Bioinformatics, 35, 2019, i530–i537
doi: 10.1093/bioinformatics/btz353
ISMB/ECCB 2019

OXFORD

Rotation equivariant and invariant neural networks for microscopy image analysis

Benjamin Chidester[1], Tianming Zhou[1], Minh N. Do[2] and Jian Ma[1,*]

[1]Computational Biology Department, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213, USA and [2]Department of Electrical and Computer Engineering, University of Illinois, Urbana, IL 61801, USA

*To whom correspondence should be addressed.

Abstract

Motivation: Neural networks have been widely used to analyze high-throughput microscopy images. However, the performance of neural networks can be significantly improved by encoding known invariance for particular tasks. Highly relevant to the goal of automated cell phenotyping from microscopy image data is rotation invariance. Here we consider the application of two schemes for encoding rotation equivariance and invariance in a convolutional neural network, namely, the group-equivariant CNN (G-CNN), and a new architecture with simple, efficient conic convolution, for classifying microscopy images. We additionally integrate the 2D-discrete-Fourier transform (2D-DFT) as an effective means for encoding global rotational invariance. We call our new method the *Conic Convolution and DFT Network* (CFNet).

Results: We evaluated the efficacy of CFNet and G-CNN as compared to a standard CNN for several different image classification tasks, including simulated and real microscopy images of subcellular protein localization, and demonstrated improved performance. We believe CFNet has the potential to improve many high-throughput microscopy image analysis applications.

Availability and implementation: Source code of CFNet is available at: https://github.com/bchidest/CFNet.

Contact: jianma@cs.cmu.edu

Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

Though the appeal of neural networks is their versatility for arbitrary classification tasks, there is still much benefit in designing them for particular problem settings. In particular, the effectiveness of neural networks can be greatly increased by encoding invariance to uniformative augmentations of the data (LeCun *et al.*, 1989). A key invariance inherent to many imaging contexts, including microscopy data, is *rotation* (Boland and Murphy, 2001). For biological imaging, since data is often scarce and difficult or expensive to acquire, improving the effectiveness and reliability of models by encoding such invariance is highly significant.

Recently, convolutional neural networks (CNNs) have been applied to the highly relevant problem of cell phenotyping based on microscopy image analysis and have demonstrated much improved performance (Kraus *et al.*, 2016, 2017). Formerly, crafted features that inherently exhibit such invariance, such as Zernike moments and Haralick texture features, were extracted and used for subsequent analysis (Boland and Murphy, 2001), whereas CNNs are able to learn relevant features directly. This has significant applications to spatial proteomics, which has enabled the systematic probing of changes of subcellular protein localizations, which are key to protein functions (Lundberg and Borner, 2019), as a response to various perturbations (Chong *et al.*, 2015; Kraus *et al.*, 2017). However, the encoding of rotation equivariance and invariance into CNNs to learn meaningful features for cell phenotyping is yet to be considered.

Several approaches have been proposed recently for improving the performance of CNNs by encoding rotation equivariance. The most promising and popular of such methods is the group-equivariant CNN (G-CNN) (Cohen and Welling, 2016), which applies convolution over groups, such as rotation, translation and flips, thereby maintaining equivariance throughout the convolutional layers. Notably, G-CNNs have recently been applied to several biological imaging tasks, including cell boundary segmentation (Bekkers *et al.*, 2018; Weiler *et al.*, 2018), annotation of cancerous regions of tumors (Veeling *et al.*, 2018) and dermoscopy image segmentation (Li *et al.*, 2018).

Here we consider the integration of rotation equivariance and invariance to analyze the localization of proteins in fluorescence images, which, to the best of our knowledge, is the first such work. Additionally, we propose a new simple and efficient

rotation-equivariant convolutional scheme, called *conic convolution* as an effective alternative to group convolution, with advantages of computational and memory savings, interpretability of learned feature maps and improved performance. Rather than convolving each filter across the entire image, as in standard or group convolution, rotated filters are convolved only over corresponding conic regions of the input feature map that emanate from the origin, thereby intuitively transforming rotations in the input directly to rotations in the output. A comparison of conic convolution with other proposed convolution schemes is shown in Figure 1.

To encode rotation *invariance*, we propose the integration of the magnitude response of the 2D-discrete-Fourier transform (2D-DFT) into a transition layer between convolutional and fully-connected layers. The 2D-DFT is able to integrate mutual orientation information between different filter responses, yielding more informative features for subsequent layers than most previous approaches. Though the insight of using the DFT to encode rotational invariance has been employed for texture classification using wavelets (Charalampidis and Kasparis, 2002; Do and Vetterli, 2002; Jafari-Khouzani and Soltanian-Zadeh, 2005; Ojala *et al.*, 2002) and for general image classification (Schmidt and Roth, 2012), as of yet, its application to CNNs has been relatively overlooked. As in these prior works, rotations of the input are transformed to circular shifts, to which the magnitude response of the 2D-DFT is invariant, in the transformed space.

We call our new method the Conic Convolution and DFT Network (CFNet). We demonstrate the effectiveness of the two novel contributions in CFNet, namely conic convolution and integration of the DFT, based on evaluations from both synthetic and real microscopy images for localizing proteins in budding yeast cells. We show that CFNet improves classification accuracy generally over the standard raster convolution formulation and over the equivariant method of G-CNN across these settings. We also show that the 2D-DFT clearly improves performance across these diverse datasets, and that not only for the proposed conic convolution, but also for group convolution.

1.1 Related work

To encode rotation equivariance for general image classification, a variety of methods exist. One straightforward strategy is to transform the domain of the image to an alternative domain, such as the log-polar domain (Henriques and Vedaldi, 2017; Schmidt and Roth, 2012) in which rotation becomes some other transformation that is easier to manage, but this can be unstable to translations and this warping will introduce distortion, as pixels near the center of the image are sampled more densely than pixels near the perimeter. Our proposed conic convolution also encodes global rotation equivariance about the origin, but without introducing such distortion, which greatly helps mitigate its susceptibility to translation. The recently developed spatial transform layer (Jaderberg *et al.*, 2015) and deformable convolutional layer (Dai *et al.*, 2017) allow the network to learn non-regular sampling patterns and can potentially help learning rotation invariance, though invariance is not explicitly enforced, which would most likely be a challenge for tasks with small training sets.

An alternative, simple means for achieving rotation equivariance and invariance was proposed in (Dieleman *et al.*, 2016), in which feature maps of standard CNNs are made equivariant or invariant to rotation by combinations of cyclic slicing, stacking, rolling and pooling. RotEqNet (Marcos *et al.*, 2017) improved upon this idea by storing, for each feature map for a corresponding filter, only the maximal response across rotations and the value of the corresponding rotation, to preserve pose information, yielding improved results and considerable storage savings. Our proposed conic convolution is most similar to these methods and further decreases storage and computation requirements. The recently developed capsule network (Sabour *et al.*, 2017) is able to auto-encode affine transformation, including rotation, by the routing-by-agreement process. However, our CFNet developed in this paper works well even without augmentation because equivariance and invariance are encoded. Another related work extended G-CNN using steerable filters (Weiler *et al.*, 2018), as proposed in H-Net (Worrall *et al.*, 2017), to provide equivariance for finer angles. This can be considered as a parallel contribution to our work, which could also use a steerable filter design. In summary, CFNet improves upon previous methods by reducing computation and storage requirements and improving interpretability and performance.

2 Materials and methods

We consider CFNet and G-CNN within the context of microscopy image analysis to classify cell features. Each network takes, as input, an image of a cell and predicts a label of interest, such as the localization of fluorescence-tagged proteins. The overall architecture of CFNet is illustrated in Figure 2. We first give a brief description of group-equivariant convolution and then describe our proposed conic convolution in CFNet, which uses similar notation from group theory. Next, we discuss the preservation of rotation equivariance through non-linear operations within a neural network as well as the efficiency of conic convolution. We then describe the integration of the 2D-DFT in CFNet as a transition layer between group or conic convolutional layers and subsequent fully-connected layers in a CNN.

2.1 Group-equivariant convolution

For convenience, as in Cohen and Welling (2016), we represent feature maps, of dimension K, $f : \mathbb{Z}^2 \to \mathbb{R}^K$ and filters, $\phi : \mathbb{Z}^2 \to \mathbb{R}^K$, of a standard CNN as functions over the 2D space $\mathbb{Z}^2$ of integers, or pixel locations in the case of images. The expression for convolution of a filter over a feature map in a standard CNN is given by:

$$f * \phi(x) = \sum_{k=0}^{K-1} \sum_{z \in \mathbb{Z}^2} f_k(z)\phi_k(z - x). \quad (1)$$

The success of CNNs can be attributed largely to the fact that standard convolution is equivariant to translations and many image classification tasks are invariant to small local translations. However, standard convolution does not in general exhibit equivariance to other important transformations, such as rotations, unless certain constraints on the filters are met. The insight of Cohen and Welling (2016) was to generalize convolution to operate on functions on *groups*, thereby achieving equivariance for other types of transformations. A group is a mathematical term referring to a particular set paired with a binary operation, which together meet certain criteria. The set of indices $\mathbb{Z}^2$ with the operation of translation is a particular instance of a group.

A more relevant group for microscopy image data is the $p4$ group, or roto-translation group, which consists of both translations and rotations about the origin of $\frac{\pi}{2}$ of $\mathbb{Z}^2$, and a function on this group is indexed not just by translation, such as the x position of feature maps of normal CNNs, but also by rotation. In this way, rotation information is preserved throughout the network and equivariance can thereby be maintained.

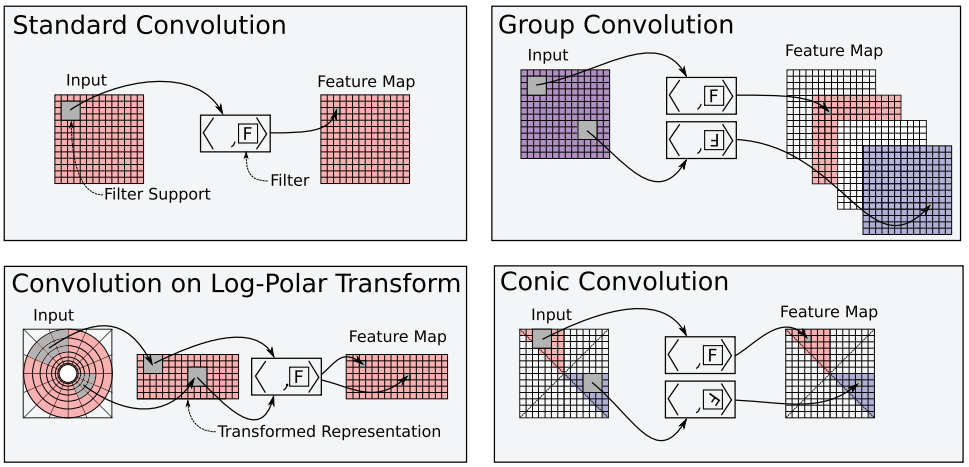

Fig. 1. Comparison of convolution schemes. The domain of filter 'F' in the input and its corresponding outputs in the feature map are colored red. That of the rotation of 'F' by 225 degrees is colored blue. The local support on the domain for the convolution at a few points for each scheme is shown in gray. Conic convolution, with rotations of 45 degrees in this example, encodes rotation equivariance without introducing distortion to the support of the filter in the original domain (unlike the log-polar transform) and without requiring additional storage for feature maps (unlike group convolution). The example shown for group convolution is the first layer of a G-CNN, mapping from $\mathbb{Z}^2$ to the roto-translation group

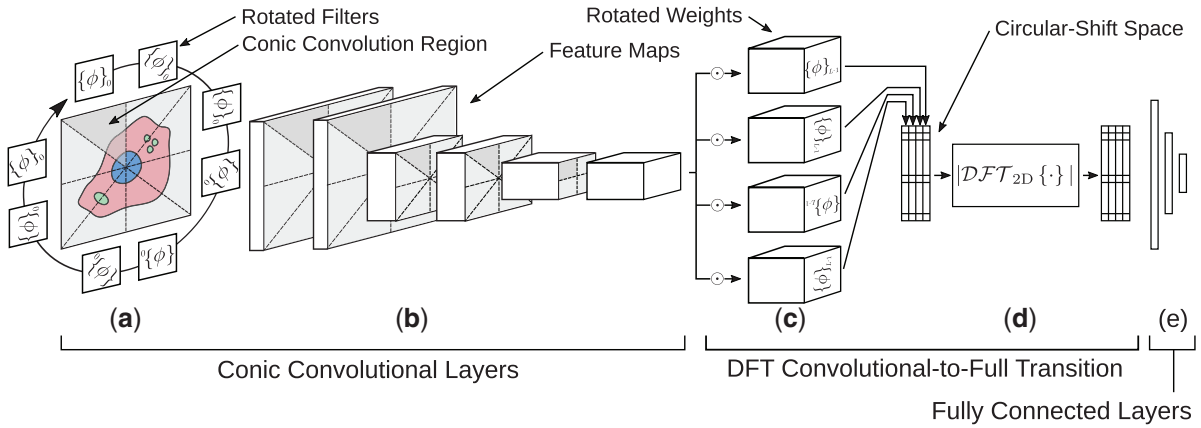

Fig. 2. The overall architecture of CFNet. (**a**) Filtering the image by various filters at rotations in corresponding conic regions preserves rotation-equivariance. (**b**) Subsequent convolutional feature maps are filtered similarly. Rotation-invariance is encoded by the transition from convolutional to fully-connected layers, which consists of (**c**) element-wise multiplication and sum, denoted by $\odot$, with rotated weight tensors, transforming rotation to circular shift and (**d**) application of the magnitude response of the 2D-DFT to encode invariance to such shifts. (**e**) This output is reshaped and passed through the final, fully-connected layers

We denote this group by G, where $g \in G$ is the transformation of rotation about the origin and a translation. The first group convolutional layer of a G-CNN operates on functions on $\mathbb{Z}^2$, over which the input image is defined, and is given by:

$$f * \phi(g) = \sum_{k=0}^{K-1} \sum_{z \in \mathbb{Z}^2} f_k(z) \phi_k(g^{-1}z). \tag{2}$$

Whereas in standard convolution, the filter is translated over the image and the inner product is computed at each translation, in group convolution, the filter is transformed by each element $g \in G$. The output of group convolution is then a function of the group G. Subsequent layers of the network must therefore operate on such functions, and group convolution for these layers is defined as:

$$f * \phi(g) = \sum_{k=0}^{K-1} \sum_{h \in G} f_k(h) \phi_k(g^{-1}h). \tag{3}$$

As shown in Cohen and Welling (2016), standard operations used in neural networks, including pooling, batch normalization and activations, can be defined on the feature maps of group convolution to preserve the equivariance property, and a full G-CNN can be defined by the composition of such operations. We refer the reader to Cohen and Welling (2016) for more details.

2.2 Rotation-equivariant quadrant convolutional layers

Rather than operating on functions on groups, conic convolution is simpler in that it maintains rotation equivariance while operating still on functions on the spatial domain $\mathbb{Z}^2$, as in standard convolution. We begin the formulation with a simpler, special case of conic convolution, which we call *quadrant convolution*. Its difference from standard convolution is that the filter being convolved is rotated by $\frac{r\pi}{2}$, $r \in \{0, 1, 2, 3\}$, depending upon the corresponding quadrant of the domain. We show that for quadrant convolution, rotations of $\frac{\pi}{2}$ of the input are straightforwardly associated with rotations of the output feature map, which is a special form of equivariance called *same-equivariance* [as coined by Dieleman *et al.* (2016)].

Relevant to our formulation is the group of two-dimensional rotation matrices of $\frac{\pi}{2}$, which we denote by G_1 and which can be easily

parameterized by $g(r)$, and which acts on points in $\mathbb{Z}^2$ by matrix multiplication, i.e. for a given point $x = (u, v) \in \mathbb{Z}^2$:

$$g(r)x = \begin{bmatrix} \cos\left(\frac{r\pi}{2}\right) & -\sin\left(\frac{r\pi}{2}\right) \\ \sin\left(\frac{r\pi}{2}\right) & \cos\left(\frac{r\pi}{2}\right) \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix}. \tag{4}$$

Let T_g denote the transformation of a function by a rotation in G_1, where $T_g f(x) \triangleq f(g^{-1}x)$ applies the inverse of g to an element of the domain of f. For an operation $\Phi : \mathcal{F} \to \mathcal{F}$, $\mathcal{F}$ being the set of K-dimensional functions f on $\mathbb{Z}^2$ (which represent feature maps), to exhibit same-equivariance, applying rotation either before or after the operation yields the same result, i.e.

$$T_g \Phi(f) = \Phi(T_g f). \tag{5}$$

Quadrant convolution can be interpreted as weighting the convolution for each rotation with a function $\omega : \mathbb{Z}^2 \to [0, 1]$ that simply 'selects' the appropriate quadrant of the domain. The weighting function for the first quadrant is defined as:

$$\omega(u, v) \triangleq \begin{cases} 1 & u > 0, v > 0, \\ \frac{1}{2} & u = 0 \oplus v = 0, \\ \frac{1}{4} & (u, v) = (0, 0), \\ 0 & \text{else.} \end{cases} \tag{6}$$

Since the origin does not strictly belong to a particular quadrant, it is handled by averaging the response of the filter at all four rotations. Boundary values are averaged over the responses of the neighboring regions. The appropriate weighting function for other quadrants is just a rotation of ω (i.e. $T_g\omega$) by the appropriate angle. The output of the layer is then given by:

$$\Phi(f) \triangleq \sum_{g \in G_1} [T_g \omega][[T_g \phi] * f]. \tag{7}$$

In our notation, parenthesis convey the parameter of a function, whereas square brackets merely clarify the order of operations. Example convolutional regions with appropriate filter rotations are shown in Figure 1.

Note that the equivariance property is established (see our detailed proof in the Supplementary Material) independent of the definition of ω, yet its definition will greatly influence the performance of the network. For example, if ω is simply the constant 1/4, it is equivalent to merely averaging the filter responses.

2.3 Generalization to conic convolutional layers

The above formulation can be generalized to *conic convolution* in which the rotation angle is decreased by an arbitrary factor of $\frac{\pi}{2R}$, for some positive integer R, instead of being fixed to $\frac{\pi}{2}$. Rather than considering quadrants of the domain, we can consider conic regions emanating from the origin and their boundaries, defined by:

$$\mathcal{C} = \left\{ (x, y) \in \mathbb{Z}^2_+ : 0 < \operatorname{arccot}(x/y) < \frac{\pi}{2R} \right\}, \tag{8}$$

$$\mathcal{B} = \left\{ (x, y) \in \mathbb{Z}^2_+ : \operatorname{arccot}(x/y) \in \left\{ 0, \frac{\pi}{2R} \right\} \right\}. \tag{9}$$

The weighting function is changed to have value one only over this conic region:

$$\omega_R(u, v) \triangleq \begin{cases} 1 & (u, v) \in \mathcal{C}, \\ \frac{1}{2} & (u, v) \in \mathcal{B}, \\ \frac{1}{4R} & (u, v) = (0, 0), \\ 0 & \text{else,} \end{cases} \tag{10}$$

of which $\omega_1 = \omega$ is a special case.

If we consider feature maps to be functions over the continuous domain $\mathbb{R}^2$ instead of $\mathbb{Z}^2$ and define the group G_R, with parameterization:

$$g_R(r)x = \begin{bmatrix} \cos\left(\frac{r\pi}{2R}\right) & -\sin\left(\frac{r\pi}{2R}\right) \\ \sin\left(\frac{r\pi}{2R}\right) & \cos\left(\frac{r\pi}{2R}\right) \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix}, \tag{11}$$

for $r \in \{0, 1, \ldots, 4R - 1\}$ and $x = (u, v) \in \mathbb{R}^2$, it is easy to show similarly as above that

$$\Phi_R(f) \triangleq \sum_{g \in G_R} [T_g \omega_R][[T_g \phi] * f] \tag{12}$$

is equivariant to G_R.

However, due to subsampling artifacts when discretizing $\mathbb{R}^2$ to $\mathbb{Z}^2$, as in an image, rotation equivariance for arbitrary values of R cannot be guaranteed and can only be approximated. In particular, the filters will have to be interpolated for rotations that are not a multiple of $\frac{\pi}{2}$. In our experiments when applying CFNet, we chose nearest neighbor interpolation, which preserves the energy of the filter under rotations. This defect notwithstanding, it can be shown that conic convolution maintains equivariance to rotations of $\frac{\pi}{2}$, and as we found in our experiments, the approximation of finer angles of rotation can still improve performance. Additionally, we note that R need not be the same for each layer, and it may be advantageous to use a finer discretization of rotations for early layers, when the feature maps are larger, and gradually decrease R.

A note must be made about subsequent nonlinear operations for a convolutional layer. It is typical in convolutional networks to perform subsampling, either by striding the convolution or by spatial pooling, to reduce the dimensionality of subsequent layers. Again, due to downsampling artifacts, rotational equivariance to rotations smaller than $\frac{\pi}{2}$ is not guaranteed. However, given that the indices of the plane of the feature map are in $\mathbb{Z}^2$ and are therefore centered about the origin, a downsampling of $D \in \mathbb{Z}_{>0}$ can be applied while maintaining rotational equivariance for rotations of $\frac{\pi}{2}$, regardless of the choice of R. After subsampling, the result is passed through a non-linear activation function $\sigma : \mathbb{R} \to \mathbb{R}$, such as ReLU, with an added offset $c_k \in \mathbb{R}$.

2.4 Computational efficiency of conic convolution

In CFNet, the response for each rotation in conic convolution is only needed over its corresponding conic region. However, since GPUs are more efficient operating on rectangular inputs, it is faster to compute the convolution over each quadrant in which the conic region resides. The output of conic convolution can be achieved by convolving over the corresponding quadrant, multiplying by the weighting function, summing the responses in each quadrant together, and then concatenating the responses of quadrants. For the special case of quadrant convolution, this process incurs negligible additional computation beyond standard convolution. Additionally, conic convolution produces only one feature map per filter as in standard convolution and therefore incurs no additional storage costs, in contrast to G-CNN and cyclic slicing, which both produce

Table 1. Test error on the rotated MNIST dataset

Algorithm	Test error (%)
Cohen and Welling (2016) (CNN)	5.03
Schmidt and Roth (2012)	3.98
Cohen and Welling (2016) (G-CNN)	2.28
G-CNN + DFT	2.00
CFNet	1.75

one map per rotation (Cohen and Welling, 2016; Dieleman *et al.*, 2016), and two for RotEqNet, one for the filter response and one for the orientation (Marcos *et al.*, 2017).

2.5 Rotation-invariant transition using the magnitude of the 2D-DFT

After the final convolutional layer of a CNN, some number of fully-connected layers will be applied to combine information from the various filter responses. In general, fully-connected layers will not maintain rotation equivariance or invariance properties. Commonly, convolution and downsampling are applied until the spatial dimensions are eliminated and the resulting feature map of the final convolutional layer is merely a vector, with dimension equal to the number of filters.

Rather than encoding invariance for each filter separately, as in most other recent works (Cohen and Welling, 2016; Weiler *et al.*, 2018), in CFNet we consider instead to transform the collective filter responses to a space in which rotation becomes circular shift so that the 2D-DFT can be applied to encode invariance. The primary advantage of the 2D-DFT as an invariant transform is that each output node is a function of every input node, and not just the nodes of a particular filter response, thereby capturing mutual information across responses.

Since the formulation of this transition involves the DFT, which is defined only for finite-length signals, we switch to represent feature maps as tensors, rather than functions. We denote the feature map generated by the penultimate convolutional layer by $f \in \mathbb{R}^{M\times M\times K}$, where $M \in \mathbb{Z}_{>1}$.

At the transition to fully-connected layers, the input f is passed through N fully-connected filters, $\phi^{(n)} \in \mathbb{R}^{M\times M\times K}$, $n \in \{0, 1, \ldots, N-1\}$. The operation of this layer can be interpreted as the inner product of the function and filter, $\langle \phi^{(n)}, f \rangle$. If we again consider rotations of the filter from the group G_R,

$$\Psi(n, r) \triangleq \langle T_{g_R(r)}\phi^{(n)}, f \rangle, \tag{13}$$

this is equivalent to the first layer of a G-CNN, mapping from the spatial domain to G_R (though this group does not include the translation group since the convolution is only applied at the origin), and rotations of the final convolutional layer f will correspond to permutations of G_R, which are just circular shifts in of the second dimension of the matrix Ψ.

The magnitude response of the 2D-DFT is applied to Ψ to transform these circular shifts to an invariant space:

$$|\mathcal{DFT}\{\Psi\}|(n, r) = \left| \sum_{n'=0}^{N-1} \sum_{r'=0}^{4R-1} \Psi(n', r') e^{-j2\pi\left(\frac{n'n}{N} + \frac{r'r}{4R}\right)} \right|. \tag{14}$$

This process of encoding rotation invariance corresponds to the 'Convolutional-to-Full Transition' in Figure 2. The result is then vectorized and passed into fully-connected layers that precede the final output layer, as in a standard CNN.

In addition, the 2D-DFT, as a rotation invariant transform, can also be integrated into other rotation-equivariant networks, such as G-CNN. At the final layer of a fully-convolutional G-CNN, since the spatial dimension has been eliminated through successive convolutions and spatial downsampling, rotation is encoded along contiguous stacks of feature maps $f \in \mathbb{R}^{N\times 4}$ of each filter at four rotations. In this way, rotations similarly correspond to circular shifts in the final dimension. This representation Ψ is then passed through the 2D-DFT, as in Eqn. 14.

3 Results

3.1 Application to rotated MNIST

We first used the rotated MNIST dataset (Larochelle *et al.*, 2007), which has been utilized as a benchmark for previous works on rotation invariance, to place CFNet against results previously reported for G-CNN. The model was trained on 10 000 images, using training augmentation of rotations of arbitrary angles as in (Cohen and Welling, 2016) (Though the paper (Cohen and Welling, 2016) did not state the use of training augmentation, code posted by the authors at https://github.com/tscohen/gconv_experiments indicates that rotations of arbitrary angles were used.), and the best model parameters were selected based on scores on a validation set of 5000 images. Our best CFNet architecture consisted of six conic convolution layers, with $R = 2$ for the first three and $R = 1$ for the next three, followed by the DFT transition and an output softmax layer of 10 nodes. Filters were three pixels in size, with 15 filters per layer, and spatial max-pooling was applied after the second layer. This architecture was similar in terms of number of layers and filters per layer as that of the G-CNN of (Cohen and Welling, 2016). As shown in Table 1, on a held-out set of 50 000 test images, CFNet achieved a 25% reduction in test error over G-CNN. To evaluate the G-CNN with the DFT, the only changes we made from the reported architecture for G-CNN was to reduce the number of filters for each layer to 7, to offset the addition of the 2D-DFT, which was applied to the output of the final convolutional layer. Incorporating the DFT transition into G-CNN further reduces the test error by 13%. These results demonstrate in a standard setting the value of incorporating mutual rotational information between filters, through the DFT, when encoding invariance and the added value of conic convolution.

3.2 Application to synthetic biomarker images

To precisely evaluate the advantage of encoding rotation equivariance, we created a set of synthetic microscopy images in which we could explicitly control the manifestation of rotations and intra- and inter-class variation. We utilized Gaussian-mixture models (GMMs), which have been used previously to emulate real-world fluorescence microscopy images of biological signals (Zhao and Murphy, 2007). Examples of synthetic images from across and within classes are shown in Figure 3a and b. Specifically, we defined 50 distribution patterns and generated 50 and 100 examples per class for training and 200 examples per class for testing. Each image consists of points sampled from several Gaussians, which have mean and variance defined by their particular class. Some intensity fluctuation, exponential noise and jitter are incorporated into the generating model to add variation. The image size was 50 pixels. A batch size of 50 examples, a learning rate of 5×10^{-3} and a weight decay ℓ_2 penalty of 5×10^{-4} were used during training. We used the Adam optimizer and decreased the

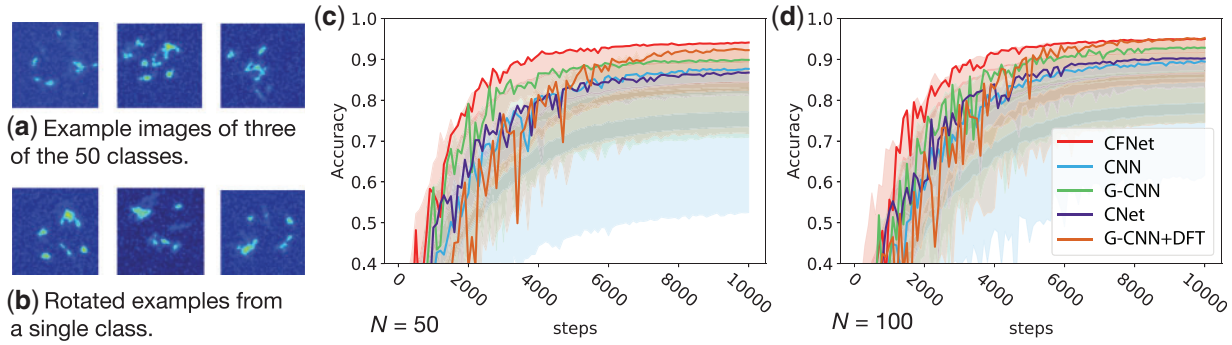

Fig. 3. Comparison of the results of CFNet, CNet (network with conic convolution but without the DFT), G-CNN, G-CNN+DFT and a standard CNN on the synthetic biomarker images. (**a, b**) Example images, shown as heat maps for detail, showing inter- and intra-class variation. The results with varying numbers *N* of training examples per class are in (**c, d**)

learning rate by 0.95 every few epochs. To help all methods, we augmented the training data by rotations and random jitter of up to three pixels, as was done during image generation. A more detailed description of the approach for generating the synthetic images is provided in the Supplementary Material.

Classification accuracies on the test dataset over training steps for various numbers of training samples, denoted by *N*, for several methods are shown in Figure 3c and d. A variety of configurations were trained for each network, and each configuration was trained three times. The darkest line shows the accuracy of the configuration that achieved the highest moving average, with a window size of 100 steps, for each method. The spread of each method, which is the area between the point-wise maximum and minimum of the error, is shaded with a light color, and three standard-deviations around the mean is shaded darker.

We observed a consistent trend of CFNet outperforming G-CNN, which in turn outperforms the CNN, both in overall accuracy and in terms of the number of steps required to attain that accuracy Figure 3c and d. Additionally, the spread of CFNet is mostly above even the best performing models of G-CNN and the CNN, demonstrating that an instance of CFNet will outperform other methods even if the best set of hyperparameters has not been chosen. We also included a network consisting of conic convolutional layers, but without the DFT, noted as 'CNet' (Fig. 3), to show the relative advantage of the DFT. CNet performs comparably to the standard CNN while requiring significantly less parameters to attain the same performance, though the true advantage of conic convolution is shown when integrated with the DFT to achieve global rotation invariance. In comparison, including the 2D-DFT increases the performance of G-CNN, to a comparable level with CFNet, though it does not train as quickly.

3.3 Application to subcellular protein localization images in budding yeast cells

To further demonstrate the advantage of rotation equivariant architectures and CFNet, we evaluated the models on real microscopy images of budding yeast cells generated from Kraus *et al.* (2017), which were collected as follow-up analysis of the data from Chong *et al.* (2015) and are more challenging, since they include more subclasses. In this dataset, cells were first modified by homologous recombination and SGA protocol to express fluorescent markers and GFP fusion query proteins. The cells were then transferred into 384-well plates and ten images (1338 × 1003 pixels) were taken per plate per channel. As shown in Figure 4, each image consists of a single or few cells and three stains, where blue shows the cytoplasmic region, pink the nuclear region and green the protein of interest. The classification for each image is the subcellular compartment in which the protein is localized and expressed, such as the cell periphery, mitochondria, or eisosomes, some of which exhibit very subtle differences. Our goal therefore is to predict the protein localization for a given image.

We compared the performance of CFNet with G-CNN and a standard CNN. Figure 4b and c shows the results of each method for classifying the protein localization for each image. To compare with DeepLoc (Kraus *et al.*, 2017), we used the same reported architecture and hyperparameters for the CNN. For CFNet and G-CNN, we removed the last convolutional layer and reduced the number of filters per layer by roughly half to offset for encoding of equivariance and invariance. The same training parameters and data augmentation were used as for the synthetic data, except that a dropout probability of 0.8 was applied at the final layer and the maximum jitter was increased to five pixels, since many examples were not well-centered. For each method, several iterations were run, and the spread and the best performing model is shown. We found that CFNet consistently outperforms G-CNN and the standard CNN representing DeepLoc, when the number of training examples per class is either 50 or 100 (see Fig. 4b and c), demonstrating that the gains of the 2D-DFT and conic convolution translate to real-world microscopy data. We note that the best reported algorithm that did not use deep learning, called ensLOC (Chong *et al.*, 2015; Koh *et al.*, 2015), was only able to achieve an average precision of 0.49 for a less challenging set of yeast phenotypes and with ∼20 000 samples, whereas all runs of CFNet achieved an average precision of between 0.60 and 0.67 with ∼10% of the data used for training.

We further analyzed the variation of performance for different protein localization labels (Fig. 4d). CFNet outperforms CNN on almost all classes. For instance, CFNet improves the accuracies on 'nuclear periphery', 'nucleolus', 'nucleus' and 'punctate nuclear' by 10, 14, 7 and 14%, respectively. Nucleolus and punctate nuclear are both structures inside the nucleus and their only difference is that punctate nuclear is generally smaller and rounder, which is rather subtle and CNN misassigns 13% of proteins that are in punctate nuclear with label 'nucleolus'. In contrast, CFNet decreases this misassignment to less than 5%. However, we also observed a few classes in particular for which CFNet could be further improved. For example, we found that CFNet tends to confuse the class 'bud' and 'budding periphery', likely because many proteins are present in both locations. Nevertheless, the application of CFNet to the

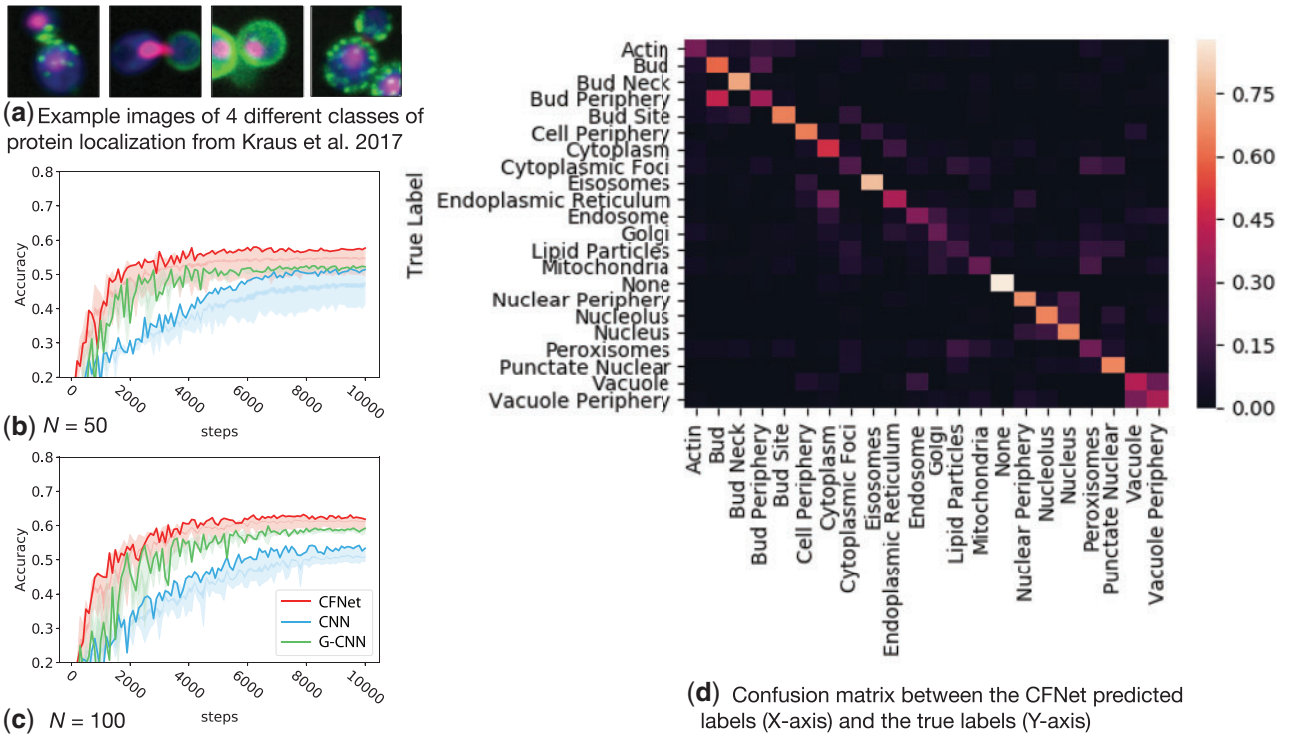

Fig. 4. Evaluation results based on subcellular protein localization images from Chong *et al.* (2015). (**a**) Example images. (**b–c**) Comparison results of CFNet, G-CNN and a standard CNN with varying numbers *N* of training examples per class. (**d**) Confusion matrix for the results for CFNet (X-axis) as compared to the true labels (Y-axis)

subcellular protein localization data demonstrates the effectiveness of the method.

One of the most significant advantages of CFNet, especially for biological knowledge discovery, is its interpretability. Figure 5 shows the activations of two particular filters from both CFNet and CNN at their third layer for example images of 'nucleus' and 'nuclear periphery' localizations, two classes that are challenging to differentiate. Since rotations of the input correspond directly to rotations of the output of conic convolution, as seen, the activations of the learned features do not change, except for rotating, thereby eliminating rotation as a confounding source of variation. It is important to note that even for rotations of 45 degrees, which conic convolution with $R = 2$ approximates, the activations are noticeably similar. Conversely, the activations for the standard CNN significantly change based upon the orientation of the image. This is especially apparent for the activation of filter 1 for the nucleus sample, which has a high response at the nucleus that splits in half under 90 degree rotation. We also observe that the activation of the CNN's filter 2 for the nuclear periphery sample only outlines the upper right boundary of the nucleus, since it is applied only at a specific orientation, whereas filter 1 of CFNet outlines the entire nucleus. The property of equivariance of conic convolution drastically enhances the ability to distinguish biological meaning of the learned representation from uninformative rotation.

4 Discussion

In this work, we explored the application of rotation equivariant and invariant neural networks to analyze cellular images. We have demonstrated the effectiveness of enforcing rotation equivariance and invariance in CNNs by means of the proposed conic convolutional layer and the 2D-DFT, even for group convolution. In addition, by applying our methods to a dataset of subcellular protein localizations, we showed that rotation equivariant models outperform the standard CNN and, in particular, CFNet with both conic convolutional layer and the 2D-DFT performs the best in our evaluations.

There are a few directions that we can further improve our models. For example, CFNet could be potentially further improved by incorporating steerable filters (Freeman and Adelson, 1991; Liu *et al.*, 2014) for convolution, as was done in (Weiler *et al.*, 2018), to enhance group-equivariant convolution and in Worrall *et al.* (2017), which allow for finer sampling of rotations of filters without inducing artifacts. Further evaluations would be needed to thoroughly assess these new approaches. Additionally, in the future, we intend to apply CFNet to full micrograph screens in a multiple-instance learning setting, as was done for CNNs in (Kraus *et al.*, 2016), since this is the setting with potentially more microscopy data and applications.

We believe that the proposed enhancements to the standard CNN will have much utility for future applications in many problem settings, in particular, high-throughput molecular and cellular imaging data, where training data is usually sparse, especially for rare cellular events. One of the most exciting frontiers in current biomedical research is to understand different cellular identities at single cell resolution, their functions and their compositions in different contexts, including various human tissues. With the datasets from large-scale projects such as the ongoing Human Cell Atlas (Rozenblatt-Rosen *et al.*, 2017) and the Human BioMolecular Atlas Program (HuBMAP) becoming available, our methods have the potential to complement existing approaches to more effectively analyze high-throughput cellular images.

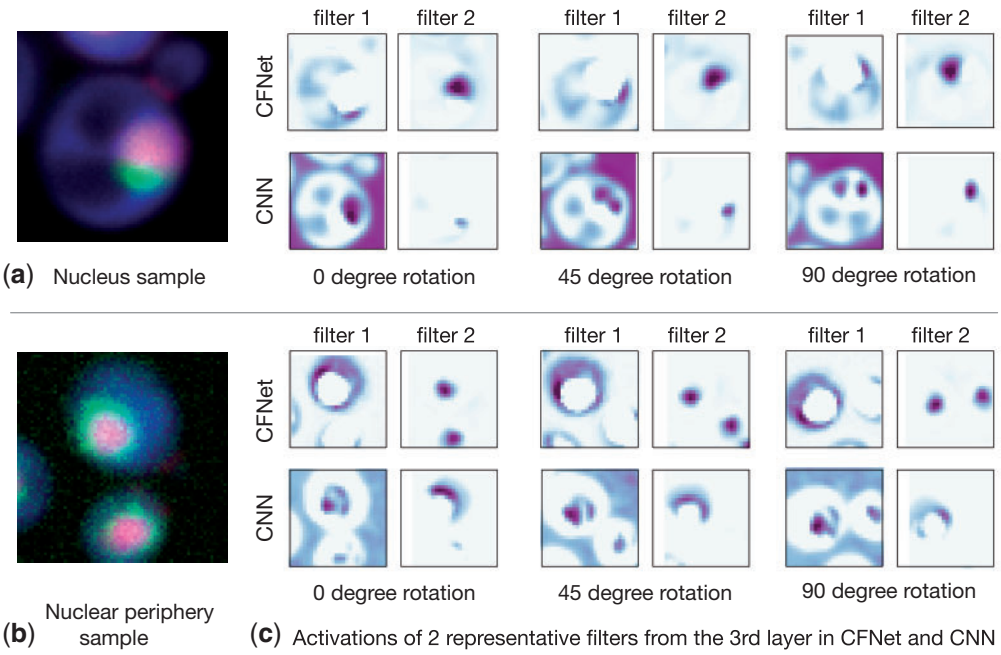

Fig. 5. Visualization of learned features of CFNet and CNN. Example images with protein localized (**a**) in the nucleus and (**b**) at the nuclear periphery. (**c**) Activations of two particular filters from the third layer in CFNet (top row) and CNN (bottom row) for each input rotated by 0, 45 and 90 degrees

Funding

This work was supported in part by the National Science Foundation grant 1717205 (J.M.).

Conflict of Interest: none declared.

References

Bekkers,E.J. *et al.* (2018) Roto-translation covariant convolutional networks for medical image analysis. In: *MICCAI, Granada, Spain*, pp. 440–448.

Boland,M.V. and Murphy,R.F. (2001) A neural network classifier capable of recognizing the patterns of all major subcellular structures in fluorescence microscope images of HeLa cells. *Bioinformatics*, **17**, 1213–1223.

Charalampidis,D. and Kasparis,T. (2002) Wavelet-based rotational invariant roughness features for texture classification and segmentation. *IEEE Trans. Image Process.*, **11**, 825–837.

Chong,Y.T. *et al.* (2015) Yeast proteome dynamics from single cell imaging and automated analysis. *Cell*, **161**, 1413–1424.

Cohen,T. and Welling,M. (2016) Group equivariant convolutional networks. In: *ICML, New York, NY, USA*, pp. 2990–2999.

Dai,J. *et al.* (2017) Deformable convolutional networks. In: *ICCV, Venice, Italy*, pp. 764–773.

Dieleman,S. *et al.* (2016) Exploiting cyclic symmetry in convolutional neural networks. In: *ICML, New York, NY, USA*, pp. 1889–1898. JMLR.org.

Do,M.N. and Vetterli,M. (2002) Rotation invariant texture characterization and retrieval using steerable wavelet-domain hidden Markov models. *IEEE Trans. Multimedia*, **4**, 517–527.

Freeman,W.T. and Adelson,E.H. (1991) The design and use of steerable filters. *IEEE Trans. Pattern Anal. Mach. Intell.*, **13**, 891–906.

Henriques,J.F. and Vedaldi,A. (2017) Warped convolutions: efficient invariance to spatial transformations. In: *ICML, Sydney, Australia*, pp. 1461–1469.

Jaderberg,M. *et al.* (2015) Spatial transformer networks. In: *NIPS, Montreal, Canada*, pp. 2017–2025.

Jafari-Khouzani,K. and Soltanian-Zadeh,H. (2005) Rotation-invariant multiresolution texture analysis using radon and wavelet transforms. *IEEE Trans. Image Process.*, **14**, 783–795.

Koh,J.L.Y. *et al.* (2015) Cyclops: a comprehensive database constructed from automated analysis of protein abundance and subcellular localization patterns in *Saccharomyces cerevisiae*. *G3 Genes Genomes Genet.*, **5**, 1223–1232.

Kraus,O.Z. *et al.* (2016) Classifying and segmenting microscopy images with deep multiple instance learning. *Bioinformatics*, **32**, i52–i59.

Kraus,O.Z. *et al.* (2017) Automated analysis of high-content microscopy data with deep learning. *Mol. Syst. Biol.*, **13**, 924.

Larochelle,H. *et al.* (2007) An empirical evaluation of deep architectures on problems with many factors of variation. In: *ICML, Corvalis, Oregon, USA*. ACM, pp. 473–480.

LeCun,Y. *et al.* (1989) Generalization and network design strategies. In: Pfeifer, *et al.* (eds) *Connectionism in Perspective, Zurich, Switzerland, Springer*, pp. 143–155.

Li,X. *et al.* (2018) Deeply supervised rotation equivariant network for lesion segmentation in dermoscopy images. In: *OR 2.0 Context-Aware Operating Theaters, Computer Assisted Robotic Endoscopy, Clinical Image-Based Procedures, and Skin Image Analysis*, Springer, pp. 235–243.

Liu,K. *et al.* (2014) Rotation-invariant hog descriptors using Fourier analysis in polar and spherical coordinates. *Int. J. Comput. Vis.*, **106**, 342–364.

Lundberg,E. and Borner,G.H. (2019) Spatial proteomics: a powerful discovery tool for cell biology. *Nat. Rev. Mol. Cell Biol.*, **20**, 285–302.

Marcos,D. *et al.* (2017) Rotation equivariant vector field networks. In: *ICCV, Venice, Italy*, pp. 5058–5067.

Ojala,T. *et al.* (2002) Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Trans. Pattern Anal. Mach. Intell.*, **24**, 971–987.

Rozenblatt-Rosen,O. *et al.* (2017) The human cell atlas: from vision to reality. *Nature*, **550**, 451.

Sabour,S. *et al.* (2017) Dynamic routing between capsules. In: Guyon,I. *et al.* (eds) *NIPS, Long Beach, CA, USA*. Curran Associates, Inc., pp. 3856–3866.

Schmidt,U. and Roth,S. (2012) Learning rotation-aware features: from invariant priors to equivariant descriptors. In: *CVPR, Providence, RI, USA*. IEEE, pp. 2050–2057.

Veeling,B.S. *et al.* (2018) Rotation equivariant CNNs for digital pathology. In: MICCAI, Granada, Spain, pp. 210–218.

Weiler,M. *et al.* (2018) Learning steerable filters for rotation equivariant CNNs. In: *CVPR, Salt Lake City, UT, USA*, pp. 849–858.

Worrall,D.E. *et al.* (2017) Harmonic networks: deep translation and rotation equivariance. In: *CVPR, Honolulu, HI, USA*, pp. 5028–5037.

Zhao,T. and Murphy,R.F. (2007) Automated learning of generative models for subcellular location: building blocks for systems biology. *Cytometry A*, **71**, 978–990.