Patch Neural Representation for Image Processing

Haojun Qiu

Abstract—This paper proposes a novel approach to represent single image. The approach involves explicitly modeling the patch-to-patch relationship through function approximation using neural networks. By utilizing function approximation, the proposed method can learn complex, non-linear relationships between patches, which leads to better generalization to new data. The proposed method encodes the information about the nearest patch or the general patch relationship within the weights of the neural network, eliminating the need for searching every time a prediction is made. The approach is evaluated on image restoration tasks such as denoising and super-resolution from a single image, and the results demonstrate the effectiveness of the proposed method. Overall, this paper presents a promising solution for learning single-image processing tasks with limited data.

Index Terms—Image Processing, Neural Network

1 Introduction

Neural networks have emerged as powerful universal function approximators [1], paving the way for significant advancements in various domains such as computer vision, and low-level image processing. Numerous research efforts have demonstrated excellence in these tasks, predominantly relying on supervised learning from large human-labeled datasets. However, collecting extensive datasets can be costly, which often necessitates learning with limited data. Our work is situated at the most extreme end of this spectrum, focusing on learning from a single image, particularly for single-image processing tasks. On the other hand, the key to success in single-image methods, sometimes augmented with an image pyramid, lies in the rich internal patch statistics that can be easily acquired. This patch distribution can be effectively learned and then applied to several useful downstream tasks such as super-resolution and denoising. Our method aims to solve these task by explicitly modeling the patch-to-patch relationship.

To illustrate the importance of modeling patch relationship more specifically, consider the case of image denoising, where we observe noisy measurements of an image and aim to recover its clean version. In many cases, noise has zero mean, but this is not necessarily always true. Some works ingeniously average similar noisy patches to recover clean patches [2]. Meanwhile, for super-resolution (SR), given a patch, the task involves finding a nearest neighbor (NN) low-resolution (LR) patch such that its high-resolution (HR) patch counterpart (in the same relative position but at a higher scale of the pyramid) can be upscaled through a simple "copy-paste" operation [3], [4]. However, the success of these methods highly depends on the algorithm for *searching* or matching similar patches (often determined by a proximity score like mean squared error). This aspect is critical for enabling the aforementioned downstream tasks and has been a subject of research for decades [5], [6]. Unfortunately, such searching method has a few limitations: (1) Nearest neighbor approaches can only represent relationships present in the training data, potentially hindering their ability to generalize well to new, unseen data points.

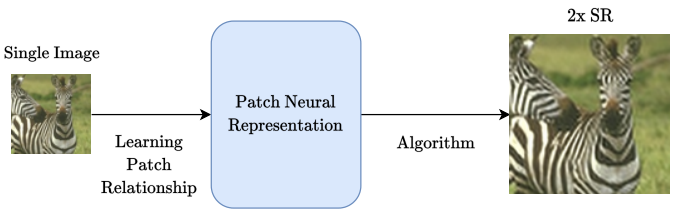

Fig. 1. Our model aims to learn the patch relationship through a neural network. After training, we can adopt the neural network for image processing taks, such as superresolution.

(2)Furthermore, every time we want to make a prediction, we must execute the same searching algorithm. This process can be costly at prediction time compared to encoding information about the nearest neighbor patch (NNP) compactly within the parameters of a function.

In this paper, we propose a novel approach that fits a mapping from a patch with some auxiliary information to another patch using function approximation with neural networks. We build on the insightful observation that all the disadvantages previously discussed can be addressed by modeling a function rather than relying on searching. By utilizing function approximation, we can learn more complex, non-linear relationships, enabling better generalization to new data. Furthermore, after training, the information about the nearest patch and the general patch relationship are encoded within the weights of the neural network, eliminating the need for searching at every inference time. Specifically, we use a Multilayer Perceptron (MLP) as a function approximator, training it to overfit the patch-to-patch relationship, conditioned on variables such as pixel displacements and relative scales, which we refer to as the delta space. From another perspective, this can be thought of as learning how to traverse locally from point to point within a patch manifold, conditioned on certain deltas. Intuitively, in the function approximation setting, the implicit matching of patches is achieved when two similar patch

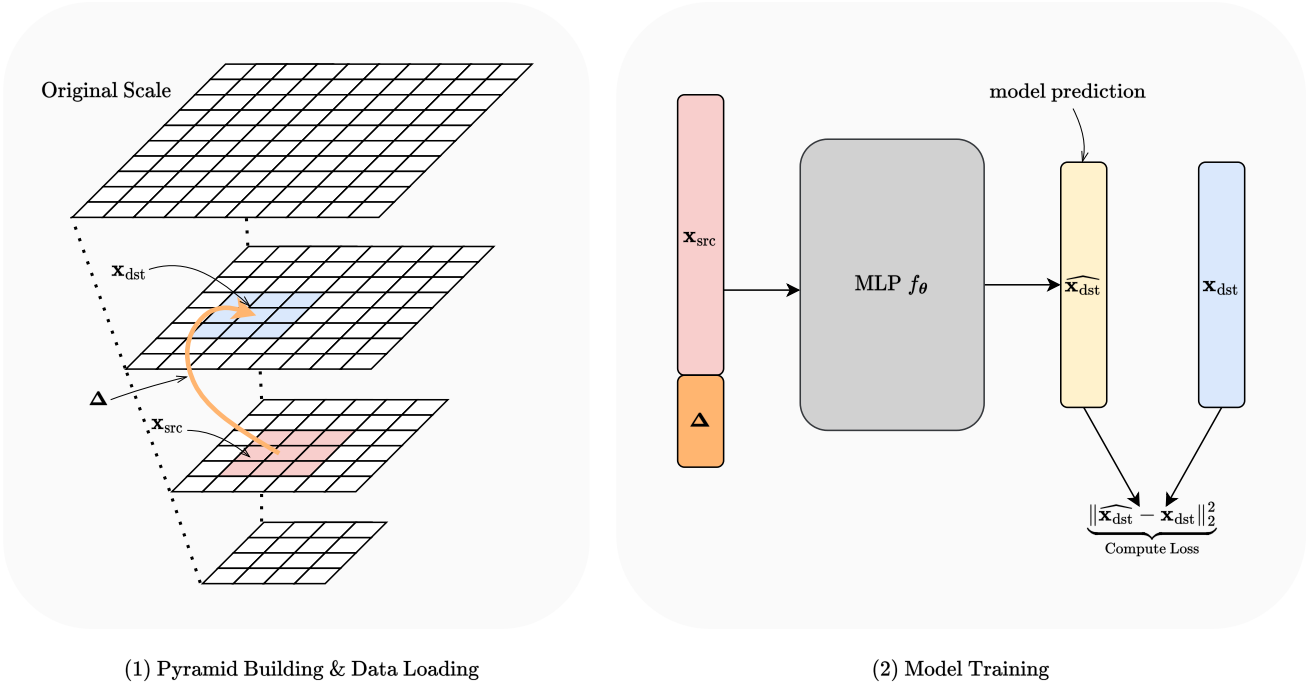

Fig. 2. Pipeline. The left demonstrate how to gather a triple of data (source patch, delta, destination patch) from a single image pyramid. The right demonstrates how the learning is done after gathering all the data.

inputs (along with the similar auxiliary input) yield similar patch outputs, when function is smooth.

Overall, we make the following contributions:

- We introduce a new patch-based neural representation scheme – mapping a patch to its neighboured patch with some auxiliary traversal information, within an image pyramid. This implicitly allows *globally* matching the behaviour/relationship of patch pairs, and enables the generality to unseen data given the form of function approximation.
- We evaluate the capability of trained model on image restoration tasks like denoising and super-resolution from a single image.

2 Related Work

2.1 Patch-Based Models

We begin by examining past methods that focus on modeling images at the patch level. Traditionally, finding the nearest neighbor patch has been the most straightforward way to model patch relationships. A patch's nearest neighbor refers to the patch that is closest to itself, as determined by a defined distance metric such as L2. Finding the exact nearest neighbor through brute force is computationally infeasible. As a result, PatchMatch [5], an approximate algorithm proven to be both practically effective and theoretically convergent, has been widely adopted. It identifies the nearest neighbor patch through a two-step iterative algorithm – propagation and random search – that iteratively processes patches in a raster scan order of pixel positions. Building upon this, Generalized PatchMatch [6] enables finding nearest neighbor patches across scale and rotation as well. However, searching algorithms may not be efficient at prediction time as we have to re-run algorithm to retrive the patch nearest neighbours, and searching algorithm could struggle to generalize to unseen data that is significantly outside the patch distribution. In the era of deep learning, some works model images as patches too, treating non-overlapping patches as a sequence of tokens. Borrowing ideas from natural language processing, they employ transformers [7] as their model backbone to introduce global attention, as seen in ViT [8] and MAE [9]. However, these patch-level transformers predominantly target high-level vision tasks like object recognition and segmentation, and require a vast amount of data for training. Our model relates neighboring patches from a single image. By using a neural network as a function approximator, it has the potential to generalize to unseen data, as demonstrated in tasks such as super-resolution.

2.2 Neural Representation for Image

Numerous methods have investigated how different architectures of neural networks (specifically, MLPs) can implicitly represent images or scenes by querying continuous coordinate-based neural networks, also known as neural fields [10], [11], [12], [13], [14]. One of the primary advantages of coordinate-based neural networks is compression: representing large images, such as gigapixels, with a relatively small network that occupies less memory. However, our work is fundamentally distinct from these methods. We do not focus on compression, nor do we represent images by querying neural networks based on spatial positions. Our innovative approach is to adapt simple MLPs for querying a patch with inputs as its neighboring patch and the small

spatial and transformation relation that associates the two patches, such as relative displacements, scale factors, or even rotation degrees. Modeling this neighboring patch relationship paves the way for image processing applications, such as denoising or super-resolution.

2.3 Single Image Processing

Many works have investigated how to process image, e.g., denoising and super-resolution using only a single image as a data source. For image denoising, [2] initially demonstrates that averaging similar patches within a local search window results in a clean patch if noise is zero-mean. Subsequently, BM3D [15] advances this idea with collaborative 3D filtering, significantly improving performance and continuing to serve as a state-of-the-art approach today. Another work, [16], comparable to BM3D, leverages the patch recurrence property across scales and the observation that less noise exists in downscaled images. For image super-resolution, the main idea stems from example-based learning [3]. This approach predicts patches at higher resolution by matching lower-resolution pairs in a repository of LR-HR pairs, with simple copy-pasting serving as the prediction step. The work [4] gathers pairs from single image pyramids, and [17] generalizes this idea to search over the space of all affine transformation of patches in the pyramid. However, these method, due to dependency on searching, all face the problem of generalization when encountering unseen data. And, they restrict themselves to relatively local search spaces for searching efficiency considerations, not exploiting the information from the whole patch statistics provided by image pyramids.

In the deep learning era, there have been a few successful single-image approaches as well. The work [18] use a CNN to map from noise to an image, enabling various applications including denoising and super-resolution. ZSSR [19] trains an image-specific CNN to map from LR to HR images, while [20] builds on that by leveraging meta-learning to further enhance performance. However, these method, due to adoption of CNNs, do not explicitly model the relationship between neighboring patches. Our method, which models the mapping from patch + delta to patch with an MLP, not only does the modeling of patch relationship, can also naturally be adopted for both denoising and super-resolution applications.

3 Proposed Method

In section 3.1, we first introduces how do we gather patch data from the image, and how we pair them up. In section 3.2, we propose a simple regression approach to model the conditional distribution as deterministic mapping, and discuss the rationality why the patch relation are restricted to small values in our practice. The overall pipeline is shown in [Fig. 2].

3.1 Multi-Scale Patch Representation

We introduce a novel approach for collecting a patch dataset from an image, this will serves as the learning data. Specifically, we build an image pyramid and gather each data as a pair of patches together with their relationship. We think that this dataset, containing enriched patch statistics, is a great learning source, as we will discuss in 3.2. Be more specific, let $\mathbf{I} \in \mathbb{R}^{H\times W\times C}$ be a given image, where H, W, and C denote height, width and number of channels, respectively. We desire to construct a image pyramid, by first fixing a constant relative scale factor $(s_H, s_W) \in \mathbb{R}$ for both the height and width dimensions, between two consecutive layers. A pyramid $\{\mathbf{I}_l\}_{l=0}^{L-1}$ of L layers with decreasing resolutions is constructed such that,

- $\mathbf{I}_0$ denotes the original image, and $\mathbf{I}_{L-1}$ denotes the image of smallest resolution;
- Each of the two layers are related roughly by the fixed scale factor, i.e., let H_l denotes the height of $\mathbf{I}_l$ (so $H_0 = H$), we have $H_{l+1}/H_l \approx s_H$ for all $l \in \{0, \dots, L-2\}$, similarly for the widths.

Provided this constructed pyramid, for a fixed patch size of $P \times P$, we are able to sample data triples

$$(\mathbf{x}_{\text{src}}, \boldsymbol{\Delta}, \mathbf{x}_{\text{dst}}), \tag{1}$$

where each consists of (1) a source patch $\mathbf{x}_{\text{src}} \in \mathbb{R}^{P\times P\times C}$ locates somewhere in the pyramid (2) the delta

$$\boldsymbol{\Delta} = (u, v, s_x, s_y) \in \mathbb{R}^4 \tag{2}$$

containing vertical and horizontal (sub-)pixel displacements and scales, governing how to transform and traverse from source patch $\mathbf{x}_{\text{src}}$ to (3) a destination patch $\mathbf{x}_{\text{dst}} \in \mathbb{R}^{P\times P\times C}$ that has the same size as source patch. We use an efficient algorithm (refer to algorithm box) to gather such data triple. This comprises a dataset that can be used to model the underlying true distribution, for which we denote as

$$\mathcal{D} = \left\{ \left(\mathbf{x}_{\text{src}}^{(i)}, \boldsymbol{\Delta}^{(i)}, \mathbf{x}_{\text{dst}}^{(i)} \right) \right\}_{i=1}^{N}, \tag{3}$$

where N is the total number of triples gathered from the algorithm, and the super-script enclosed by braces denotes the index for each data triple.

3.2 Regression with Neural Network

We aim to model an mapping from source patches with a delta to a destination patch, i.e.,

$$(\mathbf{x}_{\text{src}}, \boldsymbol{\Delta}) \mapsto \mathbf{x}_{\text{dst}}. \tag{4}$$

Naturally, we choose an simple multi-layer perceptron (MLP) as it has shown to be a good universal function approximator [1], and conduct a continuous regression on it. After training, the MLP can be fixelbly adopted for downstream tasks, and it is deemed as a type of representation for patch as it encodes patch relationship in its parameter and its output signal is a patch.

Specifically, with the same notation as before, let $f_{\boldsymbol{\theta}} : \mathbb{R}^{P\times P\times C+4} \to \mathbb{R}^{P\times P\times C}$ be a MLP parametrized by $\boldsymbol{\theta}$ such that the prediction is closed to ground truth

$$f_{\boldsymbol{\theta}} \left(\mathbf{x}_{\text{src}}^{(i)}, \boldsymbol{\Delta}^{(i)} \right) \approx \mathbf{x}_{\text{dst}}^{(i)}, \quad \forall i \in \{1, \dots, N\}. \tag{5}$$

Similar to all learning-based regression method, we calculate the mean loss between predicted destination patch and ground-truth destination patch:

$$\begin{aligned} \mathcal{L}(\boldsymbol{\theta}) &= \mathbb{E}_{\mathcal{D}}[\ell(f_{\boldsymbol{\theta}}(\mathbf{x}_{\text{src}}, \boldsymbol{\Delta}), \mathbf{x}_{\text{dst}})] \\ &= \frac{1}{N} \sum_{i=1}^{N} \ell \left(f_{\boldsymbol{\theta}} \left(\mathbf{x}_{\text{src}}^{(i)}, \boldsymbol{\Delta}^{(i)} \right), \mathbf{x}_{\text{dst}}^{(i)} \right), \end{aligned}$$

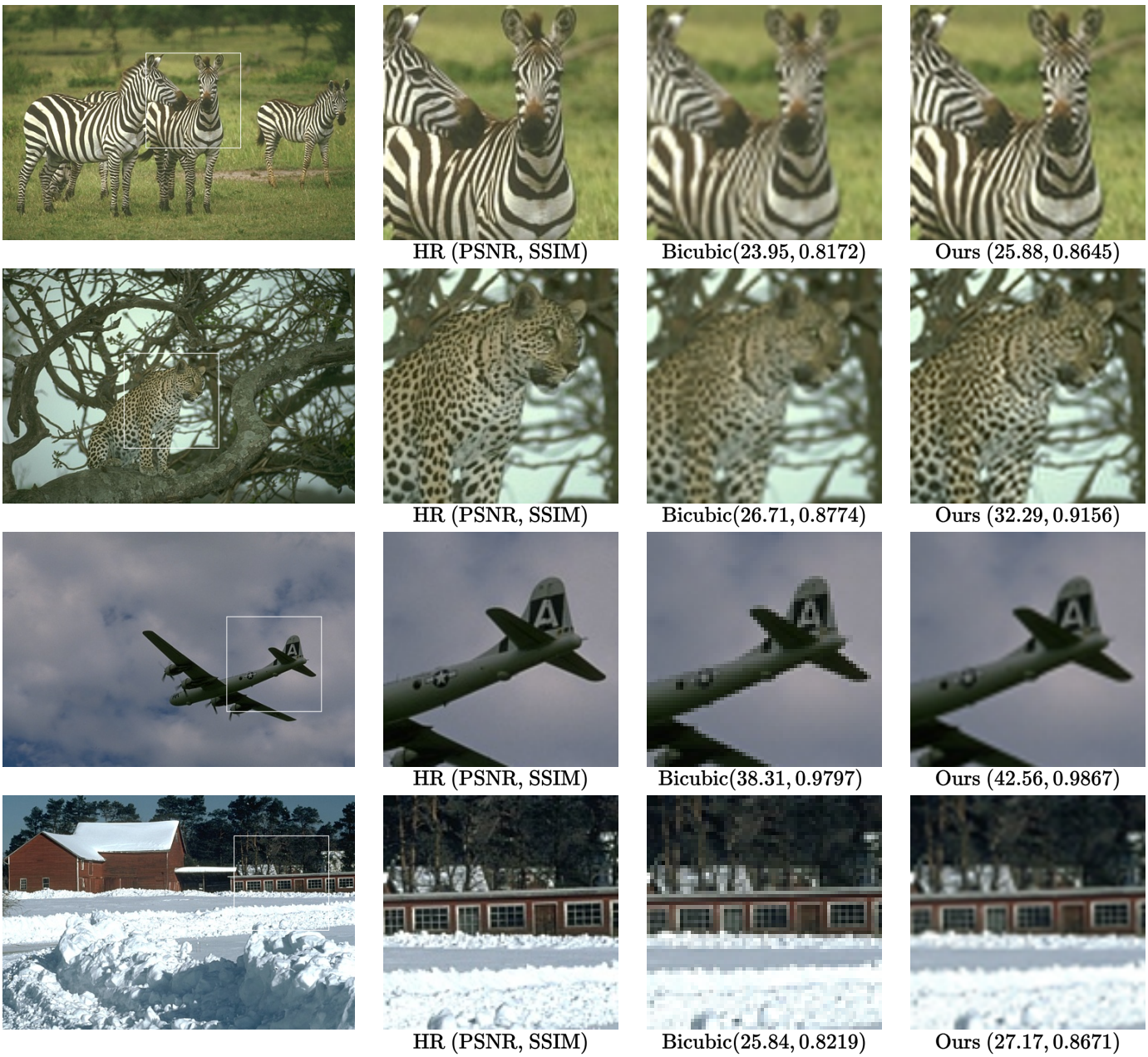

Fig. 3. The figure shows the result for 2x superesolution. The first column is the HR image, the 2nd column is a zoom-in of it. The third columns is bicubic, and the last is ours

where $\ell(\cdot, \cdot)$ is some loss function. The common gradient-based optimization can be use to iteratively update the parameter $\boldsymbol{\theta}$ through back-propagation.

For the choice of loss function, though any proper metric would satisfy, we go with the most common choice — L2 loss, i.e., $\ell(\mathbf{x}_1, \mathbf{x}_2) = \|\mathbf{x}_1 - \mathbf{x}_2\|_2^2$. We can re-write the loss

$$\mathcal{L}(\boldsymbol{\theta}) = \mathbb{E}_{\mathcal{D}} \left\| f_{\boldsymbol{\theta}}(\mathbf{x}_{\text{src}}, \boldsymbol{\Delta}) - \mathbf{x}_{\text{dst}} \right\|_2^2 .$$

Note that the regression method has the effect of taking the mean of target in the dataset as prediction [21], since its output is deterministic. This behaviour may be somewhat acceptable when the distribution is unimodal. On the other hand, by examining natural images, we can intuitively deduce that when the deltas are relatively small, i.e., when we pair patches from close neighborhoods, the variety and distribution of the destination patch are more likely to be simple. However, this observation does not hold for larger deltas. This insight serves as the motivation for our preference to select patch pairs from close neighborhoods and to constrain the delta to be relatively small when constructing the dataset from the pyramid in practice.

4 Experimental Results

4.1 Super-Resolution

Increase the resolution of an given image by scale factor $s > 1$. For a given image, we construct a pyramid where two consecutive layers differ by a small scale factor, such as $1/0.95$, with the minimum scale being about half the size of the given image. We train the neural network to map from lower

Fig. 4. Denoising Results. The first column shows clean image, second is noisy with addtive noise of $\sigma = 25$, the thired is BM3D [15] results, and the last is ours. Our algorithm, though we improve the psnr from noisy image, our method suffer from bad performance in uniform region

resolution (LR) patches to higher resolution (HR) patches within the pyramid. Specifically, for delta $\mathbf{\Delta} = (u, v, s_x, s_y)$, the u, v will be some subpixel displacements to align two patches across the scales, and s_x and s_y will be the ratio of HR to LR, we experimentally choose it to be $1/0.95$. Intuitively, the trained MLP can map a small patch in a lower resolution image to a patch (of the same size) in a slightly higher resolution image. By recursively applying this small factor super-resolution, we can reach a factor of 2.

At inference time, we recursively upsample the image by a factor of roughly $1/0.95$. We extract patches from the last predicted image (initially from the original lower resolution image) and obtain higher resolution patches by feeding the source patch and corresponding delta into the MLP. To construct an image prediction, we simply average all the spatially overlapping predicted patches. We also apply the back-projection algorithm to improve the result. Specifically, after obtaining an initial predicted HR image, we perform the following steps:

(1) Downscale it to the LR scale using an interpolation

method and calculate the difference to obtain an error map.
(2) Upscale the error map back to the predicted image scale.
(3) Use the upscaled error map to correct the initial predicted HR image.

This can be done iteratively, but we just done this once for every intermidiate predicted image.

We use the BSD100 dataset for our experiments. Our results, compared with naive Bicubic interpolation upsampling, show overall better PSNR and SSIM values. [Fig. 3] All the model outputs are trained after 200 epochs, where each epoch the model processes all the patch pairs from the pyramid once. The MLP is 256×6 in size, with residual connections.

4.2 Denoising

Remove the noise from an image and recover the underlaying clean signal. For the patch dataset, we stick with the original image scale only. We train a neural network to reconstruct patches themselves at the original scale of the image. That is, the pair of source and destination patches are the same, $\mathbf{x}_{\text{src}} = \mathbf{x}_{\text{dst}}$ and delta are just zero $\mathbf{\Delta} = \mathbf{0}$. At evaluation time, we just feed the noisy patches as source, and use the output patches as denoised patch. Similarly to super-resolution, we will average all the overlapping output patches to construct the image as denoised output. The intuition is that for a smooth function, if the input is similar, so are their corresponding values. This allows us to average patches in the target space (the reconstructed patches), globally for all spatial locations. We find that using a deterministic auto-encoder architecture improves the evaluation metric (PSNR) compared to the simple MLP, so we adopt it.

The dataset we used is Set 12 [22]. We corrupt the clean image with an addtive noise from $\mathcal{N}(0, \sigma^2)$ where $\sigma = 25$. The used auto-encoder is small-sized, 128×2 MLP for each of encoder and decorder, separately. An example result can be seen in [Fig. 4]. Our method, though we denoise the original image, especially at regions with rich texture, struggles with the uniform regions. This may be a result that the function do not recognize similar noisy patches are "neighbours", as to functions those noise is large enough to distinguish two patches.

5 Discussion & Limitation

In this section, we will briefly talk about the issues we encounter during the process of discovering the model's ability.

Super-Resolution. We observe an artifact of error accumulation in our model when we remove the back-projection component from the entire algorithm. This means that our prediction is biased at every small step towards the final HR image, but we have many such steps, making the bias larger and larger. This aligns with error accumulation observed in many auto-regressive models, as pointed out in [23]. At its core, our algorithm is auto-regressive (though deterministic) since our predictions are recursively fed back into the model as inputs to make the next predictions. Although the issue is addressed by back-projection, we believe there are many alternative options that could potentially work even better.

Denoising. Our current method does not utilize any cross-scale patch relationships to refine the prediction. Given the observation that clean signals are hidden at lower resolutions of the image as demonstrated in [16], we could attempt to incorporate this insight into our denoising algorithm. Moreover, we can extend our algorithm to align with the Noise2Noise approach [24], but applied at the patch level.

Generative Model. Most importantly, our current model relies on a deterministic mapping from the source patch with delta to the destination patch. However, we should acknowledge that the real underlying patch relationship is probabilistic. That is, given a source patch and delta, there should be a distribution of destination patches. In other words, we should instead model the underlying conditional distribution:

$$p(\mathbf{x}_{\text{dst}} \mid \mathbf{x}_{\text{src}}, \mathbf{\Delta}). \tag{6}$$

Our deterministic models can have limitations, especially when it comes to capturing the inherent uncertainty or variability in the underlying data. Generative models that incorporate randomness, like probabilistic autoregressive models, can better capture this uncertainty and produce a variety of plausible outputs given the same input.

Acknowledgments

The authors would like to thank Kyros Kutulakos, David Lindell, and Wenzhen Chen for their patient guidance and tremendous support.

References

[1] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Networks*, vol. 2, no. 5, pp. 359–366, 1989. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0893608089900208
[2] A. Buades, B. Coll, and J.-M. Morel, "A non-local algorithm for image denoising," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 2, 2005, pp. 60–65 vol. 2.
[3] W. Freeman, T. Jones, and E. Pasztor, "Example-based super-resolution," *IEEE Computer Graphics and Applications*, vol. 22, no. 2, pp. 56–65, 2002.
[4] D. Glasner, S. Bagon, and M. Irani, "Super-resolution from a single image," in *2009 IEEE 12th International Conference on Computer Vision*, 2009, pp. 349–356.
[5] C. Barnes, E. Shechtman, A. Finkelstein, and D. B. Goldman, "PatchMatch: A randomized correspondence algorithm for structural image editing," *ACM Transactions on Graphics (Proc. SIGGRAPH)*, vol. 28, no. 3, Aug. 2009.
[6] C. Barnes, E. Shechtman, D. B. Goldman, and A. Finkelstein, "The generalized PatchMatch correspondence algorithm," in *European Conference on Computer Vision*, Sep. 2010.
[7] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," 2017.
[8] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16x16 words: Transformers for image recognition at scale," *ICLR*, 2021.
[9] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, "Masked autoencoders are scalable vision learners," 2021.
[10] Z. Chen and H. Zhang, "Learning implicit fields for generative shape modeling," *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

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

[12] Y. Chen, S. Liu, and X. Wang, "Learning continuous image representation with local implicit image function," *arXiv preprint arXiv:2012.09161*, 2020.

[13] J. N. Martel, D. B. Lindell, C. Z. Lin, E. R. Chan, M. Monteiro, and G. Wetzstein, "Acorn: Adaptive coordinate networks for neural representation," 2021.

[14] T. Müller, A. Evans, C. Schied, and A. Keller, "Instant neural graphics primitives with a multiresolution hash encoding," *ACM Trans. Graph.*, vol. 41, no. 4, pp. 102:1–102:15, Jul. 2022. [Online]. Available: https://doi.org/10.1145/3528223.3530127

[15] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising by sparse 3-d transform-domain collaborative filtering," *IEEE Transactions on Image Processing*, vol. 16, no. 8, pp. 2080–2095, 2007.

[16] M. Zontak, I. Mosseri, and M. Irani, "Separating signal from noise using patch recurrence across scales," in *2013 IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 1195–1202.

[17] J.-B. Huang, A. Singh, and N. Ahuja, "Single image super-resolution from transformed self-exemplars," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 5197–5206.

[18] D. Ulyanov, A. Vedaldi, and V. Lempitsky, "Deep image prior," *arXiv:1711.10925*, 2017.

[19] M. I. Assaf Shocher, Nadav Cohen, ""zero-shot" super-resolution using deep internal learning," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[20] J. W. Soh, S. Cho, and N. I. Cho, "Meta-transfer learning for zero-shot super-resolution," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 3513–3522.

[21] M. H. Kutner, C. J. Nachtsheim, J. Neter, and W. Li, *Applied Linear Statistical Models*, 5th ed. McGraw-Hill/Irwin, 2005.

[22] D. Martin, C. Fowlkes, D. Tal, and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*, vol. 2, 2001, pp. 416–423 vol.2.

[23] S. Bengio, O. Vinyals, N. Jaitly, and N. Shazeer, "Scheduled sampling for sequence prediction with recurrent neural networks," 2015.

[24] J. Lehtinen, J. Munkberg, J. Hasselgren, S. Laine, T. Karras, M. Aittala, and T. Aila, "Noise2Noise: Learning image restoration without clean data," in *Proceedings of the 35th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, J. Dy and A. Krause, Eds., vol. 80. PMLR, 10–15 Jul 2018, pp. 2965–2974. [Online]. Available: https://proceedings.mlr.press/v80/lehtinen18a.html