

Inpainting of Remote Sensing SST Images With Deep Convolutional Generative Adversarial Network

Junyu Dong^{ID}, Member, IEEE, Ruiying Yin, Xin Sun^{ID}, Member, IEEE, Qiong Li, Yuting Yang^{ID}, and Xukun Qin

Abstract—Cloud occlusion is a common problem in the satellite remote sensing (RS) field and poses great challenges for image processing and object detection. Most existing methods for cloud occlusion recovery extract the surrounding information from the single corrupted image rather than the historical RS image records. Moreover, the existing algorithms can only handle small and regular-shaped obnubilation regions. This letter introduces a deep convolutional generative adversarial network to recover the RS sea surface temperature images with cloud occlusion from the big historical image records. We propose a new loss function for the inpainting network, which adds a supervision term to solve our specific problem. Given a trained generative model, we search for the closest encoding of the corrupted image in the low-dimensional space using our inpainting loss function. This encoding is then passed through the generative model to infer the missing content. We conduct experiments on the RS image data set from the national oceanic and atmospheric administration. Compared with traditional and machine learning methods, both qualitative and quantitative results show that our method has advantages over existing methods.

Index Terms—Cloud occlusion images, deep convolutional generative adversarial network (DCGAN), inpainting, sea surface temperature (SST) images.

I. INTRODUCTION

AS AN important research field, remote sensing (RS) [1] sea surface temperature (SST) image processing has been receiving more and more attention. It is not only because the development of satellite technology makes the acquirement of large RS SST data possible, but also because the SST data can provide enlightening information related to the ocean's properties. For example, SST [2] is one of the most significant factors of the marine hydrological state and also an important branch of oceanography that can provide guidance for piscatology, aquaculture, and navigation. Meanwhile, it is helpful for such fields as meteorology and hydroacoustics [3]. Furthermore, SST is the first property used for global monitoring of oceanic

Manuscript received July 24, 2018; revised September 2, 2018; accepted September 13, 2018. Date of publication October 3, 2018; date of current version January 21, 2019. This work was partially supported by the National Natural Science Foundation of China (No. 41576011, 41741007), and Key Research and Development Program of Shandong Province (No. GG201703140154). (Corresponding author: Xin Sun.)

J. Dong, R. Yin, X. Sun, Q. Li, and Y. Yang are with the Department of Computer Science and Technology, Ocean University of China, Qingdao 266100, China (e-mail: sunxin@ouc.edu.cn).

X. Qin is with the Department of Computer Science and Engineering, University of Minnesota Twin Cities, Minneapolis, MN 55455 USA.

Color versions of one or more of the figures in this letter are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/LGRS.2018.2870880

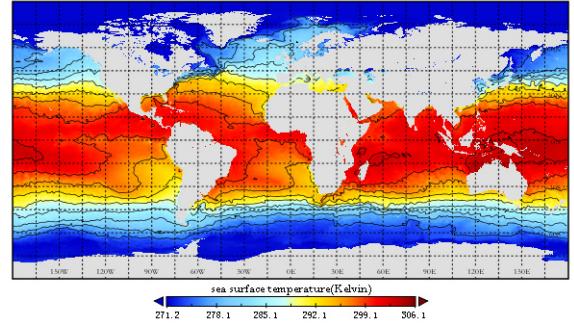


Fig. 1. Global SST sketch. In this graph, the gray regions represent land and the other regions show SST distribution in the ocean scaled to the Kelvin scale. The temperature is relevant to the color, and the higher the temperature is, the redder the color will be.

fronts from space. Fig. 1 shows a sketch of global SST, in which the gray regions represent land and the other regions show the SST distribution in the ocean. The temperature is related to the color, and the higher the temperature is, the redder the color will be. The black solid lines represent the locations where the temperature changes. These locations might be relevant for fishery. Cloud occlusion is a classic problem in the object recognition field. Nevertheless, obnubilation causes objects in SST images to be unclear and even invisible. To solve this problem, many previous methods recover the missing information using interpolation algorithms. For example, the Telea [4] algorithm is a popular method of image inpainting. The total variation (TV)-based approaches [5], [6] consider the smoothness property of images, which are useful in recovering small missing regions or removing spurious noise. Holes in textured images can be recovered by finding a similar texture from the same image. Yang *et al.* [2] propose an algorithm based on microcanonical multifractal formalism (MMF) for SST. It predicts the image with cloud occlusion using time-series images without occlusion. It is not so much an algorithm of image inpainting as an algorithm of image predicting. PatchMatch (PM) [7] searches for similar patches in the available part of the image and quickly becomes one of the most successful inpainting methods because of its efficiency. In addition, many of them utilize prior knowledge, such as statistics of patch offsets [8], planarity [9], or low rank [10]. However, these traditional methods merely work on some simple situations when the missing region is small and possibly of regular shape, because they recover corrupted images depending on the single input image. Machine learning

methods have the advantage of utilizing the global historical RS image records and have the capacity of dealing with arbitrary complex issues. Therefore, we consider solving the cloud occlusion problem with machine learning methods.

Encoder-decoder [11] is a kind of framework, which consists of two parts: an encoder and a decoder. The encoder-decoder framework looks like a funnel-shaped structure. It is also a possible way to recover the obnubilation. The deep convolutional generative adversarial network (DCGAN) [12] is a popular generative adversarial network (GAN) variant [13]–[15], which has been extensively applied to handle many different problems in fields, such as natural images, audio waveforms, and natural language; promising results have been achieved. DCGAN has the ability to learn the distribution of training data. We try to learn the distribution of an RS SST data set, and then recover information from SST images with cloud occlusion. To the best of our knowledge, this is the first work on RS SST images inpainting with DCGAN. In Section IV, we will demonstrate and explain our results.

The main contributions of this letter are summarized as follows: 1) our method can generate SST images as real as possible; 2) we introduce a new inpainting loss function to fit our specific application; 3) our method can not only perform inpainting (i.e., cloud occlusion removal) on SST images with extensive missing information, but also remove cloud occlusion regions with regular or irregular shapes; 4) our method can recover unknown-mask images compared with an encoder-decoder network [11]; and 5) we believe that the application of GAN to recover corrupted SST images will be a promising research topic.

II. NETWORK DETAILS

This section introduces our network and illustrates how it works. In Section II-A, we shortly review the traditional DCGAN architecture and propose our special loss function. In Section II-B, we introduce the SST image inpainting procedure in detail.

A. Deep Convolutional Generative Adversarial Network

As been introduced, GAN has the ability to learn the distribution of training data. However, GANs have been known to be unstable to train, often resulting in generators that produce nonsensical outputs. There have been very limited published studies in trying to understand and visualize what GANs learn, and the intermediate representations of multilayer GANs. To solve these problems, DCGAN [12] improves the architectural topology of GANs, which makes them stable to train in most settings. Compared with GAN, DCGAN modifies the network details based on the original framework structure. In consideration of all the advantages, we choose to apply the DCGAN network rather than the traditional GAN network to our SST image inpainting task. In addition, we add a supervision term to the loss function in the inpainting procedure to achieve better results.

DCGAN, as one kind of the popular variants of GAN, has the same loss function (1) as GAN. In our experiment, we optimize the generator loss function (2) and the discriminator

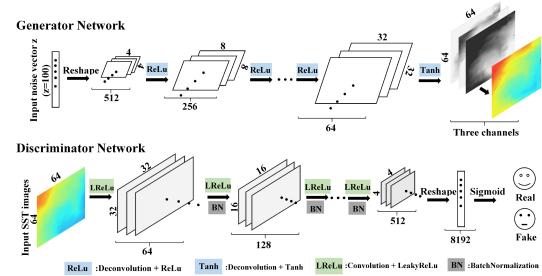


Fig. 2. DCGAN architecture for cloud occlusion inpainting. The blue module represents a deconvolution filter and the green module represents a convolution filter. ReLu, LeakyReLu, and Tanh represent three different activation functions. BN [16] represents the batch normalization operation. We optimize the DCGAN with a batch gradient descent algorithm. (Top row) Generative network including four deconvolution layers. (Second row) Discriminative network including four convolution layers. Please zoom for more details.

loss function (3) alternatively until the loss function curves converge

$$\min_G \max_D V(G, D) = E_{x \sim P_{\text{data}}(x)}[\log(D(x))] + E_{z \sim P_Z(z)}[\log(1 - D(G(z)))] \quad (1)$$

$$L_D = E_{x \sim P_{\text{data}}}[-\log(D(x))] + E_{z \sim P_Z}[-\log(1 - D(G(z)))] \quad (2)$$

$$L_G = E_{z \sim P_Z}[-\log(D(G(z)))] \quad (3)$$

G plays the role of a generator whose purpose is to generate realistic images by utilizing z vectors sampled from a uniform distribution $[-1,1]$. D plays the role of a discriminator whose purpose is to distinguish whether the image generated by G or sampled from the real data set is real or not. Just like a dual meet, two teams improve their ability to the best by rivaling with each other. Fig. 2 shows the DCGAN architecture for cloud occlusion regions recovery in detail.

B. How to Repair the Corrupted SST Images

Our inpainting method is inspired by the work of Yeh *et al.* [17], [18] that contributes our inventions for cloud occlusion recovery of SST images. However, the method proposed in [17] and [18] has the problem in that it can only recover the corrupted images with predefined obnubilation. Thus, it is inappropriate to use it to solve the cloud occlusion problem existing in SST images, because cloud occlusion is always uncertain and cannot be predefined. In contrast, our method can scan the corrupted image and automatically match a mask for the cloud occlusion area. Moreover, we propose a new loss function aimed at our specific issue. Unlike many existing traditional methods based on interpolation algorithms, our method can recover obnubilation images from the historical RS image records.

Fig. 3 shows our two-stage inpainting procedure. First, we train a DCGAN to generate SST images as real as possible. After that, we can put the discriminative part aside and consider only the generative part. Second, we train an inpainting network with trained DCGAN. Instead of recovering every pixel, we try to find the closest vector representation of the uncorrupted image in the low-dimensional space by using an input corrupted image. Suppose that we have a corrupted SST

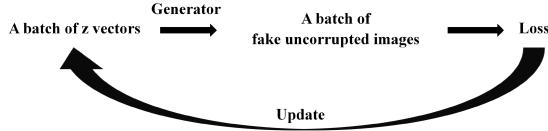


Fig. 3. Flowchart of the inpainting process. Given a batch of random z vectors sampled from a uniform distribution $[-1, 1]$, the generator can output a batch of corresponding fake images. We define an improved loss function (4) and can find the optimal z vector by updating the loss function.

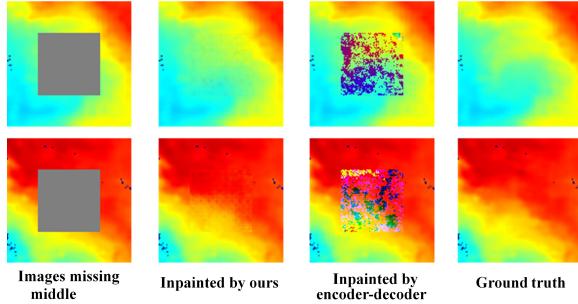


Fig. 4. Our results of recovering corrupted SST images with cloud occlusion on the center part. (First column) Corrupted SST images. (Second column) Our results. (Third column) Results inpainted by the encoder-decoder network proposed in [11]. (Last column) Groundtruth.

image, a mask-autodetection algorithm will be used to match a mask. The mask-autodetection algorithm has the ability to recognize the cloud occlusion areas. The corrupted pixels can be assigned to zero, while the uncorrupted assigned to one. The loss function (4) is as follows. The first part calculates l_1 norm between the generated sample $G(z)$ and the corrupted image x by taking advantage of the remaining available data. The second part calculates the loss, which punishes the unreal image by the discriminator D based on high-level feature representations instead of the pixelwise error as the first part does. The last part is our added supervision term whose purpose is to prohibit the situation that the result deviates from the average too much. The experimental results with and without adding it will be demonstrated in Section IV-B. The two hyperparameters are to balance the three losses, so that z is updated to fool D and generates a perceptually plausible result. We choose the right alpha and beta by judging the acceptability of the generated SST images during the first process. When we set the hyperparameters as $\alpha = \beta = 0.1$, we can get the best result. In different application scenarios, the values can be different

$$\begin{aligned} L = & \|M \odot G(z) - x_{\text{corrupted}}\| \\ & + \alpha \cdot E_{z \sim p_z}[-\log(D(G(z)))] \\ & + \beta \cdot \|(1 - M) \odot G(z) - (1 - M) \odot x_{\text{average}}\|. \quad (4) \end{aligned}$$

Here, M is the mask in which the missing part will be set to zero and the others to one. In addition, M has the same size as the input image, $G(\bullet)$ is the generator we trained, $x_{\text{corrupted}}$ is the corrupted image, $D(\bullet)$ is the discriminator, x_{average} is an average image of days of SST images centered on the corrupted SST image, and \odot denotes the multiplication of corresponding pixels.

About our mask-autodetection, there are two strategies to identify the broken parts. When the cloud occlusion situation is simple, a threshold will be set to determine where is broken

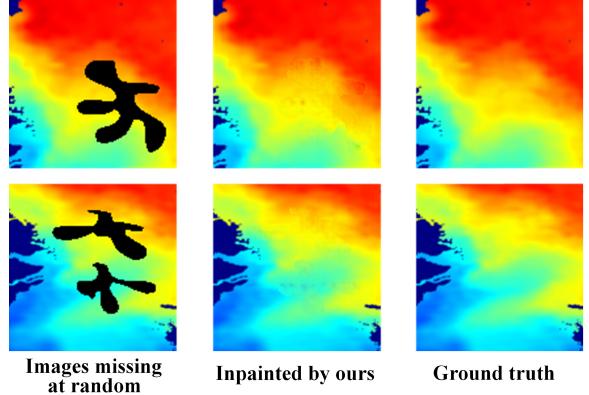


Fig. 5. Our results of recovering corrupted SST images with random cloud occlusion regions. (First column) Corrupted SST images. (Second column) Results inpainted by ours. (Third column) Groundtruth.

and where is not. Otherwise, we adopt the random-recovery strategy in which 20% of points in SST images will be randomly chosen as the broken points at each iteration. Then, our inpainting algorithm will be applied to recover the broken points. Theoretically, the probability of a point being chosen is 0.2, so that the probability of a point not being inpainted is $0.8^{100} \approx 0$ after 100 iterations. We calculate the expectation of the number of each point being taken after 100 iterations under the condition of independent and identical distribution: $E_n(X) = np$ [here, $X \sim B(n,p)$, p represents the probability of a pixel being regarded as a broken point, n represents the number of random recovery (i.e., the number of iterations), $B(\bullet)$ represents the binomial distribution, $p = 0.2$, and $n = 100$, so the $E_n(X) = 20$]. We can come to a conclusion according to probability theory that every point in an image has been recovered 20 times and we think the whole image does not have any broken points after 100 iterations.

III. IMPLEMENTATION DETAILS

We divide our algorithm into two procedures and use the mini-batch stochastic gradient descent strategy during both of them. Batch-size is set as 64. All of the parameters are initialized by a Gaussian distribution. We set the learning rate as 0.0002 and 0.01 for the two procedures, separately. During the procedure of DCGAN training, we sample a batch-size of z vectors $z \sim [-1, 1]$. And given a batch-size of corrupted images, we match a mask (M) for every corrupted image during the image inpainting procedure. The details are elaborated below as pseudocode (here, $x \sim p_{\text{real}}$, $\hat{x} \sim p_{\text{fake}}$, and $z \sim p_z$).

IV. EXPERIMENTAL RESULTS

We illustrate qualitative and quantitative results in Sections IV-A and IV-B, respectively.

A. Qualitative Results

We propose a novel method based on DCGAN to remove obnubilation regions that appear in RS SST images. Our method can handle complex situations to recover large arbitrary-shaped obnubilation regions at arbitrary locations. Fig. 4 compares our method with the encoder-decoder network

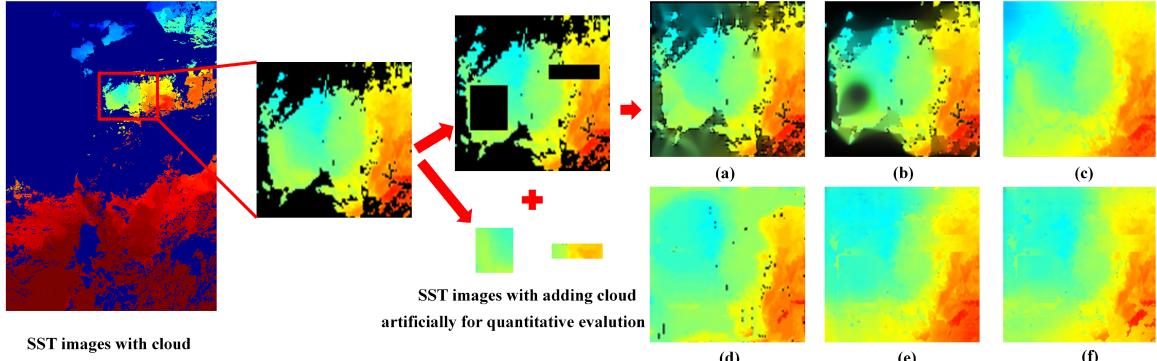


Fig. 6. Quantitative experimental results compared with traditional methods. (a) Result of Telea [4]. (b) Result of TV [5]. (c) Result of MMF [2]. (d) Result of PM [7]. (e) Result of ours without supervision term. (f) Result of ours with supervision term.

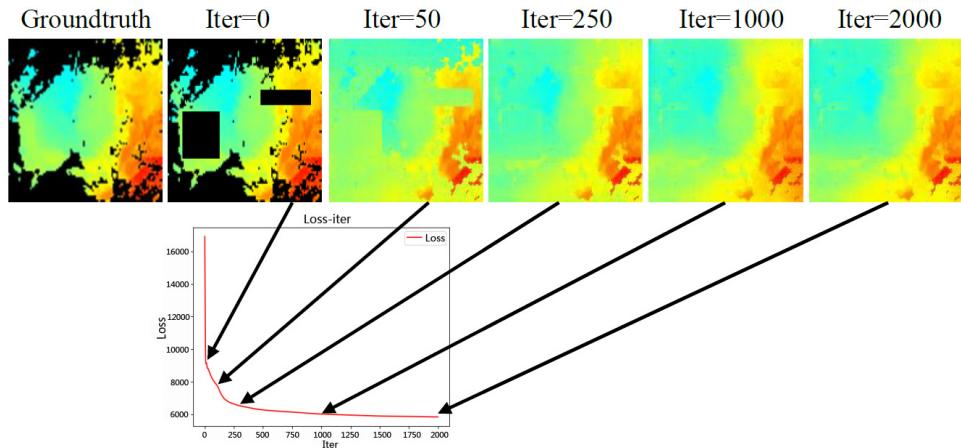


Fig. 7. Inpainting process. (First row) Changing process of an SST image. (Second row) Inpainting loss function curve.

Our Algorithm

Procedure 1: DCGAN training

While L_D loss function (2) and L_G loss function (3) not converge do:

Update L_D by stochastic gradient ascending :
 $\nabla_{\theta_d} \{-\frac{2}{batchsize} \sum [\log(D(x_i)) + \log(1 - D(G(z_i)))] - \frac{2}{batchsize} \sum [\log(D(\hat{x}_i)) + \log(1 - D(G(z_i)))]\}$
 Update L_G by stochastic gradient descending:
 $\nabla_{\theta_g} [-\frac{1}{batchsize} \sum \log(D(G(z_i)))]$

end

Procedure 2: Images Inpainting

\ find the closest z vector

While L loss function (4) not converge do:

Update z_i by stochastic gradient descending:

$\nabla_{z_i} [\|M \cdot G(z_i) - x_i_{corrupted}\|^2 + \alpha \cdot [-\log(D(G(z_i)))] + \beta \cdot \|(1-M) \odot G(z_i) - (1-M) \odot \bar{x}\|]$

end

\ get the uncorrupted SST image using pre-matched mask

$x_i_{fixed} = x_i_{corrupted} + (1-M) \odot G(z_i)$

proposed in [11] on recovering SST images. The encoder-decoder network obtains the missing pixels by regressing a per-pixel l_1 or l_2 loss function. Once an encoder-decoder network is trained using a set of aligned image pairs, it can only be applied to a single cloud occlusion situation.

For example, the situation of cloud occlusion on the center part is illustrated in Fig. 4. Because of the limited size of the RS SST data set, it cannot generate an acceptable result. Fig. 5 shows our results on recovering corrupted images with irregular cloud occlusion regions.

B. Quantitative Results

We evaluate our method by calculating a per-pixel normalized average absolute value error between the corrupted image and the recovered image, and we call it PNAR for short. The formula of PNAR e is as function

$$e = \frac{1}{m} \cdot \frac{1}{n} \sum_{i=1}^m \sum_{j=1}^n \frac{\|p_{ij}^c - p_{ij}^f\|}{255}. \quad (5)$$

Here, m is the number of SST images, n is the number of pixels in an image, p_{ij}^c represents the j th pixel in the i th corrupted image, and p_{ij}^f represents the j th pixel in the i th recovered image. Fig. 6 shows the results compared with other methods. We can see from Fig. 6 the following: the (TV) [5] leaves a black hole unsolved; the result of PM [7] produces many black broken points; the algorithm of Telea [4] cannot recover the large surrounding missing information; and the method of Yang *et al.* [2] is not so much an algorithm of inpainting as an algorithm of image predicting. Therefore, it cannot be compared with our method.

It is difficult for us to make quantitative comparisons, because we do not have the groundtruth pixel values in cloud

TABLE I
QUANTITATIVE RESULTS COMPARING WITH TRADITIONAL METHODS

methods	Telea[4]	TV[5]	Patch-match[7]	Our method(wo)	Our method(w)
error	0.0333	0.166	0.1085	0.1310	0.0862

occlusion regions. To this end, we use a small trick that adds artificial occlusion regions to the realistic cloud occlusion SST image. Then, we obtain a new SST image with the realistic cloud occlusion and the artificial occlusion. For the missing information with artificial occlusion, we can calculate the PNAR between the generated data and the groundtruth data. Fig. 6 illustrates the process. Table I presents the quantitative results. From Table I, we can see that the error of the Telea method is lower than ours. However, it is only the quantitative result for the artificial parts of cloud occlusion. From Fig. 6(a), we can see that the Telea method cannot achieve good results on the realistic cloud occlusion part. Therefore, if we consider both the realistic cloud occlusion part and the artificial occlusion part, the conclusion can be drawn that our method is the best. Fig. 7 illustrates the obnubilation removing process by the increasing iterations. The first row illustrates the changing process of an SST image. The second row is the inpainting loss function curve. We can see that the cloud occlusion part can be well inpainted by increasing iterations.

V. CONCLUSION

We introduce the DCGAN architecture to remove the cloud occlusion in SST images. To the best of our knowledge, it is the first time that GAN is used in the RS field. We propose the new loss function by adding a supervision term to solve our problem. Compared with the existing traditional and machine learning methods, we receive better results in both the qualitative and quantitative aspects. Nevertheless, our experiment still leaves much space to be improved, such as the location information we ignored and the margin of the missing part. In the future work, we will continue the research that applies the deep learning methods in the RS field and explore the possibility of training our framework in an end-to-end manner. We believe that it will be an interesting and challenging experiment.

REFERENCES

- [1] S. Shen, G. G. Leptoukh, J. G. Acker, Z. Yu, and S. J. Kempler, “Seasonal variations of chlorophyll α concentration in the northern south China sea,” *IEEE Geosci. Remote Sens. Lett.*, vol. 5, no. 2, pp. 315–319, Apr. 2008.
- [2] Y. Yang, J. Dong, X. Sun, R. Lguensat, M. Jian, and X. Wang, “Ocean front detection from instant remote sensing sst images,” *IEEE Geosci. Remote Sens. Lett.*, vol. 13, no. 12, pp. 1960–1964, Dec. 2016.
- [3] E. Lima, X. Sun, Y. Yang, and J. Dong, “Application of deep convolutional neural networks for ocean front recognition,” *J. Appl. Remote Sens.*, vol. 11, no. 4, p. 042610, 2017.
- [4] A. Telea, “An image inpainting technique based on the fast marching method,” *J. Graph. Tools*, vol. 9, no. 1, pp. 23–34, 2004.
- [5] J. Shen and T. F. Chan, “Mathematical models for local nontexture inpaintings,” *SIAM J. Appl. Math.*, vol. 62, no. 3, pp. 1019–1043, 2002.
- [6] M. V. Afonso, J. M. Bioucas-Dias, and M. A. T. Figueiredo, “An augmented Lagrangian approach to the constrained optimization formulation of imaging inverse problems,” *IEEE Trans. Image Process.*, vol. 20, no. 3, pp. 681–695, Mar. 2011.
- [7] C. Barnes, E. Shechtman, A. Finkelstein, and D. B. Goldman, “PatchMatch: A randomized correspondence algorithm for structural image editing,” *ACM Trans. Graph.*, vol. 28, no. 3, p. 24, 2009.
- [8] K. He and J. Sun, “Statistics of patch offsets for image completion,” in *Proc. Eur. Conf. Comput. Vis.*, 2012, pp. 16–29.
- [9] J. B. Huang, S. B. Kang, N. Ahuja, and J. Kopf, “Image completion using planar structure guidance,” *ACM Trans. Graph.*, vol. 33, no. 4, p. 129, 2014.
- [10] Y. Hu, D. Zhang, J. Ye, X. Li, and X. He, “Fast and accurate matrix completion via truncated nuclear norm regularization,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 9, pp. 2117–2130, Sep. 2013.
- [11] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros, “Context encoders: Feature learning by inpainting,” in *Proc. CVPR*, 2016, pp. 2536–2544.
- [12] A. Radford, L. Metz, and S. Chintala. (2015). “Unsupervised representation learning with deep convolutional generative adversarial networks.” [Online]. Available: <https://arxiv.org/abs/1511.06434>
- [13] I. J. Goodfellow *et al.*, “Generative adversarial nets,” in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2014, pp. 2672–2680.
- [14] S. Gurumurthy, R. K. Sarvadevabhatla, and R. V. Babu, “DeLiGAN: Generative adversarial networks for diverse and limited data,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 4941–4949.
- [15] X. Huang, Y. Li, O. Poursaeed, J. Hopcroft, and S. Belongie, “Stacked generative adversarial networks,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 1866–1875.
- [16] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” in *Proc. ICML*, 2015.
- [17] R. A. Yeh, C. Chen, T. Y. Lim, A. G. Schwing, M. Hasegawa-Johnson, and M. N. Do, “Semantic image inpainting with deep generative models,” in *Proc. CVPR*, 2017, pp. 6882–6890.
- [18] R. A. Yeh, C. Chen, T. Y. Lim, A. G. Schwing, M. Hasegawa-Johnson, and M. N. Do, “Semantic image inpainting with perceptual and contextual losses,” Univ. Illinois, Urbana, Urbana, IL, USA, Tech. Rep., 2016.