DeepCFL: Deep Contextual Features Learning from a Single Image (Supplementary Material)

Indra Deep Mastan and Shanmuganathan Raman
Indian Institute of Technology Gandhinagar
Gandhinagar, Gujarat, India
{indra.mastan, shanmuga}@iitgn.ac.in

	DIP [8]	**MEDS** [2]	**DIP** [8] **+ CL**	**InGAN** [7]	**DCIL** [3]	**DeepCFL (ours)**
Framework Design	Encoder-decoder framework.	Multi-level extension of encoder-decoder framework.	Encoder-decoder framework with the contextual loss (CL).	Single image GAN framework for internal patch distribution learning.	Single image GAN framework for internal patch distribution learning with contextual loss.	Single image GAN framework for learning the distribution of contextual features.
Target Application.	Image restoration.	Image restoration.	Image restoration.	Image synthesis.	Image restoration and image synthesis.	Image restoration and image synthesis.
Non-aligned image data applications.	No. Pixel to pixel comparison.	No. Pixel to pixel comparison.	Yes. The contextual loss compares image statistics at VGG features space.	Yes. Cycle consistency loss.	Yes. Cycle consistency loss.	Yes. Cycle consistency loss.
New object synthesis.	No. It lacks new object synthesis abilities of GAN framework.	No. It lacks new object synthesis abilities of GAN framework.	No. It lacks new object synthesis abilities of GAN framework.	Yes. But it may not preserve the context of the object as described in [7].	Yes. It preserves better object context as shown in [3].	Yes. It preserves object context well, as shown in Fig. 6 and Fig. 8 of the manuscript.

Table 1: In this table, we give a summary of single image deep features learning methods. DIP [8] and MEDS [2] perform well for image restoration, but they lack the image synthesis abilities achievable using the GAN framework. DIP [8] and MEDS [2] frameworks compute pixel-based loss. Therefore, they might not apply to non-aligned image data applications. DIP with contextual loss (DIP + CL) computes the MSE loss and contextual loss. The cycle consistency loss would provide the scenario of non-alignment, where the pixel correspondence between the source and the target images is not well defined, for example, content-aware image resizing. Fig. 6 and Fig. 8 of the manuscript show new objects synthesis using single image GAN models: InGAN [7], DCIL [3], and DeepCFL. DCIL [3] perform image restoration by denoising-super resolution. DeepCFL performs image restoration when the input image has missing pixels corrupted by the binary mask (*e.g.*, outpainting).

In this supplementary material, we discuss the DeepCFL framework. We also give detailed visual comparisons and quantitative comparisons for the generated images.

1. DeepCFL Framework

We have illustrated the DeepCFL framework (ours) in Fig. 2 of the main paper. In this supplementary material, we discuss the generator $\mathcal{G}$ and discriminator $\mathcal{D}$ architectures. The generator is an encoder-decoder network as shown in Fig. 1. Encoder-decoder has been reported to capture good quality low-level image statistics from the input image [2, 8]. The multi-scale discriminator shown in Fig. 2 consists of three discriminators which do not share the weights. The input to $\mathcal{D}$ is an image that could be the real image or the output of the generator. We feed the input image to VGG19 ϕ to extract the context vectors. We use

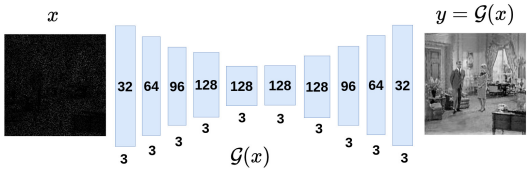


Figure 1: Generator Network. The figure shows the generator network $\mathcal{G}(\cdot)$ of DeepCFL. It is an encoder-decoder network which is used to perform image restoration and image synthesis. The encoder network encodes the image features to smaller feature representations, and the decoder performs the reconstruction from the encoded representation.

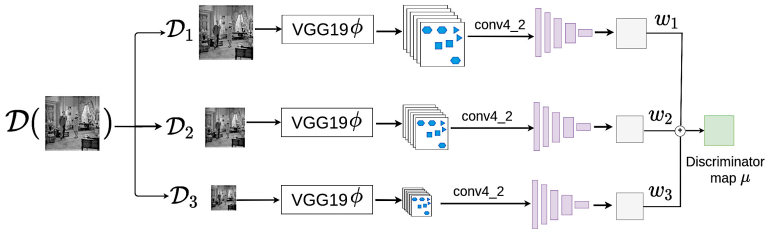


Figure 2: Discriminator architecture. In this figure, we have shown the multi-scale discriminator $\mathcal{D}$ of DeepCFL. It consists three discriminators $\{\mathcal{D}_1, \mathcal{D}_2, \mathcal{D}_3\}$. Each discriminator in $\{\mathcal{D}_i\}_{i=1}^{3}$ is operating at the different scales of the context vectors. Here, $\{w_1, w_2, w_3\}$ are the coefficients of the discriminator. The final output is a discriminator map μ, where each entry in μ denotes the probability of the context vector coming from the distribution of the contextual feature of the original image.

Figure 3: **Image resizing.** The figure shows new objects creations when performing image resize using DeepCFL. The input images are in red color frame.

the features present at `conv_4_2` layer of VGG network ϕ.

In Table 1, we summarize different frameworks for deep features learning from a single image. DIP [8] and MEDS [2] are based on the pixel-to-pixel loss. Mechrez *et al.* have shown that pixel-to-pixel loss is not applicable for non-aligned image data applications [5]. Internal patch distribution learning performs a patch-based comparison. Therefore is applicable for non-aligned image data application, e.g., image retargeting. Our DeepCFL makes the contextual features comparison. Therefore, it also applies to a non-aligned image data application. We have shown content-aware image resizing where the input source image and the target out image does not have a well-defined pixel correspondence).

2. Perceptual Loss and Contextual Loss

We now highlight the interesting difference between perceptual loss (PL) and context vector loss (CVL). Suppose two images denoted by x and y. Also suppose feature extractor VGG19 denoted as ϕ. Let $\phi(x) = \{\phi_l(x)\}_{l=1}^{k}$ denote the features extracted from the k layers of ϕ. Similarly, $\phi(y) = \{\phi_l(y)\}_{l=1}^{k}$ denotes the features of y at

the VGG19 feature space. For simplicity, consider the layer l of $\phi(x)$ and $\phi(y)$ contains N feature vectors, *i.e.*, $\phi_l(x) = \{\phi_l(x)_i\}_{i=1}^N$ and $\phi_l(y) = \{\phi_l(y)_i\}_{i=1}^N$.

The perceptual loss (PL) for the layer l is defined in Eq. 1.

$$PL(x, y, l) = \|\phi_l(x) - \phi_l(y)\|. \tag{1}$$

Here, we can observe that the perceptual loss would be computing the direct difference between feature vectors $\phi_l(x)_i$ and $\phi_l(y)_i$.

The context vector loss (CVL) is computed between the context vectors extracted from the features extractor VGG19 ϕ. The context vectors represent content information present at the higher layers (e.g., conv4_2) of ϕ. CVL is defined in Eq. 2.

$$CVL(x, y, l) = -\log(CX(\phi_l(x), \phi_l(y))). \tag{2}$$

Here, CX is computed by finding for each feature $\phi_l(y)_j$, a feature $\phi_l(x)_i$ that is most contextually similar to it and then summed for all $\phi_l(y)_j$ (Eq. 5 of the manuscript). CX is given in Eq. 3.

$$CX(\phi_l(x), \phi_l(y)) = \frac{1}{N} \sum_j \underset{i}{max}\, CX_{ij} \tag{3}$$

Here, CX_{ij} is the similarity between the context vectors $\phi^l(x)_i$ and $\phi^l(y)_j$. The contextual similarity CX_{ij} is computed by using the cosine distance between feature vectors $\phi_l(x)_i$ and $\phi_l(y)_j$ [5].

It is interesting to note that the perceptual loss does not use the contextual similarity criterion. Therefore, PL will be comparing the different pairs of image features as compared to the pairs of features in CVL.

3. InGAN Implementation.

We have implemented the internal learning of InGAN [7] for the restoration of missing pixels. We have not used the cycle consistency due to the following reason. Cycle consistency would first take input as the corrupted image and generate the restored image, and the next task is to use the restored image and generate a corrupted image (cycle consistency). Generating the corrupted image from the restored image would make the network to learn the corrupted features. Therefore, the cycle consistency would not be useful for image restoration [3]. We have used the encoder-decoder network without skip connections as the generator. We have not used the residual blocks in the generator. It is done to incorporate the generator architectures from the state-of-the-art image restoration methods in the implementation [2, 8].

4. Results.

Here, we provide additional figures and the tables for various applications described in the manuscript. We emphasize that single image deep features learning is sensitive to hyper-parameters search [2,8]. Therefore, we believe that the results of our method and the methods in the comparison could be further improvised using the hyper-parameter search.

- **Ablation Study.** Fig. 4 shows the ablation study that increasing the number of layers in the features extractor VGG19 improvised the restoration of image pixels. Moreover, using features from more layers will be computationally heavy. One trick is to sample a collection of features randomly [5]. The scope of DeepCFL is limited to the contextual features present in conv4_2 layer. We propose as future work to study the single image GAN framework with randomly sampled the context vectors for the restoration of missing pixels.
- **Image Resizing.** We have provided more results for the synthesis of new objects when performing image resize in Fig. 3. The content-aware image resizing is performed by considering the input height and width to be a multiple of 128. It could be observed that the single image GAN framework synthesize new objects features when performing the image resizing. We first performed the image resizing and then do the post-processing of the synthesized output using the style transfer similar to DCIL [3]. The resized image is taken to be the content image, and the input image is taken to be the style image. Post-processing reduces the number of iterations and makes the output image's style features consistent with the input image.
- **Outpainting.** We have provided detailed quantitative comparison for image outpainting in Table 2, Table 3, Table 4, Table 5, and Table 6. We observed that the image features could be fine-tuned by matching the patches from the corrupted input. Therefore, for outpainting, we perform the post-processing using deep image analogy [1].

- **Inpainting.** We have provided a detailed quantitative comparison in Table 7 and Table 8. The visual comparison is provided in Fig. 11.
- **Restore WC 50%.** We have shown the restoration of missing pixels in the presence of the word cloud image with the increasing number of iterations in Fig. 5. The visual quality comparison for the generated images is provided in Fig. 12 and Fig. 13. We provide a detailed quantitative comparison of the generated images in Table 9 and 10.
- **Restore 90%.** We have shown the restoration of missing pixels with the increasing number of iterations in Fig. 6. The input image contains only 10% pixels and 90% pix-

(a) Original image (b) Corrupted image (c) DeepCFL "Conv4_2 layer" (31.14, 0.96) (d) DeepCFL "Conv3_4 & Conv4_2 layers" (31.18, 0.97)

Figure 4: **Ablation Study.** The figure shows that increasing the number of layers for feature comparison using features extractor VGG19 improves performance. It could be observed that the output (d) having Conv3_4 and Conv4_2 layers achieved higher PSNR and SSIM values than the output (c) having only Conv4_2 layer.

Figure 5: The figure shows the restoration of wordcloud in the corrupted image with the increasing number of iterations for Fig. 7 of the manuscript.

	House	Peppers	Lena	Baboon	F16	Kodak-1	Kodak-2	Kodak-3	Kodak-12	**Avg**
DIP [8]	0.94	0.89	0.92	0.89	0.94	0.89	0.96	0.88	0.94	**0.91**
DIP + CL	0.86	0.94	0.93	0.91	0.91	0.87	0.94	0.93	0.95	**0.91**
MEDS [2]	0.89	0.92	0.91	0.92	0.88	0.88	0.94	0.91	0.93	**0.91**
InGAN [6]	0.94	0.88	0.91	0.88	0.93	0.89	0.95	0.94	0.95	**0.92**
DeepCFL	0.95	0.90	0.93	0.88	0.95	0.88	0.94	0.91	0.94	**0.92**

Table 2: **Image Outpainting.** A detailed quantitative comparison (SSIM values) for outpainting of 20% missing pixels.

els are corrupted. We provide more perceptual quality comparison in Fig. 14 and Fig. 15. We provide a detailed quantitative comparison for Restore 90% in Table 11.

- Table 12 shows the size of the image used in our experiments.

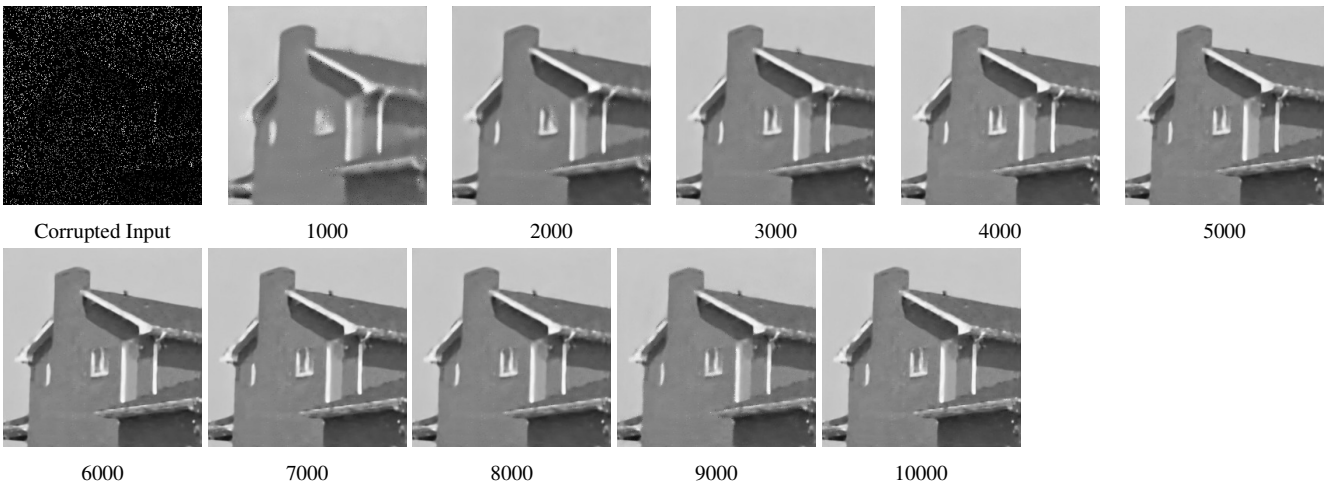


Figure 6: The figure shows the restoration of missing pixels in the corrupted image with the increasing number of iterations (Fig. 13 of the manuscript).

	House	Peppers	Lena	Baboon	F16	Kodak-1	Kodak-2	Kodak-3	Kodak-12	**Avg**
DIP [8]	24.51	20.16	21.72	24.13	22.43	23.69	30.03	22.02	24.88	**23.73**
DIP + CL	24.17	22.64	27.67	21.75	21.35	22.47	25.15	22.74	28.87	**24.09**
MEDS [2]	23.78	21.17	19.71	21.19	18.86	23.32	25.48	20.24	21.55	**21.70**
InGAN [6]	24.52	18.15	20.04	18.29	23.28	23.76	27.36	24.87	25.76	**22.89**
DeepCFL	24.84	20.61	22.35	24.09	25.19	23.46	29.79	22.52	24.36	**24.13**

Table 3: **Image Outpainting.** A detailed quantitative comparison (PSNR values) for outpainting of 20% missing pixels.

	01	*02*	*03*	*04*	*05*	***Avg***
DIP [8]	0.89	0.90	0.88	0.86	0.87	**0.88**
MEDS [2]	0.90	0.87	0.86	0.90	0.89	**0.88**
InGAN [7]	0.92	0.88	0.86	0.89	0.88	**0.89**
DeepCFL	0.91	0.88	0.87	0.92	0.92	**0.90**

	01	*02*	*03*	*04*	*05*	***Avg***
DIP [8]	13.20	22.11	18.66	22.23	18.96	**19.03**
MEDS [2]	16.4	20.35	17.54	22.85	19.65	**19.35**
InGAN [7]	20.22	21.13	18.04	20.01	17.05	**19.29**
DeepCFL	18.09	21.81	19.43	26.83	21.35	**21.50**

Table 4: Detailed PSNR comparison for outpainting on images from Set5 dataset.

	01	*02*	*03*	*04*	*05*	*06*	*07*	*08*	*09*	*10*	*11*	*12*	*13*	*14*	***Avg***
DIP [8]	0.85	0.87	0.88	0.90	0.89	0.88	0.90	0.89	0.91	0.87	0.93	0.88	0.94	0.9	**0.89**
MEDS [2]	0.87	0.88	0.87	0.90	0.86	0.92	0.90	0.90	0.91	0.86	0.89	0.89	0.87	0.87	**0.89**
InGAN [7]	0.86	0.86	0.87	0.91	0.86	0.90	0.89	0.92	0.90	0.86	0.89	0.88	0.93	0.89	**0.89**
DeepCFL	0.86	0.87	0.87	0.90	0.87	0.92	0.91	0.92	0.92	0.88	0.92	0.89	0.94	0.90	**0.90**

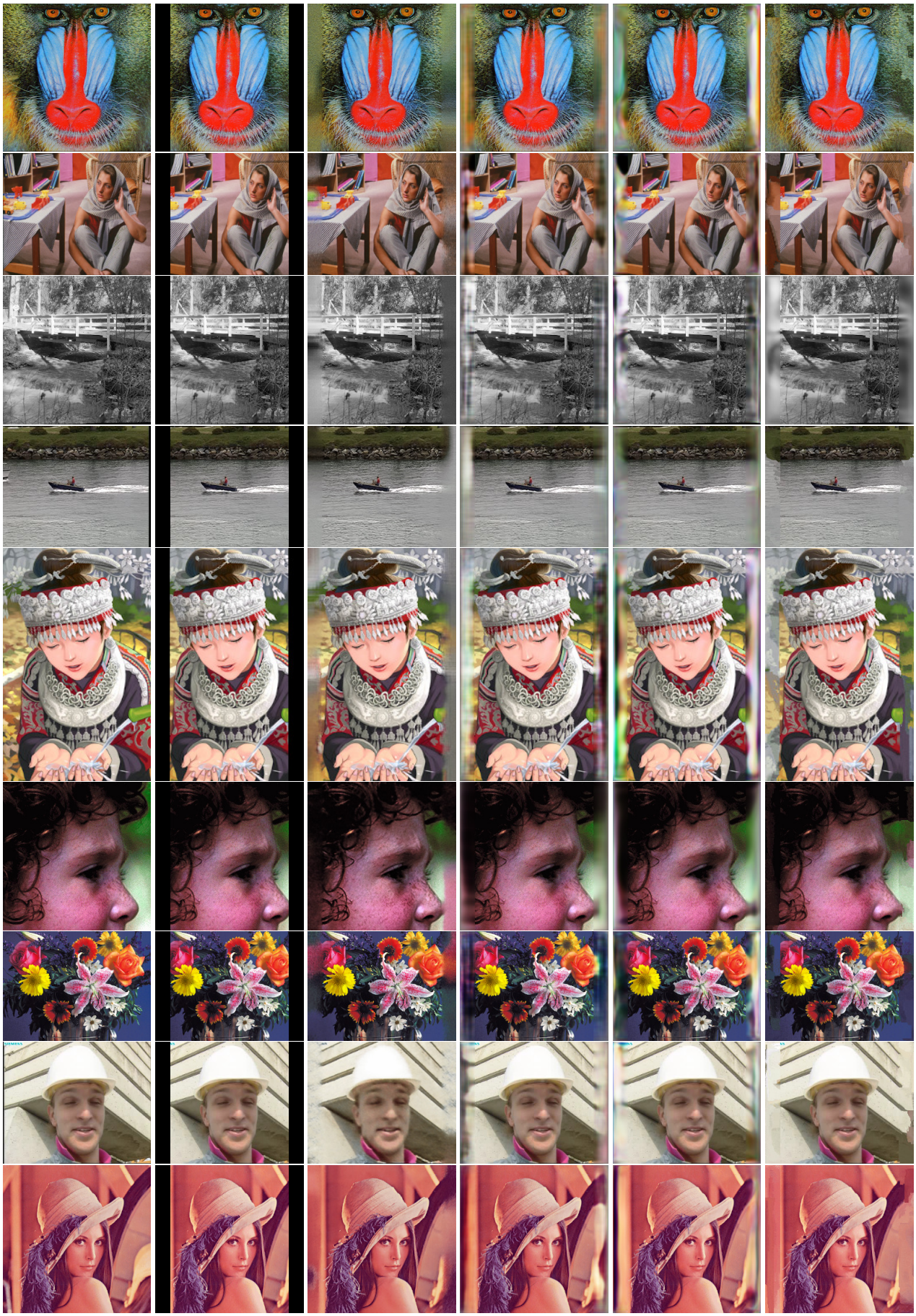
Table 5: Detailed SSIM comparison for outpainting on images from Set14 dataset.

	01	*02*	*03*	*04*	*05*	*06*	*07*	*08*	*09*	*10*	*11*	*12*	*13*	*14*	***Avg***
DIP [8]	23.15	22.13	23.23	21.61	21.61	23.92	23.16	21.12	22.99	19.54	21.88	20.19	21.75	23.41	**22.12**
MEDS [2]	22.35	20.94	21.87	19.42	18.54	23.07	23.53	19.5	20.85	19.88	21.92	19.84	11.8	19.9	**20.24**
InGAN [7]	20.51	20.08	21.47	22.89	20.72	20.74	24.24	22.20	20.31	18.77	22.03	18.41	21.63	22.66	**21.19**
DeepCFL	22.46	20.71	22.13	20.29	20.43	24.99	24.96	21.58	23.42	21.05	24.33	20.55	22.03	26.48	**22.52**

Table 6: Detailed PSNR comparison for outpainting on images from Set14 dataset.

(a) Original **(b)** Masked **(c)** DIP [8] **(d)** DIP + CL **(e)** MEDS [2] **(f)** InGAN [7] **(g)** DeepCFL

Figure 7: Outpainting results on standard data set.



(a) Original **(b)** Masked **(c)** DIP [8] **(d)** MEDS [2] **(e)** InGAN [7] **(f)** DeepCFL

Figure 8: Outpainting results on set 14 data set - I.

(a) Original **(b)** Masked **(c)** DIP [8] **(d)** MEDS [2] **(e)** InGAN [7] **(f)** DeepCFL

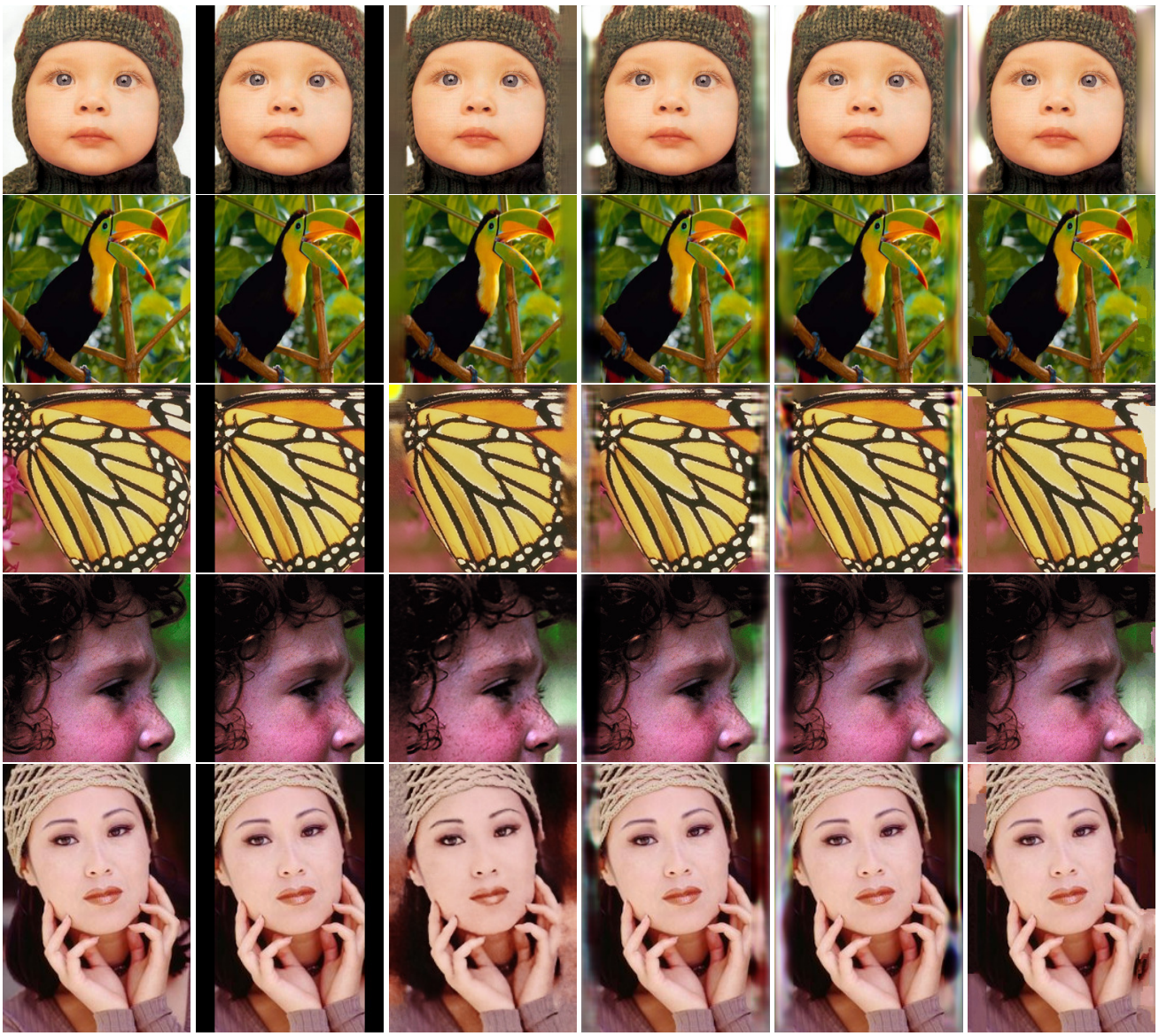
Figure 9: Outpainting results on set 14 data set - II.

	Baboon	F16	House	Lena	Peppers	Kodim01	Kodim02	Kodim03	Kodim12	**Avg**
DIP [8]	0.87	0.92	0.93	0.97	0.90	0.80	0.89	0.91	0.90	**0.90**
MEDS [2]	0.88	0.91	0.92	0.95	0.87	0.78	0.86	0.88	0.89	**0.88**
InGAN [7]	0.88	0.91	0.94	0.96	0.89	0.79	0.88	0.90	0.90	**0.89**
DeepCFL	0.89	0.93	0.95	0.96	0.91	0.80	0.89	0.91	0.91	**0.91**

Table 7: Detailed SSIM comparison for Inpainting.

	Baboon	F16	House	Lena	Peppers	Kodim01	Kodim02	Kodim03	Kodim12	**Avg**
DIP [8]	24.86	26.13	28.89	31.36	24.39	24.05	24.71	26.68	24.45	**26.16**
MEDS [2]	24.99	21.32	26.34	29.20	22.46	22.82	25.05	25.45	26.08	**24.85**
InGAN [7]	25.24	24.08	27.67	31.22	23.10	21.94	26.81	26.03	26.80	**25.87**
DeepCFL	26.04	22.56	28	31.14	22.96	24.22	26.02	26.2	27.39	**26.05**

Table 8: Detailed PSNR comparison for Inpainting.

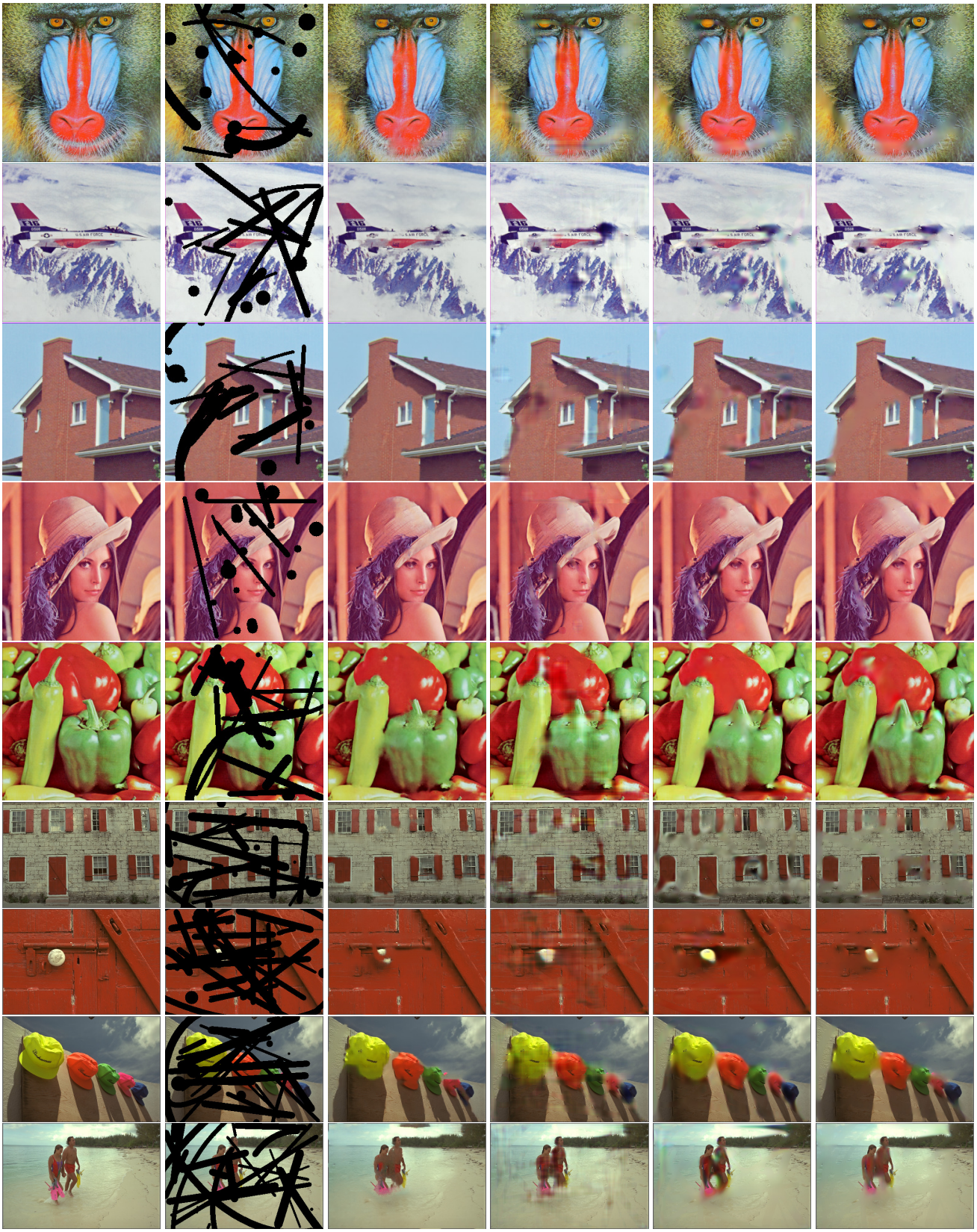


(a) Original **(b)** Masked **(c)** DIP [8] **(d)** MEDS [2] **(e)** InGAN [7] **(f)** DeepCFL

Figure 10: Outpainting results on set 5 data set.

	Barbara	Boat	Cameraman	Couple	Fingerprint	Hill	House	Lena	Man	Montage	Peppers	**Avg**
DIP [8]	0.93	0.89	0.88	0.94	0.96	0.93	0.96	0.93	0.87	0.94	0.92	**0.92**
MEDS [2]	0.93	0.92	0.93	0.93	0.96	0.92	0.95	0.95	0.92	0.92	0.92	**0.93**
InGAN [7]	0.88	0.92	0.93	0.91	0.93	0.91	0.94	0.95	0.93	0.93	0.93	**0.92**
DeepCFL	0.888	0.92	0.92	0.91	0.93	0.89	0.93	0.95	0.93	0.92	0.92	**0.92**

Table 9: Detailed SSIM comparison for Restore WC 50%.



(a) Original **(b)** Corrupted **(c)** DIP [8] **(d)** MEDS [2] **(e)** InGAN [7] **(f)** Ours

Figure 11: Visual comparison for Inpainting.

(a) Original **(b)** Corrupted **(c)** DIP [8] **(d)** MEDS [2] **(e)** InGAN [7] **(f)** Ours

Figure 12: Visual comparison for Restore WC 50%. Part I.

(a) Original **(b)** Corrupted **(c)** DIP [8] **(d)** MEDS [2] **(e)** InGAN [7] **(f)** Ours

Figure 13: Visual comparison for Restore WC 50%. Part II.

	Barbara	Boat	Cameraman	Couple	Fingerprint	Hill	House	Lena	Man	Montage	Peppers	**Avg**
DIP [8]	26.71	25.90	22.24	27.92	28.30	28.13	26.53	27.31	25.39	23.63	25.30	**26.12**
MEDS [2]	26.07	26.91	21.82	26.74	27.24	26.82	24.05	27.57	26.62	21.82	24.24	**25.44**
InGAN [7]	23.36	25.32	20.73	24.95	24.64	25.41	22.09	25.57	25.53	20.63	24.30	**23.86**
DeepCFL	24.24	26.78	23.44	26.83	25.64	26.70	22.61	28.15	26.97	22.50	21.90	**25.06**

Table 10: Detailed PSNR comparison for Restore WC 50%.

	Barbara	Boat	Cameraman	Couple	Fingerprint	Hill	House	Lena	Man	Montage	Peppers	**Avg**
DIP [8]	0.79	0.85	0.82	0.84	0.84	0.83	0.92	0.91	0.85	0.91	0.87	0.86
MEDS [2]	0.77	0.86	0.84	0.85	0.81	0.85	0.91	0.90	0.86	0.91	0.88	0.86
InGAN [7]	0.77	0.82	0.83	0.80	0.71	0.79	0.92	0.88	0.79	0.90	0.88	0.83
DeepCFL	0.78	0.82	0.83	0.81	0.73	0.80	0.92	0.90	0.83	0.89	0.89	0.84

Table 11: Detailed SSIM comparison for Restore 90%.

(a) Original **(b)** Corrupted **(c)** DIP [8] **(d)** MEDS [2] **(e)** InGAN [7] **(f)** Ours

Figure 14: Visual comparison for Restore 90%. Part I.

(a) Original **(b)** Corrupted **(c)** DIP [8] **(d)** MEDS [2] **(e)** InGAN [7] **(f)** Ours

Figure 15: Visual comparison for Restore 90%. Part II.

SI-1		SI-2		Set 5		Set 14	
Baboon	256×256	Barbara	512×512	01	256×256	01	256×256
F16	256×256	Boat	512×512	02	256×256	02	256×210
House	256×256	Cameraman	256×256	03	256×256	03	256×256
Lena	256×256	Couple	512×512	04	256×256	04	256×209
Peppers	256×256	Fingerprint	512×512	05	179×256	05	163×256
Kodim01	384×256	Hill	512×512			06	256×256
Kodim02	384×256	House	256×256			07	256×188
Kodim03	384×256	Lena	512×512			08	256×210
Kodim12	384×256	Man	512×512			09	256×256
		Montage	256×256			10	256×256
		Peppers	256×256			11	256×171
						12	256×256
						13	205×256
						14	256×171

Table 12: The table show the size of the images used for our experiments. SI-1 denotes the standard images for region inpainting and outpainting. SI-2 denotes the standard images for restoration of $x\%$ pixels.

References

[1] Deep image analogy. `https://github.com/MingtaoGuo/Deep-image-analogy-TensorFlow/`, 2018.

[2] Indra Deep Mastan and Shanmuganathan Raman. Multi-level encoder-decoder architectures for image restoration. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019.

[3] Indra Deep Mastan and Shanmuganathan Raman. Dcil: Deep contextual internal learning for image restoration and image retargeting. *WACV*, 2020.

[4] Nicholas Kolkin, Jason Salavon, and Gregory Shakhnarovich. Style transfer by relaxed optimal transport and self-similarity. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 10051–10060, 2019.

[5] Roey Mechrez, Itamar Talmi, and Lihi Zelnik-Manor. The contextual loss for image transformation with non-aligned data. *European Conference on Computer Vision (ECCV)*, 2018.

[6] Assaf Shocher, Shai Bagon, Phillip Isola, and Michal Irani. Internal distribution matching for natural image retargeting. *arXiv preprint arXiv:1812.00231*, 2018.

[7] Assaf Shocher, Shai Bagon, Phillip Isola, and Michal Irani. Ingan: Capturing and remapping the dna of a natural image. In *International Conference on Computer Vision (ICCV)*, 2019.

[8] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Deep image prior. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.