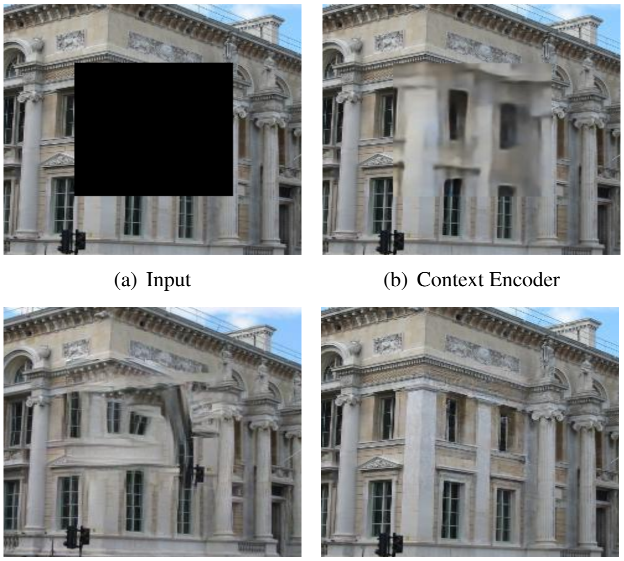
Image Inpainting via Enhanced Generative Adversarial Network

Anonymous CVPR submission

Paper ID 1377

Abstract

In this paper, we propose an Enhanced Generative Model for Image Inpainting (EGMII). Unlike most state-of-the-art algorithms using extra constraints to enforce the generator network recover semantic texture details, we construct an end-to-end network to generate both the image content and high-frequency details progressively. Our model contains a two-phase generator and two discriminators. In our generator, the previous phase restores the structure information via convolutional encoder-decoder architecture, and the following phase captures high-frequency details via residual learning. For each generator phase, we define different objective functions and further optimize the entire network via a feed-forward manner. Moreover, for the generator in the second phase, we adopt a deep residual architecture, which can eliminate the perceptual discontinuity on the border of the missing region. Experimental results on both CelebA and Oxford Buildings datasets demonstrate qualitatively and quantitatively that our model performs better than the state-of-the-art algorithms and can generate both realistic image content and high-frequency details. Our code will be released soon.



(a) Input

(b) Context Encoder

(c) Patch Synthesis

(d) Our Result

Figure 1. Qualitative illustration of the image inpainting task. Given an image (256×256) with a missing hole (128×128) (a), our model can generate sharper and more coherent content (d) comparing with Context Encoder[4] (b) and Image Inpainting with multi-scale neural patch synthesis [32](c).

1. Introduction

Image Inpainting is the process of recontructing the missing or masked regions of image. It is one of the most fundamental operation in image processing [9] , image editing [21, 12] , and low-level computer visions. The aim of image inpainting is to generate semantically plausible and context aware details. The generated contents can either be as accurate as the original, or maintain coherence well with the known context such that the recovered image appears to be realistic.

Most existing image inpainting methods address the hole-filling problem based on texture synthesis techniques [31, 21, 30, 2] which search for similar patches and synthesize the contents from surrounding regions. These methods recover missing regions by copying existing patterns or structures from surrounding regions. Wilczkowiak et al. [31] specify desired search regions to automatically detect better match patches. Barnes et al. [2] proposed the PatchMatch model which searches nearest neighbor patch to reconstruct missing regions. In [32], Yang et al. proposed a Multi-Scale Neural Patch Synthesis method which uses features extracted from middle layers of convolutional neural network to optimize the process of patch match, As shown in Figure.1 (c). Although these methods of searching for similar patches from surrounding regions performs well on background generation and inpainting, they fail to maintain the intrinsic structure of entire image sometimes and can not be used on images with key part missing.

To improve the inpainting results on key part missing images, a group of approaches based on data-driven have been proposed. These approaches [7, 12] assume that regions surrounded by similar scenes likely contain similar image fragment [12]. These approaches might get reasonable structure when the similar pattern or scenes can be searched from image dataset, but they could fail when the source image can not represent the missing parts very well, since both low-level and mid-level visual cues are not sufficient to synthesis semantically contents of missing regions.

Recently, the powerful capability of Deep Neural Networks (DNNs) [11, 23, 4, 24] has exhibited excellent performance in texture synthesis and image restoration. Phatak et al. [24] learn a convolutional neural network, named as Context Encoder, to directly predict missing image regions by learning the structured image. Figure.1(b) shows a result of Context Encoder on image inpainting. Compared to the detail generaion, this model focus more on structure restoration since the optimation with ℓ_2 loss [35] can not capture the underlying multi-modal distributions of images. Li et al. [18] apply a pre-trained parsing network for image completion. This additional semantic parsing network offers a combined reconstrution and parsing loss for face completion and works well. In [33], Yeh et al. try to find the closest encoding of the corrupted image in the latent images with a pre-trained generative model. Among the DNNs based image inpainting approaches, most state-of-the-art methods improve the performance of Context Encoder through capturing more **context or texture synthesis**. Although these approaches can achieve good performance by different network topology and training procedure, they have two major limitations:

- **Lack of high-frequency details:** Most existing approaches employ extra constraints (such as: fixed parsing net [18], trained generative model [33] or VGG net [32]) to enforce a generator recover more texture details. The generator based on Context Encoder [28, 24] can effectively restore image structure of the missing region, while applying this architecture directly to recovering high-frequency details is still suboptimal, as shown in Figure. 1.
- **Perceptual discontinuity:** In order to achieve better perceptual continuity, These approaches [24, 18, 33] require post-processing (e.g., poisson blending [8]) procedure, since only adopting deconvolution [23] to recover missing region cannot effectively perceive the consistency between missing region and its surroundings. Therefore, it is reasonable to establish an end-to-end model to eliminate the effect of perceptual discontinuity.

To overcome these limitations above, we design a joint optimization generative adversarial network with one generator and two discriminators for predicting the missing region, referred to as **E**nhanced **G**enerative **M**odel for **I**mage **I**npainting (EGMII). Specifically, we decompose the generator into two phase procedures to obtain more realistic content and high-frequency details. For the first phase, a convolutional encoder-decoder network constrained by a local loss function is proposed to generate the realistic content structure of missing region. For the second phase, we design a global loss function to constrain a deep residual learning network, and intend to recover high-frequency details among the whole image. Accordingly, different discriminators aim to improve the quality of synthesized results are presented for each phase. Moreover, the propsoed deep residual architecture ensures that our model can eliminate the perceptual discontinuity on the border of the missing region. Finally, the experimental results on both CelebA and Oxford Buildings datasets validate that our proposed EGMII can generate high-frequency details and eliminate the perceptual discontinuity. The main contributions of this paper are summarized as follows:

- We propose an end-to-end Enhanced Generative Model for Image Inpainting (EGMII). In this model, we design a two-phase generator which can synthesis both realistic content and high-frequency details for image inpainting.
- Different from optimizing multi-loss functions in the final output, we design different loss functions corresponding to the two-phase generator, and show that adding different constrain to intermediate result can effectively improve performance of the final output.
- We construct a new deep residual architecture for generating the high-frequency details. This architecture can refine the features progressively and eliminate the color difference to guarantee the visual consistency of entire image.

2. Related Works

In this section, we briefly review the most related works on our EGMII model, and roughly divide these works into two categories: **Structure Prediction** based models and **Texture Detail Recovering** based models.

Structure Prediction: Over the recent years, Generative Adversarial Networks(GAN) [10] have significantly advanced the image generative performance, as presented in [18, 3, 33, 1, 11, 27, 6, 5, 29, 32]. This framework trains two networks, a generator G, and a discriminator D. G aims to learn a mapping from a random vector z, sampled from a distribution p_g, to the image space x while D maps an input image to a single scalar. The purpose of G is to generate realistic images, while D represents the probability that the input image comes from the real image x rather than the image generated from G. We are motivated by the generative power of generative adversarial network and use it as the backbone of our model. Unlike the image generation tasks discussed in [3, 11, 27, 6, 5], where the input

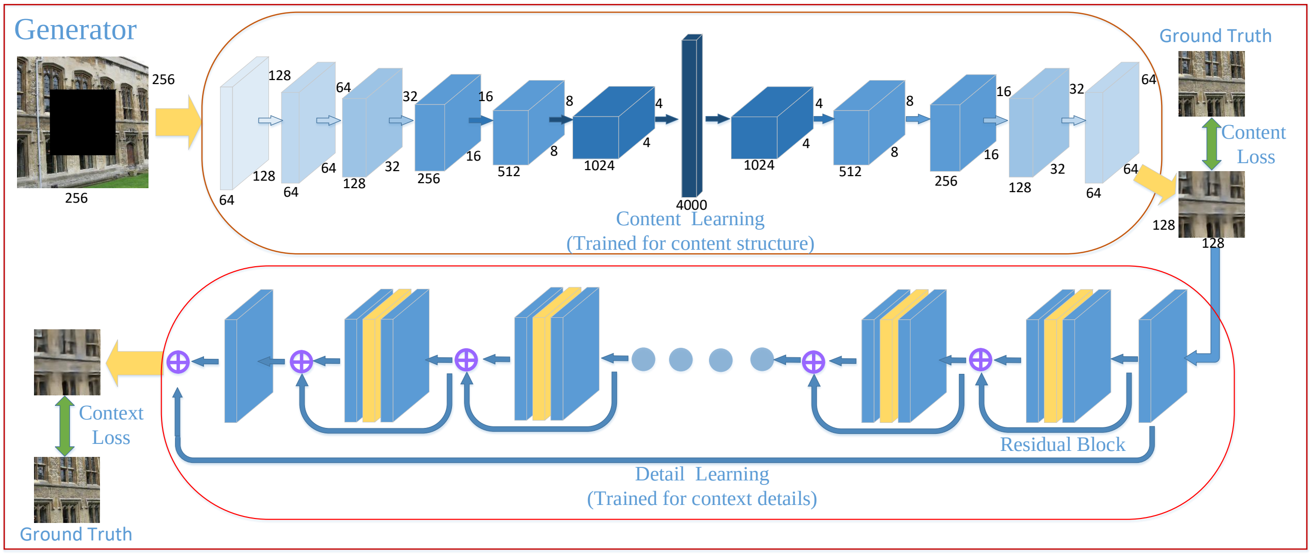


Figure 2. The illustration of our proposed generators: content learning (top) and detail learning (bottom), where the content learning phase with content loss is trained for structured information and the detail learning phase with detail loss is proposed to recover high-frequency details.

is a random noise vector and the output is an image, our goal is to predict the missing region in an image. Context Encoder [24] can train an encoder-decoder generator to directly predict the missing content. It consists of an encoder network and a decoder network. The encoder compresses the image context into a compact latent feature representation. The decoder uses this representation to reconstruct the missing content. This method verified that GAN has the ability of predicting plausible image structure. Inspired by this model, we adopt the Context Encoder architecture as the structure prediction phase of our model and fine tune the model with a local loss to improve the performance. In the following phase, we design a new network to enhance the capablity of generator to overcome the limitation of Context Encoder.

Texture Detail Recovering: In order to create more realistic image textures, our work is motivated by the recent success of Residual Network(ResNet). In [14] , He et al. first proposed ResNet with shortcut connection to solve computer vision problems such as image classification and detection. Many approaches [34, 19, 15, 17, 22] adopt ResNet to realize image restoration such as denoising, deblurring or super-resolution. As the astounding performance of deep residual learning, we design a deep residual network in our model to complete the image inpainting. In particular, we find that the deep residual learning is extremely powerful to recover fine textures or high-frequency details of natural images. In addition, we also find that deep residual architecture can refine the generated content to be consistent to the surrounding contexts.

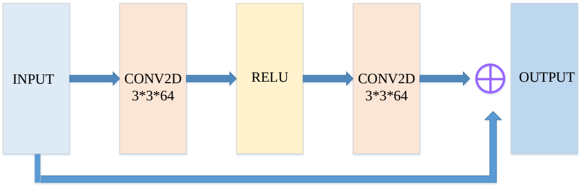


Figure 3. Residual block structure used in our network

3. Our Method

In this section, we introduce our proposed **E**nhanced **G**enerative **M**odel for **I**mage **I**npainting (EGMII). We firstly give the details of both Generator network and Discriminator network, then describe the definition of loss functions and objective function we used in learning procedure. Finally, implementation and training details are presented in Sec. 3.5.

3.1. Generator

In this section, as shown in Figure. 2, we present the generative procedure with two phases, i.e., content network and detail network. The content network as the first phase aims to reconstruct the coarse content of missing region and the detail network as the following phase aims to recover high-frequency details.

For the content network, we adopt fully convolutional encoder-decoder architecture. More specifically, our encoder consists of a set of convolutional layers, each of them is followed by a Leaky Rectified Linear Unit (LReLU) and

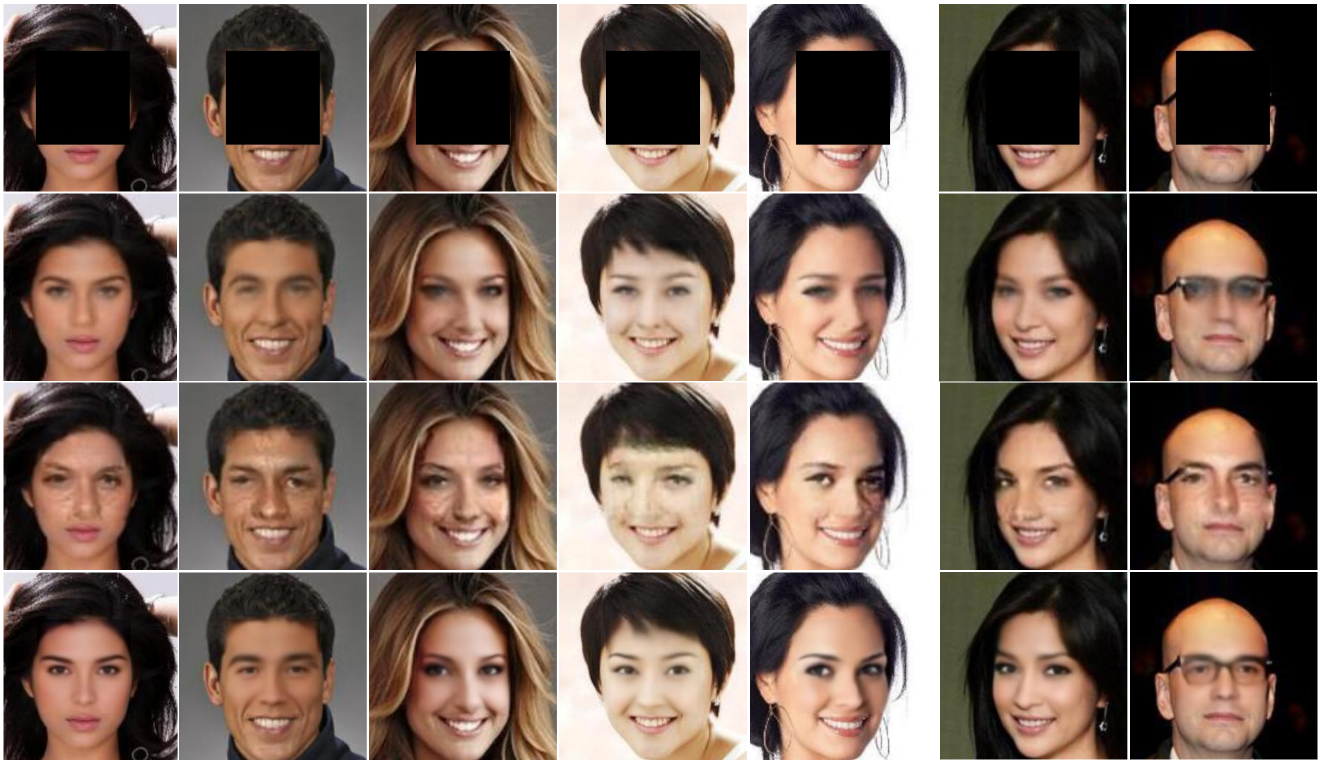


Figure 4. Visual comparisons of CelebFaces result. From top to bottom: input image, Context Encoder, Generative Face Completion, Our result. We retrain the CE model on the CelebA dataset for fair comparisons. The results of GFC offered by its author.

max pooling operation for down-sampling. At each step except the first one, we double the number of feature channels. We use a fully connected layer to map the extracted features to a hidden representation at the end of encoder. The decoder is symmetric to the encoder with deconvolutional layers.

For the detail network, we design a deep residual learning architecture to polish the result of previous phase. Compared the shallow CNN, we can design deeper architecture using residual network architecture. Also, this architecture can efficiently learn high-frequency details and remain intrinsic structure. Different from the original residual network [14] designed for object recognition, we remove the batch normalization layers of Residual Block (ResBlock) in our model. The reason is that the feature normalization damages the flexibility of the feature maps, although it can speed up the convergence in original network. To make the network have better convergence performance, we adopt skip connections in each residual blocks and the detail network. Figure. 3 shows our residual blocks.

Given a masked image, the generator firstly use encoder to map it into a latent representation, and the decoder then take the representation as input to predict the missing region. Based on the result of the content network, the detail networks can obtain sufficient receptive field to learn the contexture information via stacking enough number of convolution layers with residual blocks. This deep residual network can thus learn more high-frequency details to improve the final result. In whole process, the generator must handle three critical problems: 1) The encoder should ensure that the hidden representations can capture more semantical features and relationships between missing regions and its surroundings, which can be used for generating content information. 2) The decoder should ensure that the recovered regions have the similar structure with uncorrupted original image. 3) The detail network should recover sufficient high-frequency details and remain the pridicted structure. To achieve these purposes, we design different loss functions for two phases, as discussed in Sec. 3.3 and Sec. 3.4.

3.2. Discriminator

The generator can be used to predict sufficient image content and high-frequency details. However, it can not guarantee the visually realistic and coherent property of recovered image. To achieve more realistic results, we train two discriminators to distinguish whether its input is real or not, with each one corresponding to each phase of the generator.

As illustrated in Figure. 5, we present two discriminative networks D_l, D_g with same loss functions. For the

D_l which only provides adversarial loss for the missing region, we use the discriminator architecture of [10], which can help generate content of missing regions with sharper boundaries. For the global discriminator D_g, we adopt a set of convolutional layers to extract the high-frequency featrures F, where this layers are initialized using the same parameters as a convolutional network (VDSR [15]) pretrained on dateset BSD100. The extracted feature map F can capture more image high-frequency details information, which ensures that the generated image has the same characteristics with the real image. Therefore, taking the entire image as input in D_g can help to guarantee the generated region realistic and consistent to the surrounding contexts.

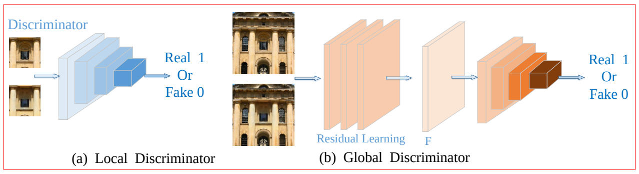


Figure 5. The overview of two discriminators: the local discriminator is presented to determine the synthesize image in the missing region as real or not, while the global discriminator focuses on determining the entire image as real or not.

3.3. The Loss Function

Let x be an input masked image, r be the ground truth of masked region and θ be the set of generator network parameters to be optimized. Our goal is to learn a generate function f for generating the missing region of image $\hat{r} = f(x; \theta)$. For each phase of our EGMII model, $\hat{r}_m$ and $\hat{r}_f$ denote the corresponding output of the content network and detail network, respectively. Instead of employing ℓ_2 loss among $(\hat{r}_m, r)$ and (r_f, r) to penalize outlier, we define the following loss functions for each phase. **Loss in pixel space:** We employ pixel loss in image space as follows:

$$\ell_{\text{pix}}(\hat{r}, r; \theta) = \frac{1}{N} \sum_{i=1}^{N} \rho(\hat{r} - r), \tag{1}$$

where $\rho(x) = \sqrt{x^2 + \varepsilon^2}$ is the Charbonnier penalty function [16] (a differentiable variant of ℓ_1), N is the total number of training samples in each batch. In addition, ε is set as $1e-3$ in this paper.

Loss in feature space: Due to the fact that reasonable objective function for generative networks can prevent it from overtraining, a loss function in the feature space is defined with the feature matching strategy. Specifically, to keep the consistency of its output and real image in feature space, we use the pre-trained VGG model to extract feature map in our model, and the new objective function is:

$$\ell_f = \frac{1}{2} \sum_{i}^{d} \left\| f_{\text{feat}}(\hat{r}_i) - f_{\text{feat}}(r_i) \right\|^2, \tag{2}$$

where $f_{\text{feat}}(r_i)$ denotes the feature extracted from a specifying intermediate layer of a pre-trained model, and d is the dimension of feature. In training stage, we firstly estimate the mean loss of the data in a minibatch in feature space, and define the feature loss function as follows:

$$\ell_{\text{feat}}(\hat{r}, r; \theta) = \frac{1}{N} \sum_{i=1}^{N} \ell_f, \tag{3}$$

where ℓ_{feat} encourages the generated data to match the real data statistically, and N is same as Eq. (1). During the training stage, the feature loss ℓ_{feat} performs better than the pixel loss in measuring structure similarity.

Adversarial Loss: In original GANs [10], the generator G and discriminator D compete in a two-player minimax game, i.e., the discriminator tries to distinguish real training data from synthesized images, and the generator tries to fool the discriminator. Concretely, the parameters φ of the discriminator are trained by minimizing:

$$\ell_D(\hat{r}, r; \varphi) = -\big[\log D_\varphi(r)\big] - \log\big(1 - D_\varphi(\hat{r})\big), \tag{4}$$

and the generator G tries to minimize :

$$\ell_G(\hat{r}) = -\log(1 - D_\varphi(\hat{r})), \tag{5}$$

where $\hat{r} = G(x; \theta)$. In our model, we define different discriminators D_l, D_g for each phase of generator, where D_l defined on masked region ensures that the predicted image has the similar structure with real image, and D_g takes the entire image as input to guarantee the consistency and visual continuity. The adversarial loss of D_l is defined as:

$$\ell_{adv_l} = \max_D \log D_\varphi(r) + \log(1 - D_\varphi(\hat{r})), \tag{6}$$

where r represents real image and $\hat{r} = G(x; \theta)$. The adversarial loss of D_g is defined as:

$$\ell_{adv_g} = \max_D \log D_\varphi(y) + \log(1 - D_\varphi(\hat{y})), \tag{7}$$

where y represents entire real image and $\hat{y}$ denotes the entire recovered image.

3.4. Objective Function

Based on the loss function defined above, we introduce two objective functions, i.e., content loss function and detail loss function to constrain our generative model.

Content Loss Function: Since the content network shares symmetric encoder-decoder architecture, and aims to generate the structure of missing regions, in this paper, we constrain this procedure via combining ℓ_2 loss, feature loss and local adversarial loss. The content loss function is defined as :

$$\ell_{\text{content}} = \ell_2 + \lambda_f \ell_{\text{feat}} + \lambda_l \ell_{adv_l}, \tag{8}$$

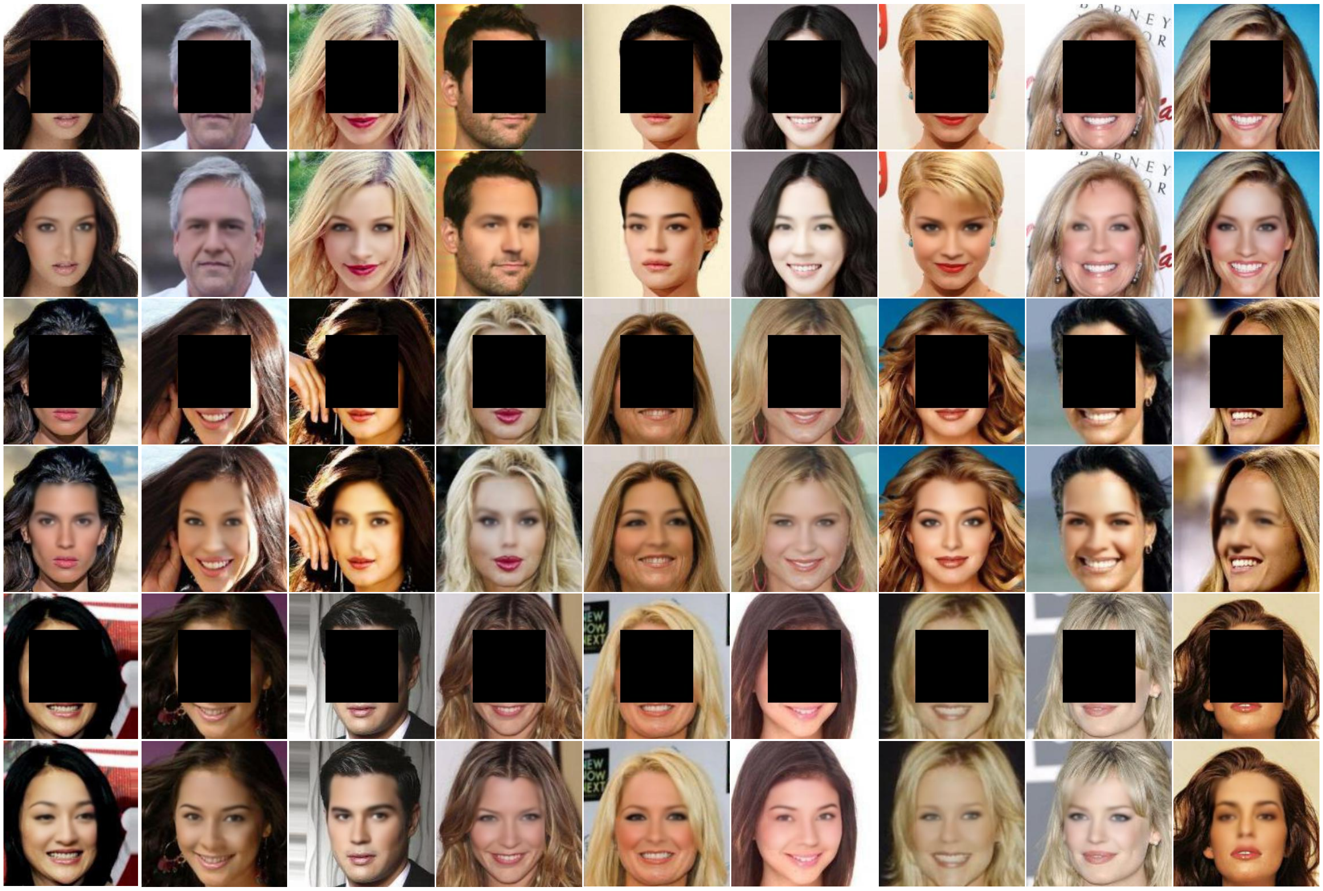


Figure 6. More results on the CelebA [20] test dataset. In each panel from top to bottom: masked inputs, the corresponding results of ours.

where λ_f and λ_l are pre-defined weights for feature loss and adversarial loss, respectively. Moreover, the advantage of using these three loss functions is that:

- The first term ℓ_2 is the most widely-used loss function for general image restoration. However, this loss penalize outliers heavily, we thus intend to smooth across various hypotheses to avoid large penalties via the following loss functions.
- The second feature loss ℓ_{feat} is good at measuring image structure similarities, and further can capture the semantics and image structure well. We thus adopt a pre-trained VGG-19 [26] network (pre-training for ImageNet classification) as the feature extracting network, and use the relu2_1 layer and the relu3_1 layer to calculate the ℓ_{feat} . This is because that empirically using a combination of relu2_1 and relu3_1 performs better than using other layers.
- The last term is the adversarial loss defined in Eq. 6, i.e., local discriminator takes the generated image or ground truth as input, and then classifies whether it is generated or not.

Detail Loss Function: The goal of the detail learning phase of generator is to learn much high-frequency details to polish the output of the first phase. To model the detail constraint effectively, we formulate the detail loss function as a combination of three terms: ℓ_{pix} loss term, global adversarial loss term and ℓ_{tv} loss term, and the loss function is defined as:

$$\ell_{\text{detail}} = \ell_{\text{pix}} + \lambda_g \ell_{\text{adv}_\text{g}} + \ell_{\text{tv}}, \tag{9}$$

where ℓ_{pix} loss is more robust loss function than ℓ_2 loss to handle outliers, and can recover more image details. Global adversarial loss $\ell_{\text{adv}_\text{g}}$ makes the generated contents to be consistent with the surrounding contexts and further enforce the generated contents to be more realistic. The last ℓ_{tv} term is used to encourage smoothness:

$$\ell_{\text{tv}} = \sum_{i,j} \left((\hat{y}_{i,j+1} - \hat{y}_{i,j})^2 + (\hat{y}_{i+1,j} - \hat{y}_{i,j})^2 \right), \tag{10}$$

Therefore, with the defined **Content Loss** and **Detail Loss**, the final objective function for the proposed EDMII can be defined as follows:

$$\ell_{\text{loss}} = \lambda \ell_{\text{content}} + (1 - \lambda)\ell_{\text{detail}}, \tag{11}$$

where λ is a pre-defined weights to banlance the content loss and detail loss.

3.5. Implementation and Training Details

In the proposed model, our generator consists of two phases, i.e. , content network and detail network, as shown in Figure. 2.

For the content network, we use an encoder-decoder architecture. Specifically, the encoder consists of a group of convolutional layers (along with batch normalization and LRelu activation) with the size of 4×4. Each layer is followed by a 2×2 max pooling operation with stride 2 for down-sampling. We double the number of feature channels at each down-sampling step. The decoder is symmetric to the encoder with deconvolutional layers.

For the detail network, it firstly transforms the output of content network to 64 feature maps. Then, n residual blocks are followed by last convolutional layers, and each convolutional layer in residual blocks consists of 64 filters with the size of 3×3. We initialize the convolutional filters using the method proposed by He et al. [13]. We pad zeros around the boundaries to preserve the size of all feature maps the same as the input of detail network. In this paper, we train the proposed model with different number of residual blocks, i.e., $n = 5, 10, 20$. Since deep networks perform better than shallow ones as the result of increasing the trained parameters, we take $n = 10$ as our baseline model with 22 convolutional layers. By optimizing the polish procedure of this phase with detail loss function (Eq. (9)), a sharper image with more contexture details is generated at the end of this stage, and the original sharp image is thus restored.

For the parameters in the training phase, we set $\lambda_f = 0.001$, $\lambda_l = 0.001$ and $\lambda_g = 0.001$ in this paper, Since setting different weight of λ to observe the different function of the content network and the detail network, we set momentum parameter to 0.9 and the weight decay to 0.0001. Notice that a smaller learning rate is critical for the convergence of the neural network, the learning rate in our experiment is initialized as 0.00001 for all layers and decreased by a factor 1 for every epoch.

4. Experiments

In this section, we present our empirical comparisons with several state-of-the-art image inpainting approaches. The experiments are conducted on CelebFaces Attribute Dataset (CelebA) [20] and Oxford Buildings Dataset [25], and some comparisons among qualitative and quantitative are presented in this section.

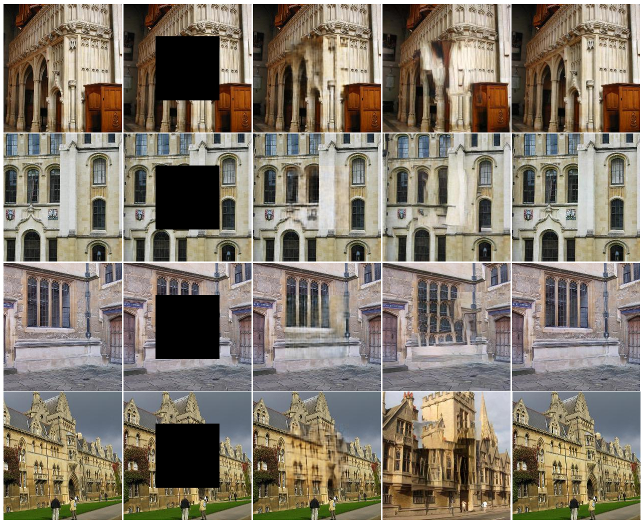
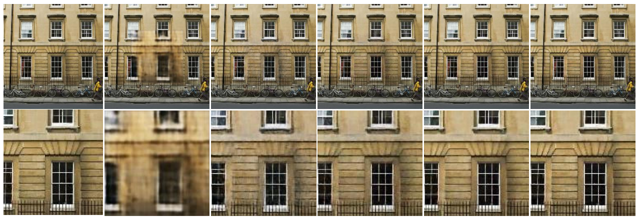


Figure 7. Visual comparisons of Oxford Buildings [25] results. From left to right: Original image,input image, Context Encoder, Neural Patch Synthesis and our result.



a) Original Image (b) λ = 1.0 (c) λ = 0.8 (d) λ = 0.6 (e) λ = 0.4 (f) λ = 0.2

Figure 8. The effect of the content network and context network. Images in (a) are ground truth images, (b) represents the results of using content network only, and (c-g) show the results of gradually decresaing the weight λ.

4.1. CelebFaces Attributes (CelebA) dataset

The CelebA dataset consists of $202,599$ face images with various view-points and expressions. For a fair comparison, we follow the standard split with $162,770$ images for training, $19,867$ for validation and $19,962$ for testing. We compare our proposed EGMII model with the state-of-the-art models, including: ContextEncoder (CE) [24] and Generative Face Completion (GFC) [18]). Since the GFC network has fully-connected layers (i.e., it needs a fixed input size of 128×128), we resize the input images to this desired size first and adjust the size of corresponding masked regions is 64×64. From the results in Figure. 4, we can notice that:

- Our model achieves the best performance among the competing models which has the same input size and masked regions, and the results also give strong support for our rationale of improving the completion performance via simultaneously considering content and texture details in the corrupted images.

	DGM	CE	GFC	Ours
SSIM	—	0.719	0.824	**0.875**
PSNR	19.4	21.3	21.51	**23.75**

Table 1. Quantitative comparison results. Higher values are better.

- More specifically, our model can well understand the difference on the individual faces from different viewpoints and discriminate reconstruct realistic eyes and eyebrows.
- To validate the generalization performance of our model, more results are presented in Figure. 6. Additionally, full results will be released soon.

4.2. Oxford Buildings Dataset

We also test our proposed model on natural images: Oxford Buildings, which contains $5,062$ building images (a training set of 3000 images and a test set of 200 images are used in this paper). Furthermore, image size in this dataset is resized to 256 $\times$ 256, and mask size is selected as 128 $\times$ 128 central square due to the fact that center area always contains the most important image information. Our model is compared with Context Encoder (ℓ_2 and adversarial loss) [24] and Neural Patch Synthesis [32]), and the results are presented in Figure. 7. As shown in Figure. 7, we have the following observations:

- Our building completion results can capture more reasonable content, clearer details, reconstruct clean lines of building and the grid pattern of windows in missing regions, which lends further evidence that our model can restore fine texture details.
- We also observe that the methods using context encoder generate results with visible reconstruction artifacts, as well as the methods using patch synthesis recover the missing region with noticeable incorrect structure. In contrast, our approache effectively suppresses such drawbacks through progressive reconstruction and joint optimization.

4.3. Quantitative comparison

We invaluate our model via two quantitative criteria: Structural Similarity Index (SSIM) and Peak Signal to Noise Ratio (PSNR). We utilize the CelebA dataset in this subsection. Specifically, SSIM is used to estimate the holistic similarity between the completion face image and the original image, and PSNR measures the difference of two images in pixel level. As the presented results in Table. 1 shows, our approach has significantly improvement than the state-of-the-art methods. On the other hand, these quantitative results show that our model achieves better perceptive on both content and details.

4.4. Effect of content and context networks

In this subsection, we intend to evaluate how the content network and context network effect the image inpainting results via varying the parameter λ in Eq. (11). Specifically, we utilize the Oxford Buildings dataset with 3000 training set and 200 test set in this subsection, and tune λ in range $[1, 0.8, 0.6, 0.4, 0.2]$, i.e., gradually decreasing the weight of content loss and increasing the detail loss. As shown in Figure. 8, we can notice that without using the content constraint, the inpainting results will be blurry and result in unpleasant visible reconstruction artifacts; when increasing the weight of the detail loss gradually, the results become more realistic-looking and more consistent with the surrounding regions. Additionally, we also find that the output of the content network is more shaper due to the constraint of detail loss function; similarly, using more detail loss may lead to incorrect image structure.

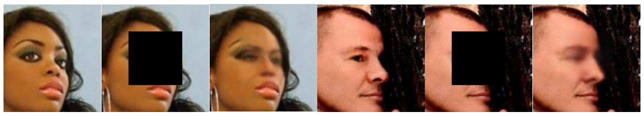


Figure 9. Failure cases of our approach. First: our model fails to generate the eye for an unaligned face. Second: our model fails to estimate correct facial pose.

4.5. Limitations

Even though our proposed EGMII model can achieve better performance on most testing images, it fails on some serious unaligned images. As shown in Figure. 9, some failure cases trained by ours are given. Notice that our model fails to reconstruct clear eyes and correct eyebrows. The reason is that the public CelebA dataset do not have sufficient profile training images similar in Figure. 9. The deep neural network thus cannot tackle the serious unaligned images which are barely appear in training data.

5. Conclusion

We have advanced the state-of-the-art results in semantic inpainting using the proposed EGMII. The comparison results show that the two-phase generator with multiple objective functions is powerful in the task of image inpainting. while the content network using content loss gives strong prior about the structure of missing regions, as well as stacking enough residual blocks also can be generate sufficient high-frequency details. This may be potentially useful to other applications such as denoising, super-resolution or other image restoring problems. There are still some cases in which our approach produces still discontinuity and artifacts when the scene or structure is complicated. In future works, we will focus on the image inpainting on more complex scene or structure.

References

[1] J. Bao, D. Chen, F. Wen, H. Li, and G. Hua. Cvae-gan: Fine-grained image generation through asymmetric training. 2017. 2

[2] C. Barnes, E. Shechtman, A. Finkelstein, and B. G. Dan. Patchmatch:a randomized correspondence algorithm for structural image editing. *Acm Transactions on Graphics*, 28(3):1–11, 2009. 1

[3] E. Denton, S. Chintala, A. Szlam, and R. Fergus. Deep generative image models using a laplacian pyramid of adversarial networks. pages 1486–1494, 2015. 2

[4] C. Doersch, A. Gupta, and A. A. Efros. *Context as Supervisory Signal: Discovering Objects with Predictable Context*. Springer International Publishing, 2014. 1, 2

[5] A. Dosovitskiy and T. Brox. Generating images with perceptual similarity metrics based on deep networks. 2016. 2

[6] A. Dosovitskiy, J. T. Springenberg, and T. Brox. Learning to generate chairs with convolutional neural networks. In *Computer Vision and Pattern Recognition*, pages 1538–1546, 2015. 2

[7] I. Drori, D. Cohen-Or, and H. Yeshurun. Fragment-based image completion. *Acm Transactions on Graphics*, 22(3):303–312, 2003. 1

[8] M. Gangnet and A. Blake. Poisson image editing. In *ACM SIGGRAPH*, pages 313–318, 2003. 2

[9] L. A. Gatys, A. S. Ecker, and M. Bethge. Texture synthesis using convolutional neural networks. In *International Conference on Neural Information Processing Systems*, pages 262–270, 2015. 1

[10] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *International Conference on Neural Information Processing Systems*, pages 2672–2680, 2014. 2, 5

[11] K. Gregor, I. Danihelka, A. Graves, D. J. Rezende, and D. Wierstra. Draw: a recurrent neural network for image generation. *Computer Science*, pages 1462–1471, 2015. 2

[12] J. Hays and A. A. Efros. Scene completion using millions of photographs. In *ACM SIGGRAPH*, page 4, 2007. 1, 2

[13] K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. pages 1026–1034, 2015. 7

[14] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Computer Vision and Pattern Recognition*, pages 770–778, 2016. 3, 4

[15] J. Kim, J. K. Lee, and K. M. Lee. Accurate image super-resolution using very deep convolutional networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1646–1654, 2015. 3, 5

[16] W. S. Lai, J. B. Huang, N. Ahuja, and M. H. Yang. Deep laplacian pyramid networks for fast and accurate super-resolution. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 5

[17] C. Ledig, L. Theis, F. Huszar, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, and Z. Wang. Photo-realistic single image super-resolution using a generative adversarial network. 2016. 3

[18] Y. Li, S. Liu, J. Yang, and M. H. Yang. Generative face completion. 2017. 2, 7

[19] B. Lim, S. Son, H. Kim, S. Nah, and K. M. Lee. Enhanced deep residual networks for single image super-resolution. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 1132–1140, 2017. 3

[20] Z. Liu, P. Luo, X. Wang, and X. Tang. Deep learning face attributes in the wild. In *IEEE International Conference on Computer Vision*, pages 3730–3738, 2015. 6, 7

[21] K. Lundbk, R. Malmros, and E. F. Mogensen. Image completion using global optimization. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 442–452, 2006. 1

[22] S. Nah, T. H. Kim, and K. M. Lee. Deep multi-scale convolutional neural network for dynamic scene deblurring. 2016. 3

[23] H. Noh, S. Hong, and B. Han. Learning deconvolution network for semantic segmentation. pages 1520–1528, 2015. 2

[24] D. Pathak, P. Krhenbhl, J. Donahue, T. Darrell, and A. A. Efros. Context encoders: Feature learning by inpainting. In *Computer Vision and Pattern Recognition*, pages 2536–2544, 2016. 2, 3, 7, 8

[25] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2007. 7

[26] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *Computer Science*, 2014. 6

[27] J. Snell, K. Ridgeway, R. Liao, B. D. Roads, M. C. Mozer, and R. S. Zemel. Learning to generate images with perceptual similarity metrics. *Computer Science*, 2015. 2

[28] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P. A. Manzagol. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *Journal of Machine Learning Research*, 11(12):3371–3408, 2010. 2

[29] X. Wang and A. Gupta. *Generative Image Modeling Using Style and Structure Adversarial Networks*. Springer International Publishing, 2016. 2

[30] Y. Wexler, E. Shechtman, and M. Irani. Space-time video completion. In *Computer Vision and Pattern Recognition*, pages I–120– I–127 Vol.1, 2004. 1

[31] M. Wilczkowiak, G. J. Brostow, B. Tordoff, and R. Cipolla. Hole filling through photomontage. In *British Machine Vision Conference 2005, Oxford, Uk, September*, 2008. 1

[32] C. Yang, X. Lu, Z. Lin, E. Shechtman, O. Wang, and H. Li. High-resolution image inpainting using multi-scale neural patch synthesis. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 1, 2, 8

[33] R. A. Yeh*, C. Chen*, T. Y. Lim, S. A. G., M. Hasegawa-Johnson, and M. N. Do. Semantic image inpainting with deep generative models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017. * equal contribution. 2

[34] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang. Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. *IEEE Transactions on Image Processing*, PP(99):1–1, 2017. 3

[35] H. Zhao, O. Gallo, I. Frosio, and J. Kautz. Is l2 a good loss function for neural networks for image processing? *Computer Science*, 2015. 2