Extraction of digital wavefront sets using applied harmonic analysis and deep neural networks

Héctor Andrade-Loarca* Gitta Kutyniok* Ozan Öktem† Philipp Petersen‡

Abstract

Microlocal analysis provides deep insight into singularity structures and is often crucial for solving inverse problems, predominately, in imaging sciences. Of particular importance is the analysis of wavefront sets and the correct extraction of those. In this paper, we introduce the first algorithmic approach to extract the wavefront set of images, which combines data-based and model-based methods. Based on a celebrated property of the shearlet transform to unravel information on the wavefront set, we extract the wavefront set of an image by first applying a discrete shearlet transform and then feeding local patches of this transform to a deep convolutional neural network trained on labeled data. The resulting algorithm outperforms all competing algorithms in edge-orientation and ramp-orientation detection.

Keywords: Wavefront set, deep learning, convolutional neural networks, shearlets.
Mathematics Subject Classification: 35A18, 65T60, 68T10.

1 Introduction

Many scientific and industrial real-world applications require a precise understanding of how a model parameter, represented by a function, is transformed under some process that is described by an operator. Such analysis easily becomes very challenging, and one attempt at simplifying it is to treat the singular (non-smooth) and smooth parts of the function separately. In fact, a significant portion of the useful information is often contained in the singular part. For images, this singular part corresponds to edges in the image.

Microlocal analysis is a powerful mathematical theory that aims to precisely describe how the singular part of a function, or more generally a distribution, is transformed when acted upon by an operator. Since its introduction in the early 1970s by Sato [42] and Hörmander [23], it has proven itself useful in both pure and applied mathematical research. The crucial underlying observation in microlocal analysis is that the information about the location of the singularities (singular support) needs to be complemented with specifying those directions along which singularities may propagate. This extra directional ("microlocal") information is key in elucidating how singularities propagate when acted upon by a certain class of operators.

Microlocal analysis is by now a well-developed theory that can be used to study how singularities propagate under certain classes of operators. The latter includes Fourier integral operators such as most differential and pseudo-differential operators as well as many integral operators arising in integral geometry. Such operators are frequently encountered in analysis, scientific computing, and physical sciences [23, 6]. Microlocal analysis is also particularly useful in inverse problems, where the goal is to reliably recover a hidden model parameter (function) from a noisy transformed version. The goal here would be to recover the wavefront set of the function (image) given the noisy realization of a transformed version of the function. Such applications frequently arise when using imaging/sensing technologies where the transform is a pseudo-differential or Fourier integral operator [27]. While computing the action of such an operator, or its inverse, on a function

[1]Institut für Mathematik, Technische Universität Berlin, 10623 Berlin, Germany, {kutyniok,andrade}@math.tu-berlin.de
[2]Department of Mathematics, KTH - Royal Institute of Technology, SE-100 44 Stockholm, Sweden, ozan@kth.se
[3]Mathematical Institute, University of Oxford, OX2 6GG, Oxford, UK, Philipp.Petersen@maths.ox.ac.uk

can be highly non-trivial, it is often possible to precisely describe how the wavefront set is transformed without performing such computations.

A prime example showing the usefulness of microlocal analysis appears in the analysis of inverse problems arising in imaging applications. Here, the image is a function describing the interior structure of the object under study and the goal is to recover this image given a noisy realization of a transformed version of it. Recovering the image is however often not possible, either because the transformation relating the image to data is not invertible or because data is incomplete. On the other hand, full knowledge of the image is not always necessary. As an example, if one is looking for a tumor, then its location and shape are often sufficient for decision making whereas the exact values of the tumor density may be ignored. The location and shape of the tumor can be readily determined from the singular part of the image.

For the above reason, there exist a plethora of applications of microlocal analysis to tomographic imaging. In these applications, the transformation relating the image to data is the ray transform. This can be interpreted as a pseudo-differential operator, so it is possible to explicitly describe the relation between the wavefront set of the function (image) and its transformed version (tomographic data). Such relationships are referred to as *(microlocal) canonical relations*.

The canonical relation can also be used to identify which singularities that can be recovered from data without explicitly computing the inverse ray transform. This was done in [37] for the case when the 2D/3D ray transform is restricted to parallel lines and in [38] for an analysis in the region-of-interest limited angle setting. A related principle was derived in [13] for the 3D ray transform restricted to lines given by helical acquisition that is common in medical imaging. Similar principles hold for transforms integrating along other types of curves, for example, ellipses with foci on the x-axis and geodesics [47].

Besides the canonical relation, another observation is that recovering the wavefront set of a function from ray transform data is less ill-posed than recovering the function itself. The latter may in fact not even be possible as in region-of-interest tomography. This was demonstrated in [12] where the severely ill-posed reconstruction problem in limited angle tomography becomes mildly ill-posed if the wavefront set of the solution is provided as prior information. See also [38] for an application of this principle to cryo-electron tomography.

Additionally, the concept of the wavefront set is applied successfully also for problems involving transformations that are not necessarily pseudo-differential operators. Directed singularities, in particular edges or ridges, play a significant role in image processing to the extent that edge detection is one of the principal problems of this field. This is due to the fact, that edges in images present boundaries of objects and carry most of the information of the associated physical scene [46, 34, 3, 4]. Additionally, it has been argued in [34] that the human visual cortex performs multiple operations of image processing, the first of which is rough sketching involving edge detection.

1.1 Extraction of the wavefront set

One critical issue that hinders the usage of the wavefront set in real-world scenarios is that its extraction is challenging. Indeed, the definition of the wavefront set depends on the asymptotic behavior of the Fourier transform after a localization procedure and thus is practically hardly accessible. This issue is especially severe since in a practical situation one can only access a finite number of samples of an underlying function.

Certain transforms from applied harmonic analysis offer an alternative possibility to identify the wavefront set. In particular, the connection between the behavior of the curvelet- and shearlet transforms and the wavefront set was analyzed in [7, 29]. We will recall these results in Section 2. These approaches are based on analyzing the rate of decay of the respective transforms. While this point of view certainly makes the wavefront set more accessible, especially since it does not depend on an unspecified localization procedure, it is still inherently asymptotic.

In practical applications, we can never access the asymptotic behavior of the shearlet or Fourier transform but have to work with data up to a finite resolution. In other words, we have to rely on heuristics that, when presented with a finite number of samples of a ground truth function f, produce an estimate of the wavefront set of f. We will demonstrate in Section 3 that such a heuristic can never be successful in full generality, i.e., for every function $f \in L^2(\mathbb{R}^2)$. In fact, we prove in Theorem 3.3 that any such heuristic fails

on a dense subset of $f \in L^2(\mathbb{R}^2)$. This statement even holds, if the number of samples depends on f and the prediction of the wavefront set of f is only required to be approximately correct. On the other hand, in most applications, we might not be interested in extracting the wavefront set for every function $f \in L^2(\mathbb{R}^2)$ from finite samples but are content with a successful wavefront set extractor for functions from a function class of interest $\mathcal{C} \subset L^2(\mathbb{R}^2)$. One example of such a relevant function class would be that of functions modeling natural signals or images. In this situation, given the non-existence of general wavefront set extractors, we should thus strive to construct a heuristic that is as closely fitted to $\mathcal{C}$ as possible and as specialized as possible. This point of view, namely that a successful wavefront set extractor necessarily needs to be highly specialized in the sense of depending strongly on the targeted situation, will be our *guiding principle*.

1.2 Data-driven approach

As mentioned before, we are primarily interested in extracting the wavefront set from real-world data. Applying our guiding principle, we aim at constructing a wavefront set extractor that is as tightly tailored to this situation as possible. However, since these classes are empirically defined without any known underlying mathematical model, our best option is to adopt a data-driven approach.

Following this philosophy, in this paper we propose the following algorithm which we call DeNSE: We assemble a considerable set of labeled training data consisting of images and the associated wavefront set, or a suitable surrogate. Then, we train a classifier—in our case a *deep neural network*—to predict the wavefront set from the shearlet coefficients of the training data. We then test the resulting wavefront set predictor on unseen data. In Section 5, we present the construction of the algorithm alongside the training data that is used. We shall see below, in Section 6, that this method outperforms all conventional edge-orientation estimators as well as alternative data-driven methods including the current state-of-the-art. Moreover, we are unaware of any wavefront set extractor in the literature that goes beyond edge or ramp detection.

1.3 Expected impact of DeNSE

We anticipate our results to have the following impacts:

- *Fast solution of inverse problems:* As outlined above many Fourier integral or pseudo-differential operators are associated with canonical relations. If the wavefront set of the measurement data is known, then these relations allow the computation of the wavefront set of the solution of an inverse problem directly, without solving the inverse problem.

  In the inverse problems described in the introduction where one is exclusively interested in the singularities of the solution, this approach can be used to significantly speed-up the numerical solution of the problem.

- *Regularization of inverse problems:* As mentioned before, a priori knowledge of the wavefront set of the solution of an inverse problem can be used to regularize such problems. A similar idea, also based on shearlets for the identification of the wavefront set, has been used in [5] in the context of limited angle tomography. In contrast to this approach, our algorithm is independent of the underlying operator and due to this versatility can be applied in a wide variety of applications.

- *Edge detection:* Detecting edges, ridges, or points of higher-order non-smoothness is a sub-problem of wavefront set detection. As we will observe below, our algorithm outperforms all competing edge-orientation detector methods on a wide range of test sets. Moreover, the detection of points of higher-order non-smoothness has—to the best of our knowledge—not been pursued in the literature, but is possible with our approach without adaptations.

1.4 Basic concepts and notation

Below, we collect the notation used throughout this manuscript. This notation is fairly standard in the literature, and hence this subsection can be skipped and only consulted if a notation is unclear.

$\mathbb{R}$, $\mathbb{N}$, and $\mathbb{Z}$ denote the set of real number, natural numbers, and integers, respectively. Next, given a fixed point $x \in \mathbb{R}^n$ and $r > 0$, we use $B_r(x)$ to denote the ball of radius r in $\mathbb{R}^n$ with center at x. Likewise, $\mathbb{S}^{n-1}$ denotes the unit sphere in $\mathbb{R}^n$. Furthermore, the boundary of a domain $\Omega \subset \mathbb{R}^d$ for $d \in \mathbb{N}$ is denoted by $\partial\Omega$.

We will also make use of a number of function spaces. Let $\Omega \subset \mathbb{R}^d$ be a fixed domain for some $d \in \mathbb{N}$. Then, $L^2(\Omega)$ is the space of Lebesgue square-integrable functions on Ω, $C^n(\Omega)$ is the space of n-times continuously differentiable functions defined on Ω, and $H^n(\Omega)$ is the space of n-times weakly differentiable functions whose weak derivatives are in $L^2(\Omega)$. The support of a measurable function $f\colon \mathbb{R}^d \to \mathbb{R}$ is denoted by $\operatorname{supp} f$. Furthermore, the Fourier transform of a function $f \in L^1(\mathbb{R}^d)$ is defined as

$$\hat{f}(\xi) \coloneqq \int_{\mathbb{R}^d} f(x) e^{-2\pi i \langle x, \xi \rangle} dx \quad \text{for } \xi \in \mathbb{R}^d.$$

The Fourier transform operator $f \mapsto \hat{f}$ can be extended to an isometry on $L^2(\mathbb{R}^n)$ by Plancherel's identity. The Paley-Wiener space are functions whose Fourier transforms are compactly supported. More precisely, it PW_Λ for $\Lambda \in \mathbb{R}_+$ is defined by

$$PW_\Lambda \coloneqq \big\{ f \in L^2(\mathbb{R}^d) : \operatorname{supp} \hat{f} \in [-\Lambda, \Lambda]^d \big\}.$$

Finally, we use the Landau symbol $\mathcal{O}$ to describe asymptotic behavior, i.e., for functions $f, g\colon \mathbb{R} \to \mathbb{R}$ we write $f(x) = \mathcal{O}(g(x))$ as $x \to a \in \mathbb{R}$ whenever there exists a constant $c > 0$ such that $|f(x)| \leq c|g(x)|$ for all x in a neighborhood of a. Similarly, we write $f(x) = \mathcal{O}(g(x))$ as $x \to \infty$ whenever there exists a constant $c > 0$ such that $|f(x)| \leq c|g(x)|$ for $|x|$ sufficiently large.

2 Directional multiscale systems and the wavefront set

We start by formally introducing the notion of a wavefront set followed by the definition of the directional multiscale system of shearlets. We then show how shearlets can indeed be used to resolve the wavefront set of a distribution. Similar results also hold for other multiscale systems, like the curvelet transform [7] and the more general continuous parabolic molecules [18].

2.1 The wavefront set

The wavefront set can be defined for any distribution on a manifold, but since we only deal with $L^2(\mathbb{R}^2)$ functions, we restrict the definition to this setting.

Definition 2.1. *[24, Section 8.1] Let $f \in L^2(\mathbb{R}^2)$ and $k \in \mathbb{N}$. A point $(x, \lambda) \in \mathbb{R}^2 \times \mathbb{S}^1$ is a* k-regular directed point of f *if there exist open neighbourhoods U_x and U_λ of x and λ, respectively and a smooth function $\phi \in C^\infty(\mathbb{R}^2)$ with* $\operatorname{supp} \phi \subset U_x$ *and $\phi(x) = 1$ such that*

$$\big|\widehat{\phi f}(\xi)\big| \leq C_k \big(1 + |\xi|\big)^{-k} \quad \text{for all } \xi \in \mathbb{R}^2 \setminus \{0\} \text{ such that } \xi/|\xi| \in V_\lambda$$

holds for some $C_k > 0$. The k-wavefront set $\mathrm{WF}_k(f)$ *is the complement of the set of all k-regular directed points and the* wavefront set $\mathrm{WF}(f)$ *is defined as*

$$\mathrm{WF}(f) \coloneqq \bigcup_{k \in \mathbb{N}} \mathrm{WF}_k(f),$$

Let us next build up some intuition on this notion. The definition of the wavefront set is based on the well-known characterization of smoothness of a function in terms of the decay of its Fourier transform. More precisely, a function is smooth at a point if its Fourier transform decays faster than any polynomial in any direction. As an example, the *singular support* of f, i.e., the smallest closed set U such that $f_{|U^c} \in C^\infty(U^c)$, can be characterized in terms of the wavefront set as

$$\big\{ x \in \mathbb{R}^2 : (x, \lambda) \in \mathrm{WF}(f) \text{ for some } \lambda \in \mathbb{S}^1 \big\}.$$

The wavefront set is a refined notion of the singular support, since it not only indicates at which points a function is not smooth, but also contains the associated directions causing the non-smoothness. A common example considers f with a jump singularity across the smooth boundary of a fixed domain $D \subset \mathbb{R}^2$. Then, one can show [45, Chapter VI, Exercise 1.1] that

$$WF(f) = \big\{(x,\lambda) \in \mathbb{R}^2 \times \mathbb{S}^1 : x \in \partial D \text{ and } \lambda = n_x \text{ where } n_x \text{ is a normal on } \partial D \text{ at } x\big\}. \tag{2.1}$$

The wavefront set is a very powerful tool in mathematical analysis, but it is difficult to compute in practice. This is mainly due to the asymptotic criteria involved in its definition, which means computing the wavefront set requires computing the "full" Fourier transform at every point.

Continuous transforms associated to certain directional multiscale systems offer a convenient remedy. As an example, the shearlet transform automatically performs the necessary time-frequency-orientation localization described above, thereby "resolving" the wavefront set in a sense described in the following subsection.

2.2 Shearlets

The shearlet transform, which was introduced in [19], is based on applying translation, anisotropic dilation, and shearing to generator functions. To dilate and shear a function, we define the following two matrices:

$$A_a \coloneqq \begin{pmatrix} a & 0 \\ 0 & \sqrt{a} \end{pmatrix}, \quad \widetilde{A}_a \coloneqq \begin{pmatrix} \sqrt{a} & 0 \\ 0 & a \end{pmatrix}, \quad \text{and} \quad S_s \coloneqq \begin{pmatrix} 1 & s \\ 0 & 1 \end{pmatrix} \quad \text{for } a > 0 \text{ and } s \in \mathbb{R}.$$

Next, given $(a,s,t) \in \mathbb{R}_+ \times \mathbb{R} \times \mathbb{R}^2$, $\psi \in L^2(\mathbb{R}^2)$, and $x \in \mathbb{R}^2$, define

$$\psi_{a,s,t,1}(x) \coloneqq a^{-\frac{3}{4}} \psi\left(A_a^{-1} S_s^{-1}(x-t)\right) \quad \text{and} \quad \psi_{a,s,t,-1}(x) \coloneqq a^{-\frac{3}{4}} \widetilde{\psi}\left(\widetilde{A}_a^{-1} S_s^{-T}(x-t)\right), \tag{2.2}$$

where $\widetilde{\psi}(x_1,x_2) \coloneqq \psi(x_2,x_1)$ for all $x = (x_1,x_2) \in \mathbb{R}^2$. Following [17], we define the continuous shearlet transform as follows:

Definition 2.2 (Continuous shearlet transform)**.** *Let $\psi \in L^2(\mathbb{R}^2)$. Then the family of functions $\psi_{a,s,t,\iota} \colon \mathbb{R}^2 \to \mathbb{R}$ parametrized by $(a,s,t,\iota) \in \mathbb{R}^+ \times \mathbb{R} \times \mathbb{R}^2 \times \{-1,1\}$ that are defined in* (2.2) *is called a* shearlet system*. The corresponding* (continuous) shearlet transform *is defined by*

$$\mathcal{SH}_\psi : L^2(\mathbb{R}^2) \to L^\infty\big(\mathbb{R}^+ \times \mathbb{R} \times \mathbb{R}^2 \times \{-1,1\}\big) \quad \textit{where} \quad \mathcal{SH}_\psi(f)(a,s,t,\iota) \coloneqq \langle f, \psi_{a,s,t,\iota}\rangle.$$

As we shall see next, if the generator function ψ has directional vanishing moments, then the asymptotic behavior as $a \to 0$ of the continuous shearlet transform of an L^2-function f characterizes its wavefront set. The precise statement in [17] reads as follows.

Theorem 2.3. *Let $f \in L^2(\mathbb{R}^2)$ and assume $(x_0,\lambda_0) \in \mathbb{R}^2 \times \mathbb{S}^1$ is a k-regular directed point of f for some $k \in \mathbb{N}$. Next, consider a continuous shearlet system with generator function $\psi \in H^l(\mathbb{R}^2)$ with Fourier transform $\widehat{\psi} \in L^1(\mathbb{R}^2)$ where ψ has $m \in \mathbb{N}$ vanishing moments in x_1-direction, i.e.,*

$$\int_{\mathbb{R}^2} \frac{\big|\widehat{\psi}(\xi_1,\xi_2)\big|^2}{|\xi_1|^{2m}} d\xi < \infty.$$

Finally, assume ψ displays the following asymptotic behavior:

$$\big|\psi(x)\big| = \mathcal{O}\big((1+|x|)^{-p}\big) \quad \textit{for } |x| \to \infty.$$

Then, there exist a neighborhood $U_0 \subset \mathbb{R}^2$ of x_0 and a neighborhood $S_0 \subset \mathbb{S}^1$ of λ_0 such that

$$\big|\mathcal{SH}_\psi(f)(a,s,x,\iota)\big| = \mathcal{O}\Big(a^{\frac{p}{2}-\frac{3}{4}} + a^{\frac{m}{4}} + a^{\frac{3k}{4}-\frac{3}{4}} + a^{\frac{3l}{4}}\Big) \quad \textit{as } a \to 0$$

for all $x \in U_0$ and all $s \in \mathbb{R}$ and $\iota \in \{-1, 1\}$ such that $\lambda(s, \iota) \in S_0$, where

$$\lambda(s,\iota) \coloneqq \begin{cases} \left(\frac{1}{\sqrt{s^2+1}}, \frac{s}{\sqrt{s^2+1}}\right) & \text{if } \iota = 1, \\ \left(\frac{s}{\sqrt{s^2+1}}, \frac{1}{\sqrt{s^2+1}}\right) & \text{if } \iota = -1. \end{cases} \tag{2.3}$$

Remark 2.4. *Under suitable assumptions on the shearlet generator ψ, the converse of Theorem 2.3 holds as well. More precisely, following [17], assume that ψ is sufficiently regular for any $k \in \mathbb{N}$. Next, let $(x_0, \lambda_0) \in \mathbb{R}^2 \times \mathbb{S}^1$ and assume there exist a neighborhood $U_0 \subset \mathbb{R}^2$ of x_0 and a neighborhood $S_0 \subset \mathbb{S}^1$ of λ_0 such that*

$$\left|\mathcal{SH}_\psi(f)(a, s, x, \iota)\right| = \mathcal{O}(a^n) \quad \text{as } a \to 0,$$

holds for sufficiently large $n \in \mathbb{N}$ uniformly for $x \in U_0$ and all $s \in \mathbb{R}$, $\iota \in \{-1, 1\}$ such that $\lambda(s, \iota) \in S_0$. Then, $(x_0, \lambda_0) \notin \mathrm{WF}_k(f)$.

Theorem 2.3 and Remark 2.4 demonstrate that the wavefront set is completely determined by the decay properties of the shearlet transform. This implies that in the continuous setting, one can compute the wavefront set of a function by first computing its continuous shearlet transform, then analyzing the pairs of point and direction where this shearlet transform exhibits rapid decay as $a \to 0$.

Theorem 2.3 and Remark 2.4 were first reported in [29] in a setting restricted to a specific shearlet generator (called the "classical shearlet"). Moreover, results similar to Theorem 2.3 and Remark 2.4 were obtained in [7] for the curvelet transform and in [14] for transforms stemming from general group representations. As shown in [18], all transforms that belong to the category of continuous parabolic molecules admit a similar characterization of the wavefront set. Finally, one can also use the shearlet transform to classify certain geometric properties of the singularities of a function that goes beyond differentiating between rapid and non-rapid decay of the shearlet transform, see e.g. [48, 20, 32].

3 Wavefront sets from sampled data

In this section, we will analyze to what extent it is possible to construct an operator that maps a finite number of point samples of a function f to an estimate of the wavefront set of f. This situation reflects practical applications, e.g., images are only represented as pixels representing point samples of a real-valued function.

To make the connection between a sampled function and its wavefront set more precise, it is convenient to adopt a point of view that is based on the Shannon sampling theorem. We will state this theorem in Subsection 3.1 and – based on it – we will introduce the notion of an approximate wavefront set extractor in Subsection 3.2. Finally, in Subsection 3.3 we show that any approximate wavefront set extractor that predicts the wavefront set of a function on $\mathbb{R}^2$ from a finite number of sample values will fail on a dense subset of $L^2(\mathbb{R}^2)$. This result holds even if the sampling density is allowed to depend on the function.

3.1 Sampling theorem and Paley-Wiener spaces

The sampling theorem states that every band-limited function f can be written as a sum of shifted cardinal sine functions weighted by point samples of f. In other words, a band-limited function is fully determined by its values on a discrete grid. To give the precise statement, we introduce the Paley-Wiener spaces. Given $\Lambda > 0$, the Paley-Wiener space $\mathcal{PW}_\Lambda \subset L^2(\mathbb{R}^d)$ is defined as

$$\mathcal{PW}_\Lambda \coloneqq \left\{ f \in L^2\left(\mathbb{R}^d\right) : \operatorname{supp}\left(\widehat{f}\right) \subset [-\Lambda, \Lambda]^d \right\}.$$

We define the d-dimensional sinc-function as

$$\operatorname{sinc}_d(x) \coloneqq \prod_{i=1}^{d} \frac{\sin(\pi x_i)}{\pi x_i}, \quad \text{where } x = (x_1, \ldots, x_d) \in \mathbb{R}^d.$$

Bearing in mind the above notation, we now state the sampling theorem, see, e.g., [33].

Theorem 3.1 (Sampling theorem)**.** *Let $f \in L^2(\mathbb{R}^d) \cap C(\mathbb{R}^d)$ and $\Lambda > 0$. Then*

$$f \in \mathcal{PW}_\Lambda \iff f(x) = \sum_{n \in \mathbb{Z}^d} f\left(\frac{n}{\Lambda}\right) \operatorname{sinc}_d(\Lambda \cdot x - n) \quad \text{for all } x \in \mathbb{R}^d.$$

In particular, for every summable sequence $(y_k)_{k \in \mathbb{Z}^2}$ we can define

$$f(x) \coloneqq \sum_{k \in \mathbb{Z}^2} y_k \operatorname{sinc}_2(\Lambda \cdot x - k), \quad \text{for } x \in \mathbb{R}^d,$$

and, by Theorem 3.1 f, is band-limited. Furthermore, since $\operatorname{sinc}_2(m-n)$ vanishes for every $m, n \in \mathbb{Z}^2$ such that $m \neq n$, we observe that $f(m/\Lambda) = y_m$ for all $m \in \mathbb{Z}^2$. In other words, every sequence on a grid defines an associated interpolating band-limited function and conversely, every band-limited function is uniquely determined by its values on a discrete grid.

As a consequence, the problem of extracting the wavefront set of a function $f \in L^2(\mathbb{R}^2)$ from its discrete sampled values $(f(m/\Lambda))_{m \in \mathbb{Z}^2}$ can be re-stated as extracting the wavefront set of f from its projection onto a Paley-Wiener space, i.e., $P_{\mathcal{PW}_\Lambda}(f) =: P_\Lambda(f)$.

3.2 Wavefront set extractors

As already stated, the problem of extracting the wavefront set from samples on a grid is equivalent to extracting the wavefront set given the projection onto a Paley-Wiener space. There are multiple conceivable notions of a wavefront set extractor. First, for $\Lambda > 0$, we could ask for a map

$$\mathrm{DWF}_\Lambda : \mathcal{PW}_\Lambda \to P\left(\mathbb{R}^2 \times \mathbb{S}^1\right) \quad \text{such that } \mathrm{DWF}(P_\Lambda f) = \mathrm{WF}(f) \text{ for all } f \in L^2(\mathbb{R}^2). \tag{3.1}$$

Here $P(\mathbb{R}^2 \times \mathbb{S}^1)$ denotes the power set of $\mathbb{R}^2 \times \mathbb{S}^1$. Essentially, this map requests extraction of the wavefront set of a function f from knowledge of the samples of f on a fixed grid. It is clear that such a map, DWF_Λ, cannot exist, since it will fail for functions f that have fine structures which cannot be detected by coarse sampling. For example, a function that vanishes on every grid point of $\mathbb{Z}^2/\Lambda$ while having a non-trivial wavefront set would be classified the same as the zero function.

A more reasonable model for a wavefront set extractor should give an approximate prediction of the wavefront set that eventually improves as the sampling density increases. To weaken this statement even further, we might only ask for approximate extraction of the wavefront set in one point. For a fixed set $W \subset \mathbb{R}^2 \times \mathbb{S}^1$ and a point $x \in \mathbb{R}^2$, we therefore define

$$W_x \coloneqq \left\{\lambda \in \mathbb{S}^1 : (x, \lambda) \in W\right\}.$$

We can now model the approximation described above by considering a sequence of wavefront set extractors given by

$$\mathrm{DWF}_j \colon \mathcal{PW}_j \to \mathcal{P}\left(\mathbb{R}^2 \times \mathbb{S}^1\right) \quad \text{for } j \in \mathbb{N} \tag{3.2}$$

such that, for fixed $x \in \mathbb{R}^2$, and all $f \in L^2(\mathbb{R}^2)$,

$$d_H\left(\overline{\mathrm{DWF}_j(P_j(f))_x}, \overline{\mathrm{WF}(f)_x}\right) \to 0. \tag{3.3}$$

Here d_H denotes the Hausdorff distance with the convention $d_H(X, \varnothing) = d_H(\varnothing, X) := 1$ for any non-empty compact subset $X \subset \mathbb{S}^1$ and $d_H(\varnothing, \varnothing) := 0$. Recall that with this definition d_H is a metric on compact subsets of $\mathbb{S}^1$ (including the empty set).

A sequence as in (3.2) satisfying (3.3) yields an approximate extraction of the wavefront set of f at x from point samples of f where the sampling density may depend on f. This observation motivates the following definition.

Definition 3.2. *A sequence* $(\mathrm{DWF}_j)_{j\in\mathbb{N}}$ *of mappings as in* (3.2) *is called an* approximate wavefront set extractor. *We say that an approximate wavefront set extractor is*

- clairvoyant *at* $x \in \mathbb{R}^2$ *if the sequence satisfies* (3.3) *at* x *for all* $f \in L^2(\mathbb{R}^2)$, *and*
- ignorant *to* $f \in L^2(\mathbb{R}^2)$ *at* $x \in \mathbb{R}^2$ *if* $d_H(\overline{\mathrm{DWF}_j(P_j(f))_x}, \overline{\mathrm{WF}(f)_x}) \not\to 0$ *as* $j \to \infty$.

3.3 Non-existence of clairvoyant approximate wavefront set extractors

We will observe below that, for every $x \in \mathbb{R}^2$, there does not exist a clairvoyant approximate wavefront set extractor. Even more severely, every approximate wavefront set extractor is ignorant to a dense subset of $L^2(\mathbb{R}^2)$ at x.

Theorem 3.3. *For every* $x \in \mathbb{R}^2$ *and every approximate wavefront extractor* $(\mathrm{DWF}_j)_{j\in\mathbb{N}}$, *there exists a dense subset* $\mathcal{M} \subset L^2(\mathbb{R}^2)$ *such that* $(\mathrm{DWF}_j)_{j\in\mathbb{N}}$ *is ignorant to all* $f \in \mathcal{M}$ *at* x. *In particular, no approximate wavefront set extractor is clairvoyant at* x.

Proof. The proof proceeds in two steps. For a given approximate wavefront set extractor, $(\mathrm{DWF}_j)_{j\in\mathbb{N}}$, and a point $x \in \mathbb{R}^2$, we construct a function $q \in L^2(\mathbb{R}^2)$ such that $(\mathrm{DWF}_j)_{j\in\mathbb{N}}$ is ignorant to q at x. Second, we show that the set of such functions is dense in $L^2(\mathbb{R}^2)$.

Step 1: Notice that Definition 2.1 implies

$$\mathrm{WF}(f_1 + f_2) = \mathrm{WF}(f_1) \quad \text{for every } f_1 \in L^2(\mathbb{R}^2) \text{ and } f_2 \in C^\infty(\mathbb{R}^2) \cap L^2(\mathbb{R}^2). \tag{3.4}$$

We now choose a function g with wavefront set $\{x\} \times \mathbb{S}^1$ where $x \in \mathbb{R}^2$ is arbitrary, e.g., a function $g \in C^\infty(\mathbb{R}^2 \setminus \{x\}) \cap L^2(\mathbb{R}^2)$ with a cusp at x. Then, by (3.4), we can conclude that $\mathrm{WF}(g - P_j g) = \{x\} \times \mathbb{S}^1$ holds for every $j \in \mathbb{N}$. Moreover, by construction, we have $d_H(\overline{\mathrm{WF}(g)_x}, \varnothing) = 1$. To define the desired function q, we first set

$$\begin{aligned} q_0 &\coloneqq P_1 g, \\ q_n &\coloneqq \begin{cases} q_{n-1} + (P_n g - P_{n-1} g) & \text{if } \mathrm{DWF}_{j-1}(P_{n-1} q_{n-1})_x = \varnothing, \\ q_{n-1} & \text{otherwise,} \end{cases} \end{aligned} \tag{3.5}$$

for all $n \geq 1$. By definition,

$$(P_n g - P_{n-1} g) \perp (P_m g - P_{m-1} g) \quad \text{for all } n \neq m.$$

Hence, by the Pythagorean theorem,

$$\sum_{n\in\mathbb{N}} \|P_n g - P_{n-1} g\|_2^2 = \|g\|_2^2 < \infty. \tag{3.6}$$

It now follows from (3.6) and (3.5) that q_n is a Cauchy sequence. Therefore q_n converges to a limit $q \in L^2(\mathbb{R}^2)$. Furthermore, one of the following statements holds:

(1) $\overline{\mathrm{DWF}_j(P_j q_j)_x}$ does not converge for $j \to \infty$;

(2) $\overline{\mathrm{DWF}_j(P_j q_j)_x}$ converges to a limit W such that $d_H(W, \overline{\mathrm{WF}(q)_x}) \geq 1/4$;

(3) $\overline{\mathrm{DWF}_j(P_j q_j)_x}$ converges to a limit W such that $d_H(W, \overline{\mathrm{WF}(q)_x}) < 1/4$.

In Cases (1) and (2), we directly obtain that DWF_j is ignorant to q at x. In Cases (3), we obtain that there exists some j_0 such that

$$d_H\big(\overline{\mathrm{DWF}_j(P_j q_j)_x}, \overline{\mathrm{WF}(q)_x}\big) < 1/2 \quad \text{for all } j \geq j_0. \tag{3.7}$$

We now consider the cases $\mathrm{WF}(q)_x = \varnothing$ and $\mathrm{WF}(q)_x \neq \varnothing$ separately. If $\mathrm{WF}(q)_x = \varnothing$, then (3.7) implies that $\mathrm{DWF}_j(P_j q_j)_x = \varnothing$ for all $j \geq j_0$ since no subset of $P(\mathbb{S}^1) \setminus \{\varnothing\}$ has a distance less than 1 to the empty set. Therefore,

$$q - P_{j_0} q = \sum_{j > j_0} (P_j g - P_{j-1} g) = g - P_{j_0} g. \tag{3.8}$$

We obtain from (3.8) that $\varnothing = \overline{\mathrm{WF}(q)_x} = \mathrm{WF}(g)_x = \mathbb{S}^1$ which is a contradiction.

If $\mathrm{WF}(q)_x \neq \varnothing$, then $d_H(\overline{\mathrm{WF}(q)_x}, \varnothing) = 1$. By the triangle inequality, this yields that there exists some j_0 such that $\mathrm{DWF}_j(P_j q_j)_x \neq \varnothing$ for all $j \geq j_0$. Therefore, $q = q_{j_0} \in \mathcal{PW}_{j_0}$ by definition, which implies that $W(q)_x = \varnothing$. Hence, Case (3) does not occur, i.e., $(\mathrm{DWF}_j)_{j\in\mathbb{N}}$ is ignorant to q at x.

Step 2: For an arbitrary $f \in L^2(\mathbb{R}^2)$, there exists $j_1 \in \mathbb{N}$ such that

$$\|f - P_{j_1} f\|_2 \leq \frac{\epsilon}{2} \quad \text{and} \quad \|g - P_{j_1} g\|_2 \leq \frac{\epsilon}{2}.$$

Define $q_{j_1} = P_{j_1} f$ and, for every $n \geq j_1$, we define q_n as in (3.5). It is clear that q_n converges to a limit q_f. Also, it is straightforward to show that $\|q_f - f\|_2 \leq \epsilon$. Now using the same arguments as in Step 1, it follows that $(\mathrm{DWF}_j)_{j\in\mathbb{N}}$ is ignorant to q_f. □

Remark 3.4. *The following observations provide additional insight into Theorem 3.3:*

(1) The Theorem 3.3 and its proof also hold when "wavefront set" is replaced by "singular support".

(2) The arguments in the proof of Theorem 3.3 are independent from the domain $\mathbb{R}^2$. Indeed, the same result holds for functions defined in an open domain $\Omega \subset \mathbb{R}^2$ and $x \in \Omega$. Here we define the wavefront set of $f \in L^2(\Omega)$ as

$$\Big\{(x,\lambda) \in \Omega \times \mathbb{S}^1 : (x,\lambda) \in \mathrm{WF}\left(\tilde{f}\right) \text{ where } \tilde{f} = f \text{ on } \Omega \text{ and } \tilde{f} = 0 \text{ elsewhere}\Big\}.$$

(3) Theorem 3.3 yields that there is no clairvoyant approximate wavefront set extractor. Even more severely, every approximate wavefront set extractor fails on a dense subset of $L^2(\mathbb{R}^2)$. As a consequence, an approximate wavefront set extractor can never be successful on an open subset of $L^2(\mathbb{R}^2)$. Hence, if we want to have any chance of building a successful wavefront set extractor, then it must be as strongly adapted to the underlying signal class as possible.

4 Wavefront set and edge detectors

The characterization of a wavefront set given in (2.1) implies that detecting the wavefront set of a piece-wise smooth function with singularities along a smooth curve is equivalent to detecting edges and their normal direction. Edge detection, which is one of the most well-studied problems in image processing, is therefore a sub-problem to wavefront set extraction. However, we are unaware of any wavefront set extractor in the literature that goes beyond edge or ramp detection.

We next recall some edge-orientation detectors from the literature. We start by approaches based on filters, which is followed by a review of methods based on directional systems from applied harmonic analysis. We then compare these methods to the guiding principle. Finally, we will comment on recent data-driven algorithms for edge detection.

4.1 Filter-based edge-orientation detectors

The traditional way of detecting edges in digital images is to convolve the image with suitable convolution kernels to enhance edge-like features. These features can then be extracted using simple rules. For example, convolution with local difference filters leads to the Roberts [40], Sobel [11, 44], and Prewitt [36] operators. In [34] the authors convolve an image with the Laplacian of a Gaussian function. Here, the zeros of the resulting image are taken as estimate for the positions of edges in the original.

Potentially the most famous edge-orientation detection algorithm of this category is the Canny edge detector [8]. In this algorithm, a function g is convolved with a Gaussian window G_σ with standard deviation $\sigma > 0$, the magnitude of the local gradient of g is then defined as $|\nabla_x g * G_\sigma|$, and the associated gradient direction is

$$\left(\cos\left(\arctan\left(\frac{\partial_{x_2}[g * G_\sigma]}{\partial_{x_1}[g * G_\sigma]}\right)\right), \sin\left(\arctan\left(\frac{\partial_{x_2}[g * G_\sigma]}{\partial_{x_1}[g * G_\sigma]}\right)\right)\right). \tag{4.1}$$

If the magnitude of the local gradient exceeds a certain threshold, then it is labeled as an edge with the associated direction given by (4.1).

The Canny edge detector has some obvious drawbacks. The choice of the standard deviation σ of the Gaussian window strongly influences the performance of the algorithm to the extent that a high σ improves the robustness against noise, but might also remove high-frequency components. In fact, there is no universally appropriate choice of σ. For example, the contrast may vary strongly even within a single image and, in this case, the edge detector can perform well in one part of the image and fail in another.

There have been many efforts in addressing the shortcomings of the Canny edge detector. In [35] one extracts the positions and orientations of not only jump singularities but also of composite edges like ramp- and hat-like singularities. Most recently, [1] develops algorithms that apply even more high-level heuristics to a set of patches retrieved from applying oriented gradient operators.

4.2 Edge-orientation detectors based on directional systems

An alternative to the filter-based approaches is to base the detector on multiscale directional transforms. These transforms include shearlets, which were introduced in Subsection 2.2, as well as curvelets, ridgelets, wavelets, or bandlets, see [25] for a survey.

The idea is to first transform the given image using a directional system. Next, one applies certain heuristics to the result by using the theoretical information on the behavior of the underlying transform at directed edge points. In principle, the transforms given by directional systems mentioned above can be written as a series of convolutions with respect to directed filters. Hence, this approach can, in a sense, be understood as a special case of the edge detectors in the previous section.

An example of such an approach is the shearlet based algorithm in [48], which we will now explain in detail. In fact, this algorithmic approach served as the main inspiration for our forthcoming algorithm in Section 5. The approach in [48] seeks to detect and classify edges of functions of the form

$$f(x) \coloneqq \sum_{i=1}^{N} u_i(x)\chi_{B_i}(x) \quad \text{for } x \in \mathbb{R}^2, \tag{4.2}$$

where $B_i \subset (0,1)^2$ are such that ∂B_i are piece-wise smooth and $u_i \in C^\infty(\mathbb{R}^2)$. The singular support of f is the set $\bigcup_{i=1}^N \partial B_i$, which is also called the *set of singularity curves of* f. The algorithm is based on computing a sampled version of the shearlet transform of f, i.e., the algorithm computes

$$\left\{\mathcal{SH}(f)(2^{-j}, 2^{-\frac{j}{2}}k, x, \iota) : j \in \mathbb{N}, k \in \mathbb{Z}, x \in c\mathbb{Z}^2, j_0 \le j \le J, |k| \le 2^{\frac{j}{2}}, \iota \in \{-1, 1\}\right\} \tag{4.3}$$

for given $j_0, J \in \mathbb{N}$ and $c > 0$. This algorithmic step can be performed using standard software libraries such as the ShearLab toolbox [30], see also Section 5.1. This computation is followed by a series of heuristics that are applied to the coefficients (4.3) and which lead to a classification of points as point singularities, directed

edges, corners, or smooth points. The heuristics are based on theoretical insights into the behavior of the shearlet transform at points of different regularities, which in turn are very closely related to the results of Theorem 2.3 (see also Remark 2.4). The classification scheme is precisely described in [48, Chapter IV] and the heuristics include:

Step 1: Fitting a polynomial (in j) to

$$\sum_{\substack{|k^*|\leq 2^{j/2}\\ \iota\in\{-1,1\}}} \mathcal{SH}(f)\left(2^{-j}, 2^{-\frac{j}{2}}k^*, x, \iota\right) \quad \text{for fixed } x \in (0,1)^2.$$

If this this polynomial does not decay for $j \to J$, then the jump is classified as a point singularity.

Step 2: Let T be a fixed threshold. If for a fixed scale j^*

$$\sum_{\substack{|k^*|\leq 2^{j^*/2}\\ \iota\in\{-1,1\}}} \mathcal{SH}(f)\left(2^{-j^*}, 2^{-\frac{j^*}{2}}k^*, x, \iota\right) < T,$$

then x is considered a regular point.

Step 3: If both cases above did not yield a classification, then one analyzes the vector

$$c_{J,x,\iota}(k) \coloneqq \left(\mathcal{SH}(f)(2^{-J}, 2^{-\frac{j}{2}}k, x, \iota)\right)_{|k|\leq 2^{J/2}} \quad \text{for fixed } x \in (0,1)^2 \text{ and } \iota \in \{-1,1\}.$$

1. If $c_{J,x,\iota}(k) \sim c$ for all k, then the point x is identified as a regular point.
2. If there is one direction k_1 such that $c_{J,x,\iota}(k) \ll c_{J,x,\iota}(k_1)$ for all $k \neq k_1$, then x is considered a point in one of the singularities of f.
3. If there are two directions k_1, k_2 such that $c_{J,x,\iota}(k_1) \sim c_{J,x,\iota}(k_2)$ and $c_{J,x,\iota}(k) \ll c_{J,x,\iota}(k_1)$ for all $k \neq k_1, k_2$, then x is classified as a corner point of the the jump curve of f.

The resulting algorithm is very powerful in detecting the set of singularity curves and the associated orientations of edges of piece-wise smooth functions. Nonetheless, the algorithm requires some significant tuning, e.g., the threshold T needs to be selected, and one needs to define criteria to distinguish between the cases 1-3 above. Moreover, the heuristics neglect the behavior of $\mathcal{SH}(f)(2^{-j}, 2^{-\frac{j}{2}}k, x, \iota)$ when j and k vary simultaneously.

Additionally, we mention an algorithm in [39] that is based on the complex shearlet transform. This algorithm computes two shearlet transforms, one with a symmetric and one with an anti-symmetric generator. The relationship between both is then used to determine if a point is an edge, a ridge, or a smooth point.

4.3 Heuristics and our guiding principle

In the previous sections, we gave a detailed account of the heuristics behind the Canny edge detector [8] and the shearlet based algorithm in [48]. We will now analyze how these heuristic-based algorithms perform with respect to the guiding principle of being as tightly adapted to the class of natural images as possible.

The Canny edge detector is not tightly adapted to images, but rather to functions that are piece-wise smooth and it assumes the jump curve has a given contrast that remains relatively constant across the image. The same argument holds for the shearlet based algorithm in [48]. In fact, its theoretical motivation comes from the treatment of piece-wise smooth functions as in (4.2), which are not natural images.

The parameters in these algorithms, such as the threshold σ in the Canny edge detector and the threshold T in the shearlet based algorithm in [48], can be adapted to the underlying functions. This offers a possibility to adapt the algorithms to natural images. However, as we argued in the respective sections, it is impossible to find a universally suitable setting for these parameters. We see that in both cases the guiding principle is violated.

Finally, it is worth noting that similar issues arise for all algorithms mentioned in the previous two subsections.

4.4 Data-driven edge-orientation detectors

As we have already advocated previously, the only way to comply with the guiding principle is to use a data-driven algorithm. For edge detection, this idea has already been followed in a series of papers. For example, [10] uses supervised learning of an edge classifier based on a technique called probabilistic boosting tree. Another approach is DeepEdge that uses a convolutional neural network taking as an input candidate edges produced by a Canny edge detector as well as patches of the original image [2]. To this input one then applies parts of the KNet [28] for feature extraction and a network with two fully-connected layers for classification. Finally, the algorithms SEAL (simultaneous edge alignment and learning) [50] and the CASENet (category-aware semantic edge detection network) [49] perform high-level edge analysis with highly complex deep neural networks. The underlying CASENet is a 101-layered network. These methods are state-of-the-art for segmentation and edge detection.

5 Computing the digital wavefront set with shearlets and deep learning

We propose an algorithm that replaces the heuristic approach of the shearlet-based edge detection and classification algorithm of [48] by a data-driven approach. Concretely, instead of hand-crafted heuristics, we train a deep neural network using a variety of training data, adapted to the classification procedure at hand. The neural network takes as input the shearlet coefficients of an image and produces a set of point-direction pairs that are classified as elements of the wavefront set. We will present the construction of the classifier below and then present the computational realization of our algorithm in Subsection5.4 at the end of this section.

5.1 Digital shearlet transform

The classifier to be constructed below is based on the shearlet transform of a digital image. Therefore, we need to work with a digitized shearlet transform, defined on a digital domain of pixel images. The digital shearlet transform was introduced in [31] and is defined as follows:

Let $M \in \mathbb{N}$, $J \subset \mathbb{N}$ be finite, $k_j \subset \mathbb{N}$ for all $j \in J$ and $K_j \coloneqq [-k_j, \dots, 0, \dots, k_j]$. Then, we pick $2\sum_{j\in J} K_J + 1$ matrices in $\mathbb{R}^{M\times M}$. We denote these matrices by ϕ^{dig} and $\psi^{dig}_{j,k,\iota}$ for $j \in J, k \in K_j, \iota \in \{-1, 1\}$. To make the connection to the classical shearlet transform, we can think of $\psi^{dig}_{j,k,\iota}$ as a digitized version of $\psi_{2^{-j},2^{-j/2}k,0,\iota}$ and of ϕ^{dig} as a digitized version of a low frequency filter. A concrete construction of the matrices ϕ^{dig} and $\psi^{dig}_{j,k,\iota}$ can be found in [31]. Then, we define the digital shearlet transform of an image $I \in \mathbb{R}^{M\times M}$ by

$$\mathrm{DSH}(I)(j,k,m,\iota) \coloneqq \begin{cases} \left\langle I, T_m\psi^{dig}_{j,k,\iota}\right\rangle & \text{if } \iota \in \{-1,1\}, \\ \left\langle I, T_m\phi^{dig}\right\rangle & \text{if } \iota = 0, \end{cases}$$

where $j \in J, k \in K_j$, $m \in \{1, \dots, M\}^2$, and $T_m : \mathbb{R}^{M\times M} \to \mathbb{R}^{M\times M}$ circularly shift the entries of the elements of a matrix by m.

Thus the digital shearlet transform of an image $I \in \mathbb{R}^{M\times M}$ is a stack of $2\sum_{j\in J}(K_J - 1) + 1$ matrices of dimension $M \times M$. In all our numerical experiments, we fixed $J = 4$ and $K_j = 2^{\lceil j/2+1\rceil} + 1$ and therefore, $2\sum_{j\in J}(K_J - 1) + 1 = 49$.

The computation of the digital shearlet transform is performed by using the Julia implementation of ShearLab [30] (`www.shearlab.org/software`).

5.2 Network architecture

We train a neural network which, given a patch of the shearlet coefficients of a function, produces a prediction of which directions belong to the wavefront set of the function at the position associated with this patch.

These patches are of size $21 \times 21 \times 49$.

The network architecture consists of four convolutional layers, with 2×2 max pooling, ReLU activation and batch normalization, followed by a fully-connected layer with 1024 neurons, softmax activation function and a one-dimensional output. The network architecture is depicted in Figure 1. We chose this architecture since it performed well in a series of tests while being of moderate size. Here we focused on networks with only a few layers because we expect that the shearlet transform already acts as the correct feature extractor of the problem. Therefore, the classifier does not need to learn the correct data representation. Nonetheless, it is conceivable that a deeper and larger neural network architecture could potentially lead to improvements for the classification results below.

We pick 180 directions $(\theta_i)_{i=1}^{180}$. For each θ_i, we then train a network Φ_i with the described architecture by passing patches of shearlet coefficients of images $I \in \mathbb{R}^{M \times M}$ of the form

$$(\mathrm{DSH}(I)(j,k,m,\iota))_{j \in J, k \in K_j, \iota \in \{-1,0,1\}, m \in [m_1^*-10, m_1^*+10] \times [m_2^*-10, m_2^*+10]}, \tag{5.1}$$

where $m^* \in \{11, \dots, M-10\}^2$, to the network. The associated label to a batch of (5.1) is 1 if I has an edge with direction θ_i at m^* and 0 else. In total, this procedure yields 180 digital classifiers. We train one more network with the same data, but the label is 1 if I has no singularity at m^* and 0 else. This additional classifier is used in test cases where all competing algorithms only perform edge detection and not edge-orientation detection.

The final classifier is constructed by putting all of these 181 networks in parallel, producing one large network with 181 outputs. For every $21 \times 21 \times 49$ patch of shearlet coefficients, this classifier generates a vector of length 181 indicating if the underlying function is smooth at the center point of the patch and listing all directions of edges present at the center point.

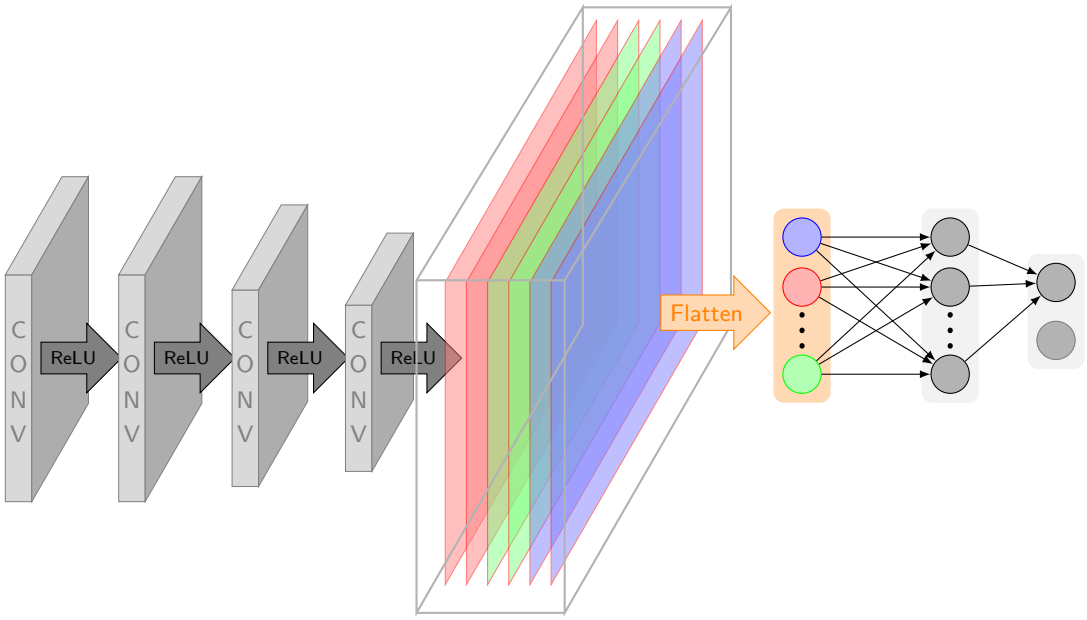

Figure 1: Illustration of the network architecture forming the foundation of the classifier. This network consists of four convolutional layers and one fully-connected layer. The colored block in the middle represents a stack of the output of the last convolutional layer. The colors correspond to the different channels.

5.3 Network training

We train the network as described in Subsection 5.2 using stochastic gradient descent to minimize the cross-entropy over a variety of training sets. We use four different data sets to train our classifier and test our

algorithm:

1) The first data set consists of patches of the shearlet transform of images made of random sums of ellipses and parallelograms of different contrasts, sizes, and orientations.

2) The second data set is again based on random sums of ellipses and parallelograms, but convolves these images with a kernel to generate a function with a higher-order wavefront set.

3) The third data set is based on the BSDS500 (Berkeley Segmentation data set) provided by the Computer Vision Group of UC Berkeley. It comprises 503 natural images of different types.

4) The fourth data set is based on the Semantic Boundaries data set (SBD) with 11355 natural images, again provided by the Computer Vision Group of UC Berkeley.

We depict examples of functions from each of the data sets in Figures 2, 3, 4 and 5.

To make these data sets suitable for our purposes, we need to equip each image of the data sets with an associated set of labels indicating the associated wavefront set or the set of edges. For the first two data sets, standard theoretical results on the wavefront sets of characteristic functions allow us to compute the associated wavefront sets analytically. The segmentation and semantic boundaries data sets, on the other hand, are natural images where such an approach is not possible. These data sets are used to assess the quality of segmentation and contour detection applications, see [21] and [1]. Therefore, every image in these data sets was annotated and has a set of ground truth edges. However, we should point out that this annotated ground truth does not contain all edges of the images, but only that between semantically different parts of the images. We depict the annotated edges in Figures 2, 3, 4 and 5.

In the following subsections, we describe the computation of the associated wavefront sets in detail.

5.3.1 Ellipses and parallelograms

The wavefront sets of characteristic functions of ellipses and parallelograms can be identified by (2.1) and the fact that if x is a vertex of a parallelogram P then $\{x\} \times \mathbb{S}^1 \subset \mathrm{WF}(\chi_P)$. For sums of these functions, we have, by basic properties of the Fourier transform that

$$\mathrm{WF}(\chi_{P_1} + \chi_{P_2}) \subset \mathrm{WF}(\chi_{P_1}) \cup \mathrm{WF}(\chi_{P_2}).$$

Note that in this relation we do not have equality in general. Indeed, if $\mathrm{WF}(\chi_{P_1}) \cap \mathrm{WF}(\chi_{P_2}) \neq \varnothing$ then cancellations can occur. We shall neglect this technicality as the probability of cancellations is sufficiently small and assume that the wavefront set of characteristic functions as described above is the union of the respective wavefront sets.

We build this data set by randomly choosing a number of parallelograms and ellipses with random positions and computing the associated ground truth of the wavefront set as described above.

5.3.2 Higher-order wavefront data set

The ellipses/parallelograms data set contains images with jump singularities only. To test our method on functions with higher-order singularities, such as ramp singularities, we computed the convolution of the elements of the ellipses/parallelograms data set with a filter h the Fourier transform of which is given by:

$$\hat{h}(\xi) = \frac{1}{1 + |\xi|}, \text{ for } \xi \in \mathbb{R}^2.$$

It is not hard to see that $P : f \mapsto h * f$ is an elliptic pseudo-differential operator and hence $\mathrm{WF}(h*g) = WF(g)$ for all $g \in L^2(\mathbb{R}^2)$, see [16, Chapter 8 G] for details. Thus, the convolutions of the elements of the ellipses/parallelograms data set with h have the same wavefront set as the associated ellipses or parallelograms, but of a higher order.

5.3.3 Segmentation and semantic boundaries data sets

In the BSDS500 and the SBD data sets, the ground truth of the edges is given in form of binary images with 0's at positions where the image is smooth and 1's at locations associated to edges. This annotated edge set is depicted in Figure 4.

To compute the orientation of the edges, we used a five-point stencil derivative on the edges to approximate the normal vectors. To detect corners and assign the appropriate orientations we used the Harris corner detector [22]. From these images, we produce patches for the training of the network classifier. However, due to the fact that the annotated image does not contain all edges we only use patches that are close to these edges for training, validation and testing.

5.4 DeNSE: Deep Network Shearlet Edge Extractor

In this section, we present our algorithm extracting the wavefront set of a digital image. For $M \in \mathbb{N}$, and a digital image $I \in \mathbb{R}^{M \times M}$, this algorithm produces, for every $m^* \in [11, M-10]^2$ a prediction of the wavefront set of I at m^*. The algorithm proceeds along the following three steps:

Step 1 Train the network classifier as of Section 5.2 on a set of labeled training data.

Step 2 For a given test image $I \in \mathbb{R}^{M \times M}$, compute the digital shearlet transform of I with 49 shearlet generators: The digital shearlet transform of I is given by $(\mathrm{DSH}(I)(j,k,m,\iota))_{j \in J, k \in K_j, \iota \in \{-1,0,1\}, m \in [1,M]^2}$.

Step 3 For every $m^* = (m_1^*, m_2^*) \in [11, M-10]^2$, pass the patch

$$(\mathrm{DSH}(I)(j,k,m,\iota))_{j \in J, k \in K_j, \iota \in \{-1,0,1\}, m \in [m_1^*-10, m_1^*+10] \times [m_2^*-10, m_2^*+10]} \tag{5.2}$$

to the classifier of Step 1. If the classifier predicts that an edge with direction θ is present, then classify (m^*, θ) as an element of the wavefront set of I.

We coin the algorithm above Deep Network Shearlet Edge Extractor (DeNSE).

6 Numerical results

We implemented the training as described in the previous section using the GPU version of Tensorflow. To evaluate the classification quality, we use two quality measures, a mini-batch test average taken over all mini-batches and the so-called MF-score. The MF-score is computed as the mean of the F-score defined as

$$F := \frac{2PR}{R+P},$$

where P is the *precision*, i.e., the number of true positives divided by the sum of true and false positives, and R is the *recall*, i.e., the number of true positives divided by the sum of true positives and false negatives, [41]. The MF-score is often used for evaluating classification performance when the distribution of classes is uneven. This is, for example, the case in edge detection, since there usually are significantly fewer edge points than smooth points in an image. Moreover, these performance measures enable us to compare with the state-of-the-art [50] on the respective data sets.

6.1 Results for ellipses/parallelograms

We train each of the 181 subnetworks as of Subsection 5.2 using 10,000 images as training data, 1,000 images as validation data, and 2,000 images as test data. For each direction θ_i we trained the associated subnetworks using a mini-batch procedure with 86 examples per batch and 3,000 training steps for each. We obtained an average test accuracy of 96.2% (taking the average over all 181 classifiers) and an MF-score of 97.1%. We also notice that the test accuracy of the individual classifiers was higher when classifying angles aligned to the discrete orientations of the underlying shearlet system.

We compared our method on this data set to other classifiers commonly use in machine learning namely: Logistic regression, Decision trees, K-nearest neighbors, Linear SVM, and Random forest. We report the performances of these classifiers in Table 1.

Method	**Test accuracy**	**MF-score**
Logistic regression	45.7	48.9
Decision trees	75.2	75.8
Linear SVM	46.5	50.3
K-nearest neighbors	72.7	73.2
Random forest	86.0	86.7
DeNSE	**96.2**	**97.1**

Table 1: Ellipses/parallelograms data set performance metrics in percentage.

By construction, the last of the 181 subnetworks corresponds to an edge-detector, where the achieved average test accuracy was 97.5%, and the MF-score was 97.9%, the performance benchmarks with other classical edge classifiers can be found in Table 2. Figure 2 shows the results on an example of the ellipses/parallelograms data set.

We depict the classification for one instance of the test set of the parallelograms/ellipses data set in Figure 2 and compare the results with the classification by the heuristic approach by Yi-Labate-Easley-Krim [48]. We observe that our method performs significantly better in low contrast regions. Moreover, our algorithm appears to be more precise when differentiating between corners and edges. Here, we classify a point as a corner point if the classifiers predict at least two different orientations that differ by more than 10 degrees. In Figures 2, 3, 4 and 5, we indicate corners by white dots.

Method	**MF-score**
Canny [8]	49.1
Sobel [44]	40.0
BEL [10]	63.3
Yi-Labate-Easley-Krim [48]	70.3
CoShREM [39]	90.6
DeNSE	**97.5**

Table 2: Edge detection performances of edge detection algorithms on the Ellipses/parallelograms data set. The MF-Score is in percentage.

Method	**MF-score**
gPb-owt-ucm [1]	73.7
gPb [1]	71.5
Mean Shift [9]	64.0
Normalized Cuts [43]	64.2
Fetzenszwalb, Huttenlocher [15]	61.0
Canny	60.3
CoShREM [39]	75.7
DeepEdge [2]	75.3
DeNSE	**95.4**

Table 3: BSDS500 (Berkeley) data set performance metrics in percentage.

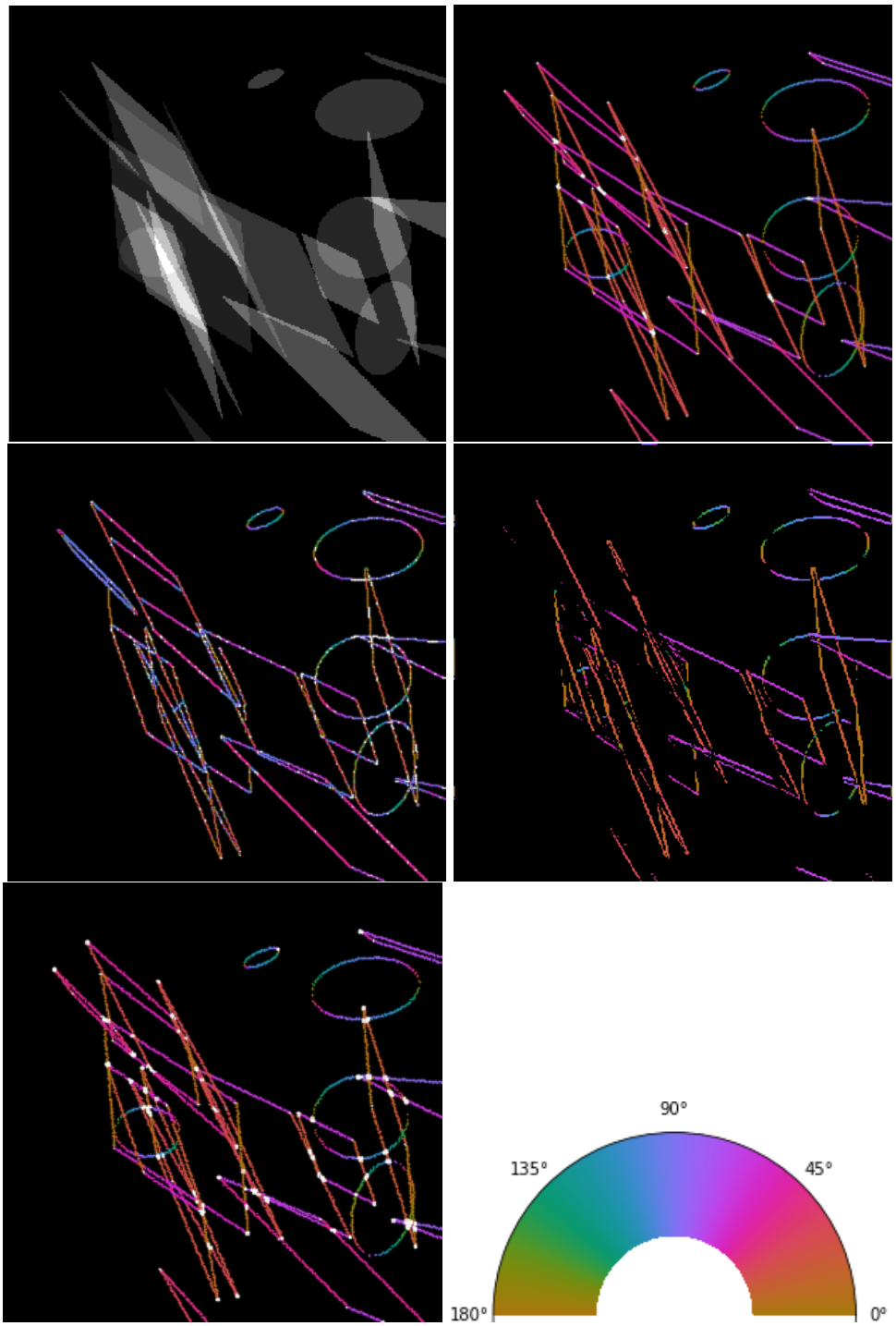

Figure 2: Computed edges and orientations of an example of the ellipses/parallelograms data set. Top-left: Input image. Top-right: Orientations, human annotation. Middle-left: Orientations predicted by Yi-Labate-Easley-Krim algorithm. Middle-right: Orientations predicted by CoShREM. Bottom-left: Orientations predicted by DeNSE algorithm. Bottom-right: Color code for normal-directions.

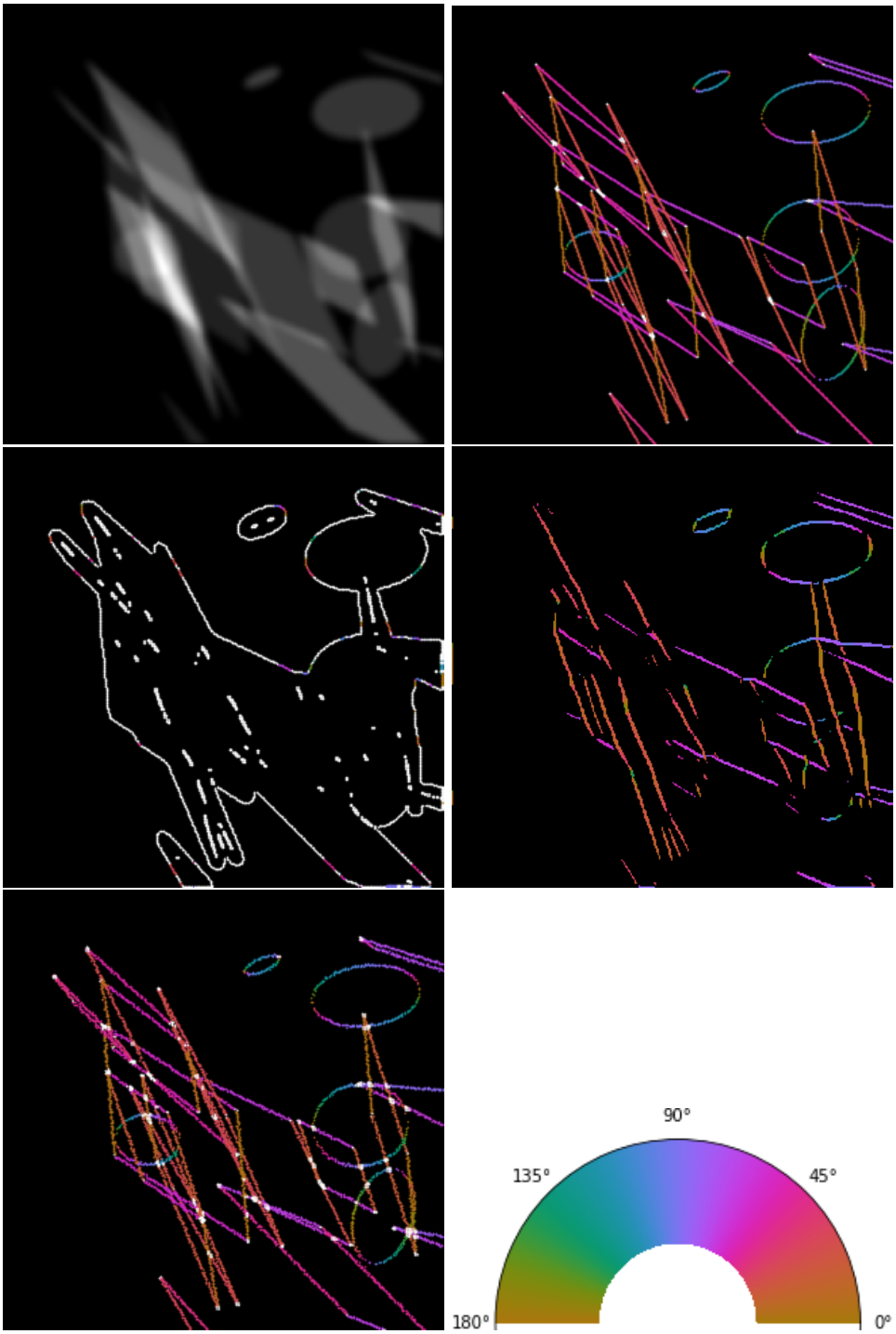

Figure 3: Computed edges and orientations of an example of the higher-order ellipses/parallelograms data set. Top-left: Input image. Top-right: Orientations, human annotation. Middle-left: Orientations predicted by Yi-Labate-Easley-Krim algorithm. Middle-right: Orientations predicted by CoShREM. Bottom-left: Orientations predicted by DeNSE algorithm. Bottom-right: Color code for normal-directions.

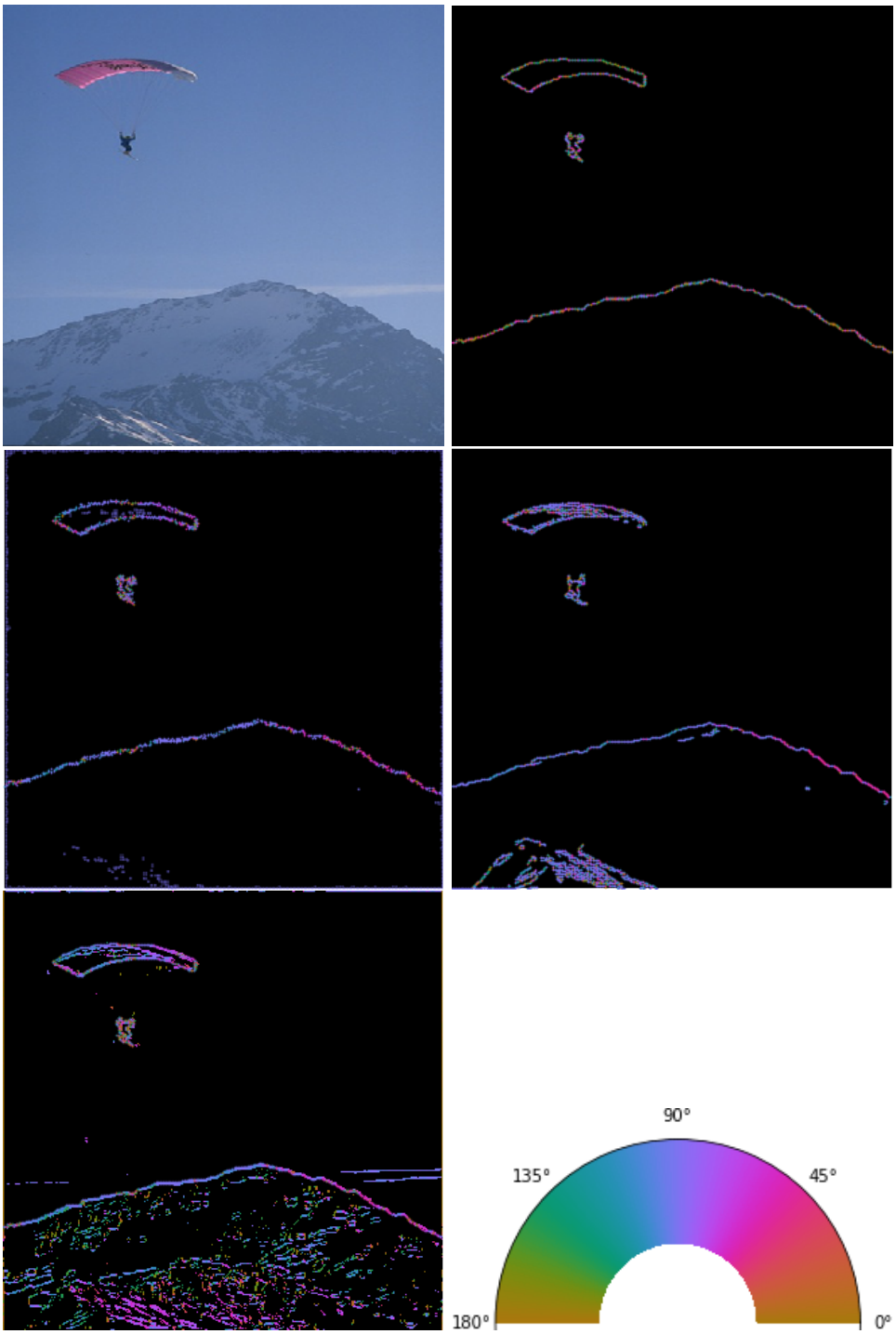

Figure 4: Computed edges and orientations of an example of the BSDS500 (Berkeley) data set. Top-left: Input image. Top-right: Orientations, human annotation. Middle-left: Orientations predicted by gPb-owt-ucm. Middle-right: Orientations predicted by CoShREM. Bottom-left: Orientations predicted by DeNSE algorithm. Bottom-right: Color code for normal-directions.

6.2 Results for higher-order wavefront set data set

Using the same procedure as in the ellipses/parallelograms classification, we performed the edge detection, and orientation classification of the higher-order wavefront set data set. In this case, we used 30,000 patches as training data, 3,000 patches as validation data, and 6,000 patches as test data. We trained on 86-sized mini-batches, with 200,000 training steps. We obtained an average test accuracy of 93.4% and an MF-score of 94.6%. We are not aware of any algorithms specifically build for higher-order wavefront set detection, which is why we do not provide a comprehensive list of results of alternative algorithms in this case.

For completeness, we added Figure 4 showing an example of the obtained results. We also add two predictions by the algorithm of Yi-Labate-Easley-Krim [48] and the method CoShREM [39].

It is important to mention that the algorithm of Yi-Labate-Easley-Krim is constructed to detect jump singularities and not ramp like singularities. Hence this algorithm is expected to fail on this data set. Indeed, the performance of the algorithm of achieves only an MF-score of 30.5%. CoShREM, on the other hand, is built to detect edges and ridges. The performance was significantly better than that of Yi-Labate-Easley-Krim and resulted in an MF-score of 65.4%.

6.3 Results for Berkeley segmentation set

In the Berkeley segmentation data set, the complexity of the images is considerably higher compared to the images from the ellipses/parallelogram data set. Therefore, we use a significantly larger training set to train the associated classifier. For the classification of each angle, we used 30,000 patches as training data (around 600 patches per image), 3,000 patches as validation data, and 6,000 patches as test data. As in the case of the ellipses/parallelograms, we train using a mini-batch procedure, with 86 examples per batch, but in this case, using 30,000 training steps for each. We obtained an average test accuracy of 93.1% and MF-score of 95.4%, which is lower than the one obtained in the ellipses/parallelogram due to the higher complexity of the patches. One advantage of this and the SBD data set is the existence of several benchmarks including state-of-the-art deep learning based algorithms.

We compared our method using the available benchmarks on this data set provided by the UC Berkeley Computer Vision Group, we refer to [1] for a more detailed explanation of these methods. In [1], just the MF-score of the competing algorithms was reported. We give the results in Table 3.

We present one example of the results obtained on the BSDS500 data set in Figure 4.

6.4 Results for semantic boundary set (SBD)

The SBD data set contains significantly more images than the BSDS500 which, as we will observe below, improves the overall classification performance slightly. In this case, we used 100,000 patches as training data, 10,000 patches as validation data, and 20,000 patches as test data. We train on 86-sized mini-batches, with 100,000 training steps. We obtained average test accuracy of 95.3% and MF-score of 96.8%.

This data set has recently been widely used for image segmentation tasks, in particular, it was used on the two deep learning based image segmentation frameworks proposed by Z. Yu *et al.*, namely the SEAL (Simultaneous Edge Alignment and Learning) [50] and the CASENet (Category-Aware Semantic Edge Detection Network) [49]. We also compared them with the deep learning image boundary detector and classifier proposed (OBDC) by J. Y. Koh *et al.* [26]. The results can be found on Table 4.

Figure 5 shows the results obtained by DeNSE on an example image of the SBD data set, as in the case of the BSDS500 data set, the obtained result admits more edges than the ground truth due to the batch-based approach. Nonetheless, the method outperforms even the specialized algorithms for segmentation over the given data sets.

6.5 Conclusion

We observe that in all performed tests, our novel algorithm DeNSE significantly outperforms all competitors. In doing so, DeNSE outperforms not only traditional methods, but also other, Deep Learning based algorithms.

Method	MF-score
OBDC	62.5
CASENet	71.8
CASENet-S	75.8
CASENet-C	80.4
CoShREM	69.7
SEAL	81.1
DeNSE	**96.8**

Table 4: Performance on the SBD data set. All values are in percentage.

Comparing the complexity of the involved neural networks reveals that the classifier of DeNSE uses a comparably small neural network.

One natural explanation for the jump in performance already with simple networks is the fact that the shearlet representation transforms the data in a much more convenient form for training and evaluation purposes at least from the point of view of wavefront set extraction. The underlying theoretically established relationship between edges and their directions and the shearlet transform also indicates that our algorithm can be easily generalized to extract not only jump-like singularities but also ramps or even higher-order non-smooth patterns in images.

As mentioned in the introduction, the presented method offers a new analysis tool for any application where information on the wavefront set can be used advantageously. This includes, in particular, inverse problems, where, through mathematical analysis, a relation between the wavefront set of a function f and the wavefront set of its transform $T(f)$ is apriori known. Using the prediction of DeNSE and the known relationship of the wavefront sets given by the underlying operator, it is possible to compute the wavefront set of the f without inverting T. This additional information can then be used as a regularization for the inverse problem.

Acknowledgements

H.A.-L. is supported by the Berlin Mathematical School. G. K. acknowledges partial support by the Bundesministerium für Bildung und Forschung (BMBF) through the Berliner Zentrum for Machine Learning (BZML), Project AP4, by the Deutsche Forschungsgemeinschaft (DFG) through grants CRC 1114 "Scaling Cascades in Complex Systems", Project B07, CRC/TR 109 "Discretization in Geometry and Dynamics', Projects C02 and C03, RTG DAEDALUS (RTG 2433), Projects P1 and P3, RTG BIOQIC (RTG 2260), Projects P4 and P9, and SPP 1798 "Compressed Sensing in Information Processing", Coordination Project and Project Massive MIMO-I/II, by the Berlin Mathematics Research Center MATH+ , Projects EF1-1 and EF1-4, and by the Einstein Foundation Berlin. The work of O.Ö. was supported by the Swedish Foundation of Strategic Research grant AM13-004. P.P is supported by a DFG Research Fellowship "Shearlet-based energy functionals for anisotropic phase-field methods".

References

[1] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. Contour detection and hierarchical image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(5):898–916, 2011.

[2] G. Bertasius, J. Shi, and L. Torresani. Deepedge: A multi-scale bifurcated deep network for top-down contour detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4380–4389, 2015.

[3] T. O. Binford. Inferring surfaces from images. *Artif. Intell.*, 17(1):205–244, 1981.

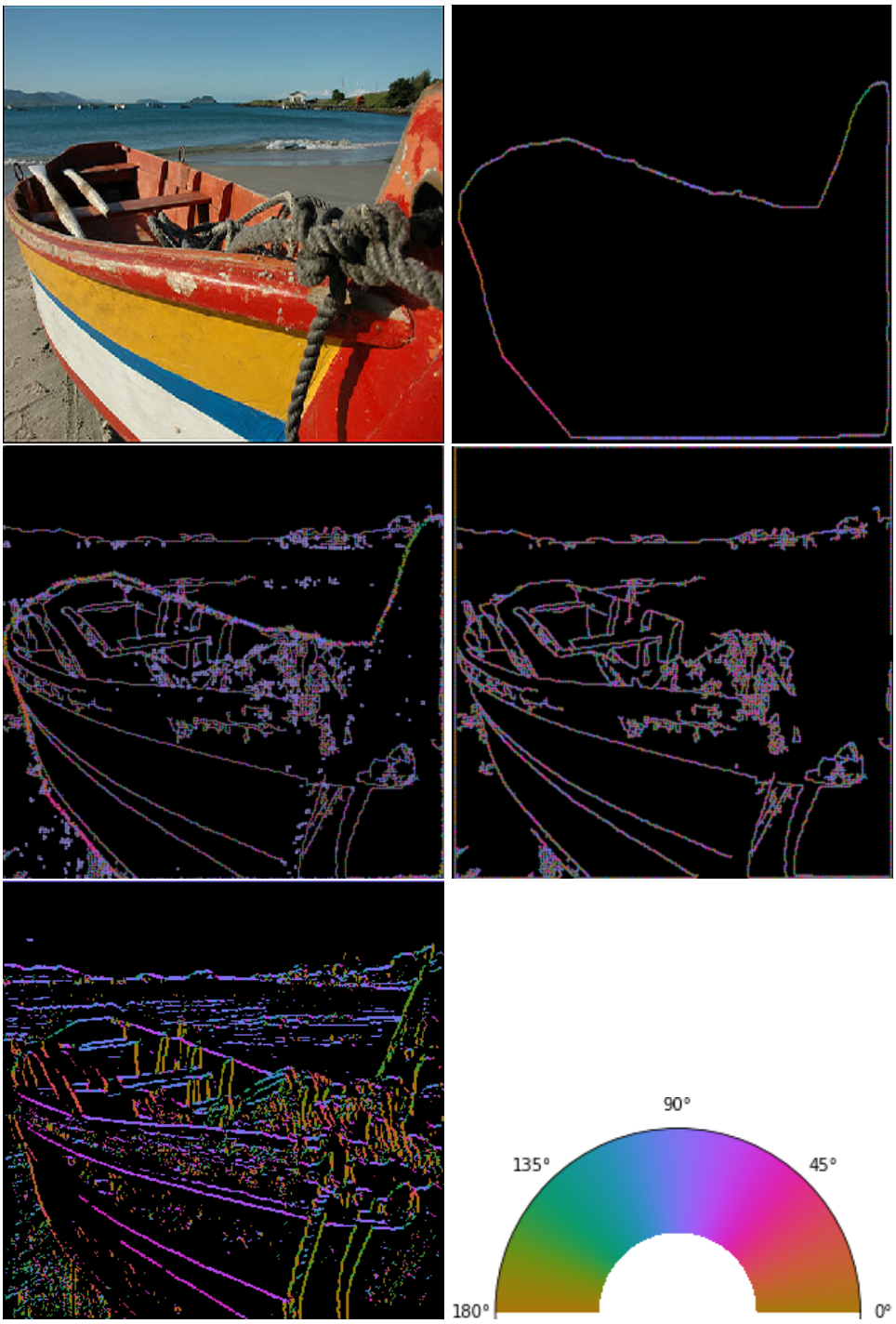

Figure 5: Computed edges and orientations of an example of the SBD data set.Top-left: Input image. Top-right: Orientations, human annotation. Middle-left: Orientations predicted by SEAL. Middle-right: Orientations predicted by CoShREM. Bottom-left: Orientations predicted by DeNSE algorithm. Bottom-right: Color code for normal-directions.

[4] M. Brady. Computational approaches to image understanding. *ACM Comput. Surv.*, 14(1):3–71, 1982.

[5] T. A. Bubba, G. Kutyniok, M. Lassas, M. März, W. Samek, S. Siltanen, and V. Srinivasan. Learning the invisible: A hybrid deep learning-shearlet framework for limited angle computed tomography. *arXiv:1811.04602*, 2018.

[6] E. Candes, L. Demanet, and L. Ying. Fast computation of Fourier integral operators. *SIAM J. Sci. Comput.*, 29(6):2464–2493, 2007.

[7] E. J. Candès and D. L. Donoho. Continuous curvelet transform: I. resolution of the wavefront set. *Appl. Comput. Harmon. Anal.*, 19(2):162–197, 2005.

[8] J. Canny. A computational approach to edge detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 8(6):679–698, 1986.

[9] D. Comaniciu and P. Meer. Mean shift: a robust approach toward feature space analysis. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2002.

[10] P. Dollár, Z. Tu, and S. Belongie. Supervised learning of edges and object boundaries. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2006.

[11] R. O. Duda and P. E. Hart. Pattern classification and scene analysis. *A Wiley-Interscience Publication, New York: Wiley, 1973*, 1973.

[12] M. E. Davison. The ill-conditioned nature of the limited angle tomography problem. *SIAM J. Appl. Math.*, 43, 04 1983.

[13] V. Faber, A. I. Katsevich, and A. G. Ramm. Inversion of cone-beam data and helical tomography. *J. Inverse Ill-Posed Probl.*, 3:429–446, 01 1995.

[14] J. Fell, H. Führ, and F. Voigtlaender. Resolution of the wavefront set using general continuous wavelet transforms. *J. Fourier Anal. Appl.*, 22(5):997–1058, 2016.

[15] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient graph-based image segmentation. *Int. J. Comput. Vis.*, 2004.

[16] G. B. Folland. *Introduction to partial differential equations.* Princeton University Press, Princeton, NJ, 1995.

[17] P. Grohs. Continuous shearlet frames and resolution of the wavefront set. *Monatsh. Math.*, 164(4):393–426, 2011.

[18] P. Grohs and Z. Kereta. Continuous parabolic molecules. *Research Report*, 2015.

[19] K. Guo, G. Kutyniok, and D. Labate. Sparse multidimensional representations using anisotropic dilation and shear operators. In *Wavelets and Splines*, pages 189–201, Nashville, TN, 2005. Nashboro Press,.

[20] K. Guo, D. Labate, and W.-Q. Lim. Edge analysis and identification using the continuous shearlet transform. *Appl. Comput. Harmon. Anal.*, 27(1):24–46, 2009.

[21] B. Hariharan, P. Arbelaez, L. Bourdev, S. Maji, and J. Malik. Semantic contours from inverse detectors. In *Proceedings of the 2011 International Conference on Computer Vision*, ICCV '11, pages 991–998, Washington, DC, USA, 2011. IEEE Computer Society.

[22] C. Harris and M. Stephens. A combined corner and edge detector. In *Fourth Alvey Vision Conference*, pages 147–151, 1988.

[23] L. Hörmander. Fourier integral operators I. *Acta Math.*, 127:79–183, 1971.

[24] L. Hormander. *The Analysis of Linear Partial Differential Operators. I, Distribution Theory and Fourier Analysis*. Grundlehren Der Mathematischen Wissenschaften. Springer, 1990.

[25] L. Jacques, L. Duval, C. Chaux, and G. Peyré. A panorama on multiscale geometric representations, intertwining spatial, directional and frequency selectivity. *Signal Proc.*, 91(12):2699–2730, 2011.

[26] J. Y. Koh, W. Samek, K.-R. Müller, and A. Binder. Object boundary detection and classification with image-level labels. In *German Conference on Pattern Recognition*, pages 153–164. Springer, 2017.

[27] V. P. Krishnan and E. T. Quinto. *Microlocal analysis in tomography*. Handbook of Mathematical Methods in Imaging. Springer, 2015.

[28] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 1097–1105, 2012.

[29] G. Kutyniok and D. Labate. Resolution of the wavefront set using continuous shearlets. *Trans. Amer. Math. Soc.*, 361(5):2719–2754, 2009.

[30] G. Kutyniok, W.-Q. Lim, and R. Reisenhofer. ShearLab 3D: Faithful digital shearlet transforms based on compactly supported shearlets. *ACM Trans. Math. Softw.*, 42(1):5:1–5:42, 2016.

[31] G. Kutyniok, W.-Q. Lim, and X. Zhuang. Digital shearlet transforms. In *Shearlets*, pages 239–282. Springer, 2012.

[32] G. Kutyniok and P. Petersen. Classification of edges using compactly supported shearlets. *Appl. Comput. Harmon. Anal.*, 42(2):245–293, 2017.

[33] R. J. I. Marks. *Introduction to Shannon sampling and interpolation theory*. Springer Science & Business Media, 2012.

[34] D. Marr and E. Hildreth. Theory of edge detection. *Proc. R. Soc. Lond. B*, 207(1167):187–217, 1980.

[35] P. Perona and J. Malik. Detecting and localizing edges composed of steps, peaks and roofs. In *Proceedings of the Third International Conference on Computer Vision*, pages 52–57. IEEE, 1990.

[36] J. M. Prewitt. Object enhancement and extraction. *Pict. Process. Psychopictorics*, 10(1):15–19, 1970.

[37] E. T. Quinto. Singularities of the X-ray transform and limited data tomography in $\mathbb{R}^2$ and $\mathbb{R}^3$. *SIAM J. Math. Anal.*, 24(5):1215–1225, 1993.

[38] E. T. Quinto and O. Öktem. Local tomography in electron microscopy. *SIAM J. Appl. Math.*, 68(5):1282–1303, 2008.

[39] R. Reisenhofer, J. Kiefer, and E. J. King. Shearlet-based detection of flame fronts. *Exp. Fluids*, 57, 11 2015.

[40] L. G. Roberts. *Machine perception of three-dimensional solids*. PhD thesis, Massachusetts Institute of Technology, 1963.

[41] Y. Sasaki. The truth of the F-measure. *Teach Tutor mater*, 1(5):1–5, 2007.

[42] M. Sato. Regularity of hyperfunctions solutions of partial differential equations. In *Actes du Congrès international des mathématiciens*, volume 2, pages 785–794, Paris, 1971. Gauthier-Villars.

[43] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Trans. on Pattern Anal. Mach. Intell.*, 1997.

[44] I. Sobel. An isotropic 3x3 image gradient operator. *Presentation at Stanford A.I. Project 1968*, 2014.

[45] M. E. Taylor. *Pseudodifferential operators*. Princeton University Press, Princeton, NJ, 1981.

[46] V. Torre and T. A. Poggio. On edge detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 8(2):147–163, 1986.

[47] G. Uhlmann and A. Vasy. The inverse problem for the local geodesic X-ray transform. *Invent. Math.*, 205, 10 2012.

[48] S. Yi, D. Labate, G. R. Easley, and H. Krim. A shearlet approach to edge analysis and detection. *IEEE Trans. Image Process.*, 18(5):929–941, 2009.

[49] Z. Yu, C. Feng, M.-Y. Liu, and S. Ramalingam. Casenet: Deep category-aware semantic edge detection. In *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA*, pages 21–26, 2017.

[50] Z. Yu, W. Liu, Y. Zou, C. Feng, S. Ramalingam, B. V. Kumar, and J. Kautz. Simultaneous edge alignment and learning. *arXiv preprint arXiv:1808.01992*, 2018.