Generative Image Inpainting With Neural Features

Haolin Liu
Institute of Information Engineering, CAS
School of Cyber Security, UCAS

Chenyu Li
Institute of Information Engineering, CAS
School of Cyber Security, UCAS

Shiming Ge*
Institute of Information Engineering, CAS

Shengwei Zhao
Institute of Information Engineering, CAS
School of Cyber Security, UCAS

Xin Jin
Department of Computer Science and Technology, Beijing Electronic Science and Technology Institute

ABSTRACT

In this paper, we propose an image inpainting approach based on generative adversarial networks (GANs). The model consists of an inpainting network, two discriminative networks, local and global respectively, and a novel neural feature network. The inpainting network generates image content to regress the missing parts with an encoder-decoder. The two discriminative networks jointly guide the synthesized content to be consistent both locally and globally. The neural feature network, which constrains feature smoothness using feature maps of lower layers of deep neural network, serves as an effective extra regularization term for the inpainting network, ensuring the generated images to preserve structure consistence. Via extensive experiments and comparison with traditional patch matching approaches, we qualitatively and quantitatively demonstrate that our approach can make good use of the features information of images, and perform efficient and realistic inpainting.

CCS CONCEPTS

• **Computing methodologies** → **Reconstruction**; *Appearance and texture representations*; *Learning latent representations*; Unsupervised learning;

KEYWORDS

Image Inpainting, Neural Features, Generative Adversarial Networks

ACM Reference Format:
Haolin Liu, Chenyu Li, Shiming Ge, Shengwei Zhao, and Xin Jin. 2018. Generative Image Inpainting With Neural Features. In *The 10th International Conference on Internet Multimedia Computing and Service(ICIMCS'18), August 17-19,2018, Nanjing, China.* ACM, Nanjing, China, 5 pages. https://doi.org/10.1145/3240876.3240902

1 INTRODUCTION

Image inpainting algorithms have a long history of development. Approaches based on partial differential equations [1, 3] are proposed earlier. It works well for small patches. Another generally studied and applied method is based on image texture synthesis [5–9, 19, 20, 22, 23, 31, 35] method. This method has achieved significant experimental results, especially in the background completion tasks, such as sky. But it fails to utilize the features of the entire image. For some circumstances, such as images of faces or some exquisite objects, the completion results can not make global structural, semantic and visually consistent.

Recently, the field of image inpainting has experienced great process thanks to convolutional neural networks (C-NNs) [24] and generative adversarial networks (GANs) [11]. These approaches could model the context information by learning from labeled observations just like some active learning methods [38, 39]. Deepak [28] et al proposed context encoder, which can generate semantically consistent completion results based on GANs [11]. It uses the auto-encoder network to generate image and allows the extraction and utilization of high-level semantic feature, making the generated content consistent with the original image. This method can apply for complex image inpainting tasks, such as human faces and complex scenes.

However, existing model show deficiency when size of input is relatively small. Consider the circumstance where a 32×32 pixel block convolves with a 3×3 convolution kernel, which can only affect the features of a pixel. It obviously causes the loss of contextual feature information due to distance. The important feature information is missing in the convolution process, and the features of interest are not selected. As a result, the completed images fail to keep consistency around the boundary, content structure distorted, and the texture not clear enough. Therefore, we propose a modified generative completion model, which add a pre-trained network based on neural features.

The algorithm we propose can be mainly divided into two parts. The first part was used to initially generate image, based on the Encoder-Decoder model. The encoder map the

[*] Corresponding author: Shiming Ge (email: geshiming@iie.ac.cn).

ICIMCS'18, August 17-19,2018, Nanjing, China
© 2018 ACM.
ACM ISBN 978-1-4503-6520-8/18/08. . . $15.00
https://doi.org/10.1145/3240876.3240902

masked image into a feature representation, which is used by the decoder to reconstruct images. With the Encoder-Decoder as the generator, we adopt two discriminators, local discriminator and global discriminator, to apply for the GANs [11] framework. The local discriminator encourages realistic and semantically consistent details in the masked region, while the global discriminator enforces the generated content to be consistent with the original image.

The second part is the features network. In this part, we adopt a pre-trained VGG-19 [32] network to extract the features of the image. We compare the feature maps of the original image with those of the generated image. The result was used to construct the loss function, which serves as regularization to the texture and structure of the synthesised content. The generate model is optimized in a gradual way and achieves satisfying performances in small image inpainting tasks. The generated images are visually and semantically consistent.

Our proposed method contributes as follows:

- We propose a neural features network, which makes the model learn texture and semantic features and promotes image inpainting performance.
- We propose a joint inpainting framework, which completes the masked image through a global content constraint, local content constrain, and texture features constraint.
- The two discriminator enhances generated content of the image.

2 RELATED WORK

Image inpainting algorithms can be divided into two major categories. The first is traditional image inpainting algorithms, including image inpainting based on image texture synthesis [5–9, 19, 20, 22, 23, 31, 35], image inpainting based on partial differential equations [1, 3] and other representative methods [12–14, 21, 29]. The second is an image inpainting method based on convolutional neural networks, which mainly uses self-learning methods [18, 36, 37] for image inpainting.

The concept of image inpainting was first proposed by Bertalmio [3]. Meanwhile, image inpainting algorithms based on partial differential equations [1, 3] emerged. The method complete images with small areas missing. The patch-based method was first used for image texture synthesis [8]. This method synthesizes the target image by sampling the original image, and extended to image stitching techniques [23]. The patchmatch [2] method can achieve real-time image processing based on these methods. However, these methods cannot fill holes with complex structures.

Recently, deep learning and GAN-based image inpainting methods [18, 27] have shown promising prospect. Context Encoder [28] is the first to apply GANs [11], which utilized an encoding and decoding network. It introduced adversaries loss, combined with L2 loss as the objective function. It exhibit good inpainting ability. Soon, targeted at the incompatibility of CE in keeping global consistency, Ishikawa [15] et al introduced two discriminator. With two discriminators judge local and global fidelity separately, the model can generate more details, while maintaining the consistency of the whole image. In face images completion, Li [26] proposed similar method. The model leverages two discriminators, dealing with local and global discrimination respectively, finally adding a face parsing network to form the generative face completion model. It can generated realistic face images, which are consistent both in pixel and semantic. But these methods often require image fusion operations on the boundaries of the mask.

3 APPROACH

As shown in Fig.1, the model consists of one generator, two discriminators and a neural feature network. The generator completes an image with an encoder-decoder, while the two discriminators determine whether the generated image is true or false in global and local respectively. In addition, the novel neural feature network is used to further improve the generated image in structure and texture. We describe these networks in detail as follows.

3.1 Generator

The generator adopts the encoder-decoder structure. We use the conv1 to conv3 layers of the VGG-19 [32] network and add a full connection layer as the encoder. The decoder adopts symmetrical structure, correspondingly using the deconv and unpooling layers. The network structure settings of the generator are shown in the Table 1 and Table 2.

Table 1: The structure of the encoder in generator

Type	conv.	pool.	conv.	pool.	conv.	pool.	conv.	pool.	fc.	defc.
Kernel	3×3	2×2	3×3	2×2	3×3	2×2	3×3	2×2	-	-
Stride	1×1	2×2	1×1	2×2	1×1	2×2	1×1	2×2	-	-
Outputs	32	-	64	-	128	-	256	-	1024	4096

Table 2: The structure of the decoder in generator

Type	Kernel	Stride	Scale	Outputs
reshape.	-	-	-	256
unsample.	-	-	2	-
conv.	3×3	1×1	-	64
unsample.	-	-	2	-
conv.	3×3	1×1	-	64
unsample.	-	-	2	-
conv.	3×3	1×1	-	32
unsample.	-	-	2	-
conv.	3×3	1×1	-	32

3.2 Discriminator

The generator can generate preliminary low-level content, but only with a fuzzy shape, often not realistic enough. In order to generate realistic images, we introduced the GANs [11]

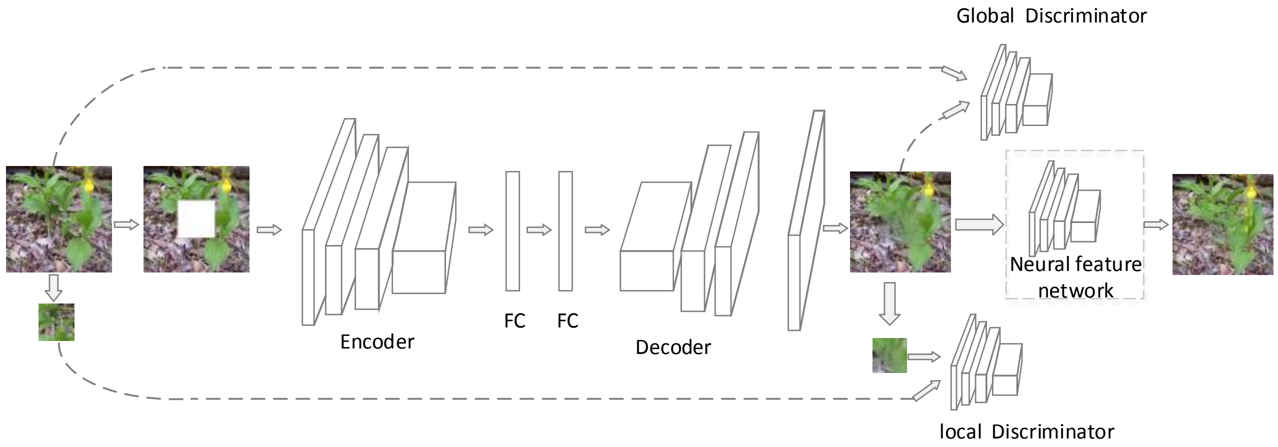

Figure 1: The network architecture. It consists of one generator, two discriminators and a neural feature network. The generator generates a preliminary image. The global discriminator determine whether the whole generated image is true or false. The local discriminator determine whether the local generated image is true or false. The neural feature network, which is a pretrained model and remains fixed. It is used to further generate a clearer image of the texture.

Table 3: The structure of global discriminator

Type	conv.	conv.	conv.	conv.	conv.	conv.	conv.	sigmoid.
Kernel	3×3	4×4	4×4	4×4	4×4	4×4	4×4	-
Stride	1×1	2×2	2×2	2×2	2×2	2×2	1×1	-
Outputs	32	32	64	128	256	128	1	1

Table 4: The structure of local discriminator

Type	conv.	conv.	conv.	conv.	conv.	sigmoid.
Kernel	3×3	4×4	4×4	4×4	4×4	-
Stride	1×1	2×2	2×2	2×2	1×1	-
Outputs	64	64	128	256	1	1

framework with some changes. We adopt two discriminators, a global discriminator and a local discriminator. The local discriminator is used to discriminate whether the synthesized mask region is real or fake. It encourages the generator to generate the details of the missing region. However, its limitations are also obvious. It can only discriminate local pixels. The new pixels it generates are limited to the pixels in the mask region. Moreover, the local discriminator can hardly affect pixels outside the mask region. This leads to inconsistency between the generated content with the rest of the image. Therefore, we propose another global discriminator, which takes the whole image as input and makes judgement. Global discriminator will not only make the missing image content realistic, but also enforce the content of the whole image consistent. The network structure settings of the discriminator are shown in the Table 3 and Table 4.

3.3 Neural Features

However, with above modules, the texture of the generated content still not clear enough. A recent study by C. Li and M. Wand [25] found that the features extracted from different convolution layers of image have different meanings [4, 10, 17, 33, 34]. In the middle convolution layer, the neural features are similar to the contents of image. But in the lower layers, The neural features of convolutional layer tend to represent the styles of image. Inspired by the above ideas, we utilized a pre-trained VGG-19 [32] classification model and introduced conv3 neural feature constraint, by extracting the conv3 feature maps of the original image and the generated image respectively. As experiments show, it can make the synthesized texture in the masked region clearer. Since the conv3 features relate to pixels not only within, but around the mask region, it further enforce the synthesized texture consistent with the surrounding.

3.4 Objective Function

First, we construct an L_2 loss function, which is the L_2 distance between the original image and the generated image. We also adopt adversarial loss, which is defined as:

$$\begin{aligned}\min_G \max_D V(G,D) =& E_{x\sim p_{data}(x)}[log(D(X))] \\ &+ E_{z\sim p_z(Z)}[log(1-(D(X)))]\end{aligned} \tag{1}$$

where $p_{data}(x)$ is the distribution of real variables, and $p_z(Z)$ is the distribution of noise variables. The adversarial loss aims to make the generator generate photorealistic image to deceive the discriminators, while the discriminator makes the best efforts to tell them apart. Corresponding to the two discriminators, we also introduced two adversarial loss, which were defined as L_{b1} and L_{b2} respectively. Moreover, to regularize the generated content in feature level, we also adopt features style loss, defined as follows:

$$D_S^L(X,Y) = \sum_{k,l}(F_{XL}(k,l) - F_{YL}(k,l))^2 \tag{2}$$

In the above formula, X is a real image and Y is a generated image. The $F_{XL}(i)$ is the ith element of the features map exported from the Lth layer of the network. We refer to the neural features loss as L_n and the total objective function is formulated as follows:

$$L = L_2 + \lambda_1 L_{b1} + \lambda_2 L_{b2} + \lambda_3 L_n \tag{3}$$

which λ_1, λ_2, λ_3 are coefficients for balancing the effects of different losses.

3.5 Training

Our hyperparameter settings in the training process are similar to those in DCGAN [30]. In the generative network, we used Batch Normalization [16] and each convolutional layer is followed by a ReLU activation layer. But the discrimination network did not use Batch Normalization.

4 EXPERIMENT

We carried out comprehensive experiments to evaluate the inpainting ability of our model. We set λ_1=200, λ_2 =200, λ_3 =0.05 during the training process.

4.1 Dataset

We used the ImageNet dataset, which covers more than 1000 objects. We used 20112 images from ImageNet as the training set, and 5500 images from standard test images as the test set. We cropped all the training images and scaled them to 24×24 pixels. During the training, the size of mask is fixed to be 8×8 pixels, with position fixed to the center. During the test, there is no restriction to the size and location of the mask. In order to avoid over-fitting, we performed data augmentation, including flip, shift, and rotation.

4.2 Image Inpainting

Visual Effects: During the testing process, the size and location of mask areas of each image are randomly selected. A batch of inpaiting results are shown in the figure 4. It is shown that the generated images are visually pleasant. We also modify the location and the size of the mask to evaluate the model performance. The results are shown in the figure 2 and figure 3. It is shown that under different size and location of mask, our model is always capable of realistic inpainting.

Table 5: Quantitative Results

Method	PSNR	SSIM
PatchMatch	17.9027	0.8997
Ours	**24.3315**	**0.9599**

Quantitative Results: we also run quantitative assessments. We performed an evaluation on the standard dataset and evaluated the results using the PSNR (peak-to-noise ratio) and SSIM (structural similarity index). The results are shown in Table 5 and show our method are better than PM.

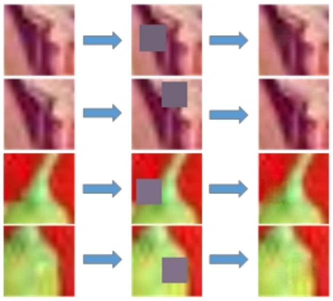

Figure 2: In each row from left to right: the original image. mask image. and generated image. In the second column, different row have different mask areas.

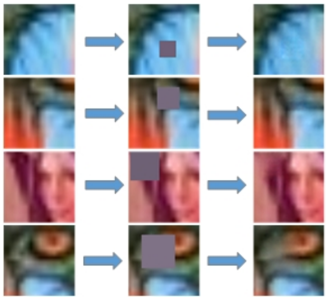

Figure 3: In each row from left to right: the original image. mask image. and generated image. In the second column, different row have different mask size.

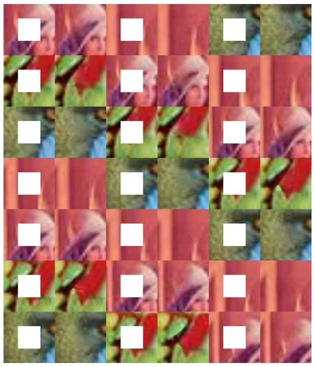

Figure 4: Batch results

5 CONCLUSION

In this work, we proposed an image inpainting approach based on GANs. The proposed model consists of a generator, two discriminators and a novel neural feature network, which jointly guide the inpainted images to be consistent both in content and structure. Both qualitative and quantitative results show that the proposed approach can generate realistic images, where the synthesised content is consistent with existing image, both in pixel and semantic level, even with large area missing. In the future, we will extend the neural feature network to other applications.

ACKNOWLEDGMENTS

This work was partially supported by grants from National Key Research and Development Plan (2016YFC0801005), National Natural Science Foundation of China (61772513), and the International Cooperation Project of Institute of Information Engineering at Chinese Academy of Sciences (Y7Z0511101). Shiming Ge is also supported by Youth Innovation Promotion Association, CAS.

REFERENCES

[1] Coloma Ballester, M Bertalmio, V Caselles, Guillermo Sapiro, and Joan Verdera. 2001. Filling-in by joint interpolation of vector fields and gray levels. *IEEE Transactions on Image Processing* 10, 8 (2001), 1200–1211.

[2] Connelly Barnes, Eli Shechtman, Adam Finkelstein, and Dan B. Goldman. 2009. PatchMatch: a randomized correspondence algorithm for structural image editing. *ACM Trans. Graph.* 28, 3 (2009), 24:1–24:11.

[3] Marcelo Bertalmio, Guillermo Sapiro, Vincent Caselles, and Coloma Ballester. 2000. Image Inpainting. In *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*. 417–424.

[4] Alex J. Champandard. 2016. Semantic Style Transfer and Turning Two-Bit Doodles into Fine Artworks. *CoRR* abs/1603.01768 (2016).

[5] Antonio Criminisi, Patrick Pérez, and Kentaro Toyama. 2003. Object Removal by Exemplar-Based Inpainting. In *IEEE Computer Vision and Pattern Recognition*. 721–728.

[6] Soheil Darabi, Eli Shechtman, Connelly Barnes, Dan B. Goldman, and Pradeep Sen. 2012. Image melding: combining inconsistent images using patch-based synthesis. *ACM Trans. Graph.* 31, 4 (2012), 82:1–82:10.

[7] Iddo Drori, Daniel Cohen-Or, and Hezy Yeshurun. 2003. Fragment-based image completion. *ACM Trans. Graph.* 22, 3 (2003), 303–312.

[8] Alexei A. Efros and William T. Freeman. 2001. Image quilting for texture synthesis and transfer. In *Conference on Computer Graphics and Interactive Techniques*. 341–346.

[9] Alexei A Efros and Thomas K Leung. 1999. Texture synthesis by non-parametric sampling. In *IEEE International Conference on Computer Vision*, Vol. 2. 1033–1038.

[10] Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. 2015. Texture synthesis and the controlled generation of natural stimuli using convolutional neural networks. *CoRR* abs/1505.07376 (2015).

[11] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative Adversarial Nets. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, Vol. 3. 2672–2680.

[12] James Hays and Alexei A. Efros. 2007. Scene completion using millions of photographs. *ACM Transactions on Graphics* 26, 3 (2007), 4.

[13] Kaiming He and Jian Sun. 2012. Statistics of patch offsets for image completion. In *European Conference on Computer Vision*. 16–29.

[14] Jia Bin Huang, Sing Bing Kang, Narendra Ahuja, and Johannes Kopf. 2014. Image completion using planar structure guidance. *Acm Transactions on Graphics* 33, 4 (2014), 1–10.

[15] Satoshi Iizuka, Edgar Simo-Serra, and Hiroshi Ishikawa. 2017. Globally and Locally Consistent Image Completion. *ACM Trans. Graph.* 36, 4 (July 2017), 107:1–107:14.

[16] Sergey Ioffe and Christian Szegedy. 2015. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. *CoRR* abs/1502.03167 (2015).

[17] Justin Johnson, Alexandre Alahi, and Fei-Fei Li. 2016. Perceptual Losses for Real-Time Style Transfer and Super-Resolution. In *European Conference on Computer Vision*. 694–711.

[18] Rolf Köhler, Christian Schuler, Bernhard Schölkopf, and Stefan Harmeling. 2014. Mask-Specific Inpainting with Deep Neural Networks. In *German Conference on Pattern Recognition*, Vol. 8753. 523–534.

[19] Nikos Komodakis. 2006. Image Completion Using Global Optimization. In *IEEE Computer Vision and Pattern Recognition*, Vol. 1. 442–452.

[20] N Komodakis and G Tziritas. 2007. Image Completion Using Efficient Belief Propagation Via Priority Scheduling and Dynamic Pruning. *IEEE Transactions on Image Processing* 16, 11 (2007), 2649–2661.

[21] Johannes Kopf, Wolf Kienzle, Steven Drucker, and Sing Bing Kang. 2012. Quality prediction for image completion. *Acm Transactions on Graphics* 31, 6 (2012), 131.

[22] Vivek Kwatra, Irfan Essa, Aaron Bobick, and Nipun Kwatra. 2005. Texture optimization for example-based synthesis. *ACM Trans. Graph.* 24, 3 (2005), 795–802.

[23] Vivek Kwatra, Arno Schödl, Irfan A. Essa, Greg Turk, and Aaron F. Bobick. 2003. Graphcut textures: image and video synthesis using graph cuts. *ACM Trans. Graph.* 22, 3 (2003), 277–286.

[24] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. 1998. Gradient-based learning applied to document recognition. *Proc. IEEE* 86, 11 (1998), 2278–2324.

[25] Chuan Li and Michael Wand. 2016. Combining Markov Random Fields and Convolutional Neural Networks for Image Synthesis. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2479–2486.

[26] Y. Li, S. Liu, J. Yang, and M. H. Yang. 2017. Generative Face Completion. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 5892–5900.

[27] Xudong Mao, Qing Li, Haoran Xie, Raymond Y. K. Lau, Zhen Wang, and Stephen Paul Smolley. 2017. Least Squares Generative Adversarial Networks. In *IEEE International Conference on Computer Vision*. 2813–2821.

[28] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A. Efros. 2016. Context Encoders: Feature Learning by Inpainting. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[29] Darko Pavi, Volker Schnefeld, and Leif Kobbelt. 2006. Interactive image completion with perspective correction. *Visual Computer* 22, 9-11 (2006), 671–681.

[30] Alec Radford, Luke Metz, and Soumith Chintala. 2015. Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. *CoRR* abs/1511.06434 (2015).

[31] Ganesh Ramanarayanan and Kavita Bala. 2007. Constrained Texture Synthesis via Energy Minimization. *IEEE Trans Vis Comput Graph* 13, 1 (2007), 167–178.

[32] Karen Simonyan and Andrew Zisserman. 2014. Very Deep Convolutional Networks for Large-Scale Image Recognition. *CoRR* abs/1409.1556 (2014).

[33] Xavier Snelgrove. 2017. High-resolution multi-scale neural texture synthesis. In *SIGGRAPH Asia 2017 Technical Briefs*. 1–4.

[34] Dmitry Ulyanov, Vadim Lebedev, Andrea, and Victor Lempitsky. 2016. Texture Networks: Feed-forward Synthesis of Textures and Stylized Images. In *Proceedings of The 33rd International Conference on Machine Learning*, Vol. 48. 1349–1357.

[35] Marta Wilczkowiak, Gabriel J. Brostow, Ben Tordoff, and Roberto Cipolla. 2005. Hole Filling Through Photomontage. In *Proceedings of the British Machine Vision Conference*.

[36] Li Xu, Jimmy S. J. Ren, Ce Liu, and Jiaya Jia. 2014. Deep Convolutional Neural Network for Image Deconvolution. In *International Conference on Neural Information Processing Systems*, Vol. 1. 1790–1798.

[37] Raymond Yeh, Chen Chen, Teck Yian Lim, Mark Hasegawa-johnson, and Minh N. Do. 2016. Semantic Image Inpainting with Perceptual and Contextual Losses. *CoRR* abs/1607.07539 (2016).

[38] Xiaoyu Zhang, Shupeng Wang, and Xiao-chun Yun. 2015. Bidirectional active learning: A two-way exploration into unlabeled and labeled data set. *IEEE Trans. Neural Netw. Learning Syst.* 26, 12 (2015), 3034–3044.

[39] Xiaoyu Zhang, Shupeng Wang, Xiaobin Zhu, Xiao-chun Yun, Guangjun Wu, and Yipeng Wang. 2015. Update vs upgrade: modeling with indeterminate multi-class active learning. *Neurocomputing* 162 (2015), 163–170.