On the Benefits of SO(3)-Equivariant Neural Networks for Spherical Image Processing

Simon Martin, Pierre-Yves Lagrave

On the Benefits of $\mathrm{SO}(3)$-Equivariant Neural Networks for Spherical Image Processing

Simon Martin[1,2,*], **Pierre-Yves Lagrave**[2]
[1]Ecole Normale Supérieure
[2]Thales Research and Technology
simon.martin@ens.fr, pierre-yves.lagrave@thalesgroup.com

Abstract

This paper deals with the use of $\mathrm{SO}(3)$-Equivariant Neural Networks for processing spherical images such as those acquired with bi-directional fish-eye lenses. By using Generative Adversarial Networks trained on the Planesnet dataset to generate augmented training sets, we first confirm recent results according to which the use of $\mathrm{SO}(3)$-equivariance mechanisms is more efficient than training data augmentation techniques for processing data with a native spherical geometry. We then explain how the action of $\mathrm{SO}(3)$ on the 2d-sphere can be projected on tangent planes, leading to an equivalent reformulation of the $\mathrm{SO}(3)$-convolution operator in euclidean spaces and allowing to achieve a wider range of robustness properties through the design of appropriate local pooling mechanisms. Finally, we articulate our projected $\mathrm{SO}(3)$-convolution with recent works on fish-eye image processing in which $\mathrm{SU}(1,1)$ and $\mathrm{SL}(2,\mathbb{R})$ Equivariant Neural Networks are coupled with hyperbolic projection mechanisms, drawing the path toward a consolidated approach for processing inputs represented as signals on homogeneous spaces.

1 Introduction

In the following, we motivate our work by reminding ourselves of the generic lack of robustness of Deep Learning algorithms, by then discussing the state-of-the-art regarding the use of data augmentation and Equivariant Neural Networks as a remediation, and by finally stating our contributions.

1.1 Data Augmentation and Equivariance

Conventional Deep Learning algorithms only encode limited priors about robustness to perturbations into their design. Taking the example of computer vision tasks, Convolutional Neural Networks (CNN) enforce local robustness with respect to translations through equivariance, weights sharing and pooling mechanisms but have been shown to lack of robustness with respect to other types of relevant transforms such as rotations, scaling, lightening effects, small noise, etc.

To remedy this issue, a practical approach referred to as data augmentation consists in increasing the size of the training set by applying well chosen transformations to the original training set, typically on-the-fly during the gradient descent routine [Shorten and Khoshgoftaar, 2019]. Data augmentation has been shown to be empirically successful and is widely used for practical applications. Several approaches such as [Cubuk *et al.*, 2019], subsequently improved in [Lim *et al.*, 2019] can be used to derive efficient data augmentation strategies for a given purpose by optimizing over a possible set of transformations sequences.

Although widely applied by practitioners because of its simplicity and implementation convenience, data augmentation is not fully satisfactory. In particular, learning invariances directly from the data consumes significant algorithmic capacity and therefore requires models with a large number of trainable parameters. Depending on the targeted applications, this process may therefore not be aligned with operational constraints such as memory footprint limitation or inference timing performance. Also, the training time is impacted by data augmentation as more epochs will typically be required to reach convergence. Another caveat with this type of approach is that, although recent attempts have been made with respect to the formalization of the method through group theory [Dao *et al.*, 2019; Lyle *et al.*, 2020; Chen *et al.*, 2020], we are still generally lacking of theoretical guarantees with respect to the behavior the algorithms trained with augmented data. Data augmentation has also been shown sub-optimal in term of sample complexity [Elesedy and Zaidi, 2021], a more efficient approach consisting in using architectures where group-based invariance/equivariance is natively enforced. These results were recently experimentally confirmed in [Gerken *et al.*, 2022] for spherical image processing.

Equivariant Neural Networks (ENN) [Cohen and Welling, 2016a; Gerken *et al.*, 2021], which belong to the field of Geometric Deep Learning [Bronstein *et al.*, 2021] are becoming more and more popular thanks to their conceptual soundness and to their ability to reach state-of-the-art accuracies for a wide range of applications [Bekkers *et al.*, 2018; Cohen *et al.*, 2018b; Finzi *et al.*, 2020]. In particular, the underlying equivariant and/or invariant layers of ENN allow building architectures robust to generic geometrical transforms, therefore making the use of ENN a reasonable alter-

*The first author contributed to this work during an internship at Thales Research and Technology France in 2022.

native to data augmentation techniques.

1.2 Previous Work and Contributions

ENN have already been proposed for processing spherical data, starting with the work [Cohen *et al.*, 2018a] where lifting mechanisms and $\mathrm{SO}(3)$-convolution operators are introduced to enforce equivariance properties in neural networks, together with efficient numerical schemes relying on spherical harmonics decomposition and Fast Fourier Transform. Steerable Neural Networks [Cohen and Welling, 2016b], which achieve equivariance through the choice of constrained kernel functions, have also been introduced to achieve $\mathrm{SO}(3)$ equivariance by leveraging on Clebsch-Gordan decomposition [Lang and Weiler, 2020]. More recently, the relevance of using other equivariance groups than $\mathrm{SO}(3)$ for spherical data processing has been highlighted by leveraging on adequate projection techniques, including the use of the Lie groups $\mathrm{SL}(2,\mathbb{C})$ [Mitchel *et al.*, 2022] or $\mathrm{SU}(1,1)$ and $\mathrm{SL}(2,\mathbb{R})$ [Lagrave and Barbaresco, 2022].

Comparing the benefits of using equivariance mechanisms with that of applying widely used data augmentation techniques has been recently considered from both theoretical [Elesedy and Zaidi, 2021; Wang *et al.*, 2022] and practical standpoints. Although still limited, the available results show that ENN are superior to data augmentation techniques for the considered set-ups and datasets.

The contribution of this paper is twofold:

- After giving some reminders with respect to group-based convolution operators and their equivariance properties, we first confirm the superiority of using ENN when compared to data augmentation techniques for the classification of spherical images that we built using the Planesnet dataset[1], a set of colored 20x20 aerial images with and without planes. We also emphasize that, compared to the existing work [Gerken *et al.*, 2022], we have investigated here a data augmentation with respect to the full data distribution $\mathbb{P} = \mathbb{P}_d \otimes \mathbb{P}_\theta$ by leveraging on Generative Adversarial Networks (GAN), while only augmentation with respect to the orientation of the images $\mathbb{P}_\theta$ was considered in [Gerken *et al.*, 2022].
- Anchoring in [Mitchel *et al.*, 2022] and [Lagrave and Barbaresco, 2022] which exploit projection mechanisms to introduce equivariance mechanisms to process spherical signals, we show that the action of $\mathrm{SO}(3)$ can be projected onto tangent planes. This leads to an equivalent reformulation of the $\mathrm{SO}(3)$-convolution operator in such spaces, which in particular allows to achieve a wider range of robustness properties through the design of appropriate local pooling mechanisms. We finally articulate this approach with those leveraging on $\mathrm{SL}(2,\mathbb{C})$, $\mathrm{SU}(1,1)$ and $\mathrm{SL}(2,\mathbb{R})$ equivariance properties, paving the way toward a unified approach of coupling projections with adequate groups actions when working with signals defined on homogeneous spaces.

[1] https://github.com/rhammell/planesnet

2 Geometric Deep Learning and Equivariant Neural Networks

In this section, we introduce the mathematical framework to generalize the equivariance property of the standard CNNs. To do so, we present a more general convolution operator defined on abstract groups (seen as a set of geometric transforms) and homogeneous spaces (accounting for the structure of the data). We then introduce the main implementation challenges for such operators.

2.1 Generalized Convolution

In computer vision, standard CNNs achieve translational equivariance by computing a convolution between the input image and a kernel containing the network's parameters. The kernel slides over the image and an accumulation is performed on the image pixels :

$$[K \star f](y) = \sum_{x \in \mathbb{Z}^2} K(y-x) f(x) \tag{1}$$

where $f : \mathbb{Z}^2 \to \mathbb{R}^d$ is the image and $K : \mathbb{Z}^2 \to \mathbb{R}^{d \times d'}$ is the kernel. The integers d, d' correspond to the number of input and output channels.
This operation is designed to be linear with respect to the input image and translational equivariant :

$$\mathcal{L}_u[K \star f] = [K \star \mathcal{L}_u f] \tag{2}$$

where $\mathcal{L}_u$ is the translational operator for $u \in \mathbb{Z}^2$, defined as $\mathcal{L}_u(f)(x) = f(x-u)$.

Equivariance can naturally be generalized to other geometric transforms thanks to equation (2). Let G be a group (whose elements should be seen as geometric transforms). We define the convolution between the function $f \in L^2(G, \mathbb{R}^d)$ and the filter $K \in L^2(G, \mathbb{R}^{d \times d'})$ as :

$$[K \star f](g) = \int_G K(h^{-1}g) f(h) \, \mathrm{d}\mu(h) \tag{3}$$

where μ is the Haar measure on G. As for the case of translations, this operation satisfies an equivariance property, with respect to the action of G:

$$\mathcal{L}_g[K \star f] = [K \star \mathcal{L}_g f] \tag{4}$$

with $\mathcal{L}_g(f)(h) = f(g^{-1}h)$ for $g, h \in G$. We mention that this convolution operation is more general that it may appear : as discussed in [Cohen *et al.*, 2019], any linear and equivariant mapping $L^2(G, \mathbb{R}^d) \to L^2(G, \mathbb{R}^{d'})$ is in fact a convolution for some kernel $K : G \to \mathbb{R}^{d \times d'}$.

However, this framework is somewhat restrictive : it requires feature maps and kernels to be defined on a group, and therefore the symmetries associated with the convolution operation are inherited from the data structure. As for the case of the sphere and the rotational symmetry group $\mathrm{SO}(3)$, we would rather need a geometrical space on which images and kernels are defined and a group of geometric transforms acting on this space. This is the context of homogeneous spaces. A topological space X is said to be

homogeneous with respect to a group G (or G-homogeneous) if G acts transitively on X, that if there exists an action $(g, x) \in G \times X \mapsto g \cdot x \in X$ such that for any $x, y \in X$, there exists a transform $g \in G$ such that $g \cdot x = y$. Under those assumptions, the space X is isomorphic to a quotient G/H for some H subgroup of G.

The standard example of homogeneous space is the sphere $S^{n-1} = \{x \in \mathbb{R}^n, \|x\| = 1\}$, homogeneous with respect to both the orthogonal group $O(n)$ and the special orthogonal group $SO(n)$. This general framework allows for more freedom in the choice of the group G which is essential as it defines the symmetries of the neural network.

In these conditions, we now consider a feature map $f : X \to \mathbb{R}^d$ and a kernel $K : X \to \mathbb{R}^{d \times d'}$. There are different ways to define homogeneous spaces convolutions : a first possibility, introduced by [Cohen *et al.*, 2018a], consists in using a convolution to lift the feature map from X to G and then use the group convolution (3). In this framework, we would have therefore a first layer with the lifting operator (5) and the group convolution (3) in all the subsequent layers.

$$[K \star f](g) = \int_X K(g^{-1} \cdot x) f(x) \, \mathrm{d}x \tag{5}$$

Another convolution, defined by [Kondor and Trivedi, 2018], allows to keep feature maps on the space X itself, but requires to introduce some reference point $x_0 \in X$:

$$[K \star f](x) = \int_G K(g^{-1} \cdot x) f(g \cdot x_0) \, \mathrm{d}\mu(g) \tag{6}$$

However, thanks to the G-homogeneity of X, the choice of x_0 is irrelevant in the convolution : all points of X are equivalents.

Homogeneous space convolution detaches the symmetry group from the space, and therefore for a given space (as for the case of the sphere) there are often many groups satisfying the homogeneity condition. Although selecting bigger groups may seem tempting, this would come with a computational cost overhead and may also lead to a decreasing expressivity of the neural network.

2.2 Numerical Considerations

Although the convolution operations previously defined are perfectly equivariant, numerical constraints require to discretize the space as well as the group, therefore the numerical equivariance is only approximate. One of the implementation challenges is to reduce to the maximum the equivariance error.

However, in some cases, perfect equivariance can be achieved, for example on the standard translational convolution of 2D CNNs. In the same way, any convolution on a finite group would reach perfect equivariance, provided that its computation is done by exhausting all the group elements.

For an image $f : X \to \mathbb{R}^d$, a kernel $K : X \to \mathbb{R}^{d \times d'}$ and a geometric transform $g \in G$, we define the equivariance error of the convolution $K \star f$:

$$\Delta_{g,K}(f) = \frac{\mathrm{Std}\left(K \star \mathcal{L}_g(f) - \mathcal{L}_g(K \star f)\right)}{\mathrm{Std}\left(K \star f\right)} \tag{7}$$

where the standard deviation is taken over the pixels of the image. We therefore aim at minimizing the general error associated with the operator : $\Delta = \langle \Delta_{g,K}(f) \rangle_{g,K,f}$ averaged over the filters, images and transforms.

Regarding the numerical implementation, a first method to implement such a convolution is to use a numerical integration scheme to compute the integral over the space or the group. Depending on the data structure and the size of the group, the use of a Monte Carlo method as introduced in [Finzi *et al.*, 2020; Lagrave and Barbaresco, 2021] could be of interest. In the case of the sphere S^2 and symmetry group $\mathrm{SO}(3)$, a simple median point method gives a convergence rate proportional to $p^{-2/3}$, p being is the total number of integration points, which is better than a Monte Carlo method. These methods require significant sampling to converge and can lead in practice to substantial equivariance error when timing performance constraints are considered.

The alternative is to use Fourier theory on the group G. This method has been extensively used in spherical convolutions [Gerken *et al.*, 2021; Lang and Weiler, 2020] and has proven to reach very low equivariance errors while being quite effective in calculation duration. For instance, [Cohen *et al.*, 2018a] reach an approximation error around 10^{-7} with their equivariant neural network on the sphere (using the lift method in equation (5)). The main constraint of this type of implementation is that it requires a well established Fourier theory on the group of interest. Although this is the case for standard transforms as translations and rotations, with corresponding efficient numerical schemes (e.g., FFT), some generalized version of Fourier theory needs to be considered when dealing with more exotic structures (e.g., non compact groups) as it was done in [Lagrave *et al.*, 2021] with the use of the Helgason-Fourier transform for computing $\mathrm{SU}(1,1)$-convolutions between signals defined on the Poincaré disk $\mathbb{D}$.

3 $\mathrm{SO}(3)$ Equivariance for Images

We now deal with the particular case of spherical images and equivariance with respect to the group $\mathrm{SO}(3)$ of 3D rotations, which is of high interest thanks to a wide range of applications (Fisheye images, autonomous cars, data at the surface of the Earth, ...).

We first prove the equivariance interest with a simple data augmentation numerical experiment, then explore the possibility to project spherical features on the plane to directly perform equivariant convolution in 2D. Finally, we go through several applications of equivariance methods in image classification.

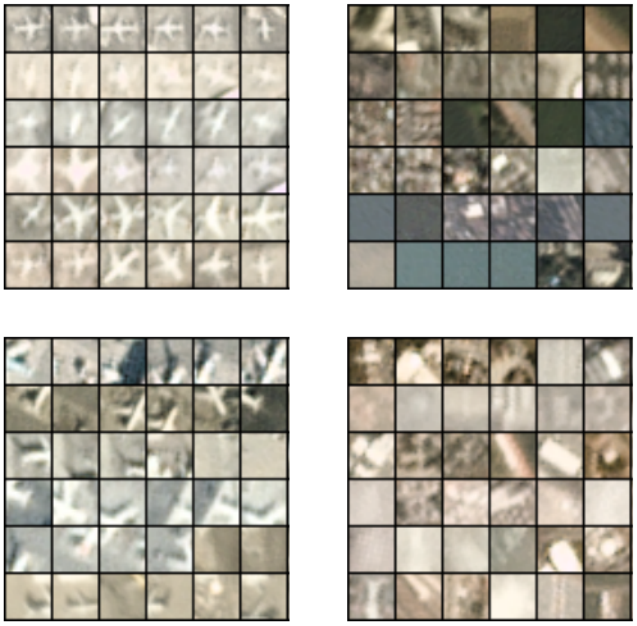

Figure 1: Examples of images inside the Planesnet dataset. Top left : images containing planes. Top right : images with no planes. Bottom left : images containing portions of planes. Bottom right : images with objects looking like planes. Images of the last three categories are labeled with no plane.

3.1 Confirming the Equivariance Benefits

We confirm the theoretical [Elesedy and Zaidi, 2021; Wang *et al.*, 2022] and previous numerical results [Gerken *et al.*, 2022] about equivariant neural networks versus data augmentation methods. To do so, we compare the learning performance of a standard 2D CNN and a $\mathrm{SO}(3)$ equivariant CNN on the sphere.

The dataset we used for this comparison is the Planesnet dataset (https://github.com/rhammell/planesnet). It is composed of RGB satellite images of size 20×20 with a label accounting for the presence of a plane on the image. Although the CNN task amounts to a simple binary classification, the dataset contains misleading images such as ones containing only portion of planes or objects with sharp edges looking like planes. All of these are labeled with no plane. Examples of such images are shown in figure 1.

Before any learning, the images are first projected on the sphere by using the representation in spherical coordinates, offering the possibility to apply random rotations during the projection. This process is shown in figure (2) for an expressive MNIST sample. Mapping the randomly rotated image in spherical coordinates causes non-linear distorsion. The fact that this distorsion is only local and depends on the position of the features on the sphere makes it very difficult for standard CNNs to reach reasonable performance on this task. We used three neural networks for our comparison, one spherical equivariant CNN (denoted S2CNN in the following) designed by [Cohen *et al.*, 2018a] with 2 convolutional layers, and two others standard CNNs, one with 2 layers (classic CNN) and the second with 3 convolutional layers (deep CNN). All three networks have roughly 160k parameters.

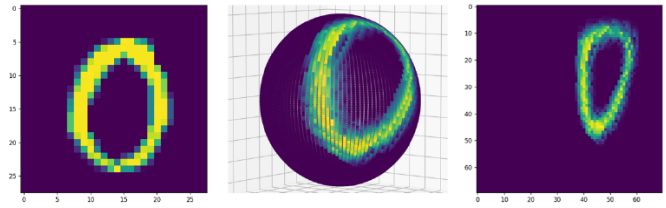

Figure 2: Illustration of the projection algorithm on the sphere. Left : original image in (x, y) coordinates. Middle : image projected on the sphere with random rotation. Right : visualization of the projection in spherical coordinates (ϕ, θ).

Figure 3: Comparison between "plane" samples of the Planesnet dataset (left) and outputs of a GAN trained on samples of the "plane" class of the same dataset (right).

There are different ways to perform data augmentation on such neural networks. It is possible to enrich the training set by creating rotated copies of the original images as it was done in [Elesedy and Zaidi, 2021]. Denoting $\mathbb{P}_d$ the distribution of the original (spherical) dataset, and $\mathbb{P}_\theta$ that of random 3d-rotations by $\mathrm{SO}(3)$ elements, we observe that the augmentation is only performed with respect to $\mathbb{P}_\theta$ in their work.

In order to perform the data augmentation with respect to the full data distribution $\mathbb{P}$=$\mathbb{P}_d \otimes \mathbb{P}_\theta$, we propose using here a generative method such as a GAN architecture to create more examples of images in the training set. This is the method we have chosen for data augmentation and figure 3 compares samples obtained with the GAN used to generate the class "plane" (another one was trained to cover the "no-plane" case) with samples of the original Planesnet dataset. We have then trained our three networks with 50 epochs on data generated by the GANs for consistency purpose. We measured their test accuracy for different sizes of training sets and the results are presented in figure 4.

Before training, each image of the training set and the testing set was projected and randomly rotated on the sphere. We then used the same rotated dataset for each of the three neural networks. Each data point on figure 4 is an average over 5 simulations.

The first conclusion is that the equivariant neural network gets largely better performances than the 2D CNNs, no

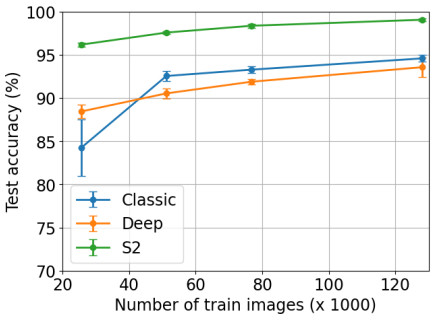

Figure 4: Test accuracies for 2D CNNs (Classic and Deep) and S2 CNN as a function of the size of the training set.

matter the augmentation factor. With a sufficient amount of data, the S2CNN almost reaches a perfect generalization accuracy on the testset. On the side of the standard CNNs, distorsion effects due to rotations are hardly handled, and even though test accuracies are low for small datasets, improvements with data augmentation tend to be modest. As a striking result, test accuracies of 2D CNNs at the maximum of data augmentation do not even reach the performance of the S2CNN without any augmentation.

This is the benefit of equivariance : implementing the symmetries directly inside the convolution operators leads to major savings in dataset sizes and therefore in learning time and allows for more compact architectures with less parameters.

3.2 $\mathrm{SO}(3)$ Action on Planes

We now explain how the action of $\mathrm{SO}(3)$ on the sphere can be projected onto tangent planes as to derive an equivalent formulation of the $\mathrm{SO}(3)$-convolution in those Euclidean spaces, hence providing a way of reaching a wider range of robustness properties through the choice of adequate pooling mechanisms within the Neural Networks architectures.

Feature maps on the sphere can be represented as planar images via some projection $\Pi : S^2 \to \mathbb{R}^2$. For the moment, we do not specify any particular projection, but most common are stereographic and logarithm projections. Due to the topology of the sphere, those projections have a singularity at a single point, which is no problem as we eventually discretize the space during implementation.

Starting from the homogeneous space convolution of equation (6), we assume the reference point $x_0 \in S^2$ to be the north pole and that the projection is done via this point, that is $\Pi(x_0) = 0$ the origin of $\mathbb{R}^2$. We denote B_Π the image of S^2 by the projection. Given a feature map $f : S^2 \to \mathbb{R}^d$, we define its projection f^Π on B_Π : $f^\Pi = f \circ \Pi^{-1}$. Taking a template kernel $K : \mathbb{R}^2 \to \mathbb{R}^{d \times d'}$, we define the projected convolution, for $v \in B_\Pi$:

$$[K \star f^\Pi](v) = \int_{SO(3)} K\left(\Pi \circ g^{-1} \circ \Pi^{-1}(v)\right) f^\Pi\big(\Pi(g \cdot x_0)\big)\, \mathrm{d}\mu(g) \tag{8}$$

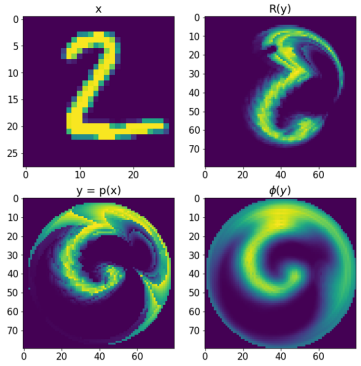

Figure 5: Illustration of the projection process with the choice of logarithm projection as stated in equation (11). Top left : original MNIST image. Bottom left : logarithm projection with random rotation. Top right : SO(3) rotation visualized via logarithm projection. Bottom right : convolution (10) of the projected image.

This encourages to define a projected version of $\mathrm{SO}(3)$, that is $\Pi(g) \equiv \Pi \circ g \circ \Pi^{-1} : B_\Pi \to B_\Pi$. $\mathrm{SO}(3)$ can now be seen as acting transitively on B_Π. What's more is that the application $g \mapsto \Pi(g)$ is a group morphism, and the convolution writes :

$$[K \star f^\Pi](v) = \int_{SO(3)} K\big(\Pi(g)^{-1}(v)\big) f^\Pi\big(\Pi(g)(0)\big)\, \mathrm{d}\mu(g) \tag{9}$$

And with the change of variable $g \mapsto \Pi(g)$, we get :

$$[K \star f](v) = \int_{\Pi(SO(3))} K\big(\omega^{-1}(v)\big) f^\Pi\big(w(0)\big) |J(w)| \mathrm{d}\omega \tag{10}$$

where the jacobian $J(\omega)$ depends on the chosen projection Π.

Applications of this framework rely on the choice of the projection. When Π is the stereographic projection, B_Π corresponds to the full plane $\mathbb{R}^2$ and in order to keep a bounded support when projecting the image, we shall restrict features on the northern hemisphere, for instance.

A direct way of obtaining a bounded support for the projected images is to use the logarithm projection which originates from differential geometry. On the sphere, it simply maps spherical coordinates to 2D cylindrical coordinates :

$$\Pi_L : \begin{pmatrix} \cos(\phi)\sin(\theta) \\ \sin(\phi)\sin(\theta) \\ \cos(\theta) \end{pmatrix} \in S^2 \mapsto \begin{pmatrix} \theta\cos(\phi) \\ \theta\sin(\phi) \end{pmatrix} \in \mathbb{R}^2 \tag{11}$$

for $\phi \in [0, 2\pi]$ and $\theta \in [0, \pi]$. As for the stereographic projection, it is ill defined at the south pole s, however it is a diffeomorphism from $S^2 \backslash \{s\}$ onto the open ball of radius π, $\{x \in \mathbb{R}^2, \|x\| < \pi\}$. See figure 5 for a visualization of rotations and convolution through this projection.

The convolution operations in equation (6) and (10) are perfectly equivalent. The real benefit of the projection method is when pooling after performing the convolution. Depending on the chosen projection, we can expect very

Figure 6: Left : Fisheye lens. Photo: Wikipedia. Right : First high-resolution color image sent by Hazard Cameras (Hazcams) of NASA's Perseverance Mars vehicle after its landing on February 18, 2021. Distortion effects are in particular visible through the curved horizon line. Photo: NASA / JPL-Caltech.

different results as features and pixels on images are shuffled during projection. Theoretical and experimental results should however support those hypothesis.

3.3 Application to Images Classification

The main application of $\mathrm{SO}(3)$ Equivariant Neural Networks is the treatment of fisheye pictures which originate from spherical shaped lenses (see figure 6). Due to the curvature of the lens, the images have a native spherical shape, which results in some distorsion when projected on the 2D plane. Such images can be treated by an equivariant CNN after being projected back on the sphere, using the method describe in section 3.1. Equivalently, using the adapted projection (that is knowing the native projection mechanism of the lens), we can use the framework introduced in section 3.2 and directly compute convolutions on the plane. This allows to avoid a pre-processing of the images before training, although it restrains the convolution algorithm to a particular projection. Note that the knowledge of the lens projection mechanism is also required to project back images on the sphere.

4 Conclusions and Further Work

By leveraging on GANs trained on the Planesnet dataset, we have first confirmed in this paper the benefits of using equivariant architectures compared to data augmentation techniques in the context of spherical image classification. We have then shown that the action of $\mathrm{SO}(3)$ on the 2d-sphere could be projected onto tangent planes and have highlighted some similarities with recent works coupling projection mechanisms with equivariance to other groups than $\mathrm{SO}(3)$ to handle spherical images.

From a theoretical standpoint, further work will include writing down a unified framework for coupling projection techniques with equivariance to appropriate groups actions when working with signals defined on homogeneous spaces. From a more practical standpoint, the scalability of these approaches will be studied by working with bigger images and larger datasets, and by investigating potential improvements to the numerical schemes used to compute group-based convolution operators.

References

[Bekkers *et al.*, 2018] Erik J Bekkers, Maxime W Lafarge, Mitko Veta, Koen AJ Eppenhof, Josien PW Pluim, and Remco Duits. Roto-translation covariant convolutional networks for medical image analysis, 2018.

[Bronstein *et al.*, 2021] Michael M. Bronstein, Joan Bruna, Taco Cohen, and Petar Veličković. Geometric deep learning: Grids, groups, graphs, geodesics, and gauges, 2021.

[Chen *et al.*, 2020] Shuxiao Chen, Edgar Dobriban, and Jane H. Lee. A group-theoretic framework for data augmentation. *Journal of Machine Learning Research*, 21(245):1–71, 2020.

[Cohen and Welling, 2016a] Taco Cohen and Max Welling. Group equivariant convolutional networks. In Maria Florina Balcan and Kilian Q. Weinberger, editors, *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 2990–2999, New York, New York, USA, 20–22 Jun 2016. PMLR.

[Cohen and Welling, 2016b] Taco S. Cohen and Max Welling. Steerable cnns. 2016.

[Cohen *et al.*, 2018a] Taco S. Cohen, Mario Geiger, Jonas Koehler, and Max Welling. Spherical cnns, 2018.

[Cohen *et al.*, 2018b] Taco S. Cohen, Mario Geiger, Jonas Köhler, and Max Welling. Spherical cnns. *CoRR*, abs/1801.10130, 2018.

[Cohen *et al.*, 2019] Taco S Cohen, Mario Geiger, and Maurice Weiler. A general theory of equivariant cnns on homogeneous spaces. In H. Wallach, H. Larochelle, A. Beygelzimer, F. Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32, pages 9145–9156. Curran Associates, Inc., 2019.

[Cubuk *et al.*, 2019] Ekin Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc Le. Autoaugment: Learning augmentation strategies from data. pages 113–123, 06 2019.

[Dao *et al.*, 2019] Tri Dao, Albert Gu, Alexander Ratner, Virginia Smith, Chris De Sa, and Christopher Re. A kernel theory of modern data augmentation. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 1528–1537. PMLR, 09–15 Jun 2019.

[Elesedy and Zaidi, 2021] Bryn Elesedy and Sheheryar Zaidi. Provably strict generalisation benefit for equivariant models, 2021.

[Finzi *et al.*, 2020] Marc Finzi, Samuel Stanton, Pavel Izmailov, and Andrew Gordon Wilson. Generalizing convolutional neural networks for equivariance to lie groups on arbitrary continuous data, 2020.

[Gerken *et al.*, 2021] Jan E. Gerken, Jimmy Aronsson, Oscar Carlsson, Hampus Linander, Fredrik Ohlsson, Christoffer Petersson, and Daniel Persson. Geometric deep learning and equivariant neural networks, 2021.

[Gerken *et al.*, 2022] Jan E. Gerken, Oscar Carlsson, Hampus Linander, Fredrik Ohlsson, Christoffer Petersson, and Daniel Persson. Equivariance versus augmentation for spherical images, 2022.

[Kondor and Trivedi, 2018] Risi Kondor and Shubhendu Trivedi. On the generalization of equivariance and convolution in neural networks to the action of compact groups. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 2747–2755, Stockholmsmässan, Stockholm Sweden, 10–15 Jul 2018. PMLR.

[Lagrave and Barbaresco, 2021] Pierre-Yves Lagrave and Frédéric Barbaresco. Generalized SU(1,1) Equivariant Convolution on Fock-Bargmann Spaces for Robust Radar Doppler Signal Classification. working paper or preprint, July 2021.

[Lagrave and Barbaresco, 2022] Pierre-Yves Lagrave and Frédéric Barbaresco. Hyperbolic Equivariant Convolutional Neural Networks for Fish-Eye Image Processing. working paper or preprint, February 2022.

[Lagrave *et al.*, 2021] P.-Y Lagrave, Yann Cabanes, and Frédéric Barbaresco. Su(1,1) equivariant neural networks and application to robust toeplitz hermitian positive definite matrix classification. *To appear in Proc. of GSI21*, 2021.

[Lang and Weiler, 2020] Leon Lang and Maurice Weiler. A wigner-eckart theorem for group equivariant convolution kernels, 2020.

[Lim *et al.*, 2019] Sungbin Lim, Ildoo Kim, Taesup Kim, Chiheon Kim, and Sungwoong Kim. Fast autoaugment. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d' Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[Lyle *et al.*, 2020] Clare Lyle, Mark van der Wilk, Marta Kwiatkowska, Yarin Gal, and Benjamin Bloem-Reddy. On the benefits of invariance in neural networks, 2020.

[Mitchel *et al.*, 2022] Thomas W. Mitchel, Noam Aigerman, Vladimir G. Kim, and Michael Kazhdan. Möbius convolutions for spherical cnns, 2022.

[Shorten and Khoshgoftaar, 2019] Connor Shorten and Taghi Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of Big Data*, 6, 07 2019.

[Wang *et al.*, 2022] Rui Wang, Robin Walters, and Rose Yu. Data augmentation vs. equivariant networks: A theory of generalization on dynamics forecasting, 2022.