EXPLOITING LEARNED SYMMETRIES IN GROUP EQUIVARIANT CONVOLUTIONS

Attila Lengyel *Jan van Gemert*

Computer Vision Lab, Delft University of Technology, The Netherlands

ABSTRACT

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. All code is publicly available at github.com/Attila94/SepGrouPy.

***Index Terms*—** group equivariant convolutions, depthwise separable convolutions, efficient deep learning

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. 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.

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

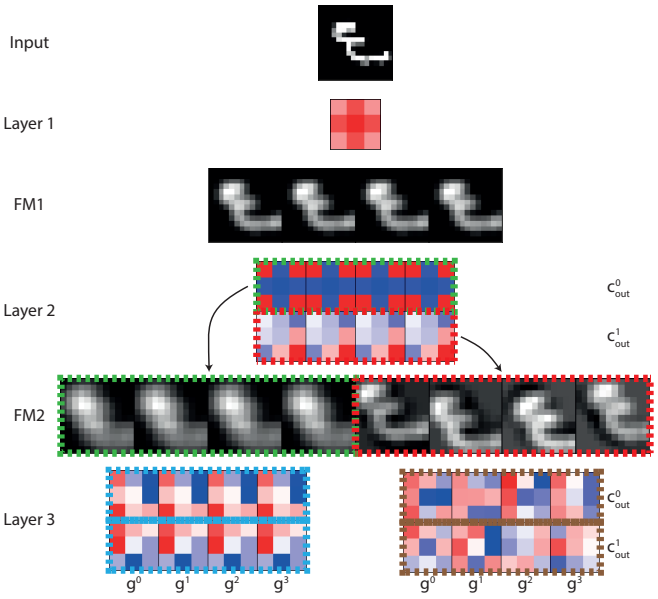


Fig. 1: Filters and feature maps of a GConv architecture trained on Rotated MNIST. Rotation invariant filters in Layer 2 result in identical feature maps FM2 (green) and cause Layer 3 to learn identical weights along the group dimension g (blue). In contrast, non-symmetric filters in Layer 2 (red) result in non-identical filters in Layer 3 (brown).

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, 8, 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 ap-

plied on each individual input channel 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, 14, 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.

3. METHOD

3.1. Group Equivariant Convolutions

Equivariance to a group of transformations G is defined as

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

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 first focus on the group $p4$ of translations and 90-degree rotations, but extend to larger groups later.

Let us denote a regular convolution as

$$X^{l+1}_{n,:,:} = \sum_{c}^{C^l} F^l_{n,c,:,:} * X^l_{c,:,:}, \tag{2}$$

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:

$$X^{l+1}_{n,h,:,:} = \sum_{c}^{C^l} \sum_{g}^{G^l} \tilde{F}^l_{n,h,c,g,:,:} * X^l_{c,g,:,:}. \tag{3}$$

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 $X^1_{c,g,:,:}$ reduces to $X^1_{c,:,:}$, whereas for all following layers G^l=4 for the $p4$ group (and G^l=8 for $p4m$).

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. 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. 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.

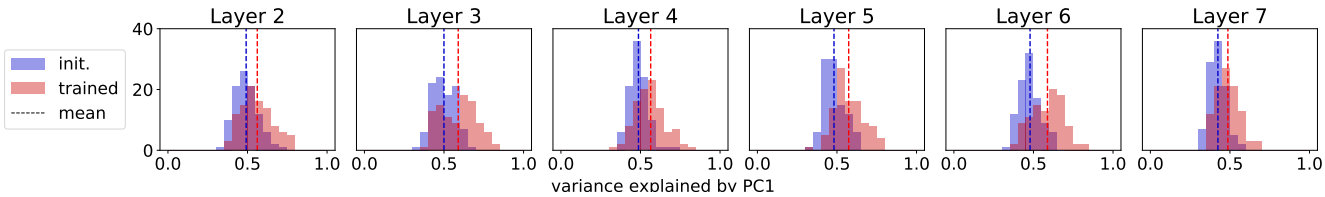


Fig. 2: Ratio of variance explained by the first principal component when decomposing a filter kernel along the group dimension, before (blue) and after (red) training on Rotated MNIST. Redundancy in filter parameters increases as the network converges.

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:

$$F^l_{n,c,g,:,:} = K^l_{n,c,:,:} \cdot w^l_{n,c,g}. \tag{4}$$

The full GConv filter bank is then constructed as

$$\tilde{F}^l_{n,h,c,g,:,:} = T_h(K^l_{n,c,:,:}) \cdot \tilde{w}^l_{n,h,c,g}, \tag{5}$$

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. (3). Alternatively, for better computational efficiency we can substitute the filter decomposition in Eq. (5) into the GConv in Eq. (3) and rearrange as follows:

$$X^{l+1}_{n,h,:,:} = \sum_c^{C^l} \sum_g^{G^l} X^l_{c,g,:,:} * \left(T_h(K^l_{n,c,:,:}) \cdot \tilde{w}^l_{n,h,c,g}\right) \tag{6}$$

$$= \sum_c^{C^l} \sum_g^{G^l} \left(X^l_{c,g,:,:} \cdot \tilde{w}^l_{n,h,c,g}\right) * T_h(K^l_{n,c,:,:}) \tag{7}$$

$$= \sum_c^{C^l} \tilde{X}^l_{n,h,c,:,:} * T_h(K^l_{n,c,:,:}) \tag{8}$$

with

$$\tilde{X}^l_{n,h,c,:,:} = \sum_g^{G^l} \left(X^l_{c,g,:,:} \cdot \tilde{w}^l_{n,h,c,g}\right). \tag{9}$$

Expanding the dimensions of $\tilde{w}^l$ to $[C^{l+1}, G^{l+1}, C^l, G^l, 1, 1]$ we can implement Eq. (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. (8). We refer to this separable GConv variants as g-GConv, denoting the summation variable in Eq. (9).

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

$$F^l_{n,c,g,:,:} = K^l_{n,:,:} \cdot w^l_{n,c,g}, \tag{10}$$

$$\tilde{F}^l_{n,h,c,g,:,:} = T_h(K^l_{n,:,:}) \cdot \tilde{w}^l_{n,h,c,g}. \tag{11}$$

Substituting $\tilde{F}^l$ in Eq. (3) and rearranging yields

$$X^{l+1}_{n,h,:,:} = \sum_c^{C^l} \sum_g^{G^l} X^l_{c,g,:,:} * \left(T_h(K^l_{n,:,:}) \cdot \tilde{w}^l_{n,h,c,g}\right) \tag{12}$$

$$= \sum_c^{C^l} \sum_g^{G^l} \left(X^l_{c,g,:,:} \cdot \tilde{w}^l_{n,h,c,g}\right) * T_h(K_{n,:,:}) \tag{13}$$

$$= \tilde{X}^l_{n,h,:,:} * T_h(K^l_{n,:,:}) \tag{14}$$

with

$$\tilde{X}^l_{n,h,:,:} = \sum_c^{C^l} \sum_g^{G^l} \left(X^l_{c,g,:,:} \cdot \tilde{w}^l_{n,h,c,g}\right). \tag{15}$$

This way the GConv essentially reduces to an inverse depthwise separable convolution with Eq. (15) being the pointwise and Eq. (14) being the depthwise component. This variant is named gc-GConv after the summation variables in Eq. (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 3.4, allows us to increase the network width given the same parameter budget resulting in better overall performance.

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 $[C^l \times G^l \times k^2 \times C^{l+1}]$ parameters in a GConv filter bank, g- and gc-GConvs require only $[C^l \times C^{l+1} \times (G^l + k^2)]$ and $[C^{l+1} \times (C^l \times G^l + k^2)]$, respectively. Similarly, a regular GConv layer performs $[C^l \times G^l \times k^2 \times W \times H \times C^{l+1} \times G^{l+1}]$ MACs, whereas g- and gc-GConvs do only $[C^l \times C^{l+1} \times G^{l+1} \times W \times H \times (G^l + k^2)]$ and $[C^{l+1} \times G^{l+1} \times W \times H \times (C^l \times G^l + k^2)]$, 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.

Table 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 (upper part) and comparable to other equivariant methods (lower part).

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]	$\mathbf{1.74 \pm 0.070}$	30	24.64 k	13.01 M
SFCNN [6]	$\mathbf{0.71 \pm 0.022}$	-	-	-
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	-

4. EXPERIMENTS

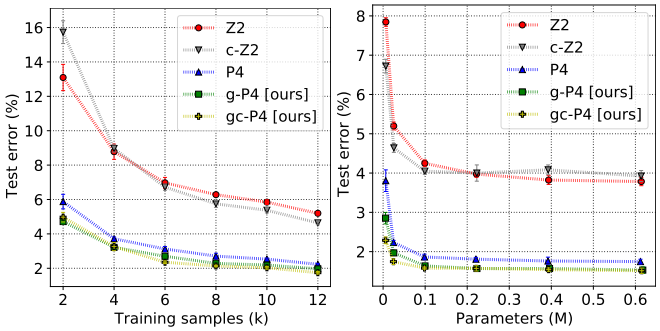
4.1. Rotated MNIST

We construct a g-separable (Eqs. 8-9) and gc-separable (Eqs. 14-15) 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 10 and 20 channels, respectively. We follow the training procedure of [2] and successfully reproduced the results.

Table 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. 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. 3b. Separable GConvs do better for all model capacities.

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 less parameters, as shown in Table 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%.



(a) (b)

Fig. 3: Test error on Rotated MNIST for varying training set (a) and model sizes (b). Architectures with separable GConvs perform consistently better.

Table 2: Test error on CIFAR10 - comparison with other $p4m$-equivariant methods. gc-$p4m$-ResNet44 performs best.

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‡.

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.

6. REFERENCES

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

[2] Taco Cohen and Max Welling, "Group equivariant convolutional networks," New York, New York, USA, 20–22 Jun 2016, vol. 48 of *Proceedings of Machine Learning Research*, pp. 2990–2999, PMLR.

[3] Daniel Haase and Manuel 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," *ArXiv*, vol. abs/2006.16867, 2020.

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

[6] Maurice Weiler, Fred A. Hamprecht, and Martin Storath, "Learning steerable filters for rotation equivariant cnns," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[7] Marysia Winkels and Taco S. Cohen, "Pulmonary nodule detection in ct scans with equivariant cnns," *Medical Image Analysis*, vol. 55, pp. 15 – 26, 2019.

[8] Daniel E. Worrall and Gabriel 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.

[9] Maurice Weiler, Mario Geiger, Max Welling, Wouter Boomsma, and Taco S Cohen, "3d steerable cnns: Learning rotationally equivariant features in volumetric data," in *Advances in Neural Information Processing Systems*. 2018, vol. 31, pp. 10381–10392, Curran Associates, Inc.

[10] Daniel Worrall and Max Welling, "Deep scale-spaces: Equivariance over scale," in *Advances in Neural Information Processing Systems*. 2019, vol. 32, pp. 7366–7378, Curran Associates, Inc.

[11] Ivan Sosnovik, Michał Szmaja, and Arnold Smeulders, "Scale-equivariant steerable networks," in *International Conference on Learning Representations*, 2020.

[12] Laurent Sifre and Prof Stéphane Mallat, "Ecole polytechnique, cmap phd thesis rigid-motion scattering for image classification author:," 2014.

[13] A. Howard, Mark Sandler, G. Chu, Liang-Chieh Chen, B. Chen, M. Tan, W. Wang, Y. Zhu, R. Pang, V. Vasudevan, Quoc V. Le, and H. Adam, "Searching for mobilenetv3," *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 1314–1324, 2019.

[14] Mingxing Tan and Quoc Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in *Proceedings of the 36th International Conference on Machine Learning*. 2019, vol. 97 of *Proceedings of Machine Learning Research*, pp. 6105–6114, PMLR.

[15] Mirgahney Mohamed, Gabriele Cesa, Taco S. Cohen, and Max Welling, "A data and compute efficient design for limited-resources deep learning," 2020.

[16] Hugo Larochelle, Dumitru Erhan, Aaron Courville, James Bergstra, and Yoshua Bengio, "An empirical evaluation of deep architectures on problems with many factors of variation," in *Proceedings of the 24th International Conference on Machine Learning*, New York, NY, USA, 2007, ICML '07, p. 473–480, Association for Computing Machinery.

[17] Junying Li, Zichen Yang, Haifeng Liu, and Deng Cai, "Deep rotation equivariant network," *Neurocomputing*, vol. 290, pp. 26–33, 2018.

[18] Daniel E. Worrall, Stephan J. Garbin, Daniyar Turmukhambetov, and Gabriel 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] David Romero, Erik Bekkers, Jakub Tomczak, and Mark Hoogendoorn, "Attentive group equivariant convolutional networks," in *Proceedings of the 37th International Conference on Machine Learning*. 13–18 Jul 2020, vol. 119 of *Proceedings of Machine Learning Research*, pp. 8188–8199, PMLR.

[20] David W. Romero and Mark Hoogendoorn, "Co-attentive equivariant neural networks: Focusing equivariance on transformations co-occurring in data," in *International Conference on Learning Representations*, 2020.

[21] Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton, "Cifar-10 (canadian institute for advanced research)," .