Journal of Mathematical Imaging and Vision (2020) 62:365–375
https://doi.org/10.1007/s10851-019-00890-3

Residual Networks as Flows of Diffeomorphisms

François Rousseau[1] · Lucas Drumetz[2] · Ronan Fablet[2]

Received: 30 November 2018 / Accepted: 21 May 2019 / Published online: 29 May 2019

Abstract

This paper addresses the understanding and characterization of residual networks (ResNet), which are among the state-of-the-art deep learning architectures for a variety of supervised learning problems. We focus on the mapping component of ResNets, which map the embedding space toward a new unknown space where the prediction or classification can be stated according to linear criteria. We show that this mapping component can be regarded as the numerical implementation of continuous flows of diffeomorphisms governed by ordinary differential equations. In particular, ResNets with shared weights are fully characterized as numerical approximation of exponential diffeomorphic operators. We stress both theoretically and numerically the relevance of the enforcement of diffeomorphic properties and the importance of numerical issues to make consistent the continuous formulation and the discretized ResNet implementation. We further discuss the resulting theoretical and computational insights into ResNet architectures.

Keywords Residual network · Diffeomorphism · Dynamical systems

1 Introduction

Deep learning models are the reference models for a wide range of machine learning problems. Among deep learning (DL) architectures, residual networks (also called ResNets) have become state-of-the-art ones [15,16]. Experimental evidences emphasize critical aspects in the specification of these architectures, for instance, in terms of network depths or combination of elementary layers as well as in their stability and genericity. The understanding and the characterization of ResNets and more wider DL architectures from a theoretical point of view remain a key issue despite recent advances for CNN [24].

Interesting insights into ResNets have recently been presented in [12,25,31] from an ordinary/partial differential equation (ODE/PDE) point of view. ResNets can be regarded as numerical schemes of differential equations. In particular, in [25], this PDE-driven setting stresses the importance of numerical stability issues depending on the selected ResNet configuration. Interestingly, it makes explicit the interpretation of the ResNet architecture as a depth-related evolution of an input space toward a new space, where the prediction of the expected output (for instance classes) is solved according to a linear operator. This interpretation is also pointed out in [13] and discussed in terms of Riemannian geometry.

In this work, we deepen this analogy between ResNets and deformation flows to relate ResNet and registration problems [27], especially diffeomorphic registration [2,3,5,30]. Our contribution is threefold: (1) We restate ResNet learning as the learning of a continuous and integral diffeomorphic operator and investigate different solutions, especially the exponential operator of velocity fields [2], to enforce diffeomorphic properties; (2) we make explicit the interpretation of ResNets as numerical approximations of the underlying continuous diffeomorphic setting governed by ordinary differential equations (ODE); (3) we provide theoretical and computational insights into the specification of ResNets and on their properties.

This paper is organized as follows. Section 2 relates ResNets to diffeomorphic registrations. We introduce in Sect. 3 the proposed diffeomorphism-based learning framework. Section 4 reports experiments. Our key contributions are further discussed in Sect. 5.

2 From ResNets to Diffeomorphic Registrations

ResNets [15,16] have become state-of-the-art deep learning architectures for a variety of problems, including, for

✉ François Rousseau
francois.rousseau@imt-atlantique.fr

[1] IMT Atlantique, LaTIM, UMR INSERM 1101, 29238 Brest, France

[2] IMT Atlantique, Lab-STICC, UMR CNRS 6285, 29238 Brest, France

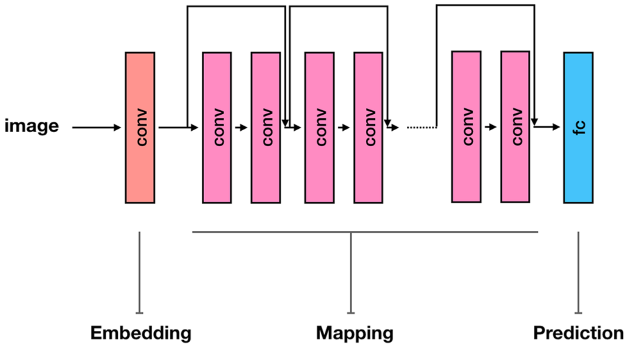

Fig. 1 A schematic view of ResNet architecture [15], decomposed into three blocks: embedding, mapping and prediction. ‘conv’ means convolution operations followed by nonlinear activations, and ‘fc’ means fully connected layer

instance, image recognition [15] or super-resolution [18]. This architecture has been proposed in order to explore performance of very deep models, without training degradation accuracy when adding layers. ResNets proved to be easier to optimize and made it possible to learn very deep models (up to hundreds layers).

As illustrated in Fig. 1, ResNets can be decomposed into three main building blocks:

- the embedding block which aims to extract relevant features from the input variables for the targeted task (such as classification or regression). In [15], the block consists of a set of 64 convolution filters of size 7 × 7 followed by nonlinear activation function such as ReLU.
- the mapping block, which aims to incrementally map the embedding space to a new unknown space, in which the data are, for instance, linearly separable in the classification case. In [15], this block consists of a series of residual units. A residual unit is defined as $\mathbf{y} = F(\mathbf{x}, \{W_i\}) + \mathbf{x}$, where the function F is the residual mapping to be learned. In [15], $F(\mathbf{x}) = W_2\sigma(W_1\mathbf{x})$, where σ denotes the activation function (biases are omitted for simplifying notations). The operation $F(\mathbf{x}) + \mathbf{x}$ is performed by a shortcut connection and element-wise addition.
- the prediction block, which addresses the classification or regression steps from the mapped space to the output space. This prediction block is expected to involve linear models. In [15], this step is performed with a fully connected layer.

In this work, we focus on the definition and characterization of the mapping block in ResNets. The central idea of ResNets is to learn the additive residual function F such that the layers in the mapping block are related by the following equation:

$$\mathbf{x}_{l+1} = \mathbf{x}_l + F(\mathbf{x}_l, W_l) \tag{1}$$

where $\mathbf{x}_l$ is the input feature to the lth residual unit. W_l is a set of weights (and biases) associated with the lth residual unit. In [16], it appears that such formulation exhibits interesting backward propagation properties. More specifically, it implies that the gradient of a layer does not vanish even when the weights are arbitrarily small.

Here, we relate the incremental mapping defined by these ResNets to diffeomorphic registration models [27]. These registration models, especially large deformation diffeomorphic metric mapping (LDDMM) [5,30], tackle the registration issue from the composition of a series of incremental diffeomorphic mappings, each individual mapping being close to the identity. Conversely, in ResNet architectures, the lth residual block provides an update of the form $\mathbf{x}_l + F(\mathbf{x}_l, W_l)$. Under the assumption that $\|F(\mathbf{x}_l, W_l)\| \ll \|\mathbf{x}_l\|$, the deformation flows generated by ResNet architectures may be expected to implement the composition of a series of incremental diffeomorphic mappings.

In [15,16], it is mentioned that the form of the residual function F is flexible. Several residual blocks have been proposed and experimentally evaluated such as bottleneck blocks [15], various shortcut connections [16] or aggregated residual transformations [32]. However, by making the connection between ResNets and diffeomorphic mappings, it appears here that the function F is a parametrization of an elementary deformation flow, constraining the space of admissible residual unit architectures.

We argue this registration-based interpretation motivates the definition of ResNet architectures as the numerical implementation of continuous flows of diffeomorphisms. Section 3 details the proposed diffeomorphism-based learning framework, in which diffeomorphic flows are governed by ODEs as in the LDDMM setting. ResNets with shared weights relate to a particularly interesting case yielding the definition of exponential diffeomorphism subgroups in the underlying Lie algebra. Overall, the proposed framework results in: (1) a theoretical characterization of the mapping block as an integral diffeomorphic operator governed by an ODE, (2) in considering deformation flows and Jacobian maps for the analysis of ResNets, (3) the derivation of ResNet architectures with additional diffeomorphic constraints.

3 Diffeomorphism-Based Learning

3.1 Diffeomorphisms and Driving Velocity Vector Fields

Registration issues have been widely stated as the estimation of diffeomorphic transformations between input and output spaces, especially in medical imaging [27]. Diffeomorphic properties guarantee the invertibility of the transformations, which includes the conservation of topological features. The parametrization of diffeomorphic transformations according to time-varying velocity vector fields has been shown to be very effective in medical imaging [21]. Beyond its computational performance, this framework embeds the group structure of diffeomorphisms and results in flows of diffeomorphisms governed by an ordinary differential equation (ODE):

$$\frac{\mathrm{d}\phi(t)}{\mathrm{d}t} = V_t\left(\phi(t)\right) \tag{2}$$

with $\phi(t)$ the diffeomorphism at time t, and V_t the velocity vector field at time t. $\phi(0)$ is the identity and $\phi(1)$ the registration transformation between embedding space $\mathcal{X}$ and output space $\mathcal{X}^*$, such that for any element X in $\mathcal{X}$ its mapped version in $\mathcal{X}^*$ is $\phi(1)(X)$. Given velocity fields $(V_t)_t$, the computation of $\phi(1)(X)$ comes from the numerical integration of the above ODE.

A specific class of diffeomorphisms refers to stationary velocity fields, that is to say velocity fields which do not depend on time ($V_t = V, \forall t$). As introduced in [2], in this case, the resulting diffeomorphisms define a subgroup structure in the underlying Lie group and yield the definition of the exponential operators. We here only briefly detail these key properties. We let the reader refer to [1] for a detailed and more formal presentation of their mathematical derivation. For a stationary velocity field, the resulting diffeomorphisms belong to the one-parameter subgroup of diffeomorphisms with infinitesimal generator V. In particular, they verify the following property: $\forall s, t, \phi(t) \cdot \phi(s) = \phi(s+t)$, where $\cdot$ stands for the composition operator in the underlying Lie group. This implies, for instance, that computing $\phi(1)$ boils down to applying n times $\phi(1/2^n)$ for any integer value n. Interestingly, this one-parameter subgroup yields the definition of diffeomorphisms $(\phi(t))_t$ as exponentials of velocity field V denoted by $(\exp(tV))_t$ and governed by the stationary ODE

$$\frac{\mathrm{d}\phi(t)}{\mathrm{d}t} = V\left(\phi(t)\right) \tag{3}$$

Conversely, any one-parameter subgroup of diffeomorphisms is governed by an ODE with a stationary velocity field. It may be noted that the above definition of exponentials of velocity fields generalizes the definition of exponential operators for matrices and finite-dimensional spaces.

3.2 Diffeomorphism-Based Supervised Learning

In this section, we view supervised learning issues as the learning of diffeomorphisms according to some predefined loss function. Let us consider a typical supervised classification issue, in which the goal is to predict a class Y from an N-dimensional real-valued observation X. Let $\mathcal{L}_\theta$ be a linear classifier model with parameter θ. Within a neural-network setting, $\mathcal{L}_\theta$ typically refers to a fully connected layer with softmax activations and parameter vector θ to the weight and bias parameters of this layer. Let $\mathcal{D}$ be the group of diffeomorphisms in $\mathbb{R}^N$. We state the supervised learning as the joint estimation of an embedding $\mathcal{E}$, a diffeomorphic mapping $\phi \in \mathcal{D}$ and linear classification model $\mathcal{L}_\theta$ according to:

$$\widehat{\mathcal{E}}, \widehat{\phi}, \widehat{\theta} = \arg\min_{\mathcal{E},\phi,\theta} \operatorname{loss}\left(\{\mathcal{L}_\theta\left(\phi\left(\mathcal{E}(X_i)\right)\right), Y_i\}_i\right) \tag{4}$$

with $\{X_i, Y_i\}_i$ the considered training dataset and $loss$ an appropriate loss function, typically a cross-entropy criterion. Considering the ODE-based parametrization of diffeomorphisms, the above minimization leads to an equivalent estimation of velocity field sequence (V_t)

$$\widehat{\mathcal{E}}, \widehat{(V_t)}, \widehat{\theta} = \arg\min_{\mathcal{E},(V_t),\theta} \operatorname{loss}\left(\{\mathcal{L}_\theta\left(\phi(1)\left(\mathcal{E}(X_i)\right)\right), Y_i\}_i\right) \tag{5}$$

$$\text{subject to} \begin{cases} \dfrac{\mathrm{d}\phi(t)}{\mathrm{d}t} & = V_t\left(\phi(t)\right) \\ \phi(0) & = I \end{cases} \tag{6}$$

When considering stationary velocity fields [2,3], this minimization simplifies as

$$\widehat{\mathcal{E}}, \widehat{V}, \widehat{\theta} = \arg\min_{\mathcal{E},(V),\theta} \operatorname{loss}\left(\{\mathcal{L}_\theta\left(\exp(V)\left(\mathcal{E}(X_i)\right)\right), Y_i\}_i\right) \tag{7}$$

We may point out that this formulation differs from the image registration problem in the considered loss function. Whereas image registration usually involves the minimization of the prediction error $Y_i - \phi(1)\left(\mathcal{E}(X_i)\right)$ with any pair $X_i, Y_i \in \mathbb{R}^N$, we here state the inference of the registration operator $\phi(1)$ according to classification-based loss function. It may also be noted that the extension to other loss functions (e.g., for regression tasks) is straightforward.

3.3 Derived NN Architecture

To solve for minimization issues (5) and (7), additional priors on the velocity fields can be considered. One may consider the introduction of an additional term in the minimization, which typically involves the integral of the norm of the gradient of the velocity fields and favors small registration displacements between the two time steps [5,33]. Parametric priors may also be considered. They come to set some parametrization for the velocity fields. In image registration studies, spline-based parametrization has, for instance, been explored [3].

Here, we combine these two types of priors. We exploit a parametric approach and consider neural-network-based representations of the driving velocity fields in ODEs (2) and (3).

More specifically, the discrete parametrization of the velocity field, $V_t(\mathbf{x})$, can be considered as a linear combination of basis functions:

$$V_{j,t}(\mathbf{x}) = \sum_i \nu_{t,j,i} f_{t,i}(\mathbf{x}) \tag{8}$$

where $V_{j,t}$ denotes component j (a scalar) of the learned velocity field, and the $\nu_{t,j,i}$ are the weights learned by the 1D convolutional layer.

Various types of shortcut connections and various usages of activation functions experimented in [16] correspond to various forms of the parametrization of the velocity field. Understanding residual units in a registration-based framework allows to provide a methodological guide to propose new valid residual units. Figure 2 shows three residual block units: original ResNet [15], improved ResNet [16] and the residual block studied in this work. For instance, it can be noticed that adding an activation function such as ReLU after the shortcut connection (i.e., after the addition layer) as in [15] (see Fig. 2a) makes the mapping no longer bijective, and thus, such an architecture may be less efficient, as shown experimentally in [16].

One way to build diffeomorphisms is to compose small perturbations of the identity. Using the same notations as in [33], let $\Omega \subset \mathbb{R}^d$ be open and bounded. We denote by $C_0^1(\Omega, \mathbb{R}^d)$ the Banach space of continuously differentiable vector fields v on Ω such that v and Dv vanish on $\partial\Omega$ and at infinity. Let $\chi_1^1(T, \Omega)$ be the set of absolutely integrable functions from $[0, T]$ to $C_0^1(\Omega, \mathbb{R}^d)$. It can be shown that the flow associated with the time-dependent vector field $v \in \chi_1^1(T, \Omega)$ is a diffeomorphism of Ω [33].

In this work, we propose a residual block suitable to build flows of diffeomorphisms. In the two case studies considered in this paper, two parametrizations of the basis functions are considered. For the case of the experiments of Sect. 4.1, on the CIFAR-10 dataset, where the inputs are images, the basis functions $f_{t,i}$ are parametrized with one convolutional layer and one ReLU layer. The linear combination of these basis functions can be represented as a second one-dimensional convolutional layer, with a filter size of 1×1. In the experiments of Sect. 4.2, on the spiral datasets, the inputs are two dimensional. In that case, the basis functions are modeled through the output of a dense layer, followed by a tanh activation function.

It has to be noticed that no biases are considered for the two convolutional layers. In order to control the magnitude of the velocity field, we propose to use in the residual block a tan$h$ layer and a scaling layer. Finally, to ensure that $v \in \chi_1^1(T, \Omega)$, we introduce a windowing layer such that v and Dv vanish on $\partial\Omega$ and at infinity. This ensures that v is a Lipschitz continuous mapping [33]. This proposed residual block is shown in Fig. 2c.

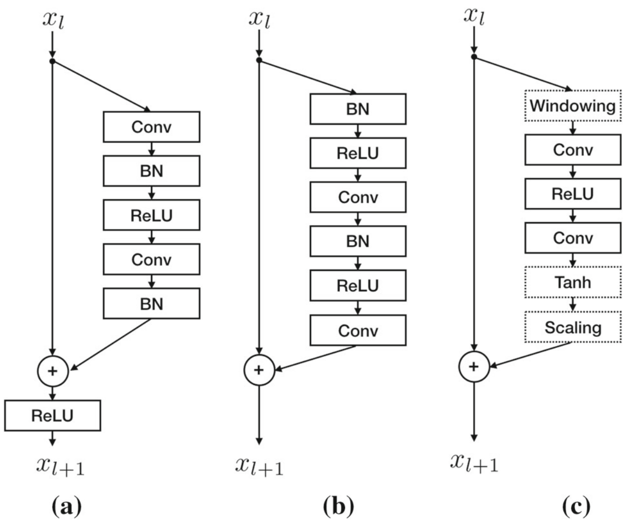

Fig. 2 Various residual units: **a** original ResNet [15], **b** improved ResNet [16], **c** proposed residual unit

In the registration-based framework considered so far, the transformation ϕ is only applied to the embedding of the observation X. This can introduce an undesirable asymmetry in the optimization process and have a significant impact on the registration performance. Inverse consistency, first introduced by Thirion [29], can be performed by adding a penalty term. In order to implement inverse consistent algorithms, it is useful to be able to integrate backward as well as forward. In the diffeomorphic framework, the inverse consistency can be written as follows:

$$\phi(1) \circ \phi(-1) = \phi(-1) \circ \phi(1) = \phi(0) \tag{9}$$

This inverse consistency can then be achieved by adding the following term in the overall loss function:

$$\begin{aligned} \widehat{\mathcal{E}}, \widehat{\phi}, \widehat{\theta} = \arg\min_{\mathcal{E},\phi,\theta} \alpha \operatorname{loss}\left(\{\mathcal{L}_\theta\left(\phi\left(\mathcal{E}(X_i)\right)\right), Y_i\}_i\right) \\ + (1-\alpha)\sum_i \left(\mathcal{E}(X_i) - \phi(-1)(\mathcal{E}(X_i)^*)\right)^2 \end{aligned} \tag{10}$$

where $\mathcal{E}(X_i)^* = \phi(1)(\mathcal{E}(X_i))$, $X_i \in \mathcal{X}$ and α is a weighting parameter. We may stress that this term does not depend on the targeted task (i.e., classification or regression) and only constraint the learning of the mapping block. Thus, this regularization term can be extended to data points that do not belong to the learning set, and more generally to points in a given domain, such that the inverse consistency property does not depend on the sampling of the learning dataset.

4 Experiments

In this section, we investigate experimentally the potential of the proposed architecture of residual blocks using the image classification dataset CIFAR-10 [22] and synthetic 2D data (spiral dataset). CIFAR-10 is used to explore the performance of the proposed residual unit with respect to other ResNet architectures. The 2D spiral dataset helps to further investigate properties of diffeomorphism-based networks and provides geometrical insights into the estimated flows.

4.1 CIFAR-10

4.1.1 Experimental Setting

The CIFAR-10 dataset contains 60,000 32×32 color images in ten different classes. A total of 10,000 images are used for the testing purpose. The overall architecture is decomposed into three main parts: embedding, mapping and prediction. First, the embedding is performed using the following layers: (1) a 5×5 convolutional layer with 128 filters, (2) batch normalization, (3) tan$h$ activation layer (which ensures that $\mathcal{E}(X_i) \in]-1,1[^p$, an open and bounded interval). Then, the network consists of several residual blocks as depicted in Fig. 2c (3×3 convolutions without bias, with 128 filters). The scaling factor of the residual units is learned and shared for every unit. At the end of the mapping block, a tan$h$ activation layer is used to ensure that $\phi(\mathcal{E}(X_i)) \in]-1,1[^p$, an open and bounded interval. Finally, the prediction step is performed using the following layers: (1) a 3×3 convolutional layer with 128 filters, (2) batch normalization, (3) tan$h$ activation layer, (4) 32×32 average pooling, (5) fully connected layer. Weights are initialized with a truncated normal distribution centered on 0 [14]. We use ℓ_2 weight-decay regularization set to 2×10^{-4} and SGD optimization method with a starting learning rate of 0.1, minibatch of 128, 100 epochs.

The goal of this experiment is to study the efficiency of the proposed residual unit with respect to original ResNet. The baseline architecture used for comparison is the ResNet architecture proposed in [16]. In this experiment, we use the Keras[1] implementation of ResNet for reproductibility purpose: ResNet56v2 (depth of 56 with increasing number of convolution filters [16]), with about 1.6M of trainable parameters.

4.1.2 Results

In this work, the dimension of the embedding space is constant throughout the mapping block. We first compare ResNet56v2 with the proposed approach with the same number of parameters, which corresponds to a network depth of 5 (i.e., 5 residual units). It can be seen in Fig. 3 that the performance of these two networks is similar when using the entire training dataset. Residual units of the Keras ResNet are built using three layers: convolution, batch normalization and ReLU. ReLU units are crucial to promote efficient sparse parametrizations of the velocity fields. However, the obtained results show that batch normalization in the residual units is not required to reach satisfactory classification accuracy. Instead, we use a tan$h$ activation layer to restrict the embedding set to be open and bounded for the whole flow of diffeomorphisms. It has to be noted that in the proposed network, the dimension of the transformed embedding space is constant. Similar to [17], the experimental results show that progressive dimension changes in the embedding space are not required, contrary to popular belief that the performance of deep neural networks is based on progressive removal of uninformative variability. To study the generalization performance of the proposed residual unit, we conduct experiments with decreasing number of training samples (see Table 1 for detailed results). It appears that the use of the proposed residual unit makes the model more robust to small training sets compared to the Keras ResNet.

We also investigate the impact of the network depth on the classification results. Increasing the depth leads to more complex models (in terms of trainable parameters). It can thus be expected to observe overfitting when using very small training datasets. However, Fig. 3 shows that increasing the depth of the proposed network does not lead to accuracy decrease. From a dynamical point of view, increasing the depth corresponds to smaller integration steps and then to smoother variations of velocity fields. The proposed residual architecture is not subject to overfitting for small datasets even with a deep diffeomorphic network.

4.2 Spiral Data

In this section, we propose to further deepen the understanding of behavior of networks based on flows of diffeomorphisms. Following the work on differential geometry analysis of ResNet architectures of Hauser et al. [13], we consider a classification task of two-dimensional spiral data.

4.2.1 Experimental Setting

No embedding layer is required in this experimental setup. The purpose of the mapping block is then to warp the input data points X_i into an unknown space $\mathcal{X}^*$, where the transformed data X^* are linearly separable. We have considered the following setting: The loss function is the binary cross-entropy between the output of a sigmoid function applied to the transformed data points X^* and the true labels. Each

[1] F. Chollet et al. Keras, 2015. https://keras.io.

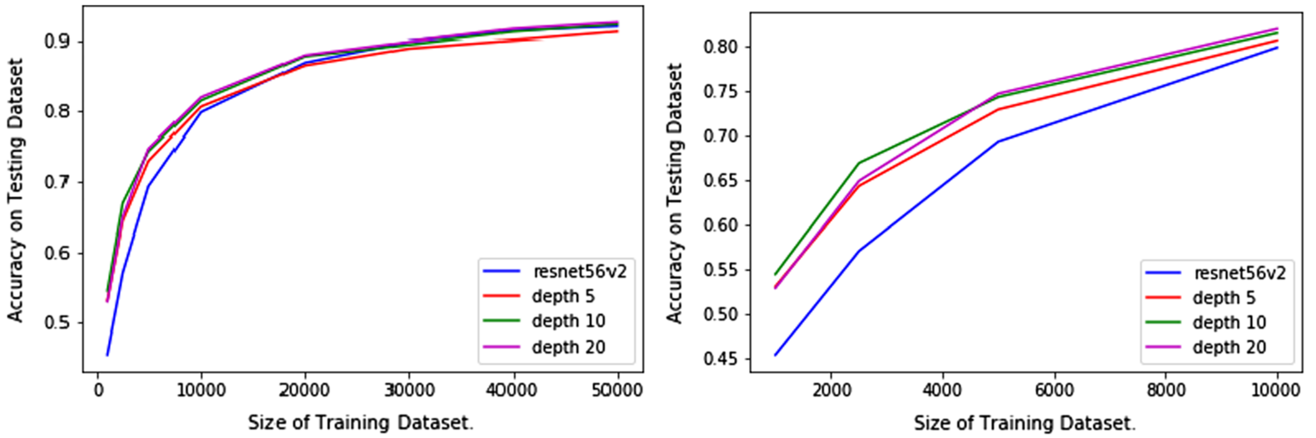

Fig. 3 Performance of various non-stationary ResNet architectures on CIFAR10, with varying sizes of the training dataset (left: from 1000 to 50,000 images, right: zoom version for very small dataset size)

Table 1 Accuracy on CIFAR10 for ResNet and proposed approaches (DiffeoNet: proposed residual unit, stationary velocity fields correspond to the use of shared weights) with respect to the number of training samples (1000 up to 50,000)

Methods	#params (M)	1k	2.5k	5k	10k	20k	30k	40k	50k
ResNet d56	1.6	0.45	0.57	0.69	0.80	0.87	0.90	0.91	0.92
DiffeoNet d5	1.6	0.53	0.64	0.73	0.81	0.86	0.89	0.90	0.91
DiffeoNet d10	3.1	0.54	0.67	0.74	0.81	0.88	0.89	0.91	0.92
DiffeoNet d20	6	0.53	0.65	0.75	0.82	0.88	0.90	0.92	0.93
Stationary velocity fields									
DiffeoNet d5	0.45	0.52	0.63	0.70	0.75	0.83	0.86	0.87	0.88
DiffeoNet d10	0.45	0.51	0.63	0.70	0.77	0.83	0.86	0.89	0.89
DiffeoNet d20	0.45	0.53	0.64	0.70	0.77	0.84	0.87	0.89	0.89
Stationary velocity fields and inverse consistency									
DiffeoNet d5	0.45	0.52	0.63	0.69	0.75	0.82	0.86	0.88	0.90
DiffeoNet d10	0.45	0.51	0.64	0.70	0.76	0.84	0.87	0.88	0.90
DiffeoNet d20	0.45	0.52	0.64	0.70	0.77	0.85	0.86	0.88	0.90

Depths (d5, d10, d20) and number of parameters are reported to perform a fair comparison

network is composed of 20 residual units, for which nonlinearities are modeled with tan$h$ activation functions and 10 basis functions (modeled by dense layers) are used for the parametrization of the velocity fields. Weights are initialized with the Glorot uniform initializer (also called Xavier uniform initializer) [9]. We use ℓ_2 weight-decay regularization set to 10^{-4} and the ADAM optimization method [20] with a learning rate of 0.001, $\beta_1 = 0.9$, $\beta_2 = 0.999$, minibatch of 300, 1000 epochs.

We consider four ResNet architectures: (a) a ResNet without shared weights (corresponding to time-varying velocity fields modeling), (b) ResNet with shared weights (corresponding to the stationary velocity fields modeling), (c) data-driven Symmetric ResNet with shared weights (considering also the inverse consistency criterion is computed over training data) and (d) domain-driven Symmetric ResNet with shared weights (where the inverse consistency criterion is computed over the entire domain using a random sampling scheme).

4.2.2 Characterization of ResNet Properties

ResNet architectures have been recently studied from the point of view of differential geometry in [13]. In this article, Hauser et al. studied the impact of residual-based approaches (compared to non-residual networks) in terms of differentiable coordinate transformations. In our work, we propose to go one step further by considering the characterization of the estimated deformation fields leading to an adapted configuration for the considered classification task. More specifically, we consider in this work the maps of Jacobian values.

The Jacobian (i.e., the determinant of the Jacobian matrices of the deformations) is defined in a two-dimensional space as follows:

$$J_\phi(\mathbf{x}) = \begin{vmatrix} \frac{\partial \phi_1(\mathbf{x})}{\partial x_1} & \frac{\partial \phi_1(\mathbf{x})}{\partial x_2} \\ \frac{\partial \phi_2(\mathbf{x})}{\partial x_1} & \frac{\partial \phi_2(\mathbf{x})}{\partial x_2} \end{vmatrix}. \qquad (11)$$

From a physical point of view, the value of the Jacobian represents the local volume variation induced by the transformation. A transformation with a Jacobian value equal to 1 is a transformation that preserves volumes. A Jacobian value greater than 1 corresponds to a local expansion, and a value less than 1 corresponds to a local contraction. The case where the Jacobian is zero means that several points are warped onto a single point: This case corresponds to the limit case from which the bijectivity of the transformation is not verified anymore, thus justifying the constraint on the positivity of the Jacobian in several registration methods [27].

4.2.3 Results

Classification algorithms are usually only evaluated using the classification accuracy (as the number of correct predictions from all predictions made). However, the classification rate is not enough to characterize the performance of a specific algorithm. In all the experiments shown in this work, the classification rate is greater than 99%. Visualization of the decision boundary is an alternative way to provide complementary insights into the regularity of the solution in the embedding space. Figure 4 shows the decision boundary for the four considered ResNets. Although all methods achieved very high classification rates, it can be seen that adding constraints such as the use of stationary velocity fields (i.e., shared weights) and inverse consistency constraints lead to smoother decision boundaries with no effect on the overall accuracy. This is regarded as critical for generalization and adversarial robustness [28].

Decision boundaries correspond to the projection of the estimated linear decision boundary in the space $\mathcal{X}^*$ into the embedding space $\mathcal{X}$. The visualization of decision boundaries does not however provide information regarding the topology of the manifold in the output space $\mathcal{X}^*$. We also study the deformation flow through the spatial configuration of data points through the network layers as in [13]. Figure 5 shows how each network untangles the spiral data. Networks with shared weights exhibit smoother layer-wise transformations. More specifically, this visualization provides insights into the geometrical properties (such as topology preservation/connectedness) of the transformed set of input data points.

To evaluate the quality of the estimation warping transformation, Fig. 6 shows the Jacobian maps for each considered network. Local Jacobian sign changes correspond to locations where bijectivity is not satisfied. It can be seen that adding constraints such as stationary velocity fields and inverse consistency leads to more regular geometrical shapes of the deformed manifold. The domain-driven regularization applied to a ResNet with shared weights leads to the most regular geometrical pattern. Adding the symmetry consistency leads to positive Jacobian values over the entire domain, guaranteeing the bijectivity of the estimated mapping.

5 Discussion: Insights into ResNet Architectures from a Diffeomorphic Viewpoint

As illustrated in the previous section, the proposed diffeomorphic formulation of ResNets provides new theoretical and computational insights into their interpretation and characterization as discussed as follows.

5.1 Theoretical Characterization of ResNet Architectures

In this work, we make explicit the interpretation of the mapping block of ResNet architectures as a discretized numerical implementation of a continuous diffeomorphic registration operator. This operator is stated as an integral operator associated with an ODE governed by velocity fields. Moreover, ResNet architectures with shared weights are viewed as the numerical implementation of the exponential of velocity fields, equivalently defined as diffeomorphic operators governed by stationary velocity fields. Exponentials of velocity fields are by construction diffeomorphic under smoothness constraints on the generating velocity fields. Up to the choice of the ODE solver implemented by ResNet architecture (in our case an Euler scheme), ResNet architectures with shared weights are then fully characterized from a mathematical point of view.

The diffeomorphic property naturally arises as a critical property in registration problems, as it relates to invertibility properties. Such invertibility properties are also at the core of the definition of kernel approaches, which implicitly defines mapping operators [26]. As illustrated for the reported classification experiments, the diffeomorphic property prevents the mapping operator from modifying the topology of the manifold structure of the input data. When not imposing such properties, for instance, in unconstrained ResNet architectures, the learned deformation flows may present unexpected topology changes.

The diffeomorphic property may be regarded as a regularization criterion on the mapping operator, so that the learned mapping enables a linear separation of the classes while guaranteeing the smoothness of the classification boundary and of the underlying deformation flow. It is obvious that a ResNet architecture with shared weights is a special case of an unconstrained ResNet. Therefore, the training of a ResNet architecture with shared weights may be viewed as the training of an unconstrained ResNet within a reduced search space. The same holds for the symmetry property which further constrains the search space during training.

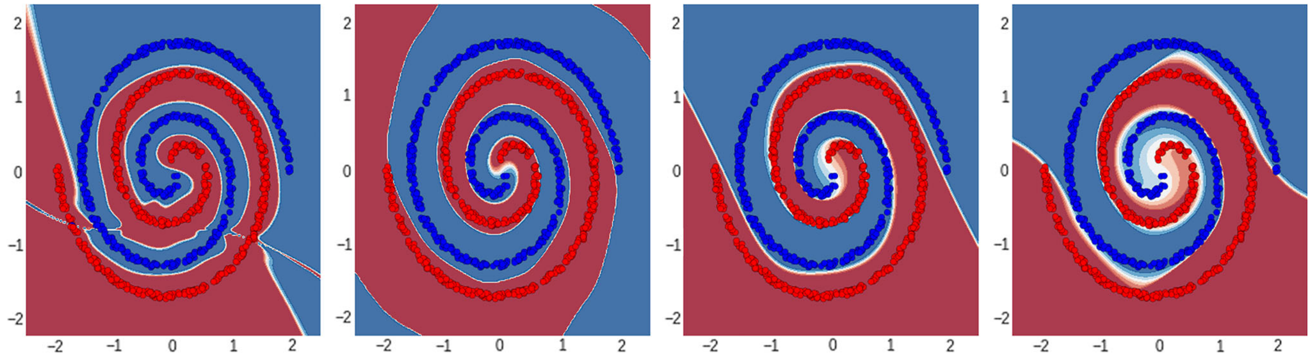

Fig. 4 Decision boundaries for the classification task of two-dimensional spiral data. From left to right: ResNet without shared weights, ResNet with shared weights, data-driven symmetric ResNet with shared weights, domain-driven symmetric ResNet with shared weights. We refer the reader to the main text for the correspondence between ResNet architectures and diffeomorphic flows

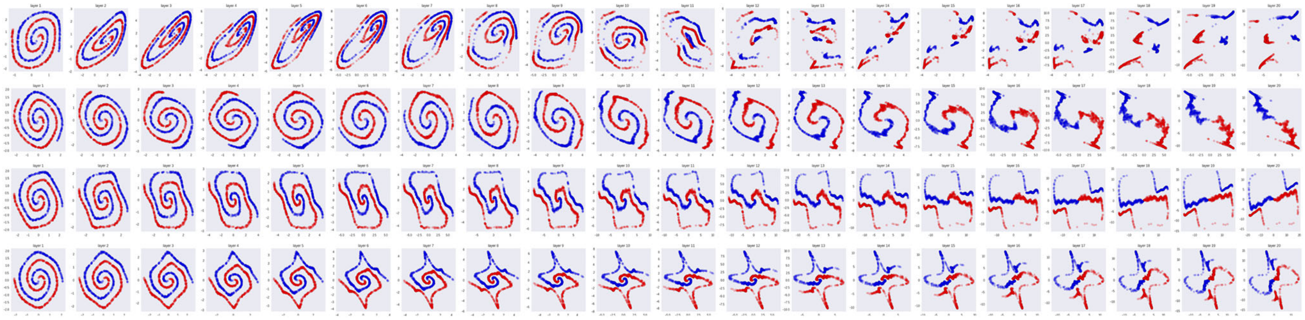

Fig. 5 Evolution of the spatial configuration of data points through the 20 residual units. From top to bottom: ResNet without shared weights, ResNet with shared weights, data-driven symmetric ResNet with shared weights, domain-driven symmetric ResNet with shared weights

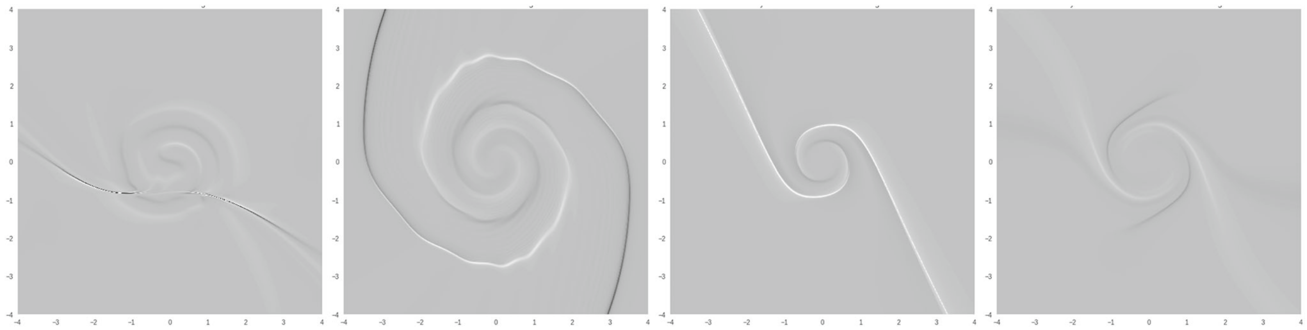

Fig. 6 Jacobian maps for the four ResNet architectures. From left to right: ResNet without shared weights ($J_{\min} = -5.59$, $J_{\max} = 6.34$), ResNet with shared weights ($J_{\min} = -1.41$, $J_{\max} = 2.27$), data-driven symmetric ResNet with shared weights ($J_{\min} = 0.55$, $J_{\max} = 5.92$), domain-driven symmetric ResNet with shared weights ($J_{\min} = 0.30$, $J_{\max} = 1.44$). (colormap: $J_{\min} = -2.5$, $J_{\max} = 2.5$, so dark pixels correspond to negative Jacobian values)

The later constraint is shown to be numerically important, so that the discretized scheme complies with the theoretical diffeomorphic properties.

Overall, this analysis stresses that over an infinity of mapping operators reaching optimal training performance one may favor those depicting diffeomorphic properties so that key properties such as generalization performance, prediction stability and robustness to adversarial examples may be greatly improved. Numerical schemes which fulfill such diffeomorphic properties during the training process could be further investigated and could benefit from the registration literature, including for diffeomorphic flows governed by non-stationary velocity fields [4,5,30]. In particular, the impact of the diffeomorphism-based network building on adversarial example estimation is an open research direction for further studies.

5.2 Computational Issues

Besides theoretical aspects, computational properties also derive from the proposed diffeomorphism-based formulation. Within this continuous setting, the depth of the network relates to the integration time step and the precision of the integration scheme. The deeper the network, the smaller the integration step. In particular, a large integration time step, i.e., a shallower ResNet architecture, may result in numerical integration instabilities and hence in non-diffeomorphic transformations.

Therefore, deep-enough architectures should be considered to guarantee numerical stability and diffeomorphic properties. The maximal integration step relates to the regularity of the velocity fields governing the ODEs. In our experiments, we only consider an explicit first-order Euler scheme. Higher-order explicit schemes, for instance, the classic fourth-order Runge–Kutta scheme, seem of great interest as well as implicit integration schemes [8]. Given the spatial variabilities of the governing velocity fields, adaptive integration schemes also appear as particularly relevant.

Using diffeomorphism-based framework leads to specific architectures of residual units. In this work, for instance, tan$h$ activation layers are used to constraint the domain Ω to be open and bounded. Such activation layer guarantees the diffeomorphic properties of the mapping block. The popular use of ReLU activation in the embedding block cannot provide such guarantee. Several other reversible architectures have been recently proposed [7,10,17], showing the potential of such frameworks for the analysis of residual networks. In the LDDMM framework [5], the parametrization of the velocity field is often carried out with Reproducing Kernel Hilbert Spaces (RKHS). Recent works have been done in this direction connecting RKHS and deep networks [6]. In our work, v and Dv vanish on $\partial\Omega$ and at infinity. This property guarantees that the learned residual units are Lipschitz continuous, which is related to recent works investigating explicit constraints on Lipschitz continuity in neural networks [11]. Moreover, this condition implies that the Hilbert space of admissible velocity fields is a RKHS [34]. Further work could focus on the parametrization of the velocity fields (i.e., residual units) using suitable kernels.

Diffeomorphic mapping defined as exponential of velocity fields was shown to be computationally more stable with smoother integral mappings. They lead to ResNet architectures with shared weights, which greatly lowers the computational complexity and memory requirements compared with the classic ResNet architectures. They can be implemented as recurrent neural networks [19,23]. Importantly, the NN-based specification of the elementary of velocity field V (8) becomes the bottleneck in terms of modeling complexity.

The parametrization (Eq. 8) may be critical to reaching good prediction performance. Here, we considered a two-layer architecture regarded as a projection of V onto basis function. Higher-complexity architecture, for instance with larger convolution supports, more filters or layers, might be considered while keeping the numerical stability of the overall ResNet architectures. By contrast, considering higher-complexity elementary blocks in a ResNet architectures without shared weights would increase numerical instabilities and may require complementary regularization constraints across network depth [15,25].

Regarding training issues, our experiments exploited a classic backpropagation implementation with a random initialization. From the considered continuous log-Euclidean prospective, the training may be regarded as the projection of the random initialization onto the manifold of acceptable solutions, i.e., solutions satisfying both the minimization of the training loss and diffeomorphic constraints. In the registration literature [27], the numerical schemes considered for the inference of the mapping usually combine a parametric representation of the velocity fields and a multiscale optimization strategy in space and time. The combination of such multiscale optimization strategy to backpropagation schemes appears as a promising path to improve convergence properties, especially the robustness to the initialization. The different solutions proposed to enforce diffeomorphic properties are also of interest. Here, we focused on the invertibility constraints, which result in additional terms to be minimized in the training loss.

6 Conclusion

This paper introduces a novel registration-based formulation of ResNets. We provide a theoretical interpretation of ResNets as numerical implementations of continuous flows of diffeomorphisms. Numerical experiments support the relevance of this interpretation, especially the importance of the enforcement of diffeomorphic properties, which ensure the stability of a trained ResNet. This work opens new research avenues to explore further diffeomorphism-based formulations and associated numerical tools for ResNet-based learning, especially regarding numerical issues.

Acknowledgements We thank B. Chapron and C. Herzet for their comments and suggestions. The research leading to these results has been supported by the ANR MAIA Project, Grant ANR-15-CE23-0009 of the French National Research Agency, INSERM and Institut Mines Télécom Atlantique (Chaire “Imagerie médicale en thérapie interventionnelle”) and Fondation pour la Recherche Médicale (FRM Grant DIC20161236453), Labex Cominlabs (Grant SEACS), CNES (Grant OSTST-MANATEE) and Microsoft trough AI-for-Earth EU Oceans Grant (AI4Ocean). We also gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan Xp GPU used for this research.

References

1. Arsigny, V.: Processing data in Lie groups: an algebraic approach. Application to non-linear registration and diffusion tensor MRI. Ph.D. thesis, Ecole Polytechnique (2006)
2. Arsigny, V., Commowick, O., Pennec, X., Ayache, N.: A log-Euclidean framework for statistics on diffeomorphisms. In: International Conference on Medical Image Computing and Computer-Assisted Intervention: MICCAI, vol. 9(Pt 1), pp. 924–931 (2006)
3. Ashburner, J.: A fast diffeomorphic image registration algorithm. NeuroImage **38**(1), 95–113 (2007)
4. Avants, B., Gee, J.C.: Geodesic estimation for large deformation anatomical shape averaging and interpolation. NeuroImage **23**, S139–S150 (2004)
5. Beg, M.F., Miller, M.I., Trouvé, A., Younes, L.: Computing large deformation metric mappings via geodesic flows of diffeomorphisms. Int. J. Comput. Vis. **61**(2), 139–157 (2005)
6. Bietti, A., Mairal, J.: Group invariance, stability to deformations, and complexity of deep convolutional representations. J. Mach. Learn. Res. **20**(25), 1–49 (2019)
7. Chang, B., Meng, L., Haber, E., Ruthotto, L., Begert, D., Holtham, E.: Reversible architectures for arbitrarily deep residual neural networks. In: Neural Network cs.CV (2017)
8. Davis, P.J., Rabinowitz, P.: Methods of Numerical Integration. Academic Press, London (1984)
9. Glorot, X., Bengio, Y.: Understanding the difficulty of training deep feedforward neural networks. In: AISTATS (2010)
10. Gomez, A.N., Ren, M., Urtasun, R., Grosse, R.B.: The reversible residual network—backpropagation without storing activations. In: NIPS (2017)
11. Gouk, H., Frank, E., Pfahringer, B., Cree, M.J.: Regularisation of neural networks by enforcing Lipschitz continuity. In: Neural Networks stat.ML (2018)
12. Haber, E., Ruthotto, L.: Stable architectures for deep neural net works. Inverse Probl. (2018). https://doi.org/10.1088/1361-6420/aa9a90
13. Hauser, M., Ray, A.: Principles of Riemannian geometry in neural networks. In: NIPS (2017)
14. He, K., Zhang, X., Ren, S., Sun, J.: Delving deep into rectifiers: surpassing human-level performance on ImageNet classification. In: IEEE ICCV (2015). https://doi.org/10.1109/ICCV.2015.123
15. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 770–778. IEEE (2016)
16. He, K., Zhang, X., Ren, S., Sun, J.: Identity mappings in deep residual networks. In: ECCV (2016). https://doi.org/10.1007/978-3-319-46493-0_38
17. Jacobsen, J.H., Smeulders, A., Oyallon, E.: i-RevNet: deep invertible networks. In: International Conference on Learning Representations (ICLR) (2018)
18. Kim, J., Lee, J.K., Lee, K.M.: Accurate image super-resolution using very deep convolutional networks. In: CVPR (2016)
19. Kim, J., Lee, J.K., Lee, K.M.: Deeply-recursive convolutional network for image super-resolution. In: CVPR (2016)
20. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization (2014)
21. Klein, A., Andersson, J., Ardekani, B.A., Ashburner, J., Avants, B., Chiang, M.C., Christensen, G.E., Collins, D.L., Gee, J., Hellier, P., Song, J.H., Jenkinson, M., Lepage, C., Rueckert, D., Thompson, P., Vercauteren, T., Woods, R.P., Mann, J.J., Parsey, R.V.: Evaluation of 14 nonlinear deformation algorithms applied to human brain MRI registration. NeuroImage **46**(3), 786–802 (2009)
22. Krizhevsky, A., Hinton, G.: Learning multiple layers of features from tiny images. University of Toronto (2009)
23. Liao, Q., Poggio, T.A.: Bridging the gaps between residual learning, recurrent neural networks and visual cortex. In: Neural Networks (2016)
24. Mallat, S.: Understanding deep convolutional networks. Philos. Trans. R. Soc. A Math. Phys. Eng. Sci. **374**(20), 20150203-16 (2016)
25. Ruthotto, L., Haber, E.: Deep neural networks motivated by partial differential equations (2018)
26. Scholkopf, B., Smola, A.J.: Learning with kernels. Support Vector Machines, Regularization, Optimization, and Beyond. MIT Press, Cambridge (2002). https://mitpress.mit.edu/books/learning-kernels
27. Sotiras, A., Davatzikos, C., Paragios, N.: Deformable medical image registration: a survey. IEEE Trans. Med. Imaging **32**(7), 1153–1190 (2013)
28. Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., Fergus, R.: Intriguing properties of neural networks. In: ICLR (2014)
29. Thirion, J.: Image matching as a diffusion process: an analogy with Maxwell's demons. Med. Image Anal. **2**(3), 243–260 (1998)
30. Trouvé, A.: Diffeomorphisms groups and pattern matching in image analysis. Int. J. Comput. Vis. **28**(3), 213–221 (1998)
31. Weinan, E.: A proposal on machine learning via dynamical systems. Commun. Math. Stat. **5**(1), 1–11 (2017)
32. Xie, S., Girshick, R.B., Dollár, P., Tu, Z., He, K.: Aggregated residual transformations for deep neural networks. In: CVPR, pp. 5987–5995 (2017)
33. Younes, L.: Shapes and diffeomorphisms. Applied Mathematical Sciences, vol. 171. Springer, Berlin (2010)
34. Younes, L.: Diffeomorphic learning (2018)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

François Rousseau received the Ph.D. degree in Signal Processing and Telecommunications from the University of Rennes, France, in 2003. From 2004 to 2005, he was a post-doctoral fellow at UCSF, USA. From 2006 to 2015, he held a full-time research position at CNRS in the field of medical image processing applied to brain imaging. In 2015, he joined the IMT Atlantique (formerly, Telecom Bretagne) as a Professor. His main interests include image analysis and applications to medical data.

Lucas Drumetz received the Ph.D. degree in 2016 in Signal and Image Processing at the University of Grenoble Alpes, for work carried out at the GIPSA-lab Laboratory. In 2017, he was a visiting assistant professor in the Department of Mathematics at the University of California, Los Angeles (UCLA). In 2017, he also completed a 10-week research stay at the RCAST Laboratory at Tokyo University in Japan. He is an associate Professor at IMT Atlantique. His areas of interest include satellite and airborne imagery (including hyperspectral and radar altimetry data), inverse problems in remote sensing, signal processing, mathematical optimization, automatic learning and parsimonious methods.

Ronan Fablet received the Ph.D. degree in Signal Processing and Telecommunications from the University of Rennes, France, in 2001. In 2002, he was an INRIA postdoctoral fellow at Brown University, RI, USA. From 2003 to 2007, he held a full-time research position at Ifremer Brest in the field of signal and image processing applied to fisheries science. In 2008, he joined the signal and communications department of IMT Atlantique (formerly, Telecom Bretagne) as an Associate Professor, and has been holding a Professor position since 2012. His main interests include signal processing and computer vision and applications to ocean remote sensing data.