 2021. URL: <https://openreview.net/forum?id=Jnspzp-olZE>.
- [16] M. Weiler, P. Forré, E. Verlinde, and M. Welling. *Coordinate Independent Convolutional Networks – Isometry and Gauge Equivariant Convolutions on Riemannian Manifolds*. 2021. DOI: 10.48550/ARXIV.2106.06020. URL: <https://arxiv.org/abs/2106.06020>.
- [17] N. Dehmamy, R. Walters, Y. Liu, D. Wang, and R. Yu. “Automatic Symmetry Discovery with Lie Algebra Convolutional Network”. In: *Advances in Neural Information Processing Systems*. Ed. by A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan. 2021. URL: https://openreview.net/forum?id=NPOWF_ZLfC5.
- [18] K. S. Tai, P. Bailis, and G. Valiant. “Equivariant Transformer Networks”. In: *Proceedings of the 36th International Conference on Machine Learning*. Ed. by K. Chaudhuri and R. Salakhutdinov. Vol. 97. Proceedings of Machine Learning Research. PMLR, June 2019, pp. 6086–6095. URL: <https://proceedings.mlr.press/v97/tai19a.html>.
- [19] E. J. Bekkers, M. W. Lafarge, M. Veta, K. A. J. Eppenhof, J. P. W. Pluim, and R. Duits. “Roto-Translation Covariant Convolutional Networks for Medical Image Analysis”. In: *Medical Image Computing and Computer Assisted Intervention – MICCAI 2018*. Springer International Publishing, 2018, pp. 440–448. ISBN: 978-3-030-00928-1.

- [20] B. S. Veeling, J. Linmans, J. Winkens, T. Cohen, and M. Welling. “Rotation Equivariant CNNs for Digital Pathology”. In: *Medical Image Computing and Computer Assisted Intervention – MICCAI 2018: 21st International Conference, Granada, Spain, September 16-20, 2018, Proceedings, Part II*. Granada, Spain: Springer-Verlag, 2018, pp. 210–218. ISBN: 978-3-030-00933-5. DOI: 10.1007/978-3-030-00934-2_24. URL: https://doi.org/10.1007/978-3-030-00934-2_24.
- [21] J. Han, J. Ding, N. Xue, and G.-S. Xia. “ReDet: A Rotation-Equivariant Detector for Aerial Object Detection”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2021, pp. 2786–2795.
- [22] J. Mitton and R. Murray-Smith. “Rotation Equivariant Deforestation Segmentation and Driver Classification”. In: *NeurIPS 2021 Workshop on Tackling Climate Change with Machine Learning*. 2021. URL: <https://www.climatechange.ai/papers/neurips2021/16>.
- [23] R. Zhang. “Making Convolutional Networks Shift-Invariant Again”. In: *Proceedings of the 36th International Conference on Machine Learning*. Ed. by K. Chaudhuri and R. Salakhutdinov. Vol. 97. Proceedings of Machine Learning Research. PMLR, June 2019, pp. 7324–7334. URL: <https://proceedings.mlr.press/v97/zhang19a.html>.
- [24] M. Amirul Islam, M. Kowal, S. Jia, K. G. Derpanis, and N. D. B. Bruce. “Global Pooling, More than Meets the Eye: Position Information is Encoded Channel-Wise in CNNs”. In: *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*. 2021, pp. 773–781. DOI: 10.1109/ICCV48922.2021.00083.
- [25] J. Xu, H. Kim, T. Rainforth, and Y. W. Teh. “Group Equivariant Subsampling”. In: *Advances in Neural Information Processing Systems*. Ed. by A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan. 2021. URL: <https://openreview.net/forum?id=CtaDl9L0bIQ>.
- [26] O. Kayhan and J. C. van Gemert. “On Translation Invariance in CNNs: Convolutional Layers can Exploit Absolute Spatial Location”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020.
- [27] D.-H. Jang, S. Chu, J. Kim, and B. Han. “Pooling Revisited: Your Receptive Field Is Suboptimal”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2022, pp. 549–558.
- [28] D. Romero, E. Bekkers, J. Tomczak, and M. Hoogendoorn. “Attentive Group Equivariant Convolutional Networks”. In: *Proceedings of the 37th International Conference on Machine Learning*. Vol. 119. Proceedings of Machine Learning Research. PMLR, July 2020, pp. 8188–8199. URL: <http://proceedings.mlr.press/v119/romero20a.html>.
- [29] H. Mo and G. Zhao. *RIC-CNN: Rotation-Invariant Coordinate Convolutional Neural Network*. 2022. DOI: 10.48550/ARXIV.2211.11812. URL: <https://arxiv.org/abs/2211.11812>.

- [30] P. Bagad, F. Eijkelboom, M. Fokkema, D. de Goede, P. Hilders, and M. Kofinas. “C-3PO: Towards Rotation Equivariant Feature Detection and Description”. In: *3rd Visual Inductive Priors for Data-Efficient Deep Learning Workshop*. 2022. URL: <https://openreview.net/forum?id=dXouQ9ubkPJ>.
- [31] D. W. Romero and S. Lohit. “Learning Partial Equivariances From Data”. In: *Advances in Neural Information Processing Systems*. Ed. by S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh. Vol. 35. Curran Associates, Inc., 2022, pp. 36466–36478.
- [32] T. van der Ouderaa, D. W. Romero, and M. van der Wilk. “Relaxing Equivariance Constraints with Non-stationary Continuous Filters”. In: *Advances in Neural Information Processing Systems*. Ed. by S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh. Vol. 35. Curran Associates, Inc., 2022, pp. 33818–33830.
- [33] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. “ImageNet: A large-scale hierarchical image database”. In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*. 2009, pp. 248–255. DOI: 10.1109/CVPR.2009.5206848.
- [34] K. He, X. Zhang, S. Ren, and J. Sun. “Deep Residual Learning for Image Recognition”. In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 770–778. DOI: 10.1109/CVPR.2016.90.
- [35] Y. LeCun, C. Cortes, and C. Burges. “MNIST handwritten digit database”. In: *ATT Labs [Online]*. Available: <http://yann.lecun.com/exdb/mnist> 2 (2010).
- [36] D. Kingma and J. Ba. “Adam: A Method for Stochastic Optimization”. In: *International Conference on Learning Representations* (Dec. 2014).
- [37] M.-E. Nilsback and A. Zisserman. “Automated Flower Classification over a Large Number of Classes”. In: *Indian Conference on Computer Vision, Graphics and Image Processing*. Dec. 2008.
- [38] A. Krizhevsky. “Learning Multiple Layers of Features from Tiny Images”. In: (2009), pp. 32–33. URL: <https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf>.
- [39] A. Chaman and I. Dokmanic. “Truly Shift-Invariant Convolutional Neural Networks”. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2021, pp. 3773–3783.

APPENDICES

5.A PROOF MIRRORING EQUIVARIANCE

The proof for the (horizontal) mirroring transformation is similar to the proof for rotation in Section 5.3.2. We again use the function *index*, which returns the input indices for a convolutional or pooling layer corresponding to the output value located at index (x, y) :

$$\text{index}\left(\begin{bmatrix} x \\ y \end{bmatrix}\right) = \left[\begin{bmatrix} sx \\ sy \end{bmatrix}, \begin{bmatrix} sx+k-1 \\ sy+k-1 \end{bmatrix}\right]. \quad (5.15)$$

Here s is the stride used for subsampling and k represents the kernel size. The output of the function is a square patch, denoted as $[\vec{u}, \vec{v}]$, where \vec{u} and \vec{v} represent the indices of the top left and bottom right corner, respectively. The sampled indices include all integer tuples within this patch.

Similarly to the R function in the paper, we now introduce a function M , which takes an index (x, y) as input and returns the indices mirrored horizontally:

$$M_n\left(\begin{bmatrix} x \\ y \end{bmatrix}\right) = \begin{bmatrix} n-1-x \\ y \end{bmatrix}, \quad (5.16)$$

where n indicates the width and height of the feature map in which the index (x, y) is located. We further generalize Eq. (5.16) to an input patch $[\vec{u}, \vec{v}]$ rather than a single coordinate, resulting in Eq. (5.17):

$$= \left[\begin{bmatrix} n-1-x_2 \\ y_1 \end{bmatrix}, \begin{bmatrix} n-1-x_1 \\ y_2 \end{bmatrix}\right] \quad (5.17)$$

In the resulting output coordinates x_1 and x_2 get interchanged due to the mirroring of the patch: the top left corner becomes the top right corner, while the bottom right corner becomes the bottom left corner.

Given that our layer takes a feature map of width and height i as input, we

can write the width and height of the output feature map as

$$o = \lfloor \frac{i-k}{s} \rfloor + 1. \quad (5.18)$$

For a layer to be exactly equivariant, determining the sampled indices and then mirroring should return the same result as mirroring first and then determining the sampled indices, which we can formally denote as

$$\text{index} \left(M_o \left(\begin{bmatrix} x \\ y \end{bmatrix} \right) \right) = M_i \left(\text{index} \left(\begin{bmatrix} x \\ y \end{bmatrix} \right) \right). \quad (5.19)$$

To solve the left-hand side, we substitute Eq. (5.16) into Eq. (5.15), yielding

$$\begin{aligned} \text{index} \left(M_o \left(\begin{bmatrix} x \\ y \end{bmatrix} \right) \right) &= \text{index} \left(\begin{bmatrix} \lfloor \frac{i-k}{s} \rfloor - x \\ y \end{bmatrix} \right) \\ &= \left[\left[s \lfloor \frac{i-k}{s} \rfloor - sx \right], \left[s \lfloor \frac{i-k}{s} \rfloor - sx + k - 1 \right] \right]. \end{aligned} \quad (5.20)$$

The same can be done for the right-hand side, by substituting Eq. (5.15) into Eq. (5.17), resulting in

$$\begin{aligned} M_i \left(\text{index} \left(\begin{bmatrix} x \\ y \end{bmatrix} \right) \right) &= R_i \left(\left[\begin{bmatrix} sx \\ sy \end{bmatrix}, \begin{bmatrix} sx + k - 1 \\ sy + k - 1 \end{bmatrix} \right] \right) \\ &= \left[\left[i - k - sx \right], \left[i - 1 - sx \right] \right]. \end{aligned} \quad (5.21)$$

Substituting Eqs. (5.20) and (5.21) into Eq. (5.19), we find:

$$s \lfloor \frac{i-k}{s} \rfloor - sx = i - k - sx, \quad (5.22)$$

$$s \lfloor \frac{i-k}{s} \rfloor - sx + k - 1 = i - 1 - sx. \quad (5.23)$$

Removing duplicate terms yields a single equation

$$s \lfloor \frac{i-k}{s} \rfloor = i - k. \quad (5.24)$$

Eq. (5.24) is identical to the constraint found for rotation equivariance. Therefore, when constructing a network that is exactly equivariant to mirroring, the same restrictions on stride, kernel size and input size hold as for rotation.

5.B STATISTICAL ANALYSIS OF SIGNIFICANCE

To determine the statistical significance of our results, we compare each pair of models using an independent t-Test testing the null hypothesis $H_0 : \mu_a = \mu_b$. We use a significance level $\alpha = 1.0 \times 10^{-2}$. However, since we perform 12 comparisons in total, we use Bonferroni correction and find a new significance level $\alpha = 8.33 \times 10^{-4}$. The p-values resulting from the t-Tests for MNIST can be found in Table 5.B.1, and in Table 5.B.2 for RotMNIST. The values were calculated using 100 repeats for each condition to ensure a representative normal distribution for the performance. The performance distribution was then visually confirmed to be a normal distribution. Due to unequal variances between the performance of P4 and Z2 networks on RotMNIST, a Welch's t-Test was used to calculate the p-value for comparisons including the Z2 network.

	P4 (27)	P4 (28)	P4 (29)
Z2 (28)	1.43×10^{-1}	7.92×10^{-82}	1.09×10^{-1}
P4 (27)	-	1.19×10^{-62}	9.97×10^{-2}
P4 (28)	-	-	3.03×10^{-57}

Table 5.B.1: p-values for two sided t-Test for different networks trained on the MNIST dataset. The input dimension of the network is indicated using parentheses.

	P4 (27)	P4 (28)	P4 (29)
Z2 (28)	2.95×10^{-99}	1.53×10^{-99}	2.95×10^{-99}
P4 (27)	-	4.54×10^{-1}	1.70×10^{-1}
P4 (28)	-	-	4.44×10^{-1}

Table 5.B.2: p-values for two sided t-Test for different networks trained on the RotMNIST dataset. For p-values of comparisons containing the Z2 network, a Welch's t-Test is used due to unequal variances. The input dimension of the network is indicated using parentheses.

For MNIST, we find a significant difference between our exactly equivariant network and the other networks. For RotMNIST we find no significant differences between the P4 equivariant networks, but we do find that the Z2 equivariant network performs significantly worse than the others.

To assert the effect size, we look at the *95%-confidence intervals*, given in

Table 5.B.3. We find that on MNIST, the exactly equivariant network has a performance drop between 0.65% and 0.91% compared to the other networks. On RotMNIST, P4 equivariant networks offer a performance increase between 4.97% and 5.62% compared to a standard CNN.

Model	Equivariance	MNIST	RotMNIST
Z2CNN	- (28)	[98.44; 98.51]	[91.35; 91.85]
P4CNN	Approx (27)	[98.47; 98.57]	[96.86; 96.97]
P4CNN	Exact (28)	[97.66; 97.72]	[96.85; 96.93]
P4CNN	Approx (29)	[98.37; 98.47]	[96.82; 96.92]

Table 5.B.3: Network accuracy confidence interval on MNIST and RotMNIST test sets. The standard deviation is calculated using 100 runs with different seeds. The equivariance column shows whether the network is exactly or approximately equivariant and the input dimensions of the network are indicated in parentheses.

6

DISCUSSION

This thesis explores the incorporation of knowledge priors into deep learning algorithms to improve data and compute efficiency, based on the premise that knowledge that is built-in no longer needs to be learned from data. The previous chapters have discussed a variety of inductive biases, ranging from physics-based reflection models (Chapter 2) to photometric (Chapter 3) and geometric (Chapters 4 and 5) transformations. Several common themes arise from these knowledge priors, which will be discussed below. Specifically, we will zoom in on the specific data and compute efficient settings in which these priors excel, discuss considerations for how to thoroughly evaluate them and mention some limitations of approximate priors. Finally, we conclude with an outlook for future work.

6.1 DATA AND COMPUTE EFFICIENCY

Knowledge priors are data efficient and improve out-of-domain generalization It is well-known that incorporating convolution improves the data efficiency of neural networks as the model is able to generalize over image locations through explicit parameter sharing. The knowledge priors discussed in this thesis demonstrate similar properties, in which we can distinguish between two forms of data efficiency: (i) out-of-domain generalization and (ii) the low-data regime. On the one hand, inductive biases improve out-of-domain generalization. In Chapter 2, the Color Invariant Convolution improves the nighttime performance of a model trained on daytime data. The hue equivariant convolution in Chapter 3 improves robustness to test-time color changes. Similarly, exact Group Equivariant Convolutions in Chapter 5 improve robustness to test-time rotations. This reduces the need for collecting a large training set, and makes more efficient use of model capacity as the

transformations are explicitly modeled in the architecture, rather than being learned from data. On the other hand, inductive biases improve accuracy in the low-data regime: the Separable GConvs in Chapter 4 show that hard-wiring parameter sharing is especially useful when the number of training samples is limited. Knowledge priors thus play a significant role in reducing the data hunger of deep learning.

On parameter count vs. compute efficiency The equivariant convolutions discussed in Chapters 3 to 5 are computationally more expensive compared to regular convolutions. For a fair comparison with a baseline model, it is common practice to downscale the network width to keep the number of parameters equal to the baseline [1]. However, it is often overlooked that even after downscaling the resulting equivariant architecture performs significantly more multiply-accumulate (MAC) operations. For example, the equivariant P4CNN architecture in Chapter 4 has a comparable parameter count to the baseline Z2CNN network, yet performs three times more MACs and as a result has significantly longer training and inference times. This does in no way mean that equivariant architectures are by definition less compute efficient; for instance our gc-P4CNN performs 40% fewer MACs yet still has a 45% relative lower test error compared to the Z2CNN. However, it does underpin the importance of evaluating and reporting compute costs. It is also important to note that theoretical compute costs do not directly correspond to practical compute speed, as it is highly dependent on hardware optimizations of various operations.

6

6.2 APPROXIMATE KNOWLEDGE PRIORS

Inexact priors are still beneficial As British statistician George Box once famously wrote, *"All models are wrong, but some are useful"*. Models do not precisely reflect reality, but are based on approximations and simplifying assumptions. This extends to the knowledge priors discussed in this thesis. For instance, the image formation model that lies at the foundation of the color invariants in Chapter 2 relies on simplifications, such as assuming purely matte reflections, non-transparent materials, and a single, spatially uniform light source - conditions that are rarely met in natural scenes. Similarly, the hue rotation introduced in the color equivariant layer in Chapter 3 captures only one facet of color changes, neglecting variations in brightness, contrast, and saturation. Furthermore, the equivariance achieved by GConvs, as discussed

in Chapters 4 and 5, is often only approximate due to errors introduced by downsampling layers in the network. Despite these imperfections, our experiments consistently demonstrate the benefits of these approximate knowledge priors. The inexact color invariants improve robustness to illumination shifts, incorporating hue transformations improves out-of-distribution generalization, and even approximate rotation equivariance consistently outperforms baseline CNNs, with or without rotation augmentation. These results underscore that inexact knowledge priors remain valuable tools for enhancing data efficiency.

Data augmentation and prior knowledge are complementary Data augmentation is a widely adopted method to improve the data efficiency of deep learning models. The core idea is to apply transformations to the input that keep the semantic meaning (e.g. class) unchanged, thereby artificially generating new samples from existing data. As seen in this thesis, data augmentation also exhibits an interesting interplay with invariant and equivariant architectures. Equivariant architectures often model discrete transformations; examples include the discrete hue shifts in Color Equivariant Convolutions (Chapter 3) and the multiples of 90-degree rotations in Group Equivariant Convolutions (Chapters 4 and 5). While augmenting data samples with the same discrete transformations would offer no benefit, applying finer augmentations does improve model generalization to more subtle transformations, e.g. 45 degree rotations for GConvs. In other words, data augmentation can be used to interpolate between the discrete transformations the model is hardwired to be equivariant to. In the case of inexact priors, such as Color Invariant Convolutions (Chapter 2) and approximate rotation equivariance (Chapter 5), the architecture fails to precisely model the intended transformation and data augmentation can be effectively used to further improve model robustness. For instance, the representation produced by the CIConv layer is not exactly invariant to illumination changes, therefore applying brightness, contrast, hue and saturation perturbations to the input helps the model to extrapolate from the training sample to other lighting conditions. Consequently, while equivariant architectures show much promise for improving the data efficiency of deep learning models, data augmentation will continue to play an important role in practical applications.

Flexible bypassing of inductive biases Invariant and equivariant architectures are designed assuming that the incorporated inductive bias is relevant to the task at hand and correctly models the data distribution. For instance, rota-

tion equivariant GConvs assume the presence of rotation symmetries in the data, while Color Invariant Convolutions assume a significant illumination-based domain shift. Unfortunately, not all data samples may satisfy these assumptions equally well, and hardwired inductive biases do generally not support flexible bypassing when not matching the properties of the data. In these cases, enforcing an inductive bias can have a detrimental effect. As discussed in Chapter 2, the robustness of color invariants to illumination changes comes at the loss of some discriminative power [2]. As such, CIConv performs worse on daytime images than using the default RGB input, and thus should in these cases be bypassed. In Chapter 5 we have found that Group Equivariant Convolutions are able to bypass their rotation bias by learning to become less equivariant when beneficial. Similarly, it has been shown that regular convolutional layers are able to learn absolute spatial location by exploiting padding, thereby bypassing their strict shift equivariance properties [3]. These examples illustrate that equivariant architectures are generally but not always useful, which is why adaptive modules should be designed that allow for more flexibility in the incorporated inductive biases.

6

6.3 FUTURE OUTLOOK

While many recent advances in deep learning can be attributed to ever increasing datasets and model sizes, this trend arguably does not provide a sustainable way forward. Already now we are witnessing debates around scientific reproducibility in our field [4, 5], partly fueled by the immense computation costs required for training state-of-the-art models, which are only affordable for large companies. Research on compute and data efficient methods is therefore more important than ever. This section briefly highlights several promising research directions based on the findings in this dissertation.

Understanding equivariance While it is generally accepted that equivariant architectures improve generalization to unseen transformations, we have found GConvs to also perform better on datasets such as ImageNet [6], where rotations are not predominantly present (Chapter 5). Understanding the exact source of empirical gains of equivariance is key to explain such unexpected side-effects, and consequently designing better and more data efficient equivariant network components.

Scaling down deep learning by scaling up equivariance Large, multimodal foundation models [7] trained on enormous datasets are becoming an increasingly important cornerstone of deep learning applications. The major innovation of these methods lies in the ability to learn from uncurated image and text data in a self-supervised manner [8], while the model architectures themselves have remained unchanged. The empirical success of GConvs on ImageNet suggests that the practical usability of equivariant architectures is not only limited to small scale datasets. Investigating the use of GCNNs for multimodal learning is therefore a promising and high-impact research direction for improving compute efficiency in large scale applications. In addition to simply scaling up GCNNs to larger model sizes, we argue for a thorough investigation on the use of well established methods for optimizing compute performance in regular CNNs, including but not limited to neural architecture search [9], quantization [10], pruning [11], and distillation [12].

Effect of architecture hyperparameters on invariance and equivariance We have found empirically that multiple architecture components influence the effectiveness of in- and equivariance, including padding, and the depth at which equivariant layers are used. Though initial results suggest that equivariance is mostly beneficial in the early layers of a network, a robust analysis on the effects of various architecture hyperparameters would provide valuable insights.

Relaxing equivariance constraints Since inductive biases do not always exactly fit the data distribution, a promising future research direction is to explore ways to relax or bypass hard-coded in- or equivariance. More specifically, CConv layers should be bypassed when the test data is originating from the daytime domain (Chapter 2), and rotation equivariance should be relaxed when rotation is not predominantly present in the dataset (Chapter 5). Initial experiments with Color Equivariant Convolutions (Chapter 3) also showed slight improvements when the rotation matrix is made learnable, which results in approximately hue-equivariant transformations.

Combining geometric and photometric equivariance Most current works regard individual transformation groups, such as rotation [1], scale [13], and hue (Chapter 3). An interesting line of future research to investigate combinations of different geometric and photometric transformations, including brightness and saturation.

6.4 FINAL WORDS

The exponential growth in deep learning, however remarkable, is arguably unsustainable in its current form. Going against the trend of increasing data and compute, this dissertation has explored the integration of knowledge priors into deep learning architectures, with a focus on invariant and equivariant approaches. These methods have proven effective in improving data efficiency and generalization on small scale problems, setting the stage for the next phase - scaling up these techniques to address larger and more complex tasks. Having a deeper understanding of the nuances of invariance and equivariance will help widespread adoption of these methods for practical use. This work aims to contribute to this understanding, thereby serving as a step towards a more sustainable and accessible deep learning landscape.

REFERENCES

- [1] T. S. Cohen and M. Welling. “Group Equivariant Convolutional Networks”. In: *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*. ICML’16. New York, NY, USA: JMLR.org, 2016, pp. 2990–2999.
- [2] J. M. Geusebroek, R. van den Boomgaard, A. W. M. Smeulders, and H. Geerts. “Color Invariance”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 23.12 (2001), pp. 1338–1350.
- [3] O. Kayhan and J. C. van Gemert. “On Translation Invariance in CNNs: Convolutional Layers can Exploit Absolute Spatial Location”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2020.
- [4] E. Raff. “A Step Toward Quantifying Independently Reproducible Machine Learning Research”. In: *Advances in Neural Information Processing Systems*. Ed. by H. Wallach, H. Larochelle, A. Beygelzimer, F. d’Alché-Buc, E. Fox, and R. Garnett. Vol. 32. Curran Associates, Inc., 2019.
- [5] B. Yildiz, H. Hung, J. H. Krijthe, C. C. Liem, M. Loog, G. Migut, F. A. Oliehoek, A. Panichella, P. Pawełczak, S. Picek, M. de Weerdt, and J. van Gemert. “Reproduced-Papers.org: Openly teaching and structuring machine learning reproducibility”. In: *International Workshop on Reproducible Research in Pattern Recognition*. Springer, 2021, pp. 3–11.
- [6] A. Krizhevsky, I. Sutskever, and G. Hinton. “ImageNet Classification with Deep Convolutional Neural Networks”. In: *Neural Information Processing Systems* 25 (Jan. 2012). DOI: 10.1145/3065386.
- [7] R. Bommasani *et al.* “On the Opportunities and Risks of Foundation Models”. In: *ArXiv* (2021). URL: <https://crfm.stanford.edu/assets/report.pdf>.
- [8] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever. “Learning Transferable Visual Models From Natural Language Supervision”. In: *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*. Ed. by M. Meila and T. Zhang. Vol. 139. Proceedings of Machine Learning Research. PMLR, 2021, pp. 8748–8763. URL: <http://proceedings.mlr.press/v139/radford21a.html>.
- [9] T. Elsken, J. H. Metzen, and F. Hutter. “Neural Architecture Search: A Survey”. In: *J. Mach. Learn. Res.* 20.1 (Jan. 2019), pp. 1997–2017. ISSN: 1532-4435.
- [10] A. Gholami, S. Kim, Z. Dong, Z. Yao, M. W. Mahoney, and K. Keutzer. “A Survey of Quantization Methods for Efficient Neural Network Inference”. In: *CoRR* abs/2103.13630 (2021). arXiv: 2103.13630. URL: <https://arxiv.org/abs/2103.13630>.

- [11] T. Liang, J. Glossner, L. Wang, S. Shi, and X. Zhang. “Pruning and Quantization for Deep Neural Network Acceleration: A Survey”. In: *Neurocomputing* 461 (July 2021). DOI: 10.1016/j.neucom.2021.07.045.
- [12] G. E. Hinton, O. Vinyals, and J. Dean. “Distilling the Knowledge in a Neural Network”. In: *ArXiv abs/1503.02531* (2015). URL: <https://api.semanticscholar.org/CorpusID:7200347>.
- [13] I. Sosnovik, M. Szmaja, and A. Smeulders. “Scale-Equivariant Steerable Networks”. In: *International Conference on Learning Representations*. 2020. URL: <https://openreview.net/forum?id=HJgpugrKPS>.

ACKNOWLEDGMENTS

This dissertation marks the end of an 11-year journey in Delft. I'd like to thank the people that made all this possible.

First of all, a big thanks to my promotors. **Jan**, during my MSc thesis you gave me an honest introduction to scientific research, including all of its ups and downs, and planted the seed in my head for pursuing a PhD. I'm happy you did. Thanks for your support, your relaxed way of supervision and for all our brainstorm sessions where you often helped me turn vague ideas into hypotheses and toy experiments. **Marcel**, thanks for your critical attitude during our meetings - your ability to pinpoint areas for improvement is always spot on. Also a huge thanks to both of you for facilitating such an incredibly pleasant social environment to work in.

Second, thanks to my office roommates and friends **Robert-Jan** and **Ombretta**. **Robert-Jan**, these four years wouldn't have been the same without your horrible jokes. Thanks for challenging me both intellectually as well as on the tennis courts. I'm glad we got to work together on so many things, with my personal highlights being our weekly whiteboard update sessions, organizing the VIPriors workshop in Israel, and playing catch with the stress ball in our office. **Ombretta**, sorry for causing you so much stress with the stress ball. Jokes aside, thanks for your always helpful and cheerful attitude. I enjoyed our fun yet productive Douwe Egberts sessions and I'm happy we got to work on some joint publications in our last year. Also thanks for enriching my PhD experience with regular cat pictures and for organizing all the dinners, bouldering sessions, (techno) parties and others that brought PRB together even more. It's been a fun ride on the PhD roller coaster with the three of us!

To the Computer Vision Lab: **Osman**, thanks for the warm welcome to the CV lab on my first day. I remember you showing me around the floor and you've been like a big bro for me in the lab ever since. **Nergis**, **Silvia**, thanks for always providing excellent input on research topics, and for co-supervising many students with me - I've learned a lot from you. **Casper**, your skills for making awesome teaching materials are amazing and you have taken the DL course to a whole new level with the lab assignments. **Aurora**, thanks for joining the ski trip, even without knowing me. I hope you didn't regret it. **Yancong**, thanks for joining the ski trip, even without knowing

how to ski. I hope you didn't regret it. **Sander**, I appreciate our shared love for stroopkoeken. Spar University is lucky to have us. **Alejandro, Amogh, Chengming, Giorgio, Hadi, Hesam, Marcos, Marian, Seyran, Xucong, Xin, Xiangwei, Yeshwanth, Yunqiang, Zhi-Yi, Ziqi** and others, thanks for being wonderful colleagues, you are all amazing. To all students I was lucky enough to supervise, it was a pleasure to collaborate with you. Special thanks to **Tom**, whose work significantly contributed to my thesis.

To PRB: **Rickard**, I'm happy that we share a love for both mountains as well as festivals. Climbing Gran Paradiso together was an epic adventure, let's keep doing more of that! Also thanks for introducing some Swedish culture to our floor, especially the *fika* sessions have been life-changing. **David, Marco**, thanks for not being blinded by bold numbers and applying some common sense to CV papers. Your critical mindsets never failed to spark a great discussion at the coffee talks. **Tom**, thanks for being the most joyful person at TU Delft. Keep that laughter going. **Arman**, thanks for our discussions about fundamental linear algebra concepts. Going back to the basics was always immensely educational. **Chirag**, it was a pleasure attending ECCV together. I'm impressed the holographic projections didn't fool you. **Jim**, thanks for introducing me to David Vunk. **Swier**, we were awesome ski-boarding down the slopes together! **Tiffany**, thanks for the pre-festival chillings in your garden. Your cooking skills are off the charts. **Amelia, Aysun, Bernd, Gabriel, Hayley, Jana, Jesse, Jin, Jose, Mahdi, Merve, Paul, Ramin, Skander, Stavros, Stephan, Stephanie, Taylan, Yasin, Yuko** and many others, thanks for all the pleasant chats, useful feedback, ideas, comments, laughs and for all being part of PRB. I couldn't have wished for a better place to do a PhD. Also thanks to **Elena, Zuza** and **Davide** for being great company whenever we met.

Many thanks to those who keep PRB a well-oiled machine. Computer magicians **Ruud** and **Bart**: you have without doubt saved me countless hours of suffering. **Azza**, thanks for taming the beast called the HPC. **Saskia** and **Marunka**, thank you for handling all paperwork with lightning speed.

Also a big thanks to my team at Google: **Eduard, Bernhard, Simon, Jan, Manuel, Tianqi, Johann, Thomas, Natalie** and **Dror**. Interning with you was fantastic, and I learned much about software development and industry work. I've always been drawn to the applied side of research and the hands-on tech experience was incredibly valuable. I also really enjoyed living in Zurich and the weekend hikes in the Swiss Alps, and my Swiss roomies **Andreas** and **Leonie** made it feel like home. Thanks for having me! Also a huge thanks to **Ehsan** and **Anton** at AIML for hosting my research visit. Our collaboration was enriching, and I gained valuable insights about multimodal models. Special thanks to **Silvia** for facilitating it and to my roommates **Trent** and **Stella** for

the Aussie immersion and fun times in Adelaide.

To my friends: **Yordi** and **Yannick**, I'm glad we kept in touch after high school despite having moved to different corners of the country. Looking forward to more weekends filled with board games and good conversations. **Arjen, Ewout, Han, Hermen, Hessel, Jaap, Joost, Matys, Pim** and **Xander**: thanks for throwing sixes. Thanks to **Joey** Travels, including **Bart, Daniel, Jan, Jelle, Jorden, Leon, Martin, Richelle, Sam, Teun, Tijs, William** and **Zacca** for the peaceful and tranquil holidays, to **Eva, Lisa, Lisa** and **Sander** for feeding me food and random trivia on Mondays, and to my housemates **Hannah, Joris, Julius, Marlissa** and **Sanne** for all the chill sessions, Christmas dinners, parties, Netflix binging, sailing trips, workouts, and for dealing with the significant portions of beach I bring home on a regular basis.

To my parents, **Ildikó** and **Attila**, thank you for your continuous support and for always believing in me. You have encouraged my interest in science and technology from the first moment I can remember, and have always motivated me to get the best out of myself.

Lastly, **Ghi**, thank you for your endless energy and optimism. I'm glad we share so many interests, as well as the inability to sit still, and I'm looking forward to all our future adventures.

*Attila Lengyel
Delft, November 2023*

LIST OF PUBLICATIONS

In this thesis

1. A. Lengyel, S. Garg, M. Milford, and J. 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)*. Oct. 2021, pp. 4399–4409
2. A. Lengyel, O. Strafforello, R.-J. Brintjes, A. Gielisse, and J. van Gemert. “Color Equivariant Convolutional Networks”. In: *Advances in Neural Information Processing Systems*. Vol. 36. 2023, pp. 29831–29850
3. A. Lengyel and J. C. van Gemert. “Exploiting Learned Symmetries in Group Equivariant Convolutions”. In: *2021 IEEE International Conference on Image Processing (ICIP)*. 2021, pp. 759–763. DOI: 10.1109/ICIP42928.2021.9506362
4. T. Edixhoven, A. Lengyel, and J. C. van Gemert. “Using and Abusing Equivariance”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*. Oct. 2023, pp. 119–128

Other publications

1. L. Zeng, A. Lengyel, N. Tomen, and J. C. van Gemert. “Copy-Pasting Coherent Depth Regions Improves Contrastive Learning for Urban-Scene Segmentation”. In: *33rd British Machine Vision Conference 2022, BMVC 2022, London, UK, November 21-24, 2022*. BMVA Press, 2022
2. J. Warchocki, T. Opreșcu, Y. Wang, A. Dămăcuș, P. Misterka, R.-J. Brintjes, A. Lengyel, O. Strafforello, and J. van Gemert. “Benchmarking Data Efficiency and Computational Efficiency of Temporal Action Localization Models”. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*. Oct. 2023, pp. 3008–3016

CURRICULUM VITÆ

Attila LENGYEL

9 July 1994 Born in Nové Zámky, Slovak Republic.

EDUCATION

2019–2023 **PhD in Computer Science**
Delft University of Technology, Delft
Thesis: On Color and Symmetries for Data
Efficient Deep Learning
Promotors: Dr. J.C. van Gemert
Prof. dr. ir. M.J.T. Reinders

2017–2019 **Master of Science in Electrical Engineering**
Delft University of Technology, Delft

2012–2017 **Bachelor of Science in Electrical Engineering**
Delft University of Technology, Delft

2006–2012 **Pre-university education (VWO)**
Esdal College, Emmen (2009–2012)
Praedinius Gymnasium, Groningen (2006–2009)

WORK EXPERIENCE

2024 - current **TomTom, Amsterdam**
Applied Scientist

2022 **Google, Zurich**
Research Intern

Propositions


accompanying the dissertation

On Color and Symmetries for Data Efficient Deep Learning

by

Attila LENGYEL

1. Inductive biases provide a good alternative in the absence of sufficient data.
2. The necessity of exact equivariance is application dependent, as approximately equivariant architectures can outperform exact ones.
3. Data augmentations are and remain necessary to complement approximate equivariance.
4. Designing a toy experiment to demonstrate a theoretical deep learning problem is easier than showing that it also applies to the real world.
5. Reporting compute efficiency in terms of multiply-accumulate (MAC) operations is of no practical use.
6. The hype surrounding generative AI obscures practical limitations and challenges, and boosts unrealistic expectations.
7. Academic code writing is in need of a culture shift toward unit tests and code reviews to accelerate research.
8. Imposing restrictions on AI research forms a great danger to society.
9. Engaging in physical exercise is the single most productive thing to do when in a time crunch.
10. Review count should be an equally important scientific metric as citation count.

 Pertains to this dissertation.

These propositions are regarded as opposable and defensible,
and have been approved as such by the promotors
Dr. J.C. van Gemert and Prof. dr. M.J.T. Reinders.