Distance transform data augmentation and stochastic patch-wise image prediction methodology for small dataset learning

Adam Hammoumi, Maxime Moreaud, Christophe Ducottet, Sylvain Desroziers

Distance transform data augmentation and stochastic patch-wise image prediction methodology for small dataset learning

Adam Hammoumi[a,*], Maxime Moreaud[a,b], Christophe Ducottet[c], Sylvain Desroziers[d]

[a] *IFP Energies nouvelles, Rond-point de l'échangeur de Solaize BP 3, 69360 Solaize, France*
[b] *MINES ParisTech, PSL-ResearchUniversity, CMM, Fontainebleau, France*
[c] *Université de Lyon, UJM-Saint-Etienne, CNRS, IOGS, Laboratoire Hubert Curien UMR5516, F-42023, Saint-Etienne, France*
[d] *IFP Energies nouvelles, 1 et 4 avenue de Bois-Préau, 92852 Rueil-Malmaison, France*

Abstract

Most recent methods of image augmentation and prediction are building upon the deep learning paradigm. A careful preparation of the image dataset and the choice of a suitable network architecture are crucial steps to assess the desired image features and, thence, achieve accurate predictions. We first propose to help the learning process by adding structural information with specific distance transform to the input image data. To handle cases with limited number of training samples (as 12 training and 2 validation images), we propose a patch-based procedure with a stratified sampling method. We illustrate our approaches on image dataset generated by an FFT-based homogeneization technique for heterogeneous media physical properties. The obtained results are evaluated using SSIM, UIQ and PSNR metrics. The proposed techniques demonstrate that the established framework is a reliable estimation method that could be used for a wide range of applications.

Keywords: image augmentation, deep learning, distance transform, patch-wise segmentation, stratified sampling

Preprint submitted to Neurocomputing on Apr 15, 2020

*Corresponding author
Email address: adam.hammoumi@ifpen.fr (Adam Hammoumi)

1. Introduction

Deep Convolutional Neural Networks (DCNNs) have demonstrated throughout the recent years their remarkable performance in handling a variety of problems in the fields of image processing and computer vision [1][2]. In particular, DCNNs are becoming a major tool for visual recognition modern tasks such as image classification, segmentation, semantic segmentation and so on. Considering the global dimension that DCNNs have taken, predominant implementations often require large datasets, which is not always possible in many domains. Regarding that, the interest in small sample learning (SSL) is increasingly growing [3]. Novel network topologies and training methodologies are required to address this issue. In the litterature, there are many approaches that attempt to face out the SSL paradigm. The augmented data approach tries to compensate the lack of samples by applying adequate transformations to the original dataset [4]. Another example is knowledge transfer of fully trained networks that can be used to fit small datasets [5]. In this paper, we propose a training methodology that is independent from network architecture and field of application. The proposed methodology consists of a patch-based procedure with stratified sampling along with a data augmentation technique. Patch-based methods were intensively used in the recent years [6][7][8]. Besides enlarging the sample size by feeding the network with a large number of extracted patches, they capture local features and don't require large memory footprint. We address the problem of data augmentation in a new way using a patch-based technique coupled with a stratified sampling strategy [9] to cancel an apparent edge effect at the patch boundaries. We also propose augmenting the training images by adding structural information to reinforce the learning process. A classical DCNN starts the process of features extraction according to a growing architecture from low level to high level features. The dynamical aspect of convolutional filters is the reason why abstract features –often, not relevant for a human observer– are recognized as important image variant characteristics. Taking advantage of the maximum implicit and explicit information that can be embedded in one image can only be beneficial for pertinent training. Used augmentation data is extracted from a distance map of the considered images [10], which provides implicit information. The method yields a mapping of each element in the background to its distance from the closest foreground element. In a symmetrical way, the distance map for foreground space is provided. In this configuration, less informative surfaces become more informative throughout all the image space. A use case of the distance transform in the framework of semantic segmentation

regularization has been addressed in recent work [11]. These concepts will be illustrated through estimation of a grayscale image from a binary image. This application study corresponds to the approximation using DCNN of the dielectric field computed initially by homogenization technique from a microstructure of a model material. Our methodology doesn't rely on any ad hoc application. The intuition behind is to offer meaningful ideas that aim to improve the network outcomes in the case of SSL.

2. Methods

2.1. Fully Convolutional Neural Network

While DCNNs for image classification are predicting a single class for a whole image, fully convolutional networks (FCNs) can be used to make dense predictions. Given an input image of any size, a FCN produces an output having the same spatial support (possibly resampled) and predicting a value associated to each input pixel (or each group of pixels). For instance, in image augmentation or prediction, each pixel must be labeled. Meaning that, a pixel-wise mask of each element is created in the initial image. The goal is to distinguish between elements by classifying every pixel to the desired labels. To formulate a problem of image prediction, a set of possible grayscale image values that represents the pixel brightness can be defined. Generally, in a convolutional neural network, the input image goes through the convolutional layers for features extraction and gets downsized by the pooling layers. The results of the convolution/pooling operations are fed to the fully connected layers (FC) to classify the image into a label. To obtain a label map instead of a single value label, an upsampling step is mandatory to calculate a pixel wise output. Our attention is drawn to FCNs and dense prediction [12][13] since they recapture the spatial information lost during downsampling operations by upsampling or deconvolution. Therefore, a FCN architecture transforms the size of the label map back to the size of input image or a subsampled version through the upsampling process so that the predictions have a pixel-to-pixel correspondence with the input image. U-Net is a popular FCN that demonstrated its performance in segmenting biomedical images [14]. It consists of a contraction path made of consecutive 3×3 convolutions followed by 2×2 max pooling matrices, an expansion path which is composed of series of 3×3 convolutions and 3×3 transposed convolution matrices. The aim from this step is to concatenate the features maps with the corresponding layers from the contraction path to regain spatial information. A final layer is a 1×1 convolution matrix that yields the final labeled image. In this work, we keep the original U-Net

architecture as shown in figure (1). It consists of a series of two convolutions per max pooling. The ReLU activation function is used for of each convolution and it is preceded by a batch normalization operation. After each max pooling, the size of the image is divided by 2 and the number of features maps (or, channels) is doubled. The transposed convolution operations are used to up scale the images to a higher resolution while the concatenation operation collects information from features map of the contraction path. The training is made on the basis of a sliding 48×48 window cropped from the initial image of the size 500×500. The output is obtained from a 1×1 convolution followed by a sigmoid activation function.

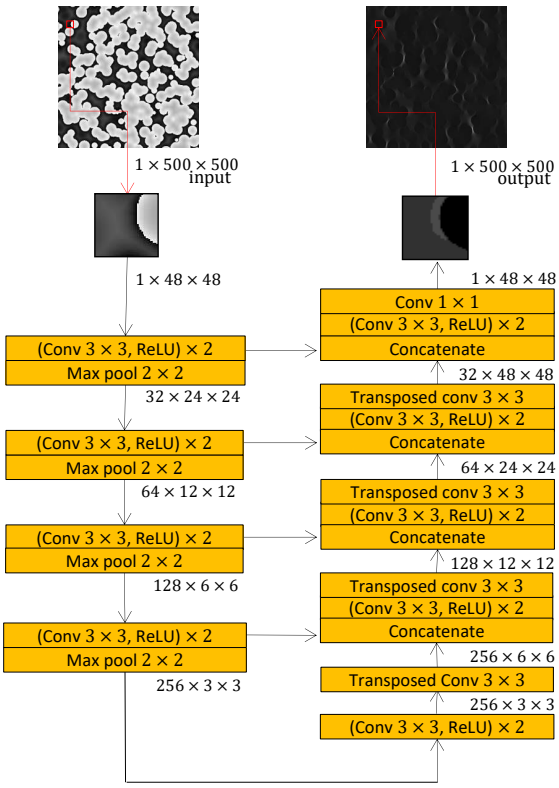


Figure 1: U-Net architecture. Left: contraction path; right: expansion path. After each set of operations, the size of the image and the number of channels is indicated. Operations are: convolution, transposed convolution, max pooling and concatenation. Input and output images are described later in the experimental results section.

2.2. Patch procedure

Training a Convolutional Neural Network at the pixel level of each image can be a challenging task in several cases. For example, in many medical applications as shown in [6], the training data is a set of high resolution images, which will require a very large memory footprint. An additional drawback to this approach is the risk to bias the training by forcing the network to only learn the most distinctive features from the whole image. On the other hand, the process of gathering a large set of training data in many domains is not always possible. Many recent works tried to address this issue. In [7], a patch-wise setup was firstly introduced to predict the class label of each pixel. This method has the big advantage to train the network on a large set of patches instead of the few original training images. The question of the use of context arises regarding this approach. Basically, the size of the patches and the number of hidden layers of the FCN control the field of view of the network and contribute explicitly in the learning of important features. For example, the authors in [8] postulate that a small patch size is not needed for their specific case, since there is little chance of finding relevant information in small image regions. Other techniques as the one shown in [15] propose to combine pixel-level and patch-level to improve the segmentation accuracy. We follow the patch-based method proposed in [7]. A patch is characterized by its size $K \times K$ with a sliding step s over the image. That makes a total number $(1 + (I_W - K)/s) \times (1 + (I_H - K)/s)$ of patches for an image of size $I_W \times I_H$, which are the width and the height of the image respectively. By trying several patch sizes and sliding steps, we found out that a patch size $K = 48$ with a sliding step $s = 24$, give the best results for the application we used to illustrate our method. We assume that overlapping patches allow to extract significant information captured in-between patches. This hypothes was verified empirically. Given the size of the training images $I_W = I_H = 500$, the total number of patches contained in each image is $L = 400$. Following this method, we augment our data training number from 12 images to a total of $12 \times 400 = 4800$ patches. Figure (2) shows the process of patch extraction.

2.3. Shift invariance

CNNs are believed to be translation invariant [16] at some degree. Although convolutional layers have a property of equivariance to translation [17], it is not exactly the same for the complete network. This general consensus is supported by the fact that the networks have the inherent ability of learning arbitrary features: important ones, but also features as affine

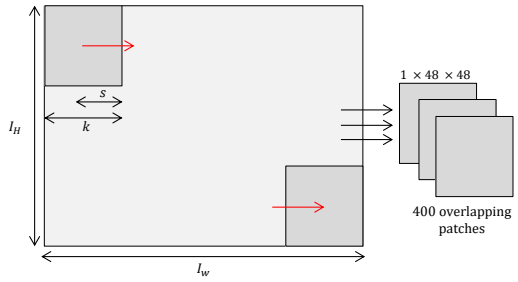


Figure 2: Illustration of the patch extraction process. Parameters: $I_H = I_W = 500$, $K = 48$ and $s = 24$

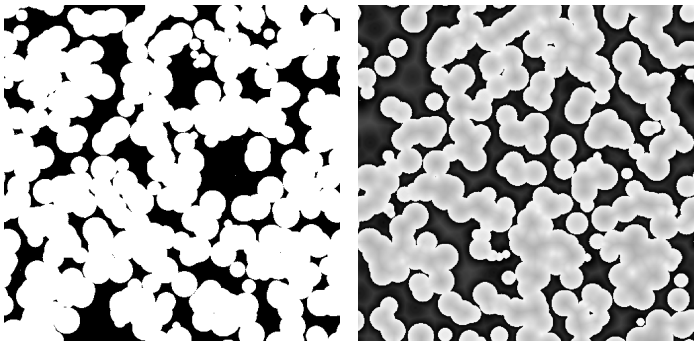
transformations that are irrelevant and must be discarded. Two main ideas in the litterature try to address this issue. According to the first, the ability to learn translation invariance is due primarily to the networks architecture, in which the succession of convolution layers augments the receptive field of neurons [18], and to pooling layers that select a value from convolution layer output regardless of its position [19]. Using translation sensitivity maps and radial translation-sensitivity to quantify shift invariance introduced in [20], it can be demonstrated that the use of appropriate input data along with data augmentation comes beyond the network architecture in terms of learning translation invariant representation. A careful examination of the extent to which the U-Net architecture is shift invariant is relevant information for data preparation. Several outcomes are provided in the experimental results section.

2.4. Pre-processing of training data

In the following, for illustration, images of binary microstructures (white: solid space, black: porosity space) and their corresponding electric field response images representing the ground truth data are used. A detailed explanation of how we obtained such images is discussed in the experimental results section.

2.4.1. Distance transform

Our focus in this part is to demonstrate the effectiveness of a novel strategy in pre-processing the training data. Following the patch-based approach and from the network perspective, roughly speaking, flat surfaces contained



(a) Original microstructure (b) Augmented microstructure

Figure 3: Illustrations of the process of adding information to the input image: (a) original microstructure; (b) augmented microstructure, expressed by Eq.(4).

in the image are less informative than textured surfaces. Obviously, the lack of information in these regions will cancel the effect of the convolution matrices of the network. Moreover, features outside the size of the matrix will not be extracted. Considering first the case of binary images as training data, distance map computation is a commonly used technique in several image processing tasks such as connected component labeling [21], skeletonization [22], Voronoi diagrams and so on. For binary images, the distance map can be computed in the following way: we consider the two dimensional metric space $E = \mathbb{R}^2$. Let $I\colon \psi \to \{0, 255\}$ be a binary image and $\psi \subset E$ the support of I. We separate objects of I into two categories: background and foreground elements, the latter denotes feature objects in the image. To probe the background space, we let the set of foreground space elements $\omega = \{x \in \psi : I(x) = 255\}$ be the reference set of features. A distance map is an image transform that substitutes the value of each element in ψ by its distance from the closest feature object of ω. The operator of the distance transform is:

$$DT^d(x) = \min_{\{y|I(y)=0\}} d(x, y) \qquad x, y \in \psi \tag{1}$$

A more general formulation of the distance transform that extends to grayscale and color images may be found in [23]. The distance between two points x and y is expressed by:

$$d(x, y) = \inf_{\Gamma \in P_{x,y}} \int_0^{l(\Gamma)} \sqrt{1 + \gamma^2(\,\nabla I(s) \cdot \Gamma'(s)^2)}\, ds \tag{2}$$

where Γ is a path parameterized by its arc length $s \in [0, l(\Gamma)]$ and $P_{x,y}$ is the set of all differentiable paths.The geodesic factor γ measures the contribution of the image gradient $\nabla I(s)$ and spatial distances. $\Gamma^{'}(s) = \partial\Gamma(s)/\partial s$ is the unit vector tangent to the direction of the path. Notice that the binary image distance transform is a special case of Eq.(2), where the image has scalar values $\{0, 255\}$ and $\gamma = 0$. In this case, Eq.(2) simplifies to the euclidean length of path Γ. Our strategy to extract a maximum amount of information from the image consists of probing both the background and foreground space. The related distance map to the latter space is:

$$DT_c^d(x) = \min_{\{y|I(y)=255\}} d(x,y) \qquad x, y \in \psi \tag{3}$$

We now define an augmented binary image as:

$$I(x) = I(x) + \alpha DT^d(x) - \beta DT_c^d(x) \quad \alpha, \beta \quad constants \tag{4}$$

It is possible to compute distance map for grayscale image using approach from [24]. It corresponds to a distance transform starting from lowest to highest grayscale intensities:

$$\bar{DT}^d(x) = \frac{1}{255} \sum_i d(x, F_i) w_i \,|F_i = \{x; I(x) \geq i\},\ w_i = 1 \tag{5}$$

Similarly, an extended symmetric distance map going from highest to lowest grayscale intensities can be defined:

$$\bar{DT}_c^d(x) = \frac{1}{255} \sum_i d(x, G_i) w_i \,|G_i = \{x; I(x) < i\},\ w_i = 1 \tag{6}$$

The resulting augmented grayscale image writes:

$$I(x) = I(x) + \alpha \bar{DT}^d(x) - \beta \bar{DT}_c^d(x) \quad \alpha, \beta \quad constants \tag{7}$$

The former distance has an important time complexity, which leads us to define an approximated distance map for grayscale images that can be deduced from a functional projected distance map $d^{\perp}$ [25]. This projected distance makes it independent of greyscale scaling. It suggests a similar formulation of the distance in the background space using the set of pixels of lowest intensities:

$$\hat{DT}^d(x) = \min_{\{y|I(y)=min(I)\}} d^{\perp}(x,y) \qquad x, y \in \psi \tag{8}$$

where $d^{\perp}$ is the projected distance of the one developed in Eq. (2). Likewise, its symmetric distance map is given by:

$$\hat{DT}_c^d(x) = \min_{\{y|I(y)=max(I)\}} d^{\perp}(x,y) \qquad x,y \in \psi \tag{9}$$

The augmented grayscale image writes:

$$I(x) = I(x) + \alpha \hat{DT}^d(x) - \beta \hat{DT}_c^d(x) \quad \alpha,\beta \quad constants \tag{10}$$

Distance transform can be computed with a two pass raster scanning algorithm which is well established in the literature [26]. The augmentation of the images lies on adding information, seen in Eqs. (4), (7) or (10), and helps the network learning accurately patterns and flat surfaces. Figure (3) illustrates this process. In our illustrations and results parts, α and β are taken equal to 1. A grayscale image example of data augmentation according to the former distance transform formalism is provided in figure (4).

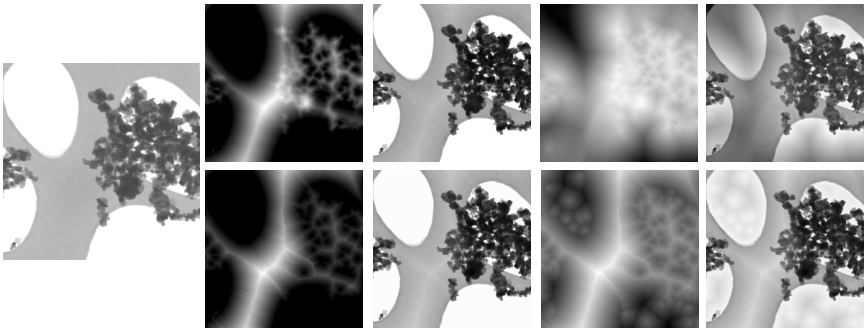


Figure 4: Left: initial image (boehmite aggregate). From left to right: distance transform, combined distance transform, symmetric distance transform , combined symmetric distance transform; with at the top, grayscale distance transform from [24], and bottom, fast approach with projected distance transform from [25].

2.5. Stratified sampling of patches

On the basis of individual patches extracted from the input image, the estimated image is assembled from clustered patches predicted by the network. If they are regularly distributed, an edge effect in the border of each patch may arises, as shown in figure (9). On the contrary, an exact convolution strategy (one patch for each pixel) leads to fuzzy and reduced quality results. We show these problems in the experimental results section. Hence, an adequate sampling strategy is required to reduce this effect. Consider

the set of patches contained in one image $\chi = \{\chi_i\}_{i\in[0,L-1]}$. L is the number of patches. Each patch occupy a total area of $A = W \times H$ where W and H are the width and the height of each patch respectively. An uniform sampling strategy consists of cropping patches as fragments of the original image following a scanning strategy from top to bottom and from left to right. Mathematically, this sampling strategy boils down to:

$$\chi_{i,j} = \{x, y \mid x \in [i, W+i] \,,\; y \in [j, H+j] \,\} \text{ with,} \tag{11}$$

$$\begin{aligned} i &= 0, s, (2\times s), (3\times s), ..., (I_w - W + s) \\ j &= 0, s, (2\times s), (3\times s), ..., (I_h - H + s) \end{aligned} \tag{12}$$

Eq.(11) correspond to a formulation of the patches and the indexation strategy shown in Eq.(12) yields a uniform sampling. To remove the edge effect on the borders, patches must be drawn stochastically. We propose using the stratified sampling strategy [9]. It consists of a uniform density of sampled points $\mathbf{U}(-\mathbf{s}, \mathbf{s})$ covering the whole size of the patch. Indexing the patches with random coordinates will guarantee the generation of fresh ones every time. We rewrite the new indexation as:

$$\begin{aligned} m &= i + \mathbf{U}(-\mathbf{s}, \mathbf{s}) \\ n &= j + \mathbf{U}(-\mathbf{s}, \mathbf{s}) \end{aligned} \tag{13}$$

Consider a set of N images representing the prediction result of the same input image. We divide this set to N subsets (or, strata). Each stratum is an image where the patches were drawn randomly. We generate $2 \times L \times N$ random points. The final result should be obtained by averaging the strata. We write:

$$I_f = 1/N \times \left(\sum_{v=1}^{N} \sum_{m,n}^{I_w - W + s, I_H - H + s} \chi_{v,m,n} \right) \tag{14}$$

where m and n have random values according to Eq.(13) and vary constantly. As a consequence, the distribution of patches in each image is unique.

2.6. Evaluation metrics

The performance of our proposed methods is investigated through several metrics. We use some of the most common reference image quality measures, in particular the ones that are based on different measuring approaches. The goal is to measure dissimilarities between two images. Hereafter in this section, we use the formulation $I_\star = \{I_\star(i,j);\ \forall i = 1, ...W,\ \forall j = 1, ...H\}$, $with \star = \{o, d\}$ for both original (ground truth) and distorted (predicted) image.

2.6.1. PSNR

Peak signal-to-noise ratio (PSNR) –related to the mean squared error (MSE)– is firstly deployed. PSNR is based upon an explicit numerical criterion which is the comparison between pixel values. Let I_o be the original image and I_d the distorted image. To perform a comparison between these images, the PSNR metric writes:

$$\mathrm{PSNR}(\mathrm{I_o}, \mathrm{I_d}) = 10 \times \log_{10} \left[\frac{(2^d - 1)^2}{MSE(I_o, I_d)} \right] \text{ , with} \tag{15}$$

$$\mathrm{MSE}(\mathrm{I_o}, \mathrm{I_d}) = \frac{1}{W \times H} \sum_{i=0}^{W-1} \sum_{j=0}^{H-1} \left(I_o(i,j) - I_d(i,j) \right)^2 \tag{16}$$

$2^d - 1$ denotes the maximum possible value that a pixel can have. For $d = 8$ byte coded image, the maximum value is 255. Eq. (16) measures the value differences between corresponding pixels of each image. PSNR is expressed in decibels which is a logarithmic unit. From Eq. (15), we can see that higher PSNR value is an indicator of highly similar images.

2.6.2. UIQ

Universal Image Quality (UIQ) [27] is another important tool to measure dissimilarities between two images in terms of their statistical properties. The UIQ index writes:

$$\mathrm{UIQ}(\mathrm{I_o}, \mathrm{I_d}) = \frac{\sigma_{I_o I_d}}{\sigma_{I_o} \sigma_{I_d}} \times \frac{2 \bar{I}_o \bar{I}_d}{\bar{I}_o{}^2 + \bar{I}_d{}^2} \times \frac{2 \sigma_{I_o} \sigma_{I_d}}{\sigma_{I_o}^2 + \sigma_{I_d}^2} \tag{17}$$

where $\bar{I}$ and σ^2 denote mean and variance value, respectively. Eq. (17) is an expression of the UIQ index as a product of three factors: loss of correlation (measures linear correlation), luminance and contrast distortion. UIQ range is [-1,1] so that the index of very similar images approaches 1.

2.6.3. SSIM

Structural similarity index measure (SSIM) [28] is an adaptation of the human visual system (HVS) that aims to assess the structural information of an image. The SSIM equation writes:

$$\mathrm{SSIM}(\mathrm{I_o}, \mathrm{I_d}) = A(I_o, I_d) \times B(I_o, I_d) \times C(I_o, I_d) \text{ , where} \tag{18}$$

$$A(I_o, I_d) = (2\bar{I}_o\bar{I}_d + C_1)/(\bar{I_o}^2 + \bar{I_d}^2 + C_1)$$
$$B(I_o, I_d) = (2\sigma_{I_o}\sigma_{I_d} + C_2)/(\sigma_{I_o}^2 + \sigma_{I_d}^2 + C_2)$$
$$C(I_o, I_d) = (\sigma_{I_oI_d} + C_3)/(\sigma_{I_o}\sigma_{I_d} + C_3)$$

$C_1 = (K_1 \times L)^2$, $C_2 = (K_2 \times L)^2$ and $C_3 = C_2/2$. l is the image dynamic, K_1, K_2 are constants and $\bar{I}$ and σ^2 are the mean and variance values, respectively. SSIM aims to identify the perceptual similarity between two images through the evaluation of luminance (A-Eq.(18)), contrast (B-Eq.(18)) and image structure (C-Eq.(18)). In the presented application, K_1 and K_2 are taken equal to 0.01 and 0.03, respectively.

3. Experimental results

In this section, a series of comparisons between predicted results and ground truth data are performed. The goal is to estimate grayscale from binary images through: a suitable transformation of input data that adds relevant information and an adequate patch sampling method. The training is performed on a dataset of 4800 training and 800 validation patches. The energy function is computed by a *sigmoid* activation function, $f(x) = 1/(1 + e^{-x})$, over the final feature map. The network is trained using the Adam [29] optimizer and *binary cross entropy loss function* defined as: $L(y, y') = -1/N \times \left(\sum_{i=1}^{N}(y \log(y'_i)) + (1 - y)\log(1 - y'_i)\right)$ y and y' being ground truth and predicted patches, respectively. The network achieves its optimal performance after one epoch and a batch size of 4. Only binary and grayscale images are used, colored illustrations are shown solely for the sake of clarity.

3.1. Image dataset

Boolean random models of spheres located by Poisson point process are considered to generate one-scale microstructures [30][31]. A generalization to this process by the Cox Boolean model [32] allows to generate multi-scale microstructures. The power of these models lies on their ability to generate realistic microstructures according to tailor-made characteristics. A specific algorithm described in [33] uses an original construction method which allows to run wide simulations with the least computational cost. We follow the guidelines of the latter algorithm to generate our training images. In this framework, a multi-scale microstructure is modeled by volume fractions that define aggregates ($V_{v,inc}$ of inclusion areas), grains inside and outside the inclusion areas (V_v and $V_{v,out}$, respectively). The training dataset is

made of 500^2 pixel images. The parameters $R = 20$ (radius of spheres), $V_{v,inc} = 0.4$,$V_v = 0.6$ and $V_{v,out} = 0.7$ are fixed for the whole image set. On the basis of the foregoing microstructures, which have known structure and properties, we use an efficient method to estimate the effective dielectric constant (more specifically, the macroscopic equivalent dielectric constant $\epsilon*$). This method lies on several works, namely [34],[35],[36] and [37]. Labeled images are representations of electric field response $E(x)$ estimated inside microstructures and necessary to compute $\epsilon*$. For dielectric constants of the phases of the binary microstucture, 0.1 and 100 (no imaginary part) are used for the black and white pixels respectively. The resulting electric field response module is converted to $8bit$ format (256 values) by uniform sampling. Dataset generations and homogenization codes can be found in the open access software "plug im!" [38].

3.2. Shift invariance

We conduct shift transformations from -5 to $+5$ pixels (in the x's axis) applied to all patches of validation images. Given a fully trained U-Net on the original dataset, we compare the *binary cross entropy* value for each shift value. The goal is to evaluate the ability of the network to predict translated images from non translated training data. Figure (5) shows that the minimum error value coincides with the prediction of a non translated image, and, thence, the network smoothly looses accuracy proportionally to increasing shift values. An alternative approach would consists of feeding a trained network a prediction image I_p that yields a prediction field F_p. We select a patch X_p of the size $W \times H$ centered at position (x, y), the network result is an estimated patch field f_p at position (x, y). Considering shift values h, patches $X_p - h$ of the same size and their respective estimations $f_p - h$ are assessed. The intersection area of these patches is of the size $(W - 2h \times H)$ centered at (x, y). For $h = [-5, 5]$, we compare the dissimilarities through the PSNR, UIQ and SSIM mean values of three different patches between the common area of each shifted patch. Table (2) and figures (5),(6) manifest these differences and quantify the shift "invariance" of the U-Net architecture. That suggests that the network, in our specific case, is not suited for a pixel-wise approach. Rather, a patch based procedure is stabilizing the network by forcing it to learn features from the whole patch dumping non-important features as the shift transformation. The use of U-Net as a predictor for central pixel of patch (pixel-by-pixel approach) leads to fuzzy results due to the averages of patch estimates that are not completely similar.

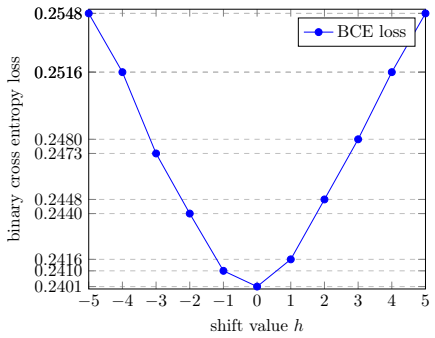


Figure 5: Evolution of the binary cross entropy loss in terms of shift transformations applied to patches of the validation dataset.

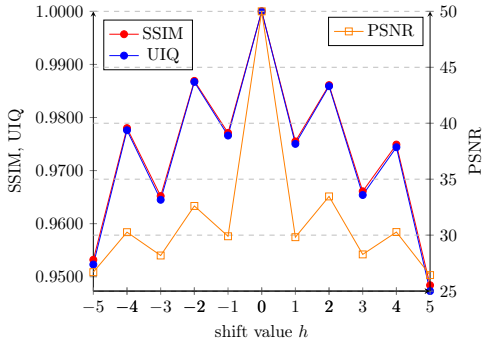
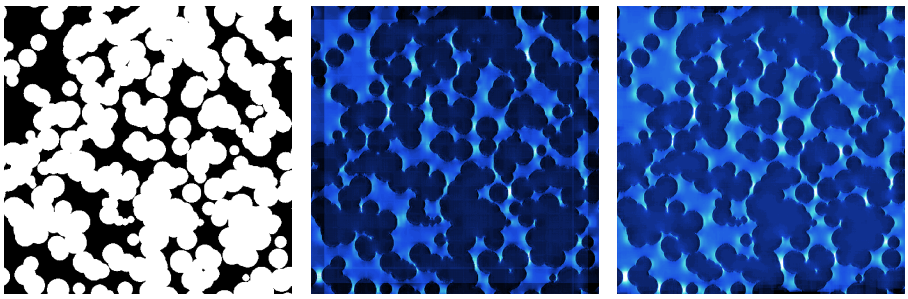


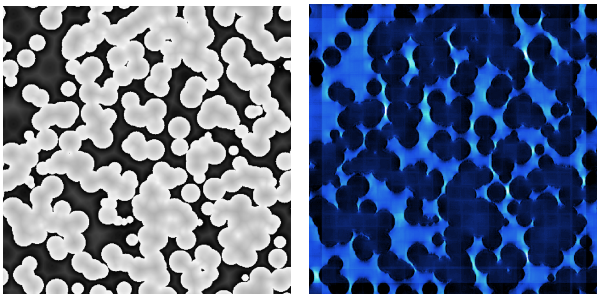
Figure 6: Dissimilarities given by PSNR, SSIM and UIQ mean values between common area of original and shifted patches.

3.3. Effects of enhancing input data

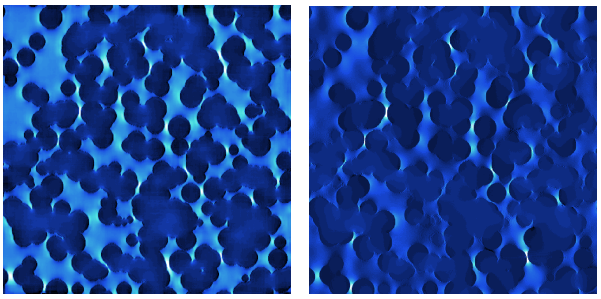
We investigate the effect of augmenting the data by combining the original binary microstructure with distance map. As pointed out in the method section, the distance map enriches the information to be extracted. A comparison of the PSNR, UIQ and SSIM indices is performed for two types of training data. First, we train the network on the original microstructures and identify the similarities between the predicted and the groundtruth image. Then, we do the same for the augmented microstructures training images. A visual comparison is shown in figure (7) and the values of the evaluation metrics are exhibited in table (1). PSNR, UIQ and SSIM indices have improved remarkably for the augmented microstructure.



(a) Original microstructure (b) Predicted image (uniform) (c) Predicted image (stratified)

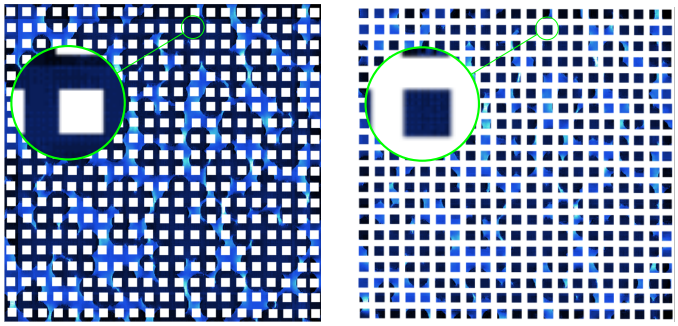


(d) Augmented microstructure (e) Predicted image (uniform)



(f) Predicted image (stratified) (g) Ground truth image

Figure 7: U-Net prediction results [(b), (c), (e) and (f)] for two types of training data: (a) original image. (d) augmented image. (g) is the ground truth image.



(a) MaskB (borders) (b) MaskI (inside borders)

Figure 8: left and right, border area mask and inside area mask respectively.

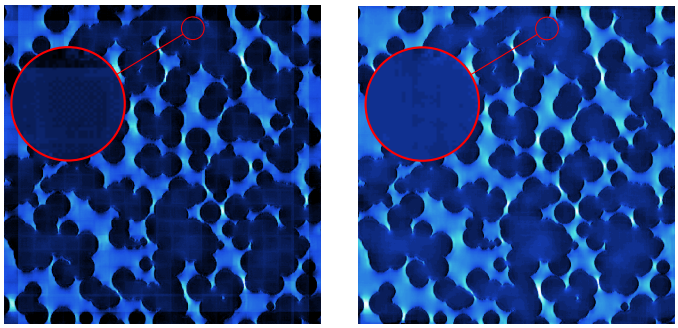
Images	Sampling	PSNR	UIQ	SSIM
Original	uniform	34.5997	0.6891	0.8305
	stratified	32.1624	0.6443	0.7820
MaskB	uniform	30.8096	0.6890	0.7777
MaskI	uniform	34.0679	0.6892	0.8137
Augmented	uniform	35.7380	0.7030	0.8803
	stratified	**37.9949**	**0.7627**	**0.9026**

Table 1: PSNR,UIQ and SSIM values comparison of predicted images in figures (7) and (8) stemming from original and augmented training datasets.

Shift values	-5	-4	-3	-2	-1	0	1	2	3	4	5
PSNR	26.6912	30.2654	28.1801	32.6074	29.8986	50	29.8177	33.4603	28.2784	30.2763	26.4302
UIQ	0.9523	0.9776	0.9645	0.9867	0.9766	1	0.97503	0.9859	0.9654	0.9744	0.9473
SSIM	0.9532	0.9780	0.9652	0.9869	0.9771	1	0.9755	0.9861	0.9661	0.9749	0.9484
BCE	0.2547	0.2516	0.2472	0.2440	0.2409	0.2401	0.2415	0.2448	0.2480	0.2516	0.2547

Table 2: PSNR, UIQ, SSIM mean values and BCE values evaluated between common area of original and shifted patches.

3.4. Effects of stratified sampling



(a) Uniform sampling (b) Stratified sampling

Figure 9: U-Net prediction results with uniform sampling (a) and stratified sampling (b).

The network processes patch by patch and yields accordingly a prediction for each one separately. To bring together the patches, a sampling strategy needs to be defined. Uniform and stratified sampling methods are investigated. In contrast to the first which consists of distributing the predicted patches uniformly over all the image size, the latter allocates random positions to each patch based on uniform sampled points. Visibly, stratified sampling reduces considerably the edge effect on the borders of patches, related results are shown in figure (9). A comparison between ground truth image and predicted results in terms of evaluation indices is shown in table (1). For augmented images, stratified sampling has proven its ability to improve results in terms of image evaluation metrics. On the other hand, predicted image from the original dataset has been reduced. We explain this by the fact that the evaluation metrics are being averaged over the entire image and does not reflect the visual feeling. Two masks isolating both the edge borders and the inside area of patches are considered. A local similarity comparison between former masks applied to figure (7- b) and to ground truth image has been conducted. As expected, PSNR, UIQ and SSIM indices for uniform sampling evaluated at the edge borders area are lower than prior evaluated indices in the case of stratified sampling. Figure (8) illustrates this operation and related results are shown in table (1).

4. Conclusion

A small sample learning methodology is proposed. It is based on the distance transform as data augmentation. The former provides intrinsic information to the original image that enhanced the learning process. A patch-based procedure has been applied to overcome the limited number of samples along with a stratified sampling method to remove border edge effect. The methodology was validated on a case study concerning the estimation of a grayscale image from a binary image using a FCN. Future work will focus on an application with distance transform augmentation applied to grayscale images as input.

Acknowledgment

The authors would like to thank Mohamed El Khamlichi for his contributions in a prior version of the used U-Net patch-based code.

References

[1] Y. Le Cun, Y. Bengio, G. Hinton, Deep learning, Nature 521 (7553) (2015) 436–444. doi:10.1038/nature14539.

[2] Y. Guo, Y. Liu, A. Oerlemans, S. Lao, S. Wu, M. S. Lew, Deep learning for visual understanding: A review, Neurocomputing 187 (2016) 27 – 48, recent Developments on Deep Big Vision. doi:https://doi.org/10.1016/j.neucom.2015.09.116.

[3] J. Shu, Z. Xu, D. Meng, Small sample learning in big data era, CoRR abs/1808.04572 (2018). arXiv:1808.04572.

[4] T. D. Kulkarni, W. F. Whitney, P. Kohli, J. Tenenbaum, Deep convolutional inverse graphics network, in: C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, R. Garnett (Eds.), Advances in Neural Information Processing Systems 28, Curran Associates, Inc., 2015, pp. 2539–2547.

[5] J. Yosinski, J. Clune, Y. Bengio, H. Lipson, How transferable are features in deep neural networks?, in: Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2, NIPS'14, MIT Press, Cambridge, MA, USA, 2014, p. 3320–3328.

[6] K. Nazeri, A. Aminpour, M. Ebrahimi, Two-stage convolutional neural network for breast cancer histology image classification, in: A. Campilho, F. Karray, B. ter Haar Romeny (Eds.), Image Analysis and Recognition, Springer International Publishing, 2018.

[7] D. C. Cireşan, A. Giusti, L. M. Gambardella, J. Schmidhuber, Deep neural networks segment neuronal membranes in electron microscopy images, in: Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 2, NIPS'12, Curran Associates Inc., Red Hook, NY, USA, 2012, p. 2843–2851.

[8] T. Araújo, G. Aresta, E. Castro, J. Rouco, P. Aguiar, C. Eloy, A. Polónia, A. Campilho, Classification of breast cancer histology images using convolutional neural networks, PLOS ONE 12 (6) (2017) 1–14. doi:10.1371/journal.pone.0177544.

[9] J. Neyman, On the two different aspects of the representative method: The method of stratified sampling and the method of purposive selection 97 (4) (1934) 558–625. doi:10.2307/2342192.

[10] G. Borgefors, Distance transformations in arbitrary dimensions, Computer Vision, Graphics, and Image Processing 27 (3) (1984) 321 – 345. doi:https://doi.org/10.1016/0734-189X(84)90035-5.

[11] N. Audebert, A. Boulch, B. L. Saux], S. Lefèvre, Distance transform regression for spatially-aware deep semantic segmentation, Computer Vision and Image Understanding 189 (2019) 102809. doi:https://doi.org/10.1016/j.cviu.2019.102809.

[12] W. Liu, Z. Wang, X. Liu, N. Zeng, Y. Liu, F. E. Alsaadi, A survey of deep neural network architectures and their applications, Neurocomputing 234 (2017) 11 – 26. doi:https://doi.org/10.1016/j.neucom.2016.12.038.

[13] F. Lateef, Y. Ruichek, Survey on semantic segmentation using deep learning techniques, Neurocomputing 338 (2019) 321 – 348. doi:https://doi.org/10.1016/j.neucom.2019.02.003.

[14] O. Ronneberger, P. Fischer, T. Brox, U-net: Convolutional networks for biomedical image segmentation, in: N. Navab, J. Hornegger, W. M. Wells, A. F. Frangi (Eds.), Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015, Springer International Publishing, Cham, 2015, pp. 234–241.

[15] T. Wang, Z. Ji, Q. Sun, S. Han, Combining pixel-level and patch-level information for segmentation, Neurocomputing 158 (2015) 13 – 25. doi:https://doi.org/10.1016/j.neucom.2015.02.010.

[16] Y. Le Cun, Learning invariant feature hierarchies, in: A. Fusiello, V. Murino, R. Cucchiara (Eds.), Computer Vision – ECCV 2012. Workshops and Demonstrations, Springer Berlin Heidelberg, Berlin, Heidelberg, 2012, pp. 496–505.

[17] I. Goodfellow, Y. Bengio, A. Courville, Deep Learning, MIT Press, 2016, `http://www.deeplearningbook.org`.

[18] R. Gens, P. M. Domingos, Deep symmetry networks, in: Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, K. Q. Weinberger (Eds.), Advances in Neural Information Processing Systems 27, Curran Associates, Inc., 2014, pp. 2537–2545.

[19] M. Jaderberg, K. Simonyan, A. Zisserman, k. kavukcuoglu, Spatial transformer networks, in: C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, R. Garnett (Eds.), Advances in Neural Information Processing Systems 28, Curran Associates, Inc., 2015, pp. 2017–2025.

[20] E. Kauderer-Abrams, Quantifying translation-invariance in convolutional neural networks, CoRR abs/1801.01450 (2018). arXiv:1801.01450.

[21] L. He, Y. Chao, K. Suzuki, A run-based two-scan labeling algorithm, in: M. Kamel, A. Campilho (Eds.), Image Analysis and Recognition, Springer Berlin Heidelberg, Berlin, Heidelberg, 2007, pp. 131–142.

[22] C. Pudney, Distance-based skeletonization of 3d images, in: Proceedings of Digital Processing Applications (TENCON '96), Vol. 1, 1996, pp. 209–214 vol.1.

[23] A. Criminisi, T. Sharp, C. Rother, P. P'erez, Geodesic image and video editing, ACM Trans. Graph. 29 (5) (Nov. 2010). doi:10.1145/1857907.1857910.

[24] I. S. Molchanov, P. Terán, Distance transforms for real-valued functions, Journal of Mathematical Analysis and Applications 278 (2) (2003) 472 – 484. doi:https://doi.org/10.1016/S0022-247X(02)00719-9.

[25] J. Chaniot, Efficient morphological characterization of materials using distance transforms, PhD thesis, (2019).

[26] P. J. Toivanen, New geodosic distance transforms for gray-scale images, Pattern Recognition Letters 17 (5) (1996) 437 – 450. doi:https://doi.org/10.1016/0167-8655(96)00010-4.

[27] Zhou Wang, A. C. Bovik, A universal image quality index, IEEE Signal Processing Letters 9 (3) (2002) 81–84.

[28] Z. Wang, A. C. Bovik, H. R. Sheikh, Structural similarity based image quality assessment, 2004.

[29] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization (2014). arXiv:1412.6980.

[30] G. Matheron, Random sets and integral geometry [by] G. Matheron, Wiley New York, 1974.

[31] J. F. C. Kingman, Poisson processes, Vol. 3 of Oxford Studies in Probability, The Clarendon Press Oxford University Press, New York, 1993, oxford Science Publications.

[32] M. Moreaud, J. Chaniot, T. Fournel, J. M. Becker, L. Sorbier, Multi-scale stochastic morphological models for 3d complex microstructures, 2018 17th Workshop on Information Optics (WIO) (2018) 1–3.

[33] D. Jeulin, M. Moreaud, Multi-scale simulation of random spheres aggregates - application to nanocomposites, Proc. 9th European Congress on Stereology and Image Analysis 1 (2005) 341–348.

[34] H. Moulinec, P. Suquet, A fft-based numerical method for computing the mechanical properties of composites from images of their microstructures, in: R. Pyrz (Ed.), IUTAM Symposium on Microstructure-Property Interactions in Composite Materials, Springer Netherlands, Dordrecht, 1995, pp. 235–246.

[35] H. Moulinec, P. Suquet, A numerical method for computing the overall response of nonlinear composites with complex microstructure, Computer Methods in Applied Mechanics and Engineering 157 (1) (1998) 69 – 94.

[36] D. Jeulin, M. Moreaud, Statistical representative volume element for predicting the dielectric permittivity of random media, in: D. Jeulin, S. Forest (Eds.), 11th International Symposium on Continuum Models and Discrete Systems CMDS 11, Sciences de la matière, Presses des mines, Paris, France, 2007, pp. 429–436, iSBN : 078-2-35671-000-0.

[37] D. J. Eyre, G. W. Milton, A fast numerical scheme for computing the response of composites using grid refinement, The European Physical Journal - Applied Physics 6 (1) (1999) 41–47. doi:10.1051/epjap:1999150.

[38] "plug im!" an open access and customizable software for signal and image processing (2020).
URL `https://www.plugim.fr`