DeepCFL: Deep Contextual Features Learning from a Single Image

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

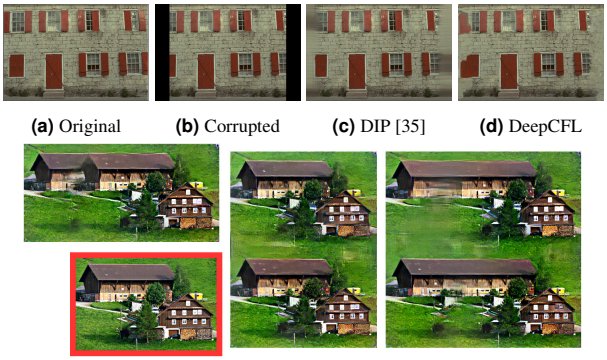
Abstract

Recently, there is a vast interest in developing image feature learning methods that are independent of the training data, such as deep image prior [35], InGAN [28, 29], SinGAN [27], and DCIL [8]. These methods perform various tasks, such as image restoration, image editing, and image synthesis. In this work, we proposed a new training data-independent framework, called Deep Contextual Features Learning (DeepCFL), to perform image synthesis and image restoration based on the semantics of the input image. The contextual features are simply the high dimensional vectors representing the semantics of the given image. DeepCFL is a single image GAN framework that learns the distribution of the context vectors from the input image. We show the performance of contextual learning in various challenging scenarios: outpainting, inpainting, and restoration of randomly removed pixels. DeepCFL is applicable when the input source image and the generated target image are not aligned. We illustrate image synthesis using DeepCFL for the task of image resizing.

1. Introduction

Recently, there has been a remarkable success for image restoration and image synthesis methods that do not use training data [7, 8, 27, 28, 30, 31, 35] . One of the major challenges for the deep feature learning methods above is the limited contextual understanding in the absence of feature learning from training samples [7]. Contextual learning is mostly studied for image inpainting [25] and image transformation tasks [23], where many pairs of source and target images are used to learn the image context.

Restoration of missing pixels in an image is a classical inverse problem [4–6, 10, 18, 32, 33, 44, 46]. It addresses various applications such as image editing, restoration of damaged paintings, image completion, and image outpainting. The image transformation model allows formulation for a variety of tasks such as style transfer, single image animation, and domain transfer [23].



(a) Original **(b)** Corrupted **(c)** DIP [35] **(d)** DeepCFL

(e) Image synthesis using DeepCFL (input image in red color frame).

Figure 1: The figure show image restoration (first row) and image synthesis (second row). Here, DIP [35] is a pixel-loss based setup. DeepCFL is a single image GAN framework for contextual learning. DeepCFL could fill the masked regions well for image restoration and also perform new object synthesis, which could not be performed using the pixel-based comparison of DIP [35].

Traditionally, image restoration is formulated as optimization problems, where the objective function includes a loss term and an image prior term, *e.g.*, sparse [1, 9] and low-rank [13] priors. The desired image is reconstructed by finding the solution for the optimization problem. Deep learning models have shown an ability to capture image priors implicitly by minimizing the loss over the training samples [3, 17, 19, 25, 36–38, 40, 43]. However, training data-based methods have their limitations, such as generalizability to new images [7, 35].

Recently, there is a growing interest in developing methods that are independent of training data to perform image restoration and image synthesis tasks [7, 8, 27, 28, 30, 31, 35]. Ulyanov *et al.* proposed deep image prior (DIP) [35], which shows that the handcrafted structure of the convolution neural network (CNN) provides an implicit image prior [35]. However, image prior learning using pixel-to-pixel loss in [35] is limited to the tasks which have a spatial correspondence between the pixels of the source image and the target image [23]. One approach would be to learn the internal

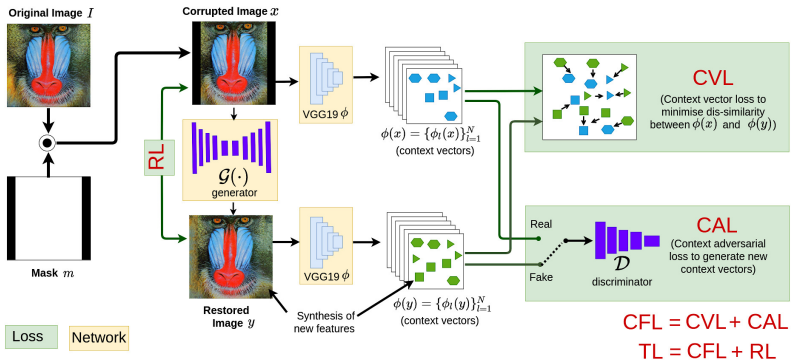


Figure 2: Contextual Features Learning (DeepCFL). The figure shows the framework for the outpainting. The corrupted image x is fed into the generator $\mathcal{G}$. Here, $\mathcal{G}$ is an encoder-decoder network which outputs an image $\mathcal{G}(x) = y$. Next, VGG19 (ϕ) computes the contextual features $\phi(x)$ of x and $\phi(y)$ of y. Then we compute contextual features loss (CFL) and reconstruction loss (RL) and minimize total loss (TL). The main idea of DeepCFL is to synthesize new features by comparing image statistics at contextual features space. Note that CFL compares the context vectors $\phi(x)$ and $\phi(y)$ using CVL and CAL (see Fig. 9 for above example).

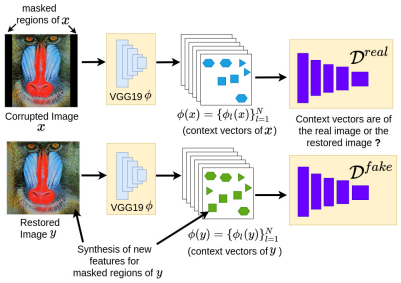


Figure 3: The figure shows the context vectors comparison in the adversarial framework. $\phi(x)$ and $\phi(y)$ are computed from the input image x and the restored image $\mathcal{G}(x) = y$. The masked regions of x are restored in y. For simplicity, we address restored regions in y as masked regions of y. Here, $\mathcal{D}^{real}$ and $\mathcal{D}^{fake}$ are the two instances of $\mathcal{D}$ which share the network parameters.

patch distribution from the input image when the source and the target images are not aligned.

The single image GAN frameworks show applications where the spatial mapping between the source and the target images is not well-defined [27–30]. Shocher *et al.* proposed an internal learning (IL) framework to synthesize realistic image patches using image-conditional GAN, called InGAN [28, 29]. Shaham *et al.* showed an unconditional generative model for image synthesis, named SinGAN [27]. Mastan *et al.* have shown the single image GAN framework for denoising-super resolution and image resizing [8].

The pixel-to-pixel loss framework in [35] and the internal patch distribution learning frameworks in [27–29] do not perform image reconstruction by considering the context of the objects. An image could be considered as a collection of high dimensional context vectors [23]. These high dimensional vectors are the image statistics captured at the intermediate layers of the features extractor such as VGG19 network [22, 23]. An interesting question would be that, given an incomplete image summary, can we synthesize new context vectors and use them to reconstruct the image. The context of an image is critical to perform image restoration and image synthesis tasks (Fig. 1 and Fig. 8) [7, 28, 30, 35]. We present a single image GAN framework (DeepCFL) which studies the contextual features in the image. The problem is *novel* as it aims to learn the distribution of the contextual features (contextual learning) in the image instead of internal patch distribution, as in the case of InGAN [28, 29] and SinGAN [27].

We have shown a pictorial representation of DeepCFL in Fig. 2. The aim is to utilize the image features of the original image I, which are present in the corrupted image x. We generate a restored image y which utilizes image features from x. We use an encoder-decoder network $\mathcal{G}$ to generate y. Then, we iteratively minimize the total loss (TL) between the corrupted image and the restored image. TL is a combination of contextual features loss (CFL) and reconstruction loss (RL). Fig. 2 shows that CFL allows feature learning using two different tools: contextual adversarial loss (CAL) and context vectors loss (CVL). The detailed description of each component of the framework and the formal definitions of the loss functions are described Sec. 3.

CAL performs distribution matching in the adversarial framework to synthesize new context vectors for the corrupted image x. CVL computes the direct difference between the context vectors extracted from the corrupted image x and the restored image y. Therefore, in CFL, CAL generates new context vectors and CVL improvises them. RL is a pixel-to-pixel loss (*i.e.*, mean squared error), which ensures the preservation of image features in the restored images. Intuitively, the main idea is to generate new context vectors using CFL and map them to the image features implicitly through pixel-based comparison using RL.

We have studied the performance of DeepCFL for the following tasks: image outpainting, inpainting of arbitrary holes, and restoration of $r\%$ pixels missing in the corrupted image. We also show the applications in the presence of non-aligned image data using image resizing. The key contributions of this work are summarized below.

- We propose a single image GAN framework for contextual features learning (DeepCFL). The framework performs well on image outpainting tasks (Fig. 4 and Ta-

ble 1). We also illustrate that DeepCFL synthesizes new objects when resizing the image (Fig. 6).
- DeepCFL investigates image reconstruction considering the contextual features. The contextual features learning is useful for the applications that use only a single image as input. We show the generalizability of DeepCFL by performing multiple applications (Sec. 4).
- We provide a detailed analysis of contextual features learning by illustrating reconstruction in various challenging setups such as arbitrary hole inpainting, restoration of a high degree of corruption, restoration of images with a word cloud, ablation studies, and limitations (Sec. 4 and Sec. 5).

2. Related work

Deep feature learning captures good image features by using the strong internal data repetitions (self-similarity prior) [11, 15, 30, 42, 45], hand-crafted structure [7, 35], and explicit regularizer [21]. DeepCFL is a single image GAN setup, which is different from features learning frameworks proposed earlier [19, 23, 24, 26, 34, 37, 39, 41]. Single image GAN frameworks performs variety of tasks such as image editing [27], retargeting [29], denoising super-resolution [8], and video inpainting [16, 42]. Our contextual learning framework is somewhat related to [7,8,28,35]. InGAN [28, 29] and SinGAN [27] are single image GAN frameworks for learning the internal patch distribution. DCIL leverage internal learning with the contextual loss [8]. DeepCFL is related to [7, 8, 28, 35] and does not employ a masked patch discriminator for CAL [37]. It does not use a features expansion network and relies on the features reconstruction capabilities of the encoder-decoder network [37].

3. Our Framework

DeepCFL is a single image GAN framework to synthesize new context vectors that are consistent with the semantics of the input source image. The task is to extract features from the source image and synthesize a new target image. The source image could be a clean or a corrupted image. The target image could be of the same size as the source image or a different size. For example, in the case of image restoration, we use a corrupted source image with missing pixel values. The contextual features are used to fill the missing regions of the corrupted image. For image synthesis, a clean image is used to synthesize new images of different sizes. Below, we discuss image restoration and context vectors before we describe the DeepCFL framework.

Let $\mathcal{I}$ denote the set of original images, X denote the set of corrupted images, and Y denote the set of restored images. Let x denotes a corrupted image, *i.e.*, $x \in X$. x is computed by removing pixels from an original image I using a binary mask m as follows: $x = I \odot m$, where $\odot$ is the Hadamard product and $I \in \mathcal{I}$. The mask m defines the underlying image restoration application. For example, in image outpainting of 20% pixels, the mask removes the 10% pixels each along the right side and the left side of the image. For the restoration of $r\%$ pixels, the mask contains $r\%$ zeros at random locations. For image inpainting, the mask contains arbitrary shapes. The objective is to restore the image details in x, which were removed by m.

Image restoration procedure. The task is to generate a new image $\mathcal{G}(x) = y$, which contains the restored pixels. Here, $\mathcal{G}$ is the generator network which maps the corrupted image to a restored image y, *i.e.*, $y \in Y$. The corrupted image x could be considered as a source image as it contains the features from the original image I. The main intuition is to estimate the context for masked regions of y based on the image features present at the unmasked regions of x (Fig. 3). The image restoration process iteratively minimizes the loss computed between x and y.

What are context vectors? The context vectors of an image I are the image statistics present at intermediate layers of a feature extractor $\phi(I)$. VGG19 has been widely used to extract image statistics. Formally, given an image I, let $\phi(I) = \{\phi_l(I)\}_{l=1}^{N}$ denote the set of context vectors extracted from I. Here, $\phi : \mathcal{I} \rightarrow CV$ is the pre-trained VGG19 network [12] which maps image $I \in \mathcal{I}$ to its context vectors $\phi(I) \in CV$. $\phi_l(\cdot)$ denotes the feature extracted from the layer l of $\phi(\cdot)$ and N is the number of layers in ϕ.

Why context vectors are important? Fig. 1 and Fig. 8 show that the contextual learning framework would allow image restoration and image synthesis based on the semantics of the input (refer Fig. 4 and Fig. 6 for more examples). For example, in the case of restoration of missing pixels, the key observation is to improve the masked regions in the restore image y using the unmasked regions in the corrupted image x. It is done by matching the distribution of the contextual features of the corrupted image $\phi(x)$ and the contextual features of the restored image $\phi(y)$ (Sec. 3.2).

DeepCFL. We now discuss the DeepCFL framework shown in Fig. 2. It consists of a generator $\mathcal{G}$, a discriminator $\mathcal{D}$, and a features extractor ϕ. The corrupted image x is fed into $\mathcal{G}$. The generator outputs an image $y = \mathcal{G}(x)$. Next, we feed x and y into $\phi(\cdot)$ to compute $\phi(x)$ and $\phi(y)$. Then we minimize the total loss (TL) computed between x and y (Eq. 1). The two primary components of TL are the contextual features loss (CFL) and the reconstruction loss (RL). CFL synthesizes new context vectors for the masked regions in $\phi(y)$, where the features learning procedure is assisted by contextual features in $\phi(x)$. $\mathcal{D}$ is used for computing CFL. RL is computed between the unmasked regions of x and y to provide image feature consistency in y.

3.1. Network Design

Generator. The generator $\mathcal{G} : X \rightarrow Y$ maps the source image $x \in X$ to the target image $y \in Y$. $\mathcal{G}$ is a depth-5 encoder-decoder network without skip connections (ED). The ED architecture works as the implicit regularizer to stabilize the image feature learning [7, 35]. It exploits the inherent self-similarity present in the source image. We use context normalization [37] to maximize features learning. Intuitively, DeepCFL is unsupervised in the sense that no training data are used to train the generator network for any of the tasks. It is a single image GAN framework which uses pre-trained VGG19 as the features extractor. VGG19 is widely used in style transfer works for defining loss at VGG features space. The feature extractor distills strong prior in the framework [8].

Discriminator. The discriminator $\mathcal{D} : CV \rightarrow \mathcal{M}$ maps the context vectors to a discriminator map $\mu \in \mathcal{M}$, where each entry in μ denotes the probability of the context vector coming from the distribution of the contextual feature of the original image. Fig. 3 illustrates the discriminator task to distinguish context vectors $\phi(x)$ and $\phi(y)$. The generator $\mathcal{G}$ learns the context vectors through its interaction with $\mathcal{D}$. We use a multi-scale discriminator (MSD), where each output is a weighted average of the output from several discriminators (we have illustrated $\mathcal{D}$ using a single CNN for simplicity in Fig. 2 and Fig. 3). Note that the discriminators in MSD would resize the context vectors.

3.2. Loss Function

The goal of the loss function is to maximize the feature learning from source x by comparing it with generated image $\mathcal{G}(x) = y$. The total loss (TL) is defined in Eq. 1.

$$\begin{aligned}\mathcal{L}_{tl}(x, y, \mathcal{G}, \mathcal{D}, \phi) = \lambda_{\mathcal{G}}\ \mathcal{L}_{cfl}(x, y, \mathcal{G}, \mathcal{D}, \phi) \\ +\lambda_{\mathcal{R}}\ \mathcal{L}_{rl}(x, y, \mathcal{G})\end{aligned} \tag{1}$$

Here, $\mathcal{L}_{cfl}$ denotes CFL and $\mathcal{L}_{rl}$ denotes RL. The terms $\lambda_{\mathcal{G}}$ and $\lambda_{\mathcal{R}}$ are the coefficients of CFL and RL. We have pictorially shown CFL and RL in Fig. 2. The total loss described in Eq. 1 compares the image features in two ways: CFL and RL. CFL provides new image features to y, which are consistent with the object context of x. RL maximizes the likelihood of randomly initialized network weights.

3.2.1 Contextual Features Loss (CFL)

The purpose of CFL is to learn the distribution of context vectors to synthesize image features in y based on the semantics of the input x. We extract context vectors $\phi(x)$ and $\phi(y)$ and then minimize the loss described in Eq. 2.

$$\begin{aligned}\mathcal{L}_{cfl}(x, y, \mathcal{G}, \mathcal{D}, \phi) = \lambda_{cal}\mathcal{L}_{cal}(\mathcal{G}, \mathcal{D}; \phi) \\ + \lambda_{cvl}\mathcal{L}_{cvl}(\phi(x), \phi(y))\end{aligned} \tag{2}$$

Here, $\mathcal{L}_{cfl}$ denotes CFL, $\mathcal{L}_{cal}$ denotes CAL, and $\mathcal{L}_{cvl}$ denotes CVL. λ_{cal} and λ_{cvl} are the coefficients of CAL and CVL. Eq. 2 shows that CFL compares the context vectors in two ways. (1) Context vector comparison in the adversarial framework using CAL. (2) Contextual features comparison by computing cosine distance in CVL. CAL is an adversarial loss computed using the generator $\mathcal{G}$ and the discriminator $\mathcal{D}$. It is aimed to synthesize new contextual features that are indistinguishable from the features of the source image. The CVL computes the difference between contextually similar vectors to make the synthesized features of y similar to the features of x.

Context Adversarial Loss (CAL). We have used the LSGAN [20] variant of the adversarial learning framework.

$$\mathcal{G}^* = \min_{\mathcal{G}} \max_{\mathcal{D}} \mathcal{L}_{cal}(\mathcal{G}, \mathcal{D}; \phi) \tag{3}$$

Here, $\mathcal{G}^*$ is the generator with optimal parameters. The loss $\mathcal{L}_{cal}$ is defined in Eq. 4.

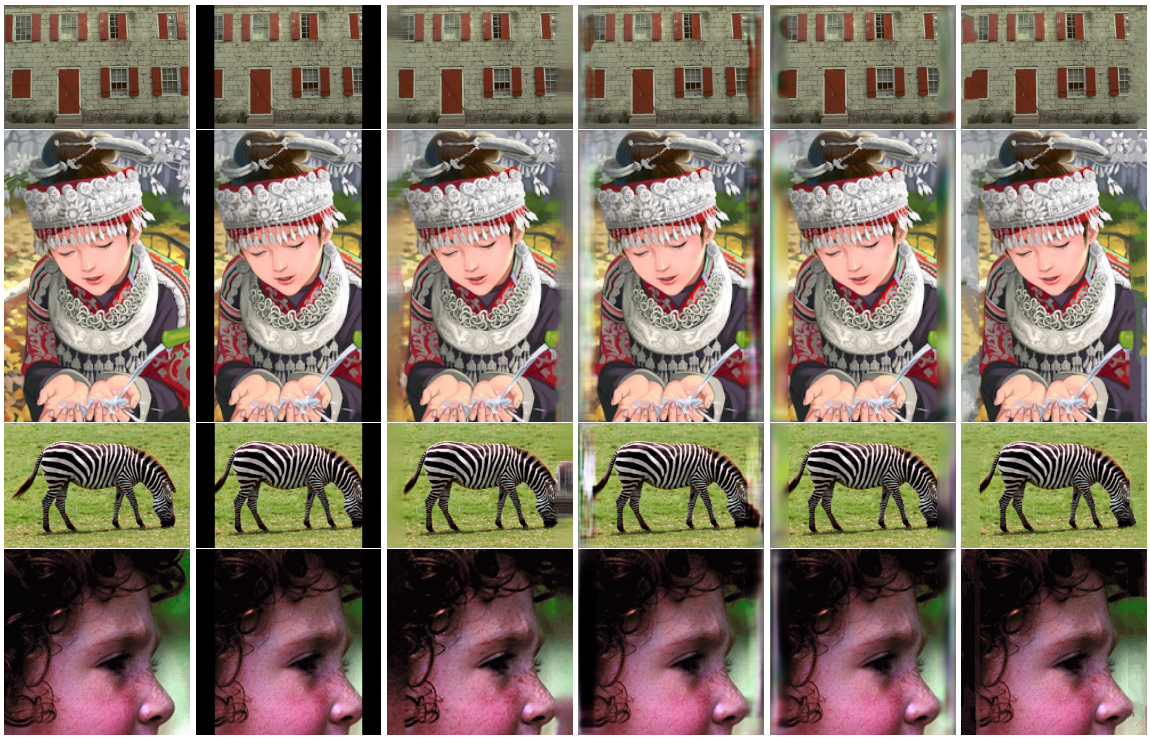
$$\begin{aligned}\mathcal{L}_{cal}(\mathcal{G}, \mathcal{D}; \phi) =& \mathbb{E}_{x \sim pdata(x)}[(\mathcal{D}(\phi(x)) - 1)^2] \\ &+ \mathbb{E}_{x \sim pdata(x)}[\mathcal{D}(\phi(\mathcal{G}(x)))^2]\end{aligned} \tag{4}$$

Eq. 4 shows the distribution matching of context vectors of the restored image $\phi(y) = \phi(\mathcal{G}(x))$ and context vectors of the corrupted image $\phi(x)$. The discriminator $\mathcal{D}$ tries to determine whether the context vectors are from x or y (see Fig. 2 and Fig. 3). Intuitively, this would help us to fill the context of the masked regions of $y = \mathcal{G}(x)$ by learning the context of the objects in unmasked areas in x. We have described $\mathcal{G}$, $\mathcal{D}$, and ϕ in Sec. 3.1.

Context Vector Loss (CVL). The main purpose of CVL is to improve the quality of contextual features in $\phi(y)$ learned by CAL. $\mathcal{L}_{cvl}(\phi_l(x), \phi_l(y))$ is the sum of the contextual loss [23] computed at each layer l in ϕ. We have defined CVL for layer l in Eq. 5.

$$\mathcal{L}_{cvl}(\phi_l(x), \phi_l(y), l) = -\log(CX(\phi_l(x), \phi_l(y))) \tag{5}$$

Here, CX is the contextual similarity defined using the cosine distance between the features contained in $\phi_l(x)$ and $\phi_l(y)$. Note that CX is computed by finding for each feature $\phi_l(y)_j$, a feature $\phi_l(x)_i$ that is most similar to it and then summed for all $\phi_l(y)_j$. Fig. 2 illustrate the matched context vectors of $\phi_l(x)_i$ and $\phi_l(y)_j$ by an arrow. Intuitively, the feature matching performed between the context vectors of masked regions of y and the context vectors of unmasked regions of x enables feature refinements for the new context vectors created by CAL. We used $conv4\_2$ layer of ϕ to compute context vectors as the higher layers capture the high-level content in terms of objects structure [12]. It is interesting to note that CVL is different from perceptual loss $\|\phi_l(x) - \phi_l(y)\|$, which computes features difference without using contextual similarity criterion.



(a) Original **(b)** Corrupted **(c)** DIP [35] **(d)** MEDS [7] **(e)** InGAN [28] **(f)** DeepCFL

Figure 4: Image outpainting. The figure shows the restoration of 20% pixels in the image. DIP [35] and MEDS [7] fill the missing regions but do not preserve the structure of the objects. Internal learning of InGAN [28] performed better, but the generated new image features are not very clear. DeepCFL incorporates the contextual understanding and is observed to perform better (Table 1).

3.2.2 Reconstruction Loss (RL).

RL is aimed to preserve image features and it is computed between corrupted image x and restored image $\mathcal{G}(x) = y$ (Fig. 2). Let $\mathcal{L}_{rl}$ denotes RL. We define $\mathcal{L}_{rl}$ in Eq. 6.

$$\mathcal{L}_{rl}(\mathcal{G}, x, y) = \|\mathcal{G}(x) \odot m - x\| \tag{6}$$

Eq. 6 shows the comparison between unmasked regions of x with the unmasked regions of y. The unmasked regions in x contains image features from I and masked regions in x are corrupted due to mask, *i.e.*, $x = I \odot m$. RL is a pixel-wise loss and it imposes a strong self-similarity prior [35].

4. Applications

Here, we discuss the following applications of DeepCFL. (1) Image outpainting: extension of an image along the sides. (2) Image inpainting of irregular holes in the image. (3) Content-aware image resizing: synthesis of new objects when we resize an image. (4) Restoration in the presence of high degree of corruption: 50% pixels[1].

[1]We have used original implementations of DIP [35], MEDS [7], and DCIL [8]. We implemented image restoration using the internal learning of InGAN [28]. We have provided the implementation details in the supplementary material.

4.1. Image Outpainting.

Image outpainting relates to image extension, which creates new features while maintaining the semantics of the scene. Image extension uses training data to learn image context and then generates the complete scene given partial information [25, 34, 37, 39]. Our outpainting task does not use any training samples and synthesize features using only the corrupted image. We address outpainting as an image extension for convenience.

A good image outpainting approach would fill the image features based on the semantics of the object present at the boundaries. The ability of the generator to synthesize new contextual features over a large spatial extent along the sides depends upon the contextual learning. Unlike pixel-to-pixel loss, the context vectors based loss functions CFL (Eq. 2) aims to fill new features in the masked regions of the restored image, which are semantically similar to the unmasked regions of the corrupted image (refer Sec. 3).

In Fig. 4, we show outpainting of 20% missing pixels, where the corrupted image is generated by removing 10% pixels along the right side and the left side. DIP [35], MEDS [7], and InGAN [28] are contextual features learning independent methods. Image outpainting is better achieved

(a) Original image **(b)** Corrupted **(c)** DIP, 0.90 **(d)** MEDS, 0.89 **(e)** InGAN, 0.90 **(f)** DeepCFL, 0.91

Figure 5: Inpainting. The figure shows the inpainting of arbitrary holes for DIP [35], MEDS [7], InGAN [28], and DeepCFL (ours). DeepCFL minimize the features spillover of trees and attains better perceptual quality.

using the semantics of the objects in the contextual learning-based DeepCFL framework. Table 1 shows the quantitative comparison on the standard datasets from [14], Set5 and Set14 datasets [7]. It could be observed that DeepCFL outperforms the other methods for outpainting. We have provided more details in the supplementary material.

	DIP [35]	**MEDS** [7]	**InGAN** [28]	**DeepCFL**
SD	0.91	0.91	**0.92**	**0.92**
	23.73	21.70	22.89	**24.13**
Set14	0.89	0.89	**0.90**	**0.90**
	22.12	20.24	21.19	**22.52**
Set5	0.88	0.88	0.89	**0.90**
	19.03	19.35	19.29	**21.50**

Table 1: Quantitative comparision using SSIM values (top) and PSNR values (bottom) for image outpainting of 20% pixels on standard dataset (SD), Set5 and Set14 datasets.

4.2. Image Inpainting.

The input image has non-uniform corrupted regions spread across the entire image in the inpainting task. It is a natural way by which an image could get corrupted [19,26]. The critical property to perform inpainting without using training data is to utilize the internal self-similarity property of the natural images [35, 42]. The computation of the MSE between the generator output and the corrupted image tends to capture strong self-similarity prior [35]. DeepCFL leverages this learning by incorporating the context vectors comparison. The features learning procedure for inpainting is similar to outpainting described in Sec 4.1.

Fig. 5 shows the visual results for arbitrary hole inpainting. It could be observed that the contextual learning of DeepCFL minimizes the features spillover between different objects and fill the arbitrary holes considering the semantics of the image. The quantitative comparison (SSIM) for inpainting is as follows: DIP [35]: 0.90, MEDS [7]: 0.88, InGAN [29] 0.90, and DeepCFL (ours): 0.91. We have provided more comparisons of generated images in the supplementary material. DeepCFL performs comparably to other frameworks. The estimation of the parameters from a single image is highly sensitive to the hyper-parameters (*e.g.*, learning rate) [7, 35]. We believe that the restoration quality of our method and other methods could be improved further using the hyper-parameter search.

4.3. Image Resize

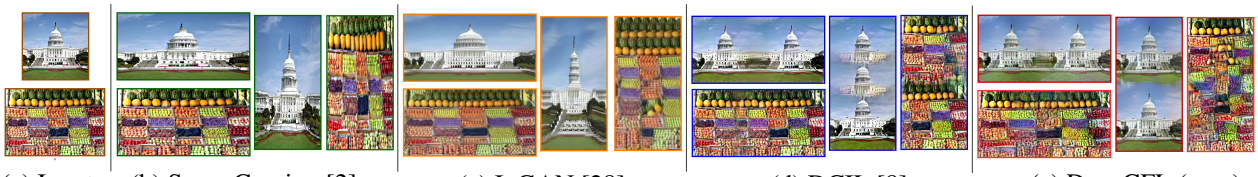
We have discussed image outpainting, which is different from content-aware image resize, where the task is to resize the image while preserving the salient objects of the image [29]. DeepCFL is able to synthesize new objects when resizing the input image (Fig. 6). The source image is scaled $2\times$ along the height and the width. Therefore, the pixel correspondence between the source and the generated target images is not well defined. The image resize is done by using the generator to scale the input and then computing the adversarial loss in a cycle consistent way.

Fig. 8 show the challenging scenario of object synthesis for various single image GAN frameworks. Inspired by InGAN [29], our framework DeepCFL studies deep contextual features. DeepCFL is different from DCIL [8] as it uses the adversarial framework on VGG features space for image outpainting. In contrast, DCIL uses the adversarial framework on the image space for Denoising-super resolution. We believe that the results of various single image GAN framework in Fig. 8 could be improvised further.

4.4. Restoration of 50% pixels.

To investigate contextual features leaning in the presence of a high degree of corruption, we perform restoration of 50% missing pixels spread across the entire image uniformly at random. It is a different setup than outpainting and inpainting, where one has to fill a missing region (*i.e.*, a contiguous array of pixels). We further increase the task difficulty by using the corrupted image containing a word cloud. We denote the above setup as RestoreWC 50% (WC denotes word-cloud). It is a challenging setup because the small font present in the corrupted image would require to fill fine image features details.

We show image restoration in RestoreWC 50% setup in Fig. 7. The quantitative comparison (SSIM) for RestoreWC 50% is as follows. DIP [35]: 0.92, MEDS [7]: 0.93, InGAN [29]: 0.92, and DeepCFL (ours): 0.92. It could be observed that DeepCFL performs comparably to other frameworks. It might be because the image features computed from the highly corrupted image might not be sufficient for



(a) Input (b) Seam Carving [2] (c) InGAN [28] (d) DCIL [8] (e) DeepCFL (ours)

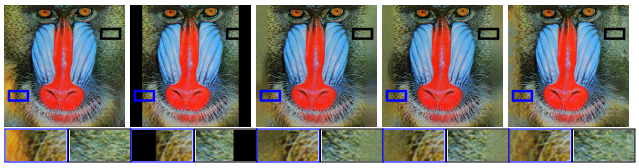
Figure 6: Image Resize. The figure shows the synthesis of small objects (fruits) and large objects (building). Seam Carving (SC) [2] does not preserve the structure well when resizing. For example, the shape of the fruits in small object synthesis is deformed in SC output. InGAN [28] preserve the structure for small objects but does not preserve for large object (building). DCIL [8] synthesizes new objects when resizing. For example, object structure is preserved well when scaling $2\times$ along the width of the building. DeepCFL also preserves object structure when synthesizing new objects. It could be observed that DeepCFL does not duplicate the objects along the expended dimension. For example, DeepCFL synthesizes the fruits when resizing (the images are best viewed after zooming).

(a) Original **(b)** Corrupted **(c)** DIP, 0.87 **(d)** MEDS, 0.92 **(e)** InGAN, 0.93 **(f)** DeepCFL, 0.93

Figure 7: RestoreWC 50%. The figure shows restoration in the presence of a word cloud. DeepCFL restore image features details comparable to DIP [35], MEDS [7], and InGAN [28].

(a) Input **(b)** InGAN [29] **(c)** DCIL [8] **(d)** DeepCFL

Figure 8: The challenge is to synthesize a new object [8, 29]. DeepCFL observed that object synthesis is achievable at a different scale, similar to DCIL [8]. DeepCFL output image with better features near the elbow, but the background is not clear.



(a) Original image **(b)** Corrupted image **(c)** DIP [35] **(d)** DIP [35] + CL **(e)** DeepCFL

Figure 9: **Ablation study (1).** The figure shows the outpainting of 20% pixels. DIP is a pixel-loss based setup. We integrated contextual loss (CL) with DIP [35] in "DIP [35] + CL" to show image restoration using CL and without GAN framework. DeepCFL is a GAN framework and it is observed to restore image features well.

restoration in the single image GAN framework. Therefore, contextual learning is a bit less effective. We believe that the pixel-based loss would not have the object synthesis abilities of the single image GAN frameworks (Fig. 8).

5. Ablation Studies and Limitations

We show the usefulness of contextual learning in the adversarial framework in Fig. 9. The restored image features are highlighted in the cropped images. It could be observed that the single image GAN framework (DeepCFL) synthesizes image features for image restoration.

In Fig. 10, we show an ablation study to disentangle the reconstruction using context vector loss (CVL), context adversarial loss (CAL), and contextual features loss (CFL) as defined in Sec. 3.2. The CFL setup performs better as it uses adversarial learning and context vector learning together.

Fig. 11 shows the restoration in the presence of two discriminator architectures setup: single scale discriminator (SSD) and multiscale discriminator (MSD). InGAN [29] shows that MSD improves the performance significantly for image synthesis. We observed that higher model capacity did not significantly improve image restoration, similar to [7] as the masked SSIM for SSD setup is (0.971) is close to MSD setup (0.976). The visual performance enhancement would be because MSD setup enforces image statistics consistency at multiple levels, which is harder than solving at a single scale SSD setup. Our intuition is that solving a hard problem would help to learn better image features [7]. Moreover, quantitative enhancement is close. Our interpretation of it is as follows. MSD in DeepCFL is operating on

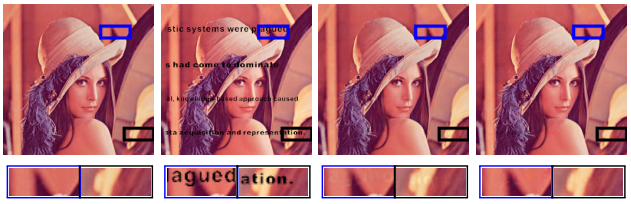
the context vectors. The scaling of the context vectors in MSD of DeepCFL and scaling the image in [8, 27, 29] are completely different operations. The performance enhancement for image restoration using the scaling of context vector might not be very effective.

Fig. 12 shows the reconstruction when the information in the corrupted image is not sufficient to fill the missing regions. The limitation is due to the lack of feature learning from the training samples in the single image GAN framework. A similar limitation has also been reported for image manipulation tasks [27]. Restoration of an object which is partially present in the image would also be exciting. However, it is not within the scope of this work.

Fig. 13 shows the the restoration of 90% pixels ($r = 90$) using image features learning from 10% pixels. It could be observed that it is difficult to understand the semantics of the scene from 10% pixels. The experiment confirms our observation that the adversarial learning of image context is less effective for the high degree of corruption. We show more results in the supplementary material.

(a) Original image **(b)** Masked image **(c)** CAL, 0.88 **(d)** CVL, 0.91 **(e)** CFL, 0.92

Figure 10: Ablation study (2). The input is the masked image, which contains 50% corrupted pixels. Here, the contextual feature loss CFL = CAL + CVL. It could be observed that CAL and CVL together enhance the restoration quality in CFL.



(a) Original **(b)** Masked **(c)** SSD **(d)** MSD

Figure 11: Ablation study (3). The figure shows text removal in the presence of single-scale discriminator (SSD) and multi-scale discriminators (MSD) setups. SSD setup makes thin marks in the restored output, which are a bit less detectable in the MSD setup.

6. Discussion

DeepCFL is a single image GAN framework. The data-driven supervised feature learning setups use paired examples of ground truth (GT) and corrupted images. The corrupted images are fed into the network and generated outputs are matched with the GT image. DeepCFL is not trained by showing training samples of GT and corrupted images. DeepCFL can be fairly compared only with training data-independent methods as they also do not use training samples. Training based methods could synthesize image feature details that are not present in the input image, which is not possible in the training data-independent setups (Fig. 12 and Fig. 13). The feature extractor VGG-19 contains layers at different scales, where each layer contains varying levels of abstractions. We believe that combining features from various VGG-19 layers would be helpful. Moreover, it would increase the model complexity. The scope of DeepCFL is limited to the contextual features present in $conv4\_2$ layer. We propose as future work to perform studies on how to increase VGG19 layers for feature comparison while minimizing the computational overhead.

(a) Original **(b)** Masked **(c)** DIP [35] **(d)** DeepCFL

Figure 12: Limitation (1). The aim is to restore a partially present object in the corrupted image (*i.e.*, head). The features in the masked image is not enough to restore head in the mirror. Therefore, we could observe that the reconstruction using DIP [35] and DeepCFL is not performed well.

(a) Real image **(b)** Masked image **(c)** DIP 0.92 **(d)** MEDS 0.91 **(e)** InGAN 0.91 **(f)** DeepCFL 0.91

Figure 13: Limitation (2). The figure shows the restoration of 90% pixels. DeepCFL preserves the image features comparable to DIP [35], MEDS [7], and InGAN [28].

7. Conclusion

We investigate deep contextual features learning (CFL) in the single image GAN framework for image restoration and image synthesis. The main challenge to accomplish the above tasks is when the information contained in the input image is not sufficient for synthesizing the necessary image features. DeepCFL synthesizes image features based on the semantics to perform outpainting, inpainting, restoration of $r\%$ pixels, and image resizing. It would be interesting to study the performance of the single image GAN framework in the setting of videos similar to [16, 42].

Acknowledgments. Indra Deep Mastan was supported by Visvesvaraya Ph.D. fellowship. Shanmuganathan Raman was supported by SERB Core Research Grant and SERB MATRICS.

References

[1] Michal Aharon, Michael Elad, Alfred Bruckstein, et al. K-svd: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on signal processing*, 54(11):4311, 2006.

[2] Shai Avidan and Ariel Shamir. Seam carving for content-aware image resizing. In *ACM Transactions on graphics (TOG)*, volume 26, page 10. ACM, 2007.

[3] Siavash Arjomand Bigdeli and Matthias Zwicker. Image restoration using autoencoding priors. *arXiv preprint arXiv:1703.09964*, 2017.

[4] Antoni Buades, Bartomeu Coll, and J-M Morel. A non-local algorithm for image denoising. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 2, pages 60–65. IEEE, 2005.

[5] Harold C Burger, Christian J Schuler, and Stefan Harmeling. Image denoising: Can plain neural networks compete with bm3d? In *2012 IEEE conference on computer vision and pattern recognition*, pages 2392–2399. IEEE, 2012.

[6] Kostadin Dabov, Alessandro Foi, Vladimir Katkovnik, and Karen Egiazarian. Image denoising by sparse 3-d transform-domain collaborative filtering. *IEEE Transactions on image processing*, 16(8):2080–2095, 2007.

[7] 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.

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

[9] Weisheng Dong, Lei Zhang, Guangming Shi, and Xin Li. Nonlocally centralized sparse representation for image restoration. *IEEE transactions on Image Processing*, 22(4):1620–1630, 2012.

[10] Alexei A Efros and Thomas K Leung. Texture synthesis by non-parametric sampling. In *Proceedings of the seventh IEEE international conference on computer vision*, volume 2, pages 1033–1038. IEEE, 1999.

[11] Michael Elad and Michal Aharon. Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Transactions on Image processing*, 15(12):3736–3745, 2006.

[12] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2414–2423, 2016.

[13] Shuhang Gu, Qi Xie, Deyu Meng, Wangmeng Zuo, Xiangchu Feng, and Lei Zhang. Weighted nuclear norm minimization and its applications to low level vision. *International journal of computer vision*, 121(2):183–208, 2017.

[14] Felix Heide, Wolfgang Heidrich, and Gordon Wetzstein. Fast and flexible convolutional sparse coding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5135–5143, 2015.

[15] Daniel Glasner Shai Bagon Michal Irani. Super-resolution from a single image. In *Proceedings of the IEEE International Conference on Computer Vision, Kyoto, Japan*, pages 349–356, 2009.

[16] Dahun Kim, Sanghyun Woo, Joon-Young Lee, and In So Kweon. Deep video inpainting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5792–5801, 2019.

[17] Christian Ledig, Lucas Theis, Ferenc Huszar, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, and Wenzhe Shi. Photo-realistic single image super-resolution using a generative adversarial network. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[18] Anat Levin. Blind motion deblurring using image statistics. In *Advances in Neural Information Processing Systems*, pages 841–848, 2007.

[19] Guilin Liu, Fitsum A Reda, Kevin J Shih, Ting-Chun Wang, Andrew Tao, and Bryan Catanzaro. Image inpainting for irregular holes using partial convolutions. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 85–100, 2018.

[20] Xudong Mao, Qing Li, Haoran Xie, Raymond YK Lau, Zhen Wang, and Stephen Paul Smolley. Least squares generative adversarial networks. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2794–2802, 2017.

[21] Gary Mataev, Peyman Milanfar, and Michael Elad. Deepred: Deep image prior powered by red. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 0–0, 2019.

[22] Roey Mechrez, Itamar Talmi, Firas Shama, and Lihi Zelnik-Manor. Learning to maintain natural image statistics. *arXiv preprint arXiv:1803.04626*, 2018.

[23] 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.

[24] Kamyar Nazeri, Eric Ng, Tony Joseph, Faisal Z Qureshi, and Mehran Ebrahimi. Edgeconnect: Generative image inpainting with adversarial edge learning. *arXiv preprint arXiv:1901.00212*, 2019.

[25] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2536–2544, 2016.

[26] Yurui Ren, Xiaoming Yu, Ruonan Zhang, Thomas H Li, Shan Liu, and Ge Li. Structureflow: Image inpainting via structure-aware appearance flow. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 181–190, 2019.

[27] Tamar Rott Shaham, Tali Dekel, and Tomer Michaeli. Singan: Learning a generative model from a single natural image. *The IEEE International Conference on Computer Vision (ICCV)*, 2019.

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

[29] 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.

[30] Assaf Shocher, Nadav Cohen, and Michal Irani. zero-shot super-resolution using deep internal learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3118–3126, 2018.

[31] Oleksii Sidorov and Jon Yngve Hardeberg. Deep hyperspectral prior: Single-image denoising, inpainting, super-resolution. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 0–0, 2019.

[32] Denis Simakov, Yaron Caspi, Eli Shechtman, and Michal Irani. Summarizing visual data using bidirectional similarity. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2008.

[33] Jian Sun, Zongben Xu, and Heung-Yeung Shum. Image super-resolution using gradient profile prior. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2008.

[34] Piotr Teterwak, Aaron Sarna, Dilip Krishnan, Aaron Maschinot, David Belanger, Ce Liu, and William T Freeman. Boundless: Generative adversarial networks for image extension. *arXiv preprint arXiv:1908.07007*, 2019.

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

[36] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[37] Yi Wang, Xin Tao, Xiaoyong Shen, and Jiaya Jia. Wide-context semantic image extrapolation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1399–1408, 2019.

[38] Chao Yang, Xin Lu, Zhe Lin, Eli Shechtman, Oliver Wang, and Hao Li. High-resolution image inpainting using multi-scale neural patch synthesis. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, page 3, 2017.

[39] Zongxin Yang, Jian Dong, Ping Liu, Yi Yang, and Shuicheng Yan. Very long natural scenery image prediction by outpainting. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 10561–10570, 2019.

[40] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Generative image inpainting with contextual attention. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5505–5514, 2018.

[41] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Free-form image inpainting with gated convolution. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4471–4480, 2019.

[42] Haotian Zhang, Long Mai, Ning Xu, Zhaowen Wang, John Collomosse, and Hailin Jin. An internal learning approach to video inpainting. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2720–2729, 2019.

[43] Kai Zhang, Wangmeng Zuo, Shuhang Gu, and Lei Zhang. Learning deep cnn denoiser prior for image restoration. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[44] Lei Zhang and Wangmeng Zuo. Image restoration: From sparse and low-rank priors to deep priors [lecture notes]. *IEEE Signal Processing Magazine*, 34(5):172–179, 2017.

[45] Maria Zontak and Michal Irani. Internal statistics of a single natural image. In *CVPR 2011*, pages 977–984. IEEE, 2011.

[46] Daniel Zoran and Yair Weiss. From learning models of natural image patches to whole image restoration. In *2011 International Conference on Computer Vision*, pages 479–486. IEEE, 2011.