Analysis of convolutional neural network image classifiers in a rotationally symmetric model *

Michael Kohler† and Benjamin Walter†,‡
Fachbereich Mathematik, Technische Universität Darmstadt, Schlossgartenstr. 7, 64289 Darmstadt, Germany, email: kohler@mathematik.tu-darmstadt.de, bwalter@mathematik.tu-darmstadt.de

October 3, 2022

Abstract
Convolutional neural network image classifiers are defined and the rate of convergence of the misclassification risk of the estimates towards the optimal misclassification risk is analyzed. Here we consider images as random variables with values in some functional space, where we only observe discrete samples as function values on some finite grid. Under suitable structural and smoothness assumptions on the functional a posteriori probability, which includes some kind of symmetry against rotation of subparts of the input image, it is shown that least squares plug-in classifiers based on convolutional neural networks are able to circumvent the curse of dimensionality in binary image classification if we neglect a resolution-dependent error term. The finite sample size behavior of the classifier is analyzed by applying it to simulated and real data.

AMS classification: Primary 62G05; secondary 62G20.

Key words and phrases: Curse of dimensionality, convolutional neural networks, image classification, rate of convergence.

1 Introduction

In image classification, the task is to assign a given image to a class, where the class of the image depends on what kind of objects are represented on the image. For several years, the most successful methods in real-world applications are based on convolutional neural networks (CNNs), cf., e.g., He et al. (2016), Goodfellow, Bengio and Courville (2016), and Rawat and Wang (2017). For some image classification problems, it does not matter whether objects are rotated by arbitrary angles concerning a correct classification. This is the case, for example, in visual medical diagnosis applications, see, Veeling et al. (2018), or in galaxy morphology prediction, see, Dieleman, Willett and Dambre (2015), and further applications, see, e.g., Delchevalerie et al. (2021) and the

*Running title: *Rotationally symmetric image classification*

†Funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) - Projektnummer 449102119.

‡Corresponding author. Tel: +49-6 151-16-23386

literature cited therein. A large number of papers demonstrate the empirical success of increasing complex network architectures, especially for image classification tasks with rotated objects, many architectures try to exploit this symmetry, e.g., by some kind of invariance to rotation, see, e.g., Delchevalerie et al. (2021), Dieleman, Willett and Dambre (2015), and Cohen and Welling (2016). However, a theoretical justification for this empirical success exists only partially, see, Rawat and Wang (2017). The aim of this article is, on the one hand, to introduce a statistical setting for image classification that includes the irrelevance of rotation of objects by arbitrary angles, and, on the other hand, to derive in this setting a rate of convergence of image classifiers based on CNNs, which is independent of the dimension of the input image.

1.1 Image classification

In order to introduce the statistical setting for image classification, we describe idealized (random) images as $[0,1]$-valued functions on the cube

$$C_h = \left[-\frac{h}{2}, \frac{h}{2}\right] \times \left[-\frac{h}{2}, \frac{h}{2}\right] \subset \mathbb{R}^2$$

for $h > 0$. The function value at position $(i,j) \in C_h$ describes the corresponding gray scale value and the width h define the size of the image area. We denote the space of all gray scaled images of width h by

$$[0,1]^{C_h} \coloneqq \{f : C_h \to [0,1] \ : \ f \text{ is a mapping}\}.$$

Since the space of all $[0,1]$-valued functions on C_h equipped with the metric induced by the supremum norm $\|\cdot\|_\infty$ defines a metric space, we obtain a measurable space

$$\left([0,1]^{C_h}, \mathcal{B}\left([0,1]^{C_h}\right)\right)$$

$([0,1]^{C_h}, \mathcal{B}([0,1]^{C_h}))$ with the corresponding Borel σ-algebra. Next we introduce our statistical setting for image classification: Let (Φ, Y), (Φ_1, Y_1), $\ldots$, (Φ_n, Y_n) be independently and identically distributed random variables with values in $[0,1]^{C_1} \times \{0,1\}$. Here the (random) image Φ has the (random) class $Y \in \{0,1\}$. In practice, we can only observe discrete images consisting of a finite number of pixels. To obtain discrete observations from our idealized images, we evaluate them on a corresponding finite grid. To obtain a corresponding grid, we divide the cube C_1 into λ^2 equal sized cubes and choose the grid points as the centers of the small cubes. Formally, this means that we define the grid $G_\lambda \subset C_1$ with resolution $\lambda \in \mathbb{N}$ by

$$G_\lambda = \left\{ \left(\frac{i - \frac{1}{2}}{\lambda} - \frac{1}{2}, \frac{j - \frac{1}{2}}{\lambda} - \frac{1}{2} \right) \ : \ i,j \in \{1, \ldots, \lambda\} \right\}. \tag{1}$$

The corresponding (continuous) function $g_\lambda : [0,1]^{C_1} \to [0,1]^{G_\lambda}$, which evaluates a idealized continuous image on the grid G_λ, is defined by

$$g_\lambda(\phi) = \left(\phi\left(\mathbf{u}\right)\right)_{\mathbf{u} \in G_\lambda} \quad (\phi \in [0,1]^{C_1}),$$

where for $[0,1]^{G_\lambda}$ we use the notation

$$A^I = \{(a_i)_{i\in I} : a_i \in A \ (i \in I)\}$$

for a nonempty and finite index set I and some $A \subseteq \mathbb{R}$. Based on the observations

$$\mathcal{D}_n = \{(g_\lambda(\Phi_1), Y_1), ..., (g_\lambda(\Phi_n), Y_n)\},$$

we aim to construct a classifier $f_n = f_n(\cdot, \mathcal{D}_n) : [0,1]^{G_\lambda} \to \{0,1\}$ such that its misclassification risk $\mathbf{P}\{f_n(g_\lambda(\Phi)) \neq Y | \mathcal{D}_n\}$ is as small as possible. The misclassification risk is minimized by the so-called Bayes classifier, which is defined as

$$f^*(\mathbf{x}) = \begin{cases} 1 & \text{, if } \eta^{(\lambda)}(\mathbf{x}) > \frac{1}{2} \\ 0 & \text{, elsewhere,} \end{cases}$$

where $\eta^{(\lambda)}$ is the a posteriori probability of class 1 for discrete images of resolution λ given by

$$\eta^{(\lambda)}(\mathbf{x}) = \mathbf{P}\{Y = 1 | g_\lambda(\Phi) = \mathbf{x}\} \quad \left(\mathbf{x} \in [0,1]^{G_\lambda}\right). \tag{2}$$

Thus we have

$$\min_{f:[0,1]^{G_\lambda}\to[0,1]} \mathbf{P}\{f(g_\lambda(\Phi)) \neq Y\} = \mathbf{P}\{f^*(g_\lambda(\Phi)) \neq Y\}$$

(cf., e.g., Theorem 2.1 in Devroye, Györfi and Lugosi (1996)). Since the a posteriori probability (2) is unknown in general we evaluate the statistical performance of our classifier f_n by deriving an upper bound on the expected misclassification risk of our classifier and the optimal misclassification risk, i.e. we want to derive an upper bound on

$$\begin{aligned} &\mathbf{E}\left\{\mathbf{P}\{f_n(g_\lambda(\Phi)) \neq Y | \mathcal{D}_n\} - \min_{f:[0,1]^{G_\lambda}\to[0,1]} \mathbf{P}\{f(g_\lambda(\Phi)) \neq Y\}\right\} \\ &= \mathbf{P}\{f_n(g_\lambda(\Phi)) \neq Y\} - \mathbf{P}\{f^*(g_\lambda(\Phi)) \neq Y\}. \end{aligned} \tag{3}$$

Here we use so-called plug-in classifiers, which are defined by

$$f_n(\mathbf{x}) = \begin{cases} 1 & \text{, if } \eta_n(\mathbf{x}) \geq \frac{1}{2} \\ 0 & \text{, elsewhere,} \end{cases}$$

where $\eta_n(\cdot) = \eta_n(\cdot, \mathcal{D}_n) : [0,1]^{G_\lambda} \to \mathbb{R}$ is an estimate of the a posteriori porbability (2). To derive nontrivial rates of convergence for (3), it is necessary to restrict the class of distributions of $(g_\lambda(\Phi), Y)$ (cf., Cover (1968) and Devroye (1982)). For this purpose, in Kohler, Krzyżak and Walter (2022) they have introduced the hierarchical max-pooling model for the a posteriori probability of class 1 for discrete images (2) (see Definition 1 in Section 2). A drawback of this model, which is also used in Kohler and Langer (2020) and in a generalized form in Walter (2021), is that it does not include some kind of symmetry against rotation of subparts of the input image.

1.2 Main results

In this article we introdue a new model for the functional a posteriori probability

$$\eta(\phi) = \mathbf{P}\{Y = 1|\Phi = \phi\} \quad \left(\phi \in [0,1]^{C_1}\right) \tag{4}$$

for continuous images. The model is based on the following three structural assumptions, which seem plausible for some image classification problems:

1. The crucial object is contained in a subpart of the image.
2. The rotation of objects by arbitrary angels is irrelevant concerning a correct classification.
3. An image is hierarchically composed of adjacent subparts.

These assumptions then lead to a model in which the a posteriori probability (4) is computed by the supremum over all possible rotated subparts of a fixed width h of the input image ϕ, inserting them into a function $f : [0,1]^{C_h} \to [0,1]$ that satisfies appropriate compository assumptions. Besides the compository assumptions, this function also satisfies smoothness assumptions, which depend on a smoothness parameter p. Due to the second point above, objects of one class can appear in differently rotated positions, which makes this new model more difficult to capture by a CNN than the discrete model from Kohler, Krzyżak and Walter (2022).

Assuming the new model for the functional a posteriori probability (4), we show that least-squares plug-in CNN image classifiers (with ReLU activation function) achieve an upper bound on the expected difference of the misclassification risk of the classifier and the optimal misclassification risk (3) of

$$\sqrt{\log(\lambda) \cdot (\log n)^4 \cdot n^{-\frac{2 \cdot p}{2 \cdot p + 4}} + \epsilon_\lambda}$$

(up to some constant factor), where ϵ_λ is an error term depending on the image resolution. Thus, if we neglect the resolution-depending error term ϵ_λ our CNN image classifiers are able to circumvent the curse of dimensionality assuming the new model for the functional a posteriori probability (4).

In the proof we use standard bounds of empirical process theory (cf., Lemma 6 in the supplement) to decompose our error (3) into an approximation error and the model complexity. The main technical novelty is then an approximation result for the functional a posteriori probability by convolutional neural networks (see Lemma 1). For model complexity, we use a slight modification of Lemma 4 from Kohler, Krzyżak and Walter (2022) that provides an upper bound on the covering number of convolutional neural networks.

1.3 Discussion of related results

A statistical theory for image classification using CNNs (with ReLU activation function) is considered in Kohler, Krzyżak and Walter (2022), Walter (2021), and Kohler and

Langer (2020). Kohler, Krzyżak and Walter (2022) and Walter (2021) study plug-in CNN image classifiers learned by minimizing the squares loss, assuming generalizations of the hierarchical max-pooling model (see Definition 1) for the a posteriori probability of class 1. The model in Kohler, Krzyżak and Walter (2022) consists of several hierarchical max-pooling models and the model in Walter (2021) generalizes the hierarchical max-pooling model in the sense that the relative distances of hierarchically combined subparts are variable. In Kohler and Langer (2020), the hierarchical max-pooling model from Definition 1 is considered, where the CNN image classifiers minimize the cross-entropy loss. All three papers achieve a rate of convergence that is independent of the input image dimension. The statistical performance of CNNs for classification problems where the data is assumed to have a low-dimensional geometric structure is studied in Liu et al. (2021). Here as well, a dimension reduction is achieved while residual convolutional neural network architectures are used, i.e., convolutional neural networks containing skip layer connections. Lin and Zhang (2019) obtained generalization bounds for CNN architectures in a setting of multiclass classification. Classification problems using standard deep feedforward neural networks were analyzed in Kim, Ohn and Kim (2021), Bos and Schmidt-Hieber (2021) and Hu, Shang and Cheng (2020).

Much more theoretical results exist in the context of regression estimation. Oono and Suzuki (2019) use a similar residual CNN network architecture as Liu et al. (2021) and obtain estimation error rates that are optimal in the minimax sense. While they show that application-preferred architectures (especially in image classification applications) perform as well as standard feedforward neural networks, they do not identify situations in which CNN architectures outperform standard feedforward neural networks. For standard deep feedforward neural networks, rate of convergence results with dimension reduction could be shown under the assumption that the regression function is a hierarchical composition of functions of small input dimension (cf., Kohler and Krzyżak (2017), Bauer and Kohler (2019), Schmidt-Hieber (2020), Kohler and Langer (2021), Suzuki and Nitanda (2019) and Langer (2021)). Kohler, Krzyzak and Langer (2019) have shown that in case where the regression function has a low local dimensionality, sparse neural network estimates achieve a dimension reduction. Imaizumi and Fukamizu (2019) obained generalization error rates for the estimation of regression functions with partitions having rather general smooth boundaries by neural networks.

Approximation results for CNNs were obtained by Zhou (2020), Petersen and Voigtlaender (2020) and Yarotsky (2018). That the gradient descent finds the global minimum of the empirical risk with squares loss is shown for CNN architectures, e.g., in Du et al. (2018). The networks used here are overparameterized. In Kohler and Krzyżak (2021), it was shown that overparametrized deep neural networks minimizing the empirical L_2 risk do not, in general, generalize well.

1.4 Notation

Throughout the paper, the following notation is used: The sets of natural numbers, natural numbers including zero, integers and real numbers are denoted by $\mathbb{N}$, $\mathbb{N}_0$, $\mathbb{Z}$ and

$\mathbb{R}$, respectively. For $\mathbf{x} = (x_1, \ldots, x_d) \in \mathbb{R}^d$ we denote the maximum norm by

$$\|\mathbf{x}\|_\infty = \max(|x_1|, \ldots, |x_d|),$$

and for $f : \mathbb{R}^d \to \mathbb{R}$

$$\|f\|_\infty = \sup_{\mathbf{x} \in \mathbb{R}^d} |f(\mathbf{x})|$$

is its supremum norm, and the supremum norm of f on a set $A \subseteq \mathbb{R}^d$ is denoted by

$$\|f\|_{A,\infty} = \sup_{\mathbf{x} \in A} |f(\mathbf{x})|.$$

Let $p = q + s$ for some $q \in \mathbb{N}_0$ and $0 < s \leq 1$. A function $f : \mathbb{R}^d \to \mathbb{R}$ is called (p, C)-smooth, if for every $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_d) \in \mathbb{N}_0^d$ with $\sum_{j=1}^d \alpha_j = q$ the partial derivative $\frac{\partial^q f}{\partial x_1^{\alpha_1} \ldots \partial x_d^{\alpha_d}}$ exists and satisfies

$$\left| \frac{\partial^q f}{\partial x_1^{\alpha_1} \ldots \partial x_d^{\alpha_d}}(\mathbf{x}) - \frac{\partial^q f}{\partial x_1^{\alpha_1} \ldots \partial x_d^{\alpha_d}}(\mathbf{z}) \right| \leq C \cdot \|\mathbf{x} - \mathbf{z}\|^s$$

for all $\mathbf{x}, \mathbf{z} \in \mathbb{R}^d$. Notice that in the case of $p \leq 1$ a function (p, C)-is smooth if and only if it is Hölder continuous with exponent p and Hölder constant C.

Let $\mathcal{F}$ be a set of functions $f : \mathbb{R}^d \to \mathbb{R}$, let $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathbb{R}^d$ and set $\mathbf{x}_1^n = (\mathbf{x}_1, \ldots, \mathbf{x}_n)$. A finite collection $f_1, \ldots, f_N : \mathbb{R}^d \to \mathbb{R}$ is called an ε– cover of $\mathcal{F}$ on $\mathbf{x}_1^n$ if for any $f \in \mathcal{F}$ there exists $i \in \{1, \ldots, N\}$ such that

$$\frac{1}{n} \sum_{k=1}^n |f(\mathbf{x}_k) - f_i(\mathbf{x}_k)| < \varepsilon.$$

The ε–covering number of $\mathcal{F}$ on $\mathbf{x}_1^n$ is the size N of the smallest ε–cover of $\mathcal{F}$ on $\mathbf{x}_1^n$ and is denoted by $\mathcal{N}_1(\varepsilon, \mathcal{F}, \mathbf{x}_1^n)$.

For $z \in \mathbb{R}$ and $\beta > 0$ we define $T_\beta z = \max\{-\beta, \min\{\beta, z\}\}$. If $f : \mathbb{R}^d \to \mathbb{R}$ is a function and $\mathcal{F}$ is a set of such functions, then we set

$$(T_\beta f)(\mathbf{x}) = T_\beta\left(f(\mathbf{x})\right) \quad \text{and} \quad T_\beta \mathcal{F} = \left\{T_\beta f \quad : \quad f \in \mathcal{F}\right\}.$$

Let I be a nonempty and finite index set. For $\mathbf{x} \in \mathbb{R}^d$ we use the notation $\mathbf{x}_I = (x_i)_{i \in I}$ and for $M \subset \mathbb{R}^d$ we define $\mathbf{x} + M = \{\mathbf{x} + \mathbf{z} : \mathbf{z} \in M\}$.

1.5 Outline of the paper

In Section 2 the new model for the functional a posteriori probability is introduced and the CNN image classifiers used in this paper are defined in Section 3. The main result is presented in Section 4 and proven in Section 6. In Section 5 the finite sample size behavior of our classifier is analyzed by applying it to simulated and real data.

2 A rotationally symmetric hierarchical max-pooling model for the functional a posteriori probability

Before we introduce the new model for the functional a posteriori probability (4), we present the hierarchical max-pooling model from Kohler, Krzyżak and Walter (2022) for the a posteriori probability for discrete images (2). Here a (random) image is directly defined as a $[0,1]^{\{1,\dots,d_1\}\times\{1,\dots,d_2\}}$-valued random variable for some image dimensions $d_1, d_2 \in \mathbb{N}$. In the hierarchical max-pooling model, the following two main ideas are used: The first idea is that the class of an image is determined by whether the image contains an object that is contained in a subpart of the image. The approach is then to estimate for all subparts of the image whether they contain the corresponding object or not. The probability that the image contains the object is then assumed to be the maximum of the probabilities of all subparts (see Definition 1 a)). The second idea is that the probabilities for the individual subparts are composed hierarchically by combining decisions from smaller subparts (see Definition 1 b)).

Definition 1 *Let $d_1, d_2 \in \mathbb{N}$ with $d_1, d_2 > 1$ and $m : [0,1]^{\{1,\dots,d_1\}\times\{1,\dots,d_2\}} \to \mathbb{R}$.*
a) *We say that m satisfies a* **max-pooling model with index set**

$$I \subseteq \{0,\dots,d_1-1\} \times \{0,\dots,d_2-1\},$$

if there exists a function $f : [0,1]^{(1,1)+I} \to \mathbb{R}$ such that

$$m(\mathbf{x}) = \max_{(i,j)\in\mathbb{Z}^2 \,:\, (i,j)+I \subseteq \{1,\dots,d_1\}\times\{1,\dots,d_2\}} f\left(x_{(i,j)+I}\right) \quad (x \in [0,1]^{\{1,\dots,d_1\}\times\{1,\dots,d_2\}}).$$

b) *Let $I = \{0,\dots,2^l-1\} \times \{0,\dots,2^l-1\}$ for some $l \in \mathbb{N}$. We say that*

$$f : [0,1]^{\{1,\dots,2^l\}\times\{1,\dots,2^l\}} \to \mathbb{R}$$

satisfies a **hierarchical model of level** *l, if there exists functions*

$$g_{k,s} : \mathbb{R}^4 \to [0,1] \quad (k = 1,\dots,l, s = 1,\dots,4^{l-k})$$

such that we have

$$f = f_{l,1}$$

for some $f_{k,s} : [0,1]^{\{1,\dots,2^k\}\times\{1,\dots,2^k\}} \to \mathbb{R}$ recursively defined by

$$\begin{aligned} f_{k,s}(\mathbf{x}) \;=\; & g_{k,s}\big(f_{k-1,4\cdot(s-1)+1}(x_{\{1,\dots,2^{k-1}\}\times\{1,\dots,2^{k-1}\}}), \\ & \quad f_{k-1,4\cdot(s-1)+2}(x_{\{2^{k-1}+1,\dots,2^k\}\times\{1,\dots,2^{k-1}\}}), \\ & \quad f_{k-1,4\cdot(s-1)+3}(x_{\{1,\dots,2^{k-1}\}\times\{2^{k-1}+1,\dots,2^k\}}), \\ & \quad f_{k-1,4\cdot s}(x_{\{2^{k-1}+1,\dots,2^k\}\times\{2^{k-1}+1,\dots,2^k\}})\big) \\ & \qquad \left(x \in [0,1]^{\{1,\dots,2^k\}\times\{1,\dots,2^k\}}\right) \end{aligned}$$

for $k = 2, \ldots, l, s = 1, \ldots, 4^{l-k}$, *and*

$$f_{1,s}(x_{1,1}, x_{1,2}, x_{2,1}, x_{2,2}) = g_{1,s}(x_{1,1}, x_{1,2}, x_{2,1}, x_{2,2}) \quad (x_{1,1}, x_{1,2}, x_{2,1}, x_{2,2} \in [0,1])$$

for $s = 1, \ldots, 4^{l-1}$.

c) *We say that* $m : [0,1]^{\{1,\ldots,d_1\} \times \{1,\ldots,d_2\}} \to \mathbb{R}$ *satisfies a* **hierarchical max-pooling model of level** l *(where* $2^l \leq \min\{d_1, d_2\}$*), if* m *satisfies a max-pooling model with index set*

$$I = \{0, \ldots, 2^l - 1\} \times \{0, \ldots 2^l - 1\}$$

and the function $f : [0,1]^{\{1,\ldots,2^l\} \times \{1,\ldots,2^l\}} \to \mathbb{R}$ *in the definition of this max-pooling model satisfies a hierarchical model with level* l.

In addition to these structural assumptions on the a posteriori probability, Kohler, Krzyżak and Walter (2022) also assume that the functions $g_{k,s}$ of the hierarchical model are (p, C)-smooth (for the definition of (p, C)-smoothness, see Section 1.4). We aim to extend the above model so that it becomes more realistic for practical applications of image classification. We do this by introducing a model for the functional a posteriori probability $\eta(\phi) = \mathbf{P}\{Y = 1 | \Phi = \phi\}$. Here we are able to introduce some kind of symmetry against rotation of subparts of the input image. In order to rotate a subpart of an image, we define the function $rot^{(\alpha)} : \mathbb{R}^2 \to \mathbb{R}^2$ given by

$$rot^{(\alpha)}(\mathbf{x}) = \begin{pmatrix} \cos(\alpha) & -\sin(\alpha) \\ \sin(\alpha) & \cos(\alpha) \end{pmatrix} \cdot \mathbf{x} \quad (\mathbf{x} \in \mathbb{R}^2)$$

which rotates its input through an angle $\alpha \in [0, 2\pi]$ about the origin $\mathbf{0} \in \mathbb{R}^2$. Furthermore, we define the translation function $\tau_{\mathbf{v}} : \mathbb{R}^2 \to \mathbb{R}^2$ with translation vector $\mathbf{v} \in \mathbb{R}^2$ by

$$\tau_{\mathbf{v}}(\mathbf{x}) = \mathbf{x} + \mathbf{v} \quad (\mathbf{x} \in \mathbb{R}^2).$$

Besides the ideas of the hierarchical max-pooling model from Kohler, Krzyżak and Walter (2022), we want to integrate the following idea into our model: We consider an image classification problem, where rotated objects correspond to each other, i.e., when asking whether an image contains a particular object, it does not matter for the correct classification whether the corresponding object is shown in some rotated position (cf., Figure 1). We solve this problem by assuming that there is a function into which we

Figure 1: All three images are assigned to the class 'dog'.

can insert differently rotated subparts of an image (this function corresponds to the

function $f : [0,1]^{C_h} \to [0,1]$ in part a) of the definition below). For a given subpart, the function estimates the probability whether the subpart contains a specific non-rotated object. We then estimate the probability whether a subpart contains the object rotated by an arbitrary angle as follows: We rotate the subpart through different angles and estimate for each angle by the above function whether the subpart contains the object. The probability that the subpart contains the object rotated by an arbitrary angle is then assumed to be the supremum of the estimated probabilities for the various rotated subparts.

In the following definition we consider subparts of images $\phi \in [0,1]^{C_1}$. The subparts will have the form of possibly rotated cubes C_h of side length $h > 0$, which are subsets of C_1. A subpart of the image $\phi \in [0,1]^{C_1}$ with side length h rotated by an angle $\alpha \in \mathbb{R}$ and located at position $\mathbf{v}$ is given by the function

$$\phi \circ \tau_{\mathbf{v}} \circ rot^{(\alpha)}\big|_{C_h} \in [0,1]^{C_h},$$

where we require $h \leq 1/\sqrt{2}$ and $\mathbf{v} \in [-1/2 + h/\sqrt{2}, 1/2 - h/\sqrt{2}]^2$ to ensure that the function $\tau_{\mathbf{v}} \circ rot^{(\alpha)}\big|_{C_h}$ maps into the image area C_1 for all angles $\alpha \in [0, 2\pi]$ (for an illustration see Figure 2). A non-rotated subpart with side length $0 < h' \leq h$ of an image $\phi \in [0,1]^{C_h}$ is then given by $\phi \circ \tau_{\mathbf{v}}\big|_{C_{h'}} \in [0,1]^{C_{h'}}$ for some $\mathbf{v} \in \mathbb{R}^2$ with $\mathbf{v} + C_{h'} \subseteq C_h$.

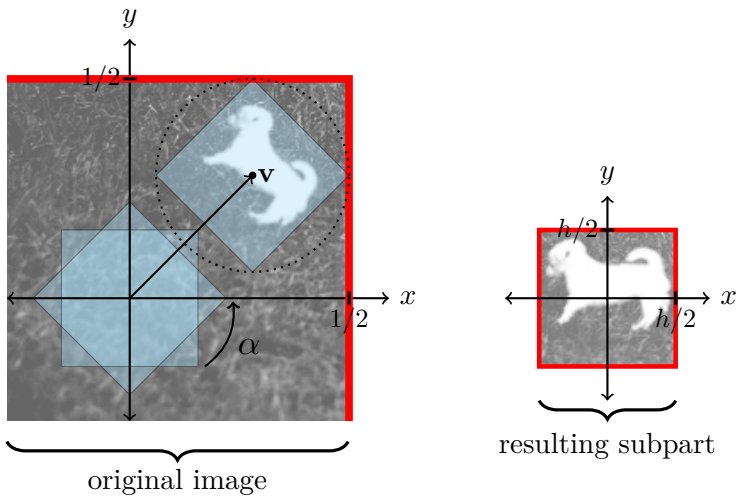

Figure 2: Illustration of an image ϕ and a subpart of the image, which is given by $\phi \circ \tau_{\mathbf{v}} \circ rot^{(\alpha)}\big|_{C_h}$ as used in Definition 2 a).

Definition 2 *Let* $m : [0,1]^{C_1} \to [0,1]$.
a) *Let* $0 < h \leq 1/\sqrt{2}$ *and let*

$$h/\sqrt{2} \leq b \leq 1/2. \tag{5}$$

We say that m *satisfies a* **rotationally symmetric max-pooling model of width** h

and border distance b, *if there exists a function* $f : [0,1]^{C_h} \to [0,1]$ *such that*

$$m(\phi) = \sup_{\mathbf{v} \in \left[-(\frac{1}{2}-b),\frac{1}{2}-b\right]^2} \sup_{\alpha \in [0,2\pi]} f\left(\phi \circ \tau_{\mathbf{v}} \circ rot^{(\alpha)}\big|_{C_h}\right) \quad (\phi \in [0,1]^{C_1}).$$

b) *Let* $l \in \mathbb{N}$ *and* $h > 0$ *and define* $h_k = h/2^{l-k}$ *for* $k \in \mathbb{Z}$. *We say that* $f : [0,1]^{C_h} \to [0,1]$ *satisfies a* **hierarchical model of level** l, *if there exists functions*

$$g_{k,s} : \mathbb{R}^4 \to [0,1] \quad (k = 1, \ldots, l, s = 1, \ldots, 4^{l-k})$$

and functions

$$f_{0,s} : [0,1]^{C_{h_0}} \to [0,1] \quad (s = 1, \ldots, 4^l) \tag{6}$$

such that we have

$$f = f_{l,1}$$

for some $f_{k,s} : [0,1]^{C_{h_k}} \to \mathbb{R}$ *recursively defined by*

$$\begin{aligned} f_{k,s}(\phi) = g_{k,s}\Big(& f_{k-1,4\cdot(s-1)+1}(\phi \circ \tau_{(-h_{k-2},-h_{k-2})}\big|_{C_{h_{k-1}}}), \\ & f_{k-1,4\cdot(s-1)+2}(\phi \circ \tau_{(h_{k-2},-h_{k-2})}\big|_{C_{h_{k-1}}}), \\ & f_{k-1,4\cdot(s-1)+3}(\phi \circ \tau_{(-h_{k-2},h_{k-2})}\big|_{C_{h_{k-1}}}), \\ & f_{k-1,4\cdot s}(\phi \circ \tau_{(h_{k-2},h_{k-2})}\big|_{C_{h_{k-1}}})\Big) \\ & \qquad\qquad (\phi \in [0,1]^{C_{h_k}}) \end{aligned}$$

for $k = 1, \ldots, l$ *and* $s = 1, \ldots, 4^{l-k}$.

c) *We say that* m *satisfies a* **rotationally symmetric hierarchical max-pooling model of level** l**, width** h **and border distance** b, *if* m *satisfies a rotationally symmetric max-pooling model with width* h *and border distance* b, *and the function* $f : [0,1]^{C_h} \to [0,1]$ *in the definition of this rotationally symmetric max-pooling model satisfies a hierarchical model of level* l.

d) *Let* $p = q + s$ *for some* $q \in \mathbb{N}_0$ *and* $s \in (0,1]$, *and let* $C > 0$. *We say that a hierarchical model is* (p, C)*–smooth if all functions* $g_{k,s}$ *in its definition are* (p, C)*–smooth.*

Remark 1. Condition (5) for the border distance ensures that the considered subparts do not extend beyond the border of the image area and that the set of centers $\mathbf{v}$ of the subparts is not empty.

For an illustration of the subdivisions of a subpart in Definition 1 b) and Definition 2 b), see Figure 3.

Figure 3: Illustration of the subdivisions used in Definition 1 b) and Definition 2 b).

3 Convolutional neural network image classifiers

In this section, we define the CNN architecture that we will use in this paper. Our network architecture consists of $t \in \mathbb{N}$ convolutional neural networks computed in parallel, followed by a fully connected standard feedforward neural network. Each of the t convolutional neural networks consists of $L \in \mathbb{N}$ convolutional layers, a linear layer and a global max-pooling layer. As activation function we use the ReLU function $\sigma : \mathbb{R} \to \mathbb{R}$, which is given by $\sigma(x) = \max\{x, 0\}$.

In the r-th convolutional layer we have $k_r \in \mathbb{N}$ channels and use filters of size $M_r \in \mathbb{N}$, where the global max-pooling layer computes the output of the convolutional neural network by a linear layer and by the computation of the maximum over (almost) all neurons of the output of the linear layer (the set of neurons whose maximum is computed depends on an output bound $B \in \mathbb{N}_0$). Our convolutional neural network architecture depends on a weight vector (so-called filters)

$$\mathbf{w} = \left(w^{(r)}_{i,j,s_1,s_2}\right)_{1\leq i,j\leq M_r, s_1\in\{1,\ldots,k_{r-1}\}, s_2\in\{1,\ldots,k_r\}, r\in\{1,\ldots,L\}},$$

bias weights

$$\mathbf{w}_{bias} = \left(w^{(r)}_{s_2}\right)_{s_2\in\{1,\ldots,k_r\}, r\in\{1,\ldots,L\}},$$

and output weights

$$\mathbf{w}_{out} = \left(w_s\right)_{s\in\{1,\ldots,k_L\}}.$$

The output of the convolutional neural network is given by a real-valued function on $[0,1]^{G_\lambda}$ of the form

$$f^{(B)}_{\mathbf{w},\mathbf{w}_{bias},\mathbf{w}_{out}}(\mathbf{x}) = \max\left\{ \sum_{s_2=1}^{k_L} w_{s_2} \cdot o^{(L)}_{(i,j),s_2} \,:\, (i,j) \in \{1+B,\ldots,\lambda-B\}^2 \right\}, \qquad (7)$$

which depends on some output bound $B \in \{0,\ldots,\lfloor(\lambda-1)/2\rfloor\}$, and where $o^{(L)}_{(i,j),s_2}$ is the output of the last convolutional layer, which is recursively defined as follows:

We start with

$$o^{(0)}_{(i,j),1} = x_{\left(\frac{i-1/2}{\lambda}-\frac{1}{2}, \frac{j-1/2}{\lambda}-\frac{1}{2}\right)} \quad \text{for } (i,j) \in \{1,\ldots,\lambda\}^2$$

and define recursively

$$o^{(r)}_{(i,j),s_2} = \sigma\left(\sum_{s_1=1}^{k_{r-1}} \sum_{\substack{t_1,t_2\in\{1,\dots,M_r\}\\ i+t_1-\lceil M_r/2\rceil\in\{1,\dots,\lambda\}\\ j+t_2-\lceil M_r/2\rceil\in\{1,\dots,\lambda\}}} w^{(r)}_{t_1,t_2,s_1,s_2}\cdot o^{(r-1)}_{(i+t_1-\lceil M_r/2\rceil,j+t_2-\lceil M_r/2\rceil),s_1} + w^{(r)}_{s_2}\right) \quad (8)$$

for $(i,j)\in\{1,\dots,\lambda\}^2$, $s_2\in\{1,\dots,k_r\}$ and $r\in\{1,\dots,L\}$. For $\mathbf{k}=(k_1,\dots,k_L)$ and $\mathbf{M}=(M_1,\dots,M_L)$ we introduce the function class

$$\mathcal{F}^{CNN}_{L,\mathbf{k},\mathbf{M},B} = \{f : f \text{ is of the form (7)}\}.$$

In definition (8) we use a so-called zero padding, which ensures that the size of a channel is the same as in the previous layer. For odd filter sizes M_r we obtain a symmetric zero padding as illustrated in Figure 4.

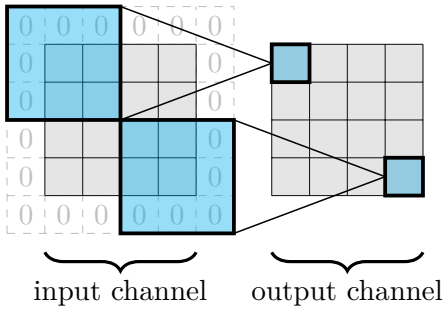

Figure 4: Example of symmetric zero padding for $M_r = 3$ and $\lambda = 4$.

A fully connected standard feedforward neural network $g_{net}:\mathbb{R}^t\to\mathbb{R}$ with ReLU activation function, $L_{net}\in\mathbb{N}_0$ hidden layers and k_r neurons in the r-th layer ($r=1,\dots,L_{net}$) is defined by

$$g_{net}(\mathbf{x}) = \sum_{i=1}^{k_{L_{net}}} w_i^{(L_{net})} g_i^{(L_{net})}(\mathbf{x}) + w_0^{(L_{net})} \quad (9)$$

for some output weights $w_0^{(L_{net})},\dots,w_{k_{L_{net}}}^{(L_{net})}\in\mathbb{R}$, where $g_i^{(L_{net})}$ is recursively defined by

$$g_i^{(r)}(\mathbf{x}) = \sigma\left(\sum_{j=1}^{k_{r-1}} w_{i,j}^{(r-1)} g_j^{(r-1)}(\mathbf{x}) + w_{i,0}^{(r-1)}\right)$$

for $w_{i,0}^{(r-1)},\dots,w_{i,k_{r-1}}^{(r-1)}\in\mathbb{R}$, $i\in\{1,\dots,r_{net}\}$, $r\in\{1,\dots,L_{net}\}$, $k_0=t$ and

$$g_i^{(0)}(\mathbf{x}) = x_i$$

for $i = 1, \dots, k_0$. We define the class of fully connected standard feedforward neural networks with L_{net} layers and $r_{net} \in \mathbb{N}$ neurons per layer by

$$\mathcal{G}_t(L_{net}, r_{net}) = \{g_{net} \ : \ g_{net} \text{ is of the form (9) with } k_1 = \cdots = k_{L_{net}} = r_{net}\}. \quad (10)$$

Our overall convolutional neural network architecture is then defined by

$$\mathcal{F}_{\boldsymbol{\theta}}^{CNN} = \left\{ f(\mathbf{x}) = g_{net}(f_1(\mathbf{x}), \dots, f_t(\mathbf{x})) : f_1, \dots, f_t \in \mathcal{F}_{L,\mathbf{k},\mathbf{M},B}^{CNN}, \ g_{net} \in \mathcal{G}_t(L_{net}, r_{net}) \right\}$$

for a parameter vector $\boldsymbol{\theta} = (t, L, \mathbf{k}, \mathbf{M}, B, L_{net}, r_{net})$.

We define the least squares estimate of $\eta^{(\lambda)}(\mathbf{x}) = \mathbf{E}\{Y = 1 | g_\lambda(\Phi) = \mathbf{x}\}$ by

$$\eta_n = \operatorname*{arg\,min}_{f \in \mathcal{F}_{\boldsymbol{\theta}}^{CNN}} \frac{1}{n} \sum_{i=1}^{n} |Y_i - f(g_\lambda(\Phi_i))|^2 \quad (11)$$

and define our classifier f_n by

$$f_n(\mathbf{x}) = \begin{cases} 1, & \text{if } \eta_n(\mathbf{x}) \geq \frac{1}{2} \\ 0, & \text{elsewhere.} \end{cases}$$

For simplicity, we assume that the minimum of the empirical L_2 risk (11) exists. If this is not the case, our result also holds for an estimator whose empirical L_2 risk is close enough to the infimum. In practical applications, cross entropy loss or hinge loss is commonly used instead of mean squared error considering classification problems. However, in theoretical analysis, further assumptions on the distribution of $(g_\lambda(\Phi), Y)$ are necessary (see, e.g., Kim, Ohn and Kim (2021), Kohler and Langer (2020), and Liu et al. (2021)), which is why we nevertheless use least squares estimates for the a posteriori probability.

4 Main result

In the sequel, let $\lambda \in \mathbb{N}$ be the resolution of the observed images defined as in Section 1.2, i.e., the discretized quadratic images consist of λ^2 pixels. Futhermore, we assume that the functional a posteriori probability $\eta(\phi) = \mathbf{P}\{Y = 1 | \Phi = \phi\}$ satisfies a (p, C)-smooth rotationally symmetric hierarchical max-pooling model of level l and width h. Before presenting the main result, we introduce two further assumptions on the a posteriori probability η. In order to formulate these assumptions we need the following notation: For a subset $A \subseteq \mathbb{R}^2$ let $1|_A : A \to \mathbb{R}$ denote the constant function with value one. Let $f_{0,s} : [0,1]^{C_{h_0}} \to [0,1]$ $(s = 1, \dots, 4^l)$ be the functions from the hierarchical model of η, where $h_0 = h/2^l$. We will use the assumptions below to approximate a rotationally symmetric hierarchical max-pooling model by a convolutional neural network. The first assumption is a smoothness assumption on the functions $f_{0,s}$ if we apply them to constant images.

Assumption 1. *For all $s \in \{1, \dots, 4^l\}$ there exists a (p, C)-smooth function $g_{0,s} : \mathbb{R} \to [0,1]$ such that*

$$g_{0,s}(x) = f_{0,s}\left(x \cdot 1|_{C_{h_0}}\right)$$

holds for all $x \in [0,1]$.

In the second assumption we bound the error that occurs if we replace the input of the function $f_{0,s}$, which is an possibly rotated subpart of an image $\phi \in [0,1]^{C_1}$ (cf., Definition 2), by a constant image whose gray scale value is chosen from the local neighborhood of the corresponding subpart. The size of the subpart, as well as the size of the neighborhood of the subpart, depends on the resolution λ, as shown in Figure 5.

Assumption 2. *There exists a measurable* $A \subset [0,1]^{C_1}$ *with* $P_\Phi(A) = 1$, $\epsilon_\lambda \in [0,1]$ *and a scaling factor* $c > 1$ *with* $h_0 \leq \min\{(c \cdot \sqrt{2})/\lambda, 1/\sqrt{2}\}$ *such that for all* $\phi \in A$, $\mathbf{v} \in [h_0/\sqrt{2} - 1/2, 1/2 - h_0/\sqrt{2}]^2$, $\alpha \in [0, 2\pi]$, *and* $s \in \{1, \dots, 4^l\}$*:*

$$\sup_{\mathbf{z} \in C_1 \,:\, \|\mathbf{v}-\mathbf{z}\|_\infty \leq \frac{c}{\lambda}} \left| f_{0,s}\Big(\underbrace{\phi \circ \tau_{\mathbf{v}} \circ rot^{(\alpha)}|_{C_{h_0}}}_{\text{subpart of } \phi \text{ with center } \mathbf{v}} \Big) - f_{0,s}\Big(\phi(\mathbf{z}) \cdot 1|_{C_{h_0}}\Big) \right| \leq \epsilon_\lambda.$$

Remark 2. Note that $\phi \circ \tau_{\mathbf{v}} \circ rot^{(\alpha)}|_{C_{h_0}}$ is a subpart of ϕ with center $\mathbf{v}$ and width h_0 rotated by α as illustrated in Figure 2. So we apply $f_{0,s}$ to an arbitrary subpart of ϕ with center $\mathbf{v}$ and let $\mathbf{z}$ be choosen from the neighborhood of $\mathbf{v}$. The condition $h_0 \leq (c \cdot \sqrt{2})/\lambda$ ensures that the subpart of width h_0 is contained in the corresponding neighborhood. As illustrated in Figure 5, for a small scaling factor c, we consider subparts whose size approximately corresponds to the resolution.

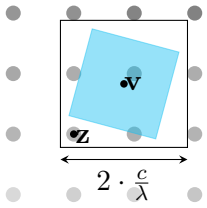

Figure 5: Illustration of a subpart with center $\mathbf{v}$ and a point $\mathbf{z}$ as in Assumption 2, where we choosed $c = 1.05$ and $h_0 = (c \cdot \sqrt{2})/\lambda$. In the background one can see possible pixel values on the corresponding grid $G_\lambda \subset C_1$.

To motivate that Assumption 2 seems realistic for some small $\epsilon_\lambda \in [0,1]$, we consider the following example:

Example 1 *We assume that there exists a resolution* $\lambda_{max} \in \mathbb{N}$ *with* $\lambda_{max} \geq 2$*, so that an image is uniquely defined by the values on the grid*

$$H_{\lambda_{max}} = \left\{ \left(\frac{i}{\lambda_{max} - 1} - \frac{1}{2}, \frac{j}{\lambda_{max} - 1} - \frac{1}{2} \right) \;:\; i, j \in \{0, \dots, \lambda_{max} - 1\} \right\}.$$

We can motivate this by the fact that humans have limited vision (cf., e.g., Gimel'farb and Delmas (2018)), but are still good at classifying images. Therefore we assume that $P_\Phi(A) = 1$ *for* $A \subset [0,1]^{C_1}$ *defined by*

$$A = \{\phi_{\mathbf{x}} \in [0,1]^{C_1} : \mathbf{x} \in [0,1]^{H_{\lambda_{max}}}\},$$

where the image $\phi_{\mathbf{x}} : C_1 \to [0,1]$ *corresponds to the bilinear interpolation of* $\mathbf{x} \in [0,1]^{H_{\lambda_{max}}}$ *(for a definition of* $\phi_{\mathbf{x}}$ *see Section E in the supplement). Furthermore, we suppose that the functions* $f_{0,s} : [0,1]^{C_{h_0}} \to [0,1]$*, which compute the decisions for the lowest level subparts, compute the average pixel value of the corresponding subpart:*

$$f_{0,s}(\phi) = \frac{1}{h_0^2} \int_{C_{h_0}} \phi(\mathbf{x}) d\mathbf{x} \quad \left(\phi \in [0,1]^{C_{h_0}}\right)$$

(without loosing generality, we can ignore measurability issues here). Then, for $\lambda \geq 32 \cdot c \cdot \lambda_{max}^2$ *we have*

$$\sup_{\mathbf{z} \in C_1 \,:\, \|\mathbf{v}-\mathbf{z}\|_\infty \leq \frac{c}{\lambda}} \left| f_{0,s}\Big(\phi \circ \tau_{\mathbf{v}} \circ rot^{(\alpha)}\big|_{C_{h_0}}\Big) - f_{0,s}\Big(\phi(\mathbf{z}) \cdot 1\big|_{C_{h_0}}\Big) \right| \leq \frac{32 \cdot c \cdot \lambda_{max}^2}{\lambda} =: \epsilon_\lambda \quad (12)$$

under the conditions of Assumption 2 (the proof of inequality (12) *can be found in the supplement in Section E).*

Theorem 1 *Let* $n \in \mathbb{N} \setminus \{1\}$ *and* $l \in \mathbb{N}$*, choose* $\lambda \in \mathbb{N}$ *with*

$$\lambda \geq 2^l + 2 \cdot l - 1, \quad (13) \qquad \text{let} \;\; 0 < h \leq \frac{2^l}{\sqrt{2} \cdot \lambda}, \quad (14) \qquad \text{set} \;\; b = \frac{2^l + 2 \cdot l - 1}{2 \cdot \lambda}, \quad (15)$$

and let $p \in [1, \infty)$*. Let* (Φ, Y)*,* (Φ_1, Y_1)*, ...,*(Φ_n, Y_n) *be independent and identically distributed* $[0,1]^{C_1} \times \{0,1\}$*-valued random variables. Assume that the functional a posteriori probability* $\eta(\phi) = \mathbf{P}\{Y = 1 | \Phi = \phi\}$ *satisfies a* (p, C)*-smooth rotationally symmetric hierarchical max-pooling model of level* l*, width* h *and border distance* b*. Furthermore, assume Assumption 1 for* (p, C)*-smooth functions* $\{g_{0,s}\}_{s=1,\ldots,4^l}$ *and Assumption 2 for some* $\epsilon_\lambda \in [0,1]$*, some measurable* $A \subset [0,1]^{C_1}$ *and some scaling factor* $c > 1$*.*

Choose $L_n = \lceil c_1 \cdot n^{2/(2p+4)} \rceil$ *for some sufficiently large constant* $c_1 > 0$*, set*

$$L = \frac{4^{l+1} - 1}{3} \cdot (L_n + 1), \quad t = \left\lceil \frac{2^{l-1/2} \cdot \pi}{c - 1} \right\rceil, \quad B = 2^{l-1} + l - 1, \quad L_{net} = \lceil \log_2 t \rceil,$$

$r_{net} = 3 \cdot t$ *and* $k_r = 5 \cdot 4^{l-1} + c_2$ $(r = 1, \ldots, L)$ *for* $c_2 > 0$ *sufficiently large, and for* $k = 0, \ldots, l$ *set*

$$M_r = \mathbb{1}_{\{k>1\}} \cdot 2^{k-1} + 3 \qquad \left(r = \sum_{i=0}^{k-1} 4^{l-i} \cdot (L_n + 1) + 1, \ldots, \sum_{i=0}^{k} 4^{l-i} \cdot (L_n + 1)\right),$$

where we define the empty sum as zero. Define f_n *as in Section 3. Then we have*

$$\begin{aligned} &\mathbf{P}\{f_n(g_\lambda(\Phi)) \neq Y\} - \min_{f : [0,1]^{G_\lambda} \to [0,1]} \mathbf{P}\{f(g_\lambda(\Phi)) \neq Y\} \\ &\leq c_3 \cdot \sqrt{\log(\lambda) \cdot (\log n)^4 \cdot n^{-\frac{2 \cdot p}{2 \cdot p + 4}} + \epsilon_\lambda} \end{aligned} \quad (16)$$

for some constant $c_3 > 0$ *which does not depend on* λ *and* n*.*

Remark 3. The constant c_3 in (16) depends polynomially on 2^l. Therefore the resolution λ occurs logarithmically in (16) only in the case where $2^l \ll \lambda$, which leads to small widths h (cf., equation (14)). Since the term

$$n^{-\frac{2 \cdot p}{2 \cdot p + 4}}$$

in (16) does not depend on the resoluation λ, our CNN image classifier is able to circumvent the curse of dimensionality in case that the a posteriori probability satisfies a (p, C)-smooth rotationally symmetric hierarchical max-pooling model if we neglect the resolution-depending error term ϵ_λ.

Remark 4. In our approximation result of Lemma 1, we can choose the function $f_{CNN} \in \mathcal{F}_{\boldsymbol{\theta}}^{CNN}$ such that its t CNNs, which are computed in parallel, share the same weights. More precisely, we can choose f_{CNN} such that each filter of any layer corresponds to a rotated filter in the same layer in a CNN computed in parallel (the weights only have different positions within the filters). Therefore, with an appropriate restriction to our function class $\mathcal{F}_{\boldsymbol{\theta}}^{CNN}$ so that the weights of the t CNNs are shared, one could improve the rate of convergence in Theorem 1 by a constant factor. In some image classification applications where rotated objects correspond to each other, such a constraint increases the performance, see, e.g., Marcos, Volpi and Tuia (2016), Dieleman, Willett and Dambre (2015), Wu, Hu and Kong (2015), and Cabrera-Vives et al. (2017). Our theoretical analysis therefore supports the use of such additional weight sharing, in addition to the weight sharing of the convolutional operation, and provides a theoretical indication of why such CNN architectures have better generalization properties.

Remark 5. Condition (13) ensures that the border distance b defined as in (15) remains less than or equal to $1/2$ and that the width h satisfies $h \leq 1/\sqrt{2}$ (cf., equation (14)). Moreover, condition (13) ensures that $h/\sqrt{2} \leq b$. In the case of maximum width $h = 2^l/(\sqrt{2} \cdot \lambda)$ and for large l, we get close to the minimum border distance $h/\sqrt{2}$, since

$$b = \frac{2^l + 2 \cdot l - 1}{2 \cdot \lambda} = \frac{h}{\sqrt{2}} \cdot \underbrace{\frac{2^l + 2 \cdot l - 1}{2^l}}_{\approx 1}.$$

Condition (13) and choice (15) are therefore no real limitations on our model and we obtain, as we have shown in Figure 6 for applications, reasonable border distances b and widths h of the subparts.

Remark 6. Some of the network parameters depend on the rotationally symmetric hierarchical max-pooling model. In applications, these network parameters can be chosen in a data-dependent way, e.g., by using the splitting of the sample technique as used in the next section.

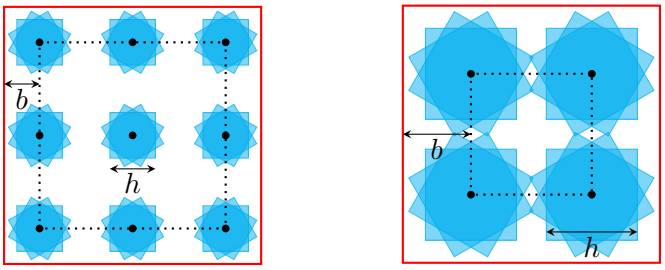

Figure 6: The figure shows possible subparts of width h for the rotationally symmetric hierarchical max-pooling model used in Theorem 1. On both sides we consider an example in which we have $\lambda = 2^9$ and $h = 2^l/(\sqrt{2} \cdot \lambda)$, where on the left hand side we have chosen $l = 7$ and on the right hand side $l = 8$.

5 Application to simulated and real data

In this section, we study the finite sample size behavior of our CNN image classifier introduced in Section 3 by applying it to synthetic and real image data sets. Furthermore, we introduce three other CNN architectures that we can motivate from our theory. We compare the performance of all four CNN image classifiers with two alternative image classifiers that are not specifically designed with the rotation aspect in mind.

We denote the function class introduced in Section 3 by $\mathcal{F}_1 = \mathcal{F}_{\boldsymbol{\theta}}^{CNN}$ for a parameter vector $\boldsymbol{\theta} = (t, L, \mathbf{k}, \mathbf{M}, B, L_{net}, r_{net})$. For the first alternative CNN architecture, we replace the fully connected feedforward neural network by simply computing the maximum over the outputs of the t convolutional neural networks:

$$\mathcal{F}_2 = \left\{ f(\mathbf{x}) = \max\{f_1(\mathbf{x}), \dots, f_t(\mathbf{x})\} : f_1, \dots, f_t \in \mathcal{F}_{L,\mathbf{k},\mathbf{M},B}^{CNN} \right\}.$$

Following the proof of Theorem 1, it is easy to see that the corresponding least squares plug-in image classifier over this function class, achieve the same rate of convergence as in Theorem 1. Our second alternative approach is inspired by the observation from Remark 4. Here we follow, e.g., Dieleman, Willett and Dambre (2015) or Cabrera-Vives et al. (2017) by applying the same CNN to multiple rotated versions of the input image and then compute the overall output as the maximum of the individual outputs. We rotate the input image by 90°, 180°, and 270°, since multiples of 90° rotations map the grid G_λ onto itself. Because it does not matter whether we rotate the input feature maps of a convolutional layer and then inversely rotate the output feature maps, or whether we rotate the corresponding filters, this architecture corresponds in our case to an architecture that has shared rotated filters (for an illustration and a more detailed explanation, see Dieleman, De Fauw and Kavukcuoglu (2016)). The rotation function $rot_{90^\circ} : [0,1]^{G_\lambda} \to [0,1]^{G_\lambda}$ which rotates a discretized image with resolution $\lambda \in \mathbb{N}$ by 90° is given by

$$\left(rot_{90^\circ}(\mathbf{x})\right)_{\left(\frac{i-1/2}{\lambda}-\frac{1}{2}, \frac{j-1/2}{\lambda}-\frac{1}{2}\right)} = x_{\left(\frac{\lambda-j+1-1/2}{\lambda}-\frac{1}{2}, \frac{i-1/2}{\lambda}-\frac{1}{2}\right)} \quad \left(\mathbf{x} \in [0,1]^{G_\lambda}\right)$$

for all $i, j \in \{1, \ldots, \lambda\}$ and our function class is defined by

$$\mathcal{F}_3 = \Big\{ f(\mathbf{x}) = \max\{g(\mathbf{x}), g(rot_{90^\circ}(\mathbf{x})), \ldots, g(\underbrace{rot_{90^\circ} \circ \cdots \circ rot_{90^\circ}}_{3 \text{ times}}(\mathbf{x}))\} : g \in \mathcal{F}_2 \Big\}.$$

For our third alternative network architecture, we extend the idea from the function class $\mathcal{F}_3$ by first rotating an input image by all angles of the discretization

$$\{\alpha_1, \ldots, \alpha_t\} = \left\{ \frac{2\pi}{t} \cdot 0, \frac{2\pi}{t} \cdot 1, \ldots, \frac{2\pi}{t} \cdot (t-1) \right\}$$

of $[0, 2\pi)$ for some $t \in \mathbb{N}$. The corresponding function class is defined by

$$\mathcal{F}_4 = \Big\{ f(\mathbf{x}) = \max\{g(f_{rot}^{(\alpha_1)}(\mathbf{x})), g(f_{rot}^{(\alpha_2)}(\mathbf{x})), \ldots, g(f_{rot}^{(\alpha_t)}(\mathbf{x}))\} : g \in \mathcal{F}_{L,\mathbf{k},\mathbf{M},B}^{CNN} \Big\},$$

where we use a nearest neighbor interpolation for the rotation function $f_{rot}^{(\alpha_i)}$, which we define and explain in detail in Section A.2 of the supplement.

In our first application, we apply our CNN image classifiers to simulated synthetic image datasets. A synthetic image dataset consists of finitely many realizations

$$\mathcal{D}_N = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)\}$$

of a $[0,1]^{G_\lambda} \times \{0,1\}$-valued random variable $(\mathbf{X}, Y)$. Here, as in Section 1, $\lambda \in \mathbb{N}$ denotes the resolution of the images and the value of Y denotes the class of the image. In our first example, we use the values $\lambda = 32$ and $\lambda = 64$. The images of both classes contain three randomly rotated geometric objects each, where images of class 0 contain three squares. The images of class 1 also contain three squares, although at least one of the squares is missing exactly one quarter (see Figure 7). For a detailed explanation of the creation of the image data sets, see Section A.1 in the supplement.

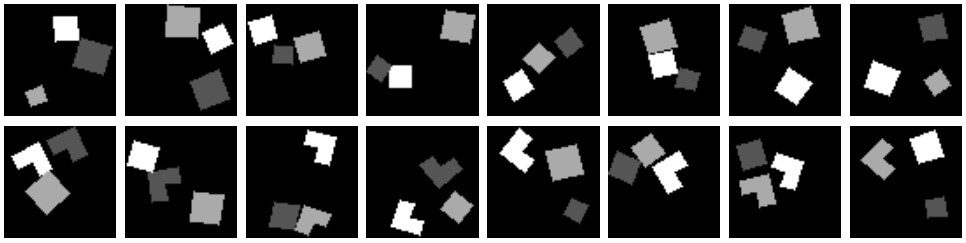

Figure 7: Some random images as realizations of the random variable $\mathbf{X}$, where the first row show images of class 0 and the lower row show images of class 1.

Since our image classifiers depend on parameters that influence their performance, we select them in a data-dependent manner by splitting our training data $\mathcal{D}_n$ into a learning set of size $n_l = \lfloor 4/5 \cdot n \rfloor$ and a validation set of size $n_v = n - n_l$. We then train our classifiers with different choices of parameter combinations on the learning set and choose the parameter combination that minimizes the empirical misclassification risk on

the validation set. Finally, we train our classifier with the best parameter combination on the entire training set $\mathcal{D}_n$. For all four network architectures, we adaptively choose the parameters $l \in \{2,3\}$, $k \in \{2,4\}$ and $L_n \in \{1,2\}$, where the network parameters are then given by $L = L_n \cdot l$, $\mathbf{k} = (k,\dots,k)$, $\mathbf{M} = (M_1,\dots,M_L)$, $B = 2^{l-1} - (l-1)$ with filter sizes $M_1,\dots,M_L$ defined by

$$M_{(r-1)\cdot L_n+1},\dots,M_{r\cdot L_n} = \mathbb{1}_{\{r>2\}} \cdot 2^{r-2} + 3 \quad (r = 1,\dots,l)$$

(note that the choice of layers and filter sizes is a simplification contrary to the choice in Theorem 1). To make the comparison of the four CNN architectures fairer, i.e., to avoid that the network architectures $\mathcal{F}_3$ and $\mathcal{F}_4$ are able to learn more angles, we adaptively choose $t \in \{4,8\}$ for the function classes $\mathcal{F}_1$ and $\mathcal{F}_2$, $t \in \{1,2\}$ for the function class $\mathcal{F}_3$ and $t = 8$ for the function class $\mathcal{F}_4$. In particular, $\mathcal{F}_3$ depends on t, since the function class $\mathcal{F}_2$ depends on t. For the function class $\mathcal{F}_1$ we additionally set $L_{net} = \lceil \log_2 t \rceil$ and $r_{net} = 3 \cdot t$. Finally, for the two alternative image classifiers that are not specifically designed with the rotation aspect in mind, we use a standard fully connected feedforward neural network (abbr. *neural-s*) with an adaptively chosen number of hidden layers and neurons per layer from $\{1,2,\dots,8\}$ and $\{10,20,50,100,200\}$, respectively, and a k_n–nearest neighbor estimator (abbr. *neighbor*) with an adaptively chosen k_n from $\{1,2,3\} \cup \{4,8,12,16,\dots,4 \cdot \lfloor \frac{n_{train}}{4} \rfloor\}$, using the *KNeighborsClassier* function from the *scikit-learn* library.

In our example, we consider $n = 200$ and $n = 400$, using the *Adam* method of the Python library *Keras* for the least-squares minimization problem (11). For the implementation of the five neural network architectures, which are all defined as least squares plug-in classifiers, we also use the *Keras* library.

The performance of each estimate is measured by its empirical misclassification risk

$$\epsilon_N(f_n) = \frac{1}{N} \sum_{k=1}^{N} \mathbb{1}_{\{f_n(\mathbf{x}_{n+k}) \neq y_{n+k}\}} \tag{17}$$

where f_n is the corresponding plug-in image classifier based on the training data and $(\mathbf{x}_{n+1}, y_{n+1}),\dots,(\mathbf{x}_{n+N}, y_{n+N})$ are newly generated independent realizations of the random variable $(\mathbf{X}, Y)$. In our example we choose $N = 10^4$. Since our estimates and the corresponding errors (17) depend on randomly chosen data, we compute the classifiers and their errors (17) on 20 independently generated data sets $\mathcal{D}_{n+N}$. Table 1 lists the median and interquartile range (IQR) of all runs. We observe that the two classifiers using the architectures $\mathcal{F}_3$ and $\mathcal{F}_4$ outperform the two CNN classifiers that do not include additional weight sharing, which supports Remark 5. In two out of four cases, the classifier with architecture $\mathcal{F}_4$ performs best. Moreover, the fourth classifier has the largest relative improvement with increasing sample size, which could be an indicator of a better rate of convergence. The fully connected neural network classifier and the k_n nearest neighbor estimator are not able to achieve satisfactory results because the errors of these estimates roughly correspond to the expected error of a classifier that always estimates the same class. We also observe that a larger resolution leads to a

	$\lambda = 32$		$\lambda = 64$	
	$n = 200$	$n = 400$	$n = 200$	$n = 400$
approach	median (IQR)	median (IQR)	median (IQR)	median (IQR)
$\mathcal{F}_1$	0.3972 (0.0998)	0.2139 (0.1553)	0.4044 (0.1379)	0.2850 (0.3038)
$\mathcal{F}_2$	0.3926 (0.0728)	0.2312 (0.0768)	0.2013 (0.2668)	0.0768 (0.0351)
$\mathcal{F}_3$	**0.1247** (0.0786)	0.0610 (0.0322)	**0.0476** (0.0263)	0.0209 (0.0114)
$\mathcal{F}_4$	0.1386 (0.0862)	**0.0357** (0.0301)	0.0521 (0.0666)	**0.0206** (0.0154)
neural-s	0.4913 (0.0132)	0.4847 (0.0066)	0.4869 (0.0088)	0.4854 (0.0102)
neighbor	0.4939 (0.0055)	0.4957 (0.0059)	0.4937 (0.0056)	0.4934 (0.0067)

Table 1: Median and interquartile range of the empirical misclassification risk $\epsilon_N(f_n)$.

better performance, which suggests that the error term ϵ_λ from Assumption 2 is small for large resolutions.

In our second application, we test our image classifiers on real images. Here we use the classes '4' and '9' of the MNIST-rot dataset (Larochelle et al. (2007)), which contains images of handwritten digits. The digits are randomly rotated by angles from $[0, 2\pi)$ (see Figure 8). The resulting data set consists of $2,400$ training images and $N = 10,000$

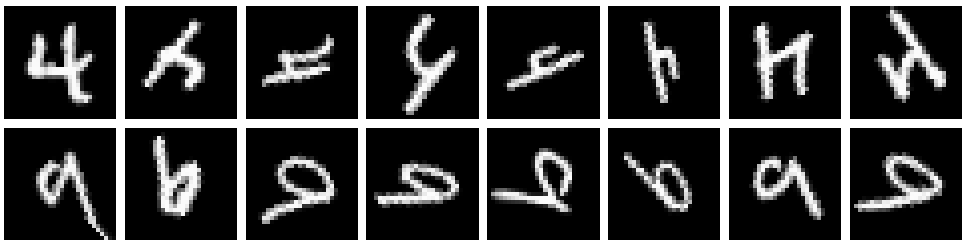

Figure 8: The first row show some images of the fours and the lower row show images of the nines of the MNIST-rot data set.

test images of resolution $\lambda = 28$. Out of the 2,400 training images, we randomly select $n/2$ training images per class and evaluate our classifiers using the corresponding N test images. We choose the parameters of our image classfiers as above. The median and interquartile range (IQR) of the empirical misclassification risk (17) of 20 runs are presented in Table 2. We observe that the classifier using the function class $\mathcal{F}_4$ outperforms the other classfiers. The CNN architecture $\mathcal{F}_3$ performs second best, while the alternative approaches not based on CNNs perform roughly as well as the CNN architectures of classes $\mathcal{F}_1$ and $\mathcal{F}_2$.

6 Proofs

6.1 An approximation result

In this subsection, we show that a rotationally symmetric hierarchical max-pooling model can be approximated by a convolutional neural network.

	$\lambda = 28$	
	$n = 200$	$n = 400$
approach	median (IQR)	median (IQR)
$\mathcal{F}_1$	0.2965 (0.0669)	0.2123 (0.0492)
$\mathcal{F}_2$	0.3201 (0.0482)	0.2153 (0.0421)
$\mathcal{F}_3$	0.1627 (0.0577)	0.1106 (0.0397)
$\mathcal{F}_4$	**0.1169** (0.0397)	**0.0771** (0.0246)
neural-s	0.3061 (0.0321)	0.2044 (0.0343)
neighbor	0.2682 (0.0158)	0.2112 (0.0107)

Table 2: Median and interquartile range of the empirical misclassification risk $\epsilon_N(f_n)$ based on the corresponding subsets of the MNIST-rot data set.

Lemma 1 *Let $n, l, \lambda \in \mathbb{N}$ with $(2^l + 2 \cdot l - 1) \leq \lambda$. Let $0 < h \leq 2^l/(\sqrt{2} \cdot \lambda)$, set $b = (2^l + 2 \cdot l - 1)/(2 \cdot \lambda)$ and let $p \in [1, \infty)$. Let $\eta : [0,1]^{C_1} \to [0,1]$ be a function that satisfies a (p, C)-smooth rotationally symmetric hierarchical max-pooling model of level l, width h and border distance b. Furthermore, assume Assumption 1 for (p, C)-smooth functions $\{g_{0,s}\}_{s=1,\dots,4^l}$ and Assumption 2 for some $\epsilon_\lambda \in [0,1]$, some measurable $A \subset [0,1]^{C_1}$ and $c > 1$. Choose the parameters L_n and $\boldsymbol{\theta} = (t, L, \mathbf{k}, \mathbf{M}, B, L_{net}, r_{net})$ as in Theorem 1. Then there exists some $f_{CNN} \in \mathcal{F}_{\boldsymbol{\theta}}^{CNN}$ such that*

$$|f_{CNN}(g_\lambda(\phi)) - \eta(\phi)|^2 \leq c_4 \cdot \left(n^{-\frac{2 \cdot p}{2 \cdot p + 4}} + \epsilon_\lambda^2 \right)$$

holds for all $\phi \in A$ and some constant $c_4 > 0$ which does not depend on λ and n.

In the points below, we first explain the steps in which we will prove Lemma 1. Afterwards we prove these steps by the auxiliary results of Lemma 2, 3 and 4 to prove Lemma 1 at the end of this subsection.

- First, we introduce a new model, namely the discretized hierarchical max-pooling model of order d (see Definition 3 below).
- *In the first step*, we then show that we can approximate the rotationally symmetric hierarchical max-pooling model by the new discretized model if the functions $\bar{g}_{k,s}^{(i)}$ of the discretized model correspond to the functions $g_{k,s}$ from the continuous model.
- *In the second step*, we show how to bound the error that occurs once the functions $g_{k,s}^{(i)}$ in the discretized hierarchical max-pooling model are replaced by approximations $\bar{g}_{k,s}^{(i)}$.
- *In the third step*, we show that we can compute a discretized hierarchical max-pooling model by a convolutional neural network from the above class $\mathcal{F}_{\boldsymbol{\theta}}^{CNN}$ if the functions $\bar{g}_{k,s}^{(i)}$ correspond to standard feedforward neural networks. This step is similar to Lemma 5 from Kohler, Krzyżak and Walter (2022) for the generalized hierarchical max-pooling model.

Since the functions $g_{k,s}$ of the continuous rotationally symmetric hierarchical max-pooling model are (p, C)-smooth, we can then use the standard feedforward neural networks from Kohler and Langer (2021) and the corresponding approximation result to bound the overall error by combining the three steps.

The new discretized hierarchical max-pooling model is similar to the hierarchical max-pooling model of Kohler, Krzyżak and Walter (2022) (see Definition 1) with the main difference that the positions of the hierarchically combined subparts are variable. Throughout this subsection we will use the following notation: For $k \in \mathbb{N}_0$ and $\lambda \in \mathbb{N}$ we define the index set

$$I^{(k)} = \left\{ -\frac{\lceil 2^{k-1} \rceil + k - 1}{\lambda}, \dots, \frac{-1}{\lambda}, 0, \frac{1}{\lambda}, \dots, \frac{\lceil 2^{k-1} \rceil + k - 1}{\lambda} \right\}^2 \subset \mathbb{R}^2,$$

where we have $I^{(0)} = \{0\} \times \{0\}$.

Definition 3 *Let $\lambda, l, d \in \mathbb{N}$ with $2^l + 2 \cdot l - 1 \leq \lambda$.*
a) *We say that $\bar{\eta} : [0,1]^{G_\lambda} \to \mathbb{R}$ satisfies a* **discretized max-pooling model of order** *d if there exists functions $\bar{f}^{(i)} : [0,1]^{I^{(l)}} \to \mathbb{R}$ for $i \in \{1, \dots, d\}$ such that*

$$\bar{\eta}(\mathbf{x}) = \max_{\mathbf{u} \in G_\lambda \, : \, \mathbf{u} + I^{(l)} \subseteq G_\lambda} \max_{i \in \{1,\dots,d\}} \bar{f}^{(i)}(\mathbf{x}_{\mathbf{u} + I^{(l)}}).$$

b) *We say that $\bar{f} : [0,1]^{I^{(l)}} \to \mathbb{R}$ satisfies a* **discretized hierarchical model of level** *l* **with functions** *$\{\bar{g}_{k,s}\}_{k \in \{0,\dots,l\}, s \in \{1,\dots,4^{l-k}\}}$, where*

$$\bar{g}_{k,s} : \mathbb{R}^4 \to \mathbb{R}_+ \quad (k = 1, \dots, l, s = 1, \dots, 4^{l-k})$$

and

$$\bar{g}_{0,s} : [0,1] \to \mathbb{R}_+ \quad (s = 1, \dots, 4^l),$$

if there exist grid points

$$\mathbf{i}_{k,s} \in \left\{ -\frac{\lfloor 2^{k-1} \rfloor + 1}{\lambda}, \dots, 0, \dots, \frac{\lfloor 2^{k-1} \rfloor + 1}{\lambda} \right\}^2 \quad (k = 0, \dots, l-1, s = 1, \dots, 4^{l-k})$$

such that we have

$$\bar{f} = \bar{f}_{l,1}$$

for some $\bar{f}_{k,s} : [0,1]^{I^{(k)}} \to \mathbb{R}$ recursively defined by

$$\bar{f}_{k,s}(\mathbf{x}) = \bar{g}_{k,s}\Big(\bar{f}_{k-1,4\cdot(s-1)+1}(\mathbf{x}_{\mathbf{i}_{k-1,4\cdot(s-1)+1} + I^{(k-1)}}), \bar{f}_{k-1,4\cdot(s-1)+2}(\mathbf{x}_{\mathbf{i}_{k-1,4\cdot(s-1)+2} + I^{(k-1)}}),$$
$$\bar{f}_{k-1,4\cdot(s-1)+3}(\mathbf{x}_{\mathbf{i}_{k-1,4\cdot(s-1)+3} + I^{(k-1)}}), \bar{f}_{k-1,4\cdot s}(\mathbf{x}_{\mathbf{i}_{k-1,4\cdot s} + I^{(k-1)}}) \Big)$$

for $k = 1, \dots, l$ and $s = 1, \dots, 4^{l-k}$ and

$$\bar{f}_{0,s}(x) = \bar{g}_{0,s}(x)$$

for $s = 1, \dots, 4^l$.

c) *We say that* $\bar{\eta} : [0,1]^{G_\lambda} \to \mathbb{R}$ *satisfies a* **discretized hierarchical max-pooling model of level** l **and order** d **with functions** $\{\bar{g}_{k,s}^{(i)}\}_{i\in\{1,\dots,d\},k\in\{0,\dots,l\},s\in\{1,\dots,4^{l-k}\}}$, *if* $\bar{\eta}$ *satisfies a discretized max-pooling model of order* d *and the functions* $\bar{f}^{(i)} : [0,1]^{I^{(l)}} \to \mathbb{R}$ *in the definition of this discretized max-pooling model satisfy a discretized hierarchical model of level* l *with functions* $\{\bar{g}_{k,s}^{(i)}\}_{k\in\{0,\dots,l\},s\in\{1,\dots,4^{l-k}\}}$ *for all* $i \in \{1, \dots, d\}$.

Lemma 2 *Let* $\lambda, l \in \mathbb{N}$ *with* $2^l + 2 \cdot l - 1 \leq \lambda$, *and set* $b = (2^l + 2 \cdot l - 1)/(2 \cdot \lambda)$. *Furthermore, let* $0 < h \leq 2^l/(\sqrt{2} \cdot \lambda)$ *and set* $h_k = h/2^{l-k}$ *for* $k \in \mathbb{Z}$. *We assume that* $\eta : [0,1]^{C_1} \to \mathbb{R}$ *satisfies a rotationally symmetric max-pooling model of level* l, *width* h, *and border distance* b *given by the functions*

$$g_{k,s} : \mathbb{R}^4 \to [0,1] \quad (k = 1, \dots, l, s = 1, \dots, 4^{l-k})$$

and functions

$$f_{0,s} : [0,1]^{C_{h_0}} \to [0,1] \quad (s = 1, \dots, 4^l),$$

and let the functions $f_{k,s} : [0,1]^{C_{h_k}} \to [0,1]$ $(k = 1, \dots, l, s = 1, \dots, 4^{l-k})$ *defined as in Definition 2. Moreover, we assume that all restrictions* $g_{k,s}|_{[0,1]^4} : [0,1]^4 \to [0,1]$ *are Lipschitz continous regarding the maximum metric with Lipschitz constant* $L > 0$ *and that Assumption 2 is satisfied for some* $\epsilon_\lambda \in [0,1]$, *some measurable* $A \subset [0,1]^{C_1}$ *and* $c > 1$. *Then there exists a discretized hierarchical max-pooling model* $\bar{\eta} : [0,1]^{G_\lambda} \to \mathbb{R}$ *of level* l *and order*

$$d = \left\lceil \frac{2^{l-1/2} \cdot \pi}{c - 1} \right\rceil \tag{18}$$

with functions $\{\bar{g}_{k,s}^{(i)}\}$, *where*

$$\bar{g}_{k,s}^{(i)} = g_{k,s} \quad (i = 1, \dots, d, k = 0, \dots, l, s = 1, \dots, 4^{l-k})$$

with $g_{0,s}(x) = f_{0,s}(x \cdot 1|_{C_{h_0}})$ $(x \in [0,1])$ *for* $s = 1, \dots, 4^l$ *such that*

$$|\bar{\eta}(g_\lambda(\phi)) - \eta(\phi)| \leq L^l \cdot \epsilon_\lambda \quad (\phi \in A).$$

Remark 7. For $p \in [1, \infty)$, the Lipschitz continuity of the restrictions $g_{k,s}|_{[0,1]^4}$ is a consequence of the (p, C)-smoothness of the functions $g_{k,s}$.

Proof. In the proof we use that for $n \in \mathbb{N}$, $a_1, \dots, a_n, b_1, \dots, b_n \in \mathbb{R}$ it holds that

$$\left| \max_{i=1,\dots,n} a_i - \max_{i=1,\dots,n} b_i \right| \leq \max_{i=1,\dots,n} |a_i - b_i|, \tag{19}$$

which follows from the fact that in case $a_j = \max_{i=1,\dots,n} a_i \geq \max_{i=1,\dots,n} b_i$ (which we can assume w.l.o.g.) we have

$$\left| \max_{i=1,\dots,n} a_i - \max_{i=1,\dots,n} b_i \right| = a_j - \max_{i=1,\dots,n} b_i \leq a_j - b_j \leq \max_{i=1,\dots,n} |a_i - b_i|.$$

Before we completely define the discretized hierarchical max-pooling model $\bar{\eta}$, i.e., before we define the corresponding grid points, we will bound $|\bar{\eta}(g_\lambda(\phi)) - \eta(\phi)|$ using equation (19). Therefore we define the grid $G = \{\mathbf{u} \in G_\lambda \ : \ \mathbf{u} + I^{(l)} \subseteq G_\lambda\}$ and the cubes

$$P_{\mathbf{u}} = \left(\mathbf{u} + \left[-\frac{1}{2\lambda}, \frac{1}{2\lambda}\right]^2\right) \cap \left[-\frac{1}{2} + b, \frac{1}{2} - b\right]^2 \quad (\mathbf{u} \in G)$$

such that the definitions of G_λ, $I^{(l)}$ and b yield

$$\begin{aligned}
\bigcup_{\mathbf{u}\in G} P_{\mathbf{u}} &= \bigcup_{\mathbf{u}\in G_\lambda \ : \ \mathbf{u}+I^{(l)}\subseteq G_\lambda} \left(\mathbf{u} + \left[-\frac{1}{2\lambda}, \frac{1}{2\lambda}\right]^2\right) \cap \left[-\frac{1}{2} + b, \frac{1}{2} - b\right]^2 \\
&= \bigcup \left\{ \mathbf{u} + \left[-\frac{1}{2\lambda}, \frac{1}{2\lambda}\right]^2 \ : \ \mathbf{u} \in \left\{ -\frac{1}{2} + \frac{2^{l-1} + l - \frac{1}{2}}{\lambda}, \ldots, \frac{1}{2} - \frac{2^{l-1} + l - \frac{1}{2}}{\lambda} \right\}^2 \right\} \\
&\quad \cap \left[-\frac{1}{2} + \frac{2^{l-1} + l - \frac{1}{2}}{\lambda}, \frac{1}{2} - \frac{2^{l-1} + l - \frac{1}{2}}{\lambda}\right]^2 \\
&= \left[-\frac{1}{2} + \frac{2^{l-1} + l - \frac{1}{2}}{\lambda}, \frac{1}{2} - \frac{2^{l-1} + l - \frac{1}{2}}{\lambda}\right]^2 \\
&= \left[-\frac{1}{2} + b, \frac{1}{2} - b\right]^2
\end{aligned} \tag{20}$$

Furthermore, definition (18) allows us to cover $[0, 2\pi]$ by intervals $\{\Theta_i\}_{i=1,\ldots,d}$ of side length $(c-1)/(2^{l-3/2})$ with centers $\{\alpha_i\}_{i=1,\ldots,d}$. Then, for $\phi \in A$ and $\mathbf{x} \coloneqq g_\lambda(\phi)$ inequality (19) and equation (20) imply

$$\begin{aligned}
&|\bar{\eta}(\mathbf{x}) - \eta(\phi)| \\
&= \left| \max_{\mathbf{u}\in G_\lambda \ : \ \mathbf{u}+I^{(l)}\subseteq G_\lambda} \max_{i\in\{1,\ldots,d\}} \bar{f}_{l,1}^{(i)}(\mathbf{x}_{\mathbf{u}+I^{(l)}}) - \sup_{\mathbf{v}\in\left[-\frac{1}{2}+b,\frac{1}{2}-b\right]^2} \sup_{\alpha\in[0,2\pi]} f_{l,1}(\phi \circ \tau_{\mathbf{v}} \circ rot^{(\alpha)}|_{C_h}) \right| \\
&= \left| \max_{\mathbf{u}\in G} \max_{i\in\{1,\ldots,d\}} \bar{f}_{l,1}^{(i)}(\mathbf{x}_{\mathbf{u}+I^{(l)}}) - \max_{\mathbf{u}\in G} \sup_{\mathbf{v}\in P_{\mathbf{u}}} \max_{i\in\{1,\ldots,d\}} \sup_{\alpha\in\Theta_i} f_{l,1}(\phi \circ \tau_{\mathbf{v}} \circ rot^{(\alpha)}|_{C_h}) \right| \\
&\leq \max_{\mathbf{u}\in G} \left| \max_{i\in\{1,\ldots,d\}} \bar{f}_{l,1}^{(i)}(\mathbf{x}_{\mathbf{u}+I^{(l)}}) - \sup_{\mathbf{v}\in P_{\mathbf{u}}} \max_{i\in\{1,\ldots,d\}} \sup_{\alpha\in\Theta_i} f_{l,1}(\phi \circ \tau_{\mathbf{v}} \circ rot^{(\alpha)}|_{C_h}) \right| \\
&\leq \max_{\mathbf{u}\in G} \sup_{\mathbf{v}\in P_{\mathbf{u}}} \max_{i\in\{1,\ldots,d\}} \sup_{\alpha\in\Theta_i} \left| \bar{f}_{l,1}^{(i)}(\mathbf{x}_{\mathbf{u}+I^{(l)}}) - f_{l,1}(\phi \circ \tau_{\mathbf{v}} \circ rot^{(\alpha)}|_{C_h}) \right|.
\end{aligned}$$

It suffices now to show that for all $i \in \{1, \ldots, d\}$ there exist grid points $\mathbf{i}_{k,s}^{(i)}$ ($k = 0, \ldots, l-1$, $s = 1, \ldots, 4^{l-k}$) of $\bar{f}_{l,1}^{(i)}$, such that

$$\left| \bar{f}_{l,1}^{(i)}(\mathbf{x}_{\mathbf{u}+I^{(l)}}) - f_{l,1}(\phi \circ \tau_{\mathbf{v}} \circ rot^{(\alpha)}|_{C_h}) \right| \leq L^l \cdot \epsilon_\lambda \tag{21}$$

for all $\mathbf{u} \in G$, $\mathbf{v} \in P_{\mathbf{u}}$, $i \in \{1, \dots, d\}$ and $\alpha \in \Theta_i$.

To show this let $\mathbf{u} \in G$, $\mathbf{v} \in P_{\mathbf{u}}$, $i \in \{1, \dots, d\}$ and $\alpha \in \Theta_i$ be fixed for the remainder of the proof. The idea is to construct the grid points $\mathbf{i}_{k,s}^{(i)}$, which do not depend on $\mathbf{u}$, $\mathbf{v}$ and α, such that we are able to prove equation (21) by showing via induction on k that

$$\left| \bar{f}_{k,s}^{(i)}(\mathbf{x}_{\mathbf{u}_{k,s}+I^{(k)}}) - f_{k,s}(\phi \circ \tau_{\mathbf{v}_{k,s}} \circ rot^{(\alpha)}|_{C_{h_k}}) \right| \leq L^k \cdot \epsilon_\lambda \tag{22}$$

for all $k = 0, \dots, l$ and $s = 1, \dots, 4^{l-k}$ where we set $\mathbf{u}_{l,1} = \mathbf{u}$ and $\mathbf{v}_{l,1} = \mathbf{v}$, and

$$\mathbf{u}_{k-1,4\cdot(s-1)+j} = \mathbf{u}_{k,s} + \mathbf{i}_{k-1,4\cdot(s-1)+j}^{(i)} \text{ and } \mathbf{v}_{k-1,4\cdot(s-1)+j} = \mathbf{v}_{k,s} + rot^{(\alpha)}\left(\mathbf{h}_{k-2}^{(j)}\right) \tag{23}$$

for $k = 1, \dots, l$, $s = 1, \dots, 4^{l-k}$ and $j = 1, \dots, 4$ with

$$\mathbf{h}_{k-2}^{(1)} = (-h_{k-2}, -h_{k-2}), \quad \mathbf{h}_{k-2}^{(2)} = (h_{k-2}, -h_{k-2}),$$
$$\mathbf{h}_{k-2}^{(3)} = (-h_{k-2}, h_{k-2}), \quad \mathbf{h}_{k-2}^{(4)} = (h_{k-2}, h_{k-2}).$$

The rest of the proof is organized in four steps. *In the first step,* we define the grid points $\mathbf{i}_{k,s}^{(i)}$ and show that they are well-defined according to Definition 3 b). *In the second step*, we show that $\mathbf{u}_{k,s}$ is 'close' to $\mathbf{v}_{k,s}$ (see Figure 9 for an example). *In the third step*, using Assumption 2, we show that equation (22) holds for $k = 0$ and *the fourth step* corresponds to the induction step for the proof of equation (22).

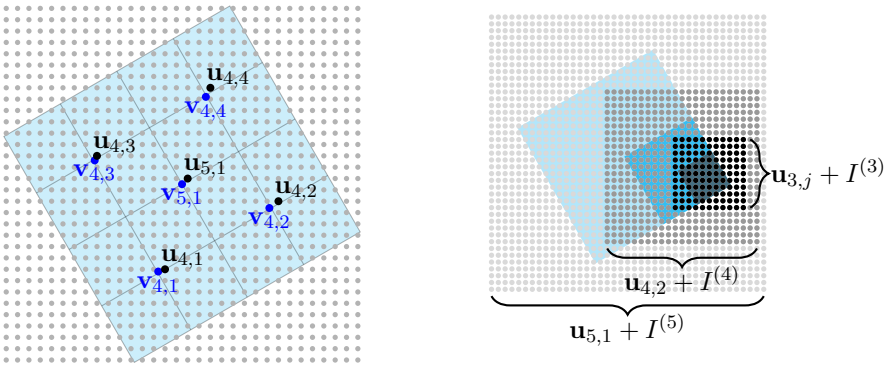

Figure 9: On the left hand side $\mathbf{v}_{k,s}$ and $\mathbf{u}_{k,s}$ are shown as used in the proof of Lemma 2, while on the right hand side one can see the corresponding grids, where $j = 4 \cdot (2-1) + 2 = 6$. We choosed $\alpha = \pi/6$, $\lambda = 100$ and $h = 2^5/(\sqrt{2} \cdot \lambda)$.

Step 1: First, we consider a subpart of width h rotated around the origin by the angle α_i, where α_i is defined as the center of the interval Θ_i. Analogous to the definition of $\mathbf{v}_{k,s}$, we divide the subpart into smaller and smaller subparts and choose the points $\mathbf{z}_{z,k}^{(i)}$ as the centers of these subparts. The idea is that $\mathbf{z}_{k,s}^{(i)}$ is then 'close' to $\mathbf{v}_{k,s} - \mathbf{v}$, as we will see in the *second step*. We set $\mathbf{z}_{l,1}^{(i)} = (0, 0)$ and recursively define

$$\mathbf{z}_{k-1,4\cdot(s-1)+j}^{(i)} = \mathbf{z}_{k,s}^{(i)} + rot^{(\alpha_i)}\left(\mathbf{h}_{k-2}^{(j)}\right)$$

for $k = 1, \ldots, l,\ s = 1, \ldots, 4^{l-k}$ and $j = 1, \ldots, 4$. Since $\mathbf{i}_{k,s}^{(i)}$ are supposed to be grid points we choose

$$\bar{\mathbf{z}}_{k,s}^{(i)} \in \arg\min_{\mathbf{z} \in I^{(l)}} \|\mathbf{z} - \mathbf{z}_{k,s}^{(i)}\|_\infty, \quad (k = 0, \ldots, l, s = 1, \ldots, 4^{l-k}) \tag{24}$$

and define

$$\mathbf{i}_{k-1,4\cdot(s-1)+j}^{(i)} = \bar{\mathbf{z}}_{k-1,4\cdot(s-1)+j}^{(i)} - \bar{\mathbf{z}}_{k,s}^{(i)} \quad (k = 1, \ldots, l, s = 1, \ldots, 4^{l-k}, j = 1, \ldots, 4).$$

To show that the grid points $\mathbf{i}_{k,s}^{(i)}$ are well-defined according to Definition 3 b) we use that $h \leq 2^l/(\sqrt{2} \cdot \lambda)$ and get

$$\left\| rot^{(\beta)} \left(\mathbf{h}_{k-2}^{(j)} \right) \right\|_\infty \leq \sqrt{2} \cdot h_{k-2} = \frac{\sqrt{2} \cdot h}{2^{l-(k-2)}} = \frac{2^{k-2}}{\lambda} \tag{25}$$

for $k = 1, \ldots, l$, $j = 1, \ldots, 4$ and an arbitrary angle $\beta \in [0, 2\pi]$ and therefore we have

$$\|\mathbf{z}_{k-1,4\cdot(s-1)+j}^{(i)}\|_\infty \leq \|\mathbf{z}_{k,s}^{(i)}\|_\infty + \|rot^{(\alpha_i)} \left(\mathbf{h}_{k-2}^{(j)} \right) \|_\infty \leq \|\mathbf{z}_{k,s}^{(i)}\|_\infty + \frac{2^{k-2}}{\lambda}$$

for $k = 1, \ldots, l,\ s = 1, \ldots, 4^{l-k}$ and $j = 1, \ldots, 4$. Since $\mathbf{z}_{l,1} = (0, 0)$ we then have

$$\|\mathbf{z}_{k,s}^{(i)}\|_\infty \leq \sum_{j=k+1}^{l} \frac{2^{j-2}}{\lambda} = \frac{1}{2 \cdot \lambda} \left(\sum_{j=0}^{l-1} 2^j - \sum_{j=0}^{k-1} 2^j \right) = \frac{2^l - 2^k}{2 \cdot \lambda}$$

and due to (24) and the definition of $I^{(l)}$ we get

$$\|\mathbf{z}_{k,s}^{(i)} - \bar{\mathbf{z}}_{k,s}^{(i)}\|_\infty \leq \frac{1}{2 \cdot \lambda} \tag{26}$$

for $k = 0, \ldots, l,\ s = 1, \ldots, 4^{l-k}$. By using the triangle inequality, inequality (26) and inequality (25) we obtain

$$\begin{aligned}
&\|\mathbf{i}_{k-1,4\cdot(s-1)+j}^{(i)}\|_\infty \\
&= \|\bar{\mathbf{z}}_{k,s}^{(i)} - \bar{\mathbf{z}}_{k-1,4\cdot(s-1)+j}^{(i)}\|_\infty \\
&\leq \|\bar{\mathbf{z}}_{k,s}^{(i)} - \mathbf{z}_{k,s}^{(i)}\|_\infty + \|\mathbf{z}_{k,s}^{(i)} - \mathbf{z}_{k-1,4\cdot(s-1)+j}^{(i)}\|_\infty + \|\mathbf{z}_{k-1,4\cdot(s-1)+j}^{(i)} - \bar{\mathbf{z}}_{k-1,4\cdot(s-1)+j}^{(i)}\|_\infty \\
&\leq \frac{1}{2 \cdot \lambda} + \|rot^{(\alpha_i)}(\mathbf{h}_{k-2}^{(j)})\|_\infty + \frac{1}{2 \cdot \lambda} \\
&\leq \frac{2^{k-2} + 1}{\lambda}
\end{aligned}$$

for $k = 1, \ldots, l,\ s = 1, \ldots, 4^{l-k}$ and $j = 1, \ldots, 4$, which together with the fact that $\mathbf{i}_{k,s}^{(i)}$ is a vector of integer multiples of $1/\lambda$ implies

$$\mathbf{i}_{k,s}^{(i)} \in \left\{ -\frac{\lfloor 2^{k-1} \rfloor + 1}{\lambda}, \ldots, 0, \ldots, \frac{\lfloor 2^{k-1} \rfloor + 1}{\lambda} \right\}^2 \quad (k = 0, \ldots, l-1, s = 1, \ldots, 4^{l-k}).$$

Step 2: For $k = 1, \ldots, l$, $s = 1, \ldots, 4^{l-k}$ and $j = 1, \ldots, 4$ we have

$$\begin{aligned}
&\|\mathbf{z}^{(i)}_{k-1,4\cdot(s-1)+j} - (\mathbf{v}_{k-1,4\cdot(s-1)+j} - \mathbf{v})\|_\infty \\
&\leq \|\mathbf{z}^{(i)}_{k,s} - (\mathbf{v}_{k,s} - \mathbf{v})\|_\infty + \left\| rot^{(\alpha_i)}\left(\mathbf{h}^{(j)}_{k-2}\right) - rot^{(\alpha)}\left(\mathbf{h}^{(j)}_{k-2}\right)\right\|_\infty \\
&= \|\mathbf{z}^{(i)}_{k,s} - (\mathbf{v}_{k,s} - \mathbf{v})\|_\infty + \left\| \begin{pmatrix} \cos(\alpha_i) - \cos(\alpha) & \sin(\alpha) - \sin(\alpha_i) \\ \sin(\alpha) - \sin(\alpha_i) & \cos(\alpha) - \cos(\alpha_i) \end{pmatrix} \mathbf{h}^{(j)}_{k-2} \right\|_\infty \\
&\leq \|\mathbf{z}^{(i)}_{k,s} - (\mathbf{v}_{k,s} - \mathbf{v})\|_\infty + 2 \cdot h_{k-2} \cdot \max\{|\sin(\alpha) - \sin(\alpha_i)|, |\cos(\alpha) - \cos(\alpha_i)|\} \\
&\leq \|\mathbf{z}^{(i)}_{k,s} - (\mathbf{v}_{k,s} - \mathbf{v})\|_\infty + h_{k-1} \cdot |\alpha - \alpha_i| \\
&\leq \|\mathbf{z}^{(i)}_{k,s} - (\mathbf{v}_{k,s} - \mathbf{v})\|_\infty + \frac{2^{k-1}}{\sqrt{2}\cdot\lambda} \cdot \frac{\sqrt{2}\cdot(c-1)}{2^l},
\end{aligned}$$

which together with $\mathbf{z}^{(i)}_{l,1} = \mathbf{v}_{l,1} - \mathbf{v} = \mathbf{0}$ implies

$$\|\mathbf{z}^{(i)}_{k,s} - (\mathbf{v}_{k,s} - \mathbf{v})\|_\infty \leq \frac{c-1}{2^l \cdot \lambda} \cdot \sum_{i=k}^{l-1} 2^i = \frac{(c-1)\cdot(2^l - 2^k)}{\lambda \cdot 2^l} < \frac{c-1}{\lambda} \tag{27}$$

for $k = 0, \ldots, l$ and $s = 1, \ldots, 4^{l-k}$. Furthermore, we have

$$\mathbf{u}_{k,s} = \mathbf{u} + \bar{\mathbf{z}}^{(i)}_{k,s} \tag{28}$$

for $k = 0, \ldots, l$, since $\bar{\mathbf{z}}^{(i)}_{l,1} = (0, 0)$ and

$$\mathbf{u}_{k-1,4\cdot(s-1)+j} = \mathbf{u}_{k,s} + \mathbf{i}^{(i)}_{k-1,4\cdot(s-1)+j} = \mathbf{u}_{k,s} + \bar{\mathbf{z}}^{(i)}_{k-1,4\cdot(s-1)+j} - \bar{\mathbf{z}}^{(i)}_{k,s}$$

for $k = 1, \ldots, l$, $s = 1, \ldots, 4^{l-k}$ and $j = 1, \ldots, 4$. Inequalities (26), (27) and (28) imply

$$\begin{aligned}
\|\mathbf{u}_{k,s} - \mathbf{v}_{k,s}\|_\infty &= \|\mathbf{u} - \mathbf{v} + \bar{\mathbf{z}}^{(i)}_{k,s} - \mathbf{z}^{(i)}_{k,s} + \mathbf{z}^{(i)}_{k,s} - \mathbf{v}_{k,s} + \mathbf{v}\|_\infty \\
&\leq \|\mathbf{u} - \mathbf{v}\|_\infty + \|\bar{\mathbf{z}}^{(i)}_{k,s} - \mathbf{z}^{(i)}_{k,s}\|_\infty + \left\|\mathbf{z}^{(i)}_{k,s} - (\mathbf{v}_{k,s} - \mathbf{v})\right\|_\infty \\
&\leq \frac{1}{2\cdot\lambda} + \frac{1}{2\cdot\lambda} + \frac{c-1}{\lambda} \\
&= \frac{c}{\lambda}
\end{aligned} \tag{29}$$

for all $k = 0, \ldots, l$ and $s = 1, \ldots, 4^{l-k}$.

Step 3: To use Assumption 2, we first show that $\mathbf{v}_{0,s} \in [h_0/\sqrt{2} - 1/2, 1/2 - h_0/\sqrt{2}]^2$ for all $s = 1, \ldots, 4^l$. By using inequality (25) we get

$$\|\mathbf{v}_{k-1,4\cdot(s-1)+j} - \mathbf{v}\|_\infty \leq \|\mathbf{v}_{k,s} - \mathbf{v}\|_\infty + \|rot^{(\alpha)}\left(\mathbf{h}^{(j)}_{k-2}\right)\|_\infty \leq \|\mathbf{v}_{k,s} - \mathbf{v}\|_\infty + \frac{2^{k-2}}{\lambda}$$

for $k = 1, \ldots, l$, $s = 1, \ldots, 4^{l-k}$ and $j = 1, \ldots, 4$, which together with $\mathbf{v}_{l,1} = \mathbf{v}$ implies

$$\|\mathbf{v}_{k,s} - \mathbf{v}\|_\infty \leq \sum_{j=k+1}^{l} \frac{2^{j-2}}{\lambda} = \frac{1}{2\cdot\lambda}\left(\sum_{j=0}^{l-1} 2^j - \sum_{j=0}^{k-1} 2^j\right) = \frac{2^l - 2^k}{2\cdot\lambda} \tag{30}$$

for $k = 0, \dots, l$ and $s = 1, \dots, 4^{l-k}$. By using inequality (30), $\mathbf{v} \in [-1/2 + b, 1/2 - b]^2$ and $h_0 \leq 1/(\sqrt{2} \cdot \lambda)$ we get

$$\begin{aligned}
\|\mathbf{v}_{0,s}\|_\infty &\leq \|\mathbf{v}\|_\infty + \|\mathbf{v}_{0,s} - \mathbf{v}\|_\infty \\
&\leq \frac{1}{2} - b + \frac{2^l - 1}{2 \cdot \lambda} \\
&\leq \frac{1}{2} - \frac{2^l + 2 \cdot l - 1}{2 \cdot \lambda} + \frac{2^l - 1}{2 \cdot \lambda} \\
&= \frac{1}{2} - \frac{l}{\lambda} \\
&\leq \frac{1}{2} - \frac{1/(\sqrt{2} \cdot \lambda)}{\sqrt{2}} \\
&\leq \frac{1}{2} - \frac{h_0}{\sqrt{2}}
\end{aligned} \tag{31}$$

for $s = 1, \dots, 4^l$. By using Assumption 2, (29) and (31) we obtain

$$\begin{aligned}
&\left| \bar{f}^{(i)}_{0,s}(\mathbf{x}_{\mathbf{u}_{0,s} + I^{(0)}}) - f_{0,s}(\phi \circ \tau_{\mathbf{v}_{0,s}} \circ rot^{(\alpha)}|_{C_{h_0}}) \right| \\
&= \left| g_{0,s}(x_{\mathbf{u}_{0,s}}) - f_{0,s}(\phi \circ \tau_{\mathbf{v}_{0,s}} \circ rot^{(\alpha)}|_{C_{h_0}}) \right| \\
&= \left| f_{0,s}(\phi(\mathbf{u}_{0,s}) \cdot 1_{C_{h_0}}) - f_{0,s}(\phi \circ \tau_{\mathbf{v}_{0,s}} \circ rot^{(\alpha)}|_{C_{h_0}}) \right| \\
&\leq \epsilon_\lambda
\end{aligned}$$

for $s = 1, \dots, 4^l$.

Step 4: Now we assume that (22) holds for some $k \in \{0, \dots, l-1\}$ and all $s \in \{1, \dots, 4^{l-k}\}$. Because of the Lipschitz assumption on the functions $g_{k,s}$, definition (23), the linearity of the function $rot^{(\alpha)}$ and the induction hypothesis (22), we conclude that

$$\begin{aligned}
&\left| \bar{f}^{(i)}_{k+1,s}(\mathbf{x}_{\mathbf{u}_{k+1,s} + I^{(k+1)}}) - f_{k+1,s}(\phi \circ \tau_{\mathbf{v}_{k+1,s}} \circ rot^{(\alpha)}|_{C_{h_{k+1}}}) \right| \\
&= \Big| g_{k+1,s}\Big(\bar{f}^{(i)}_{k-1,4\cdot(s-1)+1}(\mathbf{x}_{\mathbf{u}_{k+1,s} + \mathbf{i}^{(i)}_{k,4\cdot(s-1)+1} + I^{(k)}}), \bar{f}^{(i)}_{k,4\cdot(s-1)+2}(\mathbf{x}_{\mathbf{u}_{k+1,s} + \mathbf{i}^{(i)}_{k,4\cdot(s-1)+2} + I^{(k)}}), \\
&\qquad \bar{f}^{(i)}_{k,4\cdot(s-1)+3}(\mathbf{x}_{\mathbf{u}_{k+1,s} + \mathbf{i}^{(i)}_{k,4\cdot(s-1)+3} + I^{(k)}}), \bar{f}^{(i)}_{k,4\cdot s}(\mathbf{x}_{\mathbf{u}_{k+1,s} + \mathbf{i}^{(i)}_{k,4\cdot s} + I^{(k)}}) \Big) \\
&\quad - g_{k+1,s}\Big(f_{k,4\cdot(s-1)+1}(\phi \circ \tau_{\mathbf{v}_{k+1,s}} \circ rot^{(\alpha)} \circ \tau_{(-h_{k-1}, -h_{k-1})}|_{C_{h_k}}), \\
&\qquad f_{k,4\cdot(s-1)+2}(\phi \circ \tau_{\mathbf{v}_{k+1,s}} \circ rot^{(\alpha)} \circ \tau_{(h_{k-1}, -h_{k-1})}|_{C_{h_k}}), \\
&\qquad f_{k,4\cdot(s-1)+3}(\phi \circ \tau_{\mathbf{v}_{k+1,s}} \circ rot^{(\alpha)} \circ \tau_{(-h_{k-1}, h_{k-1})}|_{C_{h_k}}), \\
&\qquad f_{k,4\cdot s}(\phi \circ \tau_{\mathbf{v}_{k+1,s}} \circ rot^{(\alpha)} \circ \tau_{(h_{k-1}, h_{k-1})}|_{C_{h_k}}) \Big) \Big| \\
&= \Big| g_{k+1,s}\Big(\bar{f}^{(i)}_{k,4\cdot(s-1)+1}(\mathbf{x}_{\mathbf{u}_{k,4\cdot(s-1)+1} + I^{(k)}}), \bar{f}^{(i)}_{k,4\cdot(s-1)+2}(\mathbf{x}_{\mathbf{u}_{k,4\cdot(s-1)+2} + I^{(k)}}),
\end{aligned}$$

$$
\begin{aligned}
&\bar{f}^{(i)}_{k,4\cdot(s-1)+3}(\mathbf{x}_{\mathbf{u}_{k,4\cdot(s-1)+3}+I^{(k)}}), \bar{f}^{(i)}_{k,4\cdot s}(\mathbf{x}_{\mathbf{u}_{k,4\cdot s}+I^{(k)}})\Big) \\
&- g_{k+1,s}\Big(f_{k,4\cdot(s-1)+1}(\phi\circ\tau_{\mathbf{v}_{k+1,s}}\circ\tau_{rot^{(\alpha)}(\mathbf{h}^{(1)}_{k-1})}\circ rot^{(\alpha)}|_{C_{h_k}}), \\
&\qquad f_{k,4\cdot(s-1)+2}(\phi\circ\tau_{\mathbf{v}_{k+1,s}}\circ\tau_{rot^{(\alpha)}(\mathbf{h}^{(2)}_{k-1})}\circ rot^{(\alpha)}|_{C_{h_k}}), \\
&\qquad f_{k,4\cdot(s-1)+3}(\phi\circ\tau_{\mathbf{v}_{k+1,s}}\circ\tau_{rot^{(\alpha)}(\mathbf{h}^{(3)}_{k-1})}\circ rot^{(\alpha)}|_{C_{h_k}}), \\
&\qquad f_{k,4\cdot s}(\phi\circ\tau_{\mathbf{v}_{k+1,s}}\circ\tau_{rot^{(\alpha)}(\mathbf{h}^{(4)}_{k-1})}\circ rot^{(\alpha)}|_{C_{h_k}})\Big)\Big| \\
&\leq L\cdot\max_{j\in\{1,\ldots,4\}}\Big|\bar{f}^{(i)}_{k,4\cdot(s-1)+j}(\mathbf{x}_{\mathbf{u}_{k,4\cdot(s-1)+j}+I^{(k)}}) \\
&\qquad\qquad - f_{k,4\cdot(s-1)+j}(\phi\circ\tau_{\mathbf{v}_{k,4\cdot(s-1)+j}}\circ rot^{(\alpha)}|_{C_{h_k}})\Big| \\
&\leq L^{k+1}\cdot\epsilon_\lambda
\end{aligned}
$$

for all $s\in\{1,\ldots,4^{l-(k+1)}\}$. □

Lemma 3 *Let $\lambda, l, t\in\mathbb{N}$ with $2^l+2\cdot l-1\leq\lambda$, and let*

$$g^{(i)}_{k,s}:\mathbb{R}^4\to[0,1],\ \bar{g}^{(i)}_{k,s}:\mathbb{R}^4\to\mathbb{R}_+\quad(i=1,\ldots,t,k=1,\ldots,l,s=1,\ldots,4^{l-k}),$$

and

$$g^{(i)}_{0,s}:[0,1]\to[0,1],\ \bar{g}^{(i)}_{0,s}:[0,1]\to[0,2]\quad(i=1,\ldots,t,s=1,\ldots,4^l)$$

be functions such that the restrictions $\{g^{(i)}_{k,s}|_{[0,2]^4}\}_{i=1,\ldots,t,k=1,\ldots,l,s=1,\ldots,4^{l-k}}$ are Lipschitz continuous (with respect to the maximum metric) with Lipschitz constant $C>0$ and

$$\left\|\bar{g}^{(i)}_{k,s}\right\|_{[0,2]^4,\infty}\leq 2\quad(i=1,\ldots,t,k=1,\ldots,l,s=1,\ldots,4^{l-k}).$$

Let $\eta:[0,1]^{G_\lambda}\to\mathbb{R}$ be a function that satisfies a discretized hierarchical max-pooling model of level l and order t with functions $g^{(i)}_{k,s}$ and $\bar{\eta}:[0,1]^{G_\lambda}\to\mathbb{R}$ be a function that satisfies a discretized hierarchical max-pooling model of level l and order t with functions $\bar{g}^{(i)}_{k,s}$. Furthermore, we assume that the two discretized hierarchical max-pooling models have the same grid points $\{\mathbf{i}^{(i)}_{k,s}\}$. Then for any $\mathbf{x}\in[0,1]^{G_\lambda}$ it holds:

$$
\begin{aligned}
&|\eta(\mathbf{x})-\bar{\eta}(\mathbf{x})| \\
&\leq(C+1)^l\cdot\max_{\substack{i\in\{1,\ldots,t\},j\in\{1,\ldots,4^l\},\\ k\in\{1,\ldots,l\},s\in\{1,\ldots,4^{l-k}\}}}\left\{\|g^{(i)}_{0,j}-\bar{g}^{(i)}_{0,j}\|_{[0,1],\infty},\|g^{(i)}_{k,s}-\bar{g}^{(i)}_{k,s}\|_{[0,2]^4,\infty}\right\}.
\end{aligned}
$$

Proof. The result follows by applying the triangle inequality and further straightforward standard techniques. For the sake of completeness a complete proof is given in the supplement. □

Lemma 4 *Let $\lambda, l, t \in \mathbb{N}$ with $2^l + 2 \cdot l - 1 \leq \lambda$. For $L_{net}, r_{net} \in \mathbb{N}$ let*

$$g_{net,k,s}^{(i)} \in \mathcal{G}_4(L_{net}, r_{net}) \quad (i = 1, \ldots, t, k = 1, \ldots, l, s = 1, \ldots, 4^{l-k})$$

and

$$g_{net,0,s}^{(i)} \in \mathcal{G}_1(L_{net}, r_{net}) \quad (i = 1, \ldots, t, s = 1, \ldots, 4^l).$$

Assume that the function $\bar{\eta} : [0,1]^{G_\lambda} \to \mathbb{R}$ satisfies a discretized max-pooling model of level l and order t with functions $\{\bar{g}_{k,s}^{(i)}\}$, where we set

$$\bar{g}_{k,s}^{(i)} = \sigma \circ g_{net,k,s}^{(i)} \quad (i = 1, \ldots, t, k = 0, \ldots, l, s = 1, \ldots, 4^{l-k}).$$

Set $B = 2^{l-1} + (l-1)$, $L_t = \lceil \log_2 t \rceil$, $r_t = 3 \cdot t$, $k_r = 5 \cdot 4^{l-1} + r_{net}$ for $r = 1, \ldots, L$,

$$L = \frac{4^{l+1} - 1}{3} \cdot (L_{net} + 1),$$

and for $k = 0, \ldots, l$ set

$$M_r = \mathbb{1}_{\{k>1\}} \cdot 2^{k-1} + 3 \quad \left(r = \sum_{i=0}^{k-1} 4^{l-i} \cdot (L_{net} + 1) + 1, \ldots, \sum_{i=0}^{k} 4^{l-i} \cdot (L_{net} + 1) \right),$$

where we define the empty sum as zero. Then there exists some $f_{CNN} \in \mathcal{F}_{\boldsymbol{\theta}}^{CNN}$ with $\boldsymbol{\theta} = (t, L, \mathbf{k}, \mathbf{M}, B, L_t, r_t)$ such that

$$\bar{\eta}(\mathbf{x}) = f_{CNN}(\mathbf{x})$$

holds for all $\mathbf{x} \in [0,1]^{G_\lambda}$.

Proof. The proof is similar to the proof of Lemma 5 from Kohler, Krzyżak and Walter (2022) and can be found in the supplement. □

Proof of Lemma 1. Let $\bar{\eta}$ be the discretized hierarchical max-pooling model of level l and order t which is given by the functions $\{\bar{g}_{k,s}^{(i)}\}$ and grid points $\{\mathbf{i}_{k,s}^{(i)}\}$ from Lemma 2 (due to Assumption 1, the functions $\{\bar{g}_{0,s}^{(i)}\}$ have (p, C)-smooth extensions on $\mathbb{R}$), such that

$$|\eta(\phi) - \bar{\eta}(g_\lambda(\phi))| \leq c_5 \cdot \epsilon_\lambda. \tag{32}$$

for all $\phi \in A$ and some constant $c_5 > 0$. Furthermore, let $g_{net,0,s}^{(i)} \in \mathcal{G}_1(L_n, r_{net})$ and $g_{net,k,s}^{(i)} \in \mathcal{G}_4(L_n, r_{net})$ $(k > 0)$ be the standard feedforward neural networks from Kohler and Langer (2021) (cf., Lemma 7 from the supplement) which satisfy

$$\left\| \bar{g}_{k,s}^{(i)} - \sigma \circ g_{net,k,s}^{(i)} \right\|_{[0,2]^4,\infty} \leq \left\| \bar{g}_{k,s}^{(i)} - g_{net,k,s}^{(i)} \right\|_{[0,2]^4,\infty} \leq c_6 \cdot L_n^{-\frac{2 \cdot p}{4}} \leq c_7 \cdot n^{-\frac{p}{2 \cdot p + 4}}$$

for $i = 1, \ldots, t$, $k = 1, \ldots, l$, $s = 1, \ldots, 4^{l-k}$ and some constants $c_6, c_7 > 0$ and

$$\left\| \bar{g}_{0,s}^{(i)} - \sigma \circ g_{net,0,s}^{(i)} \right\|_{[0,1],\infty} \leq \left\| \bar{g}_{0,s}^{(i)} - g_{net,0,s}^{(i)} \right\|_{[0,1],\infty} \leq c_8 \cdot L_n^{-2 \cdot p} \leq c_9 \cdot n^{-\frac{p}{2 \cdot p + 1}},$$

for $i = 1, \dots, t$, $s = 1, \dots, 4^l$ and some constants $c_8, c_9 > 0$, where we choose c_1 in the definition of L_n sufficiently large such that the triangle inequality and the fact that the functions $\bar{g}_{k,s}^{(i)}$ are $[0,1]$-valued imply

$$\left\|\sigma \circ g_{net,k,s}^{(i)}\right\|_{[0,2]^4,\infty} \leq \|\bar{g}_{k,s}^{(i)}\|_{[0,2]^4,\infty} + \left\|\bar{g}_{k,s}^{(i)} - \sigma \circ g_{net,k,s}^{(i)}\right\|_{[0,2]^4,\infty} \leq 1 + c_6 \cdot L_n^{-\frac{2\cdot p}{4}} \leq 2$$

for all $k = 1, \dots, l$ and $s = 1, \dots, 4^{l-k}$ and

$$\left\|\sigma \circ g_{net,0,s}^{(i)}\right\|_{[0,1],\infty} \leq \left\|\bar{g}_{0,s}^{(i)}\right\|_{[0,1],\infty} + \left\|\bar{g}_{0,s}^{(i)} - \sigma \circ g_{net,0,s}^{(i)}\right\|_{[0,1],\infty} \leq 1 + c_8 \cdot L_n^{-2\cdot p} \leq 2$$

for all $s = 1, \dots, 4^l$. Next we define the convolutional neural network $f_{CNN} \in \mathcal{F}^{CNN}$ by using Lemma 4 such that f_{CNN} satisfies a discretized hierarchical max-pooling model which is given by the functions $\{\sigma \circ g_{net,k,s}^{(i)}\}$ and grid points $\{\mathbf{i}_{k,s}^{(i)}\}$. By using $(a+b)^2 \leq 2a^2 + 2b^2$, inequality (32) and Lemma 3 we get

$$\begin{aligned}
&|f_{CNN}(g_\lambda(\phi)) - \eta(\phi)|^2 \\
&\leq 2 \cdot |f_{CNN}(g_\lambda(\phi)) - \bar{\eta}(g_\lambda(\phi))|^2 + 2 \cdot |\bar{\eta}(g_\lambda(\phi)) - \eta(\phi)|^2 \\
&\leq c_{10} \cdot \Big(\max_{k \in \{1,\dots,l\}, s \in \{1,\dots,4^{l-k}\}, j \in \{1,\dots,4^l\}, i \in \{1,\dots,t\}} \Big\{ \|\sigma \circ g_{net,0,j}^{(i)} - \bar{g}_{0,j}^{(i)}\|_{[0,2],\infty}, \\
&\qquad\qquad \|\sigma \circ g_{net,k,s}^{(i)} - \bar{g}_{k,s}^{(i)}\|_{[0,2]^4,\infty} \Big\} \Big)^2 + 2 \cdot c_5^2 \cdot \epsilon_\lambda^2 \\
&\leq c_{11} \cdot \left(n^{-\frac{2\cdot p}{2\cdot p+4}} + \epsilon_\lambda^2 \right)
\end{aligned}$$

for some constants $c_{10}, c_{11} > 0$ which does not depend on λ and n. □

6.2 Proof of Theorem 1

We denote $\mathcal{F} \coloneqq \mathcal{F}_{\boldsymbol{\theta}}^{CNN}$ and choose $c_{12} > 0$ so large that $c_{12} \cdot \log n \geq 2$ holds (cf., Lemma 10 from the supplement). Then $z \geq 1/2$ holds if and only if $T_{c_{12}\cdot \log n} z \geq 1/2$, and consequently we have

$$f_n(\mathbf{x}) = \begin{cases} 1 & , \text{ if } T_{c_{12}\cdot\log n}\eta_n(\mathbf{x}) \geq \frac{1}{2} \\ 0 & , \text{ elsewhere.} \end{cases}$$

Because of Lemma 5 from the supplement we have

$$\begin{aligned}
&\mathbf{P}\{f_n(g_\lambda(\Phi)) \neq Y\} - \min_{f:[0,1]^{G_\lambda} \to [0,1]} \mathbf{P}\{f(g_\lambda(\Phi)) \neq Y\} \\
&\leq 2 \cdot \sqrt{\mathbf{E}\left\{ \int |T_{c_{12}\cdot\log n}\eta_n(\mathbf{x}) - \eta^{(\lambda)}(\mathbf{x})|^2 \mathbf{P}_{g_\lambda(\Phi)}(d\mathbf{x}) \right\}}
\end{aligned}$$

and hence it suffices to show

$$\mathbf{E}\left\{ \int |T_{c_{12}\cdot\log n}\eta_n(\mathbf{x}) - \eta^{(\lambda)}(\mathbf{x})|^2 \mathbf{P}_{g_\lambda(\Phi)}(d\mathbf{x}) \right\} \leq c_{13} \cdot \left(\log(\lambda) \cdot (\log n)^4 \cdot n^{-\frac{2\cdot p}{2\cdot p+4}} + \epsilon_\lambda^2 \right)$$

for some constant $c_{13} > 0$. By Lemma 6 from the supplement we have

$$\begin{aligned}
&\mathbf{E}\left\{\int |T_{c_{12}\cdot\log n}\eta_n(\mathbf{x}) - \eta^{(\lambda)}(\mathbf{x})|^2 \mathbf{P}_{g_\lambda(\Phi)}(d\mathbf{x})\right\} \\
&\leq \frac{c_{14}\cdot(\log n)^2 \cdot \sup_{\mathbf{x}_1^n}\left(\log\left(\mathcal{N}_1\left(\frac{1}{n\cdot c_{12}\cdot\log(n)}, T_{c_{12}\cdot\log(n)}\mathcal{F}, \mathbf{x}_1^n\right)\right)+1\right)}{n} \\
&\quad + 2\cdot \inf_{f\in\mathcal{F}}\int |f(\mathbf{x}) - \eta^{(\lambda)}(\mathbf{x})|^2 \mathbf{P}_{g_\lambda(\Phi)}(d\mathbf{x})
\end{aligned}$$

for some constant $c_{14} > 0$. For the first term Lemma 10 from the supplement implies

$$\begin{aligned}
&\frac{c_{14}\cdot(\log n)^2 \cdot \sup_{\mathbf{x}_1^n}\left(\log\left(\mathcal{N}_1\left(\frac{1}{n\cdot c_{12}\cdot\log(n)}, T_{c_{12}\cdot\log(n)}\mathcal{F}, \mathbf{x}_1^n\right)\right)+1\right)}{n} \\
&\leq \frac{c_{15}\cdot L^2\cdot\log(L)\cdot\log(\lambda)\cdot(\log n)^3}{n} \\
&\leq c_{16}\cdot\log(\lambda)\cdot(\log n)^4\cdot n^{-\frac{2\cdot p}{2\cdot p+4}}.
\end{aligned}$$

for some constants $c_{15}, c_{16} > 0$. Next we derive a bound on the approximation error

$$\inf_{f\in\mathcal{F}}\int |f(\mathbf{x}) - \eta^{(\lambda)}(\mathbf{x})|^2 \mathbf{P}_{g_\lambda(\Phi)}(d\mathbf{x}).$$

By using the fact that the a posteriori probability η minimizes the L_2 risk (w.r.t. the random vector (Φ, Y)), $\mathbf{P}_\Phi(A) = 1$ and Lemma 1, we get

$$\begin{aligned}
\inf_{f\in\mathcal{F}}\int |f(\mathbf{x}) - \eta^{(\lambda)}(\mathbf{x})|^2 \mathbf{P}_{g_\lambda(\Phi)}(d\mathbf{x}) &\leq \int |\bar{f}(\mathbf{x}) - \eta^{(\lambda)}(\mathbf{x})|^2 \mathbf{P}_{g_\lambda(\Phi)}(d\mathbf{x}) \\
&= \mathbf{E}\left\{|\bar{f}(g_\lambda(\Phi)) - Y|^2\right\} - \mathbf{E}\left\{|\eta^{(\lambda)}(g_\lambda(\Phi)) - Y|^2\right\} \\
&\leq \mathbf{E}\left\{|\bar{f}(g_\lambda(\Phi)) - Y|^2\right\} - \mathbf{E}\left\{|\eta(\Phi) - Y|^2\right\} \\
&= \int_A |\bar{f}(g_\lambda(\phi)) - \eta(\phi)|^2 \mathbf{P}_\Phi(d\phi) \\
&\leq c_{17}\cdot\left(n^{-\frac{2\cdot p}{2\cdot p+4}} + \epsilon_\lambda\right)
\end{aligned}$$

for $\bar{f} \in \mathcal{F}$ chosen as in Lemma 1 and some constant $c_{17} > 0$. Summarizing the above results, the proof is complete. □

7 Acknowledgment

The author would like to thank the AE and four anonymous referees for the helpful comments and suggestions to improve an early version of this manuscript.

References

[1] Bauer, B., and Kohler, M. (2019). On deep learning as a remedy for the curse of dimensionality in nonparametric regression. *Annals of Statistics*, **47**, pp. 2261–2285.

[2] Bos, T., and Schmidt-Hieber, J. (2021). Convergence rates of deep ReLU networks for multiclass classification. arXiv: 2108.00969.

[3] Cabrera-Vives, G., Reyes, I., Förster, F., Estévez, P. A., and Maureira, J. C. (2017). Deep-HiTS: Rotation Invariant Convolutional Neural Network for Transient Detection. arXiv: 1701.00458.

[4] Cohen, T. S., and Welling, M. (2016). Group Equivariant Convolutional Networks. *International Conference on Machine Learning (ICML)*, **48**, pp. 2990–2999.

[5] Cover, T. M. (1968). Rates of convergence of nearest neighbor procedures. *Proceedings of the Hawaii International Conference on Systems Siences*, pp. 413–415. Honolulu, HI.

[6] Delchevalerie, V., Bibal, A., Frenay, B., and Mayer, A. (2021). Achieving Rotational Invariance with Bessel-Convolutional Neural Networks. *Advances in Neural Information Processing Systems*.

[7] Devroye, L. (1982). Necessary and sufficient conditions for the pointwise convergence of nearest neighbor regression function estimates. *Zeitschrift für Wahrscheinlichkeitstheorie und verwandte Gebiete*, **61**, pp. 467–481.

[8] Devroye, L., Györfi, L., and Lugosi, G. (1996). *A Probabilistic Theory of Pattern Recognition*. Springer, New York.

[9] Dieleman, S., De Fauw, J., and Kavukcuoglu, K. (2016). Exploiting Cyclic Symmetry in Convolutional Neural Networks. *Proceedings of the 33rd International Conference on International Conference on Machine Learning*, **48**, pp. 1889–1898.

[10] Dieleman, S., Willett, K. W., and Dambre, J. (2015). Rotation-invariant convolutional neural networks for galaxy morphology prediction. *Monthly Notices of the Royal Astronomical Society*, **450**, pp. 1441–1459.

[11] Du, S. S., Lee, J. D., Li, H., Wang, L., and Zhai, X. (2018). Gradient Descent Finds Global Minima of Deep Neural Networks. arXiv: 1811.03804.

[12] Gimel'farb, G., and Delmas, P. (2018). *Image Processing And Analysis: A Primer*. World Scientific.

[13] Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning*. MIT Press, London.

[14] He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778.

[15] Hu, T., Shang, Z., and Cheng, G. (2020). Sharp Rate of Convergence for Deep Neural Network Classifiers under the Teacher-Student Setting. arXiv: 2001.06892.

[16] Imaizumi, M., and Fukamizu, K. (2019). Deep neural networks learn non-smooth functions effectively. *Proceedings of the 22nd International Conference on Artificial Intelligence and Statistics*. Naha, Okinawa, Japan.

[17] Kim, Y., Ohn, I., and Kim, D. (2021). Fast convergence rates of deep neural networks for classification. *Neural Networks*, **138**, pp. 179–197.

[18] Kohler, M., and Krzyżak, A. (2017). Nonparametric regression based on hierarchical interaction models. *IEEE Transactions on Information Theory*, **63**, pp. 1620–1630.

[19] Kohler, M., and Krzyżak, A. (2021). Over-parametrized deep neural networks minimizing the empirical risk do not generalize well. *Bernoulli*, **27**, pp. 2564–2597.

[20] Kohler, M., Krzyzak, A., and Langer, S. (2019). Estimation of a function of low local dimensionality by deep neural networks. arXiv: 1908.11140.

[21] Kohler, M., Krzyżak, A., and Walter, B. (2022). On the rate of convergence of image classifiers based on convolutional neural networks. *Annals of the Institute of Statistical Mathematics*, pp. 1–24.

[22] Kohler, M., and Langer, S. (2020). Statistical theory for image classification using deep convolutional neural networks with cross-entropy loss. arXiv: 2011.13602.

[23] Kohler, M., and Langer, S. (2021). On the rate of convergence of fully connected very deep neural network regression estimates. *Annals of Statistics*, **49**, pp. 2231–2249.

[24] Langer, S. (2021). Analysis of the rate of convergence of fully connected deep neuralnetwork regression estimates with smooth activation function. *Journal of Multivariate Analysis*, **182**, p. 104695.

[25] Larochelle, H., Erhan, D., Courville, A., Bergstra, J., and Bengio, Y. (2007). An empirical evaluation of deep architectures on problems with many factors of variation. *Proceedings of the 24th International Conference on Machine Learning (ICML)*.

[26] Lin, S., and Zhang, J. (2019). Generalization bounds for convolutional neural networks. arXiv: 1910.01487.

[27] Liu, H., Chen, M., Zhao, T., and Liao, W. (2021). Besov function approximation and binary classification on low-dimensional manifolds using convolutional residual networks. *Proceedings of the 38th International Conference on Machine Learning (PMLR)*, **139**, pp. 6770–6780.

[28] Marcos, D., Volpi, M., and Tuia, D. (2016). Learning rotation invariant convolutional filters for texture classification. *International Conference on Pattern Recognition (ICPR)*, pp. 2012–2017.

[29] Oono, K., and Suzuki, T. (2019). Approximation and Non-parametric Estimation of ResNet-type Convolutional Neural Networks. *In International Conference on Machine Learning*, pp. 4922–4931.

[30] Petersen, P., and Voigtlaender, F. (2020). Equivalence of approximation by convolutional neural networks and fully-connected networks. *Proceedings of the American Mathematical Society*, **148**, pp. 1567–1581.

[31] Rawat, W., and Wang, Z. (2017). Deep Convolutional Neural Networks for Image Classification: A Comprehensive Review. *Neural Computation*, **29**, pp. 2352–2449.

[32] Schmidt-Hieber, J. (2020). Nonparametric regression using deep neural networks with ReLU activation function. *Annals of Statistics*, **48**, pp. 1875–1897.

[33] Suzuki, T., and Nitanda, A. (2019). Deep learning is adaptive to intrinsic dimensionality of model smoothness in anisotropic Besov space. arXiv: 1910.12799.

[34] Veeling, B. S., Linmans, J., Winkens, J., Cohen, T., and Welling, M. (2018). Rotation Equivariant CNNs for Digital Pathology. arXiv: 1806.03962.

[35] Walter, B. (2021). Analysis of convolutional neural network image classifiers in a hierarchical max-pooling model with additional local pooling. arXiv: 2106.05233.

[36] Wu, F., Hu, P., and Kong, D. (2015). Flip-Rotate-Pooling Convolution and Split Dropout on Convolution Neural Networks for Image Classification. arXiv: 1507.08754.

[37] Yarotsky, D. (2018). Universal approximations of invariant maps by neural networks. arXiv: 1804.10306.

[38] Zhou, D.-X. (2020). Universality of deep convolutional neural networks. *Applied and Computational Harmonic Analysis*, **48**, pp. 787–794.