MATHEMATICAL MODELING AND COMPUTING, Vol. 10, No. 3, pp. 720–732 (2023)

Enhancing image inpainting through image decomposition and deep neural networks

Bellaj K., Benmir M., Boujena S.

Fundamental and Applied Mathematics Laboratory,
Department of Mathematics and Computer Sciences,
Ain Chock Science Faculty,
Km 8 Route El Jadida POB 5366 Maarif, Casablanca, Morocco

(Received 17 February 2023; Revised 17 July 2023; Accepted 18 July 2023)

A new approach to inpainting problems that combines domain decomposition methods (DDM) with deep neural networks (DNN) to solve partial differential equations (PDE) is presented. First, this article examines different existing and emerging approaches to inpainting while emphasizing their advantages and disadvantages in a unified framework. After that, we introduce an algorithm that highlights the combination of DDM and DNN techniques for solving PDEs of a proposed mathematical inpainting model. For this model, the modified approach that has been adopted uses the DNN method which is based on convolutional neural networks (CNN) to reduce the computational cost in our algorithm while maintaining accuracy. Finally, the experimental results show that our method significantly outperforms existing ones for high-resolution images in paint stains.

Keywords: *deep neural; domain decomposition methods; inpainting; machine learning; partial differential equations.*

2010 MSC: 62H35, 68U10, 94A08, 97U50 **DOI:** 10.23939/mmc2023.03.720

1. Introduction

Inpainting is the process of filling in missing or corrupted parts of an image or video [1, 2]. It can be used for a variety of applications, such as restoring old photographs, removing unwanted objects from images, and filling in missing parts of videos. There are several methods for inpainting, including:

1. Patch-based methods: These methods use small patches of known image data to fill in the missing parts. The quality of the results obtained by patch-based methods depends on the size of the patch and the method used to calculate the similarity between patches. Patch-based methods are relatively fast and can produce good results for small missing regions, but they can be less effective for larger missing regions or images with complex textures. Some popular Patch-based methods are Criminisi et al. [3], Exemplar-based Inpainting, etc. These methods are widely used in the research community and are known to produce good results for small missing regions.
2. Neural network-based methods are a type of inpainting technique that use deep learning techniques to generate the missing parts of an image. These methods typically involve training a neural network to learn a mapping between the known and missing parts of an image. Once trained, the network can be used to generate new, plausible pixels to fill in the missing regions. Neural network-based methods can produce high-quality results, even for large missing regions or images with complex textures. However, they require large amounts of training data and can be computationally intensive to train. Additionally, the quality of the results can be affected by the choice of architecture and the quality of the training data.
3. Example-based methods: These methods use small patches of an image known dataset to fill in the missing parts. Example-based methods can produce high-quality results, especially for images with complex textures. However, they require a large set of example images, and the quality of the results can be affected by the similarity between the input image and the example images.

© 2023 Lviv Polytechnic National University

Additionally, they may not work well if the missing region is large and is not similar to any part of the example image.

4. PDE-based methods are mathematical models of inpainting that use partial differential equations to propagate information from known parts of the image to fill in the missing or corrupted parts. These methods are based on the idea that the pixels in an image change over time, and the PDE describes how this change occurs. There are several types of PDE-based methods for inpainting, some of the popular ones are:
   — Linear Diffusion which uses a linear diffusion equation to describe the smoothing of an image over time. The idea is to propagate information from the known parts of the image to the missing regions by smoothing the image.
   — Nonlinear Diffusion: This method uses a nonlinear diffusion equation, which describes the smoothing of an image over time in a way that takes into account the image's structure and texture. The nonlinear diffusion equation allows for more flexibility in the inpainting process, as it can adapt to different types of images and missing regions.
   — Level set methods which is a contour-based representation of the image, and the inpainting process is formulated as a PDE-based evolution of the contour. The level set method is applied to fill in the missing regions by evolving the contour toward the desired solution.

   PDE-based methods can produce high-quality results, especially for images with complex textures. They are typically iterative, with the PDE being solved multiple times until the missing parts of the image are filled in. However, these methods can be computationally intensive and require a good initialization of the known parts of the image to produce good results.

The choice of method will depend on the type of image, the extent of the missing or corrupted parts, and the desired result. Inpainting methods continue to improve with the advancements in deep learning and computer vision.

Inpainting using a nonlinear diffusion model involves solving a PDE that describes how the pixels in an image change over time. The PDE is designed to propagate information from known parts of the image to fill in missing or corrupted parts. The nonlinearity in the diffusion equation allows for more flexibility in the inpainting process, as it can take into account the image's structure and texture.

In this method, the diffusion process is controlled by a function, which is known as the Diffusivity function. This function allows for different amounts of smoothing to be applied to different parts of the image, depending on the image's structure.

There are different variations of non-linear diffusion models like Perona-Malik model (PM), Curvature-driven Diffusion model, etc. These models are used based on the type of image and the extent of missing or corrupted parts of the image.

1.1. Contributions

This work extends our existing efforts for inpainting problems [4]. We establish a combination of the finite element method with machine learning techniques to solve PDE for inpainting problems. We also present an updated approach based on range decomposition methods and other optimizations, including steps 3 and 10 in algorithm 1, which was developed originally for high-resolution images.

1.2. Paper organization

The rest of this paper is organized as follows. In Section 2, we present a modified PM model for image inpainting. We then briefly described the discretization of the proposed modified PM model. In Section 3, the details of the proposed domain decomposition methods are described. We also discuss how DDMs and DNN can be combined with our inpainting techniques to optimize the desired results, the experimental results comparing our strategy with other advanced methods, and the related experiments are analyzed in Section 4. In Section 5, conclusions are drawn.

2. Inpainting problem's model

2.1. Perona–Malik equation

Inpainting with Perona–Malik equation involves using the nonlinear diffusion equation to propagate information from known parts of the image to fill in missing or corrupted parts [7,8]. The PM equation is a type of PDE-based method that is designed to preserve edges and textures in the image while smoothing out noise.

The PM equation is defined as:

$$\frac{\partial u}{\partial t} = \operatorname{div}\big(\rho(|\nabla u|)\nabla u\big),$$

where u is the image, t is time, ρ is a scalar function, also known as the diffusivity function, div is the divergence operator, and ∇u is the gradient of the image. To perform inpainting with Perona–Malik equation, a binary mask is typically used to define the known and missing regions of the image.

The equation is then solved using the known image data and the mask as initial conditions. The solution to the equation will propagate information from the known parts of the image to the missing regions, by filling in the missing pixels. The diffusivity function ρ plays a crucial role in the inpainting process. It is defined such that it is high in regions of high gradient and low in regions of low gradient. This helps to preserve the edges and textures in the image while smoothing out the missing regions. The Perona–Malik equation is a powerful tool for inpainting images with complex structures and textures. However, the algorithm can be computationally intensive and requires good initialization of the known parts of the image to produce good results. It should be noted that there are several other mathematical models by nonlinear partial differential equations for image processing which have recently been derived from that of Perona and Malik and which have made it possible to improve their performance [4–6]. We will focus in the following on one of these models to build our powerful algorithm.

2.2. The proposed mathematical model

Let u_0 a given damaged image occupying a domain $\Omega \subset \mathbb{R}^2$. Our proposed mathematical model for constructing the inpainting image u is defined as follows:

$$\begin{cases} \dfrac{\partial u}{\partial t} - \lambda \operatorname{div}\big(\rho(|\nabla u|)\nabla u\big) = 0, \ \text{ in } \Omega, \ \text{ for } t > 0, \\ u(x,0) = u_0(x), \forall x \in \Omega, \\ \partial u_\nu(x,t) = 0, \ \forall x \in \partial\Omega, \ \forall t > 0, \end{cases} \tag{1}$$

where the diffusion function ρ corresponds to the one used in the modified Perona–Malik model, more details can be found in [4].

This model has been successfully used in image processing to fill missing regions [4]; however, its main drawback is that it does not give a good result for images with larger sizes [9], we have coupled the DDM to the finite element method for this model and we obtained better results for high-resolution results for high resolution images. In this work, we introduce DNN within the DDM especially for the finite element method to ameliorate the accuracy and control the choice of the subdomains number.

2.3. Discretization of the proposed problem

In the following, we present the discretization of the proposed modified PM model (1) in order to solve this parabolic nonlinear problem approximately. After its discretization by using the Galerkin finite element method, we obtain a well-posed linear algebra problem.

By $V_h = \big\{w \in C^0(\Omega), w|_\Sigma \in P^1\big\}$ we denote the approximation space, where Σ is a partition of Ω.

The weak form of Perona–Malik equation using the Galerkin finite element formulation consists in finding such a function $v_h \in V_h$ that:

$$\int_\Omega \frac{\partial v_h(t)}{\partial t} w \, dx + \int_\Omega \lambda \, \rho\left(|\nabla v_h|\right) \nabla v_h \cdot \nabla w \, dx = 0, \tag{2}$$

for all $w \in V_h$ and $t \in [0,T]$.

Let $0 = t_0 < t_1 < t_2 < \ldots < t_{n+1} = T$ be a subdivision of $[0,T]$ with uniform scaling interval $\Delta t = t_n - t_{n-1}$ for some $n > 0$.

The backward Euler scheme is considered for (2) in the time discretization, and we formulate the nonlinear coefficient $\rho(|\nabla v_h|)$ by using the previous scale step value v_h^n. Thus, the discrete equation is

$$\frac{1}{\Delta t}\int_\Omega (v_h^{n+1} - v_h^n) w\, dx + \int_\Omega \lambda\, \rho\left(|\nabla v_h^n|\right) \nabla v_h^{n+1} \nabla w\, dx = 0, \tag{3}$$

for all $w \in V_h$ and $n > 0$. Assume an orthonormal basis of the finite-dimensional space V_h is $(\varphi_1, \ldots, \varphi_p)$, $p > 0$.

Taking $w = \varphi_j$ for $j = 1, 2, \ldots, p$ and using the representation $v_h^{n+1}(x) = \sum_{i=0}^{p} v_i^{n+1}\varphi_i(x)$, where v_i^{n+1} are unknown, equation (3) can be transformed into the discrete linear equation is expressed as

$$(A + \Delta t \lambda B) U^{n+1} = A U^n, \tag{4}$$

where

$$U^n = \begin{pmatrix} v_1^n \\ \vdots \\ v_p^n \end{pmatrix}, \quad A_{ij} = \int_\Omega \varphi_i \varphi_j dx, \quad B_{ij} = \int_\Omega \rho\left(|\nabla v^n|\right) \nabla\varphi_i \cdot \nabla\varphi_j dx.$$

A is a so-called mass matrix and B is the stiffness matrix. Thus, the discrete solution can be found efficiently by so-called conjugate gradient methods.

Note that the existence and uniqueness of the solution of (1) in Hilbert space are proved under the following hypothesis [10]:

1. $\rho\colon R^+ \to R^+$ is a decreasing function,
2. ρ is continuous differentiable,
3. $\rho(0) = 1 + \alpha$,
4. $\lim\limits_{s \to +\infty} \rho(s) = \alpha$,
5. $2s\,|g'(s)| \leqslant \rho(s)$, $s \geqslant 0$,

where $\alpha > 0$ is a positive parameter (generally very smaller) added to the diffusion function ρ_{pm} of PM to ensure the monotonicity of the differential operator associated with the problem (1) without changing the numerical performances of the PM model. In the following we take ρ as follows:

$$\rho(s) = \frac{1}{\sqrt{1 + (\frac{s}{k})^2}} + \alpha,$$

where $\alpha = 10^{-6}$ and $k > 0$ is a given parameter, for more details see [10].

3. The proposed upgrade technic for the calculation of solutions

3.1. Domain decomposition methods

Inpainting with the domain decomposition method involves partitioning the image into smaller regions or subdomains [11–13], and then solving for the missing or corrupted parts within each subdomain, (see Figure 1). This method can be useful when the image has distinct regions with different characteristics, such as texture or color. By solving for each subdomain separately, the inpainting process can take into account the specific characteristics of that region.

The domain decomposition method typically involves a two-step process: first, the image is divided into subdomains, and then the inpainting is performed within each subdomain. There are different ways to divide the image into subdomains, such as using a grid, clustering similar pixels, or using a level set method. Once the subdomains are defined, the inpainting process can be performed using various techniques, such as patch-based methods, PDE-based methods, or neural network-based methods. The choice of method will depend on the characteristics of the subdomain and the desired result, as mentioned in [4], for an arbitrary number of subdomains or an image with complex geometry, the standard DDM is no longer advantageous.

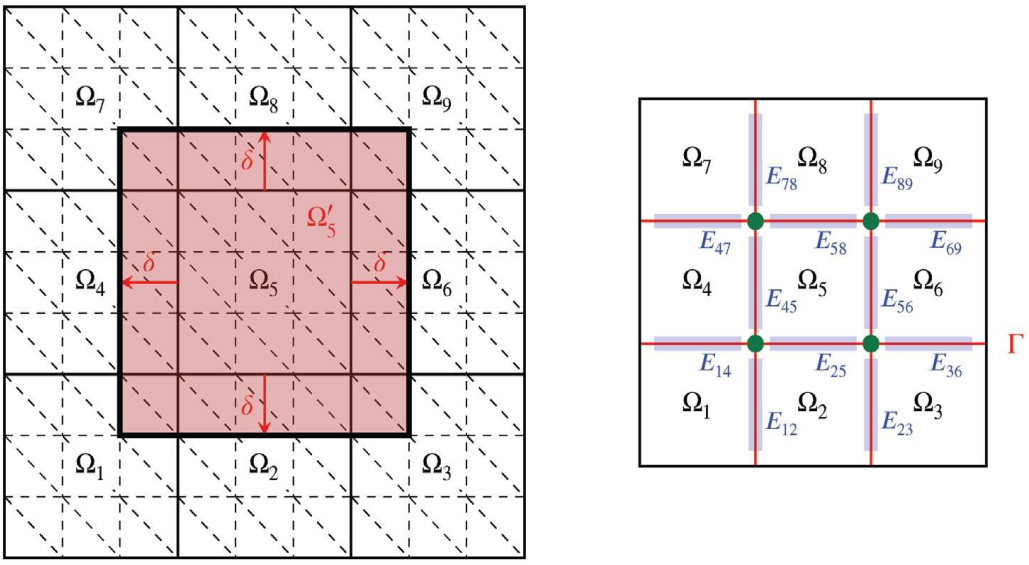


Fig. 1. Decomposition of the domain $\Omega \subset \mathbb{R}^2$ into nine non-overlapping subdomains Ω_i, $i = 1, \ldots, 9$. [Left] The decomposition of the overlapping domain is obtained by extending the non-overlapping subdomains by several layers of δ-width elements. The overlapping subdomain Ω_5 is marked in red. (Right) The Γ interface is indicated in red, the vertices in green, and the borders in blue.

In the following, an efficiency optimization method is proposed to enhance the performance by combining the DDM with the DNN algorithm, (see Figures 5 and 6 and 7).

Some common types of decomposition methods include:

1. Domain decomposition: This method divides a domain, such as an image or a piece of text, into smaller subdomains, each representing a specific feature or structure [14].
2. Functional decomposition: This method divides a complex function or system into smaller, simpler functions or sub-systems [15].
3. Matrix decomposition: This method decomposes a matrix into simpler matrices, such as eigenvectors and eigenvalues, which can make it easier to solve linear algebra problems [16].
4. Optimization decomposition: This method breaks down an optimization problem into smaller subproblems that can be solved independently and then combined to find the global solution [17].
5. Alternating Direction Method of Multipliers (ADMM): This method is a popular optimization algorithm that decomposes a large optimization problem into smaller sub-problems that can be solved in parallel, which improves computational efficiency [18].

Decomposition methods can be used to simplify complex problems and make them more tractable, by breaking them down into smaller, more manageable sub-problems. They can be used on their own or in combination with other techniques such as DNN.

3.2. Deep neural networks

DNNs are a type of machine learning algorithm that is modeled after the structure and function of the human brain. They are composed of layers of interconnected nodes, known as artificial neurons, which process and transmit information. Each layer in a DNN is responsible for extracting a different set of features from the input data, and the layers are connected in a way that allows the network to learn increasingly complex representations of the data.

DNNs are trained using large amounts of labeled data, which is used to adjust the weights of the connections between the neurons. Once trained, the DNN can be used to make predictions or classifications on new, unseen data. DNNs have been used in a wide range of applications, such as image and speech recognition, natural language processing, and self-driving cars. They have also been used to achieve state-of-the-art results in various fields such as computer vision, natural language

processing, and speech recognition. CNN and Recurrent Neural Networks (RNN) are some examples of DNNs. These networks have been very successful in various tasks such as image classification, object detection, language translation, speech recognition, and many more. DNNs are computationally intensive, requiring powerful hardware and specialized libraries to implement and train, but the results they produce are well worth the effort.

The decision for a specific number of subdomains Ω_i for the DDM is clearly a classification problem, i.e., a machine learning problem. In particular, the decision is purely based on local information, that is, the distribution of the pixels within two neighboring subdomains. To approximate a solution for subdomain classification, we use dense feed-forward neural networks or, more precisely, multilayer perceptrons [19–22]. The proposed strategy consists in train DNN to automatically and in advance perform this decision about either the number and geometry of subdomains in other words in a preprocessing phase before ever solving the problem (1).

3.3. Proposed Method for the image inpainting task

In order to maintain the load between the number of subdomains and the complex geometry of the image in equilibrium, the DDM parameters must be selected correctly. For example, in our algorithm, we used a dense neural network to predict for which edges the eigenvalue problem is required or not before resolving it. Our proposed approach is to form a neural network to take this decision automatically and in advance, that is to say in the pretreatment phase, even before solving the problems in local subdomains. Obviously, there is great potential to save the cost of computing and to make adaptive DDM more competitive by omitting our problem on certain subdomains which are unnecessary because they do not add any significant stress to the coarse problem. Indeed, for many practical and realistic modeling problems, only a small number of adaptation constraints are required to maintain a robust algorithm. Another important aspect of our contribution is that we use parallel and sequential DDMs to process images in high resolution. These improvements can be summarized as follows.

Algorithm 1

1: Data preparation: Collect and preprocess a dataset that the task is to be applied on.
2: Domain decomposition: Divide the task into smaller sub-tasks or subdomains, each representing a specific feature or structure.
3: Model design: Choose a deep learning model architecture, such as a CNN, a recurrent neural RNN or a transformer-based model that is suitable for the sub-task or subdomain.
4: Model training: Train the deep learning model on the subdomains separately. This step involves adjusting the DDM's parameters to optimize the performance of the model on the sub-task.
5: Model evaluation: Evaluate the performance of the trained model on a set of validation data. This step can be used to fine-tune the model's parameters and to check the quality of the generated results.
6: Task execution: Apply the trained models on the subdomains to generate the results.
7: Result aggregation: Combine the results generated by the trained models on the subdomains to form the final output.
8: Post-processing: Depending on the task, some post-processing steps such as denoising, data fusion, or feature extraction may be necessary.
9: Inpainting: Apply improved DDM to equation (4) to generate completed images.
10: Image reconstruction: Combine the generated subdomains to form the final completed image.
11: Post-processing: Depending on the task, some post-processing steps such as denoising, color correction, and texture synthesis may be necessary.

It is important to note that this approach is useful for handling large images or images with complex structures, where a single deep learning model may not be able to capture all the necessary details. It also allows the model to focus on specific areas of an image which requires more attention, which can improve the overall performance of the inpainting (see Figure 2).

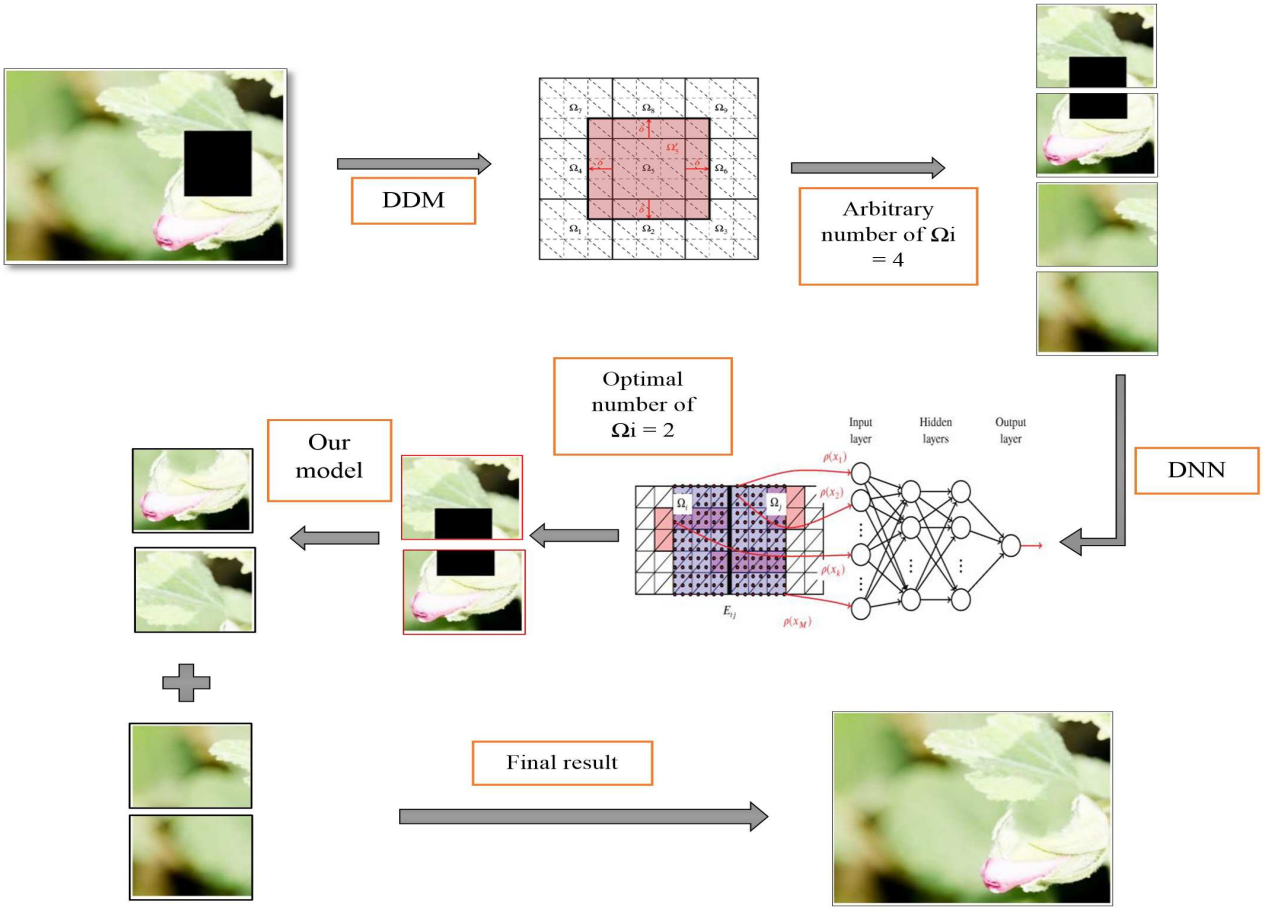


Fig. 2. An overview of our proposed framework utilized for the image inpainting.

4. Experimental results

4.1. Datasets and training setting

In this section, we evaluate the performance of the proposed framework quantitatively, and compare it with other methods. Two well-known benchmark datasets, ImageNet [23, 24] and CelebA-HQ [25, 26] are used to evaluate the performances of our proposed approach. For training purposes, we consider the following number of images, i.e. 328 500 and 30 000 from the aforementioned datasets, respectively. A batch size 48 is chosen and Adam optimizer is used to train our model with a learning rate of 0.0001. All the experiments were performed using Tensorflow 2.11 and Python 3.10.6 on a computer equipped with an Intel Core (TM) i7-8700 processor 3.900 GHz, up to 4.90 GHz with Turbo Boost, 52Gb RAM, and Windows 11. To train the network we used a DDM with default parameters. To measure the performance of the proposed method, we used the peak-signal-to-noise ratio (PSNR) providing information from different perspectives. The parameters of the model are $\Delta t = 0.24$, $dx = dy = 1$, $\alpha = 10^{-6}$ and $k = \frac{1}{4}$.

4.2. Qualitative comparisons

As the images have variable sizes (from 256x104 to 2012x2512 here), we decided to divide them into subdomains of varying sizes with a minimum overlap of 10x10. We apply DNN to each subdomain, then we average the outputs (see Figure 2). Our DNN has 5 convolutional layers and a fully connected layer with eleven outputs. In the loss function, regression to overall quality and focal loss are combined for each of the 10 presence indicators. Our machine learning model has been constructed after a preset number of training epochs which has been determined experimentally to insure convergence. We used,

for this, Adam optimizer and an exponential learning rate schedule. To facilitate the comparative evaluation of our proposal with the current state of the art, we show in Figure 3 the results of inpainting using the approach described in algorithm 1 for all types of masks (small, large, irregular). While the masks in Figure 4 mainly locate in one overlap region, the masks in Figure 3 locate in various sections in the foreground, which poses a major challenge for high-quality reconstruction of these missing regions. Based on the examples shown in Figures 4, 3, we can conclude that our method is clearly superior to the state of the art and has less noticeable mismatches in most cases. In particular, in Figure 3, with various holes in image, our method can produce a more natural and semantically plausible reconstruction, which confirms the fact that our approach to inpainting problems that combines DDM with DNN can not only explore the detailed information of the surrounding pixels but also capture the global structure of the whole image to be painted over.

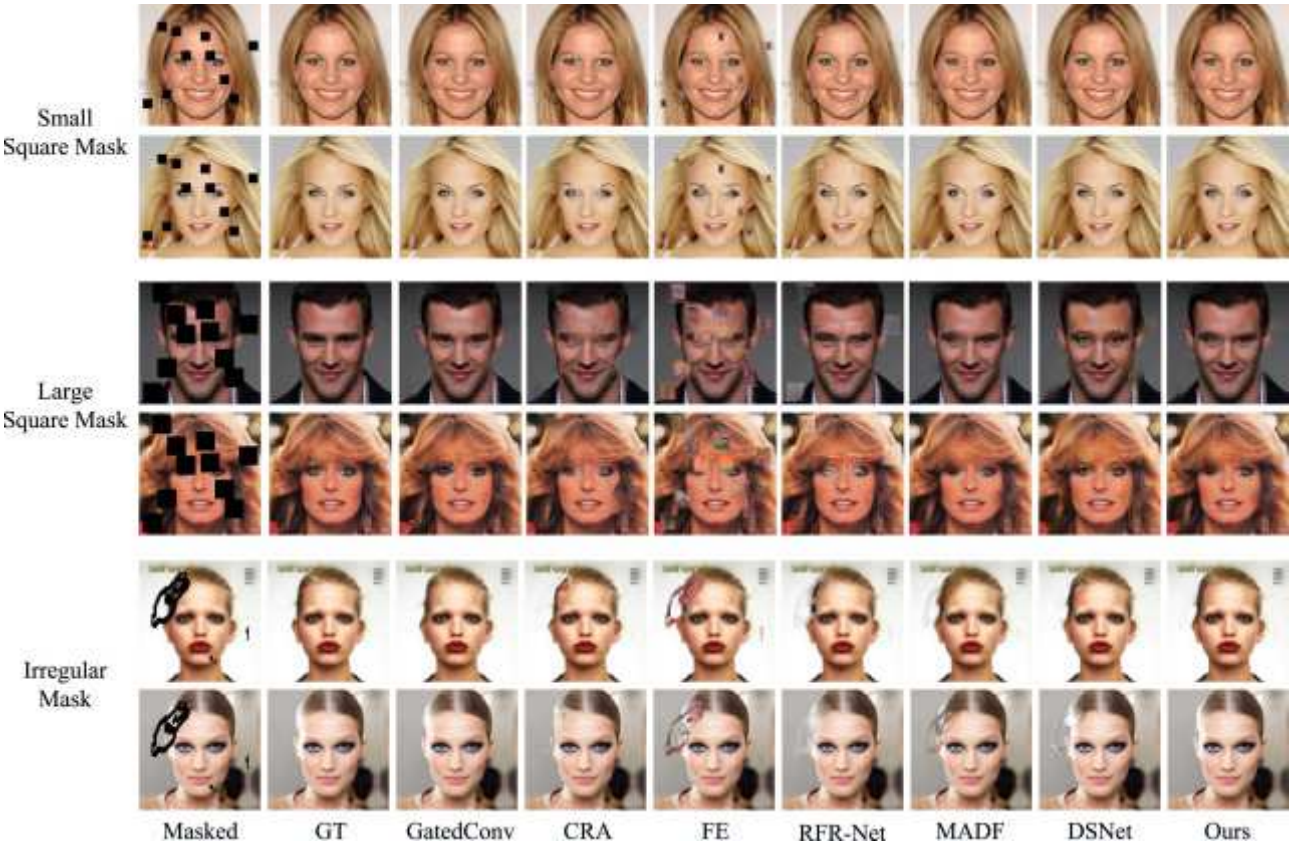


Fig. 3. Qualitative comparison results of inpainting irregular holes (small, large, irregular) on ImageNet.

It can be also seen that there exist inconsistent contents of images generated by GatedConv when inpainting images with complex structures (see c1 and c2 in Figure 4), this could be due to a number of factors, such as a lack of sufficient training data, poor network architecture, or suboptimal training parameters. And it can also be seen that the results generated by RFR-Net also have a blurring phenomenon (see e3 in Figure 4). Our proposed model can recover images with reasonable semantics, visual realism, clear texture, and consistent context (see g3 and g4 in Figure 4).

4.3. Quantitative comparison

To further demonstrate the effectiveness of our method, we perform a quantitative comparison between our method and other state-of-the-art methods as shown in Table 1. Following related image over-painting approaches, we use common evaluation metrics such as PSNR and structural similarity (SSIM) to measure the similarity between the over-painting result and the corresponding ground truth. However, the image over-painting methods are intended to focus on obtaining realistic results

Fig. 4. Comparative illustration of inpainting results achieved by our proposed and the existing benchmarks on ImageNet: (a) original image, (b) input, (b') image with DDM (subdomains with a red star are chosen by our strategy), (c) GatedConv [27], (d) CRA [28], (e) RFR-Net [29], (g) Algorithm 1.

rather than approximating the ground truth. For this reason, we also use learned perceptual image patch similarity (LPIPS) and Frechet input distance (FID) as metrics of perceptual quality that are consistent with human judgment. It can be observed that the metric values of our method significantly outperform these state-of-the-art methods. As shown in Table 1, the values of PSNR and SSIM obtained with our method are almost maximum, indicating that the image distortion obtained with our method is minimum and the repair effect is best. Our method almost achieves the minimum values of LPIPS and FID, which shows the highest quality of image generation by our method. This means that our method can better solve the problem of image distortion in larger dimensions.

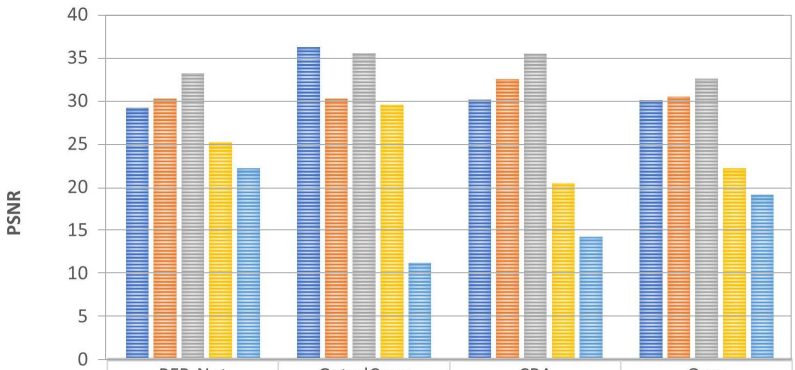
Table 1. Quantitative comparisons on CelebA-HQ and ImageNet datasets.

Method	Mask ratio	PSNR	SSIM	LPIPS	FID
GatedConv [27]	0%–10%	38.887	0.975	0.524	7.804
CRA [28]		34.296	0.975	0.527	14.563
FE [27, 30]		27.682	0.946	0.516	114.583
RFR-Net [29]		32.252	0.948	0.509	106.312
MADF [31]		37.005	0.976	0.480	12.234
DSNet [32]		32.860	0.954	0.480	18.055
Ours		32.860	0.954	0.480	18.055
GatedConv [27]	10%–20%	32.708	0.962	0.524	18.049
CRA [28]		28.602	0.938	0.534	39.253
FE [30]		24.213	0.886	0.531	177.660
RFR-Net [29]		28.999	0.922	0.511	88.637
MADF [31]		32.403	0.956	0.482	20.106
DSNet [32]		32.860	0.954	0.480	18.055
Ours		32.860	0.954	0.480	18.055
GatedConv [27]	20%–30%	29.616	0.932	0.525	29.484
CRA [28]		25.345	0.882	0.550	90.395
FE [30]		21.212	0.817	0.545	218.986
RFR-Net [29]		25.750	0.864	0.531	160.998
MADF [31]		29.589	0.937	0.500	38.178
DSNet [32]		32.860	0.954	0.480	18.055
Ours		32.860	0.954	0.480	18.055
GatedConv [27]	Irregular	33.317	0.966	0.523	25.229
CRA [26, 28]		32.614	0.960	0.524	41.758
FE [30]		27.682	0.946	0.516	114.583
RFR-Net [29]		34.470	0.967	0.480	26.150
MADF [31]		32.860	0.954	0.480	18.055
DSNet [32]		32.860	0.954	0.480	18.055
Ours		32.860	0.954	0.480	18.055

In Figures 5–7, we performed a quantitative comparison of the different number of subdomains in the ImageNet dataset to show the effects of DDM on the complementation results. For each category, 1k images with different number of subdomains (2*2, 4*4, ...) were selected and their PSNR was calculated. The experiments show that the PSNR of our method is better than that of the compared methods for a mask of $0 - 10\%$. At a mask size of $10 - 20\%$, the structural similarity between our method and the other compared methods is not significant; at a mask size of $20 - 30\%$ of the image, our method achieves the highest structural similarity. This proves the effectiveness of our approach.

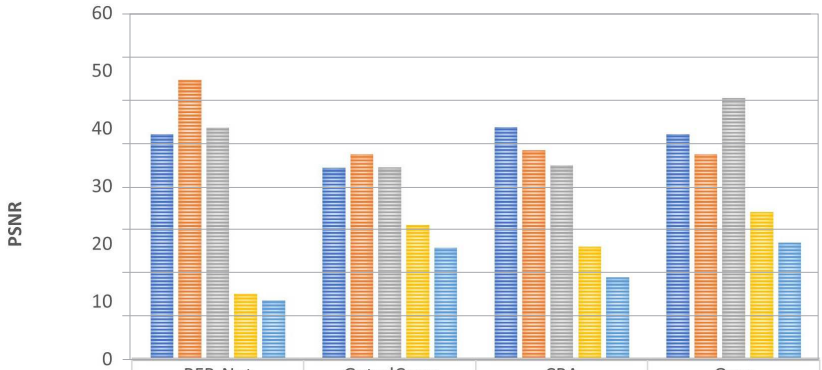
5. Conclusion

We have proposed in this paper a fast algorithm for a nonlinear model in image inpainting with specific applications of DNN and DDM. Most importantly, the proposed algorithm is a useful alternative for practical 2D data processing, especially for large sizes when the finite element method is used.



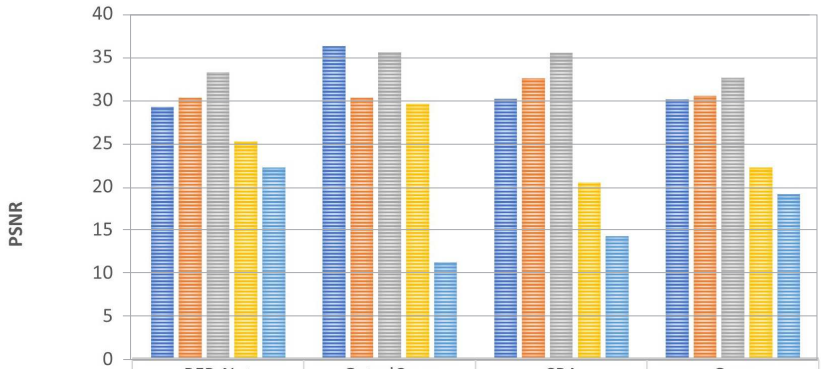
	RFR-Net	GatedConv	CRA	Ours
PSNR (sub-domains=1)	29,254	36,254	30,147	30,1
PSNR (sub-domains=4)	30,245	30,252	32,548	30,48
PSNR (sub-domains=16)	33,245	35,548	35,485	32,584
PSNR (sub-domains=64)	25,215	29,548	20,458	22,225
PSNR (sub-domains=100)	22,215	11,254	14,269	19,145

Fig. 5. Analysis of PSNR in comparison with existing methods for mask size of $0 - 10\%$.



	RFR-Net	GatedConv	CRA	Ours
PSNR (sub-domains=1)	39,027	33,256	40,258	39,044
PSNR (sub-domains=4)	48,47	35,588	36,259	35,584
PSNR (sub-domains=16)	40,22	33,269	33,698	45,28
PSNR (sub-domains=64)	11,369	23,269	19,584	25,589
PSNR (sub-domains=100)	10,258	19,365	14,269	20,269

Fig. 6. Analysis of PSNR in comparison with existing methods for mask size of $10 - 20\%$.



	RFR-Net	GatedConv	CRA	Ours
PSNR (sub-domains=1)	29,254	36,254	30,147	30,1
PSNR (sub-domains=4)	30,245	30,252	32,548	30,48
PSNR (sub-domains=16)	33,245	35,548	35,485	32,584
PSNR (sub-domains=64)	25,215	29,548	20,458	22,225
PSNR (sub-domains=100)	22,215	11,254	14,269	19,145

Fig. 7. Analysis of PSNR in comparison with existing methods for mask size of $20 - 30\%$.

[1] Elharrouss O., Almaadeed N., Al-Maadeed S., Akbari Y. Image inpainting: A review. Neural Processing Letters. **51** (2), 2007–2028 (2020).

[2] Bertalmio M., Sapiro G., Caselles V., Ballester C. Image inpainting. Proceedings of the 27th annual conference on Computer graphics and interactive techniques. 417–424 (2000).

[3] Criminisi A., Shotton J., Konukoglu E. Decision forests: A unified framework for classification, regression, density estimation, manifold learning and semi-supervised learning. Foundations and trends® in computer graphics and vision. **7** (2–3), 81–227 (2012).

[4] Boujena S., Bellaj K., El Guarmah E. M., Gouasnouane O. An improved nonlinear model for image inpainting. Applied Mathematical Sciences. **9** (124), 6189–6205 (2015).

[5] Ben-Loghfyry A., Hakim A. Time-fractional diffusion equation for signal and image smoothing. Mathematical Modeling and Computing. **9** (2), 342–350 (2022).

[6] Gouasnouane O., Moussaid N., Boujena S., Kabli K. A nonlinear fractional partial differentiation equation for image inpainting. Mathematical Modeling and Computing. **9** (3), 536–546 (2022).

[7] Kichenassamy S. The Perona–Malik paradox. SIAM Journal on Applied Mathematics. **57** (5), 1328–1342 (1997).

[8] Voci F., Eiho S., Sugimoto N., Sekibuchi H. Estimating the gradient in the Perona–Malik equation. IEEE Signal Processing Magazine. **21** (3), 39–65 (2004).

[9] Bellaj K., Boujena S., El Guarmah E. M., Gouasnouane O. One approach for image denoising based on finite element method and domain decomposition technique. International Journal of Applied Physics and Mathematics. **7** (2), 141–147 (2017).

[10] Boujena S., Pousin J., El Guarmah E. M., Gouasnouane O. An improved nonlinear model for image restoration. Pure and Applied Functional Analysis. **2** (4), 599–623 (2017).

[11] Kharazmi E., Zhang Z., Karniadakis G. E. hp-VPINNs: Variational physics-informed neural networks with domain decomposition. Computer Methods in Applied Mechanics and Engineering. **374**, 113547 (2021).

[12] Firsov D., Lui S. H. Domain decomposition methods in image denoising using Gaussian curvature. Journal of Computational and Applied Mathematics. **193** (2), 460–473 (2006).

[13] Smith B. F. Domain decomposition methods for partial differential equations. Parallel Numerical Algorithms. 225–243 (1997).

[14] Chan T. F., Mathew T. P. Domain decomposition algorithms. Acta Numerica. **3**, 61–143 (1994).

[15] Van Eck D., McAdams D. A., Vermaas P. E. Functional decomposition in engineering: a survey. International Design Engineering Technical Conferences and Computers and Information in Engineering Conference. 227–236 (2007).

[16] Mahoney M. W., Drineas P. CUR matrix decompositions for improved data analysis. Proceedings of the National Academy of Sciences. **106** (3), 697–702 (2009).

[17] Wang Z., He G., Du W., Zhou J., Han X., Wang J., He H., Guo X., Wang J., Kou Y. Application of parameter optimized variational mode decomposition method in fault diagnosis of gearbox. IEEE Access. **7**, 44871–44882 (2019).

[18] Han D.-R. A survey on some recent developments of alternating direction method of multipliers. Journal of the Operations Research Society of China. **10** (1), 1–52 (2022).

[19] Kelleher J. D. Deep learning. MIT Press (2019).

[20] Alaa K., Atountiand M., Zirhem M. Image restoration and contrast enhancement based on a nonlinear reaction-diffusion mathematical model and divide & conquer technique. Mathematical Modeling and Computing. **8** (3), 549–559 (2021).

[21] Alaa H., Alaa N., Aqel F., Lefraich H. A new Lattice Boltzmann method for a Gray–Scott based model applied to image restoration and contrast enhancement. Mathematical Modeling and Computing. **9** (2), 187–202 (2022).

[22] Alaa N., Alaa K., Atounti M., Aqel F. A new mathematical model for contrast enhancement in digital images. Mathematical Modeling and Computing. **9** (2), 342–350 (2022).

[23] Pintor M., Angioni D., Sotgiu A., Demetrio L., Demontis A., Biggio B., Roli F. ImageNet-Patch: A dataset for benchmarking machine learning robustness against adversarial patches. Pattern Recognition. **134**, 109064 (2023).

[24] Li D., Ling H., Kim S. W., Kreis K., Fidler S., Torralba A. BigDatasetGAN: Synthesizing ImageNet with Pixel-wise Annotations. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 21330–21340 (2022).

[25] Prabhu V. U., Yap D. A., Wang A., Whaley J. Covering up bias in CelebA-like datasets with Markov blankets: A post-hoc cure for attribute prior avoidance. ArXiv preprint arXiv:1907.12917 (2019).

[26] Zhu H., Wu W., Zhu W., Jiang L., Tang S., Zhang L., Liu Z., Loy C. C. CelebV-HQ: A large-scale video facial attributes dataset. European Conference on Computer Vision. 650–667 (2022).

[27] Xie K., Gao L., Lu Z., Li C., Xi Q., Zhang F., Sun J., Lin T., Sui J., Ni X. Inpainting the metal artifact region in MRI images by using generative adversarial networks with gated convolution. Medical Physics. **49** (10), 6424–6438 (2022).

[28] Yi Z., Tang Q., Azizi S., Jang D., Xu Z. Contextual residual aggregation for ultra high-resolution image inpainting. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 7508–7517 (2020).

[29] Li J., Wang N., Zhang L., Du B., Tao D. Recurrent feature reasoning for image inpainting. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 7760–7768 (2020).

[30] Zhao H., Kong X., He J., Qiao Y., Dong C. Efficient image super-resolution using pixel attention. European Conference on Computer Vision. 56–72 (2020).

[31] Zhu M., He D., Li X., Li C., Li F., Liu X., Ding E., Zhang Z. Image inpainting by end-to-end cascaded refinement with mask awareness. IEEE Transactions on Image Processing. **30**, 4855–4866 (2021).

[32] Wang N., Zhang Y., Zhang L. Dynamic selection network for image inpainting. IEEE Transactions on Image Processing. **30**, 1784–1798 (2021).

Покращення розфарбовування зображень за допомогою декомпозиції зображення та глибоких нейронних мереж

Беллаж К., Бенмір М., Бужена С.

Лабораторія фундаментальної та прикладної математики,
Кафедра математики та комп'ютерних наук,
Науковий факультет Айн Чок, Касабланка, Марокко

Представлено новий підхід до проблем розфарбовування, який поєднує методи декомпозиції домену (DDM) із глибокими нейронними мережами (DNN) для розв'язування диференціальних рівнянь у частинних похідних (PDE). По-перше, у цій статті розглядаються різні існуючі та нові підходи до малювання, підкреслюючи їхні переваги та недоліки в єдиній структурі. Після цього введено алгоритм, який висвітлює комбінацію методів DDM та DNN для розв'язування PDE запропонованої математичної моделі малювання. Для цієї моделі модифікований підхід, який був прийнятий, використовує метод DNN, який базується на згорткових нейронних мережах (CNN), щоб зменшити витрати на обчислення в нашому алгоритмі, зберігаючи при цьому точність. Накінець, експериментальні результати показують, що запропонованй метод значно перевершує існуючі для зображень високої роздільної здатності в плямах фарби.

Ключові слова: *глибокі нейронні мережи; методи декомпозиції домену; розфарбовування; машинне навчання; рівняння в частинних похідних.*