Equivariance-bridged SO(2)-Invariant Representation Learning using Graph Convolutional Network

Sungwon Hwang
shwang.14@kaist.ac.kr
Hyungtae Lim
shapelim@kaist.ac.kr
Hyun Myung
hmyung@kaist.ac.kr

School of Electrical Engineering, Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Korea

Abstract

Training a Convolutional Neural Network (CNN) to be robust against rotation has mostly been done with data augmentation. In this paper, another progressive vision of research direction is highlighted to encourage less dependence on data augmentation by achieving structural rotational invariance of a network. The deep equivariance-bridged SO(2) invariant network is proposed to echo such vision. First, Self-Weighted Nearest Neighbors Graph Convolutional Network (SWN-GCN) is proposed to implement Graph Convolutional Network (GCN) on the graph representation of an image to acquire rotationally equivariant representation, as GCN is more suitable for constructing deeper network than spectral graph convolution-based approaches. Then, invariant representation is eventually obtained with Global Average Pooling (GAP), a permutation-invariant operation suitable for aggregating high-dimensional representations, over the equivariant set of vertices retrieved from SWN-GCN. Our method achieves the state-of-the-art image classification performance on rotated MNIST and CIFAR-10 images, where the models are trained with a non-augmented dataset only. Quantitative validations over invariance of the representations also demonstrate strong invariance of deep representations of SWN-GCN over rotations.

1 Introduction

In the past few years, Convolutional Neural Network (CNN) has brought many advances, especially on many computer vision tasks. The high performance leverages the use of learned convolution filters, with phenomenal techniques to make the layers go deeper [7]. Such advancements led to the performance close to human in image classification tasks on various datasets [7, 15, 18, 28, 35]. Especially, deeper layers of CNN have been empirically shown to learn substantially more translation-invariant features in each layer, which takes account for a wide range of applicability and reliability of CNN [4, 6, 25].

However, achieving rotation invariance is another desirable property of a network, especially on applications that require inferences over arbitrarily rotated images, such as aerial

© 2021. The copyright of this document resides with its authors.
It may be distributed unchanged freely in print or electronic forms.

[30] or biomedical microscopy images [32]. To do so, the most common practice to train a neural network to yield rotation-invariant representations is data augmentation [22]. By providing rotation-augmented images for training, a network can learn representations expressed in different rotations. If associated with a correct objective function, the network can yield reasonably invariant representations or inferences regardless of the rotation of an input [26].

Yet, models trained with data augmentation may fail to capture local equivariance and entail the black-box problem [33]. Besides, with an extensive list of data augmentations to train a transformation-robust network, the exponential growth of search space for augmentations inflates the dataset size [26]. Thus, liberation from rotation augmentation during training not only decreases a substantial number of training but can also give more search space for other types of data augmentations. Considering that rotation is one of the most common types of data augmentations [26], achieving rotation-invariance of a network is a large leap towards less dependence on data augmentation.

To address this issue, TIGraNet [11] made the most recent attempt to explicitly define the aforementioned problem and to suggest a solution. The researchers validated the transformation invariance of their proposed network by training their network only with upright 2D images with no transformation augmentation and evaluating the network with images augmented with isometric transformations. We extend their view and propose an alternative over their spectral graph convolution-based method to construct deeper image representation network using an equivariance-bridged SO(2) invariant graph convolutional network.

2 Related Works

Transformation-Equivariant Networks Instead of obtaining multiple filters that represent different rotations, steerable filters can be constructed with finite linear combinations of irreducible representations in order to achieve transformation equivariance [6]. The recent work extends the concept to construct steerable CNN by obtaining homomorphisms of transformations built with base representations from transformation-equivariant filter bank [4]. Following works include parametrizing the steerable filters [32], restricting the filters to be of the form from the circular harmonic family to achieve hard-baked rotation equivariance [33], or applying convolutional filters at multiple orientations to retrieve vector field representations of deep features [21].

The most recently, E(2)-CNN [31] makes a holistic implementation of the aforementioned transformation-equivariant networks on the steerable filters. Based on the group theory, the implementation achieves E(2) (translation, rotation and reflection in Euclidean space) equivariance and achieves the state-of-the-art performance over MNIST rot [17] dataset classification task.

Deep Learning on Graphs Promising potentials of graph-based networks were demonstrated from the generalization of CNN to low-dimensional graph domains along with the extension of the convolution via the Laplacian Spectrum [1]. Then, spectral network with a graph estimation procedure enabled the graph-based network to go deeper, demonstrating its superior performance over large-scale classification problems [8].

Meanwhile, GCN [13] has been one of the most prevalent graph processing networks and was devised as renormalized first-order approximations of spectral graph convolutions to conduct semi-supervised learning on graph-structured data. GCN has been effectively applied to a wide range of fields, such as but not limited to multi-label image recognition [2], temporal action localization [36], and even solid-state material science [24].

Transformation-Invariant Networks Ti-Pooling [16] achieves rotation invariant representation by feeding multiple rotated instances of an image to a siamese network, followed by element-wise pooling. Meanwhile, Spatial Transformer Network (STN) achieves the capacity to yield transformation-invariant representation by learning the affine transformation within the data with much distortion [10]. In the meantime, graph-based isometry-invariant network [11] was proposed as a successful attempt to represent an image to be isometry-invariant using graph. Their methodology uses spectral convolutions and dynamic pooling to retrieve isometry-equivariant graph representation of an image, followed by a statistical layer over Chebyshev polynomial representations of graph signals to retrieve the isometry-invariant representations.

3 Problem Definition

3.1 Equivariance and Invariance

Given a function $f : X \to Y$, f is said to be *equivariant* to a group of transformations if every transformation $\psi \in \Psi$ of an input $\mathcal{X} \in X$ can be associated with an equal transformation of the corresponding representation $\mathcal{Y} \in Y$, or

$$f(\psi(\mathcal{X})) = \psi(f(\mathcal{X})) = \psi(\mathcal{Y}). \tag{1}$$

Meanwhile, given a function $g : Y \to Z$, g is said to be *invariant* over transformation ψ if ψ in space Y yields identity transformation in Z, or

$$g(\psi(\mathcal{Y})) = g(\mathcal{Y}). \tag{2}$$

Then, the composition of equivariant function f followed by invariant function g eventually yields an invariant function with respect to ψ as follows:

$$g(f(\psi(\mathcal{X}))) = g(\psi(f(\mathcal{X}))) = g(f(\mathcal{X})). \tag{3}$$

We bridge the equivariant network as the means to effectively project an input to a high-dimensional representation space instead of directly transforming an input image to invariant representation. The illustrative explanation can be found in Figure 1-(a).

3.2 Rethinking the Value of Equivariant Networks

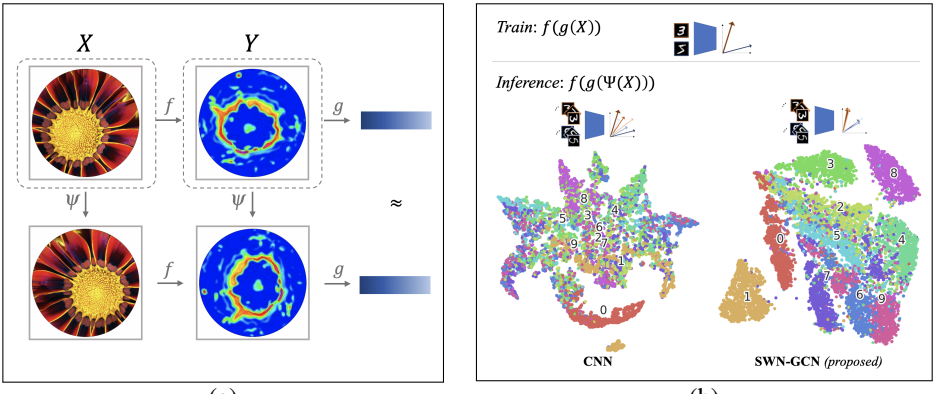
Most methodologies on equivariant networks [31, 33] validate their performances by training and testing their networks over rotation-augmented datasets, such as MNIST-rot [17]. The significance of the works is that the equivariant networks have larger capacity to learn all the different representations of rotation augmented inputs, since the representations are less variational and more equivariant, whose consistencies are easier to be adapted.

However, our objective lies on constructing a structurally invariant network that can make invariant inferences over rotations without rotation augmentations. Acquiring high-dimensional and equivariant feature space is a key step to achieving this goal by associating with a transformation-invariant function, as will be described in Section 3.3.

3.3 Equivariance-bridged SO(2) Invariant Network

Specifying X as a training dataset and given an objective function $\mathcal{L}$, our goal is to find a rotation-equivariant network $f(\cdot; w_f)$, rotation-invariant function $g(\cdot)$, and optimal parameters w_f^* of the network f that satisfy

$$w_f^* = \underset{w_f}{\operatorname{argmin}}\ \mathcal{L}(g(f(X; w_f))). \tag{4}$$



(a) (b)

Figure 1: (a) Illustration of equivariant function f, invariant function g, transformation function ψ, and their compositional relationships. Composition of f and g yields a Ψ-invariant representation (blue representation boxes); (b) Illustration of our objective and comparison of t-SNE [20] visualized feature representations of randomly rotated MNIST extracted with Convolutional Neural Networks (CNN) (left) and our proposed method (right). Both models are trained with non-augmented, upright MNIST dataset.

Then, given a rotation angle $\theta \in [0^\circ, 360^\circ)$, a corresponding rotation transformation $\mathbf{R}^\theta \in \mathbf{R}$, where $\mathbf{R}$ is a group of rotation transformation on image representation and forms a homomorphism with SO(2), and a correspondingly rotated image $\mathcal{R}^\theta = \mathbf{R}^\theta(\mathcal{X})$, our objective network and parameters should satisfy from Eq. (3) as

$$g(f(\mathcal{R}^\theta; w_f^*)) = g(f(\mathcal{X}; w_f^*)). \tag{5}$$

Thus, $g(f(\cdot; w_f^*))$ trained with the upright dataset X should be able to yield invariant representation for every $\mathcal{R}^\theta$. Such objective and t-SNE [20] visualized representations of randomly rotated input images extracted from our method compared with those extracted from ResNet-50 are available in Figure 1-(b).

4 Proposed Method

The schematics of our proposed network is summarized in Figure 2. Self-Weighted Nearest-Neighbors Graph Convolutional Network (SWN-GCN) is proposed to learn graph-based features in high dimensional and rotation-equivariant representation space, followed by Global Average Pooling (GAP) for invariance mapping of equivariant representations.

4.1 SWN-GCN

Propagation Rule GCN [13] is applied over an instance of image $\mathcal{X}$ of width W and height H expressed as an undirected graph representation $\mathcal{G} = (\mathcal{V}, A)$. $\mathcal{V}$ is a set of $|\mathcal{V}| = W \cdot H$ number of vertices and $A \in \mathbb{R}^{|\mathcal{V}| \times |\mathcal{V}|}$ is the adjacency matrix between the vertices. Specifically, we start off by representing $\mathcal{X}$ as a graph with vertices of

$$\mathcal{V}^{(0)} = [\mathcal{V}^{(0)}_{(1,1)} \cdots \mathcal{V}^{(0)}_{(w,h)} \cdots \mathcal{V}^{(0)}_{(W,H)}]^\top \in \mathbb{R}^{|\mathcal{V}| \times c^{(0)}} \tag{6}$$

where $\mathcal{V}^{(0)}_{(w,h)}$ denotes channel-wise values located at (w,h) in image $\mathcal{X}$ and $c^{(0)}$ is the size of input channel of $\mathcal{X}$, i.e. $c^{(0)} = 1$ for gray-scale images and $c^{(0)} = 3$ for color images.

The component of the adjacency matrix A that represents the adjacency between two vertices $\mathcal{V}^{(0)}_{(w,h)}$ and $\mathcal{V}^{(0)}_{(w',h')}$ is denoted as $A_{[(w,h),(w',h')]}$ and is defined as follows:

$$A_{[(w,h),(w',h')]} = \begin{cases} 1 & \text{if } 0 < d^{(w',h')}_{(w,h)} \leq \sqrt{2} \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

where $d^{(w',h')}_{(w,h)} = \sqrt{(w-w')^2 + (h-h')^2}$.

Then, given the l-th propagated set of vertices $\mathcal{V}^{(l)}$, a unit propagation of SWN-GCN comprises the two networks, which we will refer to as *Self-weighted Message Passing* network (SMP, $\zeta(\cdot)$) and *Shared-weight Graph Propagation* network (SGP, $\xi(\cdot)$), to construct a unit layer of our equivariant network $f^{(l)} : \mathcal{V}^{(l)} \rightarrow \mathcal{V}^{(l+1)}$ correspondingly as

$$\hat{\mathcal{V}}^{(l)} = \bar{D}^{-\frac{1}{2}} \bar{A}^{(l)} \bar{D}^{-\frac{1}{2}} \mathcal{V}^{(l)} := \zeta^{(l)}(\mathcal{V}^{(l)}, A) \tag{8}$$

$$\mathcal{V}^{(l+1)} = \kappa^{(l)}_2(\kappa^{(l)}_1(\hat{\mathcal{V}}^{(l)})) := \xi^{(l)}(\hat{\mathcal{V}}^{(l)}) \tag{9}$$

where $\bar{A}^{(l)} = A + \beta^{(l)} I_{|\mathcal{V}|}$ is the self-weighted adjacency matrix and $\beta^{(l)}$ is a trainable parameter. $\bar{D} \in \mathbb{R}^{|\mathcal{V}| \times |\mathcal{V}|}$ is a diagonal matrix formulated as:

$$\bar{D}_{[(w,h),(w,h)]} = 1 + \sum_{w',h'=1}^{w'=W,h'=H} A_{[(w,h),(w',h')]}. \tag{10}$$

The trainable function $\kappa^{(l)}_i(\cdot)$ in SGP is defined as

$$\kappa^{(l)}_i(\cdot) = \sigma(BN^{(l)}_i(\cdot W^{(l)}_i)) \tag{11}$$

where $c'^{(l)}, W^{(l)}_1 \in \mathbb{R}^{c^{(l)} \times c'^{(l)}}$, $W^{(l)}_2 \in \mathbb{R}^{c'^{(l)} \times c^{(l+1)}}$, BN_i, and σ are intermediate dimension size, first and second propagation parameters, batch-normalization [9], and ReLU nonlinearity, respectively.

Leveraging GCN to construct deeper model Given a diagonal matrix D where $D_{ii} = \sum_j A_{ij}$, if the linear approximation of Chebyshev polynomials of spectral graph convolution is applied over our method, SMP would be formulated as $\hat{\mathcal{V}}^{(l)} = (I_{|\mathcal{V}|} + D^{-\frac{1}{2}} A D^{-\frac{1}{2}}) \mathcal{V}^{(l)}$. However, eigenvalues of $I_{|\mathcal{V}|} + D^{-\frac{1}{2}} A D^{-\frac{1}{2}}$ range in $[0,2]$, which means that multiple stacks of these layers to construct deeper models may cause exploding or vanishing gradient problems. GCN tackled this concern directly and conducted renormalization of the operation by substituting $I_{|\mathcal{V}|} + D^{-\frac{1}{2}} A D^{-\frac{1}{2}}$ with $\tilde{D}^{-\frac{1}{2}} \tilde{A} \tilde{D}^{-\frac{1}{2}}$, where $\tilde{A} = A + I_{|\mathcal{V}|}$ and $\tilde{D}_{ii} = \sum_j \tilde{A}_{ij}$, thus allowing the network to back-propagate more stable in deeper layers. We additionally included batch normalization to reduce the typical covariance shift problem [9] in deeper models. More details on the derivation of the approximation of spectral graph convolution and the following discussion can be found in [13].

4.2 SO(2)-Equivariant Property of SWN-GCN

In this subsection, we show that the architecture of SWN-GCN yields rotation-equivariant representation by its structural nature. Primarily, representation of a rotated image must be strictly defined.

Definition 1. *Given* (u',v') *and* $\mathbf{T}^\theta \in SO(2)$ *that satisfies* $\mathbf{T}^\theta[w - \frac{W}{2}, h - \frac{H}{2}]^T = [u' - \frac{W}{2}, v' - \frac{H}{2}]^T$, *every pixel value of rotated image* $\mathcal{R}^\theta_{(u,v)}$ *is defined as*:

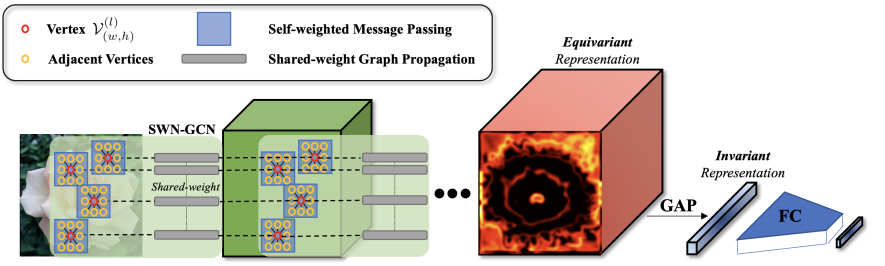


Figure 2: Self-Weighted Nearest-Neighbors Graph Convolutional Network (SWN-GCN).

$$\mathcal{R}^{\theta}_{(u,v)} := h(\mathcal{X}_{(w,h)}; (u', v')) \tag{12}$$

where $h(\cdot;(u',v'))$ is an interpolation function [29] to assign interpolated values to the closest pixel at (u,v) in $\mathcal{R}^{\theta}$. We may say that $\mathcal{R}^{\theta}_{(u,v)}$ and $\mathcal{X}_{(w,h)}$ forms a spatial correspondence.

Then, we can straightforwardly show rotation equivariance of the l-th propagated vertex set $\mathcal{V}^{(l)}$ by showing that every vertex of an upright image and the spatially corresponding vertex of a rotated image are equal throughout the propagation as the following proposition shows.

Proposition 1. *Let $\mathcal{H}$ be the vertex set representation of a rotated image $\mathcal{R}^{\theta}$. Then, for all $w \in \{1 \cdots W\}$, $h \in \{1 \cdots H\}$ and $l \in \{0 \cdots L_f\}$, the following approximation holds*

$$\mathcal{H}^{(l)}_{(u,v)} \approx \mathcal{V}^{(l)}_{(w,h)}. \tag{13}$$

Proof. Inductive method is used to prove the proposition.

(a) When $l = 0$: or when every vertex representation is the original pixel value, we can safely make the following assumption:

Assumption 1. $\mathcal{H}^{(0)}_{(u,v)} \approx \mathcal{V}^{(0)}_{(w,h)}$ *for all* $w \in \{1 \cdots W\}$, $h \in \{1 \cdots H\}$.

The assumption is reasonable from our definition of spatial correspondence in Eq. (12) given that interpolation does not significantly change the value of most pixels.

(b) Then, provided that Eq. (13) holds when $l = n$, we need to show that the equation holds when $l = n+1$, or

$$\xi^{(n)}(\zeta^{(n)}(\mathcal{V}^{(n)}))_{(w,h)} = \xi^{(n)}(\zeta^{(n)}(\mathcal{H}^{(n)}))_{(u,v)}. \tag{14}$$

First, SMP does preserve rotational invariance of spatial correspondence, or $\zeta^{(n)}(\mathcal{V}^{(n)})_{(w,h)} = \zeta^{(n)}(\mathcal{H}^{(n)})_{(u,v)}$. Since there are only 9 vertices including itself (assume that image is rotated with zero padding) that yields non-zero adjacency for every vertex according to Eq. (7), degree matrix $\bar{D}$ in Eq. (10) can be approximated as $9I_{|\mathcal{V}|}$. The degree matrix can thus be approximated as a scalar multiplication of an identity matrix, from which Eq. (8) can be rearranged as

$$\bar{D}^{-\frac{1}{2}}\bar{A}^{(l)}\bar{D}^{-\frac{1}{2}} = \bar{D}^{-1}\bar{A}^{(l)}. \tag{15}$$

Then, Eq. (15) can be utilized to express $\hat{\mathcal{V}}^{(n)}_{(w,h)} = \zeta^{(n)}(\mathcal{V}^{(n)})_{(w,h)}$ as

$$\hat{\mathcal{V}}^{(n)}_{(w,h)} = \frac{1}{9}(\beta^{(n)}\mathcal{V}^{(n)}_{(w,h)} + \sum_{\substack{i=w-1,j=h-1 \\ i\neq w, j\neq h}}^{w+1,h+1} \mathcal{V}^{(n)}_{(i,j)}). \tag{16}$$

Likewise, $\hat{\mathcal{H}}^{(n)}_{(u,v)} = \zeta^{(n)}(\mathcal{H}^{(n)})_{(u,v)}$ can also be expressed with the same process we leveraged to obtain Eq. (16) as

$$\hat{\mathcal{H}}^{(n)}_{(u,v)} = \frac{1}{9}(\beta^{(n)}\mathcal{H}^{(n)}_{(u,v)} + \sum_{\substack{i=u-1,j=v-1\\ i\neq u,j\neq v}}^{u+1,v+1} \mathcal{H}^{(n)}_{(i,j)}). \tag{17}$$

In fact, the expectation of sum of adjacent vertices of $\mathcal{V}^{(n)}_{(w,h)}$ and $\mathcal{H}^{(n)}_{(u,v)}$ in Eq. (16) and Eq. (17) are equal. Specifically, we have local rotational consistency as shown in the following assumption:

Assumption 2. *(Local Rotational Consistency)*

$$\mathbb{E}\left[\sum_{\substack{i=w-1,j=h-1\\ i\neq w,j\neq h}}^{w+1,h+1} \mathcal{V}^{(n)}_{(i,j)}\right] = \mathbb{E}\left[\sum_{\substack{i=u-1,j=v-1\\ i\neq u,j\neq v}}^{u+1,v+1} \mathcal{H}^{(n)}_{(i,j)}\right]. \tag{18}$$

Rigid rotation of an image does not change the list of adjacent vertices, which means that their sum remains constant under rigid rotation. Yet, the values of vertices retrieved from images with rotations that are not multiples of 90° may be slightly different due to interpolations. However, it is reasonable to assume that the sum of the adjacent vertices, each of which has slightly deviating values from interpolation, is acceptably constant. Experimental results in the later section show that this assumption is reasonable enough to yield the most invariant representation out of all baselines (see Section 6.1). Meanwhile, we are given with $\beta^{(n)}\mathcal{V}^{(n)}_{(w,h)} \approx \beta^{(n)}\mathcal{H}^{(n)}_{(u,v)}$ from the inductive assumption, from which we can conclude $\zeta^{(n)}(\mathcal{V}^{(n)})_{(w,h)} \approx \zeta^{(n)}(\mathcal{H}^{(n)})_{(u,v)}$.

Then, $\xi^{(n)}(\hat{\mathcal{V}}^{(n)})_{(w,h)} \approx \xi^{(n)}(\hat{\mathcal{H}}^{(n)})_{(u,v)}$ straightforwardly holds because multiplied weights are shared and ReLU is not a one-to-many function. Meanwhile, batch-normalization does not strictly but acceptably preserves the approximate equality (see Section A in supplementary material for details). Thus, we can conclude $\mathcal{V}^{(n+1)}_{(w,h)} \approx \mathcal{H}^{(n+1)}_{(u,v)}$ given $\mathcal{V}^{(n)}_{(w,h)} \approx \mathcal{H}^{(n)}_{(u,v)}$ and finalize the inductive proof. □

4.3 Global Average Pooling for Invariant Mapping and Classification of Invariant Representations

Recent graph-based networks [11, 34] employ the statistical layer, which computes means and variances of graph signals using graph Chebyshev polynomials of order up to $K_{\max}$, to map an equivariant vertex set to an invariant representation. Given set of vertices after the L_f-th propagation, where every vertex is represented in $\mathbb{R}^{c^{(L_f)}}$, the statistical layer calculates $c^{(L_f)} \cdot (K_{\max}+1)$ number of means and variances. However, such process may be burdensome on equivariant representations expressed in high representation space retrieved from deep networks like ours, where the dimension of representation space can go up to $c^{(L_f)} = 512$.

Meanwhile, deep network such as PointNet [23] or Residual Network [7] demonstrated that a global permutation-invariant operation, such as max or average operation, is capable of efficiently aggregating high-dimensional representations. To this end, we employ GAP to aggregate rotation-equivariant set of vertices, $\mathcal{V}^{(L_f)}$, to invariant representation $\mathbf{z} \in \mathbb{R}^{c^{(L_f)}}$ as

$$\mathbf{z} = \frac{1}{W \cdot H} \sum_{w=1,h=1}^{W,H} \mathcal{V}^{(L_f)}_{(w,h)}. \tag{19}$$

Since sum is a permutation-invariant operation, GAP yields rotation-invariant $\mathbf{z}$ out of rotation-equivariant vertices $\mathcal{V}^{(L_f)}$, where the permutation occurs within the rotation-equivariant set of vertices. Unlike direct use of GAP [19] that requires the last feature representation dimension equal to the number of classification classes, our equivariant space is mapped into much higher dimension before GAP and the final classification is conducted with fully connected layers non-linearized with ReLU. The summary of the methods can be found in Algorithm 1.

Algorithm 1 SWN-GCN for SO(2)-invariant representation of an image

given $\mathcal{X}$ (input image), L (Layer configurations)
$\mathcal{G} = (\mathcal{V}, A) \leftarrow$ *Image2Graph*($\mathcal{X}$) ▷ Eq. 6, Eq. 7
for l in L **do**
  $\hat{\mathcal{V}} \leftarrow \zeta^{(l)}(\mathcal{V}, A)$ ▷ Eq. 8
  $\mathcal{V} \leftarrow \xi^{(l)}(\hat{\mathcal{V}})$ ▷ Eq. 9
end for
$z \leftarrow$ *GAP*($\mathcal{V}$) ▷ Eq. 19
return *linear_classifier*(z)

5 Experiment

5.1 Dataset

MNIST The MNIST dataset [18] comprises of images with grayscale, handwritten digits of 10 classes ranging from 0 to 9. The dataset includes 60,000 train and 10,000 test images.

CIFAR-10 The CIFAR-10 dataset [14] consists of 60,000 32×32 images in 10 classes with 6,000 images per class. The dataset includes 50,000 training images and 10,000 test images. All images are circularly masked to yield minimum interference caused by black spaces on edges created after rotation for the sake of the experiment.

5.2 Experimental Setup

We compare our model with the state-of-the-art E(2)-equivariant network, E(2)-CNN [31], as well as Harmonic Network [33] and TIGraNet [11]. The most prevalent CNNs such as VGG-19 [27] and ResNet-50 [7] are also compared. For E(2)-CNN, we selected C_8, as it showed the best performance out of the rotation groups from our experiment. GAP is applied for rotation equivariant models with no specific implementation of invariance mapping. The number of parameters for all models are kept approximately constant, except for VGG-19 and ResNet-50 since these are not designed for rotation equivariance but are included as baseline models to demonstrate the degree of rotation-invariance of other state-of-the-art models.

5.3 Training Details

80% of the training sets are used for training, and the remainders for validation dataset. Models are optimized with ADAM optimizer [12] with learning rate of 10^{-4} and batch size of 64 until the models show no improvement over validation dataset for 40 consecutive epochs. All models are trained three times, and all the reports on classification accuracies are their averages. Also, models are deliberately trained with no augmentation, including rotation as well as typical augmentation methods such as flips or random crop, in order to demonstrate the strict measure of rotation invariance.

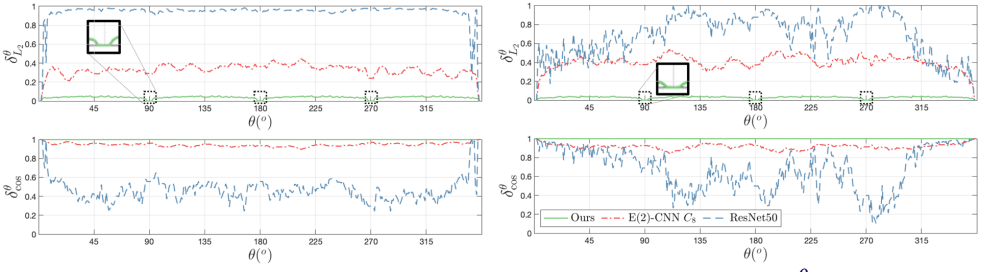


Figure 3: (L-R): Plots of the averages of rotational L_2 norm of variance ($\delta^\theta_{L_2}$) and rotational cosine distance of invariance ($\delta^\theta_{\cos}$) of invariant representations over rotation angle θ for 10 randomly selected images from (L) MNIST and (R) CIFAR-10 dataset. For $\delta^\theta_{L_2}$, the lower, the better and for $\delta^\theta_{\cos}$, the higher, the better.

5.4 Metrics

Invariance of Representations Besides the classification accuracy, metrics are defined to evaluate the degree of invariance of the representations. Let $\mathbf{z}^\theta$ be the representation of the input image $\mathcal{R}^\theta$. Then, relative L_2 norm of rotational variance ($\delta^\theta_{L_2}$) and absolute cosine similarity of rotational invariance ($\delta^\theta_{\cos}$) are defined correspondingly as

$$\delta^\theta_{L_2} = \frac{||\mathbf{z}^\theta - \mathbf{z}^0||}{||\mathbf{z}^0||}, \ \delta^\theta_{\cos} = \frac{||\mathbf{z}^\theta \cdot \mathbf{z}^0||}{||\mathbf{z}^\theta||\ ||\mathbf{z}^0||} \tag{20}$$

where $||\cdot||$ is the L_2 norm.

6 Results and Discussions

6.1 Validations over Invariance

Quantitative Validation on Invariance Given 10 randomly sampled images from MNIST and CIFAR-10 each, $\delta^\theta_{L_2}$ and $\delta^\theta_{\cos}$ are measured for $\theta \in [0, 360)$ with ResNet-50, E(2)-CNN C_8, and SWN-GCN, and the averages of the results are reported in Figure 3. SWN-GCN produces $\delta^\theta_{L_2}$ closer to 0 and $\delta^\theta_{\cos}$ closer to 1 over all angles of rotations than other baseline models, which shows that SWN-GCN yields the most rotation-invariant representations out of the baselines. Especially, note that ours yields the exact value of $\delta^\theta_{L_2} = 0$ for θ with multiples of 90°. E(2)-CNN yields significantly better invariance than ResNet-50, but still yields noticeable variances in both $\delta^\theta_{L_2}$ and $\delta^\theta_{\cos}$ over all rotation angles.

6.2 Rotation-Invariant Image Classification

We also demonstrate the classification accuracy over rotation augmented dataset of our network with other baselines, where all the models are trained with upright images only. As shown in Table 1, classification accuracy of our proposed model outperformed all baseline models on the overall accuracy (OA) on test datasets that are augmented with random rotation angles.

On top of the highest classification accuracy on the rotation augmented dataset, SWN-GCN shows significant improvements in classification accuracy over the largest range of rotation. Table 1 reports classification accuracies over dataset augmented with fixed angles with multiples of 30°. Even though ResNet-50 shows the highest performance over upright image classification, all other baselines yield higher classification accuracy on rotated images than ResNet-50 in most rotation angles. In particular, SWN-GCN outperforms TIGraNet, the state-of-the-art graph-based isometry-invariant network, in all rotation angles.

Table 1: Classification accuracies over test datasets rotated by fixed angles. Overall Accuracy (OA) is the result over all range of rotation.

		Classification Accuracy (%)												
Dataset	Models	0°	30°	60°	90°	120°	150°	180°	210°	240°	270°	300°	330°	OA
MNIST	ResNet-50 [□]	**99.5**	91.9	47.7	28.1	29.7	48.8	57.4	51.1	33.7	23.5	35.2	90.0	42.4
	E(2)-CNN C_8 [▭]	99.3	**98.1**	**95.9**	96.3	86.2	74.9	70.7	71.1	81.8	95.1	**92.9**	**97.0**	87.5
	TIGraNet [▭]	89.1	82.7	79.8	89.1	82.7	79.8	89.1	82.7	79.8	89.1	82.7	79.8	85.1
	SWN-GCN	96.5	89.8	87.3	**96.5**	**89.8**	**87.3**	**96.5**	**89.8**	**87.3**	**96.5**	89.8	87.3	**91.8**
CIFAR-10	ResNet-50	**85.1**	54.5	34.1	18.3	27.5	26.9	35.6	27.0	24.9	33.8	33.2	52.5	36.1
	E(2)-CNN C_8	77.1	**57.8**	44.3	48.5	34.4	30.8	37.8	31.9	35.4	49.4	45.0	**56.0**	46.2
	TIGraNet	38.9	37.0	36.8	38.9	37.0	36.8	38.9	37.0	36.8	38.9	37.0	36.8	38.1
	SWN-GCN	51.3	49.6	**50.1**	**51.3**	**49.6**	**50.1**	**51.3**	**49.6**	**50.1**	**51.3**	**49.6**	50.1	**50.5**

Meanwhile, one may observe from Table 1 and Figure 3 that for some rotation angles, E(2)-CNN yields better $\delta_{L_2}^{\theta}$ and $\delta_{\cos}^{\theta}$ over SWN-GCN yet shows higher classification performances than SWN-GCN (i.e. $\theta = 0°, 30°, 60°$ for MNIST and $\theta = 0°, 30°$ for CIFAR-10). For these angles, the impact of difference in expressibility of representations between E(2)-CNN and SWN-GCN is more significant than the disruptions on representations introduced by input image rotation on E(2)-CNN. However, disruptions introduced by a moderate amount of input image rotation on E(2)-CNN easily overwhelm the advantage of E(2)-CNN on expressibility, as proven by SWN-GCN outperforming E(2)-CNN in most range of rotations for image classifications.

Such trade-off of advantages between the two models can be explained by making an analogous comparison with CNN by referring to Eq. (16) for SWN-GCN. All parameters in convolution kernels, i.e. 3×3, are trainable in CNN, meaning that these kernels are as expressive as they can be. However, these parameters have different values within the convolution kernel and break the local rotation consistency. Whereas for SWN-GCN, only the self-vertex is multiplied with trainable parameters, and the magnitude of adjacency with adjacent vertices are uniform and fixed for the sake of preserving local rotational consistency.

7 Conclusion

We proposed a network that yields equivariant representation with SWN-GCN and invariant representation using GAP. We showed structural equivariance of SWN-GCN and invariance of GAP, and validated the properties with experimental results. Our method achieved the state-of-the-art performances on rotated MNIST and CIFAR-10 image classification, where the models were trained with upright images only.

8 Acknowledgement

This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (2021-0-01112, *"Development of location identification algorithms that are robust to spatial changes"*). The students are supported by the BK21 FOUR from the Ministry of Education (Republic of Korea).

References

[1] Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann LeCun. Spectral networks and locally connected networks on graphs. *arXiv preprint arXiv:1312.6203*, 2013.

[2] Zhao-Min Chen, Xiu-Shen Wei, Peng Wang, and Yanwen Guo. Multi-label image recognition with graph convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5177–5186, 2019.

[3] Taco Cohen and Max Welling. Group equivariant convolutional networks. In *Proceedings of the International Conference on Machine Learning*, pages 2990–2999, 2016.

[4] Taco S Cohen and Max Welling. Steerable cnns. *arXiv preprint arXiv:1612.08498*, 2016.

[5] William T Freeman, Edward H Adelson, et al. The design and use of steerable filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(9):891–906, 1991.

[6] Ian Goodfellow, Honglak Lee, Quoc Le, Andrew Saxe, and Andrew Ng. Measuring invariances in deep networks. In *Advances in Neural Information Processing Systems*, pages 646–654, 2009.

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[8] Mikael Henaff, Joan Bruna, and Yann LeCun. Deep convolutional networks on graph-structured data. *arXiv preprint arXiv:1506.05163*, 2015.

[9] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the International Conference on Machine Learning*, pages 448–456. PMLR, 2015.

[10] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial transformer networks. In *Advances in Neural Information Processing Systems*, pages 2017–2025, 2015.

[11] Renata Khasanova and Pascal Frossard. Graph-based isometry invariant representation learning. In *Proceedings of the International Conference on Machine Learning*, pages 1847–1856, 2017.

[12] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[13] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

[14] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

[15] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017.

[16] Dmitry Laptev, Nikolay Savinov, Joachim M Buhmann, and Marc Pollefeys. Ti-pooling: transformation-invariant pooling for feature learning in convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 289–297, 2016.

[17] 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 International Conference on Machine Learning*, pages 473–480, 2007.

[18] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[19] Min Lin, Qiang Chen, and Shuicheng Yan. Network in network. *arXiv preprint arXiv:1312.4400*, 2013.

[20] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(Nov):2579–2605, 2008.

[21] Diego Marcos, Michele Volpi, Nikos Komodakis, and Devis Tuia. Rotation equivariant vector field networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5048–5057, 2017.

[22] Luis Perez and Jason Wang. The effectiveness of data augmentation in image classification using deep learning. *arXiv preprint arXiv:1712.04621*, 2017.

[23] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 652–660, 2017.

[24] Jonathan Schmidt, Mário RG Marques, Silvana Botti, and Miguel AL Marques. Recent advances and applications of machine learning in solid-state materials science. *npj Computational Materials*, 5(1):1–36, 2019.

[25] Uwe Schmidt and Stefan Roth. Learning rotation-aware features: From invariant priors to equivariant descriptors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2050–2057, 2012.

[26] Connor Shorten and Taghi M Khoshgoftaar. A survey on image data augmentation for deep learning. *Journal of Big Data*, 6(1):60, 2019.

[27] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[28] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–9, 2015.

[29] Philippe Thévenaz, Thierry Blu, and Michael Unser. Image interpolation and resampling. *Handbook of Medical Imaging, Processing and Analysis*, 1(1):393–420, 2000.

[30] Shidong Wang, Yi Ren, Gerard Parr, Yu Guan, and Ling Shao. Invariant deep compressible covariance pooling for aerial scene categorization. *IEEE Transactions on Geoscience and Remote Sensing*, 2020.

[31] Maurice Weiler and Gabriele Cesa. General $e(2)$-equivariant steerable cnns. In *Advances in Neural Information Processing Systems*, pages 14334–14345, 2019.

[32] 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*, pages 849–858, 2018.

[33] 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*, pages 5028–5037, 2017.

[34] Qin Yang, Chenglin Li, Wenrui Dai, Junni Zou, Guo-Jun Qi, and Hongkai Xiong. Rotation equivariant graph convolutional network for spherical image classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4303–4312, 2020.

[35] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *Proceedings of the European Conference on Computer Vision*, pages 818–833. Springer, 2014.

[36] Runhao Zeng, Wenbing Huang, Mingkui Tan, Yu Rong, Peilin Zhao, Junzhou Huang, and Chuang Gan. Graph convolutional networks for temporal action localization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7094–7103, 2019.