Learning Continuous Implicit Representation for Near-Periodic Patterns

Bowei Chen[1], Tiancheng Zhi[1], Martial Hebert[1], and Srinivasa G. Narasimhan[1]

Carnegie Mellon University, Pittsburgh PA 15213, USA
{boweiche,tzhi,mhebert,srinivas}@andrew.cmu.edu

Table of Contents

1 Dataset

For PSU Near-Regular Texture Database (NRTDB), we collect all the perspective NPP images, and filter those that fail to be rectified using TILT [42]. The resultant dataset has 165 NPP images from PSU Near-Regular Texture Database (NRTDB), which contains facades, friezes, bricks, fences, grounds, Mondrian images, wallpapers, and carpets.

For Describable Textures Dataset (DTD), we only consider the official test set (the first split) since some of our baselines are trained on DTD [8]. In the test set, we select the categories whose images are more likely to be near-periodic. These categories are banded, chequered, grid, honeycombed, lined, meshed, perforated, polka-dotted, zigzagged. Then we manually filter out non-NPP images, resulting in 258 NPP images in the dataset.

For the Facade dataset, we directly use an official CVPR 2010 subset provided by [30], which contains 109 rectified images of facades. Some of these facades are strictly *not* NPP because often the windows are not arranged periodically. But nonetheless we include these to evaluate our approach when the NPP assumption is not strictly satisfied.

The sampled images of NRTDB, DTD, Facade datasets are shown in Figure 1, Figure 2 and Figure 3, respectively. For our self-collected dataset, we show all of them in qualitative results.

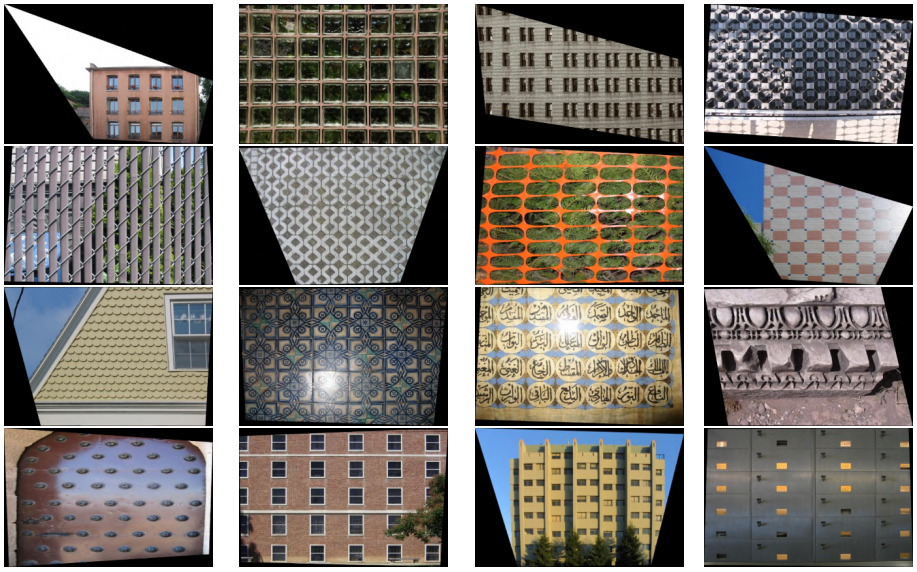


Fig. 1: Sampled NPP images (rectified) of the NRTDB dataset. All examples are resized for the visualization. More images are included in the qualitative comparison.

Fig. 2: Sampled NPP images (rectified) of the DTD dataset. All examples are resized for the visualization. More images are included in the qualitative comparison.

Fig. 3: Sampled NPP images (rectified) of the Facade dataset. All examples are resized for the visualization. More images are included in the qualitative comparison.

2 NPP-Net

2.1 Displacement Vectors Transformation

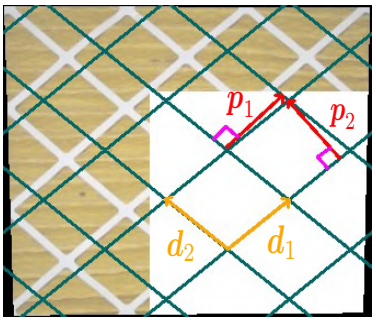


Fig. 4: The illustration of a periodicity. It consists of two displacement vectors $\boldsymbol{d}_1$ and $\boldsymbol{d}_2$ (orange), which can be transformed to periodicity vectors $\boldsymbol{p}_1$ and $\boldsymbol{p}_2$ (red), respectively.

Lattice Patterns: Assuming a 2D lattice arrangement, a periodicity can be represented as two displacement vectors $\boldsymbol{d}_1$ and $\boldsymbol{d}_2$ (orange arrows in Figure 4). A perfect infinite periodic pattern is invariant if shifted by $\alpha\boldsymbol{d}_1 + \beta\boldsymbol{d}_2 (\alpha, \beta \in \mathbb{Z})$. In order to incorporate the displacement vectors into the Periodicity-Aware Input Warping module, we need to transform $\boldsymbol{d}_1$ and $\boldsymbol{d}_2$ into pattern periods and orientations, visualized as the magnitudes and orientations of the red arrows ($\boldsymbol{p}_1$ and $\boldsymbol{p}_2$, called **periodicity vectors**).

We assume $\boldsymbol{d}_1 = (d_{1x}, d_{1y})$ and $\boldsymbol{d}_2 = (d_{2x}, d_{2y})$ are known (from Periodicity Proposal). Note that we let $d_{1x}, d_{2x} \in \mathbb{Z}$ and $d_{1y}, d_{2y} \in \mathbb{N}$ to remove identical vectors in another half circle. As we mentioned in the main paper, we can obtain the periodicity vectors by solving:

$$\boldsymbol{p}_1 \cdot \boldsymbol{d}_2 = \boldsymbol{d}_1 \cdot \boldsymbol{p}_2 = \boldsymbol{0}. \tag{1}$$

$$\boldsymbol{p}_1 \times \boldsymbol{d}_2 = \boldsymbol{d}_1 \times \boldsymbol{p}_2 = \boldsymbol{d}_1 \times \boldsymbol{d}_2, \tag{2}$$

where the cross product is defined using the corresponding 3D vectors on the plane $z = 0$.

Let $\boldsymbol{p_1} = (p_1 \cos\theta_1, p_1 \sin\theta_1)$, where θ_1 is in the range $[0, \pi)$. We can compute θ_1 according to Equation 1, given by:

$$\begin{aligned} \boldsymbol{p_1} \cdot \boldsymbol{d_2} = p_1 \cos\theta_1 d_{2x} + p_1 \sin\theta_1 d_{2y} &= 0 \\ \tan\theta_1 &= -\frac{d_{2x}}{d_{2y}} \\ \theta_1 &= \arctan -\frac{d_{2x}}{d_{2y}} \end{aligned} \tag{3}$$

Then the magnitude p_1 can be solved based on Equation 2, given by:

$$\begin{aligned} \boldsymbol{p_1} \times \boldsymbol{d_2} &= \boldsymbol{d_1} \times \boldsymbol{d_2} \\ p_1 |d_2| \sin\frac{\pi}{2} &= |d_1||d_2| \sin\phi \\ p_1 &= \sqrt{(d_{1x})^2 + (d_{1y})^2} \sin\phi \end{aligned} \tag{4}$$

where $\phi = \arccos \frac{\boldsymbol{d_1} \cdot \boldsymbol{d_2}}{|d_1| \cdot |d_2|}$ is the angle between $\boldsymbol{d_1}$ and $\boldsymbol{d_2}$. Similarly, we can obtain $\boldsymbol{p_2}$ using Equation 1 to Equation 4.

Circular Patterns: Our method can be modified to handle circular periodic patterns. Define periodicity of circular NPP by a rotation centroid $\boldsymbol{c} = (c_x, c_y)$ and an angular period p (in radians). The periodicity proposal module can be applied to circular NPP to estimate $\boldsymbol{c}$ and p. In addition, two modifications are needed:

(1) Equation 1 (main paper) is rewritten as two functions:

$$\begin{aligned} f_{\boldsymbol{c},p}(x, y) &= \sqrt{(x - c_x)^2 + (y - c_y)^2} \\ g_{\boldsymbol{c},p}(x, y) &= atan2^*(y - c_y, x - c_x) \mod p \end{aligned} \tag{5}$$

where we define $atan2^*(y, x) = (atan2(y, x) + 2\pi) \mod 2\pi$.

(2) Periodicity-based patch sampling strategy is modified. Let center of predicted patch be $\boldsymbol{x}$. Instead of shifting $\boldsymbol{x}$ to obtain centers of known patch $\boldsymbol{x}_{\alpha\beta}$ in lattice patterns, we directly rotate the input image around $\boldsymbol{c}$ based on $\alpha \cdot p$, where α is an integer constant. Then we sample known patch centers in circular images at position $\boldsymbol{x}$.

2.2 Implementation Details

We apply the following settings below (for the final pipeline) for all applications. Same as [20], we set the number of frequency d in positional encoding as 10.

Periodicity Proposal. In our implementation, the key hyperparameter for periodicity detection method [19] is an integer q, which defines the range of possible displacements in $\boldsymbol{d}_1$ and $\boldsymbol{d}_2$. Specifically, $\boldsymbol{d}_1$ and $\boldsymbol{d}_2$ are in the set R, given by:

$$R = \{(x, y) | -\frac{W}{q} < x < \frac{W}{q}, 0 \le y < \frac{H}{q}\} - \{(x, y) | -\frac{W}{q+1} < x < \frac{W}{q+1}, 0 \le y < \frac{H}{q+1}\}, \tag{6}$$

where H and W are the image height and width. The periodicity detection method outputs the best periodicity in R.

In the periodicity searching, we evaluate different q in $\{i|i \in \mathbb{Z}^+, i < 10\}$ for multiple periodicities, which are ranked based on reconstruction errors. Specifically, we generate Top-M (M=3) pseudo square masks whose centers are far away from the image boundary or unknown regions. The mask size for each mask is empirically set to $\frac{5L}{6\sqrt{2}}$, where L is the distance from the center pixel to the nearest invalid pixel. Then we run the initial pipeline for evaluation of each q based on its reconstruction error in these masked regions.

Network Architecture. The first MLP contains 9 fully-connected Snake layers [44] with 512 channels. It also includes a skip connection that concatenates the Top-1 coordinate and original features to the fifth layer's activation. The outputs of this MLP are concatenated with the Top 2nd to K-th coordinate features and then fed to the second MLP. The second MLP has 4 fully-connected Snake layers with 512, 512, 256, and 3 channels. The output features of the first MLP are also concatenated with the second layer activation of the second MLP for the skip connection.

Single-Image Optimization. In the patch loss, we sample square ground truth patches for supervision. We set the patch size s in $\{64, 96, 128, 160\}$ based on the Top-1 periodicity, where larger periods are matched with a larger patch size. For the patch center $\boldsymbol{x}$, we filter out the shifted patch centers $\boldsymbol{x}_{\alpha\beta}$ with their known patch area smaller than 70% of the whole patch area, and we use $N = 3$ (size of T_N). We set λ_p in $\mathcal{L}_{patch}$ to 0.4, and λ_c to 1, 10, 5 in completion, remapping and segmentation (explained later), respectively. In final Loss $\mathcal{L}$, we set $\lambda_1 = 1$, $\lambda_2 = 0.001$, $|B_1| = 8192$, and $|B_2| = 2$. Further, B_2 contains half of sampled centers in unknown regions and half of those in known regions.

For training details, we use the Adam optimizer with the learning rate that starts with 5×10^{-4} and decays every 500 epochs. Furthermore, for every 2000 epoch, we shrink the patch size by twice and increase the number of the sampled patches by twice to encourage the network to focus on finer-level details.

Runtime. The periodicity proposal takes 100 seconds (for 10 candidate periodicities). The optimization takes around 8 minutes, 5 minutes, and 3 minutes for NPP completion, NPP remapping, and NPP segmentation to converge on a single NVIDIA Titan XP GPU, respectively.

3 NPP Completion

3.1 Mask Generation

We apply different mask generation strategies to test the methods on different scales. For each image in NRTDB [1], we generate an unknown mask with height and width sampled from $[100, 500]$. The top-most and left-most coordinates of the mask are sampled from $[\frac{H}{2} - 250, \frac{H}{2} + 250]$ and $[\frac{W}{2} - 250, \frac{W}{2} + 250]$, respectively. We crop the mask when it exceeds the image boundary. For each image in DTD [8], we generate an unknown mask with height and width equal to 70% of the image height and width (starting from the bottom right) respectively. This mask generation strategy allows us to evaluate the extrapolation performance of the methods. For each image in Facade, we generate a mask in the center of the image, where the mask size is $(\frac{H}{6}, \frac{W}{3})$.

3.2 More Metrics

We also evaluate the models using FID [21] and RMSE. FID evaluates the distance between the distribution of ground truth and outputs images at the dataset level. But, FID requires all the images to be resized to a fixed size. Thus for images with low resolution, this resizing operation introduces aliasing even if we use the recent proposed clean FID implementation [21]. RMSE is a pixel-wise metric to compare the difference of pixel values.

3.3 Ablation Study

More results for experiments in main paper: We show the quantitative results with FID and RMSE (evaluated only in unknown regions) and all metrics (evaluated in the full images) in Table 1, Table 2 and Table 3. Note that "No Periodicity" variant simply overfits the known regions without learning any structural information. Also, the facade dataset may have different performance because it contains some images that are strictly not NPP images. Figure 5, Figure 6, and Figure 7 show the qualitative results for variants without periodicity prior and those with various combination of loss functions. Figure 8 shows the comparison with variants using different settings of periodicity augmentation. In summary, NPP-Net outperforms all the tested variants on NRTDB and DTD datasets.

More experiments for choice of loss: We further tested three variants of NPP-Net in NRTDB: (1) replace robust L2 loss by L2 loss, (2) replace contextual loss by perceptual loss, (3) replace perceptual loss by contextual loss. The LPIPS of variants (1)(2)(3) are 0.206, 0.226, and 0.209. They are still worse than the full model (0.188).

Choice of activation function: We replaced SNAKE activation function in our MLP with Sine function (following SIREN), and the LPIPS in NRTDB is 0.194 (3% worse). This is because the periodic activation function in SIREN helps in handling fine details but not necessarily complex periodic patterns.

Category	Method	Only Unknown Regions		Full Images				
		RMSE ↓	FID ↓	LPIPS ↓	SSIM ↑	PSNR↑	RMSE ↓	FID ↓
NPP-Net Variants	No Periodicity	8.890	223.7	0.134	0.861	23.34	**4.526**	47.85
	Pixel Only	8.703	153.5	0.180	0.859	23.65	5.756	47.84
	Patch Only	9.270	128.1	0.313	0.426	15.98	8.698	98.41
	Pixel + Random	8.241	100.0	0.104	0.902	24.52	4.770	26.06
	Initial Pipeline	8.246	81.99	0.124	0.893	24.29	5.160	27.46
	Top1 + Offsets	8.228	74.83	0.109	0.900	24.48	4.856	23.51
	Top5 + Offsets	8.266	74.83	**0.100**	0.900	24.55	4.721	23.08
	Top3 w/o Offsets	8.259	78.20	0.119	0.892	24.31	5.038	26.01
NPP-Net	Top3 + Offsets	**8.164**	**70.43**	**0.100**	**0.908**	**24.71**	4.839	**20.46**

Table 1: Comparing different variants of NPP-Net for NPP completion in NRTDB. NPP-Net outperforms all other variants. Note that "No Periodicity" variant overfits the known regions in the full images case.

Category	Method	Only Unknown Regions		Full Images				
		RMSE ↓	FID ↓	LPIPS ↓	SSIM ↑	PSNR↑	RMSE ↓	FID ↓
NPP-Net Variants	No Periodicity	8.758	186.3	0.256	0.661	19.35	**6.082**	86.97
	Pixel Only	8.400	145.7	0.263	0.685	20.43	6.545	81.80
	Patch Only	9.097	122.4	0.382	0.335	13.62	9.020	115.0
	Pixel + Random	7.981	89.26	0.173	0.742	20.85	6.202	54.09
	Initial Pipeline	8.100	96.64	0.212	0.704	20.00	6.516	65.91
	Top1 + Offsets	8.061	93.93	0.201	0.714	20.42	6.431	61.02
	Top5 + Offsets	8.024	86.97	0.171	0.730	20.85	6.264	50.97
	Top3 w/o Offsets	8.079	90.48	0.193	0.718	20.15	6.407	57.70
NPP-Net	Top3 + Offsets	**7.918**	**85.39**	**0.162**	**0.744**	**21.15**	6.161	**50.39**

Table 2: Comparing different variants of NPP-Net for NPP completion in DTD. NPP-Net outperforms all other variants.

Category	Method	Only Unknown Regions		Full Images				
		RMSE ↓	FID ↓	LPIPS ↓	SSIM ↑	PSNR↑	RMSE ↓	FID ↓
NPP-Net Variants	No Periodicity	9.597	264.3	**0.040**	**0.961**	**27.20**	**4.646**	**16.13**
	Pixel Only	9.902	281.6	0.137	0.914	24.80	7.196	53.69
	Patch Only	9.987	156.6	0.535	0.166	11.04	9.993	234.9
	Pixel + Random	9.540	132.7	0.073	0.939	25.43	6.289	18.15
	Initial Pipeline	**9.399**	102.1	0.092	0.932	24.70	6.607	18.65
	Top1 + Offsets	9.472	101.5	0.083	0.933	24.70	6.463	18.37
	Top5 + Offsets	9.474	92.34	0.067	0.936	25.55	6.149	17.14
	Top3 w/o Offsets	9.403	96.38	0.085	0.935	24.96	6.475	17.68
NPP-Net	Top3 + Offsets	9.464	**91.05**	0.063	0.944	25.51	6.155	16.94

Table 3: Comparing different variants of NPP-Net for NPP completion in Facade dataset. NPP-Net outperforms all other variants. Besides, the Facade dataset contains a number of non-NPP images, resulting in a different performance from the other two datasets. "No Periodicity" variant overfits the known regions in the full images case.

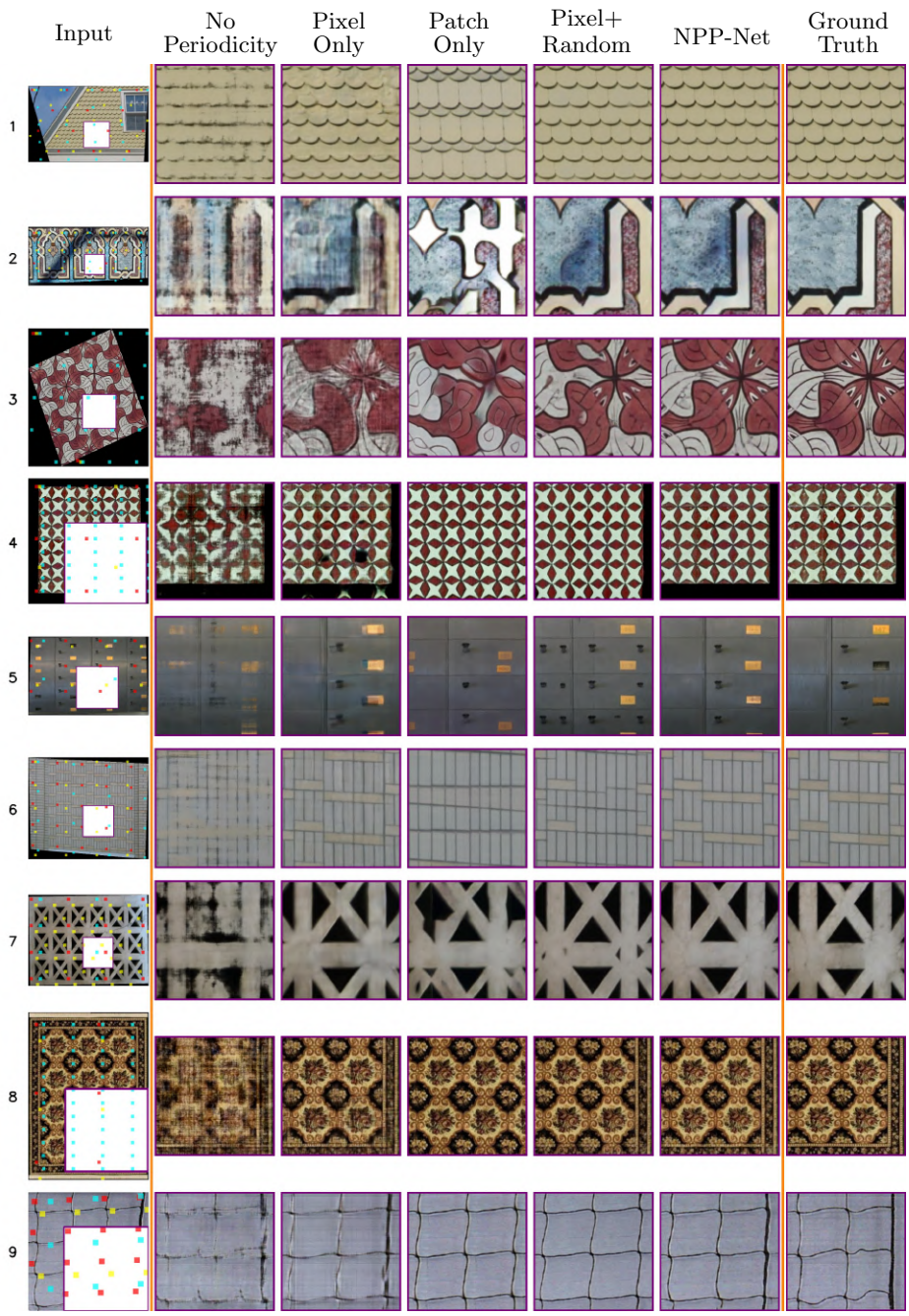


Fig. 5: Qualitative results for variants without periodicity prior and with different loss functions. NPP-Net outperforms all other variants. Note that periods in the fifth and seventh row are not scaled by 2.

Fig. 6: Qualitative results for variants without periodicity prior and with different loss function. NPP-Net outperforms all other variants.

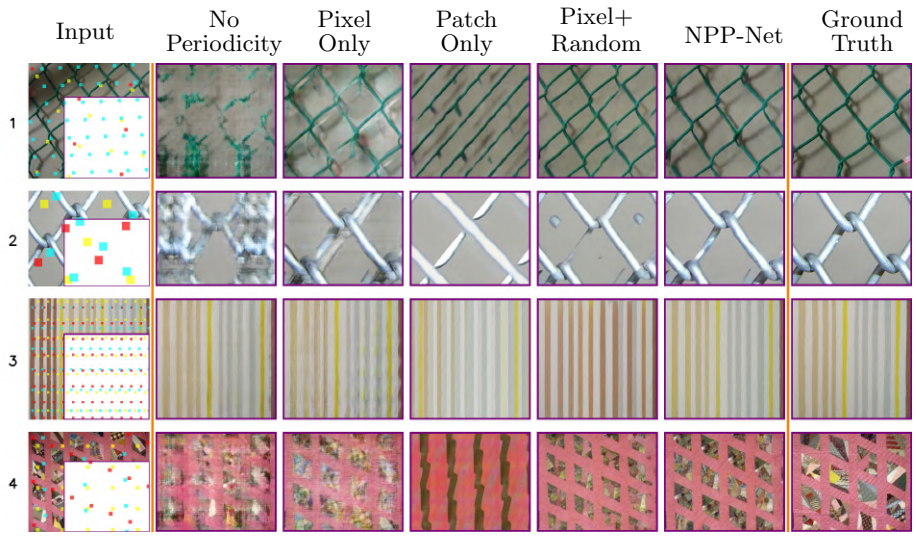


Fig. 7: Qualitative results for variants without periodicity prior and with different loss function. NPP-Net outperforms all other variants.

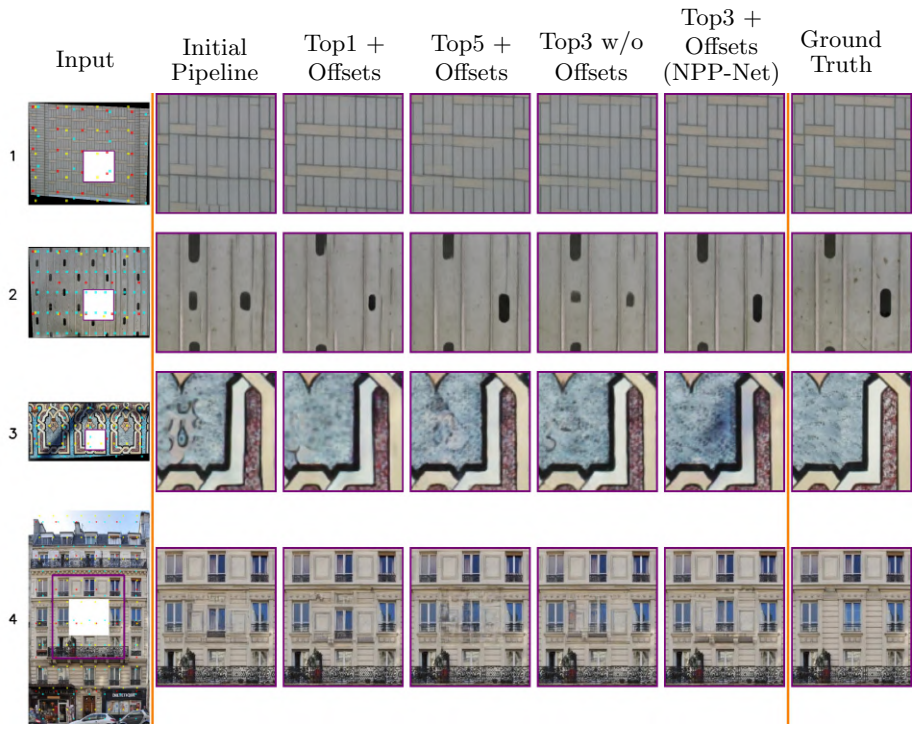


Fig. 8: Qualitative results for periodicity augmentation. NPP-Net outperforms all other variants.

3.4 Comparison with Baselines

We show the quantitative results with FID and RMSE (evaluated only in unknown regions) and all metrics (evaluated in the full images) in Table 4, Table 5, and Table 6. For NRTDB and DTD datasets, NPP-Net outperforms all the baselines. For RMSE in Table 4, DIP and Siren perform better in the full image even if they generate bad results in unknown regions because they simply overfit the known ones. For FID in Table 5, image quilting performs better. One possible reason is that the images in this dataset have smaller variations in the motifs thus simply tiling them may produce reasonable results.

The Facade dataset contains some images that are not strictly NPP images. Thus the performance is different from the other two datasets. But NPP-Net can still outperform other baselines (except for Lama that trained on large datasets) by optimizing only on a single image. Similarly, DIP and Siren perform better in the full image since they simply overfit the known ones.

We also show more qualitative results with all baselines. Figure 9 to Figure 14 show that NPP-Net outperforms baselines in terms of global consistency. Figure 15 and Figure 16 are for local variations, including boundaries and lighting effects. As shown in the results, NPP-Net can handle the NPP images that only have periodic patterns along one direction. This is because NPP-Net tends to focus on this direction while ignoring the second one since this minimizes the loss in the known regions better. Note that pretrained PEN-Net only accepts images with a fixed resolution, thus we resize the input image to satisfy the constraint, which may lead to blurry results. Lama also performs well since it implicitly learns scene prior, such as periodicity, from large datasets, while NPP-Net is only optimized on a single image.

In addition, We show the results in the main paper with all the baselines in Figure 17 and Figure 18. Besides, Figure 17 also shows more examples for extrapolation and different mask shapes. The results show NPP-Net outperforms baselines (especially for those single image-based counterparts) in terms of global consistency and local variations with various shapes and sizes of masks.

Please note that, although we minimizes LPIPS in known regions during training, we provides evaluation of all methods only in unknown regions (Table 1 in main paper). Even removing LPIPS loss, NPP-Net still obtains 0.204 LPIPS (24% better than the best baseline (BPI) that does not optimize LPIPS). In addition, NPP-Net performs the best in SSIM and FID (main paper and supp), which are not explicitly minimized by all methods.

Category	Method	Only Unknown Regions		Full Images				
		RMSE ↓	FID ↓	LPIPS ↓	SSIM ↑	PSNR↑	RMSE ↓	FID ↓
Large Datasets	PEN-Net [39]	9.231	204.1	-	-	-	-	-
	ProFill [40]	9.137	187.4	-	-	-	-	-
	Lama [29]	8.567	86.18	-	-	-	-	-
Single Image	Image Quilting [11]	9.739	89.53	-	-	-	-	-
	PatchMatch [3]	8.956	86.67	-	-	-	-	-
	DIP [31]	8.906	242.3	0.177	0.829	22.32	4.702	79.65
	Siren [25]	9.859	303.4	0.183	0.791	21.17	**4.418**	85.18
	Huang *et al.* [13]	8.921	104.7	-	-	-	-	-
	BPI [18]	9.213	71.77	-	-	-	-	-
	NPP-Net	**8.164**	**70.43**	**0.100**	**0.908**	**24.71**	4.839	**50.39**

Table 4: Comparing with different baselines for NPP completion in NRTDB dataset. NPP-Net outperforms all baselines.

Category	Method	Only Unknown Regions		Full Images				
		RMSE ↓	FID ↓	LPIPS ↓	SSIM ↑	PSNR↑	RMSE ↓	FID ↓
Large Datasets	PEN-Net [39]	8.952	166.5	-	-	-	-	-
	ProFill [40]	9.024	199.7	-	-	-	-	-
	Lama [29]	9.137	93.14	-	-	-	-	-
Single Image	Image Quilting [11]	9.431	**83.39**	-	-	-	-	-
	PatchMatch [3]	8.637	105.4	-	-	-	-	-
	DIP [31]	9.350	263.32	0.344	0.604	16.23	6.549	153.74
	Siren [25]	9.895	314.1	0.393	0.544	16.21	6.220	174.9
	Huang *et al.* [13]	8.520	97.97	-	-	-	-	-
	BPI [18]	8.910	85.93	-	-	-	-	-
	NPP-Net	**7.918**	85.39	**0.162**	**0.744**	**21.15**	**6.161**	**50.39**

Table 5: Comparing with different baselines for NPP completion in DTD dataset. NPP-Net outperforms all baselines.

Category	Method	Only Unknown Regions		Full Images				
		RMSE ↓	FID ↓	LPIPS ↓	SSIM ↑	PSNR↑	RMSE ↓	FID ↓
Large Datasets	PEN-Net [39]	9.745	133.3	-	-	-	-	-
	ProFill [40]	9.566	138.0	-	-	-	-	-
	Lama [29]	**9.434**	**83.07**	-	-	-	-	-
Single Image	Image Quilting [11]	10.16	121.46	-	-	-	-	-
	PatchMatch [3]	9.741	95.34	-	-	-	-	-
	DIP [31]	9.660	239.9	**0.055**	**0.951**	**26.71**	5.080	20.98
	Siren [25]	10.17	404.7	0.060	0.938	24.39	**2.730**	33.54
	Huang *et al.* [13]	9.447	104.9	-	-	-	-	-
	BPI [18]	9.992	98.52	-	-	-	-	-
	NPP-Net	9.464	99.50	0.063	0.944	25.51	6.155	**16.94**

Table 6: Comparing with different baselines for NPP completion in Facade dataset. Note that, Facade dataset has different performance since it contains some non-NPP images.

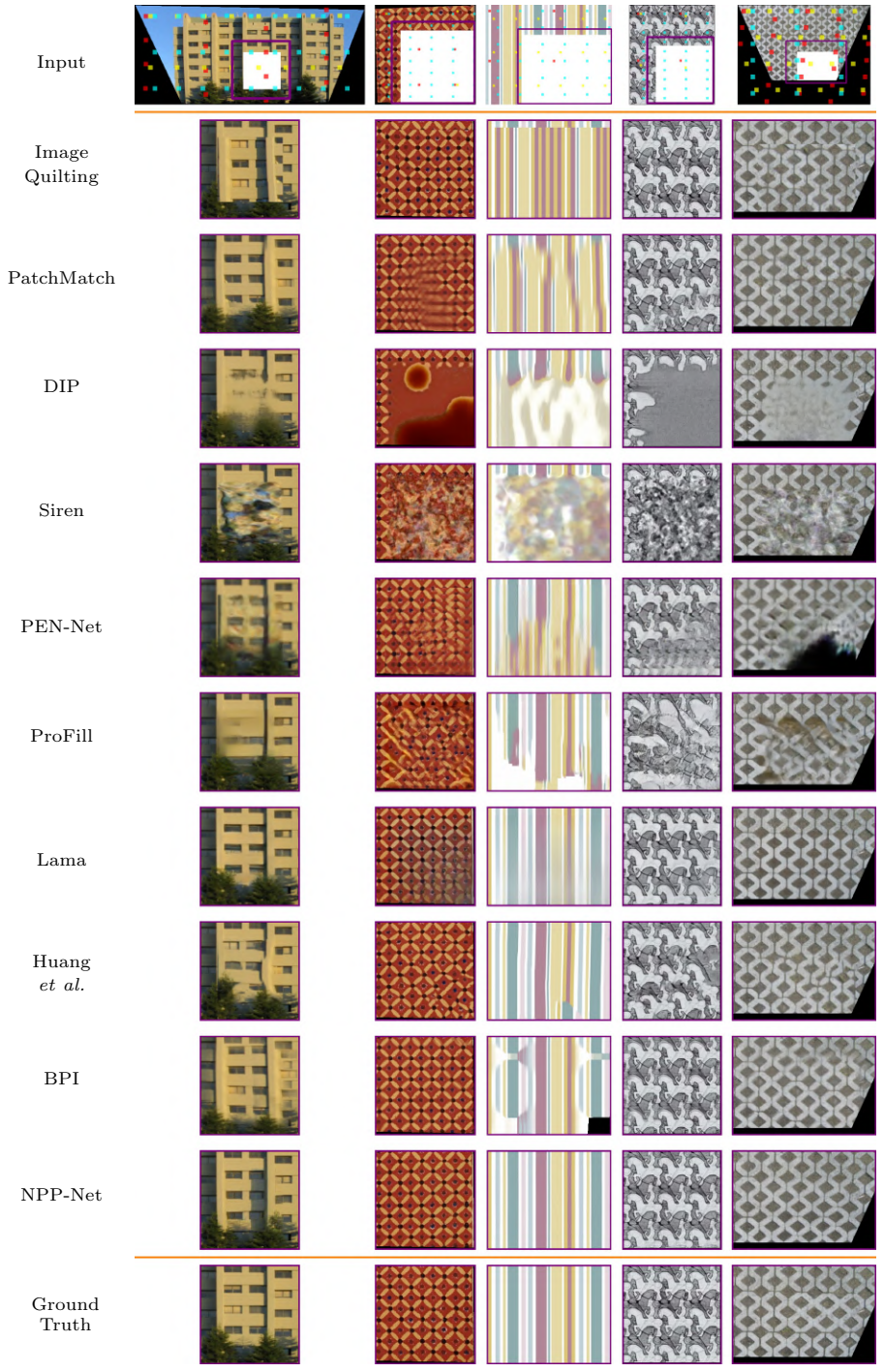


Fig. 9: Comparison with other baselines for image completion.

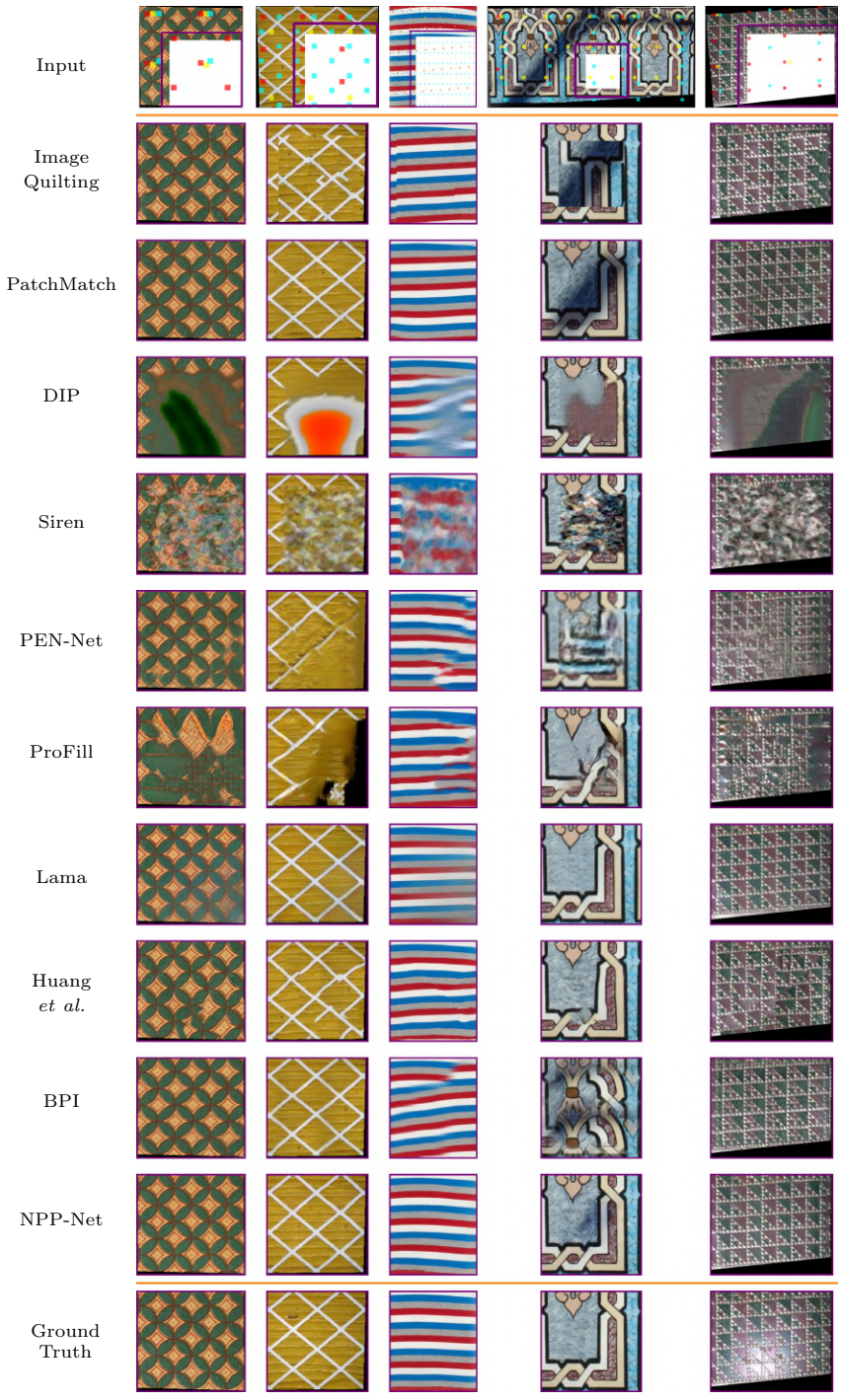


Fig. 10: Comparison with other baselines for image completion. Note that periods in the second column are not scaled by 2.

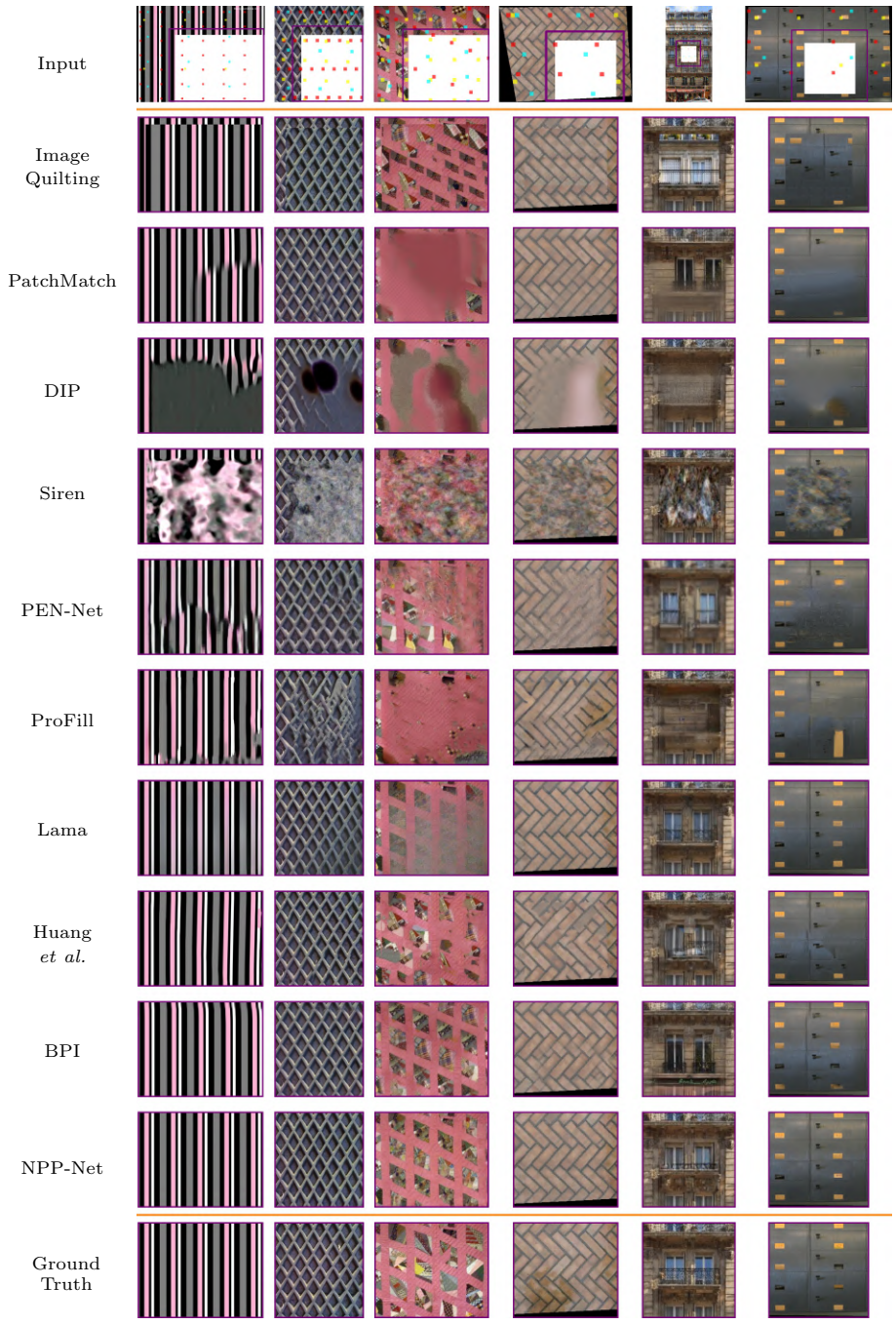


Fig. 11: Comparison with other baselines for image completion. Note that periods in the fifth column are not scaled by 2.

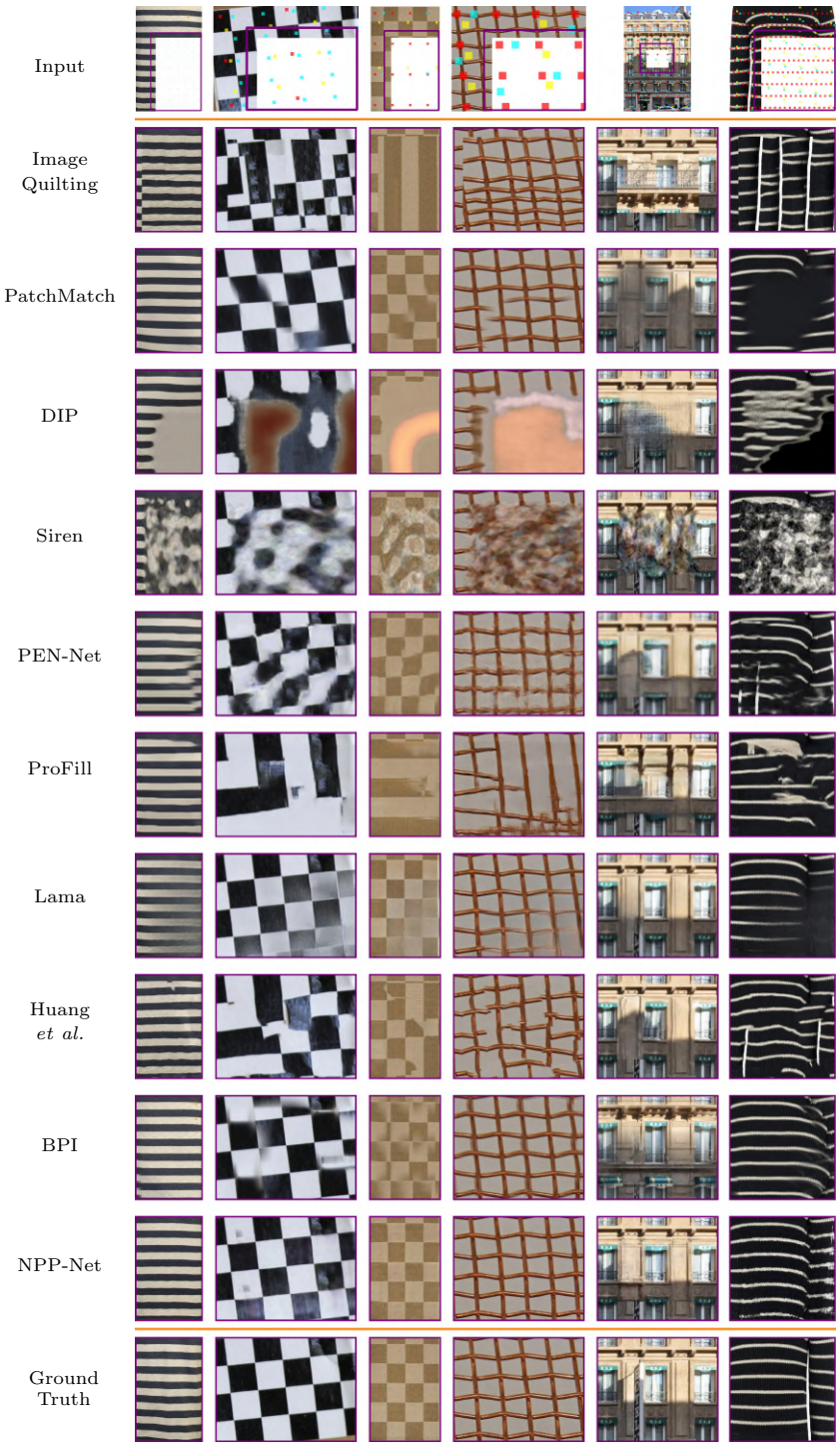


Fig. 12: Comparison with other baselines for image completion.

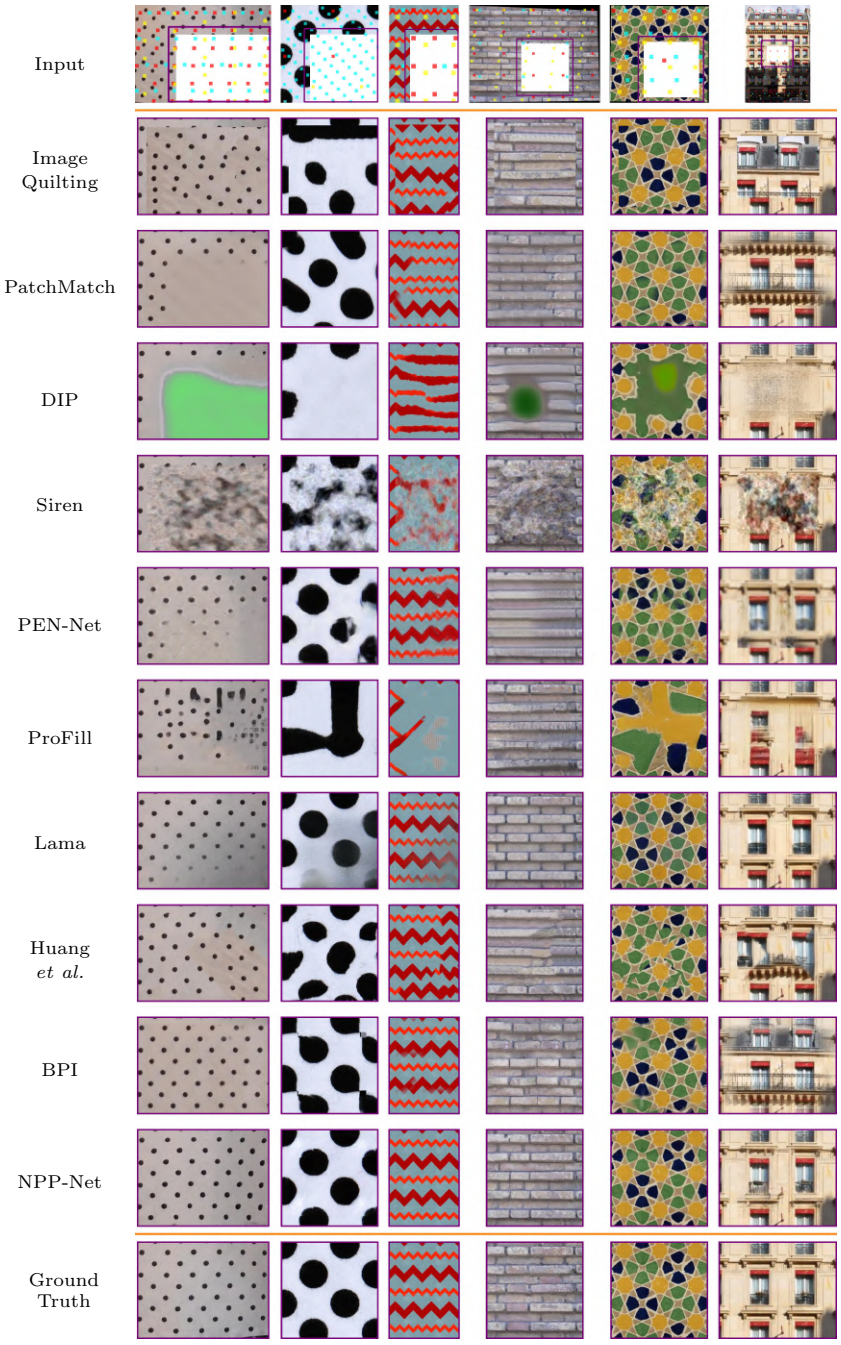


Fig. 13: Comparison with other baselines for image completion. Note that periods in the fifth column are not scaled by 2.

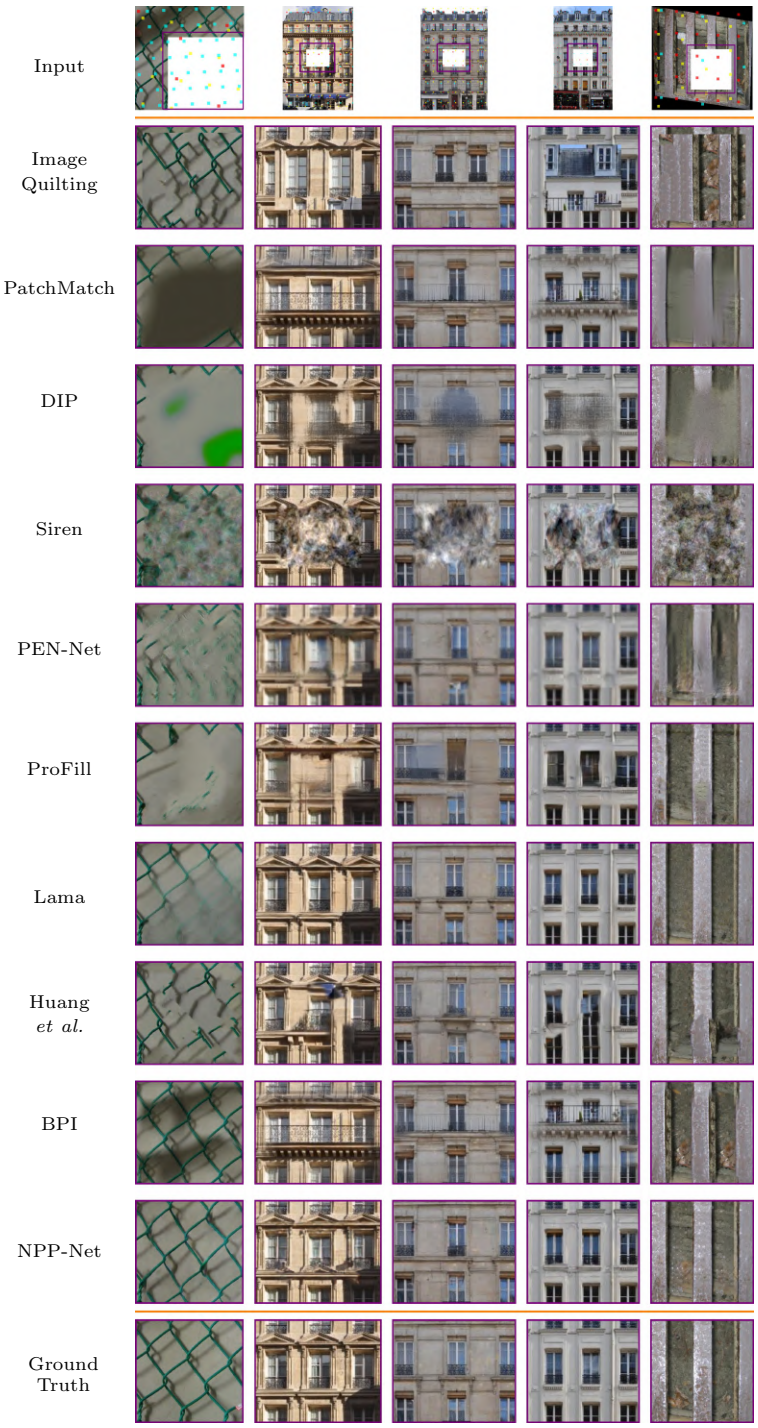


Fig. 14: Comparison with other baselines for image completion.

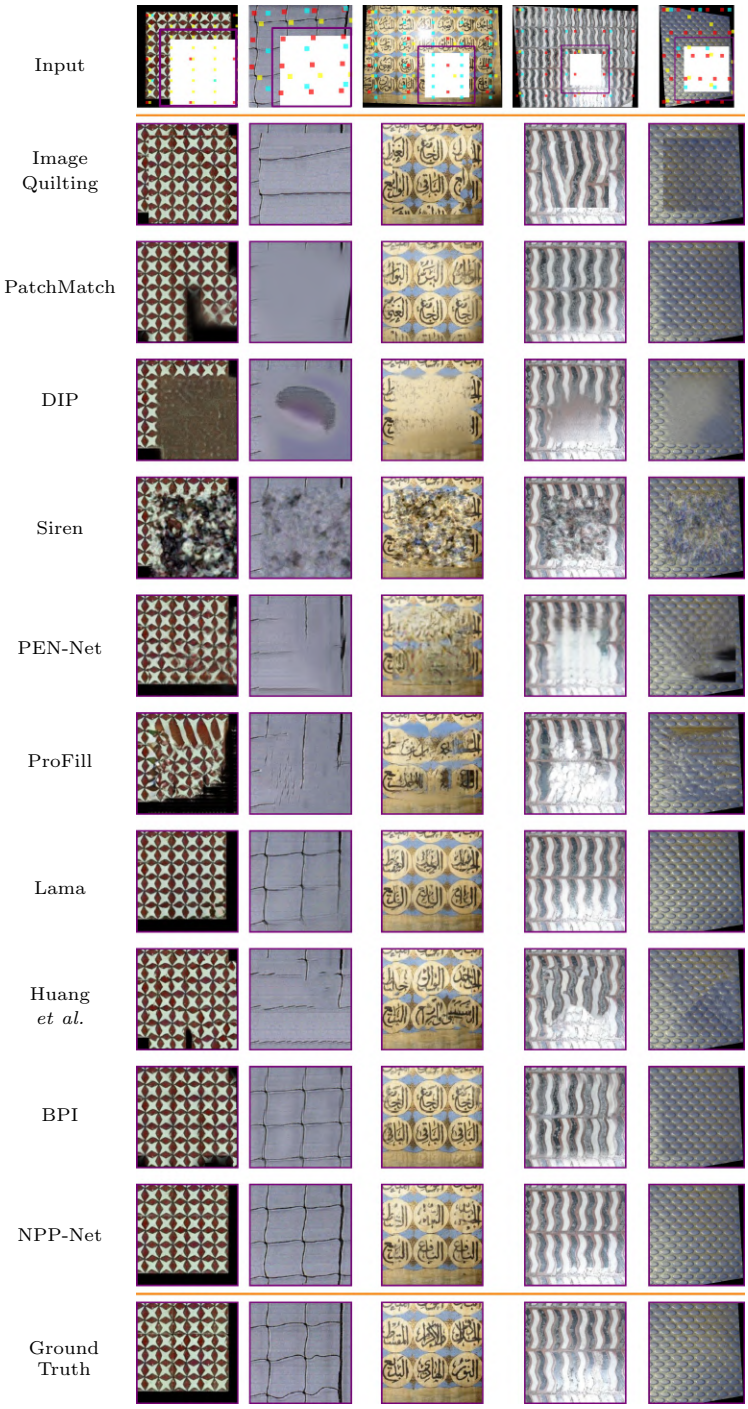


Fig. 15: Comparison with other baselines for image completion.

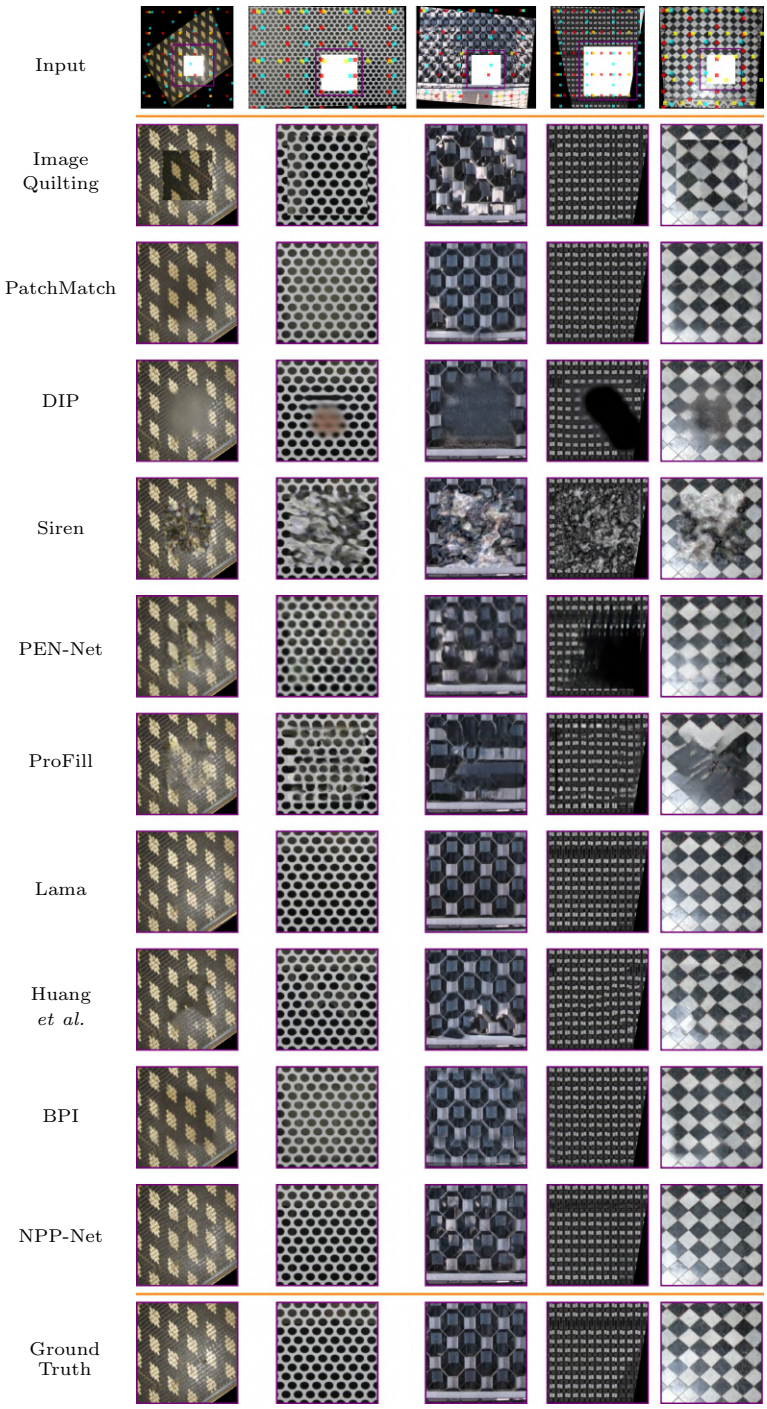


Fig. 16: Comparison with other baselines for image completion.

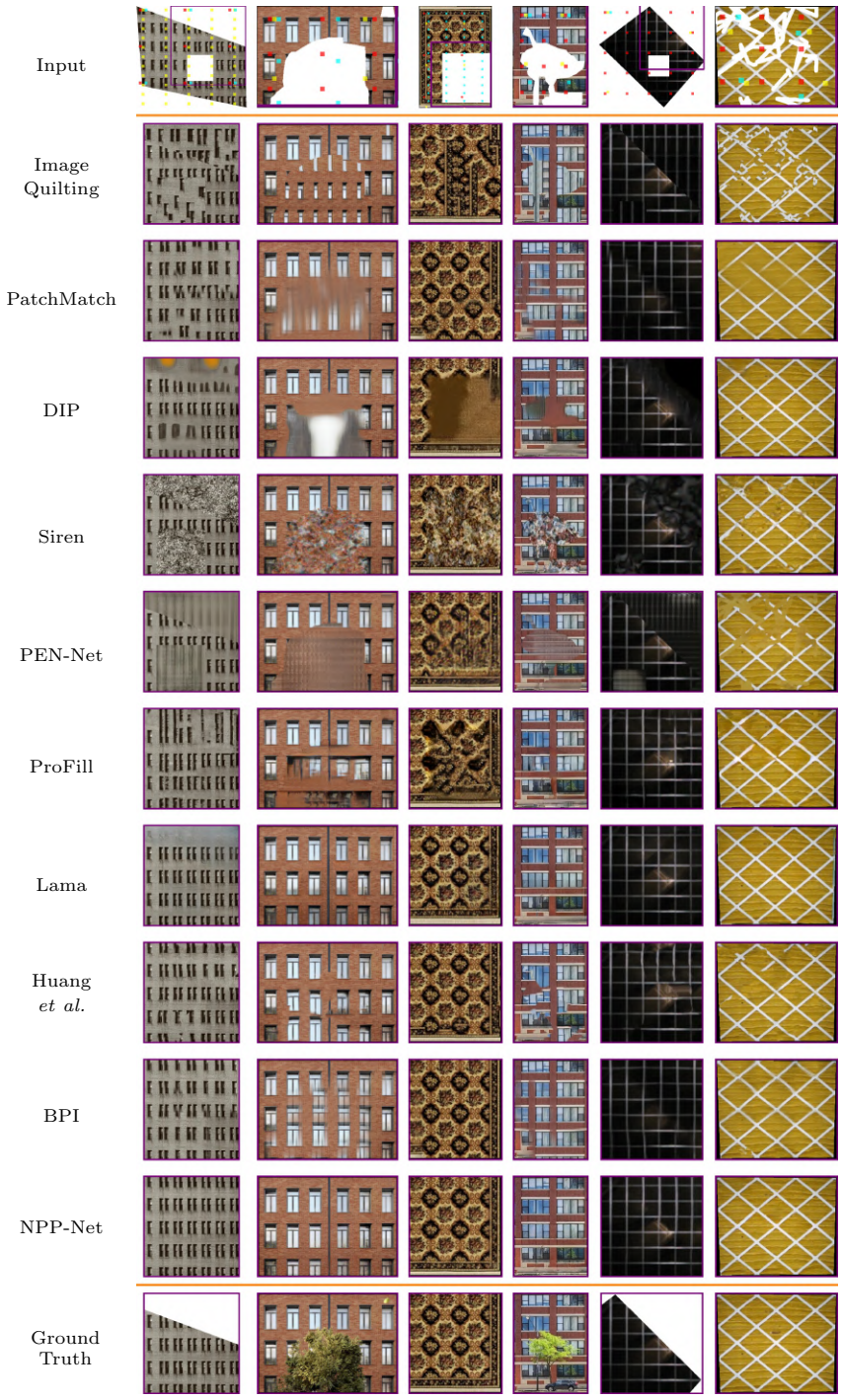


Fig. 17: Comparison with other baselines for image completion. Note that periods in the last column are not scaled by 2.

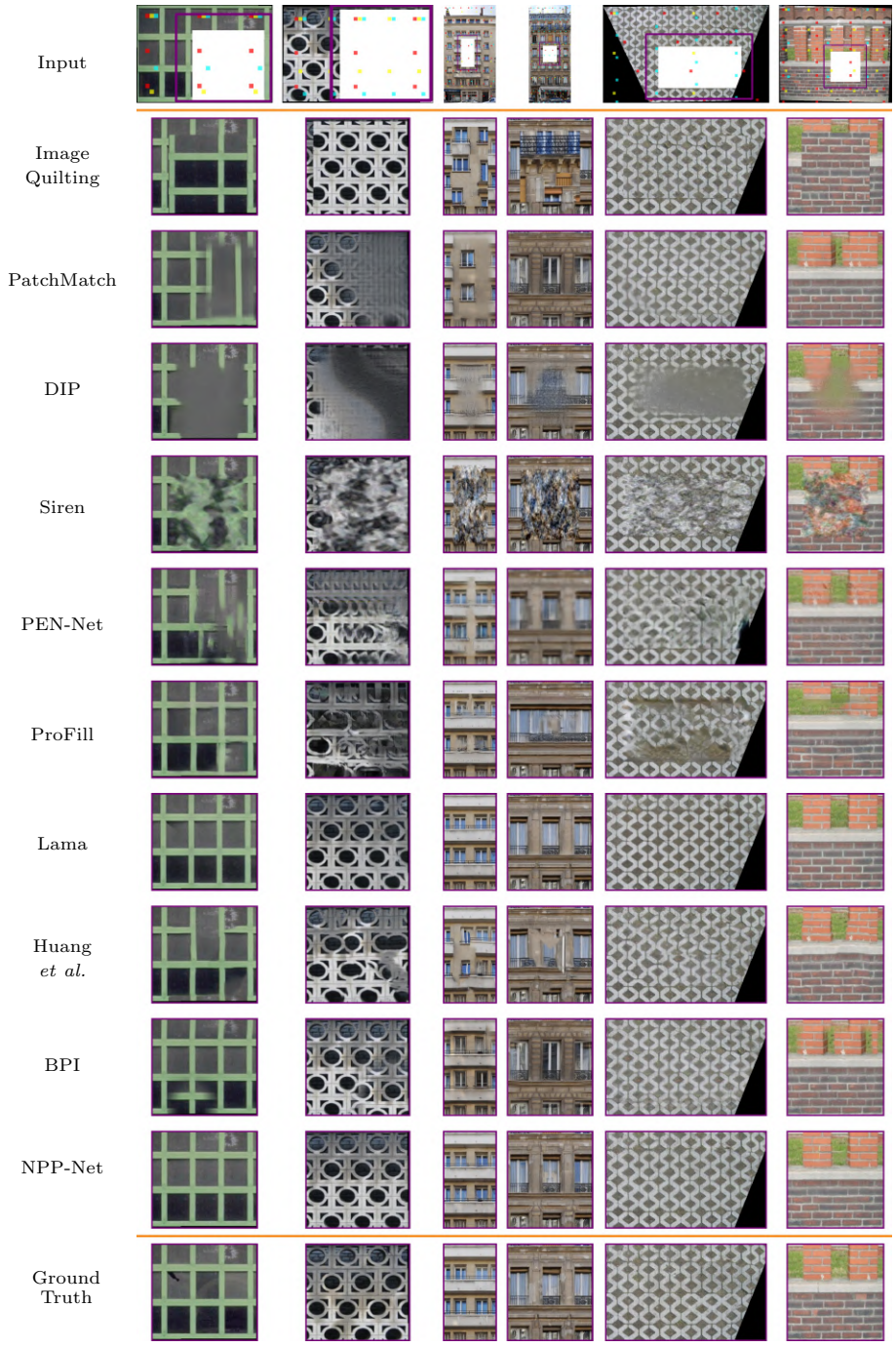


Fig. 18: Comparison with other baselines for image completion.

3.5 Influence of Mask Size

As mentioned in the main paper, we perform two experiments to study the influence of different mask sizes.

First, for each image in NRTDB dataset, if the K-th periodicity has the smallest error among Top-3 periodicities, we assign the image to the K-th periodicity. We show the number of images assigned to each periodicity with different mask sizes in Figure 19. While the Top 1st periodicity is the best one for most of the images with small mask size, this number decreases in the large mask case (64% of the image). This demonstrates that the other periodicities contain better periodicity and leveraging them by our periodicity augmentation strategy can be helpful for learning NPP representation, especially when the mask is large.

To compute the periodicity error, we manually annotate the periodicity with the smallest period as ground truth periodicity for the dataset. For each periodicity, we generate a 2D point cloud, defined as $\{\alpha \boldsymbol{d}_1 + \beta \boldsymbol{d}_2 | \alpha, \beta \in \mathbb{Z}\}$. We also filter points that are out of image range. The periodicity error is calculated using the average L2 distance between every point in proposed point clouds and its nearest neighbor in the ground truth point cloud (one-directional chamfer distance).

In the second experiment, we show all the evaluation metrics of image completion task to analyze the influence of mask size in NRTDB and DTD datasets in Figure 20 and Figure 21, respectively. While LPIPS, SSIM, PSNR, and RMSE are evaluated only in unknown regions, we evaluate FID in the full image since FID is inaccurate if the image resolution is very small (*e.g.*, 4% of original image). NPP-Net outperforms other baselines, especially when the mask size is large. Figure 22 to Figure 23 show the qualitative results, where NPP-Net performs the best among all methods.

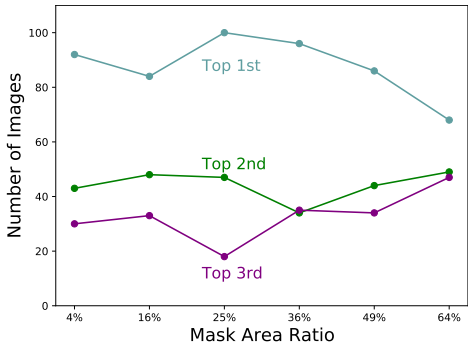
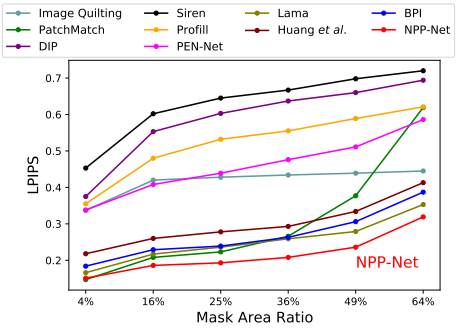
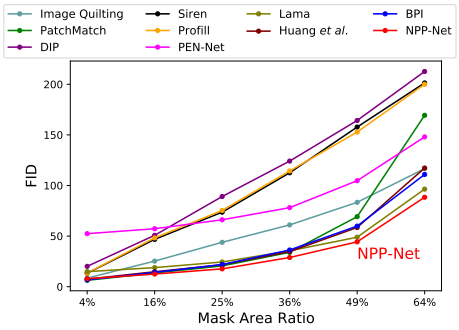


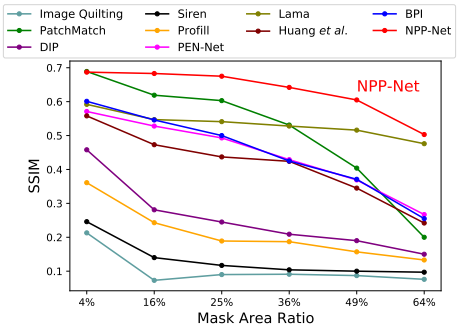
Fig. 19: The number of NPP images for each periodicity that has the smallest periodicity errors (among Top-3) with different mask sizes in NRTDB. As the mask size grows, the best periodicity emerges in the other periodicities, thus utilizing them in NPP-Net is useful.



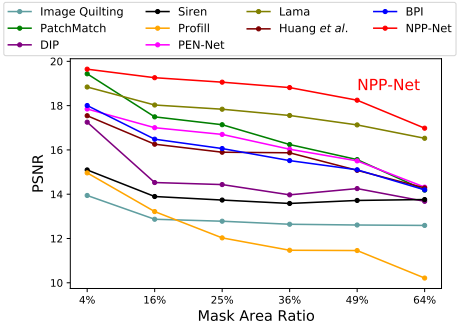
(a) LPIPS



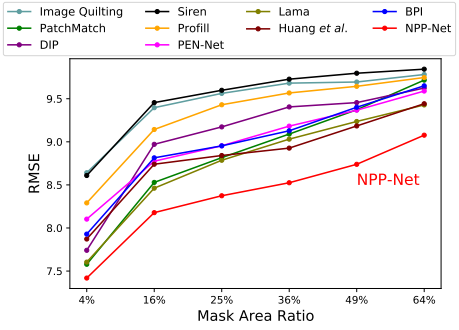
(a) FID



(b) SSIM

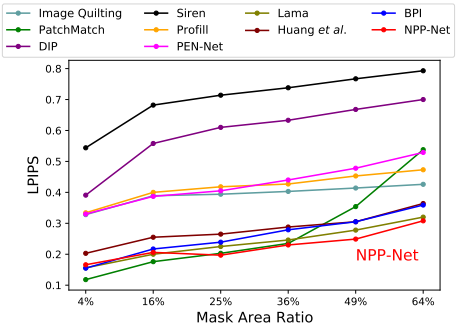


(c) PSNR

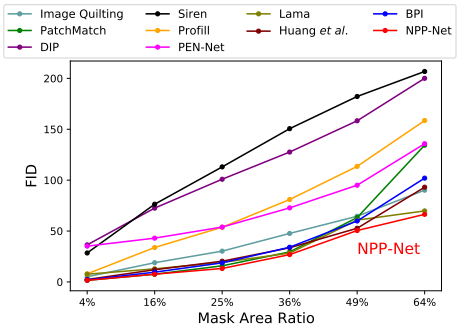


(d) RMSE

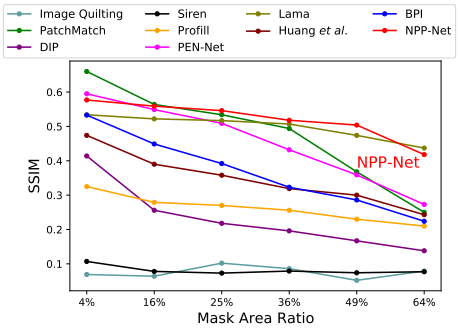
Fig. 20: Comparison of model performances for different mask sizes in the NRTDB dataset. FID is evaluated in the full image, and the other three metrics are tested in the unknown regions.



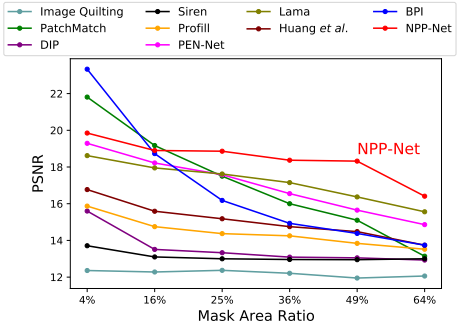
(a) LPIPS



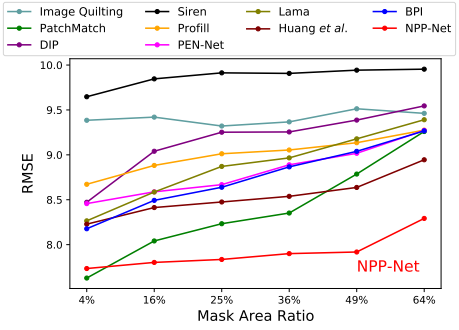
(b) FID



(c) SSIM



(d) PSNR



(e) RMSE

Fig. 21: Comparison of model performances for different mask sizes in the DTD dataset. FID is evaluated in the full image, and the other four metrics are tested in the unknown regions.

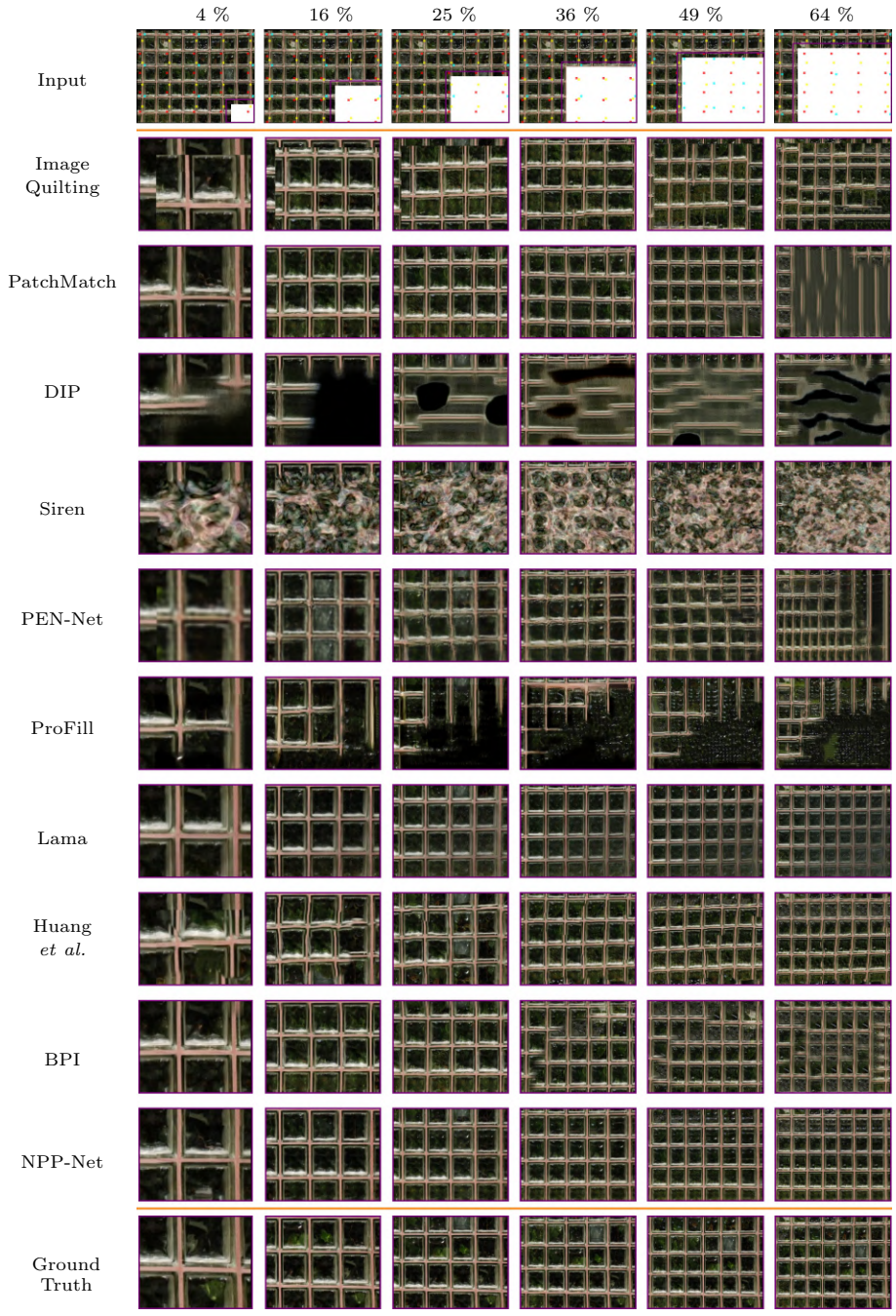


Fig. 22: Qualitative results for different size of masks. From left to right, the mask sizes are 4%, 16%, 25%, 36%, 49% and 64%. NPP-Net outperforms all the baselines.

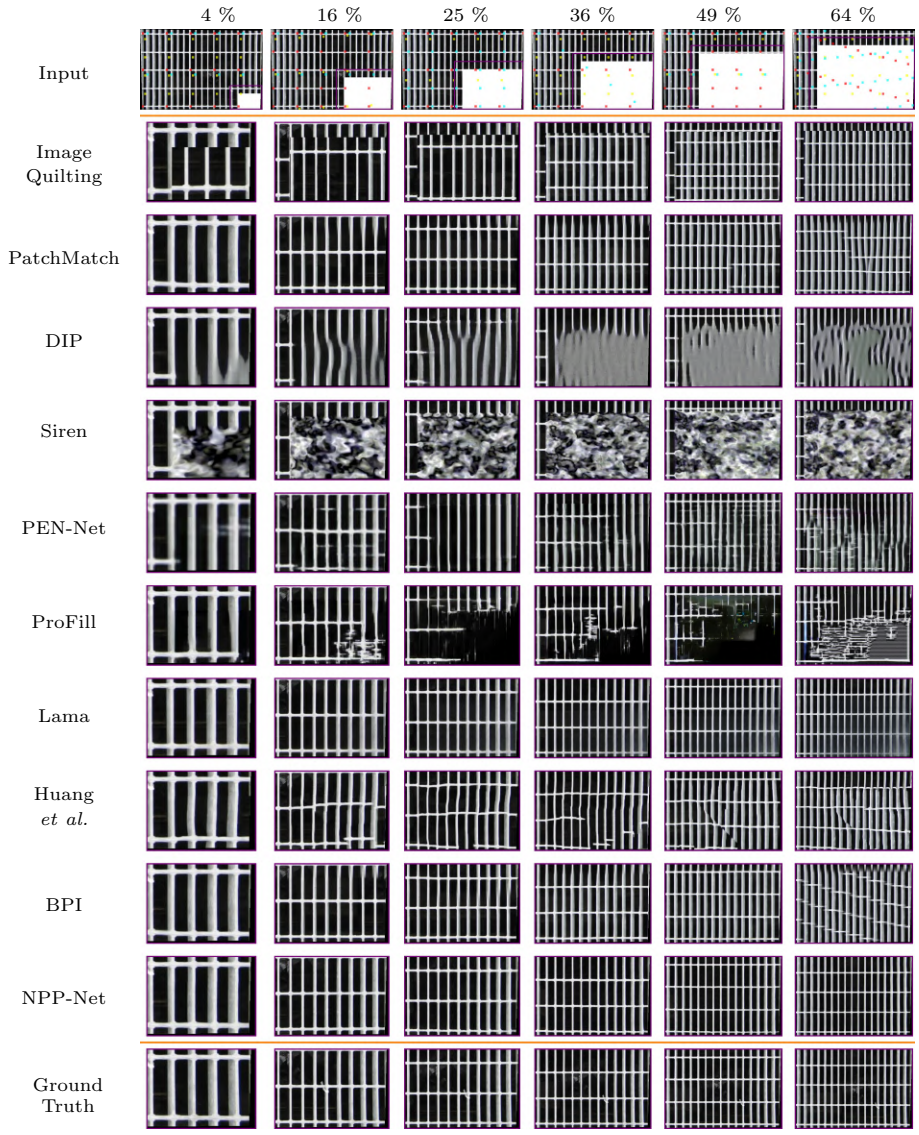


Fig. 23: Qualitative results for different size of masks. From left to right, the mask sizes are 4%, 16%, 25%, 36%, 49% and 64%. NPP-Net outperforms all the baselines.

3.6 Periodicity Refinement

The completed output of NPP-Net can also improve the periodicity detection since more known and reliable information is available for the detection, especially for large masks. The better periodicity can help BPI and NPP-Net achieve better results. Note that, we do not adopt this strategy on Huang *et al.* because it is not easy to incorporate the detection method into their pipeline.

Specifically, given a masked NPP image, we perform our pipeline to complete the image. Then we input the completed image to our pipeline for periodicity proposal, but still, train NPP-Net only in the originally known regions for better completion results. A similar process is also applied to BPI. In practice, we only refine the masked images with the ratio of unknown regions to the sum of unknown and known ones larger than 40%.

Table 7 and Figure 24 show the quantitative and qualitative results for the refinement in NRTDB dataset, respectively. Methods with refinement perform better than those without refinement. Among all the methods, NPP-Net with refinement performs the best, demonstrating the effectiveness of refinement for large masks.

Method	Only Unknown Regions				
	LPIPS ↓	SSIM ↑	PSNR ↑	FID ↓	RMSE ↓
BPI	0.252	0.385	14.83	82.77	9.496
BPI (Refine)	0.244	0.402	14.93	90.16	9.462
NPP-Net	0.186	0.626	17.55	74.10	9.001
NPP-Net (Refine)	**0.171**	**0.636**	**17.87**	**68.20**	**8.904**

Table 7: Comparison of BPI and NPP-Net with and without periodicity refinement. NPP-Net with refinement outperforms all other compared methods.

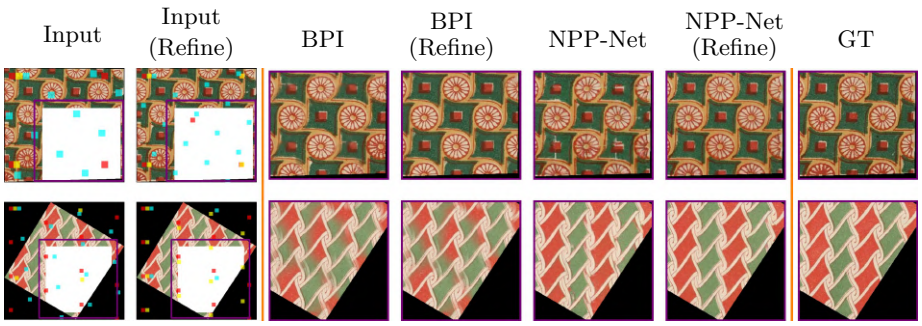


Fig. 24: Qualitative results for periodicity refinement. NPP-Net with refinement outperforms all other compared methods.

3.7 Learnable Periodicity

There are several texture synthesis methods [7,15] that directly treat periodicity as a learnable parameter during training. However, as mentioned in the main paper, this does not work for many real-world NPP images due to the presence of local variations. Predicting periodicity before training stabilizes the network optimization. Figure 25 shows four texture synthesis samples from PSGAN.

(a) Input (b) PSGAN Output

Fig. 25: Texture synthesis results of PSGAN. The left image shows the input and the right image shows four synthesized samples of PSGAN.

4 NPP Segmentation

4.1 Method

We aim to segment the non-periodic regions in an NPP image in an unsupervised manner. The key idea is to detect initial non-periodic regions, treat them as the unknown mask, and relabel regions with low reconstruction error as periodic regions.

Specifically, we first apply a traditional image segmentation method [16] to generate an initial segmentation for non-periodic regions, treated as unknown regions. This segmentation method first divides image pixels as superpixels and uses Gaussian Mixture Model for a coarse segmentation, which is further refined by GraphCut. We treat the class that contains the largest number of pixels as the periodic class, and the rest of the classes as non-periodic classes for initialization. One drawback of this method is that it often over-segments the non-periodic regions. Thus we use NPP-Net to refine the initial segmentation. In detail, we use the same pipeline of image completion to complete the unknown (non-periodic) regions. For every pixel $\boldsymbol{x}$ in non-periodic regions, we can compute the reconstruction error using two metrics since the ground-truth value of $\boldsymbol{x}$ is known. The first metric is the L1 distance, comparing the difference between output and ground truth pixels, given by:

$$S_1(\boldsymbol{x}) = |\hat{G}(\boldsymbol{x}) - G(\boldsymbol{x})|, \tag{7}$$

where $\hat{G}(\boldsymbol{x})$ and $G(\boldsymbol{x})$ are the grayscale value of output and ground truth at $\boldsymbol{x}$, respectively. The second metric is perceptual distance [41] based on pretrained network, given by:

$$S_2(\boldsymbol{x}) = ||P_{\hat{I}}(\boldsymbol{x}) - P_I(\boldsymbol{x})||_2, \tag{8}$$

where $P_{\hat{I}}$ and P_I denote the normalized perceptual activation image (first layer of AlexNet) of the output and ground truth NPP image, respectively. Only $\boldsymbol{x}$ with low reconstruction error is eligible for relabelling, and these $\boldsymbol{x}$ are defined as a set S, given by:

$$S = \{\boldsymbol{x} | S_1(\boldsymbol{x}) < \epsilon_1, S_2(\boldsymbol{x}) < \epsilon_2\}, \tag{9}$$

where ϵ_1 and ϵ_2 are constants. Finally, we relabel $\boldsymbol{x} \in S$ to periodic class to obtain our segmentation. For implementation details, we set $\epsilon_1 = 0.15$ and $\epsilon_2 = 0.3$. Also, we blur the input NPP image before using our pipeline to remove fine details in the image because we only focus on the global structure.

4.2 Comparison with Baselines

We compare with HRNet [27] and Borovec *et al.* [16]. The first one is a network trained on a large dataset (ADE20K [43]), and the latter one is an unsupervised method based on Gaussian Mixture Model and Graphcut.

We show NPP segmentation results in Figure 26, where NPP-Net outperforms all the chosen baselines. HRNet cannot produce reasonable results because it is trained on a dataset of objects and scenes rather than periodic patterns. Borovec *et al.* often over-segments the non-periodic regions since it is based on low-level features without high-level periodicity understanding.

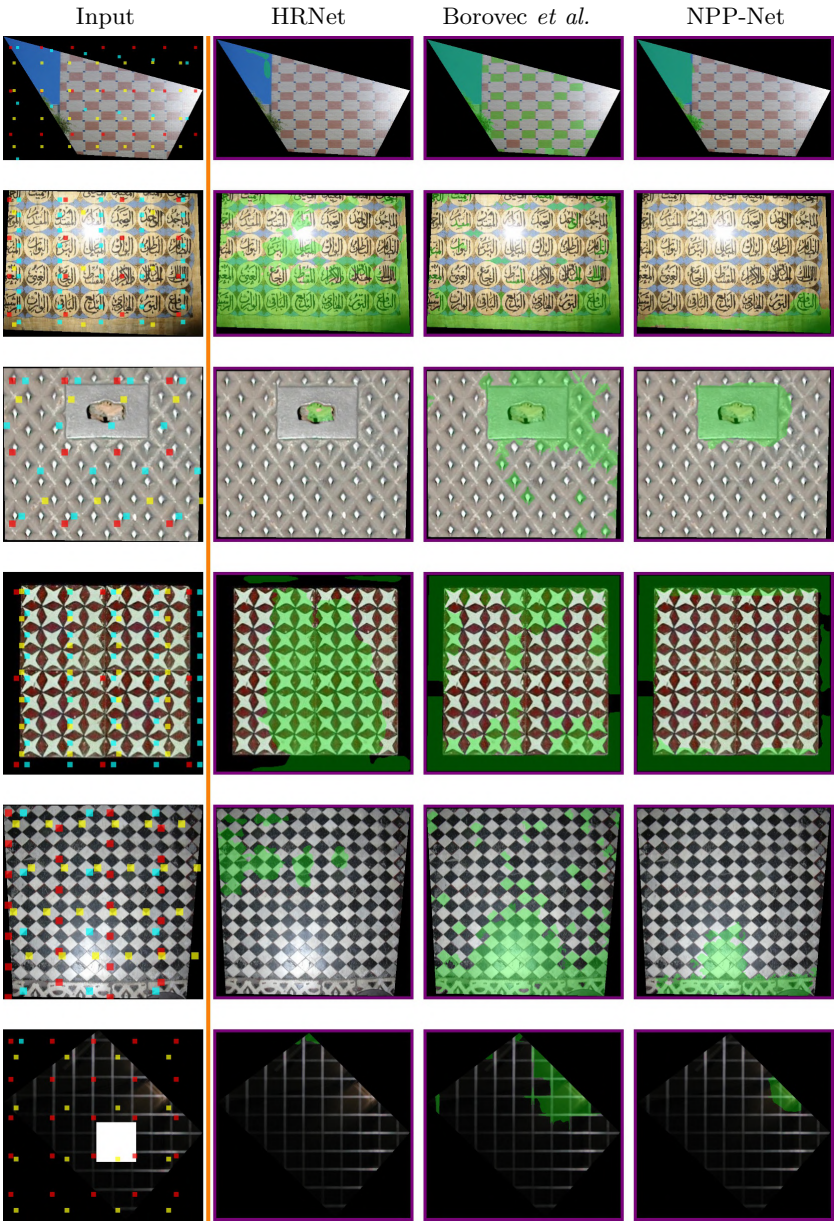


Fig. 26: Qualitative results of non-periodic region segmentation. The green regions refer to non-periodic regions.

4.3 NPP Classification

The NPP segmentation can be extended to classify whether an input image is NPP or not. This can serve as a pre-filtering step before applying NPP-Net for completion. Also, it can be adopted to collect NPP images from a large collection of natural images.

Given an arbitrary image with an unknown mask, we label the regions around the unknown mask as initial non-periodic regions. The remaining known regions are treated as periodic regions for training. Then we apply NPP segmentation to relabel the non-periodic regions. If more than 50% of pixels are relabeled, we classify it as an NPP image.

We test this classification method on the DTD dataset. We select 40 non-NPP images from the class "potholed" in the original dataset, together with 258 images in the chosen DTD dataset for experiments. We generate an unknown mask with height and width equal to 50% of the image height and width (starting from the bottom right) respectively. We initialize the initial non-periodic region as a mask with height and width equal to 70% of the image height and width (but excluding the unknown region).

Quantitatively, the precision of NPP classification for all images (298 images), only non-NPP images (258 images), and only NPP images (40 images) are 82.2%, 81.9%, 84.6% respectively.

Figure 27 shows the qualitative results of the NPP classification. NPP-Net correctly classifies the images even if the strong local variations are presented (row 1 column 4). Besides, we also show the failure cases in Figure 28, where NPP-Net fails for the images with homogenous contents or complicated backgrounds.

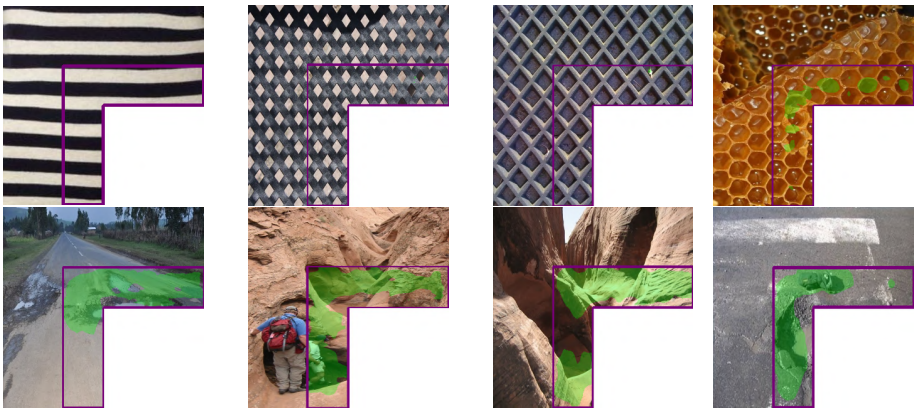


Fig. 27: The qualitative results of NPP classification. The regions inside the purple box are the initial non-periodic regions, and the regions highlighted in green refer to the final periodic regions. NPP-Net correctly classifies images in the first row (NPP images) and second row (non-NPP images). The more regions in green, the more likely the image is a non-NPP image.

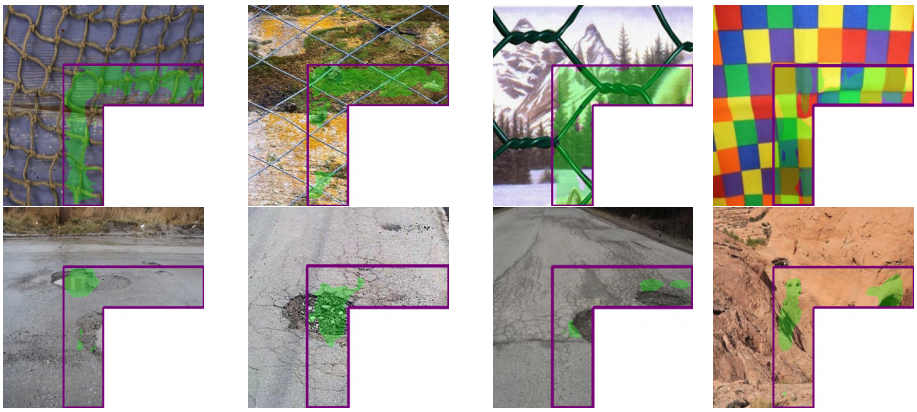


Fig. 28: The failure cases of NPP classification. The regions inside the purple box are the initial non-periodic regions, and the regions highlighted in green refer to the final periodic regions. NPP-Net fails to correctly classify images in the first row (NPP images) and second row (non-NPP images) because they have homogenous contents or complicated backgrounds. The more regions in green, the more likely the image is a non-NPP image.

5 NPP Remapping

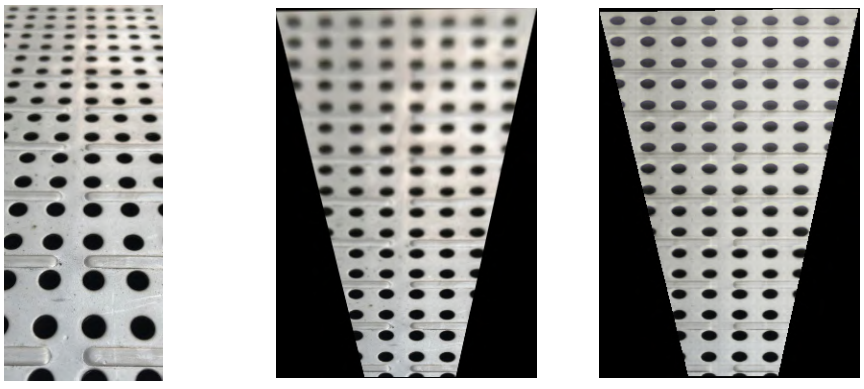
5.1 Method

NPP textures are usually adopted in various applications such as rendering. A perspective camera can be used to obtain real-world NPP texture by rectifying the captured NPP perspective image. However, the potential blurry texture issue prevents us to obtain high-quality texture. Consider an NPP captured by a perspective camera in a tilted angle, as shown in Figure 29 (a). The far-away motifs are blurry because the regions are out of focus. This problem becomes severe after rectification due to the remapping issue, shown in Figure 29 (b). The goal of NPP remapping is to recover the blurry regions (caused by image remapping errors) in the NPP images. It outputs clear NPP images by preserving local variations and global structure, shown in Figure 29 (c).

The key idea is to detect the blurry regions from the input and treat them the same as the unknown regions in the completion task with minor modification. Specifically, we first detect the blurry regions using [26], treated as unknown regions. This prevents the patch loss from sampling ground truth patches in blurry regions for supervision. Simply treating it as a completion problem cannot preserve the local variations (*e.g.*, lighting) in the blurry regions, thus we modify the pixel loss to account for this issue. Instead of computing this loss only in known (clear) regions, we also compute the loss on unknown (blurry) regions, given by:

$$\mathcal{L}_{pixel}(\boldsymbol{x}) = M(\boldsymbol{x})\mathcal{L}_{rob}(\hat{C}(\boldsymbol{x}), C(\boldsymbol{x})) + \sigma(1 - M(\boldsymbol{x}))\mathcal{L}_{rob}(\hat{C}(\boldsymbol{x}), C(\boldsymbol{x})), \quad (10)$$

where σ is a constant weight. $M(\boldsymbol{x})$ is 1 if $\boldsymbol{x}$ in clear regions, and 0 in blurry regions. We keep the remaining part the same as completion to optimize NPP-Net. For implementation details, we set $\sigma = 0.3$.



(a) Perspective NPP (b) Rectified NPP (c) Output

Fig. 29: Illustration of an NPP Remapping scenario.

5.2 Comparison with Baselines

We consider two baselines. MPRNet [37] and SelfDeblur [22] are state-of-the-art deblurring methods trained on large datasets and a single image, respectively.

We show more qualitative results in Figure 30 and Figure 31. Note that in Figure 30, MPRNet produces pixelated pattern in row 3. NPP-Net outperforms all the baselines for the blurry region recovery.

Also, Figure 31 demonstrates another application for image remapping [9]: the indoor panorama may have an NPP floor with far-away blurry floor regions. We can use a pre-trained panorama segmentation method to segment the floor regions, and rectify them into orthographic view, shown as the inputs in Figure 31. After recovering the blurry regions using NPP-Net, we can warp it back to the original panorama, resulting in a panorama with clearer floor regions.

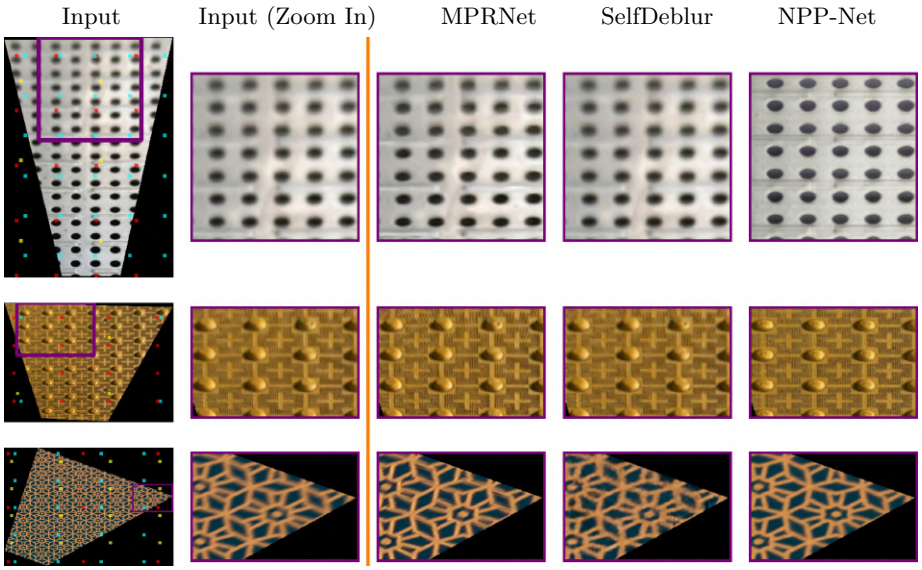


Fig. 30: Qualitative results for image remapping. MPRNet generates pixelated result in row 3. NPP-Net outperforms all other baselines.

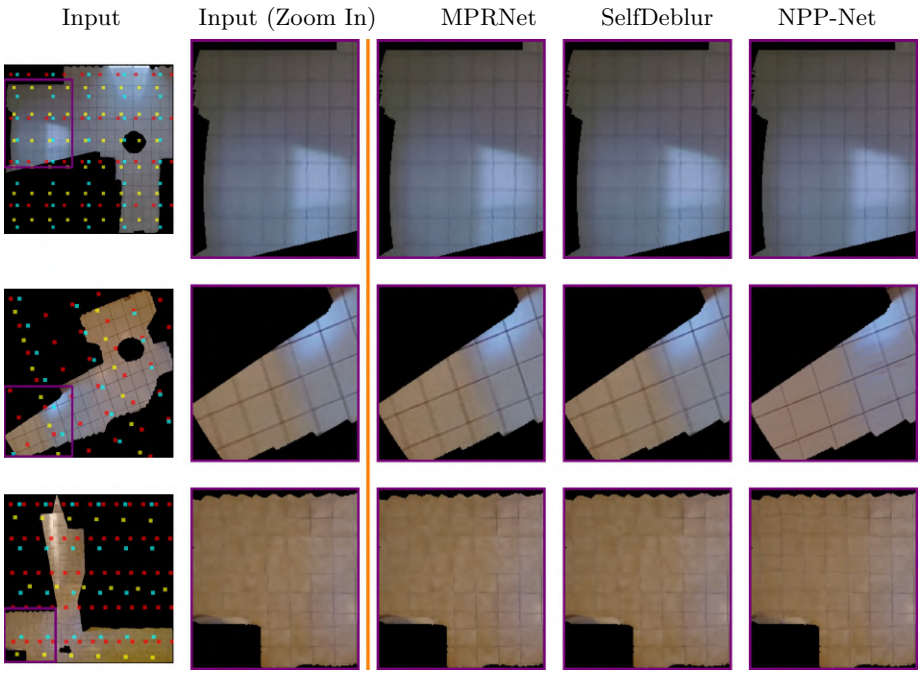


Fig. 31: Qualitative results of remapped (orthographic) floor regions in indoor panorama for image remapping. The small black masks inside known regions is the tripod. NPP-Net outperforms all other baselines.

6 Multi-Plane NPP Completion

NPP-Net can be extended to multi-planar scenes. The key idea is to automatically segment and rectify each plane, apply NPP-Net on each plane for completion, and project the completed image back to the original image.

Given a masked image (Figure 32 (a)) with different NPPs on different planes, we first adopt a pre-trained plane segmentation network [36] to obtain a coarse plane segmentation. Since this network is not trained on images with masks, we first use our "No Periodicity" variant to inpaint the masked (unknown) regions. The output is shown in Figure 32 (b). This process takes about 30 seconds. Then we can input this inpainted image in (b) to the pre-trained network to generate the coarse plane segmentation in Figure 32 (c).

For each segmented plane, we select a bounding box using a similar strategy as the pseudo mask generation in periodicity searching (Sec 2.2). This bounding box is utilized as a reference to rectify the plane using TILT [42]. Thus we do not require accurate segmentation since it is only used for bounding box selection. Figure 32 (c) visualizes bounding boxes for two plane, and Figure 32 (d) shows the rectified planes.

For each rectified plane, we detect the Top-3 periodicity (Figure 32 (d)) only using the rectified bounding box regions to remove the influence of other planes. Then we perform NPP segmentation to segment the non-periodic regions (mainly from other planes) and treat them as invalid pixels. In this case, the initial non-periodic regions are defined as all image regions excluding the bounding box regions. The segmentation results are shown in Figure 32 (e). After that, we run NPP completion on each plane, and the results are in Figure 32 (f).

Finally, we transform all completed planes back to the original image coordinate system and recompose the image. The final inpainted and ground truth images are in Figure 32 (g) and (h) respectively.

The qualitative comparisons with baselines are shown in Figure 33 to Figure 36.

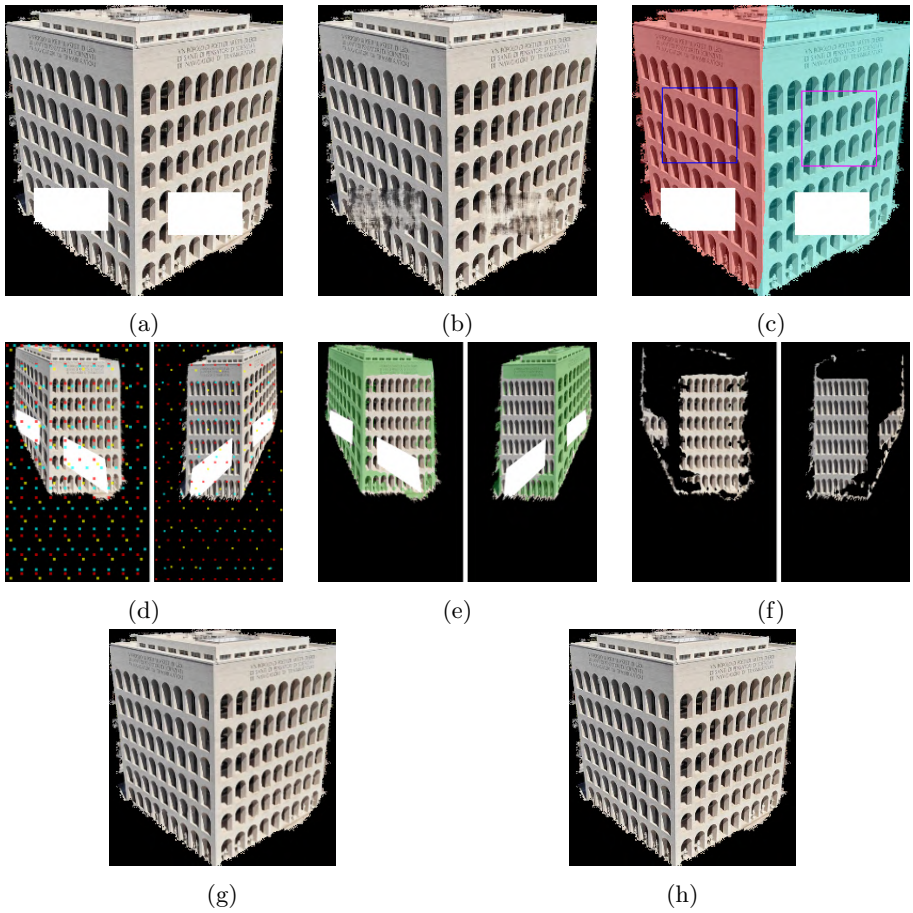
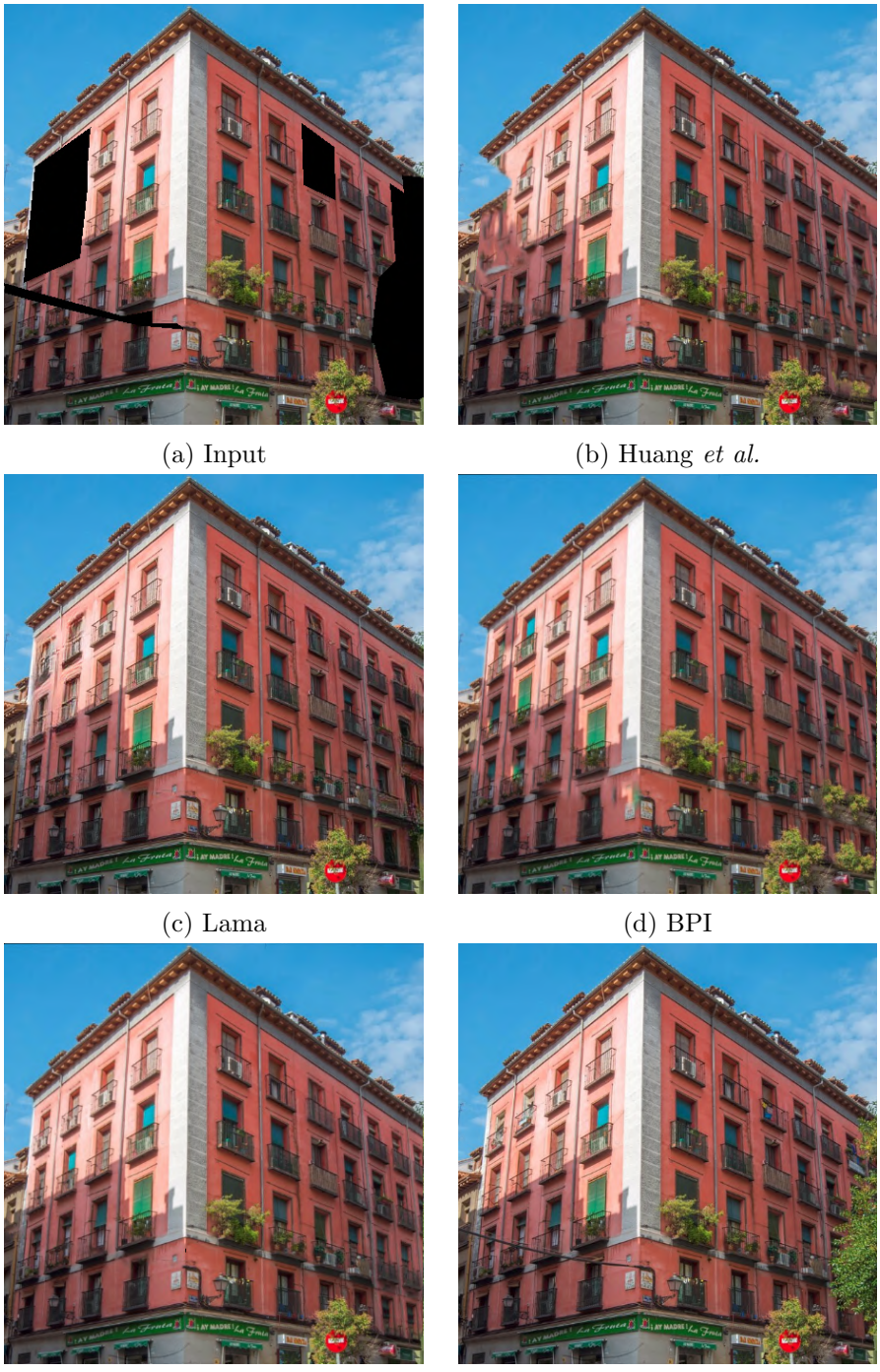


Fig. 32: Illustration of the multi-planar image completion extension for NPP-Net.

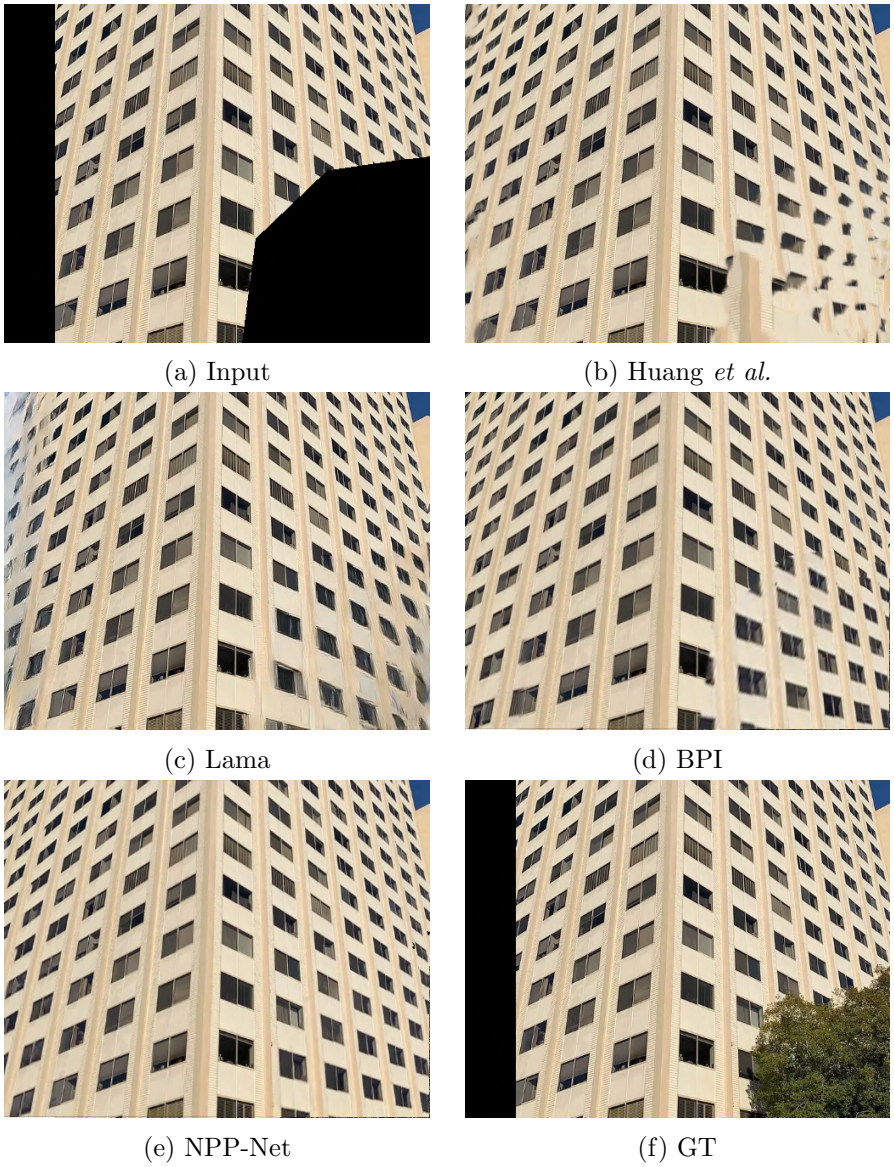


(a) Input (b) Huang *et al.*

(c) Lama (d) BPI

(e) NPP-Net (f) GT

Fig. 33: Qualitative results of a multi-planar NPP scene.



(a) Input (b) Huang *et al.*

(c) Lama (d) BPI

(e) NPP-Net (f) GT

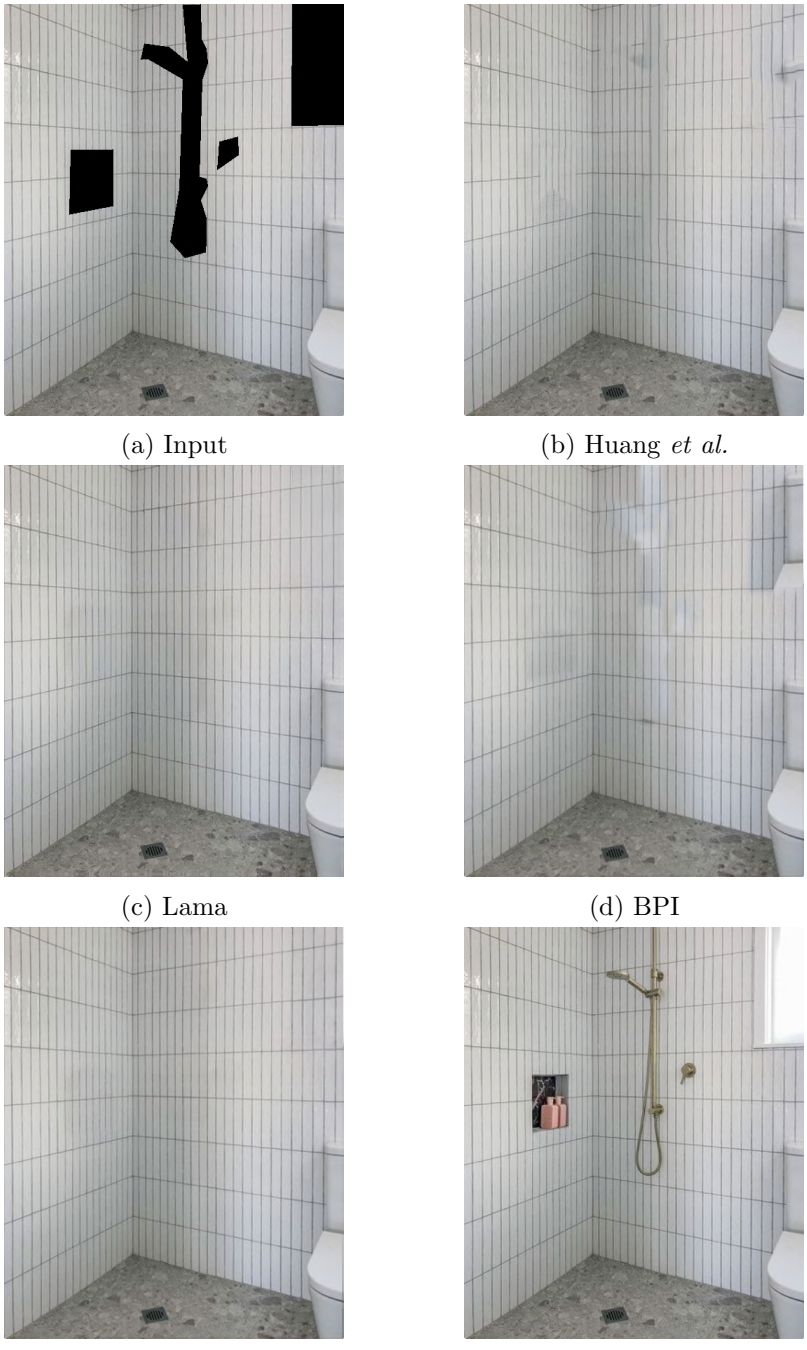
Fig. 34: Qualitative results of a multi-planar NPP scene.

(a) Input (b) Huang *et al.*

(c) Lama (d) BPI

(e) NPP-Net (f) GT

Fig. 35: Qualitative results of a multi-planar NPP scene. All images are resized to half of the original size to reduce file size.



(a) Input (b) Huang *et al.*

(c) Lama (d) BPI

(e) NPP-Net (f) GT

Fig. 36: Qualitative results of a multi-planar NPP scene.

7 Failure Case and Future Work

Our method fails in the case of non-planar NPP scene. Figure 37 shows a failure case for non-planar scene.

For future work, We plan to explore: (1) NPP scenes with more complicated geometry (*e.g.*, non-planar scenes). (2) a few-shot learning strategy for NPP-Net that can incorporate prior knowledge from only a few NPP images.

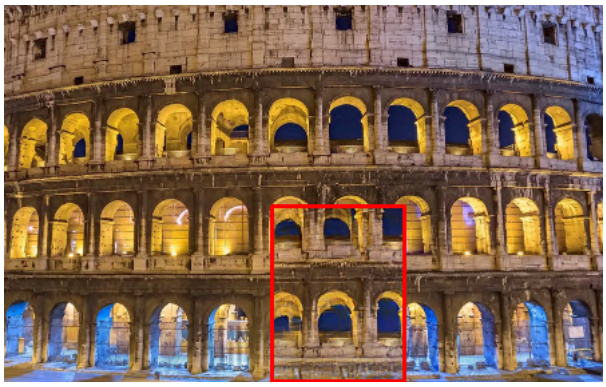


Fig. 37: Failure case of NPP-Net for non-planar scene, where region inside red box is inpainted.

8 Use of Existing Assets

CodeBase: We implement NPP-Net based on the existing assets [35,23,24,17,12,4]. For metrics, we adopt the following implementations: LPIPS [41], FID [21], RMSE [6], SSIM [34], and PSNR [5]. Also, we adopt the following implementations for baselines: Image Qualiting [2], PatchMatch [32], DIP [10], Siren [33], ProFill [38], Huang *et al.* [14,28], and BPI [32].

References

1. PSU Near-Regular Texture Database. http://vivid.cse.psu.edu/
2. axu2: Image quilting for texture synthesis and transfer. https://github.com/axu2/image-quilting (2019)
3. Barnes, C., Shechtman, E., Finkelstein, A., Goldman, D.B.: Patchmatch: A randomized correspondence algorithm for structural image editing. ACM Trans. Graph. p. 24 (2009)
4. Borda: Image segmentation toolbox. https://github.com/Borda/pyImSegm (2020)
5. Bradski, G.: The OpenCV Library. Dr. Dobb's Journal of Software Tools (2000)
6. Buitinck, L., Louppe, G., Blondel, M., Pedregosa, F., Mueller, A., Grisel, O., Niculae, V., Prettenhofer, P., Gramfort, A., Grobler, J., Layton, R., VanderPlas, J., Joly, A., Holt, B., Varoquaux, G.: API design for machine learning software: experiences from the scikit-learn project. In: ECML PKDD Workshop: Languages for Data Mining and Machine Learning. pp. 108–122 (2013)
7. Chen, H., Liu, J., Chen, W., Liu, S., Zhao, Y.: Exemplar-based pattern synthesis with implicit periodic field network. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3708–3717 (2022)
8. Cimpoi, M., Maji, S., Kokkinos, I., Mohamed, S., , Vedaldi, A.: Describing textures in the wild. In: Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR) (2014)
9. Cruz, S., Hutchcroft, W., Li, Y., Khosravan, N., Boyadzhiev, I., Kang, S.B.: Zillow indoor dataset: Annotated floor plans with 360º panoramas and 3d room layouts. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2133–2143 (June 2021)
10. DmitryUlyanov: Deep image prior. https://github.com/DmitryUlyanov/deep-image-prior (2018)
11. Efros, A.A., Freeman, W.T.: Image quilting for texture synthesis and transfer. In: International Conference on Computer Graphics and Interactive Techniques (SIGGRAPH). p. 341–346. Association for Computing Machinery (2001)
12. fled: Blurred region detection using singular value decomposition (svd). https://github.com/fled/blur_detection (2018)
13. Huang, J.B., Kang, S.B., Ahuja, N., Kopf, J.: Image completion using planar structure guidance. ACM Transactions on graphics (TOG) pp. 1–10 (2014)
14. jbhuang0604: Structcompletion. https://github.com/jbhuang0604/StructCompletion (2014)
15. Jetchev, N., Bergmann, U., Vollgraf, R.: Texture synthesis with spatial generative adversarial networks. arXiv preprint arXiv:1611.08207 (2016)
16. Jiri, B., Jan, S., Jan, K., Habart, D.: Supervised and unsupervised segmentation using superpixels, model estimation, and graph cut. Journal of Electronic Imaging (2017)
17. jonbarron: robust_loss_pytorch. https://github.com/jonbarron/robust_loss_pytorch (2020)
18. Li, Y., Mao, J., Zhang, X., Freeman, W.T., Tenenbaum, J.B., Snavely, N., Wu, J.: Multi-plane program induction with 3d box priors. In: Neural Information Processing Systems (NeurIPS) (2020)
19. Li, Y., Mao, J., Zhang, X., Freeman, W.T., Tenenbaum, J.B., Wu, J.: Perspective plane program induction from a single image. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 4434–4443. IEEE (2020)

20. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: Nerf: Representing scenes as neural radiance fields for view synthesis. In: European conference on computer vision (ECCV). pp. 405–421. Springer (2020)
21. Parmar, G., Zhang, R., Zhu, J.Y.: On aliased resizing and surprising subtleties in gan evaluation. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2022)
22. Ren, D., Zhang, K., Wang, Q., Hu, Q., Zuo, W.: Neural blind deconvolution using deep priors. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 3341–3350 (2020)
23. richzhang: Perceptualsimilarity. `https://github.com/richzhang/PerceptualSimilarity` (2020)
24. S-aiueo32: contextual loss pytorch. `https://github.com/S-aiueo32/contextual_loss_pytorch` (2020)
25. Sitzmann, V., Martel, J., Bergman, A., Lindell, D., Wetzstein, G.: Implicit neural representations with periodic activation functions. Advances in Neural Information Processing Systems (2020)
26. Su, B., Lu, S., Tan, C.L.: Blurred image region detection and classification. In: Proceedings of the 19th ACM International Conference on Multimedia (MM) (2011)
27. Sun, K., Xiao, B., Liu, D., Wang, J.: Deep high-resolution representation learning for human pose estimation. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2019)
28. SunskyF: Structcompletion. `https://github.com/SunskyF/StructCompletion-python` (2016)
29. Suvorov, R., Logacheva, E., Mashikhin, A., Remizova, A., Ashukha, A., Silvestrov, A., Kong, N., Goka, H., Park, K., Lempitsky, V.: Resolution-robust large mask inpainting with fourier convolutions. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 2149–2159 (2022)
30. Teboul, O., Simon, L., Koutsourakis, P., Paragios, N.: Segmentation of building facades using procedural shape priors. In: 2010 IEEE computer society conference on computer vision and pattern recognition. pp. 3105–3112. IEEE (2010)
31. Ulyanov, D., Vedaldi, A., Lempitsky, V.: Deep image prior. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 9446–9454. IEEE (2018)
32. vacancy: Patchmatch based inpainting. `https://github.com/vacancy/PyPatchMatch` (2019)
33. vsitzmann: Implicit neural representations with periodic activation functions. `https://github.com/vsitzmann/siren` (2020)
34. Van der Walt, S., Schönberger, J.L., Nunez-Iglesias, J., Boulogne, F., Warner, J.D., Yager, N., Gouillart, E., Yu, T.: scikit-image: image processing in python. PeerJ **2**, e453 (2014)
35. yenchenlin: Nerf-pytorch. `https://github.com/yenchenlin/nerf-pytorch` (2020)
36. Yu, Z., Zheng, J., Lian, D., Zhou, Z., Gao, S.: Single-image piece-wise planar 3d reconstruction via associative embedding. In: CVPR. pp. 1029–1037 (2019)
37. Zamir, S.W., Arora, A., Khan, S., Hayat, M., Khan, F.S., Yang, M.H., Shao, L.: Multi-stage progressive image restoration. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2021)
38. Zeng: Profill: High-resolution image inpainting with iterative confidence feedback and guided upsampling, eccv 2020. `https://zengxianyu.github.io/iic/` (2020)
39. Zeng, Y., Fu, J., Chao, H., Guo, B.: Learning pyramid-context encoder network for high-quality image inpainting. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1486–1494 (2019)

40. Zeng, Y., Lin, Z., Yang, J., Zhang, J., Shechtman, E., Lu, H.: High-resolution image inpainting with iterative confidence feedback and guided upsampling. In: European Conference on Computer Vision (ECCV). pp. 1–17. Springer (2020)
41. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2018)
42. Zhang, Z., Ganesh, A., Liang, X., Ma, Y.: Tilt: Transform invariant low-rank textures. International journal of computer vision (IJCV) pp. 1–24 (2012)
43. Zhou, B., Zhao, H., Puig, X., Fidler, S., Barriuso, A., Torralba, A.: Scene parsing through ade20k dataset. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 633–641 (2017)
44. Ziyin, L., Hartwig, T., Ueda, M.: Neural networks fail to learn periodic functions and how to fix it. In: Neural Information Processing Systems (NeurIPS). pp. 1583–1594 (2020)