

A NOVEL STATISTICAL APPROACH TO ANALYZE IMAGE CLASSIFICATION

BY JUNTONG CHEN^{1,a}, SOPHIE LANGER^{2,b} AND JOHANNES SCHMIDT-HIEBER^{3,c}

¹*School of Mathematical Sciences, Xiamen University, juntong.chen@xmu.edu.cn*

²*Fakultät für Mathematik, Ruhr-Universität Bochum, s.langer@rub.de*

³*Department of Applied Mathematics, University of Twente, a.j.schmidt-hieber@utwente.nl*

The recent statistical theory of neural networks focuses on nonparametric denoising problems that treat randomness as additive noise. Variability in image classification datasets does, however, not originate from additive noise but from variation of the shape and other characteristics of the same object across different images. To address this problem, we introduce a tractable model for supervised image classification. While from the function estimation point of view, every pixel in an image is a variable, and large images lead to high-dimensional function recovery tasks suffering from the curse of dimensionality, increasing the number of pixels in the proposed image deformation model enhances the image resolution and makes the object classification problem easier. We introduce and theoretically analyze three approaches. Two methods combine image alignment with a one-nearest neighbor classifier. Under a separation condition, it is shown that perfect classification is possible. The third method fits a convolutional neural network (CNN) to the data. We derive a rate for the misclassification error that depends on the sample size and the complexity of the deformation class. An empirical study corroborates the theoretical findings.

1. Introduction. From a machine learning perspective, object recognition is typically framed as a high-dimensional classification problem, where each pixel is treated as an independent variable. The objective of the classification rule is to learn the functional relationship between the pixel values of an input image and the corresponding conditional class probabilities or the labels. However, for high-resolution images with many pixels, the domain of the function is a high-dimensional space, leading to slow convergence rates due to the curse of dimensionality. To align the strong empirical performance of convolutional neural networks (CNNs) with theoretical guarantees, one common approach is to assume that the true functional relationship between inputs and outputs has a latent low-dimensional structure; see, for example, [32]. This assumption allows the convergence rate to depend only on this low-dimensional structure, potentially circumventing the curse of dimensionality.

A functional data perspective is to treat images as highly structured objects that can be represented by a bivariate function, where each pixel value corresponds to a local average of the function over its location. From this viewpoint, variations of the same object in different images are interpreted as deformations of a template image, which introduces additional complexity for classification compared to the pixelwise approach. This concept has been explored in foundational work on pattern recognition. Grenander and Mumford ([24, 46]) distinguish between *pure images* and *deformed images*, with the latter being generated from pure images through specific deformations. Since then, several generative models for object deformation on images have been proposed. For instance, [12, 13, 41] study a rich class of

Received August 2025; revised December 2025.

MSC2020 subject classifications. Primary 62H30; secondary 62G05.

Key words and phrases. Image classification, convolutional neural networks, statistical learning, functional data, misclassification error.

local deformations, while [47, 48] extend these models to address more complex deformations such as noise, blur, multiscale superposition, and domain warping. Generative models are becoming increasingly important in fields such as medical image registration or computer vision [5, 29, 58, 69]. While existing work focuses on algorithms that can effectively handle image deformations, statistical modeling and theoretical generalization guarantees are, however, underexplored.

This paper aims to bridge this gap by introducing a tractable image deformation model that addresses a fundamental yet rich class of geometric transformations, including common variations in object positioning, scaling, brightness, and rotation. We focus on a binary image classification setting, where datasets consist of n labeled images of two objects, such as the digits 0 and 4, with each image representing a random deformation of one of these objects. In the case of digits, these deformations can capture natural variations found in, for instance, individual handwriting.

Our statistical analysis differs significantly from the wide range of well-understood classification problems that rely on local smoothing. In these settings, randomness arises because the covariates (or inputs) do not fully determine the class label, requiring classifiers to aggregate training data with similar covariate values to effectively denoise. The resulting convergence rates for standard smoothness classes typically align with those seen in nonparametric regression and suffer from the curse of dimensionality for high-resolution images [9].

In the proposed statistical setting, the randomness occurs due to the different deformations that can arise on images within one class. The objective of the classification is, therefore, to remain invariant to these uninformative variations.

We approach this nonstandard classification problem by first constructing classifiers exploiting the specific structure of the random object deformation model. These classifiers interpolate the data and can be interpreted as one-nearest neighbor classifiers in a transformed space. At low image resolutions, however, distinguishing between highly similar objects becomes impossible. We prove that if the two objects satisfy a separation condition that depends on the image resolution, then the classifiers can perfectly discriminate between the two classes on test data. Interestingly, the sample size n plays a minor role in the analysis; it suffices to observe one training sample for each class. The imposed separation condition is also necessary in the sense that any smaller separation would result in nonidentifiability of the classes, making accurate discrimination impossible (see Theorem 3.8).

A key contribution of this work is the misclassification error rates for CNN classifiers, showing that CNNs can adapt to various geometric deformations. As a first result, we prove that for a suitably chosen network architecture, specific parameter assignments in a CNN can effectively discriminate between the two classes. This shows that among all classifiers that are representable by a given CNN architecture, there exist classifiers that are (nearly) invariant with respect to the possible deformations of an object in an image. Based on this and statistical learning techniques, we derive misclassification bounds for CNN classifiers in Theorems 4.1 and 4.4. For specific deformation classes, the obtained rates depend on the sample size and the number of pixels and the dependence on the input dimension is much more favorable than the curse of dimensionality observed in standard nonparametric convergence rates. The proposed setting has the potential to provide a more refined understanding of phenomena such as overparametrization or the improved performance through data augmentation.

The article is structured as follows. In Section 2 we introduce the image deformation model. As theoretical benchmarks, we introduce two classifiers for this deformation model in Section 3. Section 4 analyzes a CNN-based classifier. The simulation study in Section 5 compares the three classifiers. A literature overview is provided in Section 6. We conclude in Section 7 with a discussion of potential extensions and future research directions.

Notation. For a real number x , $\lfloor x \rfloor$ represents the largest integer that is less than or equal to x , whereas $\lceil x \rceil$ represents the smallest integer that is greater than or equal to x . We denote vectors and matrices by bold letters, for example, $\mathbf{v} := (v_1, \dots, v_d)$ and $\mathbf{W} = (W_{i,j})_{i=1,\dots,m; j=1,\dots,n}$. As usual, $\|\mathbf{v}\|_p := (\sum_{i=1}^d |v_i|^p)^{1/p}$ and $\|\mathbf{v}\|_\infty := \max_i |v_i|$. For a matrix $\mathbf{W} = (W_{i,j})_{i=1,\dots,m; j=1,\dots,n}$, we define the maximum entry norm as $\|\mathbf{W}\|_\infty = \max_{i=1,\dots,m; j=1,\dots,n} |W_{i,j}|$. For two sequences $(a_n)_n$ and $(b_n)_n$, we write $a_n \lesssim b_n$ if there exists a constant C such that $a_n \leq C b_n$ for all n . For $m \geq 2$ and numbers $a_1, a_2, \dots, a_m$, we define $a_1 \vee a_2 \vee \dots \vee a_m = \max\{a_1, a_2, \dots, a_m\}$ and $a_1 \wedge a_2 \wedge \dots \wedge a_m = \min\{a_1, a_2, \dots, a_m\}$. For functions, $\|\cdot\|_{L^p(D)}$ denotes the L^p -norm on the domain D . When $D = [0, 1]^2$, we also write $\|\cdot\|_p$. For a function $A = (a_1, a_2) : \mathbb{R}^2 \rightarrow \mathbb{R}^2$, we define $\|A\|_{L^\infty(D)} := \max_{i=1,2} \sup_{\mathbf{z} \in D} |a_i(\mathbf{z})|$ and set $\|A\|_\infty := \|A\|_{L^\infty([0,1]^2)}$. For B a set, the indicator function is denoted by $\mathbb{1}(x \in B)$. It takes the value 1 if $x \in B$ and 0 otherwise. Since we frequently work with bivariate functions, we write $f(\cdot, \cdot)$ for a function $(x, y) \mapsto f(x, y)$. In this notation the dots $\cdot$ represent different variables.

2. Image deformation models. We first discuss a specific case and then introduce the full image deformation model. For any integers $j, \ell \in \mathbb{Z}$, define

$$I_{j,\ell} = \left[\frac{j-1}{d}, \frac{j}{d} \right) \times \left[\frac{\ell-1}{d}, \frac{\ell}{d} \right),$$

representing a square with side length $1/d$. A $d \times d$ image with d^2 pixels, as illustrated in Figure 1(a), can be expressed as a bivariate function $f : \mathbb{R}^2 \rightarrow [0, \infty)$, where the grayscale value of the (j, ℓ) th pixel is given by

$$(1) \quad \bar{f}_{j,\ell} = d^2 \int_{I_{j,\ell}} f(u, v) du dv, \quad j, \ell \in \{1, \dots, d\},$$

representing the average intensity of f on $I_{j,\ell}$. The pixel value decodes the grayscale with smaller function values corresponding to darker pixels. To deal with image deformations, it is more convenient to define f on $\mathbb{R}^2$ instead of $[0, 1]^2$.

The support of a function g is defined as the set of all x for which the function value $g(x)$ is nonzero. For a function f representing an image, we refer to f restricted to its support as the *object*. The *background* is defined as the complement of the support, that is, the set

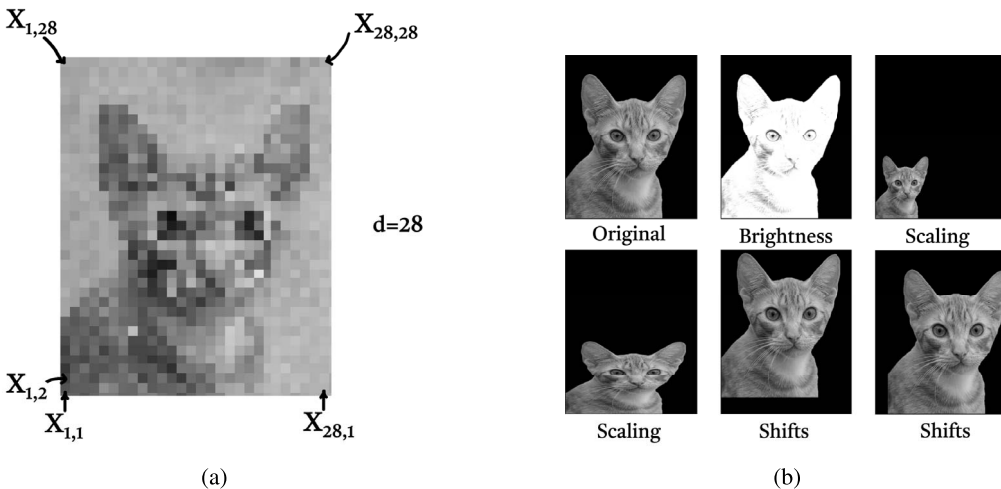


FIG. 1. Cat image represented by pixels and different deformations of a cat image.

of x with $f(x) = 0$. Assuming that the images have zero background, all positive pixels are considered as part of the object itself.

Next, we explore how simple transformations such as scaling, shifting, and brightness affect the function f and, consequently, the image. Multiplying the function values by a factor $\eta > 1$ brightens the image, making the pixel values appear whiter, while multiplying by $0 < \eta < 1$ darkens the image. Shifting the object within the image, either horizontally or vertically, corresponds to translating the function f by a vector (τ, τ') , changing the function values to $f(x - \tau, y - \tau')$. Stretching or shrinking the object along the x - or y -axis transforms the function value to $f(\xi x, \xi' y)$, where $\xi < 1$ or $\xi' < 1$ stretches the object along the x - or y -axis and $\xi, \xi' > 1$ shrinks it. Combining these transformations, the function becomes $(x, y) \mapsto \eta f(\xi x - \tau, \xi' y - \tau')$, capturing the effects of brightness adjustment, translation, and scaling on the image; see Figure 1(b) for an example of a deformed image of a cat under different scaling, shifting, and brightness adjustments.

To distinguish between images of different object classes, the underlying idea of the data generating model is to assume that images from different classes correspond to different *template functions* f . By drawing the parameters $(\eta, \tau, \tau', \xi, \xi')$ randomly, each observed image in the dataset is then a random transformation of its corresponding template function. This means we observe n independently generated pairs, each consisting of a $d \times d$ image and its corresponding class label. In the case of a supervised binary classification problem, these pairs are denoted by $(\mathbf{X}_i, k_i) \in [0, \infty)^{d \times d} \times \{0, 1\}$. Here $k_i \in \{0, 1\}$ is the i th label and the i th image is represented by a $d \times d$ matrix $\mathbf{X}_i = (X_{j,\ell}^{(i)})_{j,\ell=1,\dots,d}$ with entries

$$(2) \quad X_{j,\ell}^{(i)} = d^2 \eta_i \int_{I_{j,\ell}} f_{k_i}(\xi_i u - \tau_i, \xi'_i v - \tau'_i) du dv,$$

where f_0, f_1 are the two unknown template functions and $\eta_i, \xi_i, \xi'_i, \tau_i, \tau'_i$ are unobserved independent random variables. Each image consists of d^2 pixels. The brightness factor η_i is assumed to be positive. Throughout the article we assume that the template functions f_0, f_1 are nonnegative. This implies that all pixel values $X_{j,\ell}^{(i)}$ are also nonnegative.

In summary, model (2) generates images of the two objects using template functions f_0, f_1 , where the shifts τ, τ' , scaling factors ξ, ξ' , and brightness η are all random variables.

To extend model (2), we introduce a more general framework in which the random transformations are deformations $A = (a_1, a_2) : \mathbb{R}^2 \rightarrow \mathbb{R}^2$, belonging to a class of mappings $\mathcal{A}$. In this generalized model, the deformed template function is expressed as

$$f \circ A(u, v) := f(A(u, v)) = f(a_1(u, v), a_2(u, v)),$$

with $A \in \mathcal{A}$. Let A_i denote the deformation applied to the i th image in the sample. The image can then be represented by a $d \times d$ matrix $\mathbf{X}_i = (X_{j,\ell}^{(i)})_{j,\ell=1,\dots,d}$ with entries

$$(3) \quad X_{j,\ell}^{(i)} = d^2 \eta_i \int_{I_{j,\ell}} f_{k_i} \circ A_i(u, v) du dv.$$

Since each image can be viewed as an observation of a randomly deformed function, the proposed framework can be interpreted as a functional data analysis model adapted to image classification. This connection, along with its distinctions, will be further discussed in Section 6. For more on classification for functional data, see [18, 28, 52, 54, 67].

We now present some examples of specific deformation models. While these examples are parametric, the framework also allows for nonparametric models; see Section 3.

Affine transformations. Affine transformations have been widely discussed in the fields of image processing and computer vision; see, for example, [34, 63]. The deformation is of the form

$$(4) \quad A(u, v) = \begin{pmatrix} b_1 & b_2 \\ b_3 & b_4 \end{pmatrix} \begin{pmatrix} u \\ v \end{pmatrix} - \begin{pmatrix} \tau \\ \tau' \end{pmatrix},$$

with real parameters $b_1, \dots, b_4, \tau, \tau'$. We recover model (2) as a special case by choosing deformations $b_1 = \xi$, $b_4 = \xi'$, and $b_2 = b_3 = 0$. Moreover, rotated, scaled and translated images $\mathbf{X}_i = (X_{j,\ell}^{(i)})_{j,\ell=1,\dots,d}$ with brightness adjustment can be described by composing a scaling in x - and y -direction with a rotation by an angle $\gamma \in [0, 2\pi)$, that is, choosing

$$(5) \quad \begin{pmatrix} b_1 & b_2 \\ b_3 & b_4 \end{pmatrix} = \begin{pmatrix} \xi \cos \gamma & -\xi' \sin \gamma \\ \xi \sin \gamma & \xi' \cos \gamma \end{pmatrix} = \begin{pmatrix} \cos \gamma & -\sin \gamma \\ \sin \gamma & \cos \gamma \end{pmatrix} \begin{pmatrix} \xi & 0 \\ 0 & \xi' \end{pmatrix}.$$

Nonlinear deformations. Choosing $a_1(u, v) = u - \tau(u, v)$ and $a_2(u, v) = v - \tau'(u, v)$ for bivariate Lipschitz-continuous functions $\tau(u, v)$ and $\tau'(u, v)$ generates a class of nonlinear and local deformations [12, 13, 41]. Of particular interest among nonlinear deformations are wave-like deformations

$$a_1(u, v) = u + \alpha \sin(2\pi v/\lambda), \quad \text{and} \quad a_2(u, v) = v,$$

which are used to model periodic textures, noise patterns, image warping, and spatial distortions such as those caused by lens aberrations [23, 48, 61]. The parameter α describes the amplitude of the wave-like deformation, and $\lambda \neq 0$ controls the wavelength.

3. Classification via inverse mapping and image alignment. We construct and analyze classifiers that are specifically tailored to the proposed image deformation models (2) and (3).

3.1. Classification via inverse mapping. Under the general deformation model (3), each image $\mathbf{X} = (X_{j,\ell})_{j,\ell=1,\dots,d}$ is generated by

$$X_{j,\ell} = d^2 \eta \int_{I_{j,\ell}} f \circ A(u, v) du dv,$$

with f the template function and A the transformation modeling the deformation. Throughout this section we assume A is invertible.

To define the classifier, we also interpret an image $\mathbf{X}$ as a bivariate function on $\mathbb{R}^2$, via

$$(6) \quad \mathbf{X}(u, v) := \sum_{j,\ell \in \mathbb{Z}} X_{j,\ell} \mathbb{1}((u, v) \in I_{j,\ell})$$

for all $(u, v) \in \mathbb{R}^2$, and setting $X_{j,\ell} := 0$ if $j, \ell \notin \{1, \dots, d\}$. This means that $\mathbf{X}$, viewed as a function, assigns to any point within the pixel its corresponding pixel value. As the random deformations $A \in \mathcal{A}$ do not contain information about the class label, a classifier should not depend on these deformations. To achieve this, we consider the set of inverse transformations $\mathcal{A}^{-1} = \{A^{-1} : A \in \mathcal{A}\}$. For computational feasibility, instead of using $\mathcal{A}^{-1}$ directly, we approximate it by a discretized subset $\mathcal{A}_d^{-1}$, which covers $\mathcal{A}^{-1}$ with balls of radius $1/d$ on a given domain $D_{\mathcal{A}}$, meaning that, for any $A^{-1} \in \mathcal{A}^{-1}$, there exists a transformation $B \in \mathcal{A}_d^{-1}$ such that

$$(7) \quad \|A^{-1} - B\|_{L^\infty(D_{\mathcal{A}})} \leq \frac{1}{d}.$$

To obtain theoretical guarantees, one needs to choose the domain $D_{\mathcal{A}}$ sufficiently large and depending on the regularity conditions imposed on the deformation class.

To construct the classifier, we apply each transformation $B \in \mathcal{A}_d^{-1}$ to the input of the bivariate image function and obtain

$$(8) \quad \mathbf{X} \circ B(u, v) := \mathbf{X}(B(u, v)) = \sum_{j, \ell \in \mathbb{Z}} X_{j, \ell} \mathbb{1}(B(u, v) \in I_{j, \ell}) \quad \text{for all } (u, v) \in \mathbb{R}^2.$$

Given that any possible deformation A is invertible, there exists an approximate inverse $B \in \mathcal{A}_d^{-1}$ such that $B \approx A^{-1}$. For such a mapping B , $\mathbf{X} \circ B$ should generate a nearly deformation-free representation of the image. To account for the effects of the image brightness factor η , we normalize the pixel values and obtain

$$(9) \quad T_{\mathbf{X} \circ B} := \frac{\mathbf{X} \circ B}{\|\mathbf{X} \circ B\|_{L^2(\mathbb{R}^2)}}.$$

Combining these steps, the proposed classifier $\hat{k}$ assigns a label to the new image $\mathbf{X}$ by first applying all possible transformations from $\mathcal{A}_d^{-1}$ to both, the new image and each training image $\mathbf{X}_i$. It then finds the training image whose transformed version $T_{\mathbf{X}_i \circ B_i}$ is in Euclidean distance the closest fit to the transformed version $T_{\mathbf{X} \circ B}$ of the new image. The label of this closest matching training image is assigned to the new image. This defines the *inverse mapping classifier*

$$(10) \quad \hat{k} := k_{\hat{i}} \quad \text{with } \hat{i} \in \arg \min_{i \in \{1, \dots, n\}} \min_{B_i, B \in \mathcal{A}_d^{-1}} \|T_{\mathbf{X}_i \circ B_i} - T_{\mathbf{X} \circ B}\|_{L^2(\mathbb{R}^2)}.$$

It can be interpreted as a one-nearest neighbor estimator in a transformed space.

We now state the assumptions for the statistical analysis. To make the theory tractable, we impose a Lipschitz condition on the template function.

ASSUMPTION 1. The supports of the two template functions f_0, f_1 are contained in a rectangle $[\beta_{\text{left}}, \beta_{\text{right}}] \times [\beta_{\text{down}}, \beta_{\text{up}}] \subseteq [0, 1]^2$. Additionally, f_0, f_1 are Lipschitz continuous, in the sense that there exists a positive constant C_L such that, for any real numbers u, v, u', v' ,

$$(11) \quad |f_k(u, v) - f_k(u', v')| \leq C_L \|f_k\|_1 (|u - u'| + |v - v'|), \quad k = 0, 1.$$

We also impose conditions to prevent the deformation from moving (part of) the object outside the image. Assumption 1 ensures that the support of f is contained in $[\beta_{\text{left}}, \beta_{\text{right}}] \times [\beta_{\text{down}}, \beta_{\text{up}}] \subseteq [0, 1]^2$. The deformed object remains fully visible, if the support of the deformed function $f \circ A$ also lies in $[0, 1]^2$. This is the case if $[\beta_{\text{left}}, \beta_{\text{right}}] \times [\beta_{\text{down}}, \beta_{\text{up}}] \subseteq A([0, 1]^2)$.

ASSUMPTION 2. (i). The class $\mathcal{A}$ contains the identity. For any $A = (a_1, a_2) \in \mathcal{A}$, $[\beta_{\text{left}}, \beta_{\text{right}}] \times [\beta_{\text{down}}, \beta_{\text{up}}] \subseteq A([0, 1]^2)$, and the functions a_1, a_2 have continuous partial derivatives on $\mathbb{R}^2$, bounded in the supremum norm by a constant C_A . (ii). Assume $C_{\mathcal{J}}^2 := \inf_{A \in \mathcal{A}} \inf_{(u, v) \in \mathbb{R}^2} |\det(J_A(u, v))| > 0$ with J_A the Jacobian matrix of A .

To ensure correct classification, we also need to guarantee that two images from different object classes cannot be represented as transformations of the template function corresponding to the opposite class, as otherwise, distinguishing between the two classes becomes impossible.

To formalize this separation between the two object classes with template function f_0, f_1 , we introduce the separation quantity $D := D(f_0, f_1) \vee D(f_1, f_0)$ with

$$(12) \quad D(f, g) := \frac{\inf_{a \in \mathbb{R}, \{A_i\}_{i=1}^4 \subseteq \mathcal{A}} \|af \circ A_1 \circ A_2^{-1} - g \circ A_3 \circ A_4^{-1}\|_{L^2(\mathbb{R}^2)}}{\|g\|_{L^2(\mathbb{R}^2)}}.$$

The quantity $D(f, g)$ measures the normalized minimal L^2 -distance between all possible deformations of g and f under transformations from $\mathcal{A} \circ \mathcal{A}^{-1}$. When $\mathcal{A}$ forms a group, this expression simplifies to $D(f, g) = \inf_{a \in \mathbb{R}, A' \in \mathcal{A}} \|af \circ A - g \circ A'\|_{L^2(\mathbb{R}^2)} / \|g\|_{L^2(\mathbb{R}^2)}$, measuring the normalized minimal L^2 -distance between all possible deformations of g and f under transformations from $\mathcal{A}$.

THEOREM 3.1. *Let $(\mathbf{X}, k), (\mathbf{X}_1, k_1), \dots, (\mathbf{X}_n, k_n)$ be defined as in (3). Suppose that the labels 0 and 1 occur at least once in the training data, that is, $\{i : k_i = 0\} \neq \emptyset$ and $\{i : k_i = 1\} \neq \emptyset$. Assume moreover that f_0 and f_1 satisfy Assumption 1 with Lipschitz constant C_L and that Assumption 2 holds with constants $C_A, C_{\mathcal{J}}$. If $D_A = [-2C_A - 1, 2C_A + 1]^2$ in (7) and*

$$(13) \quad D > C(C_L, C_A, C_{\mathcal{J}})/d,$$

where D is as defined in (12), and $C(C_L, C_A, C_{\mathcal{J}})$ is a sufficiently large constant only depending on $C_L, C_A, C_{\mathcal{J}}$, then the classifier $\hat{k}$ defined in (10) will recover the correct label, that is,

$$\hat{k} = k.$$

The proof of Theorem 3.1 is deferred to Section A.1 of the Supplementary Material [14]. The result shows that classifier (10) guarantees perfect classification of any given image if the separation quantity satisfies $D > C/d$ for a sufficiently large constant C , and the dataset contains at least one image from each class. As $d \rightarrow \infty$, $C/d \rightarrow 0$ and the condition $D > C/d$ typically holds for all sufficiently large d . This aligns with the intuition that a minimal image resolution is necessary for classification. The following result provides a simple tool to check condition (13) if $\mathcal{A}$ is a group (see also Figure 2). The proof is given in Section A.1 of the Supplementary Material [14].

LEMMA 3.2. *Let f_0, f_1 be two template functions, and let the deformation set $\mathcal{A}$ be a group satisfying Assumption 2(i), with constant C_A . Then condition (13) is satisfied, whenever*

$$d > \sqrt{2}C(C_L, C_A, C_{\mathcal{J}})C_A \sup_{A \in \mathcal{A}} \frac{\|f_1\|_{L^2(\mathbb{R}^2)}}{\|f_1\|_{L^2((\text{supp}(f_0 \circ A))^c)}}.$$

Separation under distance D is different from the Euclidean distance-based criterion that is typically employed in the 1-nearest neighbor method. The Euclidean distance is significantly

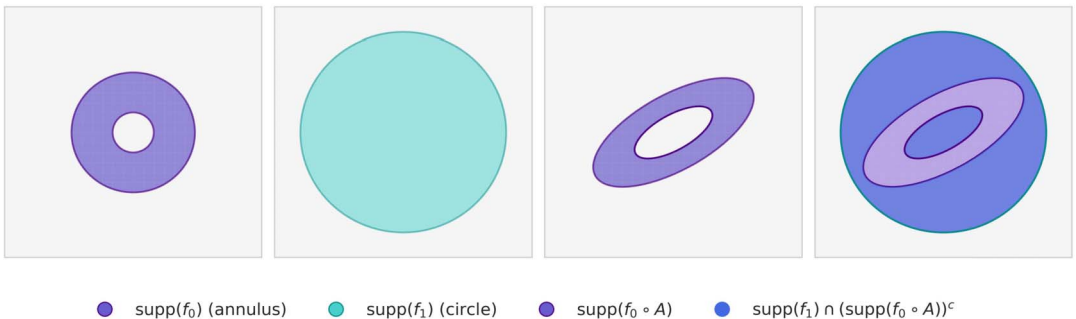


FIG. 2. Illustration of why the separation distance can be bounded from below, as in Lemma 3.2, if deformations of the template functions cannot map the support on each other.

more rigid than the proposed distance metric D , making it inherently less suitable for classification in deformation-sensitive contexts. If, for example, the deformation set $\mathcal{A}$ includes random shifts, then two shifted functions $f_i \circ A$ and $f_i \circ A'$ based on the same template function f_i will have a small D -distance, but their Euclidean distance may be large if A and A' shift in different directions.

The classifier can account for a wide range of image deformations, but discretization of the entire set of inverse mappings can be computationally demanding. Therefore, we consider this classifier more as a theoretical benchmark for the general image deformation model (3) rather than an effective method for practical applications. When dealing with specific deformation models, the computational demands can be significantly reduced. For instance, if the deformation class $\mathcal{A}$ is a group, then $\mathcal{A}^{-1} = \mathcal{A}$, and we can instead consider the classifier

$$(14) \quad \hat{k} := k_{\hat{i}} \quad \text{with } \hat{i} \in \arg \min_{i \in \{1, \dots, n\}} \min_{A \in \mathcal{A}_d} \|T_{\mathbf{x}_i} - T_{\mathbf{x} \circ A}\|_{L^2(\mathbb{R}^2)},$$

where $\mathcal{A}_d$ denotes a $1/d$ -covering of $\mathcal{A}$.

This classifier resembles traditional registration techniques in medical imaging and computer vision. Given a (moving) source image S and a target image T , the goal of image registration is to find a map $\phi : \mathbb{R}^2 \rightarrow \mathbb{R}^2$ such that $S \circ \phi(\mathbf{x}) \approx T(\mathbf{x})$ [15, 58]. This is typically formulated as an optimization problem

$$(15) \quad \hat{\phi} = \arg \min_{\phi} L(S \circ \phi, T) + R(\phi),$$

where L is a loss function measuring similarity and $R(\cdot)$ is a regularizer. The regularizer can be omitted for low-dimensional transformation models, such as rigid or affine transformations [70].

In Theorem 3.8 we show that if the set $\mathcal{A}$ is sufficiently rich, the considered separation criterion $D \gtrsim 1/d$, as defined in (12), is optimal. This means that any smaller bound could result in deformed versions of one template function being representable by a template function of the opposite class, thereby making it impossible to distinguish between the two classes; see Figure 3 for an illustration of two deformed MNIST images with labels 7 and 1 that can be transformed into each other by a rotation, making classification ambiguous.

We now verify the imposed assumptions for the specific deformation models introduced in Section 2, with proofs of the following lemmas provided in Section A.1 of the Supplementary Material [14]. Let $y_+ := \max\{y, 0\}$.

LEMMA 3.3. *Let $[\beta_{\text{left}}, \beta_{\text{right}}] \times [\beta_{\text{down}}, \beta_{\text{up}}] = [1/4, 3/4] \times [1/4, 3/4]$. The class of deformations in (2) with $1/2 \leq |\xi|, |\xi'| \leq C_{\mathcal{A}}, |\tau|, |\tau'| \leq \ell_s$,*

$$(16) \quad -(-\xi)_+ - \frac{1}{4} \leq \tau \leq \xi_+ - \frac{3}{4} \quad \text{and} \quad -(-\xi')_+ - \frac{1}{4} \leq \tau' \leq \xi'_+ - \frac{3}{4},$$

satisfies Assumption 2 with constants $C_{\mathcal{A}}$ and $C_{\mathcal{T}} = 1/2$. Moreover, there exists a $1/d$ -covering of $\mathcal{A}^{-1}$ with cardinality $|\mathcal{A}_d^{-1}| \asymp d^4$.

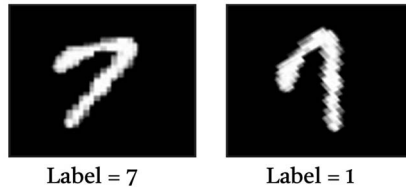


FIG. 3. Deformed MNIST images of the digits 7 and 1.

LEMMA 3.4. *The deformation model described in (4) and (5) is*

$$A(u, v) = \begin{pmatrix} \cos \gamma & -\sin \gamma \\ \sin \gamma & \cos \gamma \end{pmatrix} \begin{pmatrix} \xi & 0 \\ 0 & \xi' \end{pmatrix} \begin{pmatrix} u \\ v \end{pmatrix} - \begin{pmatrix} \tau \\ \tau' \end{pmatrix}.$$

Let $[\beta_{\text{left}}, \beta_{\text{right}}] \times [\beta_{\text{down}}, \beta_{\text{up}}] = [1/4, 3/4] \times [1/4, 3/4]$. The class of deformations with $1/2 \leq |\xi|, |\xi'| \leq C_{\mathcal{A}}, |\tau|, |\tau'| \leq \ell_s, \gamma \in [0, \pi/2)$, and

$$\begin{aligned} -(-\xi)_+ - \frac{1}{4}(\cos \gamma + \sin \gamma) &\leq \tau \cos \gamma + \tau' \sin \gamma \leq \xi_+ - \frac{3}{4}(\cos \gamma + \sin \gamma), \\ -(-\xi')_+ - \frac{1}{4}\cos \gamma + \frac{3}{4}\sin \gamma &\leq \tau' \cos \gamma - \tau \sin \gamma \leq \xi'_+ - \frac{3}{4}\cos \gamma + \frac{1}{4}\sin \gamma, \end{aligned}$$

satisfies Assumption 2 with constants $C_{\mathcal{A}}, C_{\mathcal{J}} = 1/2$ and there exists a $1/d$ -covering of $\mathcal{A}^{-1}$ with $|\mathcal{A}_d^{-1}| \asymp d^5$.

Consider the nonlinear deformations $A = (a_1, a_2) \in \mathcal{A}$,

$$(17) \quad a_1(u, v) = h_1(u, v) \quad \text{and} \quad a_2(u, v) = h_2(v),$$

where $h_2(v)$ is strictly monotone with respect to v , and for any fixed $v \in \mathbb{R}$, $h_1(u, v)$ is strictly monotone with respect to u . In this case the transformation is invertible because one can always retrieve v from $h_2(v)$ and then u from $h_1(u, v)$. A specific example is the wave-like deformation model discussed in Section 2. The following describes the construction of $\mathcal{A}_d^{-1}$.

LEMMA 3.5. *Let $[\beta_{\text{left}}, \beta_{\text{right}}] \times [\beta_{\text{down}}, \beta_{\text{up}}] = [1/4, 3/4] \times [1/4, 3/4]$. For the class of deformations*

$$A(u, v) = (a_1(u, v), a_2(u, v)) = (u + \alpha \sin(2\pi v/\lambda), v),$$

with $|\lambda| \geq C_{\text{lower}} > 0$, and $|\alpha| \leq 1/4$, Assumption 2 holds with $C_{\mathcal{A}} = \max\{\pi/(2C_{\text{lower}}), 1\}$ and $C_{\mathcal{J}} = 1$. Moreover, there exists a $1/d$ -covering $\mathcal{A}_d^{-1}$ of $\mathcal{A}^{-1}$ with cardinality $|\mathcal{A}_d^{-1}| \asymp d^2$.

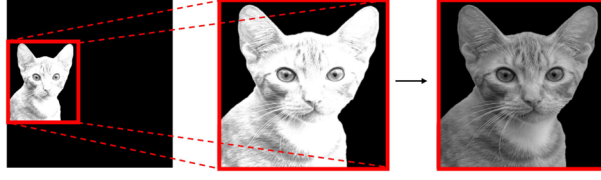
For the general nonlinear model (17), the set $\mathcal{A}_d^{-1}$ can be constructed analogously to Lemma 3.5. Specifically, if $a_1 \in \mathcal{F}_1$, $a_2 \in \mathcal{F}_2$, with $\mathcal{F}_1^\delta \subseteq \mathcal{F}_1$ and $\mathcal{F}_2^\delta \subseteq \mathcal{F}_2$ denoting δ -coverings with respect to the sup-norm, then

$$\mathcal{A}_\delta := \{(a_1, a_2) : a_1 \in \mathcal{F}_1^\delta, a_2 \in \mathcal{F}_2^\delta\}$$

forms a δ -covering of $\mathcal{A} = \{(a_1, a_2) : a_1 \in \mathcal{F}_1, a_2 \in \mathcal{F}_2\}$. If for all $A = (a_1, a_2) \in \mathcal{A}$, we have $|\partial_u a_1(u, v)|, |\partial_v a_2(u, v)| \geq \mathcal{K} > 0$, then $\mathcal{A}^{-1}$ admits a $\delta' = \delta/\mathcal{K}$ -covering formed by inverting the elements of $\mathcal{A}_\delta$.

The next lemma considers compositions of deformation classes, with the proof provided in Section A.1 of the Supplementary Material [14]. This allows to verify Assumption 2 for more involved deformation classes.

LEMMA 3.6. *Let $\mathcal{A}_1$ and $\mathcal{A}_2$ satisfy Assumption 2 with respective constants $C_{\mathcal{A}_1}, C_{\mathcal{J}_1}$ and $C_{\mathcal{A}_2}, C_{\mathcal{J}_2}$. If for any $A_1 \in \mathcal{A}_1$, $[0, 1]^2 \subseteq A_1([0, 1]^2)$, then the composite deformation class $\mathcal{A}_2 \circ \mathcal{A}_1 := \{A_2 \circ A_1 : A_1 \in \mathcal{A}_1, A_2 \in \mathcal{A}_2\}$ satisfies Assumption 2 with constants $C_{\mathcal{A}} = 2C_{\mathcal{A}_1}C_{\mathcal{A}_2}$ and $C_{\mathcal{J}} = C_{\mathcal{J}_1}C_{\mathcal{J}_2}$.*

FIG. 4. $\mathbf{X}$, $Z_{\mathbf{X}}$, and $T_{\mathbf{X}}$.

3.2. *Classification via image alignment.* We now focus on the specific image deformation model (2) that incorporates random scaling, shifts, and brightness adjustment. An image $\mathbf{X} = (X_{j,\ell})_{j,\ell=1,\dots,d}$ is then generated by

$$(18) \quad X_{j,\ell} = d^2 \eta \int_{I_{j,\ell}} f(\xi u - \tau, \xi' v - \tau') du dv,$$

with $(\eta, \xi, \xi', \tau, \tau')$ the random deformation parameters. In this setting, one can find a transformation that aligns the images, in the sense that the transformed images are nearly independent of the deformation parameters. We propose a one-nearest-neighbor classifier based on the aligned training and test images. This approach is similar to curve registration in functional data analysis; see, for instance, [43]. The classifier can be efficiently computed but relies on this specific deformation model.

The first step of the construction is to approximately detect the object within the image by identifying the smallest axis-aligned rectangle that contains all nonzero pixel values; see the left image in Figure 4 for an illustration. We refer to this as the *rectangular support*. To determine the rectangular support, we denote the smallest and largest indices corresponding to the nonzero pixels in the image by

$$(19) \quad j_- := \arg \min\{j : X_{j,\ell} > 0\}, \quad j_+ := \arg \max\{j : X_{j,\ell} > 0\}$$

and

$$(20) \quad \ell_- := \arg \min\{\ell : X_{j,\ell} > 0\}, \quad \ell_+ := \arg \max\{\ell : X_{j,\ell} > 0\}.$$

The rectangular support of the image is then given by the rectangle $[j_-/d, j_+/d] \times [\ell_-/d, \ell_+/d]$. Similarly, we define the rectangular support of a function as the smallest rectangle containing the support. From the definition of the model (2), it follows that the rectangular support of the image $\mathbf{X}$ should be close to the rectangular support of the underlying deformed function $f(\xi \cdot - \tau, \xi' \cdot - \tau')$.

We now rescale the rectangular support of the image to the unit square $[0, 1]^2$. The line $[0, 1] \ni t \mapsto j_- + t(j_+ - j_-)$ starts at j_- and ends for $t = 1$ at j_+ . We define the rescaled pixel values as

$$(21) \quad Z_{\mathbf{X}}(t, t') := X_{\lfloor j_- + t(j_+ - j_-) \rfloor, \lfloor \ell_- + t'(\ell_+ - \ell_-) \rfloor},$$

with $\lfloor \cdot \rfloor$ the floor function. The function $(t, t') \mapsto Z_{\mathbf{X}}(t, t')$ runs through the pixel values on the rectangular support, now rescaled to the unit square $[0, 1]^2$; see the middle image of Figure 4 for an illustration. This rescaling makes $Z_{\mathbf{X}}(t, t')$ approximately invariant to random shifts and scalings of the image, up to smaller-order effects.

To find a quantity that is independent of the brightness adjustment η , we normalize the pixel values by $Z_{\mathbf{X}}/\|Z_{\mathbf{X}}\|_2$. The image alignment transformation is then given by

$$(22) \quad T_{\mathbf{X}} := \frac{Z_{\mathbf{X}}}{\|Z_{\mathbf{X}}\|_2};$$

see again Figure 4 for an illustration. Based on these aligned and normalized images, we study a one-nearest-neighbor classifier $\widehat{k}$ that assigns the label of $\mathbf{X}_i$ from the training set to $\mathbf{X}$, where $T_{\mathbf{X}_i}$ is the closest to $T_{\mathbf{X}}$. The *image alignment classifier* is defined as

$$(23) \quad \widehat{k} := k_{\widehat{i}} \quad \text{with } \widehat{i} \in \arg \min_{i \in \{1, \dots, n\}} \|T_{\mathbf{X}} - T_{\mathbf{X}_i}\|_2.$$

This is an interpolating classifier, in the sense that if the new $\mathbf{X}$ coincides with one of the images in the training set $\mathbf{X}_i$, then $T_{\mathbf{X}} = T_{\mathbf{X}_i}$, $\widehat{i} = i$, and $\widehat{k} = k_i$.

To study this model, we assume that $\xi, \xi' \geq 1/2$. Applying Lemma 3.3 leads to the following assumption on the parameters to ensure full visibility of the objects on the deformed images.

ASSUMPTION 2'. The supports of the two template functions f_0, f_1 are contained in $[1/4, 3/4]^2$, and the random parameters (τ, τ', ξ, ξ') satisfy $\xi, \xi' \geq 1/2$,

$$-\frac{1}{4} \leq \tau \leq \xi - \frac{3}{4}, \quad \text{and} \quad -\frac{1}{4} \leq \tau' \leq \xi' - \frac{3}{4}.$$

Assumption 2' indicates that the range of possible shifts τ, τ' increases as ξ, ξ' become larger. This is reasonable, as larger values of ξ, ξ' shrink the object. Consequently, larger shifts τ, τ' can be applied without moving parts of the object out of the image.

Under the deformation model (2), we have $f \circ A(u, v) = f(\xi u - \tau, \xi' v - \tau')$. Based on this, we consider the separation quantity $D = D(f_0, f_1) \vee D(f_1, f_0)$ with

$$(24) \quad D(f, g) := \frac{\inf_{a, b, c, b', c' \in \mathbb{R}} \|af(b \cdot -c, b' \cdot -c') - g\|_{L^2(\mathbb{R}^2)}}{\|g\|_{L^2(\mathbb{R}^2)}}.$$

It measures the normalized minimal L^2 -distance between the function g and any potential deformation of the function f due to rescaling, shifting, and change in brightness.

THEOREM 3.7. Let $(\mathbf{X}, k), (\mathbf{X}_1, k_1), \dots, (\mathbf{X}_n, k_n)$ be defined as in (2). Suppose that the labels 0 and 1 occur at least once in the training data, that is, $\{i : k_i = 0\} \neq \emptyset$ and $\{i : k_i = 1\} \neq \emptyset$. Assume moreover that f_0 and f_1 satisfy Assumption 1 with Lipschitz constant C_L and that Assumption 2' holds. Set $\Xi_n := \max\{1, \xi, \xi', \xi_1, \xi'_1, \dots, \xi_n, \xi'_n\}$. If $D > 4K(C_L \vee C_L^2)\Xi_n^2/d$, with D as defined in (24), and K the universal constant in Lemma A.9, then the classifier $\widehat{k}$ as defined in (23) will recover the correct label, that is,

$$\widehat{k} = k.$$

The proof of Theorem 3.7 is postponed to Section A.2 of the Supplementary Material [14]. The result indicates that, under proper conditions, the classifier accurately identifies the label when the template functions f_0 and f_1 are separated by $\gtrsim 1/d$ in L^2 -norm, consistently across all conceivable image deformations. This finding aligns with the theoretical performance outlined in Theorem 3.1 for the general classification approach. The advantage of implementing the image alignment approach is that it eliminates the need to discretize the set of inverse mappings, thereby substantially improving computational efficiency.

We further prove a corresponding lower bound, showing that a $1/d$ -rate in the separation criterion is necessary. Without this condition, the same image could be represented by deformations of both template functions, making classification impossible.

THEOREM 3.8. *For any τ, τ', ξ, ξ' satisfying Assumption 2', there exist nonnegative Lipschitz continuous functions f_0, f_1 with Lipschitz constants C_{f_0} and $C_{f_1} = C_{f_1}(\xi, \xi')$, respectively, such that, for any $d \geq 32(\xi \vee \xi')$,*

$$\frac{\|f_1 - f_0\|_{L^2(\mathbb{R}^2)}}{\|f_0\|_{L^2(\mathbb{R}^2)}} \geq \frac{1}{28d},$$

and the data generating model (18) can be written as

$$X_{j,\ell} = d^2 \eta \int_{I_{j,\ell}} f_1(\xi u - \tau, \xi' v - \tau') du dv = d^2 \eta \int_{I_{j,\ell}} f_0(\xi u - \tau, \xi' v - \tau') du dv.$$

Consequently, the same pixel values are generated under both classes.

The proof of Theorem 3.8 is deferred to Section A.2 of the Supplementary Material [14], which shows that the separation rate $1/d$ arises from the Lipschitz continuity of f_0 and f_1 . If, instead, we assume Hölder regularity with index $\beta \leq 1$, we expect the lower bound to be of order $d^{-\beta}$, which we also conjecture to be the optimal separation rate in this case.

Since the deformation model (2) is a specific case of the general model (3), the lower bound derived in Theorem 3.8 also applies to the general deformation model (3). This indicates that the rate $1/d$ is indeed necessary to distinguish between the two classes.

In the presence of background noise, finding the rectangular support of the object is hard, as nonzero pixel values in the image may belong to the background. As an alternative, one could instead rely on t -level sets $\{\mathbf{x} : g(\mathbf{x}) > t\}$ and define the t -rectangular support as the smallest rectangular containing the t -level set. For nonnegative g , the previously introduced rectangular support corresponds to $t = 0$. To construct a classifier, we can first normalize the pixel values to eliminate the brightness factor η , then follow a similar strategy as in the zero-background case by determining the t -rectangular support for each image in the dataset. While increasing t enhances robustness to background noise, it also reduces the t -rectangular support and causes larger constants in the separation condition between the two classes.

If an image contains multiple nonoverlapping objects, we suggest to first apply an image segmentation method (see, e.g., [26, 45]) to isolate each object. The image alignment classifier can then be applied to each segment separately.

The analysis of both classifiers generalizes to the multiclass case with K classes, provided that the label of each class appears at least once in the training data, and the separation quantity between each pair of class templates, defined as $D_{i,j} := D(f_i, f_j) \vee D(f_j, f_i)$, satisfies $D_{i,j} \gtrsim 1/d$ for all $i, j = 1, \dots, K$ with $i \neq j$. Under the conditions outlined above and Assumptions 1 (adapted to the multiclass setting) and 2, the classifier will correctly recover the class labels.

The image alignment step in the construction of the classifier leads to a representation of the image that is, up to discretization effects, independent to rescaling and shifting of the object; see Figure 4. While the proposed image alignment transformation is natural and mathematically tractable for this specific deformation model, other transformations could be employed instead, such as Fourier transform, Radon transform, and scattering transform [12, 13, 41].

4. Classification with convolutional neural networks. Convolutional neural networks (CNNs) have achieved remarkable practical success, particularly in the context of image recognition [35, 36, 53, 55]. In this section we analyze the performance of CNN-based classifiers within the framework of the general deformation model (3), introduced in Section 2. We begin by introducing the mathematical notation to formalize the structure of a CNN. Here we focus on a particular CNN structure and refer to [53, 68] for a broader introduction.

4.1. Convolutional neural networks. We analyse a CNN with a rectified linear unit (ReLU) activation function and a softmax output layer. Generally, a CNN consists of three fundamental components: Convolutional, pooling, and fully connected layers. The input to a CNN is a $d \times d$ matrix representing the pixel values of an image. In the convolutional layer, so-called filters (i.e., weight matrices of predefined size) slide across the image, performing convolutions at each spatial location. Finally, an elementwise nonlinear activation function $\sigma : \mathbb{R} \rightarrow \mathbb{R}$, in our case the ReLU function, is applied to the outcome of the convolutions, producing the output matrices known as feature maps.

In this work we consider CNNs with a single convolutional layer followed by one pooling layer. For mathematical simplicity, we introduce a compact notation tailored to our setting and refer to [32, 33] for a general mathematical definition. Recall that the input to the network is an image represented by a $d \times d$ matrix $\mathbf{X}$. For a $d \times d$ matrix $\mathbf{W}$, we define its *quadratic support* $[\mathbf{W}]$ as the smallest square submatrix of $\mathbf{W}$ that contains all its nonzero entries. For instance,

$$[\mathbf{W}] = \begin{pmatrix} 1 & 1 \\ 0 & 0 \end{pmatrix}$$

is the quadratic support of the matrix

$$\mathbf{W} = \begin{pmatrix} 0 & 1 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}.$$

In this context, $[\mathbf{W}]$ represents the network filter. To describe the action of the filter on the image, denoted as $[\mathbf{W}] \star \mathbf{X}$, assume that $[\mathbf{W}]$ is a filter of size $\ell \in \{1, \dots, d\}$. We extend the matrix $\mathbf{X}$ by padding it with zero matrices on all sides. Specifically, we define the enlarged matrix as

$$\mathbf{X}' := \begin{bmatrix} 0_{\ell \times \ell} & 0_{\ell \times d} & 0_{\ell \times \ell} \\ 0_{d \times \ell} & \mathbf{X} & 0_{d \times \ell} \\ 0_{\ell \times \ell} & 0_{\ell \times d} & 0_{\ell \times \ell} \end{bmatrix},$$

where $0_{j \times k}$ denotes a $j \times k$ zero matrix. The (i, j) th patch is defined as the $\ell \times \ell$ submatrix $\mathbf{X}'_{i,j} := (X'_{i+a, j+b})_{a,b=0, \dots, \ell-1}$. We further define $([\mathbf{W}] \star \mathbf{X})_{i,j}$ as the entrywise sum of the Hadamard product of $[\mathbf{W}]$ and $\mathbf{X}'_{i,j}$. Thus, the matrix $[\mathbf{W}] \star \mathbf{X}$ contains all entrywise sums of the Hadamard product of $[\mathbf{W}]$ with all patches. Finally the ReLU activation function $\sigma(x) = \max\{x, 0\}$ is applied elementwise. A feature map can then be expressed as

$$\sigma([\mathbf{W}] \star \mathbf{X}).$$

This extension of the matrix $\mathbf{X}$ to $\mathbf{X}'$ is a form of zero padding, which ensures that the in-plane dimension of the input remains equal after convolution [27]. A pooling layer is typically applied to the feature map. While max-pooling extracts the maximum value from each patch of the feature map, average-pooling computes the average over each patch. In this work we consider CNNs with *global* max-pooling in the sense that the max-pooling extracts from every feature map $\sigma([\mathbf{W}] \star \mathbf{X})$ the largest absolute value. The feature map after global max-pooling is then given by

$$\mathbf{O}(\mathbf{X}) = |\sigma([\mathbf{W}] \star \mathbf{X})|_{\infty}.$$

For k filters described by the matrices $\mathbf{W}_1, \dots, \mathbf{W}_k$, we obtain the k values

$$(25) \quad \mathbf{O}_s(\mathbf{X}) = |\sigma([\mathbf{W}_s] \star \mathbf{X})|_{\infty}, \quad s = 1, \dots, k.$$

For $\alpha \in (0, 1)$, define $\mathcal{K}(i) := \{[d^{1-\alpha} + 1](i-1) + 1, \dots, ([d^{1-\alpha} + 1]i) \wedge d\}$. We say that a $d \times d$ filter matrix $\mathbf{W} = (W_{j,\ell})_{j,\ell=1, \dots, d}$ has an (α, d) -block structure if $W_{j,\ell} = W_{j',\ell'}$

whenever $j, j' \in \mathcal{K}(i)$ and $\ell, \ell' \in \mathcal{K}(i')$. By convention, any $d \times d$ matrix $\mathbf{W}$ is said to have a $(1, d)$ -block structure. An illustration of the (α, d) -block structure is shown in Figure 3 of the Supplementary Material [14]. For $0 < \alpha \leq 1$, we denote by

$$(26) \quad \mathcal{F}^C(\alpha, k)$$

the class of all CNN layers computing k outputs of the form (25), where each filter matrix $\mathbf{W}_s$ has an (α, d) -block structure and all parameters take values in the interval $[-1, 1]$. The output of the last convolutional layer is flattened, this means, it is transformed into a vector before several fully connected layers with ReLU activation function are applied.

For any vector $\mathbf{v} = (v_1, \dots, v_r)^\top$, $\mathbf{y} = (y_1, \dots, y_r)^\top \in \mathbb{R}^r$, we define $\sigma_{\mathbf{v}}\mathbf{y} = (\sigma(y_1 - v_1), \dots, \sigma(y_r - v_r))^\top$. In the context of binary classification, the last layer of the network should extract a two-dimensional probability vector. To achieve this, the softmax function

$$(27) \quad \Phi(x_1, x_2) = \left(\frac{e^{x_1}}{e^{x_1} + e^{x_2}}, \frac{e^{x_2}}{e^{x_1} + e^{x_2}} \right)$$

is typically applied. A feedforward neural network with L fully connected hidden layers and width vector $\mathbf{m} = (m_0, \dots, m_{L+1}) \in \mathbb{N}^{L+2}$, where m_i denotes the number of hidden neurons in the i th hidden layer, can then be described by a function $f: \mathbb{R}^{m_0} \rightarrow \mathbb{R}^{m_{L+1}}$ with

$$\mathbf{x} \mapsto f(\mathbf{x}) = \psi \sigma_{\mathbf{v}_{L+1}} \mathbf{W}_L \sigma_{\mathbf{v}_L} \mathbf{W}_{L-1} \sigma_{\mathbf{v}_{L-1}} \cdots \mathbf{W}_1 \sigma_{\mathbf{v}_1} \mathbf{W}_0 \mathbf{x},$$

where $\mathbf{W}_j$ is a $m_j \times m_{j+1}$ weight matrix, $\mathbf{v}_j$ is the bias vector in layer j , and ψ is either the identity function $\psi = id$ or the softmax function $\psi = \Phi$. We consider the class of fully connected neural networks in which all entries of the weight matrices and bias vectors are bounded in absolute value by 1 and denote this class by

$$(28) \quad \mathcal{F}_\psi(L, \mathbf{m}).$$

We will construct CNN classifiers based on a CNN architecture of the form

$$(29) \quad \mathcal{G}(\alpha, m) := \{f \circ g : f \in \mathcal{F}_\Phi(1 + 2\lceil \log_2 m \rceil, (2m, 4m, \dots, 4m, 2)), g \in \mathcal{F}^C(\alpha, 2m)\},$$

with m a positive integer and $0 < \alpha \leq 1$. Given that we only consider one convolutional and one pooling layer, the number of feature maps equals the input dimension of the fully connected subnetwork.

As the filters are applied to all patches of the image, CNNs are translation-invariant, meaning that up to boundary and discretization effects, the CNN classifier does not depend on the values of the shifting parameters τ and τ' in the deformation model. For instance, if a cat in an image is moved from the upper-left corner to the lower-right corner, the convolutional filter will produce, up to discretization effects, the same feature values at potentially different locations within the feature map; see Figure 5 for an illustration. A shift of the image pixels causes, therefore, a permutation of the values in the feature map. Since the global max-pooling layer is invariant to permutations, the CNN output is thus invariant under translations.

More challenging for CNNs are varying object sizes and rotation angles. [22, 51, 65] argue that scale-invariance is undesirable in image classification, as classifiers can benefit from scale information of the object and that architectures can assign different filters to capture different scales. Similarly, [42] employs separate filters for different rotation angles, thereby achieving rotation invariance for texture classification. Data augmentation enhances the learning of CNNs in the presence of rotation and scale deformations at the expense of additional computational cost. Before training, data augmentation applies simple deformations, such as rotations and different scaling to the training images, and learns a CNN on the augmented

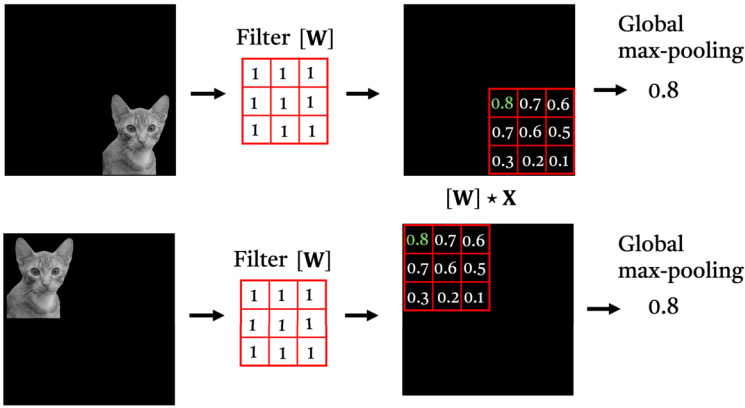


FIG. 5. Shift invariance of CNNs.

dataset consisting of the original and the transformed training samples (see, e.g., [56]). Group equivariant convolutional networks [16] extend CNNs to handle invariances induced by arbitrary groups. An essential part of the derived theory in the next section shows that, for rich classes of deformations, CNNs are expressive enough to separate deformed images.

4.2. Misclassification bounds for CNN-based classifiers. We suppose that the training data consists of n i.i.d. data points, generated as follows: For $\pi \in [0, 1]$ and each i , we draw a label $k_i \in \{0, 1\}$ from the Bernoulli distribution with success probability π . Let $\mathcal{Q}_{\mathcal{A}}$ be a distribution over the deformation class $\mathcal{A}$, and let $\mathcal{Q}_{\eta}$ be a distribution on $(0, \infty)$ for the random brightness factor. Here we assume that η and A are independent. The i th sample is $(\mathbf{X}_i, k_i)$, where $\mathbf{X}_i$ is an independent draw from the general model (3) with template function f_{k_i} , deformation $A_i \in \mathcal{A}$ generated from $\mathcal{Q}_{\mathcal{A}}$, and brightness factor η_i generated from $\mathcal{Q}_{\eta}$. The full dataset is denoted by

$$(30) \quad \mathcal{D}_n = ((\mathbf{X}_1, k_1), \dots, (\mathbf{X}_n, k_n)).$$

In expectation, the dataset consists of $n(1 - \pi)$ samples from class 0 and $n\pi$ samples from class 1.

The parameters of a CNN are then fitted to the normalized images

$$(31) \quad \bar{\mathbf{X}}_i = (\bar{X}_{j,\ell}^{(i)})_{j,\ell=1,\dots,d} \quad \text{with} \quad \bar{X}_{j,\ell}^{(i)} := \frac{X_{j,\ell}^{(i)}}{\sqrt{\sum_{j,\ell=1}^d (X_{j,\ell}^{(i)})^2}}.$$

This normalization can be viewed as preprocessing step. It ensures that the images are invariant to variations of the brightness η and all pixel values lie between 0 and 1.

We fit a CNN by minimizing the empirical error under the 0–1 loss and the cross-entropy loss. As the most natural choice, empirical risk minimization based on the 0–1 loss has been extensively studied in classification theory within the standard statistical learning framework [4, 10, 66], but optimization with respect to the 0–1 loss is considered to be computationally intractable due to the loss function’s nonconvexity and discontinuity. The cross-entropy loss serves as a tractable surrogate for the 0–1 loss and is widely adopted in practice due to its favorable optimization properties [8, 72].

For the 0–1 loss, the empirical risk minimizer over the CNN class (29) is defined as

$$(32) \quad \hat{\mathbf{p}} = (\hat{p}_1, \hat{p}_2) \in \arg \min_{\mathbf{q}=(q_1, q_2) \in \mathcal{G}(\alpha, m)} \frac{1}{n} \sum_{i=1}^n \mathbb{1} \left(\mathbb{1} \left(q_2(\bar{\mathbf{X}}_i) > \frac{1}{2} \right) \neq k_i \right),$$

with $\mathbb{1}(\hat{p}_2(\bar{\mathbf{X}}_i) > 1/2)$ the predicted label of the i th sample based on the network $\hat{\mathbf{p}} = (\hat{p}_1, \hat{p}_2) \in \mathcal{G}(\alpha, m)$. The learned network $\hat{\mathbf{p}}$ outputs estimates for the two conditional class probabilities $p_1(\mathbf{x}) = \mathbf{P}(k = 0 | \mathbf{X} = \mathbf{x})$ and $p_2(\mathbf{x}) = \mathbf{P}(k = 1 | \mathbf{X} = \mathbf{x})$. These probabilities sum to one and optimizing over q_2 suffices.

For a new test image $\bar{\mathbf{X}}$ that has been normalized according to (31), the classifier $\hat{k}(\mathbf{X}) := \mathbb{1}(\hat{p}_2(\bar{\mathbf{X}}) > 1/2)$ assigns the label 1 if the estimated probability belonging to class 1 exceeds $1/2$ and assigns class label 0 otherwise.

For the theory, we consider the general deformation model (3) and impose the following assumption.

ASSUMPTION 3 (Covering of deformation class). For any $\alpha \in (0, 1]$, the deformation class $\mathcal{A}$ contains a finite subset $\mathcal{A}_{d_\alpha}$ such that, for any $A \in \mathcal{A}$, there exists an $A' \in \mathcal{A}_{d_\alpha}$ and indices $j, \ell \in \{1, \dots, d\}$ such that $A'(\cdot + j/d, \cdot + \ell/d)$ satisfies Assumption 2(i) and

$$\left\| A' \left(\cdot + \frac{j}{d}, \cdot + \frac{\ell}{d} \right) - A \right\|_\infty \leq d^{-\alpha}.$$

Similarly to the derivation of $\mathcal{A}_d^{-1}$ in Section 3, the subset $\mathcal{A}_{d_\alpha}$ can be obtained through suitable discretization of the deformation class $\mathcal{A}$. The cardinality of the discretized class $\mathcal{A}_{d_\alpha}$ is typically of the order $d^{\alpha r}$ with r the number of free parameters that are not related to the shifts. For instance, two out of the four parameters in the deformation model (2) control the shift such that $|\mathcal{A}_{d_\alpha}| \asymp d^{2\alpha}$. Adding one parameter controlling the rotation, as in (4) and (5), yields $|\mathcal{A}_{d_\alpha}| \asymp d^{3\alpha}$.

For the CNN-based method under the deformation model (3), we consider the separation quantity $D = D(f_0, f_1) \vee D(f_1, f_0)$ with

$$(33) \quad D(f, g) := \frac{\inf_{a, s, s' \in \mathbb{R}, A, A' \in \mathcal{A}} \|af \circ A(\cdot + s, \cdot + s') - g \circ A'\|_{L^2(\mathbb{R}^2)}}{\|g\|_{L^2(\mathbb{R}^2)}}.$$

It measures the normalized minimal L^2 -distance between any deformed versions of f and g (up to spatial shifts) under transformations from the class $\mathcal{A}$.

The next result states the misclassification bound for the CNN-based classifier with 0–1 loss.

THEOREM 4.1. *Consider the general deformation model (3) and suppose Assumptions 1, 2(i), and 3 hold. Let $\hat{\mathbf{p}}$ be the estimator in (32), based on the CNN class $\mathcal{G}(\alpha, |\mathcal{A}_{d_\alpha}|)$ defined in (29), with $|\mathcal{A}_{d_\alpha}| \geq 2$. Suppose a new data point $(\mathbf{X}, k)$ is independently drawn from the data distribution (30). If the separation quantity in (33) satisfies $D \geq \sqrt{\kappa/d^\alpha}$, where $\alpha \in (0, 1]$ and κ is a constant depending only on C_L and C_A , then for the classifier $\hat{k}(\mathbf{X}) = \mathbb{1}(\hat{p}_2(\bar{\mathbf{X}}) > 1/2)$, there exists a universal constant $C > 0$ such that*

$$(34) \quad \mathbf{P}(\hat{k}(\mathbf{X}) \neq k) \leq C \frac{|\mathcal{A}_{d_\alpha}|(d^{2\alpha} + |\mathcal{A}_{d_\alpha}|)}{n} \log^3(d|\mathcal{A}_{d_\alpha}|) \log n,$$

where $\mathbf{P}$ denotes the distribution over all randomness in the data and the new sample $\mathbf{X}$.

The proof is postponed to Section B of the Supplementary Material [14]. For fixed α , since the cardinality $|\mathcal{A}_{d_\alpha}|$ depends on d , the upper bound (34) grows in the image resolution d . At first glance, this might seem counterintuitive, as a high-image resolution typically provides more information about the object and should lead to improved misclassification bounds. However, as d increases, the template functions can become closer in their separation distance (33), making the two objects more similar and thus harder to separate.

The CNN employs $2|\mathcal{A}_{d_\alpha}|$ filters and contains $\lesssim (d^{2\alpha}|\mathcal{A}_{d_\alpha}| + |\mathcal{A}_{d_\alpha}|^2) \log(|\mathcal{A}_{d_\alpha}|)$ parameters (see Lemma B.11). Up to logarithmic factors, (34) means that we obtain a consistent classifier if the sample size is of larger order than the number of network parameters. More generally, Theorem 4.1 implies that, for sufficiently large sample sizes, the misclassification error of the proposed CNN-based classifier can become arbitrary small. Compared to the image classifiers discussed in Section 3, the CNN-based classifier only requires knowledge of the deformation class for the choice of the architecture. Additionally, for Theorem 4.1 and all subsequent results in this section, Assumption 2(i) on the deformation set $\mathcal{A}$ (necessary in Section 3) can be relaxed to Lipschitz continuity. This indicates that the CNN-based approach does not require invertible deformations. To guarantee that a sufficient amount of “information” is still preserved under the deformations, one instead needs to assume the existence of a universal constant $c_{\mathcal{A}} > 0$ such that, for any $A \in \mathcal{A}$, $\|f_i\|_1 \leq c_{\mathcal{A}}\|f_i \circ A\|_1$ with $i \in \{0, 1\}$.

That the CNN misclassification error can become arbitrarily small is in line with the nearly perfect classification results of deep learning for a number of image classification tasks in practice. Interestingly, most of the previous statistical analysis for neural networks considers settings with asymptotically nonvanishing prediction error. Those are statistical models where every new image contains randomness that is independent of the training data and can, therefore, not be predicted by the classifier. To illustrate this, consider the nonparametric regression model $Y_i = f(\mathbf{X}_i) + \sigma \varepsilon_i$, $i = 1, \dots, n$ with fixed noise variance σ^2 . The squared prediction error of the predictor $\hat{Y} = \hat{f}_n(\mathbf{X})$ for Y is $\mathbf{E}(\hat{Y} - Y)^2 = \sigma^2 + \mathbf{E}[(\hat{f}_n(\mathbf{X}) - f(\mathbf{X}))^2]$. This implies that, even if we can perfectly learn the function f from the data, the prediction error is still at least σ^2 . Taking a highly suboptimal but consistent estimator $\hat{f}_n$ for f yields a prediction error $\sigma^2 + o(1)$, thereby achieving the lower bound σ^2 up to a vanishing term. For instance, the one-nearest neighbor classifier does not employ any smoothing and results in suboptimal rates for conditional class probabilities but is optimal for the misclassification error up to a factor of 2, [17]. This shows that optimal estimation of f only affects the second-order term of the prediction error. As classification and regression are closely related, the same phenomenon also occurs in classification, whenever the conditional class probabilities lie strictly between 0 and 1. The only possibility to achieve small misclassification error requires that the conditional class probabilities are consistently close to either zero or one. This means that the covariates $\mathbf{X}$ contain (nearly) all information about the label k , [9]. The main source of randomness lies then in the sampling of the covariates $\mathbf{X}$. An example are the random deformations considered in this work that only affect the covariates but not the labels. This highlights the main difference from the existing standard classification settings.

Theorem 4.1 differs from the generalization bounds in [39] and related works, such as [7, 49], which focus on generalization performance of nonlearned functions (this means the functions are not allowed to depend on the data), a distinct perspective that can be traced back to [6]. For example, as can be seen in Theorem 2.1 of [39], their bound involves the term $\sum_{i=1}^n \ell(f(\mathbf{X}_i), k_i)$, with ℓ the loss function used. Since f does not depend on the data, this term cannot be further specified, and consequently, no statistical convergence rates for estimators can be deduced. In contrast, the derived misclassification error $\mathbf{P}(\hat{k}(\mathbf{X}) \neq k)$ accounts for all sources of randomness, including both the training data and the test sample. Furthermore, the derivation of $\hat{k}$ links the generalization analysis to approximation theory, enabling us to explicitly construct a suitable CNN model and thus avoid any implicit terms in the bound.

To prove Theorem 4.1, one can decompose the misclassification error into an approximation error and a stochastic error term (see Lemma B.10). The stochastic error can be bounded via statistical learning tools such as the Vapnik–Chervonenkis (VC) dimension of the CNN class $\mathcal{G}(\alpha, |\mathcal{A}_{d_\alpha}|)$; see Lemma B.11. The approximation error vanishes as CNNs from the class $\mathcal{G}(\alpha, |\mathcal{A}_{d_\alpha}|)$ with suitably chosen parameters can achieve perfect classification. This

result is stated in the following theorem and is proved in Section B of the Supplementary Material [14].

THEOREM 4.2. *If Assumptions 1, 2(i), and 3 hold and the separation quantity D in (33) satisfies $D \geq \sqrt{\kappa/d^\alpha}$, with $\alpha \in (0, 1]$ and κ a constant depending only on the constants C_L and C_A , then for any $(\mathbf{X}, k)$ generated from the same distribution as the data in (30), there exists a network $\mathbf{p} = (p_1, p_2) \in \mathcal{G}(\alpha, |\mathcal{A}_{d_\alpha}|)$ such that the corresponding classifier $\tilde{k}(\mathbf{X}) = \mathbb{1}(p_2(\tilde{\mathbf{X}}) > 1/2)$ satisfies*

$$\tilde{k}(\mathbf{X}) = k \quad \text{almost surely.}$$

To ensure the existence of the interpolating classifier in the previous result, the best achievable order of the separation quantity D with respect to the resolution level d is

$$D \gtrsim \frac{1}{\sqrt{d}},$$

which is more restrictive if compared to the lower bound $D \gtrsim 1/d$ imposed on the image classification methods discussed in Section 3. This discrepancy arises from the construction of the proposed CNN architecture. To handle different deformations, the CNN construction uses $2|\mathcal{A}_{d_\alpha}|$ separate filters to test whether the image was approximately generated by applying one of the deformations in the discretized set $\mathcal{A}_{d_\alpha}$ to either of the two possible template functions. More specifically, consider a given input image that has been generated by deforming the template function of class $k \in \{0, 1\}$ by A . In Proposition B.5, we show that the convolutional filter corresponding to the correct class k and the deformation in $\mathcal{A}_{d_\alpha}$ that is closest to the true deformation A produces the highest activation, that is, the highest output value after convolution and global max-pooling; see Figure 6 for an illustration. This requires $D \gtrsim 1/\sqrt{d}$ when $\alpha = 1$.

The $1/d$ separation rate can be obtained, under additional conditions, if we instead take $2|\mathcal{A}_{d_1}|^2$ many convolutional filters, resulting in a CNN architecture with $\lesssim (|\mathcal{A}_{d_1}|^2 d^2 + |\mathcal{A}_{d_1}|^4) \log(|\mathcal{A}_{d_1}|)$ network parameters. The argument is outlined in the Supplementary Material [14] after the proof of Theorem 4.2. We want to emphasize that the rate $1/\sqrt{d}$ already covers most practical application scenarios, and the goal of this work is to analyze the performance of commonly used CNN architectures with as few filters as possible.

A consequence of the approximation result is that, under the conditions of Theorem 4.2, the label can be retrieved from the image.

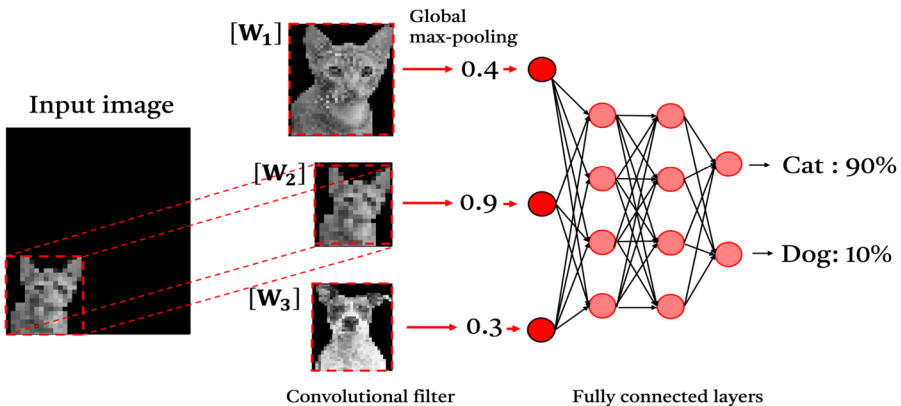


FIG. 6. Effect of different filters on the same image. The global max-pooling layer will generate the largest values for filters that are most similar to the object.

LEMMA 4.3. *If Assumptions 1, 2(i), and 3 hold and the separation quantity D in (33) satisfies $D \geq \sqrt{\kappa/d^\alpha}$, with $\alpha \in (0, 1]$ and κ a constant depending only on the constants C_L and C_A , then for any $(\mathbf{X}, k)$ generated from the same distribution as the data in (30), its label k can be written as a deterministic function $k(\cdot)$ evaluated at $\mathbf{X}$, and we have*

$$p(\mathbf{X}) = k(\mathbf{X}),$$

where $p(\mathbf{x}) = \mathbf{P}(k = 1 | \mathbf{X} = \mathbf{x})$.

A consequence is that under the imposed conditions

$$\min_{q: [0, 1]^2 \rightarrow \{0, 1\}} \mathbf{P}(q(\mathbf{X}) \neq k(\mathbf{X})) = 0.$$

Next, we consider CNN-based classifiers trained using the commonly employed cross-entropy loss. For a pair $(\mathbf{X}, k)$ with image $\mathbf{X}$ and its label $k \in \{0, 1\}$, drawn from the data distribution, and $\bar{\mathbf{X}}$ the preprocessed image, the objective is to minimize the population risk

$$-\mathbf{E}[k \log(f(\bar{\mathbf{X}})) + (1 - k) \log(1 - f(\bar{\mathbf{X}}))]$$

over a suitable class of CNNs. Since the true distribution of $(\mathbf{X}, k)$ is unknown, one instead minimizes the empirical version of this risk.

To avoid degeneracy in the cross-entropy loss, we consider the CNN class $\mathcal{G}(\alpha, m)$ with a slight modification that prevents the outputs to be too close to the boundaries 0 and 1. Specifically, we define

$$(35) \quad \tilde{\mathcal{G}}(\alpha, m) := \{(1 - \tilde{g}, \tilde{g}) : \tilde{g} = (g \vee \rho(1)) \wedge \rho(-1), (1 - g, g) \in \mathcal{G}(\alpha, m)\},$$

where $\rho(z) := 1/(1 + e^z)$. The specific values $\rho(1)$, $\rho(-1)$ are convenient but not essential. The empirical risk minimizer over all CNNs of the form (35) is

$$(36) \quad \begin{aligned} \hat{\mathbf{p}}^{\text{CE}} &= (\hat{p}_1^{\text{CE}}, \hat{p}_2^{\text{CE}}) \\ &\in \arg \min_{\mathbf{q}=(q_1, q_2) \in \tilde{\mathcal{G}}(\alpha, m)} -\frac{1}{n} \sum_{i=1}^n [k_i \log(q_2(\bar{\mathbf{X}}_i)) + (1 - k_i) \log(1 - q_2(\bar{\mathbf{X}}_i))], \end{aligned}$$

where $\bar{\mathbf{X}}_i$ denotes the normalized image as defined in (31). The misclassification bound for this CNN based classifier

$$\hat{k}^{\text{CE}}(\mathbf{X}) = \mathbb{1}(\hat{p}_2^{\text{CE}}(\bar{\mathbf{X}}) > 1/2)$$

is given below.

THEOREM 4.4. *Consider the general deformation model (3), and suppose Assumptions 1, 2(i), and 3 hold. Let $0 < \alpha \leq 1$ and $\hat{\mathbf{p}}^{\text{CE}}$ be the estimator in (36), based on the CNN class $\tilde{\mathcal{G}}(\alpha, |A_{d_\alpha}|)$ defined in (35). Suppose a new data point $(\mathbf{X}, k)$ is independently drawn from the same distribution as the data in (30). If $|A_{d_\alpha}| \geq (d^\alpha \vee 2)$ and the separation quantity in (33) satisfies $D \geq \sqrt{\kappa/d^\alpha}$ for a constant κ only depending on (C_L, C_A) , then there exists a universal constant $C > 0$ such that, for any $\gamma > 0$ and any sufficiently large n ,*

$$(37) \quad \mathbf{P}(\hat{k}^{\text{CE}}(\mathbf{X}) \neq k) \leq C \frac{|A_{d_\alpha}|(d^{2\alpha} + |A_{d_\alpha}|)}{n} \log(|A_{d_\alpha}|) \log^{1+\gamma} n,$$

where $\mathbf{P}$ denotes the distribution over all randomness in the data and the new sample $\mathbf{X}$.

The proof of Theorem 4.4 is postponed to Section B of the Supplementary Material [14]. Up to logarithmic terms, CNN-based classifiers trained with the cross-entropy loss exhibit similar generalization performance to those trained with the 0–1 loss, and the misclassification error of both classifiers converges to zero at the rate V/n , where V denotes the VC dimension of the underlying CNN architecture. The rate V/n is minimax optimal; see Theorem 4 in [44].

The CNN-based approach discussed in this section can be readily extended to the multi-class classification setting. Generalizing from 2 to $K > 2$ classes, the number of filters and the width of all layers in the fully-connected network have to be multiplied by $K/2$. The softmax function is then given by $\Phi(x_1, \dots, x_K) = (e^{x_1} / \sum_{i=1}^K e^{x_i}, \dots, e^{x_K} / \sum_{i=1}^K e^{x_i})$. Regarding the approximation theory, if Assumptions 1, 2(i), and 3 hold and the separation quantity satisfies the pairwise lower bound $D_{i,j} \gtrsim d^{-\alpha/2}$, for all $i, j = 1, \dots, K$, with $i \neq j$, then there exists a network $\mathbf{p} = (p_1, \dots, p_K)$ with the architecture described above such that the corresponding classifier $\hat{k}(\mathbf{X}) = \arg \max_{\ell \in \{1, \dots, K\}} p_\ell(\bar{\mathbf{X}})$ satisfies $\hat{k}(\mathbf{X}) = k$, almost surely. Following the proof of Lemma B.14, the covering number of the model class can be shown to be K times that of the binary case (up to a logarithmic factor). With both the approximation and covering number results, one can apply existing error decomposition results, such as Theorem 3.5 (for the cross-entropy loss) in [9] or Lemma A.9 (for the hinge loss) in [30], to bound the final misclassification error in the multiclass setting. Compared to the binary case, the resulting misclassification error bound scales with the number of classes K .

On the optimization side, current deep learning theory still cannot incorporate the algorithm (with the common initialization schemes and without excessive number of restarting) into the learning process for general nonparametric estimation problems. The NTK regime allows to study gradient descent, but in a wide limit where the method converges to kernel regression and thus the feature learning is lost. Therefore, to derive insights on how deep networks can learn functions, the statistical theory literature focuses on the empirical risk minimizer (ERM). The discrepancy might be less severe given the claims in the literature that gradient based methods find local minima with training loss close to the global minimum. For large networks it can be rigorously shown that gradient descent can reach zero training loss, and thus gradient descent eventually converges to an ERM [1, 2, 20, 21, 37, 73].

5. Numerical results. We empirically compare the three proposed methods: the inverse mapping classifier (10), the image alignment classifier (23), and CNN-based classifiers with three different architectures.

5.1. Inverse mapping and image alignment. Theorems 3.1 and 3.7 show that both the inverse mapping classifier and the image alignment classifier can recover the correct label, provided each class appears at least once in the training data. We computed the empirical performance for both methods using a balanced design with $n = \{2, 4, 8, 16, 32\}$ labeled samples, meaning that each of the two classes contributes $n/2$ images.

We consider the deformation model (2) and generate the template functions from the FashionMNIST and CIFAR-100 datasets. FashionMNIST contains 28×28 grayscale images from different fashion categories, while CIFAR-100 includes 32×32 color images across 100 classes. In our setup this corresponds to $d = 28$ and $d = 32$, respectively. In the experiments using FashionMNIST, we select as template functions one image with label “pullover” and one image with label “coat.” For CIFAR-100, the template functions are generated from the classes “fox” and “flatfish” and preprocessed to grayscale images with black backgrounds. Training samples are then generated by applying random shifts (τ, τ') , scalings (ξ, ξ') , and brightness changes (η) using the Keras ImageDataGenerator. For FashionMNIST, examples of generated deformed images are displayed in Figure 7.

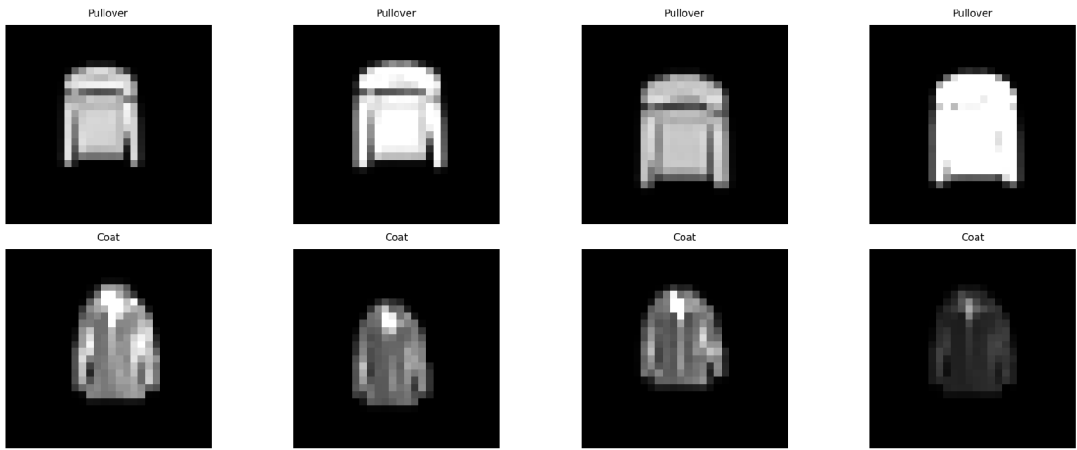


FIG. 7. Augmented images with label “pullover” (top row) and “coat” (bottom row) from the FashionMNIST dataset.

As explored in Section 3.1, the inverse mapping classifier (10) is based on a discretization of the deformation class. For correct classification, the grid size of this discretization should be $\leq$ “small constant” $\times$ “separation distance between the template functions.” The theory focuses on bounds that also apply to the worst-case separation distance $D \asymp 1/d$. Thus, the grid size becomes of the order $1/d$. In practice, the separation distance could be much larger, which means that a less fine discretization of the deformation class is sufficient. We empirically compare the image alignment classifier with the inverse mapping classifier based on different grid sizes. The performance of each classifier $\hat{k}$ is evaluated by the empirical misclassification risk (test error)

(38)
$$R_N = \frac{1}{N} \sum_{i=1}^N \mathbb{1}(\hat{k}(\mathbf{X}_{n+i}) \neq k_{n+i}),$$

based on test data $(\mathbf{X}_{n+1}, k_{n+1}), \dots, (\mathbf{X}_N, k_N)$, that are independently generated from the same distribution as the training data. For $N = 50$, Figure 8 reports the median of the test error based on 10 repetitions in each setting.

The results shown in Figure 8 align with the theory presented in Section 3. Specifically, for the image alignment method, the misclassification error is zero as long as each class appears

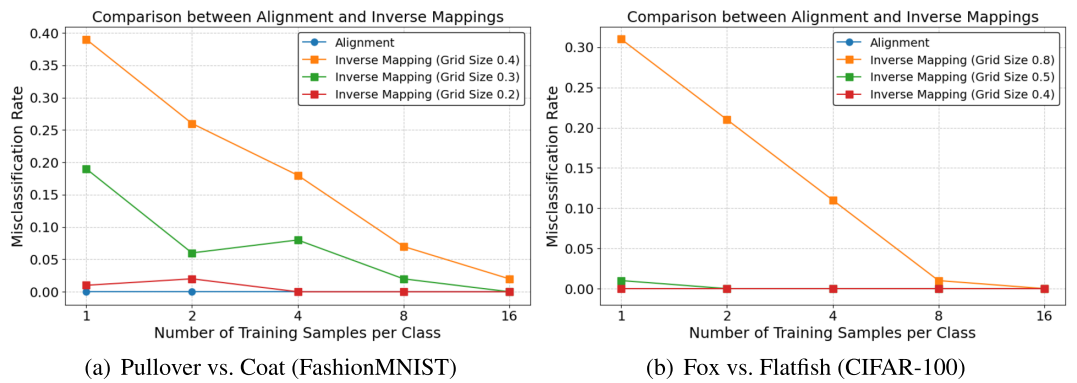


FIG. 8. Comparison of the image alignment classifier (blue) and inverse mapping classifiers with different discretization grid sizes (orange, green, red). The plots show median test errors over 10 repetitions for varying training sample sizes. In (b), both the alignment classifier and the inverse mapping classifier with grid size 0.4 achieve zero misclassification, resulting in the blue and red lines overlapping.

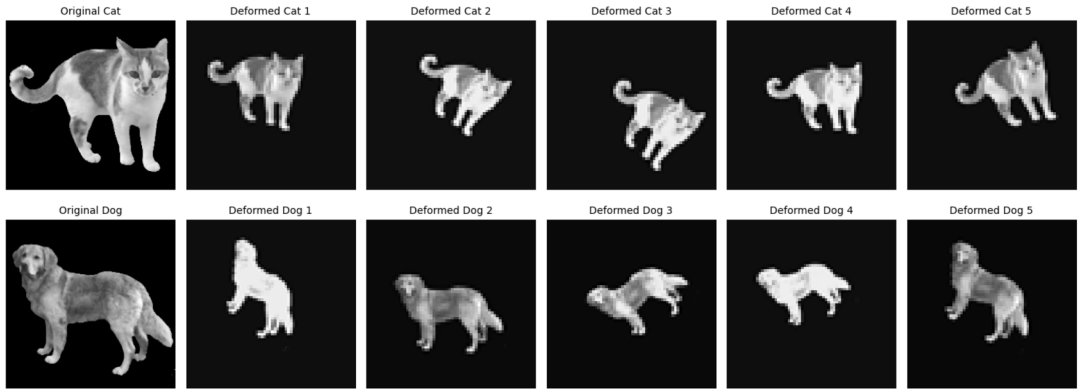


FIG. 9. The template cat and dog images (224×224) and their corresponding deformed samples (64×64).

at least once in the training data. For the inverse mapping classifier, the misclassification error decreases as the discretization becomes finer, eventually also reaching zero misclassification error. Moreover, the misclassification error of all inverse mapping classifiers decreases as the training sample size increases. The reason is that more samples yield a denser set of discretized inverse mappings to compare to, thereby improving prediction accuracy.

5.2. Classifiers via CNN-based approach. This section investigates the learning performance of three different CNN architectures with template functions generated from MNIST, FashionMNIST, CIFAR-100, and ImageNet. The MNIST dataset consists of 28×28 pixel grayscale images of handwritten digits (0 – 9). ImageNet consists of labeled high-resolution (typically 224×224) color images. We select the template functions by drawing images with label “0” vs. “4” (MNIST), “T-shirt” vs. “dress” (FashionMNIST), “fox” vs. “flatfish” (CIFAR-100), and “cat” vs. “dog” (ImageNet).

The training and test samples are generated using the same procedure as in Section 5.1, but with a more complex deformation model described by (4) and (5), which also incorporates random rotations. To accelerate training, we rescale the ImageNet images from 224×224 to 64×64 . Examples of template and deformed images from ImageNet are shown in Figure 9, and examples from the other datasets are provided in Section C of the Supplementary Material [14].

We consider three different CNN architectures, each with one convolutional layer as in Section 4, sharing the same filter size but differing in the number of filters and the width of the feed-forward layers, as summarized in Table 1. All CNNs are trained using cross-entropy loss and the Adam optimizer in Keras (with the TensorFlow backend), employing the default learning rate of 0.001. For each task, training is conducted on balanced samples with $n/2 = \{2^6, 2^7, 2^8, 2^9, 2^{10}, 2^{11}\}$ samples per class. The performance of each classifier $\hat{k}$ is evaluated using the misclassification error in (38), computed on i.i.d. test samples. For $N = 200$, Figure 10 reports the median test errors over 10 repetitions for each CNN classifier across various classification tasks.

TABLE 1
Details of three CNN architectures

	Filter size	Number of filters	Width of feed-forward layers
CNN1	10×10	32	128
CNN2	10×10	64	256
CNN3	10×10	128	512

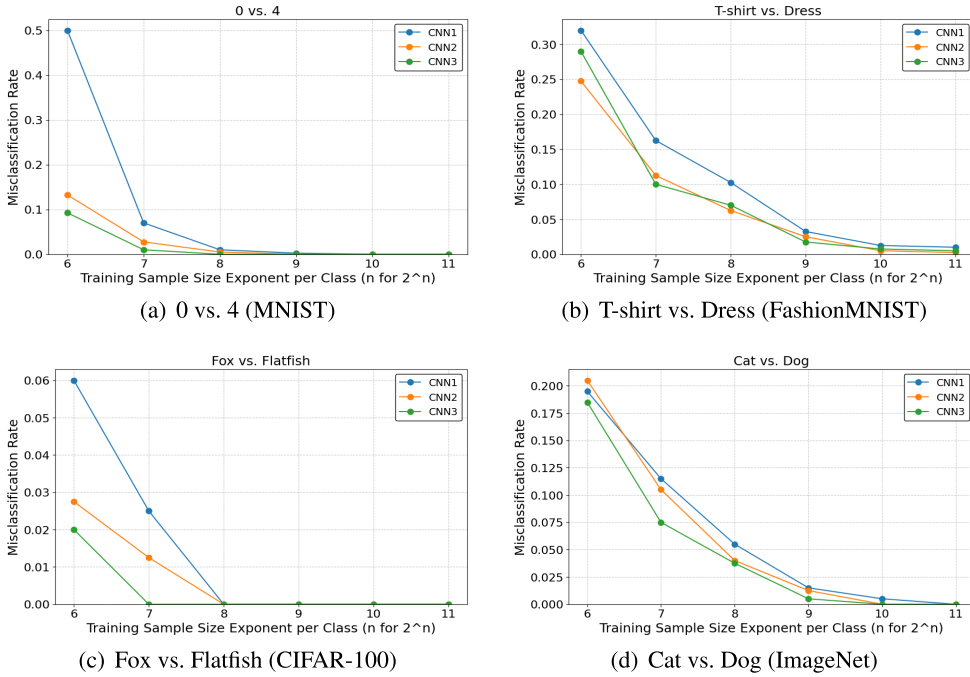


FIG. 10. Comparison of three trained CNNs. Reported is the median of the test error with $N = 200$ over 10 repetitions.

For all four classification tasks and all three CNN-based classifiers, the misclassification error decreases and eventually converges to zero as the training sample size increases. The rate of this decay varies across tasks. Among the four tasks, classifying the labels “0” vs. “4” and “fox” vs. “flatfish” is relatively easy, achieving near-zero error with fewer samples. In contrast, classifying “T-shirt” vs. “dress” and “cat” vs. “dog” is more challenging, requiring larger sample sizes to reach similarly low error rates. In the comparison among CNN1, CNN2, and CNN3, increasing the number of filters and the width of the fully connected layers enhances performance by enabling a finer approximation of the deformation set $\mathcal{A}$ (Assumption 3). These numerical results are consistent with the theoretical findings presented in Section 4.

6. Overview of deformation-based analysis. We already discussed the similarity with the optimization problem (15) in image registration.

In functional data analysis (FDA), curves typically exhibit amplitude (vertical) and phase (horizontal) variation. Let $y_1, \dots, y_n : \mathbb{R}_+ \rightarrow \mathbb{R}$ be observed functions with both types, and $x_1, \dots, x_n$ the amplitude-only functions. They can be related by unknown time-warping functions $h_i \in \mathcal{H} : \mathbb{R}_+ \rightarrow \mathbb{R}_+$ such that $y_i(t) = x_i(h_i(t))$ [31, 38]. Depending on the application, h_i may take various forms, such as uniform scaling, shifts, or more generally affine transformations. More involved statistical analysis often requires conditions such as smoothness and orientation preservation (i.e., $h_i(0) = 0$, $h_i(1) = 1$, and strict monotonicity); for example, [60] considers orientation-preserving diffeomorphisms. Additionally, some works impose the unbiasedness condition $\mathbf{E}[h_i(t)] = t$ [50, 62].

When phase variation is treated as a nuisance [43, 60], the goal is to remove it to improve feature estimation, such as the mean curve. This is achieved through curve registration (also referred to as curve alignment in biology), encompassing a wide range of techniques, including earlier heuristics like dynamic time warping (DTW) and landmark registration. These

alignment techniques use a template function x_0 as the target, seeking a time-warping function g_i such that $y_i \circ g_i \approx x_0$, where closeness is measured by a chosen metric. For simplicity of illustration, let us assume that $\mathcal{H}$ admits a group structure with identity, inversion, and closure under composition. Most methods optimize objective functions of the form

$$(39) \quad \mathcal{L}_{\lambda,i}[\mu] = \inf_{h_i \in \mathcal{H}} (\|y_i \circ h_i - \mu\|^2 + \lambda \mathcal{R}(h_i))$$

with $\mathcal{R}$ a regularizer. In [62] the authors take $\mu = y_j$, compute pairwise warping functions h_{ij} for each j via (39), and average them to obtain g_i . They establish an upper bound on the sup-norm distance between estimated and true warping functions. In contrast, [31] adopts a template-based method, assigning each i a distinct $\mu_i = \sum_{j=1}^p \beta_{ij} \xi_j$, where the ξ_j are data-driven basis elements closely related to the principal components. Besides implementing optimization as in (39), another approach models amplitude and phase variations through equivalence classes, drawing inspiration from shape analysis [19] and pattern theory [25]. This absorbs nonshape attributes like translation, rotation, and scaling into equivalence classes [59, 71], motivating the use of a metric d that satisfies the invariance for all $h \in \mathcal{H}$,

$$(40) \quad d(x_1, x_2) = d(x_1 \circ h, x_2 \circ h).$$

In [40, 60, 64], the authors use the Fisher–Rao Riemannian metric and represent each function $x(t)$ by its square-root velocity function (SRVF) $q(t) := x'(t)/\sqrt{|x'(t)|}$. This links the Fisher–Rao metric between functions x_1, x_2 to the Euclidean distance between their SRVFs q_1, q_2 , making distance computation feasible. The strict monotonicity of h is essential for the derivation. Under certain conditions, [60] prove that their alignment algorithm, based on computing the Fréchet mean, yields a consistent estimator of the original warping-free function. From the same perspective, [40] developed a Bayesian model for estimating warping functions, supported by numerical experiments. Compared with the above literature, our work may be regarded as an adaptation of the concept of phase variation to the context of 2D image classification, where such variability is modeled as geometric deformations. This adaptation is nontrivial, as conditions commonly required in FDA, such as orientation preservation, are no longer appropriate in the context of 2D image classification. Additionally, while the concept of equivalence classes is intuitive for classification tasks, pursuing strict invariance often goes beyond what is necessary or beneficial in this setting. Indeed, if each class with deformations is regarded as an orbit, the goal is not to have two parallel orbits, as in (40), but rather to ensure that the distance between orbits is sufficiently large to offset errors introduced by pixel-based image representation.

Another related line of research is Mallat’s series of works on the scattering transform [12, 13, 41] in pattern recognition, which offers an alternative representation approach to achieving invariance. Specifically, given a signal f on $\mathbb{R}^d$, they focus on deformations modeled as small diffeomorphisms close to translations, expressed by $L_\tau f(x) = f(x - \tau(x))$, where $\tau(x) \in \mathbb{R}^d$ is a displacement field. The scattering representation Φ is constructed using a wavelet scattering network, which cascades wavelet transforms with nonlinear modulus and averaging operators [11]. As demonstrated in [41], this representation is translation-invariant (Theorem 2.10) and Lipschitz continuous with respect to the action of diffeomorphisms on compactly supported functions (Corollary 2.15), satisfying

$$\|\Phi(L_\tau f) - \Phi(f)\| \lesssim \|f\|_2 \left(\sup_x |\nabla \tau(x)| + \sup_x |H\tau(x)| \right),$$

where $\nabla \tau$ denotes the deformation gradient tensor and $H\tau$ the Hessian tensor. With additional constructions, scattering networks can also be made rotation-invariant (Section 5 of [41]). A key difference from CNNs, which we study in this paper, is that scattering networks

are not learned from data but explicitly designed based on prior requirements to ensure invariance or stability to specific signal deformations. This inherently limits their flexibility to predesigned invariances. These works contribute to the broader field of invariant representation learning. Related research includes [3, 57], where [3] considers deformations forming a compact group and shows that representations defined by nonlinear group averages are invariant under the group actions.

7. Conclusion and extensions. This paper introduces a novel statistical framework for image classification. Instead of treating each pixel as a variable and analyzing a nonparametric denoising problem where randomness occurs as additive noise, the proposed deformation framework models the variability of objects within one class as geometric deformations of template functions. The abstract framework encompasses a wide range of linear and nonlinear deformations, including commonly considered transformations such as rotations, shifts, and rescaling.

In real world images, deformations can be highly complex, but the analytical approach discussed here may still offer insights. Extensions to background noise and multiple objects have been briefly discussed in Section 3.2. The CNN approach in Section 4 seems quite robust to independent background noise in the image as it relies on inner products, where, by the CLT, the noise should be rather small relative to the object signal. However, analyzing complex, structured backgrounds is more challenging and requires proper statistical modeling. Modifying the approach in Sections 3 and 4, misclassification error bounds can be derived for the case where fixed images from two different classes are randomly occluded. Closely related are extensions to partially visible objects. In this scenario relying on global characteristics, such as the full support of the object, seems unreasonable. However, it might still be possible to construct classifiers, that provide similar theoretical guarantees by focusing on local properties instead. The presence of sharp edges corresponds to a locally large (or even infinite) Lipschitz constant C_L in the template functions and requires a more refined analysis.

A further potential extension is to incorporate perspective transformations from the computer vision literature [29, 69]. The underlying idea is that images captured from different perspectives can be modeled as

$$\begin{pmatrix} \tilde{a}_1(u, v) \\ \tilde{a}_2(u, v) \\ w(u, v) \end{pmatrix} = \begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{pmatrix} \begin{pmatrix} u \\ v \\ 1 \end{pmatrix},$$

where h_{ij} are the parameters of the nonsingular homography matrix and w is the so-called scaling factor. In this framework, $a_1(u, v)$ and $a_2(u, v)$ are obtained by normalizing the output by w , namely, $a_1(u, v) = \tilde{a}_1(u, v)/w(u, v)$ and $a_2(u, v) = \tilde{a}_2(u, v)/w(u, v)$. Affine transformations can be recovered as a special case by choosing $h_{31} = h_{32} = 0$ and $h_{33} = 1$. To include perspective transformations, one needs to relax the partial differentiability imposed in Assumption 2.

The work in [47] and Section 5 of [48] discusses further deformation classes, including those that account for noise and blur, multiscale superposition, domain warping, and interruptions.

Additionally, various sophisticated image deformation models have been proposed for medical image registration. As mentioned in (15), image registration seeks an optimal transformation mapping the target image to the source image. To study and compare image registration methods, it is essential to construct realistic image deformation models describing the generation of the deformed image from the template. The survey article [58] classifies these deformation models into several categories, such as ODE/PDE based models, interpolation-based models, and knowledge-based models. For instance, a simple ODE based random image deformation model takes $\mathbf{X}$ as template/source image and generates a random vector field

u . This can be achieved by selecting a basis and generating independent random coefficients according to a fixed distribution. Given the vector field u , a continuous image deformation $\mathbf{X}(t)$ is generated by solving the differential equation $\partial_t \mathbf{X}(t) = u(\mathbf{X}(t))$ with $\mathbf{X}(0) = \mathbf{X}$. The randomly deformed image is then $\mathbf{X}(1)$. The DARTEL algorithm [5] is a widely recognized approach for image registration within this deformation model. A statistical analysis of these methods is still lacking.

Acknowledgments. The authors are grateful to the Editor, the Associate Editor, and the referees for helpful remarks and suggestions. We thank Yanis Bosch and Mark Podolskij for pointing out a mistake in a previous version of the manuscript. Most of J.C.’s work on this project was done while she was a postdoctoral researcher at the University of Twente.

Funding. The research of J. S.-H. and J.C. was supported by the NWO Vidi grant VI.Vidi.192.021.

S.L. was supported in part by the NWO Veni grant VI.Veni.232.033.

SUPPLEMENTARY MATERIAL

Supplement to “A novel statistical approach to analyze image classification” (DOI: [10.1214/26-AOS2618SUPP](https://doi.org/10.1214/26-AOS2618SUPP); .pdf). The supplement contains all the proofs of the theorems and lemmas stated in the paper.

REFERENCES

- [1] ALLEN-ZHU, Z., LI, Y. and LIANG, Y. (2019). Learning and generalization in overparameterized neural networks, going beyond two layers. In *Neural Information Processing Systems*.
- [2] ALLEN-ZHU, Z., LI, Y. and ZHAO, S. (2019). A convergence theory for deep learning via over-parameterization. In *International Conference on Machine Learning* 242–252.
- [3] ANSELM, F., ROSASCO, L. and POGGIO, T. (2016). On invariance and selectivity in representation learning. *Inf. Inference* **5** 134–158. [MR3516856 https://doi.org/10.1093/imaiai/iaw009](https://doi.org/10.1093/imaiai/iaw009)
- [4] ARLOT, S. and BARTLETT, P. L. (2011). Margin-adaptive model selection in statistical learning. *Bernoulli* **17** 687–713. [MR2787611 https://doi.org/10.3150/10-BEJ288](https://doi.org/10.3150/10-BEJ288)
- [5] ASHBURNER, J. (2007). A fast diffeomorphic image registration algorithm. *NeuroImage* **38** 95–113.
- [6] BARTLETT, P. L. (1998). The sample complexity of pattern classification with neural networks: The size of the weights is more important than the size of the network. *IEEE Trans. Inf. Theory* **44** 525–536. [MR1607706 https://doi.org/10.1109/18.661502](https://doi.org/10.1109/18.661502)
- [7] BARTLETT, P. L., FOSTER, D. J. and TELGARSKY, M. J. (2017). Spectrally-normalized margin bounds for neural networks. In *Neural Information Processing Systems*.
- [8] BARTLETT, P. L., JORDAN, M. I. and MCAULIFFE, J. D. (2006). Convexity, classification, and risk bounds. *J. Amer. Statist. Assoc.* **101** 138–156. [MR2268032 https://doi.org/10.1198/016214505000000907](https://doi.org/10.1198/016214505000000907)
- [9] BOS, T. and SCHMIDT-HIEBER, J. (2022). Convergence rates of deep ReLU networks for multiclass classification. *Electron. J. Stat.* **16** 2724–2773. [MR4406243 https://doi.org/10.1214/22-ejs2011](https://doi.org/10.1214/22-ejs2011)
- [10] BOUCHERON, S., BOUSQUET, O. and LUGOSI, G. (2005). Theory of classification: A survey of some recent advances. *ESAIM Probab. Stat.* **9** 323–375. [MR2182250 https://doi.org/10.1051/ps:2005018](https://doi.org/10.1051/ps:2005018)
- [11] BRUNA, J. (2013). Scattering representations for recognition. PhD thesis, Ecole Polytechnique.
- [12] BRUNA, J. and MALLAT, S. (2011). Classification with scattering operators. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition* 1561–1566.
- [13] BRUNA, J. and MALLAT, S. (2013). Invariant scattering convolution networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **35** 1872–1886.
- [14] CHEN, J., LANGER, S. and SCHMIDT-HIEBER, J. (2026). Supplement to “A novel statistical approach to analyze image classification.” <https://doi.org/10.1214/26-AOS2618SUPP>
- [15] CHEN, J., LIU, Y., WEI, S., BIAN, Z., SUBRAMANIAN, S., CARASS, A., PRINCE, J. L. and DU, Y. (2025). A survey on deep learning in medical image registration: New technologies, uncertainty, evaluation metrics, and beyond. *Med. Image Anal.* **100** 103385.
- [16] COHEN, T. and WELLING, M. (2016). Group equivariant convolutional networks. In *International Conference on Machine Learning* 2990–2999.

- [17] COVER, T. and HART, P. (1967). Nearest neighbor pattern classification. *IEEE Trans. Inf. Theory* **13** 21–27.
- [18] DELAIGLE, A. and HALL, P. (2012). Achieving near perfect classification for functional data. *J. R. Stat. Soc. Ser. B. Stat. Methodol.* **74** 267–286. [MR2899863](#) <https://doi.org/10.1111/j.1467-9868.2011.01003.x>
- [19] DRYDEN, I. L. and MARDIA, K. V. (2016). *Statistical Shape Analysis with Applications in R*, 2nd ed. *Wiley Series in Probability and Statistics*. Wiley, Chichester. [MR3559734](#) <https://doi.org/10.1002/9781119072492>
- [20] DU, S. S., LEE, J. D., LI, H., WANG, L. and ZHAI, X. (2019). Gradient descent finds global minima of deep neural networks. In *International Conference on Machine Learning* 1675–1685.
- [21] DU, S. S., ZHAI, X., PÓCZOS, B. and SINGH, A. (2019). Gradient descent provably optimizes over-parameterized neural networks. In *International Conference on Learning Representations*.
- [22] GLUCKMAN, J. (2006). Scale variant image pyramids. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition* 1069–1075.
- [23] GONZALEZ, R. C. and WOODS, R. E. (2008). *Digital Image Processing*. Pearson, New York.
- [24] GRENDER, U. (1976–1981). *Lectures in Pattern Theory I, II and III: Pattern Analysis, Pattern Synthesis and Regular Structures*. Springer, New York.
- [25] GRENDER, U. (1993). *General Pattern Theory: A Mathematical Study of Regular Structures*. *Oxford Mathematical Monographs*. Clarendon, Oxford. [MR1270904](#)
- [26] HARALICK, R. M. and SHAPIRO, L. G. (1985). Image segmentation techniques. *Comput. Graph. Image Process.* **29** 100–132.
- [27] HASHEMI, M. (2019). Enlarging smaller images before inputting into convolutional neural network: Zero-padding vs. interpolation. *J. Big Data* **6** 98.
- [28] JACQUES, J. and PREDA, C. (2014). Functional data clustering: A survey. *Adv. Data Anal. Classif.* **8** 231–255. [MR3253859](#) <https://doi.org/10.1007/s11634-013-0158-y>
- [29] KHATRI, N., DASGUPTA, A., SHEN, Y., ZHONG, X. and SHIH, F. Y. (2022). Perspective transformation layer. In *International Conference on Computational Science and Computational Intelligence* 1395–1401.
- [30] KIM, Y., OHN, I. and KIM, D. (2021). Fast convergence rates of deep neural networks for classification. *Neural Netw.* **138** 179–197.
- [31] KNEIP, A. and RAMSAY, J. O. (2008). Combining registration and fitting for functional models. *J. Amer. Statist. Assoc.* **103** 1155–1165. [MR2528838](#) <https://doi.org/10.1198/016214508000000517>
- [32] KOHLER, M., KRZYŻAK, A. and WALTER, B. (2022). On the rate of convergence of image classifiers based on convolutional neural networks. *Ann. Inst. Statist. Math.* **74** 1085–1108. [MR4498394](#) <https://doi.org/10.1007/s10463-022-00828-4>
- [33] KOHLER, M. and LANGER, S. (2025). Statistical theory for image classification using deep convolutional neural network with cross-entropy loss under the hierarchical max-pooling model. *J. Statist. Plann. Inference* **234** Paper No. 106188, 9. [MR4760228](#) <https://doi.org/10.1016/j.jspi.2024.106188>
- [34] KORMAN, S., REICHMAN, D., TSUR, G. and AVIDAN, S. (2013). FasT-match: Fast affine template matching. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition* 2331–2338.
- [35] KRIZHEVSKY, A., SUTSKEVER, I. and HINTON, G. E. (2017). ImageNet classification with deep convolutional neural networks. *Commun. ACM* **60** 84–90.
- [36] LECUN, Y., BENGIO, Y. and HINTON, G. E. (2015). Deep learning. *Nature* **521** 436–444.
- [37] LI, Y. and LIANG, Y. (2018). Learning overparameterized neural networks via stochastic gradient descent on structured data. In *Neural Information Processing Systems*.
- [38] LIU, X. and MÜLLER, H.-G. (2004). Functional convex averaging and synchronization for time-warped random curves. *J. Amer. Statist. Assoc.* **99** 687–699. [MR2090903](#) <https://doi.org/10.1198/016214504000000999>
- [39] LONG, P. M. and SEDGHI, H. (2020). Generalization bounds for deep convolutional neural networks. In *International Conference on Learning Representations*.
- [40] LU, Y., HERBEL, R. and KURTEK, S. (2017). Bayesian registration of functions with a Gaussian process prior. *J. Comput. Graph. Statist.* **26** 894–904. [MR3765353](#) <https://doi.org/10.1080/10618600.2017.1336444>
- [41] MALLAT, S. (2012). Group invariant scattering. *Comm. Pure Appl. Math.* **65** 1331–1398. [MR2957703](#) <https://doi.org/10.1002/cpa.21413>
- [42] MARCOS, D., VOLPI, M. and TUIA, D. (2016). Learning rotation invariant convolutional filters for texture classification. In *International Conference on Pattern Recognition* 2012–2017.
- [43] MARRON, J. S., RAMSAY, J. O., SANGALLI, L. M. and SRIVASTAVA, A. (2015). Functional data analysis of amplitude and phase variation. *Statist. Sci.* **30** 468–484. [MR3432837](#) <https://doi.org/10.1214/15-STS524>
- [44] MASSART, P. and NÉDÉLEC, É. (2006). Risk bounds for statistical learning. *Ann. Statist.* **34** 2326–2366. [MR2291502](#) <https://doi.org/10.1214/009053606000000786>

- [45] MINAEE, S., BOYKOV, Y. Y., PORIKLI, F., PLAZA, A. J., KEHTARNAVAZ, N. and TERZOPOULOS, D. (2022). Image segmentation using deep learning: A survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **44** 3523–3542.
- [46] MUMFORD, D. (1996). Pattern theory: A unifying perspective. In *Perception as Bayesian Inference* 25–62. Cambridge Univ. Press, Cambridge.
- [47] MUMFORD, D. (2000). The statistical description of visual signals. Unpublished manuscript.
- [48] MUMFORD, D. and DESOLNEUX, A. (2010). *Pattern Theory: The Stochastic Analysis of Real-World Signals. Applying Mathematics*. AK Peters, Natick, MA. [MR2723182](#)
- [49] NEYSHABUR, B., LI, Z., BHOJANAPALLI, S., LECUN, Y. and SREBRO, N. (2018). Towards understanding the role of over-parametrization in generalization of neural networks. In *International Conference on Learning Representations*.
- [50] PANARETOS, V. M. and ZEMEL, Y. (2016). Amplitude and phase variation of point processes. *Ann. Statist.* **44** 771–812. [MR3476617](#) <https://doi.org/10.1214/15-AOS1387>
- [51] PARK, D., RAMANAN, D. and FOWLKES, C. C. (2010). Multiresolution models for object detection. In *European Conference on Computer Vision* 241–254.
- [52] RAMSAY, J. O. and SILVERMAN, B. W. (2005). *Functional Data Analysis*, 2nd ed. *Springer Series in Statistics*. Springer, New York. [MR2168993](#)
- [53] RAWAT, W. and WANG, Z. (2017). Deep convolutional neural networks for image classification: A comprehensive review. *Neural Comput.* **29** 2352–2449. [MR3866781](#) https://doi.org/10.1162/neco_a_00990
- [54] ROSSI, F. and VILLA, N. (2006). Support vector machine for functional data classification. *Neurocomputing* **69** 730–742.
- [55] SCHMIDHUBER, J. (2015). Deep learning in neural networks: An overview. *Neural Netw.* **61** 85–117.
- [56] SIMARD, P. Y., STEINKRAUS, D. and PLATT, J. C. (2003). Best practices for convolutional neural networks applied to visual document analysis. In *International Conference on Document Analysis and Recognition* 958–963.
- [57] SOATTO, S. (2009). Actionable information in vision. In *IEEE International Conference on Computer Vision* 2138–2145.
- [58] SOTIRAS, A., DAVATZIKOS, C. and PARAGIOS, N. (2013). Deformable medical image registration: A survey. *IEEE Trans. Med. Imag.* **32** 1153–1190.
- [59] SRIVASTAVA, A., KLASSEN, E., JOSHI, S. H. and JERMYN, I. H. (2011). Shape analysis of elastic curves in Euclidean spaces. *IEEE Trans. Pattern Anal. Mach. Intell.* **33** 1415–1428.
- [60] SRIVASTAVA, A., WU, W., KURTEK, S., KLASSEN, E. and MARRON, J. S. (2011). Registration of functional data using Fisher–Rao metric. arXiv preprint. Available at [arXiv:1103.3817](#).
- [61] SZELISKI, R. (2010). *Computer Vision: Algorithms and Applications*. Springer, Berlin.
- [62] TANG, R. and MÜLLER, H.-G. (2008). Pairwise curve synchronization for functional data. *Biometrika* **95** 875–889. [MR2461217](#) <https://doi.org/10.1093/biomet/asn047>
- [63] TARASIUK, P. and PRYCEK, M. (2016). Geometric transformations embedded into convolutional neural networks. *J. Appl. Comput. Sci.* **24** 33–48.
- [64] TUCKER, J. D., WU, W. and SRIVASTAVA, A. (2013). Generative models for functional data using phase and amplitude separation. *Comput. Statist. Data Anal.* **61** 50–66. [MR3063000](#) <https://doi.org/10.1016/j.csda.2012.12.001>
- [65] VAN NOORD, N. and POSTMA, E. (2017). Learning scale-variant and scale-invariant features for deep image classification. *Pattern Recognit.* **61** 583–592.
- [66] VAPNIK, V. N. (1998). *Statistical Learning Theory. Adaptive and Learning Systems for Signal Processing, Communications, and Control*. Wiley, New York. A Wiley-Interscience Publication. [MR1641250](#)
- [67] WANG, J.-L., CHIOU, J.-M. and MÜLLER, H.-G. (2016). Functional data analysis. *Annu. Rev. Stat. Appl.* **3** 257–295.
- [68] YAMASHITA, R., NISHIO, M., DO, R. K. G. and TOGASHI, K. (2018). Convolutional neural networks: An overview and application in radiology. *Insights Imaging* **9** 611–629.
- [69] YAN, X., YANG, J., YUMER, E., GUO, Y. and LEE, H. (2016). Perspective transformer nets: Learning single-view 3D object reconstruction without 3D supervision. In *Neural Information Processing Systems*.
- [70] YANG, X., KWITT, R., STYNER, M. and NIETHAMMER, M. (2017). Quicksilver: Fast predictive image registration—a deep learning approach. *NeuroImage* **158** 378–396.
- [71] YOUNES, L., MICHOR, P. W., SHAH, J. and MUMFORD, D. (2008). A metric on shape space with explicit geodesics. *Atti Accad. Naz. Lincei, Rend. Lincei, Mat. Appl.* **19** 25–57. [MR2383560](#) <https://doi.org/10.4171/RLM/506>
- [72] ZHANG, T. (2004). Statistical behavior and consistency of classification methods based on convex risk minimization. *Ann. Statist.* **32** 56–85. [MR2051001](#) <https://doi.org/10.1214/aos/1079120130>
- [73] ZOU, D., CAO, Y., ZHOU, D. and GU, Q. (2020). Gradient descent optimizes over-parameterized deep ReLU networks. *Mach. Learn.* **109** 467–492. [MR4075425](#) <https://doi.org/10.1007/s10994-019-05839-6>