Pattern Recognition 151 (2024) 110378

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

Neural Knitworks: Patched neural implicit representation networks

Mikolaj Czerkawski [a,*], Javier Cardona [b], Robert Atkinson [a], Craig Michie [a], Ivan Andonovic [a], Carmine Clemente [a], Christos Tachtatzis [a]

[a] Department of Electronic and Electrical Engineering, University of Strathclyde, UK
[b] Department of Chemical Engineering, University of Strathclyde, UK

ARTICLE INFO

Dataset link: https://github.com/cidcom/neural-knitworks

Keywords:
Generative models
Image denoising
Image inpainting
Image super-resolution
Image synthesis
Internal learning
Zero-shot learning

ABSTRACT

Optimizing images as output of a neural network has been shown to introduce a powerful prior for image inverse tasks, capable of producing solutions of reasonable quality in a fully internal learning context, where no external datasets are involved. Two potential technical approaches involve fitting a coordinate-based Multilayer Perceptron (MLP), or a Convolutional Neural Network to produce the result image as output. The aim of this work is to evaluate the two counterparts, as well as a new framework proposed here, named Neural Knitwork, which maps pixel coordinates to local texture patches rather than singular pixel values. The utility of the proposed technique is demonstrated on the tasks of image inpainting, super-resolution, and denoising. It is shown that the Neural Knitwork can outperform the standard coordinate-based MLP baseline for the tasks of inpainting and denoising, and perform comparably for the super-resolution task.

1. Introduction

The research on utilizing coordinate-based Multilayer Perceptron (MLP) networks for image synthesis has developed significantly, yielding a range of impressive results [1–8]. However, the approaches for encoding images using coordinate-based MLPs proposed to date do not explore the potential of introducing additional patch-based constraints. This work attempts to fill this gap, by introducing *Neural Knitworks*, an extension of coordinate-based MLP that maps coordinates to local texture patches, rather than singular RGB values. Learning a neural field of patches, rather than colors, allows for introducing some additional losses that can narrow the space of solutions for image inverse tasks. This design is based on the intuition that local patches and the relationships between them, are an important and relevant factor for solving internal image inverse problems.

The idea of operating on patches is also inspired by the advancements made using models that focus on patch distributions like InGAN [9], SinGAN [10], and the Swapping Autoencoder [11]. The proposed framework can be treated as an extension of a conventional coordinate MLP architecture, where instead of a single color value the network predicts a color patch (or a multi-scale stack thereof) with additional constraints imposed. The purpose of these constraints is to match the distributions of predicted and reference patches and encourage spatial consistency between the predictions. The resulting method can be applied to several image inverse problems, such as image inpainting, super-resolution and denoising, as shown in Fig. 1.

The motivation of this work is to explore the feasibility of the 2D coordinate-based MLPs for the internal image inverse problems, such as inpainting, super-resolution, and denoising, and further compare that to the patch-oriented Neural Knitwork extension proposed. Both of these approaches are also evaluated against the established internal learning techniques for solving image inverse problems that are based on Convolutional Neural Network (CNN) architectures.

It is shown that the Neural Knitwork improves the performance of a coordinate-based MLP solution for the tasks of inpainting and denoising, and results in comparable performance for the super-resolution task. Furthermore, it is found that on the inpainting task, the CNN-based baseline performs better than the coordinate-based approaches. The source code is available at https://github.com/cidcom/neural-knitworks.

2. Related work

The potential of applying an MLP network as an encoding of a signal has been explored in a number of works [1–4,6,12–25]. The learned signals can be of any dimensionality, however, MLP encoding of spatial coordinates is a particularly popular theme, involving a network that learns to produce given scalar values based on the input coordinates. This allows for considerable flexibility and leads to applications such as self-supervised learning of natural images or videos.

Coordinate-Based MLP Networks. The interest in using fully connected networks to represent signals in an implicit manner (also known

* Corresponding author.
E-mail address: mikolaj.czerkawski@strath.ac.uk (M. Czerkawski).

https://doi.org/10.1016/j.patcog.2024.110378
Received 24 August 2022; Received in revised form 14 September 2023; Accepted 25 February 2024

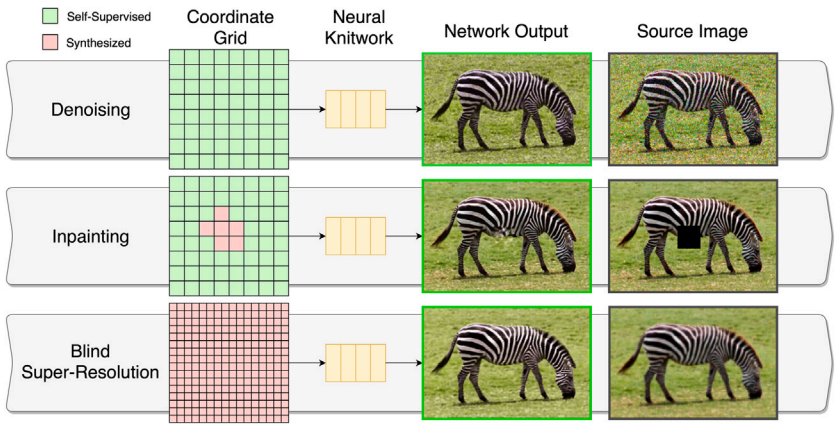


Fig. 1. The introduced model trained on a single sample can perform a number of different image synthesis tasks with very low memory requirements.

as Neural Fields [23]) has grown over the last few years, which can be attributed to the potential of such methods to be used for 3D shape representations [3,4,6,15–17].

An important issue for learning coordinate-based representations is the tendency of neural networks to interpolate and attenuate high-frequency changes in the output [1,2,26]. Two effective solutions to this problem are to either map the input coordinates (known as positional encoding) [1] or use sinusoidal activation functions [2]. However, neither of the two approaches does address the challenge of synthesizing new regions. As demonstrated in a subsequent section (Fig. 3), a standard MLP encoding input with random Fourier features mapping does not synthesize new outputs in a convincing manner.

The novel techniques of random Fourier feature encoding of spatial coordinates gave rise to Neural Radiance Field (NeRF) networks, which can synthesize high fidelity novel views of 3D scenes in an efficient manner [3]. This contribution was soon followed by further developing works, focusing on aspects such as unbounded 3D scenes [4], synthesizing based on few (or only one) images [5], taking advantage of compositionality of 3D scenes [6], concept grounding [18], or applying a similar approach in other domains such as acoustics [19]. Lastly, the work of NeRFusion [20] is an extension of NeRF that derives the radiance and volume density based not only on coordinates but also on estimated per-voxel features.

There have been some works where coordinate-based MLP networks are used as a core for a generative model using techniques such as a hypernetwork predicting the weights of a sample coordinate MLP [13], or by modulating the weights of a base coordinate MLP [14]. These approaches are fundamentally different as they attempt to create a wide generative model based on a large-scale dataset, while the proposed approach focuses on data-agnostic internal learning tasks and uses a disparate architecture. Finally, Local Implicit Image Functions introduced in [27] are trained in a self-supervised manner and are based on latent feature maps used to synthesize an image at different resolutions. However, the architecture relies on a convolutional feature encoder, applies a fixed downsampling operation, and is trained to generate images based on a selected dataset. The proposed architecture is purely based on MLP networks, requires no pretraining, and directly maximizes self-similarity between the synthesized and known patches.

Finally, there exists a line of work, where spatial coordinates are translated into coordinates in texture space, such as Neutex [21], or Neural Atlases for Video [22], but they do not make use of coordinate-to-patch mapping like Neural Knitwork.

Internal Learning for Image Inverse Problems. Patches have been identified as crucial representation features of image in various works [10,28–40]. The introduction of Generative Adversarial Network (GAN)s [41] made it possible to learn patch distributions of images in an adversarial manner [10,11]. Additionally, internal learning approaches relying on the priors contained in convolutional architectures have been proposed [37,42]. To the best of the authors' knowledge, no attempt of introducing these techniques to coordinate-based MLP networks has been made until now.

External Learning for Image Inverse Problems. Many of the recent state-of-the art techniques for image inverse tasks rely on training on external datasets, with well-performing solutions to the problems such as inpainting [43–49] or super-resolution [50–58]. Solutions based on external learning enable the introduction of domain knowledge for solving the task, which is an important source of prior. Yet, there is a significant amount of information often existing in the sole input image, which is where internal learning approaches prove to be useful. Ultimately, a robust and performing model for solving an image inverse task should use both internal and external sources of information to produce the desired output. The scope of this work is focused on the internal learning context, meaning that the core question deals with how well can various models extract information from the input sample alone to solve several synthesis problems. For this reason, the externally learned solutions are not considered relevant in the exact context of this work.

3. Method

The core structure of the proposed network is presented in Fig. 2. It consists of three small networks: (i) Patch MLP for translating from the original coordinate domain to the patch domain (ii) the discriminator responsible for assessing patch likelihoods, and (iii) Pixel MLP for mapping the patch domain to individual pixel color.

The resulting architecture performs the equivalent operation to a conventional coordinate-based MLP since the network ultimately predicts a single pixel value. However, the intermediate patch-based representation of the proposed architecture forces the model to establish the natural relationship between the encoded coordinates. This property can also be used as a useful prior for internal learning scenarios, similar to using convolutional kernels in CNN architectures. Further, the patch representation allows the proposed model to be trained with an adversarial patch loss to match the internal patch distribution with that of the reference image.

3.1. Patch synthesis

The Patch MLP is a network of 4 ReLU layers with 256 units, identical to the one used in [1]. The role of this component is to map each coordinate vector to an appropriate pixel patch. The coordinate input is mapped using random Fourier features before passing to the network. This processing step is known as positional encoding and has been described in detail in [1].

The output of this network approximates the implicit representation function $\phi_{(\mathbf{x})}$ for a query coordinate vector $\mathbf{x}$ along with values of

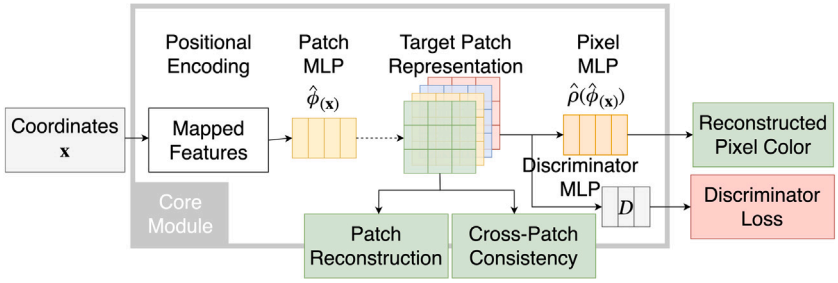


Fig. 2. Neural Knitwork architecture consists of 3 shallow MLPs. The network knits patches for related coordinates by enforcing consistency of predictions and optimizing likelihoods of individual patches. Each patch stack is translated back to a single color by the Pixel MLP. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

neighboring coordinates. The required receptive field depends on the spectral content of the image and can be adjusted by either increasing the patch size to provide more spatial bandwidth or using multi-scale patches. Here, the latter approach is applied as it is more efficient for large spatial spans, allowing for easily configurable scope covered by the output patches at low computational cost. Patches of fixed size 3 by 3 are used for all experiments. For extraction of the patches with scales larger than one, a Gaussian filter is applied to the image to reduce aliasing.

Patch Reconstruction Loss Since the core module is an MLP with multi-scale patch output, a direct way of computing the error is taking the difference of the predicted patches $\hat{\phi}_{(\mathbf{x})}$ and ground truth reference $\phi_{(\mathbf{x})}$. For inpainting tasks, not all pixel values for the patch stack are known and, hence, an appropriate mask $\mathbf{m}_{(\mathbf{x})}$ is applied to this loss. For other tasks, the mask will be an identity tensor. This loss is referred to as patch reconstruction loss $\mathcal{L}_{Recon}$, which is effectively a masked Mean Squared Error (MSE) computed for patches at N sampling coordinates.

$$\mathcal{L}_{Recon} = \sum_{\mathbf{x}}^{N} \frac{(\phi_{(\mathbf{x})} - \hat{\phi}_{(\mathbf{x})})^2 * \mathbf{m}_{(\mathbf{x})}}{|\phi_{(\mathbf{x})}|} \tag{1}$$

The effect of learning patch-based representation rather than direct pixel values has been illustrated in Fig. 3 as part of the ablation study included in the experiments. It becomes quite clear that patch-based representation alone (third column), while helpful, may not yield satisfactory results for challenging synthesis tasks. Instead, additional constraints must be applied to control the relationships between the synthesized values.

Cross-Patch Consistency Loss The ability to produce likely pixels or patches does not necessarily lead to consistent network output when the entire learned image is considered. By default, all patches for which ground truth is available, are optimized to be close to that reference, but this does not guarantee that all patches contribute to a single coherent image for coordinates with no ground truth. For new synthesized regions, the output patches may be convincing on their own (due to the bias component learned by the network from the known region) but display limited coherence between each other.

To encourage consistency, a cross-patch consistency loss is designed that computes the difference between predictions for each pixel from all patches and for the entire image scope. In practice, a way to enforce this, is to use the predictions from the central element of the lowest-scale patch as a reference. The following notation is defined: $\hat{\phi}_{(\mathbf{x})}[\mathbf{i}]$ represents the value of a patch element $\mathbf{i}$ predicted for coordinate $\mathbf{x}$ where $\mathbf{i}$ belongs to the set of I elements across all scales. In a similar fashion, the $\hat{\phi}_{(\mathbf{x})}[\mathbf{o}]$, represents the value of the central element (constant index of $\mathbf{o}$) of the lowest scale patch predicted from coordinate $\mathbf{x}$.

The central reference $\hat{\phi}_{(\mathbf{x})}[\mathbf{o}]$ is compared with element $\hat{\phi}_{(\mathbf{x}+\mathbf{s})}[\mathbf{i}]$ that corresponds to the same pixel of the output image evaluated at coordinates $\mathbf{x}+\mathbf{s}$, where $\mathbf{s}$ indicates the appropriate shift, dependent on $\mathbf{i}$. The terms with values $\mathbf{x}+\mathbf{s}$ outside of the image bounds are naturally excluded from the summation.

$$\mathcal{L}_{\text{X-patch}} = \sum_{\mathbf{x}}^{N} \sum_{\mathbf{i}}^{I} (\hat{\phi}_{(\mathbf{x}+\mathbf{s})}[\mathbf{i}] - \hat{\phi}_{(\mathbf{x})}[\mathbf{o}])^2 \tag{2}$$

Reconstructed Pixel Loss The transition from predicting isolated pixel colors to patches introduces a new trade-off between imposing spatial relationships of the pixel colors and obtaining a high fidelity image with accurate detail. In practice, there will be some disagreement between the predictions for the same pixel from different patches and scales. The naive approach of averaging all predictions for a given coordinate value leads to blurring. To avoid this, a separate Pixel MLP network is used to translate from a multi-scale patch representation to a single color value, by approximating the color extraction function $\rho(\hat{\phi}_{(\mathbf{x})})$, as shown in Fig. 2. The error made by this final output network constitutes the reconstructed pixel loss, encouraging the entire model to produce accurate pixel colors based on a stack of patches.

The pixel reconstruction loss is computed as a ℓ_1 loss between the network pixel color output $\hat{\rho}(\hat{\phi}_{(\mathbf{x})})$ and the color ground truth $\mathbf{c}(\mathbf{x})$

$$\mathcal{L}_{\text{Pixel}} = \sum_{\mathbf{x}}^{N} |\hat{\rho}(\hat{\phi}_{(\mathbf{x})}) - \mathbf{c}(\mathbf{x})| \tag{3}$$

3.2. Patch discriminator

Another important property to enforce, especially when some parts of the signal need to be synthesized, is for all predicted patches to come from a distribution of likely patches, derived from the available information in the source image. This is achieved with the aid of a discriminator tasked to predict which patches come from the original distribution and which do not. The approach is partly inspired by a number of existing works that take advantage of self-similarity between patches in natural images [10,11,31,36,37]. In this case, the discriminator is another MLP consisting of 3 Leaky ReLU layers and taking a flattened patch representation as input.

Discriminator Loss The discriminator network takes a single multi-scale patch and outputs a confidence score. At each training step of the discriminator, all real and all synthesized patches are fed into it and compute the output confidence for them. Furthermore, one-sided label smoothing [59] of the real labels is applied with a factor of 0.1 when computing the discriminator loss in order to penalize over-confidence of this network module. A standard binary cross-entropy loss is used on the discrimination scores.

3.3. Complete objective function

The objective function is a minimax loss where the generator loss term is composed of the four losses contained parameterized by weights α, β, and γ.

$$\mathcal{L}_{Total,G} = \mathcal{L}_{Recon} + \alpha\mathcal{L}_{\text{X-patch}} + \beta\mathcal{L}_{\text{Pixel}} + \gamma\mathcal{L}_{\text{BCE,G}} \tag{4}$$

The discriminator term only includes a single binary-cross entropy loss. Further details about the implementation and the hyperparameters can be found in the appendix.

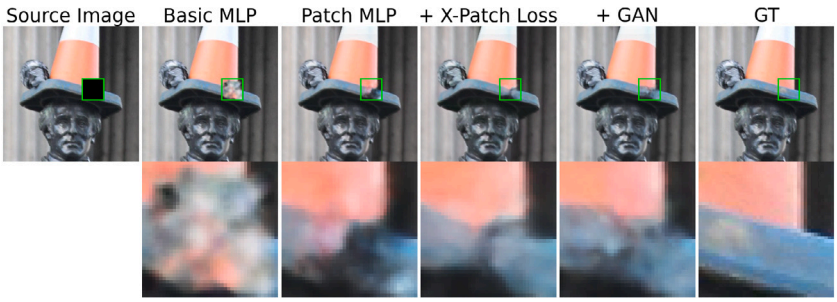


Fig. 3. Ablation study of Neural Knitwork components. Conventional MLP does not produce coherent inpainted region and this is improved with the introduction of patches. Further, imposing cross-patch consistency constraint increases the quality of the synthesized region while employing a GAN approach ensures patches of high likelihood.

4. Experiments

The capabilities of Neural Knitworks are demonstrated by utilizing a similar model with only minor adjustments for several tasks commonly investigated in the field of computer vision: (1) image inpainting (2) super-resolution and (3) denoising. The following section describes the key implementation details for each task and presents corresponding qualitative results. Quantitative measures are provided by applying each method to Set5 [60], Set14 [61], BSD100 [62], and Urban100 [63]. The key objectives of the following experiments is to estimate the potential benefits of the Neural Knitwork framework over a coordinate-based MLP, as well as compare the two to other internal learning techniques that utilize CNN architectures for the same tasks (Deep Image Prior and SinGAN).

4.1. Ablation study

The analysis begins with an ablation study of the proposed architecture to demonstrate the utility of each introduced loss component. Fig. 3 illustrates the effect of the following adjustments to the conventional coordinate MLP network (second column): (i) patch output (third column), (ii) cross-patch consistency loss (fourth column), (iii) patch discrimination (fifth column). It can be observed that the introduction of patch output alone can lead to a more convincing synthesis. However, some distortion can be observed in the synthesized region, which is reduced when cross-patch consistency loss is used. Finally, the addition of a GAN loss leads to improved region consistency.

4.2. Image inpainting

For the image inpainting task, a rectangular section is cut out from the source image to be used as the inpainted region. The coordinates of the cutout are used for producing a mask indicating whether the source signal exists for a given pixel. The mask is used to backpropagate the reconstruction losses only from the pixels outside the inpainted region. The specific configuration used for this task is shown in Table 2.

The results of the inpainting for the Neural Knitwork are compared to a conventional coordinate MLP model and to DIP [42], a CNN-based internal learning approach. Fig. 4 contains the resulting output for the three tested models. The reconstruction quality of the whole image is comparable for the three tested methods. However, when inpainted region is concerned, a significant improvement of over 4 dB can be identified for the Neural Knitwork compared to the conventional coordinate MLP and 2 dB less than the CNN-based technique. For some of the results, the Neural Knitwork was, in fact, able to outperform DIP. Table 1 contains the evaluation across the entire datasets for different fill ratios, which supports that the Neural Knitwork will most often outperform the conventional approach and but generally achieves lower performance than the DIP. More examples can be found in the appendix.

Table 1
Comparison of inpainting performance for different fill ratios. The three approaches appear comparable PSNR (↑) and SSIM (↑) for *whole images*. For the *inpainted region*, the Neural Knitwork comes close to the level of performance of Deep Image Prior (DIP), while conventional MLP is inferior.

Dataset	Fill ratio	MLP	DIP [42]	Ours
	1%	**32.99/0.98**	32.53/0.95	32.00/0.96
Set5	2%	28.65/0.97	29.35/0.92	**29.81/0.94**
Whole image	4%	25.85/0.96	26.22/0.88	**27.96/0.95**
	1%	13.96/0.36	**20.66/0.68**	18.28/0.58
Set5	2%	11.89/0.28	**18.50/0.57**	17.79/0.57
Inpainted region	4%	11.89/0.32	**15.89/0.52**	14.95/0.48
	1%	**28.97/0.95**	28.22/0.90	27.65/0.91
Set14	2%	26.38/0.94	**27.08/0.89**	26.03/0.91
Whole image	4%	24.00/0.93	**25.53/0.89**	24.44/0.89
	1%	11.85/0.23	**16.32/0.41**	15.50/0.40
Set14	2%	10.79/0.23	**15.32/0.40**	13.94/0.39
Inpainted region	4%	10.67/0.24	**14.08/0.37**	12.35/0.36
	1%	**30.04/0.97**	29.60/0.92	28.25/0.94
BSD100	2%	27.61/0.96	**28.46/0.92**	26.70/0.93
Whole image	4%	24.34/0.94	**26.72/0.91**	24.62/0.92
	1%	12.62/0.25	**16.87/0.36**	12.71/0.26
BSD100	2%	11.84/0.24	**15.88/0.36**	12.48/0.27
Inpainted region	4%	11.03/0.23	**15.16/0.35**	12.09/0.28
	1%	**28.79/0.96**	28.60/0.92	27.32/0.93
Urban100	2%	26.64/0.96	**27.84/0.93**	26.12/0.93
Whole image	4%	23.74/0.94	**25.93/0.92**	23.71/0.91
	1%	11.38/0.20	**16.86/0.43**	12.36/0.25
Urban100	2%	10.81/0.19	**15.69/0.38**	11.96/0.25
Inpainted region	4%	10.23/0.17	**14.33/0.33**	11.39/0.24
Parameters		263K	2400K	512K

Table 2
Configuration of the inpainting Neural Knitwork.

Parameter	Value
Patch size	5×5
Patch scales	[1,8]
$\mathcal{L}_{X-patch}$ weight (α)	10^{-5}
$\mathcal{L}_{Pixel}$ weight (β)	1.0
$\mathcal{L}_{GAN}$ weight (γ)	10^{-4}
Learning rate	$4 \cdot 10^{-3}$
Steps	4000

4.3. Super-resolution

To perform super-resolution, a Neural Knitwork has to generate a new image consisting of patches of finer scale, based on the information available in the image of original resolution. This can be done by optimizing the probability distributions of patches across scales, as done in several earlier works [10,35,36,39]. For blind super-resolution, where the underlying downsampling operation is not known, the default model of a Neural Knitwork is utilized with adjusted losses as illustrated in Fig. 5 (see Table 3 for network parameters). The queried coordinates

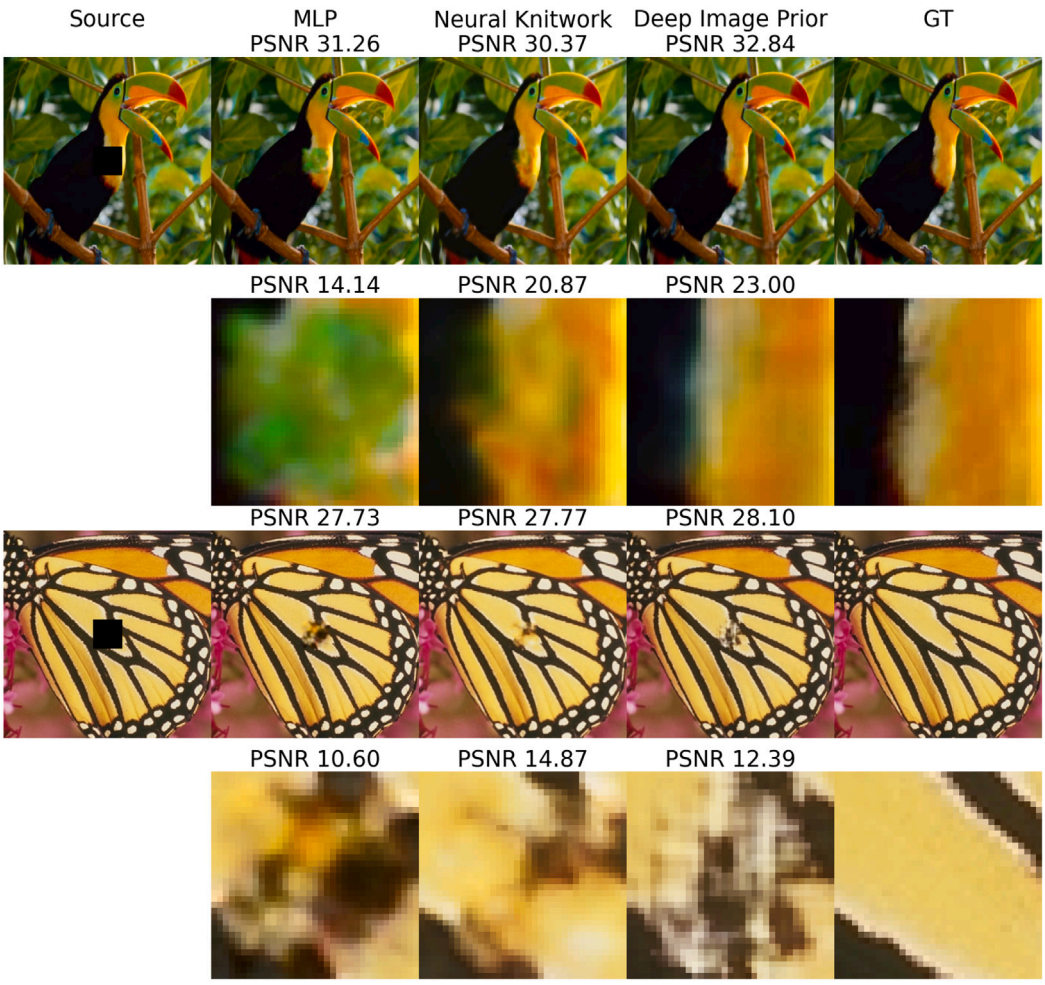


Fig. 4. Image inpainting results for a fill ratio of 2%. For the inpainted region Neural Knitworks and DIP perform comparably, and both outperform conventional MLP.

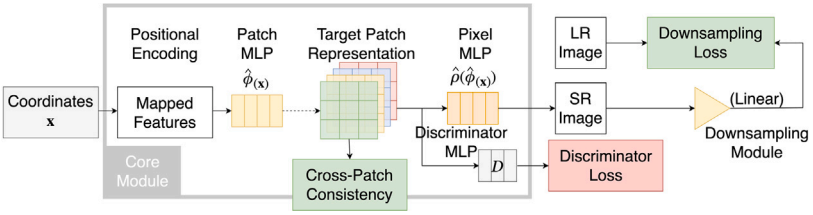


Fig. 5. The blind super-resolution framework utilizes the core module with the addition of a linear network to blindly infer the downsampling kernel. In this case the patch reconstruction loss cannot be computed.

for a patch MLP network include all high-resolution coordinates, which means that it is not possible to compute the patch reconstruction loss in this mode, since the high-resolution image is not available. However, it is possible to compute the cross-patch consistency loss as well as discriminate the patches to match the source image distribution. This alone could yield an output image resembling the low-resolution source without guaranteed structural coherence. To enforce coherence, spatially-aware supervision is applied by downsampling the output of the network and computing the downsampling loss with the reference to the low-resolution source image.

The downsampling operation can be implemented in several ways. If the downsampling kernel is known, then the best approach is to simply backpropagate through that kernel (assuming it is differentiable). Otherwise, a trainable downsampling module representing the kernel can be created with its weights optimized in an end-to-end manner. The technique introduced in [39] is revisited by using an identical deep linear network to approximate the kernel. Their method relies on the assumption that a satisfactory kernel should preserve the distribution of patches in the image. For Neural Knitworks, there is no need to introduce a new loss term accommodating this since the core module objective imposes matching patch distribution by default.

Table 3
Configuration of the super-resolution Neural Knitwork.

Parameter	Value
Patch size	3×3
Patch scales	[1,2,4]
$\mathcal{L}_{X-patch}$ weight (α)	10^{-2}
$\mathcal{L}_{Pixel}$ weight (β)	1.0
$\mathcal{L}_{GAN}$ weight (γ)	10^{-2}
Learning rate	10^{-3}
Steps	4000

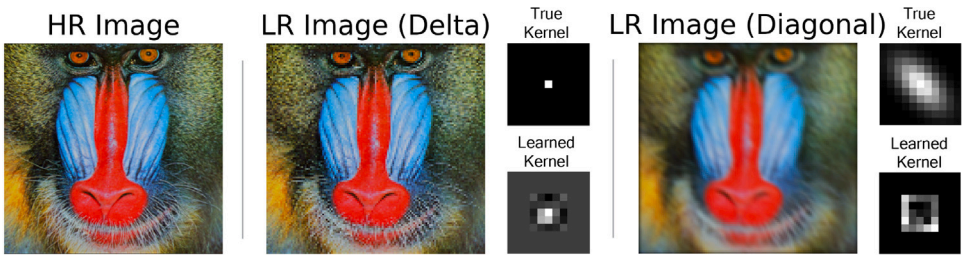


Fig. 6. The method approximates the downsampling kernel depending on the source image.

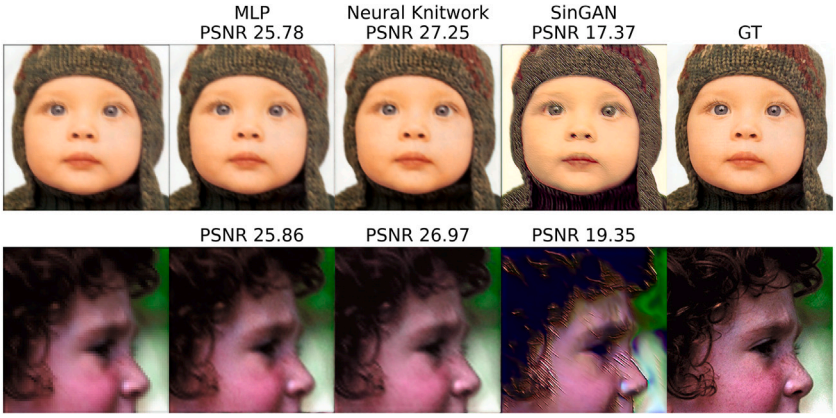


Fig. 7. Comparison of blind image super-resolution for a diagonal Gaussian kernel and upscaling factor of 4×. Neural Knitwork can outperform conventional coordinate MLP network and achieve higher PSNR. SinGAN, while generating a considerable amount of high frequency details, results in significant artifacts.

Table 4
The blind super-resolution performance achieved by a conventional coordinate MLP, a CNN-based internal learning framework of SinGAN and Neural Knitwork. PSNR (↑) and SSIM (↑) are computed for a number of upscaling factors and downsampling kernels (**Delta**, **Diagonal** - Diagonal Gaussian, **Round** - Round Gaussian).

Dataset	Kernel	MLP	SinGAN [10]	Ours
Set5 (2×)	Delta	**31.58/0.95**	19.22/0.65	27.39/0.88
	Diagonal	23.78/0.83	19.95/0.72	**24.62/0.82**
	Round	24.95/0.86	21.59/0.75	**25.48/0.84**
Set5 (4×)	Delta	**25.38/0.85**	17.16/0.53	23.81/0.81
	Diagonal	23.47/0.81	19.15/0.66	**24.22/0.82**
	Round	24.61/0.84	20.75/0.72	**25.36/0.84**
Set14 (2×)	Delta	**27.22/0.89**	14.21/0.41	24.31/0.82
	Diagonal	**22.09/0.75**	16.96/0.56	22.08/0.74
	Round	**22.96/0.78**	17.21/0.57	22.35/0.75
Set14 (4×)	Delta	**22.45/0.76**	14.32/0.33	21.72/0.75
	Diagonal	21.90/0.73	17.75/0.56	**22.05/0.73**
	Round	**22.52/0.76**	18.65/0.62	21.56/0.71
BSD100 (2×)	Delta	**26.83/0.88**	18.77/0.58	23.84/0.80
	Diagonal	22.02/0.73	21.67/0.69	**22.06/0.73**
	Round	22.70/0.76	22.94/0.74	**23.07/0.77**
BSD100 (4×)	Delta	**22.46/0.73**	17.70/0.45	21.39/0.70
	Diagonal	**21.81/0.71**	19.88/0.60	21.69/0.71
	Round	22.33/0.73	20.67/0.64	**22.37/0.74**
Urban100 (2×)	Delta	**24.77/0.87**	13.32/0.32	20.84/0.73
	Diagonal	**19.79/0.66**	15.03/0.40	19.76/0.65
	Round	20.59/0.70	15.07/0.42	**20.86/0.71**
Urban100 (4×)	Delta	**19.61/0.69**	14.34/0.37	18.38/0.62
	Diagonal	19.60/0.64	16.76/0.50	**19.70/0.64**
	Round	20.19/0.67	17.55/0.55	**20.23/0.68**
Parameters		263K	2381K	608K

In Fig. 6, the downsampling effect is demonstrated for two non-standard kernels: (i) delta function (leading to aliasing) and (ii) diagonal Gaussian kernel. Different types of artifacts can be observed depending on the kernel. During training, Neural Knitwork blindly approximates the downsampling kernel based on the image content. The true and learned kernels are illustrated in the figure.

In the case of blind super-resolution, the selected CNN baseline is SinGAN [10]. The rationale for selecting SinGAN is based on the fact, that just like the proposed Neural Knitwork, the images are produced by discriminating patches in the network output to increase similarity between them and the source image. Hence, the framework allows for modeling a variety of potential downsampling operations, just like the Neural Knitwork.

Fig. 7 contains results for a diagonal kernel and upscaling factor of 4, for the proposed Neural Knitwork, the conventional MLP and SinGAN, another image super-resolution method based on internal learning. The results show that SinGAN has the lowest performance in terms of PSNR but it also creates distinguishable artifacts. Table 4 shows how Neural Knitwork compares to counterparts along with the model sizes. Interpolation with conventional MLP directly implies delta kernel and hence, they perform best in this instance. For other kernels, a Neural Knitwork can boost the performance in some instances by adjusting to the kernel. The SinGAN method performs worse than either neural implicit representation method in nearly every case.

In this case, the extensions introduced as part of the Neural Knitwork framework do not provide a consistent advantage over the MLP baseline, which performs very well on the super-resolution task given its simplicity, efficiency, and narrow assumptions about the downsampling operation.

4.4. Denoising

Denoising is the final tested use case for the internal learning techniques explored in this work. In this case, the configuration of the Neural Knitwork is adjusted with the values shown in Table 5. In this case, similarly to the inpainting tests, the CNN baseline used for the task is the Deep Image Prior [42].

As demonstrated in Fig. 8, a standard MLP network has limited denoising capability because it attempts to fit all pixel color values with

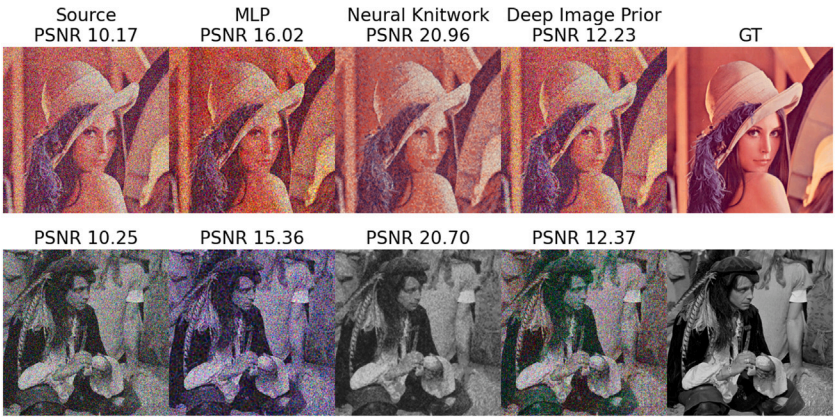


Fig. 8. Neural Knitwork demonstrates superior performance for severe levels of noise, in this case $\sigma = 40$. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 5
Configuration of the denoising Neural Knitwork.

Parameter	Value
Patch size	5×5
Patch scales	[1,4]
$\mathcal{L}_{X-patch}$ weight (α)	10^{-1}
$\mathcal{L}_{Pixel}$ weight (β)	1.0
$\mathcal{L}_{GAN}$ weight (γ)	10^{-2}
Learning rate	10^{-3}
Steps	4000

Table 6
Comparison of achieved denoising performance. For higher power levels, the Neural Knitwork achieves higher PSNR (↑) and SSIM (↑) than a conventional MLP and DIP.

Dataset	σ	MLP	DIP [42]	Neural Knitwork (ours)
Set5	10	23.58/0.70	**27.56/0.85**	26.69/0.83
	20	17.76/0.42	19.15/0.49	**21.69/0.63**
	40	12.43/0.19	11.6/0.15	**15.91/0.37**
Set14	10	24.94/0.77	**26.95/0.84**	26.15/0.85
	20	19.56/0.55	19.70/0.53	**23.08/0.73**
	40	14.55/0.30	12.62/0.19	**18.75/0.56**
BSD100	10	24.66/0.76	**26.63/0.84**	25.83/0.84
	20	18.85/0.48	18.68/0.46	**21.43/0.64**
	40	13.83/0.24	11.92/0.15	**16.23/0.37**
Urban100	10	25.02/0.83	**26.75/0.88**	25.33/0.87
	20	19.69/0.61	20.07/0.62	**21.81/0.73**
	40	14.51/0.35	12.70/0.26	**16.63/0.49**
Parameters		263K	2400K	512K

no additional constraints. In contrast, a Neural Knitwork aims to reconstruct both patches and pixel colors reliably while imposing additional consistency constraint on the derived solution. In the illustrated result with severe noise levels of $\sigma = 40$, PSNR approximately 4 dB higher than in the case of a conventional coordinate MLP is achieved.

Table 6 confirms that the Neural Knitwork model strongly outperforms both other methods for higher noise levels, by a wide margin. The only exception is the low level of $\sigma = 10$, where Deep Image Prior achieves the highest score. Still, in every single case, the conventional coordinate-based MLP achieves much lower performance than the Neural Knitwork extension framework.

5. Conclusion

Neural Knitworks constitute a hybrid architectural approach for internal learning applications, where a coordinate-based model is optimized with a patch context.

For the inpainting task, the coordinate-based solutions do not match the performance of the CNN baseline (DIP), but the proposed Neural Knitwork technique results in a considerable improvement over a standard coordinate-based MLP. For the super-resolution task, the performance of the coordinate-based MLP is comparable to the Neural Knitwork, while the CNN baseline (SinGAN) is consistently much worse than the two. Finally, the Neural Knitwork outperforms both coordinate-based MLP and CNN baseline for the denoising of images at high levels of noise, with quite a wide margin.

In the conducted experiments, the Neural Knitwork used in the experiments is 5× smaller than CNN internal learning counterparts with an additional benefit of being fully parallelizable; that is, all coordinate outputs can be computed independently. Apart from the significant potential for speed up, Neural Knitworks have the advantage of precise control over the output image size by adjusting the set of input coordinates.

In the conducted experiments, the following limitations have been identified. The Neural Knitworks are able to outperform the standard coordinate-based MLP for the task of inpainting and denoising, but achieve quite comparable performance on the task of blind super-resolution. This indicates a potential for improving the techniques based on implicit neural representation by introducing the context of local patches. At present, the experimentation shows that Neural Knitworks can be sensitive to hyperparameters such as individual loss weights, patch sizes, and learning rates, however, the configuration used in the experiments has been shown to offer stable performance.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Available at https://github.com/cidcom/neural-knitworks.

Acknowledgments

This work was funded by EPSRC, United Kingdom under Grant EP/R513349/1

Appendix A. Supplementary data

Supplementary material related to this article can be found online at https://doi.org/10.1016/j.patcog.2024.110378.

References

[1] M. Tancik, P.P. Srinivasan, B. Mildenhall, S. Fridovich-Keil, N. Raghavan, U. Singhal, R. Ramamoorthi, J.T. Barron, R. Ng, Fourier features let networks learn high frequency functions in low dimensional domains, NeurIPS (2020).

[2] V. Sitzmann, J.N. Martel, A.W. Bergman, D.B. Lindell, G. Wetzstein, Implicit neural representations with periodic activation functions, in: Proc. NeurIPS, 2020.

[3] B. Mildenhall, P.P. Srinivasan, M. Tancik, J.T. Barron, R. Ramamoorthi, R. Ng, NeRF: Representing scenes as neural radiance fields for view synthesis, in: ECCV 2020, in: LNCS, vol. 12346, 2020, pp. 405–421, http://dx.doi.org/10.1007/978-3-030-58452-8_24, arXiv:2003.08934.

[4] K. Zhang, G. Riegler, N. Snavely, V. Koltun, NeRF++: Analyzing and improving neural radiance fields, 2020, pp. 1–9, arXiv:2010.07492.

[5] A. Yu, V. Ye, M. Tancik, A. Kanazawa, pixelNeRF: Neural radiance fields from one or few images, in: Proc. IEEE Conf. on Computer Vision and Pattern Recognition, CVPR, 2021.

[6] M. Niemeyer, A. Geiger, GIRAFFE: Representing scenes as compositional generative neural feature fields, in: Proc. IEEE Conf. on Computer Vision and Pattern Recognition, CVPR, 2021.

[7] Y. Guan, A. Chubarau, R. Rao, D. Nowrouzezahrai, Learning neural implicit representations with surface signal parameterizations, Comput. Graph. 114 (2023) 257–264, http://dx.doi.org/10.1016/j.cag.2023.06.013.

[8] T. Novello, G. Schardong, L. Schirmer, V. da Silva, H. Lopes, L. Velho, Exploring differential geometry in neural implicits, Comput. Graph. 108 (2022) 49–60, http://dx.doi.org/10.1016/j.cag.2022.09.003.

[9] A. Shocher, S. Bagon, P. Isola, M. Irani, InGAN: Capturing and retargeting the 'DNA' of a natural image, in: Proceedings of the IEEE International Conference on Computer Vision, Vol. 2019-Octob, 2019, pp. 4491–4500, http://dx.doi.org/10.1109/ICCV.2019.00459.

[10] T.R. Shaham, T. Dekel, T. Michaeli, SinGAN: Learning a generative model from a single natural image, in: Proceedings of the IEEE International Conference on Computer Vision, Vol. 2019-Octob, 2019, pp. 4569–4579, http://dx.doi.org/10.1109/ICCV.2019.00467, arXiv:1905.01164.

[11] T. Park, J.Y. Zhu, O. Wang, J. Lu, E. Shechtman, A.A. Efros, R. Zhang, Swapping autoencoder for deep image manipulation, 2020, (NeurIPS).

[12] E. Dupont, A. Goliński, M. Alizadeh, Y.W. Teh, A. Doucet, COIN: Compression with implicit neural representations, in: ICLR 2021 Neural Compression Workshop, 2021, pp. 1–12.

[13] I. Skorokhodov, S. Ignatyev, M. Elhoseiny, Adversarial generation of continuous images, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR, 2021, pp. 10753–10764.

[14] I. Anokhin, K. Demochkin, T. Khakhulin, G. Sterkin, V. Lempitsky, D. Korzhenkov, Image generators with conditionally-independent pixel synthesis, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 14278–14287.

[15] J.J. Park, P. Florence, J. Straub, R. Newcombe, S. Lovegrove, Deepsdf: Learning continuous signed distance functions for shape representation, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Vol. 2019-June, 2019, pp. 165–174, http://dx.doi.org/10.1109/CVPR.2019.00025.

[16] K. Genova, F. Cole, D. Vlasic, A. Sarna, W.T. Freeman, T.A. Funkhouser, Learning shape templates with structured implicit functions, 2019, CoRR abs/1904.06447.

[17] M. Atzmon, Y. Lipman, SAL: sign agnostic learning of shapes from raw data, 2019, CoRR abs/1911.10414.

[18] Y. Hong, Y. Du, C. Lin, J.B. Tenenbaum, C. Gan, 3D concept grounding on neural fields, in: A.H. Oh, A. Agarwal, D. Belgrave, K. Cho (Eds.), Advances in Neural Information Processing Systems, 2022.

[19] A. Luo, Y. Du, M. Tarr, J. Tenenbaum, A. Torralba, C. Gan, Learning neural acoustic fields, Adv. Neural Inf. Process. Syst. 35 (2022) 3165–3177.

[20] X. Zhang, S. Bi, K. Sunkavalli, H. Su, Z. Xu, NeRFusion: Fusing radiance fields for large-scale scene reconstruction, 2022, pp. 5439–5448, http://dx.doi.org/10.1109/cvpr52688.2022.00537, arXiv:2203.11283.

[21] F. Xiang, Z. Xu, M. Hašan, Y. Hold-Geoffroy, K. Sunkavalli, H. Su, NeuTex: Neural texture mapping for volumetric neural rendering, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2021, pp. 7115–7124, http://dx.doi.org/10.1109/CVPR46437.2021.00704.

[22] Y. Kasten, D. Ofri, O. Wang, T. Dekel, Layered neural atlases for consistent video editing, ACM Trans. Graph. 40 (6) (2021) 1–12, http://dx.doi.org/10.1145/3478513.3480546.

[23] Y. Xie, T. Takikawa, S. Saito, O. Litany, S. Yan, N. Khan, F. Tombari, J. Tompkin, V. Sitzmann, S. Sridhar, Neural fields in visual computing and beyond, Comput. Graph. Forum 41 (2) (2022) 641–676, http://dx.doi.org/10.1111/cgf.14505, arXiv:2111.11426.

[24] S. Lee, J.-B. Jeong, E.-S. Ryu, Entropy-constrained implicit neural representations for deep image compression, IEEE Signal Process. Lett. 30 (2023) 663–667, http://dx.doi.org/10.1109/LSP.2023.3279780.

[25] Y. Wu, Z. Zou, Z. Shi, Remote sensing novel view synthesis with implicit multiplane representations, IEEE Trans. Geosci. Remote Sens. 60 (2022) 1–13, http://dx.doi.org/10.1109/TGRS.2022.3197409.

[26] N. Rahaman, A. Baratin, D. Arpit, F. Draxler, M. Lin, F. Hamprecht, Y. Bengio, A. Courville, On the spectral bias of neural networks, in: K. Chaudhuri, R. Salakhutdinov (Eds.), Proceedings of the 36th International Conference on Machine Learning, in: Proceedings of Machine Learning Research, vol. 97, PMLR, 2019, pp. 5301–5310.

[27] Y. Chen, S. Liu, X. Wang, Learning continuous image representation with local implicit image function, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 8628–8638.

[28] A.A. Efros, W.T. Freeman, Image quilting for texture synthesis and transfer, in: Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 2001, 2001, pp. 341–346, http://dx.doi.org/10.1145/383259.383296.

[29] V. Kwatra, A. Schödl, I. Essa, G. Turk, A. Bobick, Graphcut textures: Image and video synthesis using graph cuts, ACM Trans. Graph. 22 (3) (2003) 277–286, http://dx.doi.org/10.1145/882262.882264.

[30] D. Simakov, Y. Caspi, E. Shechtman, M. Irani, Summarizing visual data using bidirectional similarity, in: 26th IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 2008, http://dx.doi.org/10.1109/CVPR.2008.4587842.

[31] D. Glasner, S. Bagon, M. Irani, Super-resolution from a single image, in: Proceedings of the IEEE International Conference on Computer Vision, 2009, pp. 349–356, http://dx.doi.org/10.1109/ICCV.2009.5459271.

[32] M. Zontak, M. Irani, Internal statistics of a single natural image, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, IEEE, 2011, pp. 977–984, http://dx.doi.org/10.1109/CVPR.2011.5995401.

[33] M. Zontak, I. Mosseri, M. Irani, Separating signal from noise using patch recurrence across scales, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, IEEE, 2013, pp. 1195–1202, http://dx.doi.org/10.1109/CVPR.2013.158.

[34] L.A. Gatys, A.S. Ecker, M. Bethge, Texture synthesis using convolutional neural networks, Adv. Neural Inf. Process. Syst. 2015-January (2015) 262–270.

[35] T. Michaeli, M. Irani, Blind deblurring using internal patch recurrence, in: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), in: LNCS, vol. 8691, 2014, pp. 783–798, http://dx.doi.org/10.1007/978-3-319-10578-9_51.

[36] A. Shocher, N. Cohen, M. Irani, Zero-shot super-resolution using deep internal learning, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2018, pp. 3118–3126, http://dx.doi.org/10.1109/CVPR.2018.00329, arXiv:1712.06087.

[37] Y. Gandelsman, A. Shocher, M. Irani, 'Double-dip': Unsupervised image decomposition via coupled deep-image-priors, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Vol. 2019-June, 2019, pp. 11018–11027, http://dx.doi.org/10.1109/CVPR.2019.01128.

[38] R. Mechrez, E. Shechtman, L. Zelnik-Manor, Saliency driven image manipulation, Mach. Vis. Appl. 30 (2) (2019) 189–202, http://dx.doi.org/10.1007/s00138-018-01000-w, arXiv:1612.02184.

[39] S. Bell-Kligler, A. Shocher, M. Irani, Blind super-resolution kernel estimation using an internal-GAN, in: H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché Buc, E. Fox, R. Garnett (Eds.), Advances in Neural Information Processing Systems, Vol. 32, Curran Associates, Inc., 2019.

[40] T. Zhang, Y. Fu, D. Zhang, C. Hu, Deep external and internal learning for noisy compressive sensing, Neurocomputing 531 (2023) 61–73, http://dx.doi.org/10.1016/j.neucom.2023.01.092.

[41] I.J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial nets, Adv. Neural Inf. Process. Syst. 3 (January) (2014) 2672–2680, http://dx.doi.org/10.3156/jsoft.29.5_177_2.

[42] D. Ulyanov, A. Vedaldi, V. Lempitsky, Deep image prior, Int. J. Comput. Vis. 128 (7) (2020) 1867–1888, http://dx.doi.org/10.1007/s11263-020-01303-4.

[43] R. Suvorov, E. Logacheva, A. Mashikhin, A. Remizova, A. Ashukha, A. Silvestrov, N. Kong, H. Goka, K. Park, V. Lempitsky, Resolution-robust large mask inpainting with Fourier convolutions, in: Proceedings - 2022 IEEE/CVF Winter Conference on Applications of Computer Vision, WACV 2022, 2022, pp. 3172–3182, http://dx.doi.org/10.1109/WACV51458.2022.00323.

[44] C. Saharia, W. Chan, H. Chang, C. Lee, J. Ho, T. Salimans, D. Fleet, M. Norouzi, Palette: Image-to-image diffusion models, Proc. ACM SIGGRAPH 1 (1) (2022) 1–10, http://dx.doi.org/10.1145/3528233.3530757.

[45] H. Xiang, Q. Zou, M.A. Nawaz, X. Huang, F. Zhang, H. Yu, Deep learning for image inpainting: A survey, Pattern Recognit. 134 (2023) 109046, http://dx.doi.org/10.1016/j.patcog.2022.109046.

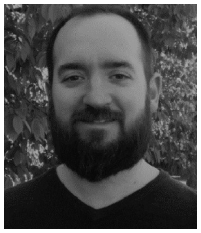
[46] W. Huang, Y. Deng, S. Hui, Y. Wu, S. Zhou, J. Wang, Sparse self-attention transformer for image inpainting, Pattern Recognit. (2023) 109897, http://dx.doi.org/10.1016/j.patcog.2023.109897.

[47] J. Wang, C. Yuan, B. Li, Y. Deng, W. Hu, S. Maybank, Self-prior guided pixel adversarial networks for blind image inpainting, IEEE Trans. Pattern Anal. Mach. Intell. 45 (10) (2023) 12377–12393, http://dx.doi.org/10.1109/TPAMI.2023.3284431.

[48] G. Liu, A. Dundar, K.J. Shih, T.-C. Wang, F.A. Reda, K. Sapra, Z. Yu, X. Yang, A. Tao, B. Catanzaro, Partial convolution for padding, inpainting, and image synthesis, IEEE Trans. Pattern Anal. Mach. Intell. 45 (5) (2023) 6096–6110, http://dx.doi.org/10.1109/TPAMI.2022.3209702.

[49] C. Cao, Q. Dong, Y. Fu, ZITS++: Image inpainting by improving the incremental transformer on structural priors, IEEE Trans. Pattern Anal. Mach. Intell. 45 (10) (2023) 12667–12684, http://dx.doi.org/10.1109/TPAMI.2023.3280222.
[50] K. Hayat, Multimedia super-resolution via deep learning: A survey, Digit. Signal Process. 81 (2018) 198–217, http://dx.doi.org/10.1016/j.dsp.2018.07.005.
[51] S. Menon, A. Damian, S. Hu, N. Ravi, C. Rudin, PULSE: Self-supervised photo upsampling via latent space exploration of generative models, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2020, pp. 2434–2442, http://dx.doi.org/10.1109/CVPR42600.2020.00251, arXiv:2003.03808.
[52] C. Saharia, J. Ho, W. Chan, T. Salimans, D.J. Fleet, M. Norouzi, Image super-resolution via iterative refinement, IEEE Trans. Pattern Anal. Mach. Intell. (2022) http://dx.doi.org/10.1109/TPAMI.2022.3204461, arXiv:2104.07636.
[53] I.H. Lee, W.Y. Chung, C.G. Park, Style transformation super-resolution GAN for extremely small infrared target image, Pattern Recognit. Lett. 174 (2023) 1–9, http://dx.doi.org/10.1016/j.patrec.2023.08.013.
[54] Y. Yu, J. Jiang, H. Chang, H. Zheng, S. Wang, Face super-resolution via joint edge information and attention aggregation network, Comput. Electr. Eng. 111 (2023) 108931, http://dx.doi.org/10.1016/j.compeleceng.2023.108931.
[55] Y. Wu, R. Cao, Y. Hu, J. Wang, K. Li, Combining global receptive field and spatial spectral information for single-image hyperspectral super-resolution, Neurocomputing 542 (2023) 126277, http://dx.doi.org/10.1016/j.neucom.2023.126277.
[56] N. Yuan, B. Sun, X. Zheng, Unsupervised real image super-resolution via knowledge distillation network, Comput. Vis. Image Underst. 234 (2023) 103736, http://dx.doi.org/10.1016/j.cviu.2023.103736.
[57] S. Angarano, F. Salvetti, M. Martini, M. Chiaberge, Generative adversarial super-resolution at the edge with knowledge distillation, Eng. Appl. Artif. Intell. 123 (2023) 106407, http://dx.doi.org/10.1016/j.engappai.2023.106407.
[58] C. Ma, Y. Rao, J. Lu, J. Zhou, Structure-preserving image super-resolution, IEEE Trans. Pattern Anal. Mach. Intell. 44 (11) (2022) 7898–7911, http://dx.doi.org/10.1109/TPAMI.2021.3114428.
[59] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, X. Chen, Improved techniques for training GANs, Adv. Neural Inf. Process. Syst. (2016) 2234–2242.
[60] M. Bevilacqua, A. Roumy, C. Guillemot, M. line Alberi Morel, Low-complexity single-image super-resolution based on nonnegative neighbor embedding, in: Proceedings of the British Machine Vision Conference, BMVA Press, 2012, pp. 135.1–135.10, http://dx.doi.org/10.5244/C.26.135.
[61] R. Zeyde, M. Elad, M. Protter, On single image scale-up using sparse-representations, in: Proceedings of the 7th International Conference on Curves and Surfaces, Springer-Verlag, 2010, pp. 711–730, http://dx.doi.org/10.1007/978-3-642-27413-8_47.
[62] D. Martin, C. Fowlkes, D. Tal, J. Malik, A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics, in: Proceedings of the IEEE International Conference on Computer Vision, Vol. 2, 2001, pp. 416–423, http://dx.doi.org/10.1109/ICCV.2001.937655.
[63] J.B. Huang, A. Singh, N. Ahuja, Single image super-resolution from transformed self-exemplars, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Vol. 07-12-June-2015, 2015, pp. 5197–5206, http://dx.doi.org/10.1109/CVPR.2015.7299156.

Mikolaj Czerkawski received the B.Eng. degree in electronic and electrical engineering in 2019 from the University of Strathclyde in Glasgow, United Kingdom, and the Ph.D.degree in 2023 at the same university, specialising in applications of computer vision to Earth observation data. His research interests include image synthesis, generative models, and use cases involving restoration tasks of satellite imagery. He is currently a research fellow at the European Space Agency in ESRIN, Frascati, Italy.



Javier Cardona received the B.Eng degree (Hons.) in chemical engineering from the University of Alicante, Alicante, Spain in 2011, and the M.Sc. and Ph.D. degrees in chemical and process engineering from the University of Strathclyde, Glasgow, U.K., in 2012 and 2016, respectively. He is currently a Lecturer and Chancellor's Fellow with the Departments of Electronic and Electrical Engineering and Chemical and Process Engineering at the University of Strathclyde. His current research interests include multiscale modeling and the application of data analytics and machine learning methods in digital manufacturing, including chemical and pharmaceutical manufacturing and agricultural processes.

Robert Atkinson received the B.Eng. degree (Hons.) in electronic and electrical engineering, the M.Sc. degree in communications, control, and digital signal processing, and the Ph.D. degree in mobile communications systems from the University of Strathclyde, Glasgow, U.K., in 1993, 1995, and 2003, respectively. He is currently a Reader with the institution. His research interests include data engineering and the application of machine learning algorithms to industrial problems, including cyber-security.

Craig Michie received the B.Sc. degree in electronics and electrical engineering and the Ph.D. degree in coherent optical communications from the University of Glasgow, Glasgow, U.K., in 1983 and 1988, respectively. He is currently a Professor of electronics and electrical engineering with the Department of Electronic and Electrical Engineering, University of Strathclyde, Glasgow. He is the author of over 170 journal and conference papers in sensing and communications. His current research interests include wireless sensor systems.

Ivan Andonovic has held a Royal Society Industrial Fellowship in collaboration with British Telecommunications (BT) Labs investigating novel approaches to broadband networking. He has edited two books and authored/coauthored six chapters in books and over 380 journal and conference papers and secured funding for research and development in excess of £10M. He is a Fellow of the IET. He was a member of flagship Scottish Enterprise (Government agency for economic growth) team of the Intermediary Technology Institutes (ITIs), aimed at bridging the gap between basic research and company growth, has been a Visiting Scientist at the Communications Research Laboratories of Japan, a Visiting Professor with the City University of Hong Kong and Princeton University, the Topical Editor for the IEEE Transactions on Communications and the Technical Programme Co-Chair for the IEEE International Conference in Communications (ICC07).

Carmine Clemente received the Laurea (cum laude) (B.Sc.) and (M.Sc.) Laurea Specialistica (cum laude) degrees in telecommunications engineering from Universita degli Studi del Sannio, Benevento, Italy, in 2006 and 2009, respectively, and the Ph.D. degree in signal processing from the Department of Electronic and Electrical Engineering, University of Strathclyde, Glasgow, U.K.

He is currently a Reader with the Department of Electronic and Electrical Engineering, University of Strathclyde working on advanced Radar signal processing algorithm, MIMO radar systems, and micro-Doppler analysis. His research interests include synthetic aperture radar (SAR) focusing and bistatic SAR focusing algorithms development, micro-Doppler signature analysis and extraction from multistatic radar platforms, micro-Doppler classification, and statistical signal processing.

Christos Tachtatzis received the B.Eng. degree (Hons.) in communication systems engineering from the University of Portsmouth in 2001, and the M.Sc. degree in communications, control and digital signal processing and the Ph.D. degree in electronic and electrical engineering from Strathclyde University, in 2002 and 2008, respectively. He was Chancellor's Fellow at the University of Strathclyde from 2016 to 2021 and he is currently a Reader of sensor systems and asset management with the University of Strathclyde. He has authored over 100 journal and conference publications. He has 14 years experience, in Sensor Systems ranging from electronic devices, networking, and communications and signal processing. His current research interests include extracting actionable information from data using machine learning and artificial intelligence.