Resolution-Invariant Image Classification based on Fourier Neural Operators*

Samira Kabri[1(✉)], Tim Roith[1], Daniel Tenbrinck[1], and Martin Burger[2,3]

[1] Friedrich-Alexander-Universität Erlangen-Nürnberg, 91058 Erlangen, Germany
[2] Deutsches Elektronen-Synchrotron, 22607 Hamburg, Germany
[3] Universität Hamburg, Fachbereich Mathematik, 20146 Hamburg, Germany
✉ samira.kabri@fau.de

Abstract. In this paper we investigate the use of Fourier Neural Operators (FNOs) for image classification in comparison to standard Convolutional Neural Networks (CNNs). Neural operators are a discretization-invariant generalization of neural networks to approximate operators between infinite dimensional function spaces. FNOs—which are neural operators with a specific parametrization—have been applied successfully in the context of parametric PDEs. We derive the FNO architecture as an example for continuous and Fréchet-differentiable neural operators on Lebesgue spaces. We further show how CNNs can be converted into FNOs and vice versa and propose an interpolation-equivariant adaptation of the architecture.

Keywords: neural operators · trigonometric interpolation · Fourier neural operators · convolutional neural networks · resolution invariance

1 Introduction

Neural networks, in particular CNNs, are a highly effective tool for image classification tasks. Substituting fully-connected layers by convolutional layers allows for efficient extraction of local features at different levels of detail with reasonably low complexity. However, neural networks in general are not resolution-invariant, meaning that they do not generalize well to unseen input resolutions. In addition to interpolation of inputs to the training resolution, various other approaches have been proposed to address this issue, see, e.g., [3,14,17]. In this work we focus on the interpretation of digital images as discretizations of functions. This allows to model the feature extractor as a mapping between infinite dimensional spaces with the help of so-called neural operators, see [13]. In Section 2, we use established results on Nemytskii operators to derive conditions for well-definedness, continuity, and Fréchet-differentiability of neural operators

* This work was supported by the European Union's Horizon 2020 programme, Marie Skłodowska-Curie grant agreement No. 777826. TR and MB acknowledge the support of the BMBF, grant agreement No. 05M2020. SK and MB acknowledge the support of the DFG, project BU 2327/19-1. This work was carried out while MB was with the FAU Erlangen-Nürnberg.

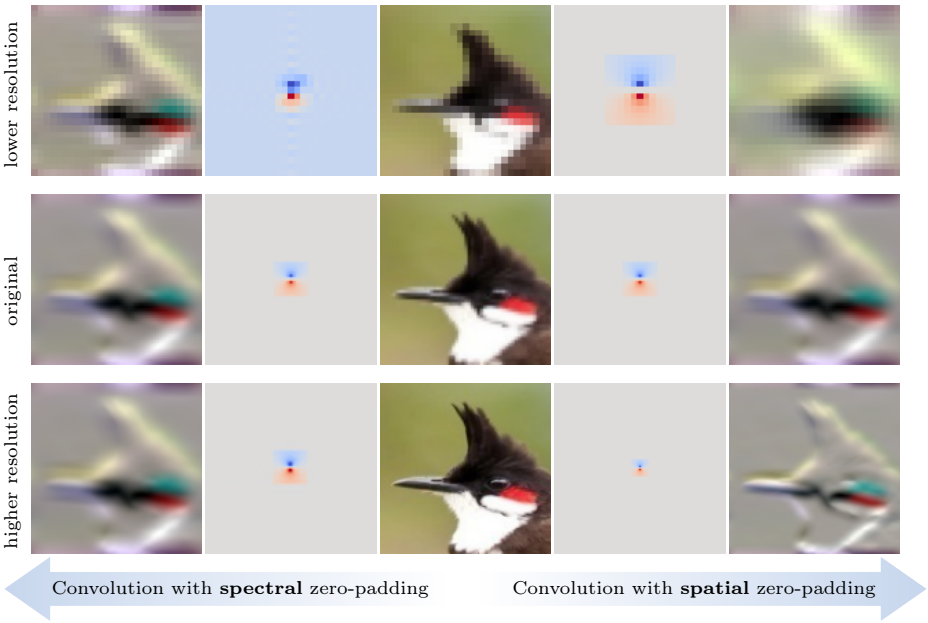


Fig. 1: Effects of applying a convolutional filter on the same image[4] with different resolutions. Spatial zero-padding (standard CNN-implementation) changes the relation of kernel support to image domain, while spectral zero-padding (FNO-implementation) captures comparable features for all resolutions.

on Lebesgue spaces. We specifically show these properties for the class of FNOs proposed in [15] as a discretization-invariant generalization of CNNs.

The key idea of FNOs is to parametrize convolutional kernels by their Fourier coefficients, i.e., in the spectral domain. Using trainable filters in the Fourier domain to represent convolution kernels in the context of image processing with neural networks has been studied with respect to performance and robustness in recent works, see e.g., [4,19,25]. In Section 3 we analyze the interchangeability of CNNs and FNOs with respect to optimization, parameter complexity, and generalization to varying input resolutions. While we restrict our theoretical derivations to real-valued functions, we note that they can be naturally extended to vector-valued functions as well. Our findings are supported by numerical experiments on the FashionMNIST [24] and Birds500 [18] data sets in Section 4.

2 Construction of Neural Operators on Lebesgue Spaces

2.1 Well-definedness and Continuity

A neural operator as defined in [13] is a composition of a finite but arbitrary number of so-called operator layers. In this section we derive conditions on the

[4] This image depicts a red whiskered bulbul taken from the Birds500 dataset [18].

components of an operator layer, such that it is a well-defined and continuous operator between two Lebesgue spaces. More precisely, for a bounded domain $\Omega \subset \mathbb{R}^d$ and $1 \leq p, q \leq +\infty$ we aim to construct a continuous operator $\mathcal{L} : L^p(\Omega) \to L^q(\Omega)$, such that an input function $u \in L^p(\Omega)$ is mapped to

$$\mathcal{L}(u)(x) = \sigma\left(\Psi(u)(x)\right) \qquad \text{for a.e. } x \in \Omega, \tag{1}$$

where we summarize all affine operations with an operator Ψ, such that

$$\Psi(u) = Wu + \mathcal{K}u + b. \tag{2}$$

Here, the weighting by $W \in \mathbb{R}$ implements a residual component and the kernel integral operator $\mathcal{K} : u \mapsto \int_\Omega \kappa(\cdot, y)\, u(y)\, dy$, determined by a kernel function $\kappa : \Omega \times \Omega \to \mathbb{R}$ generalizes the discrete weighting performed in neural networks. Analogously, the bias function $b : \Omega \to \mathbb{R}$ is the continuous counterpart of a bias vector. The (non-linear) activation function $\sigma : \mathbb{R} \to \mathbb{R}$ is applied pointwise and thus acts as a Nemytskii operator (see e.g., [6]). Thus, with a slight abuse of notation, the associated Nemytskii operator takes the form

$$\sigma : v \mapsto \sigma(v(\cdot)), \tag{3}$$

where we assume σ to be a measurable function. In order to ensure that the associated Nemytskii operator defines a mapping $\sigma : L^p(\Omega) \to L^q(\Omega)$ for $1 \leq p, q \leq \infty$ we require the following conditions to hold:

$$\begin{aligned} \underline{p, q < \infty} &: |\sigma(x)| \leq K + \beta |x|^{\frac{p}{q}} \text{ for all } x \in \mathbb{R} \text{ and constants } \beta, K \in \mathbb{R}, \\ \underline{p = \infty} &: |\sigma(x)| \leq K(c) \text{ for every } c > 0 \text{ for all } x, |x| < c \\ &\qquad \text{and a constant } K(c) \in \mathbb{R} \text{ depending on } c, \\ \underline{p < \infty, q = \infty} &: |\sigma(x)| \leq K \text{ for all } x \in \mathbb{R} \text{ and a constant } K \in \mathbb{R}, \end{aligned} \tag{4}$$

which were used in [6].

Lemma 1. *For $1 \leq p, q \leq \infty$ assume that σ fulfills (4). Then we have that the associated Nemytskii operator is a mapping $\sigma : L^p(\Omega) \to L^q(\Omega)$.*

Proof. Similar to [6, Th. 1], follows directly by employing the estimates in (4).

Since we are interested in continuity properties of the layer in (1) we consider the following continuity result for Nemytskii operators.

Lemma 2. *For $1 \leq p \leq \infty, 1 \leq q < \infty$ assume that the function $\sigma : \mathbb{R} \to \mathbb{R}$ is continuous and uniformly continuous in the case $q = \infty$. If the associated Nemytskii operator is a mapping $\sigma : L^p(\Omega) \to L^q(\Omega)$ then it is continuous.*

Proof. For $q < \infty$ the proof can be adapted from [23, p. 155-158]. For the case $q = \infty$ we refer to [6, Th. 5].

Remark 1. For $1 \leq p \leq q < \infty$ it is sufficient for σ to be p/q-Hölder continuous or locally Lipschitz continuous for $p, q = \infty$. In that case the Hölder and respectively the Lipschitz continuity transfers to the Nemytskii operator, see [22].

Example 1. The ReLU (Rectified Linear Unit, see [5]) function $\sigma(x) = \max(0, x)$ generates a continuous Nemytskii operator $\sigma : L^p(\Omega) \to L^q(\Omega)$ for any $p \geq q$. To show this, we note that the function σ is Lipschitz-continuous and with $p \geq q$ we have for all $x \in \mathbb{R}$ that $|\sigma(x)| \leq |x| \leq 1 + |x|^{\frac{p}{q}}$.

Proposition 1. *For $1 \leq p, q \leq \infty$ let $\mathcal{L}$ be an operator layer given by (1) with an activation function $\sigma : \mathbb{R} \to \mathbb{R}$. If there exists $r \geq 1$ such that*

(i) the affine part defines a mapping $\Psi : L^p(\Omega) \to L^r(\Omega)$,
(ii) the activation funtion σ generates a Nemytskii operator $\sigma : L^r(\Omega) \to L^q(\Omega)$,

then it holds that $\mathcal{L} : L^p(\Omega) \to L^q(\Omega)$. If additionally Ψ is a continuous operator on the specified spaces and the function σ is continuous, or uniformly continuous in the case $q = \infty$, the operator $\mathcal{L} : L^p(\Omega) \to L^q(\Omega)$ is also continuous.

Proof. With the assumptions on σ we directly have $\mathcal{L} = \sigma \circ \Psi : L^p(\Omega) \to L^q(\Omega)$. The continuity of $\mathcal{L}$ follows from Lemma 2.

Example 2. On the periodic domain $\Omega = \mathbb{R}/\mathbb{Z}$ consider an affine operator Ψ as defined in (2), where the integral operator is a convolution operator, i.e., $\kappa(x, y) = \kappa(x - y)$ with a slight abuse of notation. If for $1 \leq p, r, s \leq \infty$ we have that $\kappa \in L^s(\Omega)$ with $1/r + 1 = 1/p + 1/s$, it follows from Young's convolution inequality (see e.g., [8, Th. 1.2.12]) that $\mathcal{K} : L^p(\Omega) \to L^r(\Omega)$ is continuous. If further $b \in L^r(\Omega)$ and $W = 0$ in the case $r > p$, it follows directly that $\Psi : L^p(\Omega) \to L^r(\Omega)$ is continuous.

2.2 Differentiability

To analyze the differentiability of the neural operator layers we first transfer the result for general Nemytskii operators from [6, Th. 7] to our setting.

Theorem 1. *Let $1 \leq q < p < \infty$ or $q = p = \infty$ and $\sigma : \mathbb{R} \to \mathbb{R}$ a continuously differentiable function. Furthermore, let the Nemytskii operator associated to the derivative σ' be a continuous operator $\sigma' : L^p(\Omega) \to L^s(\Omega)$, with coefficient $s = pq/(p - q)$ for $q < p$ and $s = \infty$ for $q = p = \infty$. Then, the Nemytskii operator associated to σ is Fréchet-differentiable and its Fréchet-derivative $D\sigma(v) : L^p(\Omega) \to L^q(\Omega)$ in $v \in L^p(\Omega)$ is given by*

$$D\sigma(v)(h) = \sigma'(v) \cdot h, \qquad \text{for all } h \in L^p(\Omega).$$

Since the ReLU activation function from Example 1 is not differentiable, it does not fulfill the requirements of Theorem 1. An alternative is the so-called Gaussian Error Linear Unit (GELU), proposed in [10].

Example 3. The GELU function $\sigma(x) = x\,\Phi(x)$, where Φ denotes the cumulative distribution function of the standard normal distribution, generates a Fréchet-differentiable Nemytskii operator with derivative $D\sigma(v) : L^p(\Omega) \to L^q(\Omega)$ for any $p \geq q$ and $v \in L^p(\Omega)$. To show this, we compute $\sigma'(x) = \Phi(x) + x\phi(x)$, where $\phi(x) = \Phi'(x)$ is the standard normal distribution. We see that σ' is continuous and further $|\sigma'(x)| \leq 1 + |x|/\sqrt{2\pi} \leq 1 + 1/\sqrt{2\pi} + |x|^{\frac{p}{q}}/\sqrt{2\pi}$ for all $p \geq q$.

Proposition 2. *For $1 \leq p, q \leq \infty$, let $\mathcal{L}$ be an operator layer given by (1) with affine part Ψ as in (2). If there exists $r > q$, or $r = q = \infty$ such that*

(i) the affine part is a continuous operator $\Psi : L^p(\Omega) \to L^r(\Omega)$,
(ii) the activation function $\sigma : \mathbb{R} \to \mathbb{R}$ is continuously differentiable
(iii) and the derivative of the activation function generates a Nemytskii operator $\sigma' : L^r(\Omega) \to [L^r(\Omega) \to L^s(\Omega)]$ with $s = rq/(r-q)$,

then it holds that $\mathcal{L} : L^p(\Omega) \to L^q(\Omega)$ is Fréchet-differentiable in any $v \in L^p(\Omega)$ with Fréchet-derivative $D\mathcal{L}(v) : L^p(\Omega) \to L^q(\Omega)$

$$D\mathcal{L}(v)(h) = \sigma'(\Psi(v)) \cdot \tilde{\Psi}(h),$$

where $\tilde{\Psi}$ denotes the linear part of Ψ, i.e., $\tilde{\Psi} = \Psi - b$.

Proof. Theorem 1 yields that $D\sigma(v) : L^r(\Omega) \to L^q(\Omega)$ is well defined and continuous for $v \in L^r(\Omega)$. Fréchet-differentiability of linear and continuous operators on Banach spaces (see e.g., [1, Ex. 1.3]) yields the continuity of $\Psi : L^p(\Omega) \to L^r(\Omega)$ in all $v \in L^p(\Omega)$ with $D\Psi(v)(h) = \tilde{\Psi}(h)$. The claim follows from the chain-rule for Fréchet-differentiable operators, see [1, Prop. 1.4 (ii)].

For $p < q$ Fréchet-differentiability of a Nemytskii operator implies that the generating function is constant, and respectively affine linear for $p = q < \infty$, see [6, Ch 3.1]. Therefore, unless $p = \infty$, Fréchet-differentiability of neural operators with non-affine linear activation functions is only achieved at the cost of mapping the output of the affine part into a less regular space.

Example 4. For a continuous convolutional neural operator layer as constructed in Example 2, we consider a parametrization of the kernel function by a set of parameters $\hat{\theta} = \{\hat{\theta}_k\}_{k \in I} \subset \mathbb{C}$, where I is a finite set of indices, such that

$$\kappa_{\hat{\theta}}(x) = \sum_{k \in I} \hat{\theta}_k \, b_k(x), \tag{5}$$

with Fourier basis functions $b_k(x) = \exp(2\pi i \, kx)$ for $x \in \Omega$. Effectively, this amounts to parametrizing the kernel function by a finite number of Fourier coefficients. The resulting linear operator and the operator layer are denoted by $\Psi_{\hat{\theta}}$ and $\mathcal{L}_{\hat{\theta}}$. We note that FNOs proposed in [15] are neural operators that consist of such layers. It is easily seen that the kernel function defined by (5) is bounded and thus $\kappa_{\hat{\theta}} \in L^\infty(\Omega)$. Therefore, for suitable activation functions, Proposition 2 yields Fréchet-differentiability of $\mathcal{L}_{\hat{\theta}}$ with respect to its input function v, which was similarly observed in [16]. Additionally, for fixed v we consider the operator $\mathcal{L}_{(\cdot)}(v) : \hat{\theta} \mapsto \mathcal{L}_{\hat{\theta}}(v)$ which maps a set of parameters to a function. With the arguments from Proposition 2 we derive the partial Fréchet-derivatives of an FNO-layer with respect to its parameters for $h_k = (1 + i) \, e_k$ as

$$D_{\hat{\theta}_k} \mathcal{L}_{\hat{\theta}}(v) := D\mathcal{L}_{\hat{\theta}}(v)(h_k) = \sigma'(\Psi_{\hat{\theta}}(v)) \, D\Psi_{\hat{\theta}}(v)(h_k),$$

where e_k denotes the k-th canonical basis vector. Computing the Fréchet-derivative of Ψ in the sense of Wirtinger calculus ([20, Ch. 1]), this can be rewritten as $D_{\hat{\theta}_k} \mathcal{L}_{\hat{\theta}}(v) = \sigma'(\Psi_{\hat{\theta}}(v)) \, \bar{\hat{v}}_k \, \bar{b}_k$, where $\hat{v}_k$ denotes the k-th Fourier coefficient of v. Here, for a complex number $z \in \mathbb{C}$, we denote by $\bar{z}$ its complex conjugate.

3 Connections to Convolutional Neural Networks

In this section we analyze the connection between FNOs and CNNs. Thus, for the remainder of this work, we set the domain to be the d-dimensional torus, i.e., $\Omega = \mathbb{R}^d/\mathbb{Z}^d$. As described in Example 4, the main idea of FNOs is to parametrize the convolution kernel by a finite number of Fourier coefficients $\hat{\theta} = \{\hat{\theta}_k\}_{k\in I} \subset \mathbb{C}$, where $I \subset \mathbb{Z}^d$ is a finite set of indices. Making use of the convolution theorem, see e.g., [8, Prop. 3.1.2 (9)], the kernel integral operator can then be written as

$$\mathcal{K}_{\hat{\theta}} v = \mathcal{F}^{-1}\left(\hat{\theta} \cdot \mathcal{F} v\right), \tag{6}$$

where $\mathcal{F}\colon [\Omega \to \mathbb{C}] \to \left[\mathbb{Z}^d \to \mathbb{C}\right]$ denotes the Fourier transform on the torus (see e.g., [8, Ch. 3]) and $\cdot$ denotes elementwise multiplication in the sense that

$$\left(\hat{\theta} \cdot \mathcal{F} v\right)_k = \begin{cases} \hat{\theta}_k \ (\mathcal{F} v)_k & \text{for } k \in I, \\ 0 & \text{otherwise.} \end{cases} \tag{7}$$

We only consider parameters such that $\mathcal{K}$ maps real-valued functions to real-valued functions. This is equivalent to Hermitian symmetry, i.e., $\hat{\theta}_k = \overline{\hat{\theta}_{-k}}$ and in particular $\hat{\theta}_0 \in \mathbb{R}$. As proposed in [15], for $N \in \mathbb{N}$ we choose the set of multi-indices $I_N := \{-\lceil (N-1)/2 \rceil, \dots, 0, \dots, \lfloor (N-1)/2 \rfloor\}^d$, which corresponds to parametrizing the N lowest frequencies in each dimension. This is in accordance to the universal approximation result for FNOs derived in [12]. At this point, we assume N to be an odd number to avoid problems with the required symmetry and expand the approach to even choices of N in Section 3.3. Although an FNO is represented by a finite number of parameters, a discretization of (6) is needed to process discrete data, e.g., digital images. We therefore define the set of spatial multi-indices $J_N := \{0, \dots, N-1\}^d$ and write $v \in \mathbb{R}^{J_N}$ for mappings $v : J_N \to \mathbb{R}$. Furthermore, we discretize the Fourier transform for $v \in \mathbb{R}^{J_N}$ as

$$(Fv)_k = \frac{1}{\lambda} \sum_{j \in J_N} v_j \, e^{-2\pi i \left\langle k, \frac{j}{N} \right\rangle} \qquad \text{for all } k \in I_N$$

and its inverse for $\hat{v} \in \mathbb{C}^{I_N}$ as

$$\left(F^{-1}\hat{v}\right)_j = \frac{\lambda}{|J_N|} \sum_{k \in I_N} \hat{v}_k \, e^{2\pi i \left\langle k, \frac{j}{N} \right\rangle} \qquad \text{for all } j \in J_N,$$

where $\lambda \in \{1, \sqrt{|J_N|}, |J_N|\}$ determines the normalization factor. The discretized convolution operator parametrized by $\hat{\theta} \in \mathbb{C}^{I_N}_{\text{sym}} := F(\mathbb{R}^{J_N})$ is then defined by

$$K(\hat{\theta})(v) = F^{-1}\left(\hat{\theta} \cdot Fv\right) \qquad \text{for } v \in \mathbb{R}^{J_N}.$$

For the remainder of this work, we refer to the above implementation of convolution as the FNO-implementation. In the following we compare the FNO-implementation to the standard implementation of the convolution of θ and

$v \in \mathbb{R}^{J_N}$ in a conventional CNN, which can be expressed as

$$C(\theta)(v)_j = \sum_{\tilde{\jmath} \in J_N} \theta_{j-\tilde{\jmath}}\, v_{\tilde{\jmath}} \qquad \text{for all } j \in J_N.$$

For the sake of simplicity, we handle negative indices by assuming that the values can be perpetuated periodically, although this is usually not done in practice.

3.1 Extension to Higher Input-Dimensions by Zero-Padding

So far, the presented implementations of convolution require the dimensions of the parameters θ, or $\hat{\theta}$ and the input v to coincide. In accordance to (7), the authors of [15] propose to handle dimension mismatches by zero-padding of the spectral parameters. More precisely, a low-dimensional set of parameters $\hat{\theta} \in \mathbb{C}^{I_M}$ is adapted to an input $v \in \mathbb{R}^{J_N}$ with odd $N \in \mathbb{N}$ by setting

$$\hat{\theta}_k^{M \to N} = \begin{cases} \hat{\theta}_k & \text{for } k \in I_N \cap I_M, \\ 0 & \text{for } k \in I_N \backslash I_M. \end{cases}$$

Since we choose N to be odd, the required symmetry is not hurt by the above operation. The extended FNO-implementation of the convolution is then given for $\hat{\theta} \in \mathbb{C}^{I_M}$ and $v \in \mathbb{R}^{J_N}$ by

$$K(\hat{\theta})(v) := K(\hat{\theta}^{M \to N})(v).$$

Analogously, in the conventional CNN-implementation the convolution of parameters $\theta \in \mathbb{R}^{J_M}$ and $v \in \mathbb{R}^{J_N}$ with $N \geq M$ is computed as

$$C(\theta)(v) := C(\theta^{M \to N})(v),$$

where again, $\theta^{M \to N} \in \mathbb{R}^{J_M}$ denotes the zero-padded version of θ. We stress that, although the technique to generalize the implementations to higher input dimensions is the same, the outcome differs substantially. This was already mentioned in [13, Sec. 4] and is discussed further in Section 3.4.

3.2 Convertibility and Complexity

Deriving FNOs from convolutional neural operators using the convolution theorem suggests that there is a way to convert one implementation of convolution into the other as long as the input dimension is fixed. The following Lemma shows that this is indeed possible.

Lemma 3. *Let $M \leq N$ both be odd and let $T : \mathbb{R}^{J_N} \to \mathbb{C}^{I_N}$ be defined for $\theta \in \mathbb{R}^{J_N}$ as $T(\theta) = \lambda F(\theta)$. For any $\theta \in \mathbb{R}^{J_M}$ and $v \in \mathbb{R}^{J_N}$ it holds true that*

$$C(\theta)(v) = K(T(\theta^{M \to N}))(v)$$

and for any $\hat{\theta} \in \mathbb{C}^{I_M}_{sym}$ and $v \in \mathbb{R}^{J_N}$ it holds true that

$$K(\hat{\theta})(v) = C(T^{-1}(\hat{\theta}^{M \to N}))(v).$$

Proof. By the definition of the extension to higher input dimensions we can assume $M = N$. For $\theta, v \in \mathbb{R}^{J_N}$ we derive the discrete analogon of the convolution theorem by inserting the definitions of the discrete Fourier transform as

$$F(C(\theta)(v)) = \lambda\, F(\theta) \cdot F(v).$$

Employing that $F : \mathbb{R}^{J_N} \to \mathbb{C}^{I_N}_{\text{sym}}$ is a bijection, it follows that

$$C(\theta)(v) = F^{-1}(\lambda\, F(\theta)\, F(v)) = K(T(\theta))(v).$$

The second statement can be proven analogously. We note that T^{-1} is well-defined since $\lambda \geq 1$ for odd N.

Although the above Lemma proves convertibility for a fixed set of parameters and fixed input dimensions, a conversion can increase the amount of required parameters, as in general, the dimension of the converted parameters has to match the input dimension. It becomes clear that spatial locality cannot be enforced with the proposed FNO-parametrization and spectral locality cannot be enforced with the CNN-parametrization. Therefore, different behavior during the training process is to be expected if the parameter size does not match the input size. Moreover, the following Lemma shows that even for matching dimensions, equivalent behavior for gradient-based optimization like steepest descent requires careful adaptation of the learning rate, since the computation of gradients is not equivariant with respect to the function T.

Lemma 4. *For odd $N \in \mathbb{N}$ and $v, \theta \in \mathbb{R}^{J_N}$ and $\hat{\theta} = T(\theta)$ it holds true that*

$$\nabla_{\hat{\theta}} K(\hat{\theta})(v) = \frac{1}{|J_N|}\, T\left(\nabla_\theta C(\theta)(v)\right).$$

Proof. Inserting $\hat{\theta} = T(\theta)$ it follows with the chain rule from Lemma 3 that

$$\frac{\partial K(\hat{\theta})(v)_l}{\partial \hat{\theta}_k} = \sum_{j \in J_N} \frac{\partial C(\theta)(v)_l}{\partial \theta_j} \frac{\partial T^{-1}(\hat{\theta})_j}{\partial \hat{\theta}_k} = \frac{1}{|J_N|} \sum_{j \in J_N} \frac{\partial C(\theta)(v)_l}{\partial \theta_j}\, e^{-i2\pi\langle k, \frac{j}{N}\rangle}$$

for $k \in I_N$. The claim now follows by inserting the definition of T.

3.3 Adaptation to Even Dimensions

For the remainder of this paper we consider the special case $\Omega = \mathbb{R}^2/\mathbb{Z}^2$ and adapt the FNO-implementation to even dimensions. For odd dimensions M, N, zero-padding of a set of spectral coefficients does not violate the requirement $\hat{\theta}^{M \to N} \in \mathbb{C}^{I_N}_{\text{sym}}$. This property is lost in general for even dimensions. Since for odd dimensions, zero-padding in the spectral domain is equivalent to trigonometric interpolation, we perform the adaptation of dimensions such that $\hat{\theta}^{M \to N}$ is a trigonometric interpolator of a real-valued function (see [2] for an exhaustive study on this topic). In practice, this means splitting the coefficients corresponding to the Nyquist frequencies to interpolate from an even dimension to

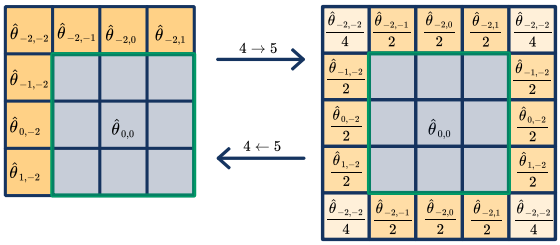


Fig. 2: Nyquist splitting for spectral parameters to extend real-valued trigonometric interpolation to even dimensions.

the next higher odd dimension, or to invert this splitting to interpolate from an odd dimension to the next lower even dimension (see Figure 2). The real-valued trigonometric interpolation of $v \in \mathbb{R}^{J_M}$ to a dimension N is then given by

$$v^{M \xrightarrow{\Delta} N} := F^{-1}\left((Fv)^{M \to N}\right).$$

We extend the FNO-implementation to parameters $\hat{\theta} \in \mathbb{C}^{I_N}_{\text{sym}}$ and inputs $v \in \mathbb{R}^{J_N}$ with even $N \in \mathbb{N}$ by defining

$$K(\hat{\theta})(v) := \left(K(\hat{\theta}^{N \to \tilde{N}})(v^{N \xrightarrow{\Delta} \tilde{N}})\right)^{\tilde{N} \xrightarrow{\Delta} N}, \tag{8}$$

where $\tilde{N} = N + 1$. We note that by this choice we lose the direct convertibility to the CNN-implementation as in general for even dimensions

$$K(\hat{\theta})(v) \neq F^{-1}(T^{-1}(\hat{\theta})\, Fv),$$

as the right hand side corresponds to zero-padding of the spectral coefficients regardless of the oddity of the dimensions. However, we can still convert the FNO-implementation to the CNN-implementation and vice versa, by adapting the magnitude of coefficients to the effects of the Nyquist splitting.

3.4 Interpolation Equivariance

Our motivation to perform the adaptation to even dimension as proposed in the preceding section, is that the resulting implementation of convolution is equivariant with respect to (real-valued) trigonometric interpolation.

Corollary 1. *For $\hat{\theta} \in \mathbb{C}^{I_M}_{sym}$, $v \in \mathbb{R}^{J_N}$, $M \leq N$ it holds true for any $L \geq M$ that*

$$K(\hat{\theta})(v^{N \xrightarrow{\Delta} L}) = \left(K(\hat{\theta})(v)\right)^{N \xrightarrow{\Delta} L}.$$

Proof. We first note that it holds for any choice of $M \leq N, L$ that

$$K(\hat{\theta})(v^{N \xrightarrow{\Delta} L}) = \left(K(\hat{\theta}^{M \to \tilde{M} \to \tilde{L}})(v^{N \xrightarrow{\Delta} \tilde{N} \xrightarrow{\Delta} \tilde{L}}) \right)^{\tilde{L} \xrightarrow{\Delta} L},$$

where $\tilde{L} = L + (1 - L\%2)$, $\tilde{M} = M + (1 - M\%2)$, $\tilde{N} = N + (1 - N\%2)$ and $\%$ denotes the modulo operation. Therefore, we can assume L, M and N to be odd without loss of generality and thus $\tilde{L} = L, \tilde{M} = M$ and $\tilde{N} = N$. Regarding the discrete Fourier coefficients then reveals that

$$F(K(\hat{\theta})(v^{N \xrightarrow{\Delta} L}))_k = \begin{Bmatrix} \hat{\theta}_k \, F(v)_k & \text{for } k \in I_M, \\ 0 & \text{otherwise} \end{Bmatrix} = F((K(\hat{\theta})(v))^{N \xrightarrow{\Delta} L})_k.$$

Applying the inverse Fourier transform completes the proof.

4 Numerical Examples

In this section we compare the discussed implementations of convolution numerically in the context of image classification.[5] Here, the task is to assign a label from $s \in \mathbb{N}$ possible classes to a given image $v : [0,1]^2 \to \mathbb{R}^{n_c}$, with $n_c \in \mathbb{N}$ denoting the number of color channels. Solving this task numerically requires discrete input images of the form $v^N = v|_{J_N/N} \in \mathbb{R}^{J_N \times n_c}$, where $N \in \mathbb{N}$ denotes the dimension. We note that since we consider a fixed function domain the dimension is proportional to the resolution. If we assume N to be fixed the network is a function $f_\theta : \mathbb{R}^{J_N \times n_c} \to \mathbb{R}^s$. Given a finite training set $D \subset \mathbb{R}^{J_N \times nc} \times \mathbb{R}^s$ we optimize the parameters θ by minimizing the empirical loss based on the cross-entropy [7, Ch. 3]. The networks we use for our experiments consist of several convolutional layers for feature extraction followed by one fully connected classification layer. To make all architectures applicable to inputs of any resolution, we insert an adaptive average pooling layer between the feature extractor and the classifier.

4.1 Expressivity for Varying Kernel Sizes

In the first experiment (see Fig. 3) we train a CNN without any residual components on the FashionMNIST[6] dataset. The network has two convolutional layers with periodic padding and without striding, followed by an adaptive pooling layer and a linear classifier. Since we do not observe major performance changes on the test set for different kernel sizes, we conclude that on this data set the expressivity of the small kernel architectures is comparable to large kernel architectures. We then convert the convolutional layers of the CNNs with 3×3- and 28×28-kernels to FNO-layers, employing varying numbers of spectral parameters. Here, we observe decreasing performance with smaller spectral kernel sizes,

[5] Our code is available online: github.com/samirak98/FourierImaging.

[6] This dataset consists of $60,000$ training and $10,000$ test 28×28 images (grayscale).

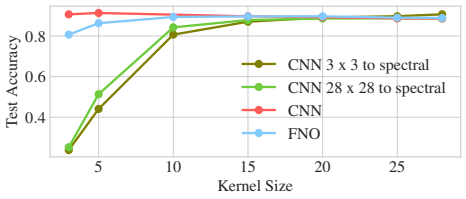


Fig. 3: Test accuracy of CNNs and FNOs for varying kernel sizes.

indicating that the learned spatial kernels cannot be expressed well by fewer frequencies. However, in this example, training an FNO with the same structure almost closes this performance gap. This implies the existence of low frequency kernels with sufficient expressivity. We refer to [11] for a study on training with a spectral parametrization.

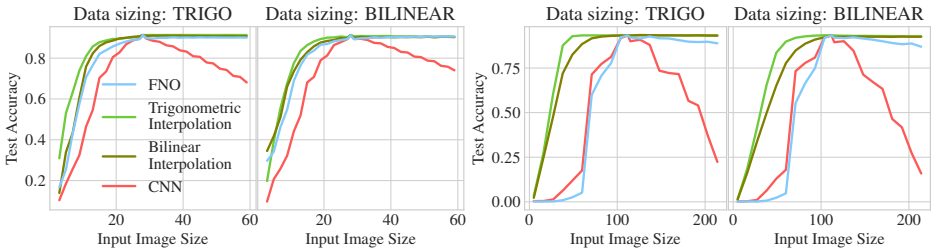
4.2 Resolution Invariance

In the second experiment, we investigate the resolution invariance of the different convolution implementations. In Fig. 4a we compare the accuracy on test data resized to different resolutions with trigonometric, or bilinear interpolation, respectively. Here, CNN refers to the conventional CNN-implementation with 5×5 kernel, where dimension mismatches are compensated for by spatial zero-padding of the kernel. FNO refers to the FNO-implementation, where the kernels are adapted to the input dimension by trigonometric interpolation. Additionally, we show the behavior of the CNN for inputs rescaled to the training resolution. Applying trigonometric interpolation before a convolutional layer can be interpreted as an FNO-layer with predetermined output dimensions.

The performance of the CNN varies drastically with the input dimension and peaks for the resolution it was trained on. This result is in accordance with the effect showcased in Fig. 1: Dimension adaption via spatial zero-padding modifies the locality of the kernel and consequently captures different features for different resolutions. While trigonometric interpolation performs best, we see that the FNO adapts very well. In particular, the performance for higher input resolutions deters only slightly, which is not the case for the standard CNN.

Additionally (see Fig. 4b), we train a ResNet18 [9] on the Birds500 data set[7] with a reduced training size of 112×112. To regularize the generalization to different resolutions, especially for the FNO-implementation, we replace the standard striding operations by trigonometric downsampling. Compared to the first experiment it stands out that the FNO performs worse for inputs with resolutions below 112×112, but only slightly diminishes for higher resolutions. We attribute this fact to the dimension reduction operations in the architecture.

[7] We employ a former version of the data set, which consists of $76,262$ RGB images for training and $2,250$ images for testing of size 224×224, where the task is to classify birds out of 450 possible classes.



(a) FMNIST (b) Birds500

Fig. 4: Performance for different interpolation methods on test data that has been resized with the interpolation method denoted on top of the plots.

5 Conclusion and Outlook

In this work, we have studied the regularity of neural operators on Lebesgue spaces and investigated the effects of implementing convolutional layers in the sense of FNOs. Based on the theoretical derivation of the convertibility from standard CNNs to FNOs, our numerical experiments show that it is possible to convert a network that was trained with the standard CNN architecture into an FNO. By this, we could combine the benefits of both approaches: Enforced spatial locality with a small number of parameters during training and an implementation that generalizes well to higher input dimensions during the evaluation. However, we have seen that the trigonometric interpolation of inputs outperforms all other considered approaches. In future work, we want to investigate how the ideas of FNOs and trigonometric interpolation can be incorporated into image-to-image architectures like U-Nets as proposed in [21]. Additionally, we want to further explore the effects of training in the spectral domain, for example with respect to adversarial robustness.

References

1. Ambrosetti, A., Prodi, G.: A Primer of Nonlinear Analysis. Cambridge University Press (1993)
2. Briand, T.: Trigonometric polynomial interpolation of images. Image Processing On Line **9**, 291–316 (10 2019)
3. Cai, D., Chen, K., Qian, Y., Kämäräinen, J.K.: Convolutional low-resolution fine-grained classification. Pattern Recognition Letters **119**, 166–171 (2019)
4. Chi, L., Jiang, B., Mu, Y.: Fast fourier convolution. Advances in Neural Information Processing Systems **33**, 4479–4488 (2020)
5. Fukushima, K.C.: Cognitron: A self-organizing multilayered neural network. Biol. Cybernetics **20**, 121–136 (1975)
6. Goldberg, H., Kampowsky, W., Tröltzsch, F.: On Nemytskij operators in lp-spaces of abstract functions. Mathematische Nachrichten **155**(1), 127–140 (1992)
7. Goodfellow, I., Bengio, Y., Courville, A.: Deep Learning. MIT Press (2016)

8. Grafakos, L.: Classical Fourier Analysis. Graduate Texts in Mathematics, Springer, New York, NY, 3 edn. (2014)
9. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE CVPR. pp. 770–778 (2016)
10. Hendrycks, D., Gimpel, K.: Gaussian error linear units (GELUs). arXiv:1606.08415 (2016)
11. Johnny, W., Brigido, H., Ladeira, M., Souza, J.C.F.: Fourier neural operator for image classification. In: 2022 17th Iberian Conference on Information Systems and Technologies (CISTI). pp. 1–6 (2022)
12. Kovachki, N.B., Lanthaler, S., Mishra, S.: On universal approximation and error bounds for fourier neural operators. Journal of Machine Learning Research (2022)
13. Kovachki, N.B., Li, Z., Liu, B., Azizzadenesheli, K., Bhattacharya, K., Stuart, A.M., Anandkumar, A.: Neural operator: Learning maps between function spaces. arXiv:2108.08481 (2021)
14. Koziarski, M., Cyganek, B.: Impact of low resolution on image recognition with deep neural networks: An experimental study. International Journal of Applied Mathematics and Computer Science **28**(4), 735–744 (2018)
15. Li, Z., Kovachki, N.B., Azizzadenesheli, K., Liu, B., Bhattacharya, K., Stuart, A.M., Anandkumar, A.: Fourier neural operator for parametric partial differential equations. In: 9th International Conference on Learning Representations (ICLR) (2021)
16. Li, Z., Zheng, H., Kovachki, N., Jin, D., Chen, H., Liu, B., Azizzadenesheli, K., Anandkumar, A.: Physics-informed neural operator for learning partial differential equations. arXiv preprint arXiv:2111.03794 (2021)
17. Peng, X., Hoffman, J., Stella, X.Y., Saenko, K.: Fine-to-coarse knowledge transfer for low-res image classification. In: 2016 IEEE International Conference on Image Processing (ICIP). pp. 3683–3687. IEEE (2016)
18. Piosenka, G.: Birds 500 - species image classification (2021), https://www.kaggle.com/datasets/gpiosenka/100-bird-species
19. Rao, Y., Zhao, W., Zhu, Z., Lu, J., Zhou, J.: Global filter networks for image classification. Advances in Neural Information Processing Systems **34**, 980–993 (2021)
20. Remmert, R.: Theory of Complex Functions. Springer New York, New York, NY (1991)
21. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: Navab, N., Hornegger, J., Wells, W.M., Frangi, A.F. (eds.) Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015. pp. 234–241. Springer International Publishing, Cham (2015)
22. Tröltzsch, F.: Optimal Control of Partial Differential Equations: Theory, Methods, and Applications, Graduate Studies in Mathematics, vol. 112. American Mathematical Society, Providence, Rhode Island (2010)
23. Vaĭnberg, M.M.: Variational method and method of monotone operators in the theory of nonlinear equations. No. 22090, John Wiley & Sons (1974)
24. Xiao, H., Rasul, K., Vollgraf, R.: Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. arXiv:1708.07747 (2017)
25. Zhou, M., Yu, H., Huang, J., Zhao, F., Gu, J., Loy, C.C., Meng, D., Li, C.: Deep fourier up-sampling. arxiv:2210.05171 (2022)