A Study on Group Equivariant CNNs

Polina BARABANSHCHIKOVA
Institut Polytechnique de Paris
pbaraban-23@ip-paris.fr

Anshuman SINHA
Institut Polytechnique de Paris
anshuman.sinha@ip-paris.fr

ABSTRACT

Group equivariance in Convolutional Neural Networks (CNNs) develop an intuitive generalizable strategy on regular CNNs. G-CNNs increase the representative capacity of such models with minimal computational overhead. For now, these networks deal with discrete groups such as rotations, reflections and translations. This report aims at giving a brief summary of the theory, methodology and provides a detailed analysis of further experiments conducted on segmentation and classification tasks using the same, with their theoretical and computational connotations.

All implementations are made publicly available at: https://github.com/pollinab/GroupEquivariantCNN/.

KEYWORDS

Groups, Convolutional Neural Networks, Equivariance, Invariance

. . In *Proceedings of MVA Project Report (Geometric Data Analysis).*

1 MOTIVATION

Deep Convolutional Neural Networks (CNNs) excel in processing sensory data by leveraging weight sharing in convolutional layers and exploiting the inherent translation symmetry in perception tasks. It enables CNNs to analyze image components using the same weights, reducing parameters while retaining the ability to learn diverse transformations. Moreover, CNNs maintain translation equivariance across all layers, preserving symmetry, that can be defined mathematically through groups of transformations. The base paper [2] extends CNNs to be equivariant to larger symmetry groups like rotations and reflections, enhancing their capacity for comprehensive representation learning.

Geometric Data Analysis, 2023, MVA Project Report

The paper introduces representations in a space where each vector possesses an associated pose transformable by a pre-defined group of transformations, denoted by G. This setup allows G-CNN filters to detect feature co-occurrences with specific relative poses and match them across various global poses using G-convolution. Additionally, it underscores the significance of structure-preserving layers Φ that map one representation space to another, emphasizing the need for transformations to uphold the structure between connected representation spaces. For G-spaces this means that Φ has to be equivariant:

$$\Phi(T_g\, x) \;=\; T_g^{'}\, \Phi(x) \tag{1}$$

In the applications of deep learning, general equivariance is more beneficial than invariance as it helps determine whether features are in the right spatial configuration. Moreover, equivariance preserves the rotation (or any other transformation) dependent features of the data whereas invariance loses it, as we see in the latter sections. Our work applies algorithms from the article and reproduces results for the rotated MNIST dataset (Rot-MNIST). The contribution extends to the exploration of a segmentation task on a histopathology image dataset. For both tasks, we see results on how the CNN feature maps are not rotation equivariant whereas G-CNNs are. Such geometrical guarantees naturally make such models more reliable for several tasks.

In segmentation, regular CNNs do not have any checks in place to ensure that an output segmentation map gets transformed in the same way that the input is. For example, a user naturally expects a trained model to segment the rotated image of a blood sample exactly in the same way as it segments an upright image. Group Equivariant CNNs with rotation as one of its groups, has strong theoretical underpinnings to guarantee the previous results. On the other hand, regular CNNs can also be made to act similarly equivariant by being fed augmentations of the data, which in turn multiplies the training time and demands higher computational resources.

This is especially interesting as nowadays deep learning also caters to the medical field where data is not always as much in abundance as necessary to reach the accuracies acceptable to healthcare. Hence, methods that induce inherent equivariances in data prevent explicit augmentation

computations altogether. Experiments showcased include varying degrees of discreteness within particular groups. Discussions involve analyses on the actions of hidden layers of these models.

2 THEORY

The underlying symmetries of data typically have the mathematical structure of a group. Formally, a set of symmetric transformations G is closed under the composition of its elements and satisfies three group axioms: associativity, existence of the identity element, and existence of the inverse element.

For instance, it is easy to see that a set of translations of a grid is a group under the operation of composition. Its elements are identified with vectors in $\mathbb{Z}^2$ and a composition of two translations t_1 and t_2 is a translation by vector $t_1 + t_2$. We may say that a translation t modifies a grid by moving each point by the vector t. This concept is formalized by the notion of the group action.

If G is a group with identity element e, and X is a set, then a group action of G on X is a function $\alpha : G \times X \to X$ such that for all $x \in X$, $gh \in G$:

$$\alpha(e,\ x) \ = \ x \tag{2}$$

$$\alpha(g,\ \alpha(hx)) \ = \ \alpha(gh,\ x) \tag{3}$$

We will use the standard simplified notation: $gx \coloneqq \alpha(g, x)$. Hence, the action of the group of translations on a grid is given by $tx = t + x$.

In machine learning settings, one deals with images and feature maps, represented by functions defined on the grid of points. Therefore, it is necessary to define a group action on a function. For a function $f : X \to \mathbb{R}^K$, the action of a transformation $g \in G$ on f is an operator L_g given by

$$[L_g\, f](x) = f(g^{-1}x) \tag{4}$$

For example, to shift the value of the image by t at the point x, we have to find the value at the point $x - t$ in the original image.

Now, we can formalize the concept of G-equivariance. An operator Φ that maps one representation to another one is called G-equivariant if $\forall\, f : X \to \mathbb{R}^K$, $x \in X$, $g \in G$

$$[L_g\ [\Phi\, f]](x) = [\Phi\ [L_g\, f]](x) \tag{5}$$

By definition, if Φ and Ψ are G-equivariant operators, then their composition is also G-equivariant:

$$[L_g[\Phi \cdot \Psi\, f]](x) = [\Phi\ [L_g\ \Psi\, f]](x) = [\Phi\Psi[L_g\, f]](x) \tag{6}$$

Therefore, to construct a G-equivariant neural network it is sufficient to ensure that its layers are G-equivariant.

In the base paper [2], the authors propose a G-equivariant generalization of convolutional layers. A G-correlation operator $\star$ convolves a stack of feature maps $f : X \to \mathbb{R}^K$ with a set of filters $\psi : X \to \mathbb{R}^K$:

$$[f \star \psi](g) = \sum_{x \in X} \sum_k f_k(x)\psi_k(g^{-1}x) \tag{7}$$

for every $g \in G$.

Note that this operator maps a function f on the set X to the function $[f \star \psi]$ defined on the group G. Therefore, in G-CNN, filters of all layers after the first must be defined on G. The first layer is called a lifting convolution since it maps the space X to the more complex space G.

Simple estimations show that $\forall\, h \ \in \ G$,

$$[L_h[f \star \psi]](g) = [[L_h f] \star \psi](g), \tag{8}$$

thus, G-correlation is G-equivariant.

Moreover, the authors of [2] show that non-linearities and batch normalization across G-feature maps preserve the transformation properties of the previous layer. Subsampling operator is H-equivariant only in the case of subsampling on the subgroup $H \subset G$.

3 IMPLEMENTATION

As described in the paper [2], G-convolutions can be efficiently implemented for discrete groups with *split* property. This property guaranties that each element $g \in G$ is a composition of a translation $t \in \mathbb{Z}^2$ and a transformation s that acts trivially on the origin. The set of all elements that does not transform the origin is also called the stabilizer of the origin.

In case of the *split* group, G-correlation takes the form

$$\begin{aligned}[f \star \psi](ts) &= \sum_{x \in X} \sum_k f_k(x)\psi_k(s^{-1}t^{-1}x) \\ &= \sum_{x \in X} \sum_k f_k(x) L_t[L_s\psi_k](x).\end{aligned}$$

Here, the terms $[L_s\psi_k]$ denote "filter transformations" that can be pre-computed for all elements of the stabilizer. Subsequently, the set of transformed filters is used in an efficient planar correlation procedure applied to the feature map f.

Then, the additional computational costs depend only on the size of the stabilizer.

Below, we describe the structure of several *split* groups and the corresponding stabilizers.

3.0.1 ***The groups*** $p4$ ***and*** pn. $p4$ is a group of translations and rotations by multiples of 90 degrees in a grid. An element g of this group can be decomposed into an integer translation $t \in \mathbb{Z}^2$ and a rotation $r \in C_4$, where C_4 is a cyclic group of order 4. The group element $g = (t, r)$ acts on the point of the grid by first rotating this point r times and then translating it with t.

In our experiments, we exploit a generalized version of this group, denoted as pn. This group includes all compositions of translations and rotations by angles of $2\pi/n$. Its stabilizer of the origin is a cyclic group of order n.

3.0.2 ***The group*** $p4m$. The $p4m$ group consists of translations, mirror reflections, and rotations by 90 degrees around any center in a square grid. An element g in this group can be identified with an integer translation $t \in \mathbb{Z}^2$ and a flip-rotation $(r, f) \in D_4$, where D_4 is a dihedral group. The group element $g = (t, r, f)$ acts on a grid point by first flipping it $f \in \{0, 1\}$ times, then rotating it $r \in \{0, 1, 2, 3\}$ times, and finally translating it by t.

4 EXPERIMENTS

4.1 Classification

Classification tasks in image processing usually involve a set of images containing one (or more in a few cases) theme per image. Therefore, the dataset contains a finite number of such classes that each image is assigned to. CNN models to solve such tasks usually output a confidence score pertaining to each class and the highest score gets assigned to the predicted value.

For example, CIFAR-10 containing 10 classes such as airplanes, dogs, cats, etc. For a sample image from the dataset, a reliable model is expected to make identical predictions irrespective of whether the image is rotated by a specific angle or has undergone an affine translation. In other words, we expect such a model to ideally be invariant to those groups of transformations.

4.1.1 ***Dataset***. MNIST is a labelled dataset containing 60,000 images of handwritten digits. Being one of the classic image classification datasets, basic techniques such as KNNs can solve the task with high accuracies of over 95%.

In this project, we propose to try and compare traditional CNNs against group-equivariant ones on a modified version of the dataset. Herein, the images are randomly rotated by an angle in $[0,\ 2\pi)$. Needless, to say this is a rotation-invariant task hence, the labels remain the same. This new dataset is known as the Rot-MNIST.

4.1.2 ***Model***. Our chosen base model is a CNN architecture with 5 layers of 3×3 convolutions with 8, 16, 32, 64 and 128 channels in the respective layers, ReLu activation functions, batch normalization, dropout, and 3D max-pooling.

Next, for P4CNN architecture we replaced each convolution layer by a p4-group convolution layer and divided the number of filters by $\sqrt{4} = 2$ to approximately fix the total number of trainable parameters. For the subsequent models P6CNN and P8CNN, the number of channels is similarly adjusted to keep the total parameters constant. As we see in table 1, P4CNN almost doubles the accuracy of models trained on MNIST and tested on randomly rotated images. The resulting feature maps [fig: 1] consist of rotation-invariant features with the same transformations as the input image.

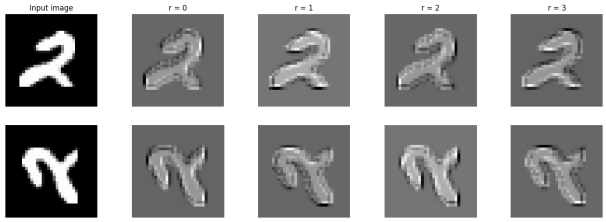

Figure 1: Rotation equivariant feature maps

4.1.3 ***Results***. First the different models were trained on raw MNIST images (original unrotated dataset) for 30 epochs and varying the number of channels to keep the net trainable parameters constant, as mentioned previously.

The table 1 shows the results of these trained models tested on the MNIST test set on the second column and the accuracy of testing on he same images with random rotations between 0 and 180°.

Table 1: Accuracy of models trained on MNIST

Model	MNIST test accuracy	MNIST (+transforms) test accuracy
CNN	**98.2**%	34.1%
P4CNN	95.8%	62.9%
P6CNN	96.0%	42.0%
P8CNN	95.9%	63.5%
D4CNN	94.2%	**79.6**%

As expected, for the testing on the original test set, the CNN performs best as it does not spend any of its parameters tying to encode rotational information. Since, the rotational and reflection equivariance is not beneficial to this set of images, there is no benefit in training those parameters. As a result we see a clear decline in performance over the other equivariant models.

Secondly, we also test on randomly transformed (rotations, affine translations and flips) images from the test set. The intention of this exercise is to prove the effectiveness of group equivariant models. At once, it can be observed that the CNN accuracy drops to a third. The best accuracy is seen for the D4CNN model with p4m-equivariant layers. A closer inspection reveals better accuracies for P4 and P8 than for P6. The reasoning behind this may be hypothesised as P6 lacking the multiples of 90°, hence only having interpolated pixel values.

Another interesting question might be as to why there is a difference between the accuracies of D4CNN between the two testing sets if it is group equivariant. The reasoning is simple and slightly disheartening, when one ideally thinks of the objectives of the technique. The rotation group that $D4$ is equivariant to, only consists of the 4 discrete multiples o $\pi/2$ and not the entire continuous set of rotation angles. As a result, it is specifically equivariant to a select few angles in the range. However, in practice, this equivariance produces substantially good feature maps for angles in between by interpolation.

Table 2: Accuracy of models on MNIST Rotated

Model	Test Accuracy
CNN	92.43%
P4CNN	**96.17**%
P6CNN	94.82%
P8CNN	96.01%
D4CNN	93.95%

The table 2 shows the results of the same models trained and tested on the MNIST-Rot dataset as described in 4.1.1.

As expected, the CNN model performs the worst, due to the unavailability of learnable parameters to explicitly encode rotational features. $P4$ and $P8$ outperform $P6$ again, for similar reasons as cited previously. $D4CNN$ does not give impressive results because there are no mirroring transformations in the dataset, for it to be equivariant to. As a result, it is effectively a version of $P4$ with fewer channels.

4.2 Segmentation

The task of semantic segmentation can be formulated as follows. For a space of images $\mathcal{X} \in \mathbb{R}^{C\times H\times W}$ and a finite label set $\mathcal{Y}$, the goal is to find the map $f : \mathcal{X} \to \mathcal{Y}^{H\times W}$ that determines the class for each pixel of the image. This mapping function is typically learned from a labeled training dataset that consists of pairs $\{X, Y\}$, where $X \in \mathcal{X}$ is an image and $Y \in Y^{H\times W}$ is the corresponding mask.

In most natural cases, the segmentation map is equivariant to the affine transformations of the input image, that is $f(t(X)) = tf(X)$. This makes segmentation task an ideal use-case for the equivariant models. We restrict our attention to the binary segmentation, namely $\mathcal{Y} = \{0, 1\}$.

For the experiments we took the PH2 Dataset for lesion segmentation. It contains 200 images of dermascopic images paired with lesion masks that we divided into train, validation and test sets of size 100, 50, and 50, respectively. Additionally, we build a "test rotated" set by rotating test images by multiples of 90 degrees. The skin images are naturally equivariant to shifts, rotations, and flips.

4.2.1 ***Model analysis.*** For the baseline model we chose a encoder-decoder architecture based on the SegNet ([1]). Both the encoder and the decoder consist of 4 blocks with 2 layers of 3×3 convolutions, batch normalizations and ReLu activations. Each encoder block is followed by a max-pooling operator and each decoder block is preceded by an upsampling.

To construct a group equivariant model we introduce the following changes: first, we replace the first convolution with a lifting convolution and all other convolutions with group equivariant convolutions. Then, we replace 2D batch normalizations with 3D batch normalizations to ensure that for each G-feature map a single scale and bias parameter is computed. Finally, we adapt the size of the input image and restrict maxpooling operators to the spatial dimension of the G-feature maps, so that subsampling is done on the subgroup $H \subset G$ containing all rotations, flips (if $G = p4m$), and shifts by multiples of 2 pixels. Moreover, we add a final maxpooling layer over the stabilizer dimension to obtain two dimensional prediction.

We note that our model is not fully translation equivariant. It happens because of two reasons. The first one is edge-effects that occur because padding is unaware of the extension of the real image. Moreover, each time we subsample on a subgroup $H \subset G$ the output feature maps becomes equivariant to H instead of G. Both of these effects appear in all kinds of convolutional neural networks. In contrast,

we emphasize that the final maxpooling layer is equivariant to all transformations of its input. Indeed, this layer can be modeled as an operator P that maps a G-feature maps $f : G \to \mathbb{R}$ to a feature map $Pf : G/H \to \mathbb{R}$ by the rule

$$Pf(g) = \max_{k \in gH} f(k), \tag{9}$$

where g is a representative of an equivalence class in G/H. In our case $H \in G$ is a stabilizer of the origin.

To verify the correctness of the definition, we check that it does not depend on the choice of the representative. Since H is a group, for every $g \in G$, $h \in H$, we have

$$Pf(gh) = \max_{k \in ghH} f(k) = \max_{k \in gH} f(k) = Pf(g) \tag{10}$$

We claim that upsampling preserves the transformation properties of the previous layers. In fact, it is a special case of the transposed convolution layer with constant filters of size $n \times n$ and stride n. The transposed convolution with stride transforms a stack of H-feature maps into a G-feature map, where H is a subgroup of G. It is modelled by the operator

$$[f * \psi](g) = \sum_{h \in H} \sum_{k} f_k(h)\ \psi_k(h^{-1}g) \tag{11}$$

The equivariance of this operator is derived in analogy to the equivariance of the G-correlation, by the substitution $h \to uh$:

$$\begin{aligned}
[[L_u\, f]\ *\ \psi](g) &= \sum_{h \in H} \sum_{k} f_k(u^{-1}h)\ \phi_k(h^{-1}g) \\
&= \sum_{h \in H} \sum_{k} f_k(u^{-1}h)\ \phi_k(h^{-1}g) \\
&= \sum_{h \in H} \sum_{k} f_k(h)\ \phi_k(h^{-1}(u^{-1}g)) \\
&= [L_u\, [f * \psi]](g).
\end{aligned}$$

Hence, the upsampled feature map is equivariant to H.

4.2.2 ***Results.*** In the first set of experiments we compare our SegNet model and its equivariant versions for groups p4 (P4SegNet) and p4m (D4SegNet) on the segmentation task. The numbers of filters are chosen independently for each model so they all have approximately the same number of parameters.

For each model, we computed Intersection over Union (IoU) metric on the test and "rotated test" sets. In addition, we trained SegNet and p4-equivariant SegNet on the train set augmented with random flips and rotations by multiples of 90 degrees, we refer to these models as SegNet+ and P4SegNet+.

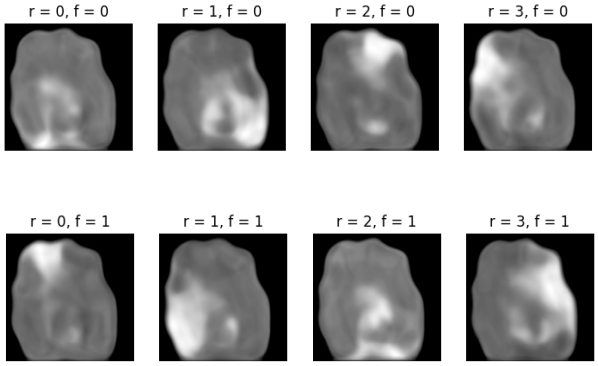

Figure 2: Feature maps taken from the last layer of D4SegNet before pooling across the stabilizer dimension

The experimental results are presented in Table 3. Figure 3 shows the predictions of the models. P4SegNet outperforms other models when training with and without augmentations. Although, since the set of augmentations is mostly covered by the group p4, P4SegNet+ demonstrates the same performance as P4SegNet.

Table 3: Accuracy of models based on SegNet

Model	test IoU	rotated test IoU
SegNet	0.718	0.646
P4SegNet	**0.786**	**0.786**
D4SegNet	0.745	0.745
SegNet+	0.767	0.771
P4SegNet+	**0.788**	**0.788**

As expected, since P4SegNet and D4SegNet are equivariant to rotations they show same results on the test and the "rotated test" sets. At the same time the SegNet model does not generalize well on the rotated data, experiencing a considerable drop in the score. However, when trained with augmentations it learns to capture data symmetries and beat D4SegNet on both test and the "rotated test" sets. But since the scores on the two sets differ, we conclude that SegNet+ is not fully equivariant to rotations.

It is important to note that although SegNet+ is inferior to P4SegNet and does not come with equivariance guaranties, it trains almost 1.5 faster.

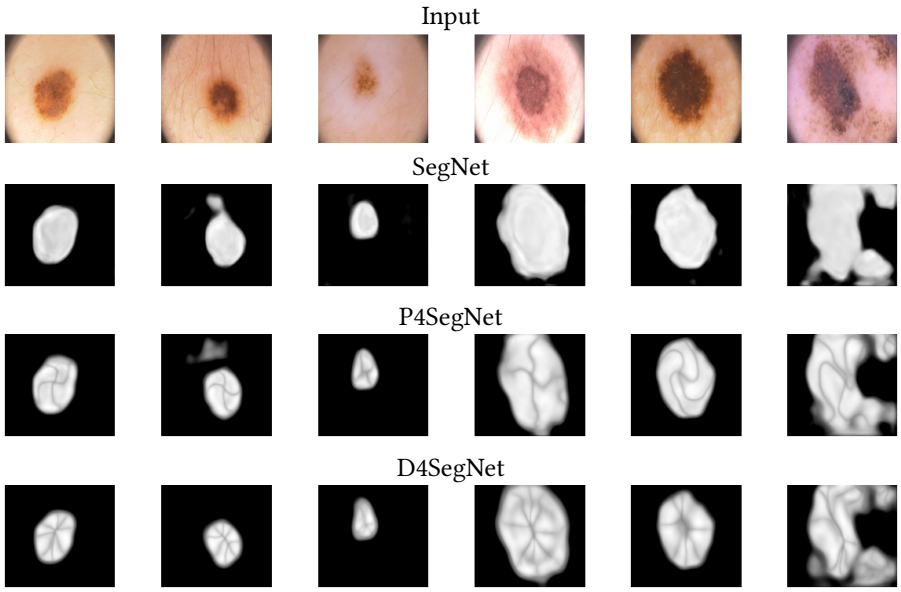

Figure 3: Predictions of the segmentation models

For the second set of experiments, we modified our models by replacing upsampling layers with transposed convolutions. The resulting models are called SegNet$_T$, P4SegNet$_T$ and D4SegNet$_T$. The IoU scores computed for this models on train and test subsets are presented in Table 4. D4SegNet$_T$ demonstrates the best result on the test set and it also appears to overfit less on the training data. Thus, we may hypothesize that enhancing weight-sharing can prevent overfitting.

Table 4: Accuracy of SegNet models with Transposed Convolutions

Model	train IoU	test IoU
SegNet$_T$	0.829	0.657
P4SegNet$_T$	0.842	0.67
D4SegNet$_T$	0.834	**0.693**

5 CONCLUSION

In all, for specific datasets and tasks that require the model to learn rotational and reflection symmetries through equivariance, this technique works very effectively, even for low resolutions of $SO(2)$ such as in the $p4$ algorithm.

However, a drawback of these methods is that they are not completely rotation equivariant. This argument follows the fact that they are equivariant to certain discrete rotation elements of $SO(2)$ sampled at a certain rate.

Some other works of literature provide ways to represent group elements in different ways, for example using steerable filters [3] authored by the same people. This paper received state of the art results on the CIFAR dataset They went on to extend that theory into 3D Steerable CNNs [4] designing $SE(3)$ equivariant convolutions. The effectiveness was measured on protein structure classification datasets.

6 ACKNOWLEDGMENTS

We acknowledge the support of Professor Erik Bekkers' online lectures on Group Equivariant CNNs from the Amsterdam Machine Learning lab at the University of Amsterdam in the Netherlands. We are also grateful towards our course supervisor Dr Jean Feydy, for giving us the opportunity to learn and explore the beautiful concept of Group Equivariant CNNs in depth.

REFERENCES

[1] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. 2015. SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. In *CoRR*. Vol. abs/1511.00561.

[2] Taco Cohen and Max Welling. 2016. Group equivariant convolutional networks. In *Proceedings of The 33rd International Conference on Machine Learning*. Vol. 48. (20–22 Jun 2016), 2990–2999.

[3] Taco Cohen and Max Welling. 2017. Steerable CNNs. In *International Conference on Learning Representations*.

[4] Maurice Weiler, Mario Geiger, Max Welling, Wouter Boomsma, and Taco Cohen. 2018. 3D Steerable CNNs: Learning Rotationally Equivariant Features in Volumetric Data. In *32nd Conference on Neural Information Processing Systems*.