Group Equivariant Convolutional Networks

GUILLAUMÉ Romain
ENS Paris Saclay, Mines Paris PSL
Paris, France
romain.guillaume@etu.minesparis.psl.eu

LEBOUCHER Grégoire
ENS Paris Saclay, Mines Paris PSL
Paris, France
gregoire.leboucher@etu.minesparis.psl.eu

INTRODUCTION

Empirical studies have shown that weight sharing and depth in the network are two key factors to obtain some good results in Convolutional Neural Networks. The translation symmetry present in CNNs allows to obtain some good results in both domains. Perception tasks are usually translation invariant: the objects the model wants to detect can be anywhere on the image. That is why it is possible to use the same weights for pixels from very different positions on an image, and thus use an architecture that contains this symmetry (CNN). This technique actually "shares weights", leading to a more efficient training and some better results than a traditional neural network technique. Moreover, convolutional layers are translation equivariant: adding the same translation transformation on the features or the outputs of your layer will give you the same result. This unlocks the possibility to stack up the layers while preserving this symmetry, permitting those networks to be deep.

Imposing some translation symmetries increases the performances of the perception tasks models. This observation can be declined to a more generalized approach, concerning larger groups of symmetries, involving reflections and rotations. This idea is supported by the fact that the content of an image does not change according to any of these symmetries, the desired output of the model is then the same.

Group theory provides a rigorous mathematical framework for describing and working with transformations that preserve structure. A group is a nonempty set G equipped with a binary operation $* : G \times G \rightarrow G$ respecting the following conditions:

- Closure: if $a, b \in G, a * b \in G$
- Associativity: for $a, b, c \in G, a * (b * c) = (a * b) * c$
- Identity: there is an element e in G such that $\forall a \in G, a * e = e * a = e * a = a$
- Inverse: For all a in G, there exists b in G such that: $b * a = e = e = a * b$

The authors, Welling and Cohen, focus on three specific groups related to geometric symmetries of the lattice $\mathbb{Z}^2$:

- **The group** $\mathbb{Z}^2$ represents the group of **translations** on an image: if $(a, b), (m, n) \in \mathbb{Z}^2, (a + m, b + n) \in \mathbb{Z}^2$ and represent the point (a, b) translated by (m, n).
- **The group p4** represented by matrices of the form:

$$g(r, u, v) = \begin{bmatrix} \cos(r\pi/2) & -\sin(r\pi/2) & u \\ \sin(r\pi/2) & \cos(r\pi/2) & v \\ 0 & 0 & 1 \end{bmatrix}.$$

  with $r \in \{0, 1, 2, 3\}$ and $(u, v) \in \mathbb{Z}^2$. This group represents all the compositions of translations and rotations by 90 degrees about any center of rotation in a square grid. Any point of the feature maps is represented as $x = \begin{bmatrix} u' \\ v' \\ 1 \end{bmatrix}$ such that:

$$g \cdot x = \begin{bmatrix} \cos(r\pi/2) & -\sin(r\pi/2) & u \\ \sin(r\pi/2) & \cos(r\pi/2) & v \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u' \\ v' \\ 1 \end{bmatrix}$$

- **The group p4m** represented by the matrices:

$$g(m, r, u, v) = \begin{bmatrix} (-1)^m \cos(r\pi/2) & -(-1)^m \sin(r\pi/2) & u \\ \sin(r\pi/2) & \cos(r\pi/2) & v \\ 0 & 0 & 1 \end{bmatrix}.$$

with $m \in \{0, 1\}$ with $r \in \{0, 1, 2, 3\}$ and $(u, v) \in \mathbb{Z}^2$. This group represents all the compositions of translations, reflexions and rotations by 90 degrees about any center of rotation in a square grid.

To develop the idea of conserving symmetries, the article focuses on linear G-spaces based on a group G (either $\mathbb{Z}^2$, p4 or p4m), where every element of the group G represents a transformation of the G-space. The aim is to build a network which maps symmetric inputs to symmetric outputs. The mathematical notion encoding this idea is called equivariance. A function, network or layer $\Phi : X \rightarrow Y$ is said to be equivariant regarding the G-spaces X and Y if for any $g \in G$:

$$\forall x \in X \quad \Phi(T_g\, x) = T_g^{'} \Phi(x)$$

Here, T and T' are both linear representations of G, but not necessarily the same (T acts on X, T' acts on Y). This definition of equivariance highlights the fact that invariance is a specific case of equivariance:

$$\Phi(T_g\, x) = \Phi(x) = T_g^{'} \Phi(x) \quad (T_g^{'} = \mathrm{id} \ \ \forall g \in G)$$

1 CONTEXT

In 2004, Lowe [7] first introduced this idea of looking at invariance inside image detections, introducing SIFT (Scale Invariant Feature Transform), identifying the similarities between the object on the image and the targeted object. The idea is that, two similar objects, even if the context is different, will have a similar SIFT whereas different objects will have a distant SIFT. Similarly, Jaderberg et al [6] introduced in 2015 Spatial Transformer Networks (STNs), that are a mechanism for dynamically learning spatial transformations of input data to achieve invariance to geometric variations, such as rotations, translations, and scaling.

The idea of leveraging symmetries in data to improve efficiency is also present in [2] the Bruna and Mallat article (2013) , by using predefined wavelet filters and group averaging to create features that are stable to deformations and invariant to specific transformations. This method reuses the SIFT technique proposed by Lowe et al. This article sets the scene for the Sifre and Mallat (2013) [9]. They introduced scattering transforms invariant to rotations translations and scaling. This enabled the architecture to handle more complex geometric transformations in data and to obtain some unprecedented results with it on image recognition tasks.

Gens and Domingos [5] also proposed to include symmetries in Convolutional Networks proposing Deep Symmetry Networks (symnets). This architecture generalizes CNNs, adding feature maps created over arbitrary symmetry groups. This idea was then applied by Dieleman et al (2015) [4] on the specific problem of galaxy morphology prediction.

Aditionnally, Welling and Cohen [3] presented in 2014 a new probabilistic model of compact commutative Lie groups, a class of continuous groups that contains translations and rotations. This model creates equivariant and untangled representations of the data, which set the stage for viewing G-CNNs as mechanisms for disentangling transformations.

G-CNNs can finally be seen as a direct synthesis of all the research made on group equivariance within convolution networks, using the theoretical foundations on scattering networks and disentangled transformations, the extensions made to some complex equivariant groups (rotations, reflections), to propose a new architecture for convolutional networks.

2 THE G-CONVOLUTION

2.1 Definition and properties

Before defining group convolutions, Welling and Cohen [1] start from regular convolutions and prove the famous property that the convolution layer, at the heart of the Convolutional Neural Network architecture, is translation equivariant.

To make this statement more precise, we first define some terms: a $\mathbb{Z}^2$-feature map is a function $\mathbb{Z}^2 \to \mathbb{R}$ supported on a finite set (usually rectangular). A G-feature map (where G is a discrete group) is defined similarly. In a CNN, inputs and outputs of convolutional layers are stacks of $\mathbb{Z}^2$-feature maps i.e. images with K channels ($\mathbb{Z}^2 \to \mathbb{R}^K$).

We consider G a symmetry group of $\mathbb{Z}^2$. We can naturally extend its action to the space of images $\mathbb{Z}^2 \to \mathbb{R}^K$ by defining the action L:

$$\forall x \in \mathbb{Z}^2 \quad [L_g f](x) = f(g^{-1} \cdot x) \tag{1}$$

One can easily verify that this defines an action (i.e. $L_g L'_g = L_{gg'}$, $(L_g)^{-1} = L_{g^{-1}}$ and $L_{\mathrm{id}} = \mathrm{id}$).

Welling and Cohen prove that for images, the convolution operator $*$ with respect to a filter $\psi : \mathbb{Z}^2 \to \mathbb{R}^K$, which maps $f : \mathbb{Z}^2 \to \mathbb{R}^K$ to the feature map $f * \psi : \mathbb{Z}^2 \to \mathbb{R}$, is translation equivariant. That is, for any translation $t : \mathbb{Z}^2 \to \mathbb{Z}^2$, we have $(L_t f) * \psi = L_t(f * \psi)$, where $*$ is defined by:

$$\forall s \in \mathbb{Z}^2 \quad [f * \psi](s) = \sum_{k=1}^{K} \sum_{x \in \mathbb{Z}^2} f_k(x) \psi_k(s - x) \tag{2}$$

We are looking to generalize this definition of the convolution in order to create convolutions that are equivariant to other kind of symmetries (not just integer translations). To this end, we notice that the variable s in (2) (i.e. the input of the resulting convoluted function), could be not interpreted as a position in $\mathbb{Z}^2$ but as an element of the translation group of $\mathbb{Z}^2$, the group for which this convolution is equivariant! It is somehow "fortuitous" that this group happens to be isomorphic to $(\mathbb{Z}^2, +)$ itself. Indeed, this isomorphism causes the convolution by ψ to be an endomorphism of the space $\mathbb{Z}^2$-feature maps. This may give us the impression that the convolution of an image ought to be an image, but requiring that much structure is not necessary.

Realizing that, we may try to replace s in (2) by an element $g \in G$ in order to obtain a G-equivariant convolution. This would make $f * \psi$ a G-feature map. We should understand $s - x$ as $-(g^{-1} \cdot x)$, which is well defined because by definition G is a symmetry group of $\mathbb{Z}^2$. We obtain a definition for a G-convolution mapping $\mathbb{Z}^2$-feature maps to G-feature maps:

$$[f * \psi](g) = \sum_{k=1}^{K} \sum_{x \in \mathbb{Z}^2} f_k(x) \psi_k(-(g^{-1} \cdot x)) \tag{3}$$

Just as for the previous definition, $f * \psi$ is finitely supported if both f and ψ are.

The layer coming after a G-convolution in the network would thus have to accept a G-feature map as input (or a stack of them if $K > 1$ filters were used like in CNNs). On that account the authors [1] define a G-convolution that maps stack of G-feature maps to G-feature maps by adapting

in a straightforward manner the previous definition (since G naturally acts on itself):

$$[f * \psi](g) = \sum_{k=1}^{K} \sum_{h \in G} f_k(h) \psi_k(h^{-1}g) \tag{4}$$

Here the filter ψ is a function $G \to \mathbb{R}^K$.

In a very similar manner than for the regular convolutions, the authors showed that these G-convolutions are G-equivariant.

Together with its bias and point-wise activation function, a complete G-convolution layer can be written:

$$f^{(i+1)} = \sigma \circ (f^{(i)} * \psi + b)$$

Since adding a constant bias and point-wise activation functions are equivariant, the complete layer is G-equivariant.

2.2 Pooling and subsampling

Pooling and subsampling layers are ... . Formally, we define the max-pooling (non strided) layer P mapping G-feature maps to G-feature maps as:

$$(Pf)(g) = \max_{h \in gU} f(h)$$

where $U \subset G$ is usually a neighborhood of the identity transformation of G, and gU is the g-shifted version of U. One easily verifies that P is G-equivariant.

A pooling layer is often followed by a subsampling layer which only keeps a subset of the G-feature map. Requiring that a pooling + subsampling layer is G-equivariant reduces drastically the possibilities: the neighborhood U must be a subgroup $H \subset G$ and the subsampling set must be the corresponding quotient space G/H.

For $p4$, the subsampling can be done over the group of four rotations $R = \{r^0, r^1, r^2, r^3\}$ where r is the 90° rotation. This yields a $p4/R$-feature map, which is identified with a $\mathbb{Z}^2$-feature map since $p4/R \simeq \mathbb{Z}^2$ (illustration on Figure 1).

We noticed that in the case of regular CNNs, subsampling is often done on a set such as $2\mathbb{Z}^2$ so that patches of 2 by 2 pixels get represented by one pixel. According to the previous reasoning applied to $G = \mathbb{Z}^2$, such a pooling + subsampling layer is not translation equivariant (it is equivariant for the subgroup $2\mathbb{Z}^2 \subset \mathbb{Z}^2$). However this layer is still used in CNNs because it is convenient and almost translation equivariant given that the pixels values are often continuous.

2.3 G-convolutions are all you need

In the section, we will go a bit further than the authors in their paper [1], showing something stronger and more general. We start by generalizing the setting, motivated by the idea of being able to work with more diverse data. We then prove, in this generalized setting, that not only G-convolutions are G-equivariant, but that they are the only linear maps to be so. The proof is adapted from [11] and [12].

We start from a space X of interest and suppose that we have some data defined on this space. We will only consider data that can be modeled as a function $\mathcal{X} \to \mathbb{R}^K$ (this stays very general). This space X could be $\mathbb{R}^2$ or its discretization $\mathbb{Z}^2$ if we are interested in planar images. It could be the sphere if we are interested in 360° images or if we wish to study a physical quantity at the surface of a planet. It could be $\mathbb{R}$ or its discretization $\mathbb{Z}$ if we work with time signals. It could be a fixed finite graph.

This space $\mathcal{X}$ often has a specific structure (adjacency, metric, topological, differential, etc.) which defines a group G of symmetries of this space (the automorphisms). As we have seen before with (1), this group naturally acts on $\mathcal{X} \to \mathbb{R}^K$, the representation space of the data. We would like to process the data defined on $\mathcal{X}$ in a way that gives similar results for inputs that are related by a symmetry: our model should be G-equivariant. For neural networks, this can be ensured by using only G-equivariant layers: the i-th layer Φ_i transforms equivariantly an intermediate representation of the data $f : Y_i \to \mathbb{R}^{K_i}$ into another one $\Phi_i(f) : Y_{i+1} \to \mathbb{R}^{K_{i+1}}$, where the Y_i's must be spaces on which G acts upon (with $Y_0 = \mathcal{X}$).

We want to understand what kind of constraint equivariance enforces on linear layers. To this end, we restrict the Y_i's to be either discrete spaces or spaces that embed (i.e. together with their structure) in $\mathbb{R}^n$. The Y_i's are therefore topological and measured spaces. We denote Φ the linear part of i-th layer and we let $X = Y_i$ and $Y = Y_{i+1}$. We suppose for convenience that $\mathbb{R}^{K_i} = \mathbb{R}^{K_{i+1}} = \mathbb{R}$ (the proof can be generalized quite easily to vector valued functions). In that context, we can represent $\Phi : \mathbb{R}^X \to \mathbb{R}^Y$ using the Dunford-Pettis Kernel Representation Theorem [13]. It states that any bounded linear map between the space $L^1(X) \subset \mathbb{R}^X$ of integrable functions over X to the space $L^\infty(Y) \subset \mathbb{R}^Y$ of bounded functions over Y can be represented as an integral operator. Therefore we can write:

$$\Phi(f)(y) = \int_X k(y, x) f(x) d\mu(x) \tag{5}$$

where $k : Y \times X \to \mathbb{R}$ is a kernel that characterizes Φ.

For Φ to be G-equivariant we must have:

$\forall f : X \to \mathbb{R}$ and $\forall g \in G$

$$\Phi(g \cdot f) = g \cdot \Phi(f) \tag{6}$$
$$\text{i.e.} \quad \Phi(f \circ g^{-1}) = (\Phi(f)) \circ g^{-1}$$

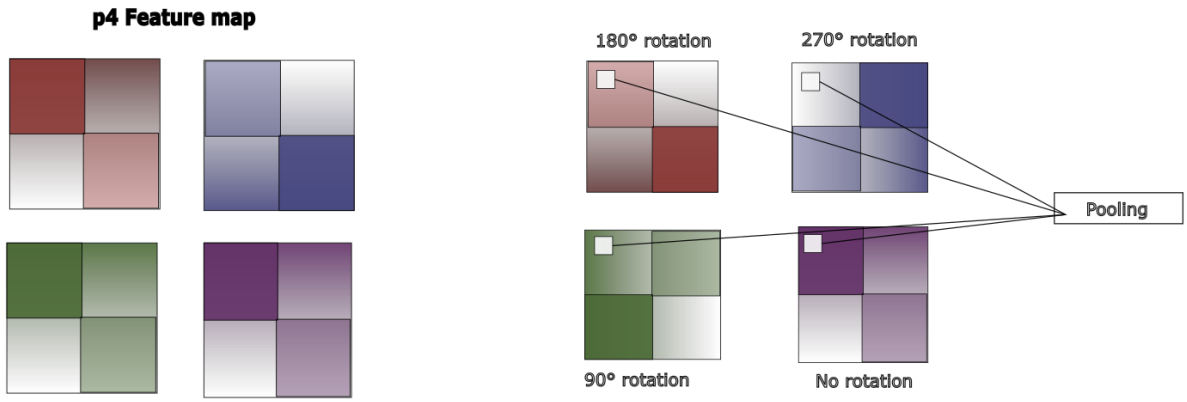

Figure 1: Pooling a p4-feature map over the 4 rotations. The p4-feature map is represented wih four 2D-arrays as a p4 element can be parametrized by (r, u, v) with $r \in \{0°, 90°, 180°, 270°\}$.

Let $y \in Y$. We focus on the right hand side:

$$\begin{aligned}\Phi(f)(g^{-1}y) &= \int_X k(g^{-1}y, x) f(x) d\mu(x) \\ &= \int_X k(g^{-1}y, g^{-1}u) f(g^{-1}u) d\mu(g^{-1}u) \\ &= \int_X k(g^{-1}y, g^{-1}u) f(g^{-1}u) \frac{1}{|\det g|} d\mu(u) \\ &= \int_X \frac{k(g^{-1}y, g^{-1}x)}{|\det g|} f(g^{-1}x) d\mu(x)\end{aligned}$$

In the sequence of equalities above, μ is the measure on X. It is the counting measure if X is discrete and the Lebesgue measure (or related to it) if $X \subset \mathbb{R}^n$. On the second line, we used a change of variable $x \mapsto g^{-1} \cdot x$. Then we used the identity $d\mu(g^{-1} \cdot x) = \frac{1}{|\det g|} d\mu(x)$ where $\det g = 1$ if μ is the counting measure and otherwise is the determinant of the linear map $x \mapsto g \cdot x$. In the case where $X \subset \mathbb{R}^n$, we indeed suppose that G acts linearly. Most of the time we can embed X in a large enough $\mathbb{R}^n$ so that the action of G becomes linear (for example if $X = \mathbb{R}^2$ and G acts by translation, one can embed X in $\mathbb{R}^3$ using the homogeneous coordinates and represent translations by specific 3×3 matrices).

On the other hand, the left hand-side is $\int_X k(y, x), f(g^{-1} \cdot x) d\mu(x)$ so for (6) to hold for all f, we must have:

$$\forall g \forall y \forall x \quad k(y, x) = \frac{k(g^{-1}y, g^{-1}x)}{|\det g|} \tag{7}$$

To obtain a stronger constraint on k, we suppose that Y is a homogeneous space for G. Informally, this means all points in Y play the same role with respect to the transformations $g \in G$. Formally, we must have $\forall y_0, y \in Y,\ \exists g_y \in G$ such that $g_y \cdot y_0 = y$. This property is equivalent to the existence of a subgroup $H \subset G$ such that the quotient space G/H is isomorphic to Y. From now on, we identify Y with G/H and identify $H \in G/H$ with some fixed $y_0 \in Y$.

Let $x \in X, y \in Y$ and let $g_y \in G$ such that $g_y \cdot y_0 = y$. We have:

$$\begin{aligned}k(y, x) &= k(g_y \cdot y_0, x) \\ &= k(g_y^{-1} \cdot g_y \cdot y_0,\ g_y^{-1} \cdot x) \ / \ |\det g_y| \\ &= k(y_0,\ g_y^{-1} \cdot x) \ / \ |\det g_y|\end{aligned}$$

where we used (7) with $g = g_y$. Since y_0 is fixed, we can therefore redefine the kernel as a one argument function $k(y, x) = k(g_y^{-1}x)/|\det g_y|$. The choice of g_y is in general not unique and should not matter. Since we identified Y with G/H, we have $y = g_y \cdot y_0 \iff g_y \in y$ where we interpret y as a coset in G/H. Therefore, the single argument kernel must satisfy the following symmetry constraint:

$$\forall x \in X,\ \forall y \in G/H,\ \forall g, \tilde{g} \in y,\ \frac{k(g^{-1}x)}{|\det g|} = \frac{k(\tilde{g}^{-1}x)}{|\det \tilde{g}|}.$$

Using (7), this condition can be simplified to:

$$\forall x \in X,\ \forall h \in H,\ k(x) = \frac{k(h^{-1}x)}{|\det h|} \tag{8}$$

.

Substituting in (5) the new kernel form, we obtain that in order for a linear layer Φ to be G-equivariant, it must write:

$$\Phi(f)(y) = \int_X \frac{1}{|\det g_y|} k(g_y^{-1}x) f(x) d\mu(x) \tag{9}$$

where the choice of g_y do not matter i.e. k must satisfy (8). We recognize in the integral a generalized group convolution. For example, if we set $X = Y = G$ and consider a discrete group G, we will find the G-correlation (= G-convolution with inverse filter) as defined in (4) (for $K = 1$).

If the input f and the kernel k are compactly supported (i.e. finitely supported if X is discrete), then the output is too. This ensures in particular that the ouput $\Phi(f) \in L^1(Y)$ (satisfying the Dunford-Pettis theorem's requirement).

This new definition is valid for discrete and continuous groups. It maps X-feature maps to Y-feature maps where Y is homogeneous for G i.e. Y can be identified with a quotient space of G. Therefore, in a network, all Y_i's except $Y_0 = \mathcal{X}$ should be quotient spaces. The kernel symmetry condition can seem quite limiting but it disappears if Y can be identified with the trivial quotient G. A good choice to make the symmetry constraint disappear is therefore to choose $Y_i = G$.

When $X = G$ and Y is a strict quotient space, we see that Φ act as a linear G-equivariant pooling + subsampling layer (for a constant kernel it would be mean-pooling).

3 IMPLEMENTATION

The first step when computing a G-CNN, is too apply the G-transformations on the filters. Before doing this, we can notice the split property that $\mathbb{Z}^2$, p4 and p4m all have. This means that we can always break down an element g in G as the composition of a translation and a rotation about the origin and a reflexion. If we take for instance the group p4m, g can be written as

$$g = tsr$$

with

$$t = \begin{bmatrix} 1 & 0 & u \\ 0 & 1 & v \\ 0 & 0 & 1 \end{bmatrix}$$

,

$$s = \begin{bmatrix} \cos(r\pi/2) & -\sin(r\pi/2) & 0 \\ \sin(r\pi/2) & \cos(r\pi/2) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

and

$$r = \begin{bmatrix} (-1)^m & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

t represents here a translation, s a rotation and r a reflexion. As $L_{tsr} = L_t \circ L_s \circ L_r$ We can then rewrite the G-correlation:

$$f \star \psi(g) = \sum_{h \in X} \sum_k f_k(h) L_t [Ls[L_r(\psi(h))]]$$

This means in the architecture that the G-transformation can be separated in three filters for p4m or two for the group p4. The first filter implements the rotation and the second one is for the transformation and the last one is a reflexion. This is represented on figure 2 with the group p4 (no reflexions). K^l is the number of channels at layer l, S^l the number of translations in G that leave the origin invariant (1 in $\mathbb{Z}^2$, 4 in p4 and 8 in p4m) and n is the spatial extent of the filter. The number of filters at layer l after the G-transformations is $K^l \times K^{l-1} \times S^{l-1} \times S^l \times n \times n$.

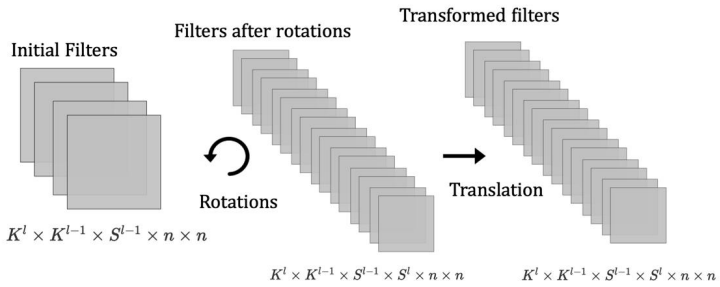

Figure 2: Construction of the augmented set of filters necessary to compute a p4-convolution using a regular correlation routine.

Once those filters are calculated, the network can be seen as a usual convolutional neural network with Relu activation functions, using dropout and Batch Normalization for a better weight sharing. The pooling needs to respect the rule described on the Figure 1. An example of a G-CNN architecture can be observed on Figure 3.

The architecture that obtained the best result was a ResNet convolutional network with 44 layers, including 14 convolutional layers and $k_i = 32, 64, 128$. The number of parameters in the p4m-CNN was adjusted, divided by $\sqrt{8} \approx 3$, to obtain the same number of parameters than the traditional the CNN. The results obtained on the CIFAR Dataset were 33% better for the p4m-CNN (6.46 vs 9.45 error rate) and around 15% better on the CIFAR+ Dataset (4.94 vs 5.61 error rate).

4 LIMITATIONS AND EXTENSIONS

This paper has shown that taking into account more symmetries of the data such as reflections and rotations symmetries lead to better results for image recognition tasks. The idea comes from the fact that an object should be processed by the model in a way that does not depend on its position and orientation in space. However, the p4 and p4m discrete groups considered in this paper are not continuous: they only contain 4 different rotations. This limitation comes from the fact that we tend to use images whose pixels are arranged on a square grid modeled as $\mathbb{Z}^2$. This restricts us to working with groups of symmetries of $\mathbb{Z}^2$. The biggest of these symmetry groups (for the metric structure of $\mathbb{Z}^2$ inherited

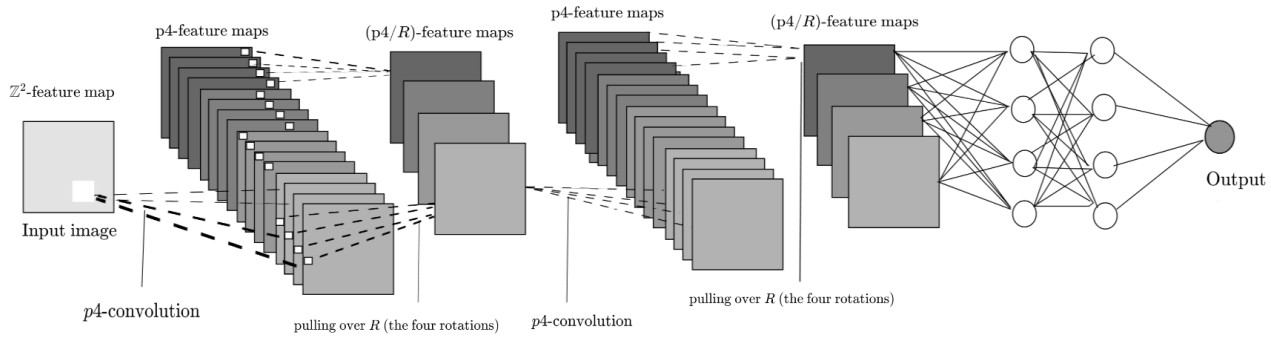

Figure 3: Example of a G-CNN architecture (the group G is p4).

from $\mathbb{R}^2$) is indeed p4m. This group is far from being as rich as the $E(2)$ group of rigid motions of the plane. Welling and Cohen thought about using the hexagonal lattice (the one generated by the third roots of unity), which is different than $\mathbb{Z}^2$ and allows for 6 rotations instead of 4. However very few data today is defined on such a lattice. Nonetheless, We can imagine interpolating square grid images in order to produce images defined on an hexagonal lattice. This would indeed allow for the use of p6m-equivariant network (p6m being the biggest group of isometries of the hexagonal lattice).

As the preliminary work by Welling and Cohen in 2014 was done on compact commutative Lie Groups, an idea could be to use a G-CNN related method that focuses on a continuous subgroup of Lie type (for instance the group of rotations in 2 dimensions) to apply those results.

A even larger problem would be to work with non-commutative Lie Groups such as the group $E(3)$ of rigid motions in 3 dimensions, or its subgroup $SO(3)$ that contains only rotations.

There is a important work to do to be able to "discretize" the behavior of these groups in order to make their processing possible in the discrete world of computation.

We can also argue that requiring full rigorous G-equivariance may be too hard of a constraint on the architecture of the network. As stated in the section on pooling, regular CNNs achieve very good results even while using max-pooling layers which are not totally translation equivariant, which we understand by thinking about the continuity of the pixels values.

REFERENCES

[1] Taco S. Cohen, Max Welling. Group Equivariant Convolutional Networks. *Proceedings of The 33rd International Conference on Machine Learning*, PMLR 48:2990-2999, 2016.

[2] Bruna, J. and Mallat, S. Invariant scattering convolu- tion networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 35(8):1872–86, aug 2013.

[3] Cohen, T. and Welling, Learning the Irreducible Representations of Commutative Lie Groups, in *Proceedings of the 31st International Conference on Machine Learn- ing (ICML)*, volume 31, pp. 1755–1763, 2014.

[4] Dieleman, S., Willett, K. W., and Dambre, J. Rotation-invariant convolutional neural networks for galaxy morphology prediction. *Monthly Notices of the Royal Astronomical Society*, 450(2), 2015.

[5] Gens, R. and Domingos, P. Deep Symmetry Networks. In *Advances in Neural Information Processing Systems (NIPS)*, 2014

[6] Jaderberg, M., Simonyan, K., Zisserman, A., and Kavukcuoglu, K. Spatial Transformer Networks. In *Advances in Neural Information Processing Systems 28 (NIPS 2015)*, 2015.

[7] Lowe, D.G. Distinctive Image Features from Scale Invariant Keypoints. *International Journal of Computer Vision*, 60(2):91–110, nov 2004.

[8] Ma, P.W., Chan, T.H.H., 2023. A feedforward unitary equivariant neural network. Neural Networks 161, 154–164. doi:https://doi.org/10.1016/j.neunet.2023.01.042.

[9] Sifre, Laurent and Mallat, Stephane. Rotation, Scaling and Deformation Invariant Scattering for Texture Discrimination. *IEEE conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.

[10] Carlos Esteves. Rotation, Theoretical Aspects of Group Equivariant Neural Networks. *arXiv*, https://arxiv.org/abs/2004.05154, 2020.

[11] Taco S. Cohen, Mario Geiger, Maurice Weiler. A general theory of equivariant CNNs on homogeneous spaces. *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, 2019.

[12] E. Bekkers. B-Spline CNNs on Lie Groups. *International Conference on Learning Representations*, 2019.

[13] Wolfgang Arendt, Alexander V. Bukhvalov. Integral representations of resolvents and semigroups. *Forum mathematicum*, 1994.