Evaluations of Image Completion Algorithms: Exemplar-Based Inpainting vs. Deep Convolutional GAN

Chenduo Huang
Stanford University
cdhuang@stanford.edu

Koki Yoshida
Stanford University
kokiy@stanford.edu

Abstract

Image completion, also known as image inpainting, is an active computer vision research problem that aims to automatically fill in a missing portion of an image in a content-aware way. Various approaches have been engineered by academia dedicated to this problem, modern ones even applying deep convolutional neural networks to have a smoother and more realistic output images. In this project, we aim to compare and evaluate two most widely accepted image inpainting algorithms: Exemplar-Based Inpainting (EBI)[1] and Deep Convolutional Generative Adversarial Nets (DCGAN)[5] based image inpainting algorithm.

1. Introduction

Image completion, also known as image inpainting, is an active computer vision research problem that aims to automatically fill in a missing portion of an image in a content-aware way. By content-aware, it means that an algorithm should consider the neighbor pixel information of the missing portion of the image it is completing when it produces the final completed output. State of the art such algorithms are often realized with convolutional neural networks, and in this paper, we aim to reproduce one of the novel research results of such neural network structures, Deep Convolutional GAN, apply it to construct an image inpainting algorithm, and evaluate our reproduced algorithm qualitatively by comparing it with Exemplar-Based Inpainting, and quantitatively by evaluating its output completed images' L1 distances to the original images.

The input to our algorithms is two datasets of human face images. For the GAN based algorithm, we use the dataset to train a GAN model, producing a discriminator D and a generator G. We then try to find the optimal input noise z that generates the best-fitting image for our image completion purpose. Finally, we use corresponding area of G(z) to fill in missing region(s) of the image.

There are two folds for our algorithms evaluation: to compare EBI with DCGAN based approaches, we simply compare the outputs of the algorithms by letting objective audience choose the better image of the output pair generated. To evaluate the DCGAN based approach by itself, we try to formalize it by comparing its output images with respect to the original images, computing average L1 distances between its outputs and originals.

The rest of this paper is structured as follows: section 2 discusses related work of image completion problem; section 3 elaborates on the image data set we chose to use for this project; section 4 and 5 go deep on the details of the algorithms we chose to evaluate and actual evaluation of their performances; section 6 addresses future works and section 7 concludes our project work.

2. Related Work

For our related work, we should first note that since our work spans from traditional image processing techniques[1], to modern application of convolutional neural networks in image completion context, we introduce our related work in a similar manner: we first introduce some related work in traditional image processing area, and then move on to introduce more related work on deep learning in image completion context.

2.1. Traditional Image Processing Techniques

The work by Bertalmio et al.[2] first formalizes the image inpainting problem and summarizes its application purposes. Followed by them were numerous image processing techniques for image inpainting problem. Criminisi et al.[1] introduced a milestone algorithm that formed the basis of the Exemplar-Based Inpainting algorithm used for this project, and we will elaborate on the algorithm in section 4. Sun et al.[7] proposed an image completion algorithm by emphasizing the underlying structures in the images by looking at the entire image rather than nearby pixels around filling regions. Their algorithm also differs from the algo-

[1] By traditional, we mean that the algorithm does not have any machine learning component.

rithm by Criminisi et al. in ways they define and compute the structure of images: they require users to manually input important structure information, and the algorithm works by computing the pixel structures following the input structures from users. Later Telea[21] proposed an image inpainting algorithm that runs faster but maintains competitive quality of output images to the work by Criminisi et al. This is achieved by essentially looking at an even smaller region of neighboring pixels compared to EBI. All the previous works compute the pixels in masked region by considering each pixel individually. However, rather than computing the next pixel in the masked region following similar manner, Avidan et al.[8] engineered an image inpainting algorithm which uses a technique called seam carving. Seam carving is a technique that extracts image content information from an optimal 8-connected path of pixels from top to bottom, or left to right, and Yan et al.[9] later engineered an algorithm which uses seam carving in a more efficient way: only considering seams in the pixel regions outside the target filling regions.

All of the above image inpainting algorithms are based on traditional image processing techniques with no machine learning involved, and we chose to focus on the algorithm by Criminisi et al. since it was the first milestone research work that performed reasonably well with various input images.

2.2. Image Completion with Deep Learning

Our project builds upon GAN network concept first introduced by Goodfellow et al.[10], which defines a generator neural network and discriminator neural network. Generator transforms random input noise, noted as z, into a fake image that is aimed to look like a random one from training set, and then discriminator takes in this generated image, and tries to distinguish if this image is a generated fake image or a real image from training set. The original GAN structure was defined with fully connected layers, and the work by Radford et al.[5] extended this GAN concept into DCGAN by replacing fully connected layers with convolutional layers, making generator and discriminator convolutional neural networks instead. However, being notoriously finicky with hyperparameters, previous networks engineered such as normal GAN and DCGAN requires huge commitment during its training process. State of the art networks such as WGAN by Arjovsky et al.[6] improves the stability of the training process by avoiding problems such as mode collapse, making hyperparameter searches on GANs easier and more stable than before.

Above related works are different GAN structures proposed that formed the basis of our research project, and the followings discuss more on the intersection between deep learning and image completion algorithms. Work by Fawzi et al.[11] proposes an approach to transform a pre-trained neural network on images into a tool to patch large corrupted regions of pixels. Schuler et al.[12] proposes a method to learn direct mapping from masked regions to their completed counterparts. Last but not least, Yeh et al.[13] proposed a method to use GANs to complete masked regions of images, and their method of image completion is elaborated in our method section as we formed our GAN based approach image completion algorithm following this piece of research work.

The work by Schuler et al. requires less test time for performing image completion task, and we first considered forming our project based on this piece of research work. However, despite its fast runtime, its algorithm requires prior knowledge of the masking information such as mask size and mask location before the training phase, which is not very scalable to be used in comparison with traditional image processing techniques. However, the work by Yeh et al. perfectly solves this problem by only considering the masking information in the test time, thus making the trained networks more generalizable as traditional image processing techniques are.

2.3. Evaluations of Different Image Inpainting Algorithms

Patel et al.[14] previously have published a work that reviews and evaluates different image completion algorithms, and their reviewed algorithms are all traditional image processing techniques that require no machine learning involved. Sangeetha et al.[15] also published a piece of work reviewing different image completion algorithms, but this work only looks at those algorithms that used exemplar-based inpainting. Tiefenbacher et al.[16] presented a relatively new piece of work that evaluates state-of-the-art image completion algorithms in 2015, and gives a conclusion that some of the formal metrics defined for evaluating image quality are not suited for evaluating quality of image completion algorithms.

Our project work follows a similar suit of the above pieces of works, but we choose to do an evaluation between EBI and DCGAN based approach to represent a comparison between traditional image processing techniques and modern deep learning based approaches.

3. Datasets

For our DCGAN based image inpainting algorithm, we used CelebA[17] dataset and LFW[18] dataset which in total comprises roughly 215,599 images of human faces collected from the web. The CelebA dataset is mainly used for training, which contains in total of 202,599 human face images, and we used LFW dataset for validation and testing, which contains roughly 13,000 images. Although these two datasets are completely independent, after the following preprocessing steps, we are able to essentially use them

(a) Sample images from CelebA

(b) Sample images from LFW

Figure 1: Sample images of our datasets

as an aggregated one big dataset.

Before we feed our raw images of human faces into our model for training, we first preprocess it using openface library[19] to (1): crop the images down to 64 pixels * 64 pixels in size, and (2): keep only the human face part and remove all background pixels. From then we further preprocess the input images by dividing all pixel values with 127.5 (a convenient mean value chosen for pixel values ranging from 0 to 255), and then subtract by one. This above step is introduced to map all pixel values from [0, 255] to [-1, 1], enhancing the stability of DCGAN training process.

For EBI algorithm, since there is no machine learning involved, there was no specific dataset used for engineering the algorithm. We used several images we had in hand and some found on the web for the algorithm's sample outputs. However, for comparison purposes, we also included output of EBI algorithm on some sample images from the above LFW dataset in section 5.

Some sample images we used from the above two datasets and for EBI algorithm are presented in Fig. 1.

4. Methods

In this section, we discuss the methodology and implementation for the the algorithms we are evaluating. As for EBI, we chose to use the online open source implementation of the algorithm by Li et al.[20], and for DCGAN based image completion algorithm, we modified based on the existing MIT Licensed implementation of DCGAN based image completion algorithm by Amos[22]. On top of the code modified, we performed our original reproduction and evaluation of the two algorithms.

4.1. Exemplar-Based Inpainting (EBI)

As suggested by its name, exemplar-based inpainting uses an iterative solution to generate the unknown region based on the source region of the image. The pixel synthesis of the fill-region begins at the fill-front, or the edge between the known/unknown region, and gradually moves inwards to complete the missing zone. The intuition of this algorithm came from fluid dynamics and partial differential equations. The filling mechanism is derived from Dr. Bertalmo, Bertozzi and Sapiros concept of smooth continuation of information in the level-lines direction.[3] According to Bertalmo's 2001 CVPR paper, the filling rule is to extend the isophotes, or linear structures, while matching gradient vectors at the contiguous edge of the fill-region.[4]

The EBI algorithm consists of a Laplacian-based edge detection, followed by iterations of two major filling steps: determining pixel filling priority, and calculating the weighted pixel value. There are variations to the implementation of this method. While traversing along the fill-front, the order, or priority, of the filling is critical to the output. Microsoft researchers Criminisi et al.'s implementation features the use of a "confidence term", prioritizing the filling of pixels locating closest to the source (known) region.[1] More specifically, this confidence term evaluates each edge pixel with its surrounding pixels (depending on custom path radius), and gives a ratio of pixel location in the fill versus source region. For example, if a pixel is located on the fill-front and has 2 out of 9 surrounding pixels located within the source region, then it will obtain a lower fill priority than a pixel with 5 out of 9 surrounding pixels located within the source region. Li et al.[20] implemented an advanced version of Criminisi et al.'s algorithm by adding a similarity term based on Non-Local-Mean method, which measures how similar the current pixel patch is to the rest of the regions within the image. For our evaluation of the EBI algorithm, the priority term is defined with both the confidence term and the similarity term according to Li et al[20]:

$$C_{ij} = \frac{\Sigma_{mn\in\Psi_{ij}\cap(I-\Omega)} C_{mn}}{|\Psi_{ij}|} \tag{1}$$

$$S_{ij} = \| \frac{1}{Z_i} \exp^{-\frac{\Sigma_{mn}(k_{mn} v((N_i)_{mn} - (N_j)_{mn}))_2^2}{h^2}} \|_2 \tag{2}$$

$$Z_i = \Sigma_j \exp^{-\frac{\|v(N_i) - v(N_i)\|_{2,a}^2}{h^2}} \tag{3}$$

For each iteration, the pixel with the highest priority enters the filling stage and has its value assigned by a normalized weighted sum of its surrounding source region pixels. The pixel weight estimation with L2-norm gives emphasis to those pixels located closer to the inpainting pixel and the boundary normal.[20][21] Upon every pixel update, the fill-front pixel priority is re-evaluated and the new pixel with highest priority proceeds to the filling stage. The algorithm iterates until the entire fill region is complete.

4.2. DCGAN

For our DCGAN approach, the general idea is to generate an image that looks close to the target image and then take the corresponding area to fill in the target image. First, using a dataset of images as input, we train a DCGAN model following the algorithm proposed in Goodfellow et al.[10]. As show in Fig. 2, the discriminator uses four convolution 2D layer with relu activation and finally a fully connected linear layer. The generator, shown in Fig. 3, is almost the exact opposite. It's first a linear layer combined with a reshape to transform the input noise z into appropriate dimensions followed by four convolution 2D transpose layers with relu activation. Adam optimizer is used with standard parameters to ensure reliable training updates. We then have a discriminator D that discriminates between "real" and "fake" (from the dataset or generated) images and a generator G that takes in a random input noise z and generates a realistic image with respect to the dataset.

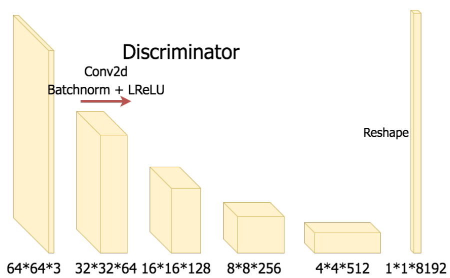


Figure 2: Discriminator architecture

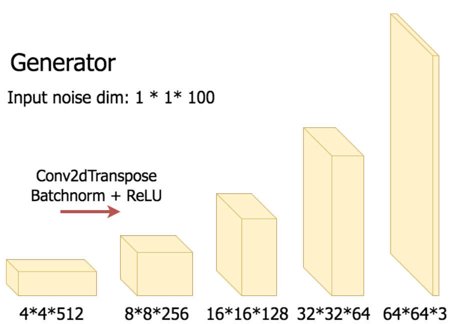


Figure 3: Generator architecture

We want to obtain an input z that allows the generator to produce a desired image to use for image completion. The work flow of this process is captured by Fig. 4. For this optimization problem, we define the loss as a weighted sum of the contextual loss and the perpetual loss of the generated image. Contextual loss captures the intuition that if the generated image $G(z)$ have similar values at places where the original image is not missing, then the part of $G(z)$ where we use to complete the original image should fit reasonably well. More specifically, we define mask M as a binary matrix with the same size as our input image. M has value 1 at pixel locations where the image is not cropped out and 0 otherwise. We then formalize the loss as the L1 distance between the masked original image and the masked generated image as shown in equation 1. This ensures that the image we generate is as similar as possible to the target image in unmasked parts. We have also tried to use l2 distance between the two images but l1 distance seems to be empirically better. In contrast, we define perpetual loss to ensure our generated image is realistic. Intuitively, although our generator is trained to produce images that look like they come from the training dataset, there is no guarantee that passing in a random input noise z through the generator will get us one. Naturally, we can pass a generated image G(z) through discriminator D and evaluate the loss as below. After many iterations of gradient descent, we obtain a realistic image that looks like our target image. In our case, we simply hard coded 1000 iterations of gradient descent, but we could have easily defined a converging condition just like any other gradient descent problem. Finally we use the corresponding area of the generated image to complete the target image.

$$L_{conceptual}(z) = ||M \otimes I - M \otimes G(z)||_1 \tag{4}$$
$$L_{perceptual}(z) = log(1 - G(z)) \tag{5}$$
$$L(z) = L_{conceptual}(z) + \lambda * L_{perceptual}(z) \tag{6}$$
$$\hat{z} = \arg \min_z L(z) \tag{7}$$
$$x_{reconstructed} = M \otimes I + (1 - M) \otimes G(z) \tag{8}$$

5. Results and Evaluation

In this section, we present our evaluation results for the above two algorithms, mainly focusing on their compared quality of output images as well as their realism to human audiences. The dataset we used for validation and test is LFW dataset, and sample outputs are presented in the following subsections.

5.1. Objective Audience Evaluation

Originally, we planned to have an experimentation process as depicted in Fig. 5. An experiment set is a set of image pairs that focuses on a category of image completion tasks. For instance, image completion is often a tool used in object removal, and we planned to set up an experiment set which contains original images that require some objects to be removed. Then we let a group of people go through the experiment set, and for each image pair, we planned to

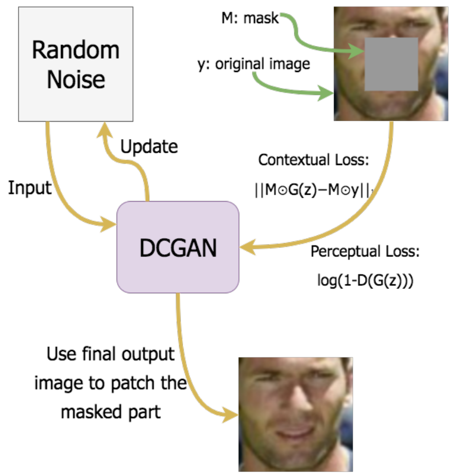


Figure 4: Completion process

record the better ones scored by audiences and discuss their indications.

However, we soon realized a problem that made this experiment inaccessible: for each new category of image completion task we choose, we need to find a dedicated training dataset to train our DCGAN algorithm. This would be an impossible amount of workload for the limited time frame we have. Thus, for comparing between EBI and DCGAN based algorithms, we still continued with the scoring approach, but since the results generated by EBI on human faces are far from ideal, we had to ask the audience to compare outputs of EBI and DCGAN based approach that were completed from two different original images in order to get competitive results. We also tried to train DCGAN on larger scenery pictures, since those are the type of images EBI performs better[2]. However, the training time it takes for DCGAN to train on such large images were too long for us to do an efficient and meaningful hyperparameter search. Thus we had to devise an objective scoring system that can address the gap in categories of original images that two algorithm run on, which we elaborate below.

As for objectively scoring the images, we asked the audience to score based on: (1): image complexity; (2): image realism; and (3): "cost performance" of the images generated. The image complexity factor accounts for the algorithm's ability to perform image completion on "harder" images. This "hardness" is introduced as an effort to map different contents of original images to the same space in image completion: if EBI is completing an image with complexity score of 8, and DCGAN based approach is also completing a region with complexity score of 8, then we state that the two algorithms are completing roughly the same level of tasks. Image realism factor accounts for the algorithm's successfulness in terms of convincing people, since after all the purpose of the algorithm is to generate realistic completed images. "Cost performance" factor accounts for the time cost that audience has to pay in order to get the completed images. In other words, the amount of time audience has to wait to "pay" for the image completion task to be finished.

[2]Our experimentation result show that EBI performs the best roughly when the input images are bigger than 300*300 with repetitive patterns

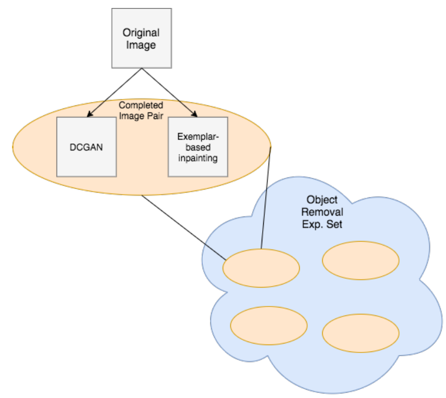


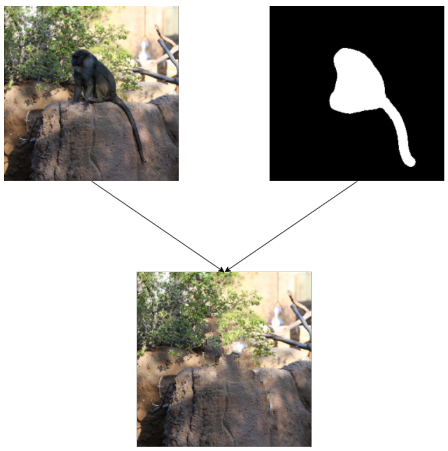
Figure 5: Example of the objective audience evaluation process.

The sample outputs of the two algorithms are in Fig. 6, and sample result of the comparison of the two algorithms is summarized in table 1. To obtain scores for the comparison sets, we asked 30 people with different demographic backgrounds to ensure enough randomization in the scoring process. We collected scores on 5 different comparison sets that have different levels of our subjective complexity ratings from us to ensure our experiment is covering a wide range of image content difficulties. The corresponding comparison set to the table 1 is provided in Fig. 7. The rest of the four comparison sets are attached in appendix section.

From the results we obtained, it was clear that EBI's quality of output images is heavily influenced by the pixel information it sees in the images it is completing. The more complex the original structure is, the worse it performs since it becomes harder to draw patterns from the surrounding complex image structures. Fig. 8 presents a failed sample output by EBI on a set of human faces. From the same line of perspective, DCGAN based approaches tend to perform much more stable and better on quality of output images. However, interestingly from the data points we collected, despite the fact that the output images look better, it shows that the audience do not necessarily scores more on the cost performance factor for DCGAN based approaches than they do on EBI (the details on the algorithm's runtime is elaborated in the following subsection). This shows that the audience we have interviewed has a non-negligible

Algorithm	Image Complexity (0-10)	Image Realism (0-10)	Image Test Time Cost Performance (0-10)
EBI	6.0	9.3	9.3 (˜1.2 min)
DCGAN	7.6	8.2	7.2 (˜7.5 min)

Table 1: Sample result of the scores collected on scoring set 1. All scores are scored on range 0 to 10 with 10 being the most ideal.



(a) Sample output of Exemplar-Based Inpainting. The algorithm uses the mask fill in the masked portion with the surrounding pixel information.

(b) Sample output of DCGAN based approach (left: original, right: completed).

Figure 6: Sample outputs of the two image completion algorithms.

preference on the runtime of the algorithm, and for any application that uses image completion algorithm, developer should think carefully about the trade off between output image quality and the runtime of algorithm.

5.2. Difference between Output Images and Original Images

Aside from the scoring evaluation we performed on our objective audience, we also analyzed our DCGAN based image completion algorithm's performance by recording its L1 distances between its output images and original images. Since the original images are real images, we can state that the closer our outputted images are to the original images, the more realistic completed images it produces.

We chose to perform this task on our test set, which comprises 1,000 images from LFW dataset. In Fig. 9 we plotted the L1 distances of a sample batch under 4000 iterations while performing completion task, and Table 2 summarizes our L1 distance stats on all of the 1,000 images under different number of training iterations.

Iters	Max	Min	Mean	Std.	Pxl. Avg.
500	3837.5	966.3	1285.5	449.6	0.31
1000	3837.5	865.5	1095.7	370.8	0.27
2000	3837.5	816.1	964.6	293.3	0.24
3000	3837.5	806.25	913.2	250.3	0.22
4000	3837.5	801.49	885.7	221.9	0.21

Table 2: L1 Distance Stats on Test Set

As indicated from the above data, we can see that our DCGAN based image completion algorithm eventually achieves an average L1 distance of 885.7, which comes down to an average loss per pixel of mere 0.21. This implies that our DCGAN based image completion algorithm's outputs are very close to the original images (each generated pixel is only differing from the original by 0.21!), and the visual outputs (Fig. 6) can further confirm our conclusion obtained quantitatively from above.

However, we can also see from the above figures that

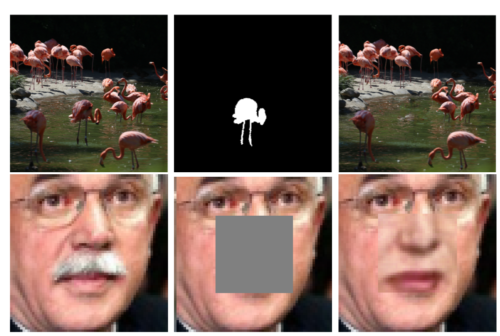


Figure 7: Example of one of the comparison sets we used (corresponding to result presented in Table 1). From left to right: original images, masks, completed images.

Figure 8: Sample of running EBI algorithm on human faces.

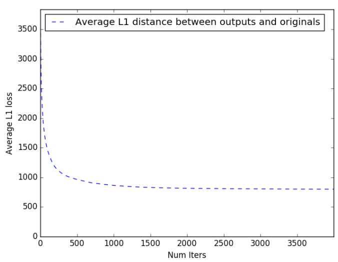


Figure 9: Average L1 distance over number of iterations

the output convergence starts around 1000 iterations seconds, which takes roughly 3 4 minutes. This long test time in fact negatively penalizes the user quality of experience by a non-negligible amount. Results attached in Appendix further proves this point: DCGAN based image completion algorithm has a stable and high scores on image realism, but it hardly has high scores on ”cost performance” factor.

6. Future Work

As discussed in evaluation section, currently our DCGAN based image completion algorithm is significantly slower than EBI, or other deep learning based approaches which directly learn the mapping from masked regions to completed images. Thus, the runtime optimization of our GAN based approach could be our next step for this project. One possible solution to address this problem is to prune unlikely outputs early. For example, if the unmasked region of the input has a skin color of black, then we might better off starting from input noise z that generates a human face that has skin color of black rather than just starting randomly in the z space. We would then also have to implement a convergence criteria during optimization of z instead of hard coding a certain number of iterations.

Additionally, our current DCGAN based approach does not smooth out the edge of the masked region after it has patched it with generated fake image. This leaves final completed image to appear a bit discontinuous when looked closely. For future development, we can introduce additional image processing algorithm to smooth out the edges so that the output images will look even more realistic than what they are now.

Moreover, DCGAN based algorithm's ability to complete images is limited to the training dataset it has seen before. We could have used a larger model which can potentially maintain more knowledge about the training images, so that users do not have to train individual networks for each different category of image completion problems. However, this solution obviously does not scale well as the categories become richer, and generalizability of neural networks is another area of research that is beyond the scope of this project.

7. Conclusion

In this paper, we presented two different image inpainting algorithms, namely Exemplar-Based Inpainting and Deep Convolutional GAN based image completion algorithm, and presented our evaluations of them. Exemplar-Based Inpainting algorithm typically runs faster, and it applies to larger types of images since it does not require any training, but its quality of output images is also heavily dependent on the surrounding pixel information it sees, and generally fails to patch images with large masked regions or with complex structures. On the other hand, DCGAN based approach tend to be slower during the test time as it needs to forward and backward propagate multiple times to search for the best input noise z. However, its quality of output images is notably better than Exemplar-Based Inpainting, and it can even extend the complex structures of the images in a very realistic way to complete the masked regions.

Given its fast runtime and acceptable quality of output, we think that Exemplar-Based Inpainting is suitable for applications where users require quick responsiveness. For example, mobile applications that automatically ”photoshop” pictures taken (to remove some small corrupted points). On the other hand, despite its requirement of huge commitment during training process and long runtime during testing, we think that DCGAN based approach is suitable for a different set of scenarios, scenarios that can tolerate time consuming completion process but require high quality and accuracy, such as scene reconstruction for criminal investigation, or human body imaging (tumor detection) for medical purposes, etc.

8. Appendices

Other four comparison sets are included here: Fig. 10, 11, 12, 13 presents the comparison sets and Table 3 presents the average scorings obtained from our audience.

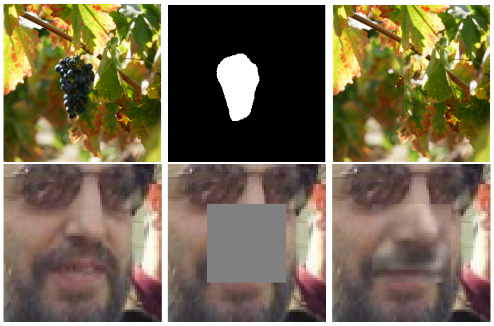


Figure 10: Comparison set 2.

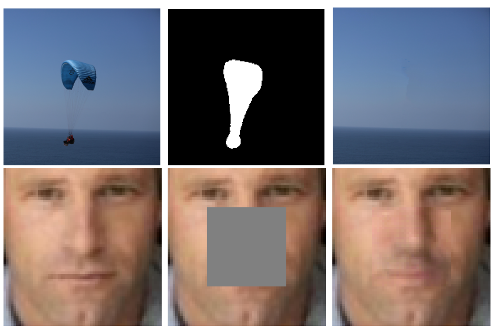


Figure 11: Comparison set 3.

Figure 12: Comparison set 4.

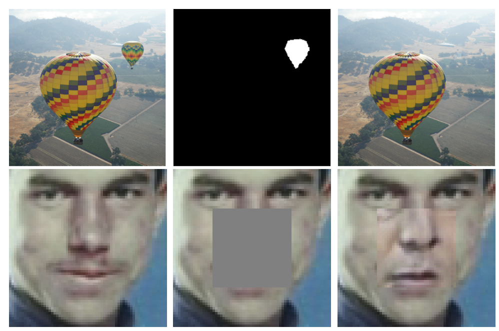


Figure 13: Comparison set 5.

Set 2	Image Complexity (0-10)	Image Realism (0-10)	Image Test Time Cost Performance (0-10)
EBI	7.9	9.4	9.2 (˜1 min)
DCGAN	7.7	8.2	7.3 (˜7.5 min)
Set 3	–	–	–
EBI	5.2	9.9	9.7 (˜0.8 min)
DCGAN	6.4	8.3	7.4 (˜7.5 min)
Set 4	–	–	–
EBI	8.2	8.1	8.0 (˜1.1 min)
DCGAN	7.9	8.2	8.0 (˜7.5 min)
Set 5	–	–	–
EBI	5.3	9.0	9.2 (˜0.6 min)
DCGAN	6.4	8.8	8.0 (˜7.5 min)

Table 3: Results of comparison sets 2 - 4

References

[1] A. Criminisi, P. Perez, and K. Toyama, *Object removal by exemplar-based inpainting*, CVPR, 2003. Proceedings of the 2003 IEEE Computer Society Conference on (Vol. 2, pp. II-II).

[2] Bertalmio, Marcelo, Guillermo Sapiro, Vincent Caselles, and Coloma Ballester. *Image inpainting.* In Proceedings of the 27th annual conference on Computer graphics and interactive techniques, pp. 417-424. ACM Press/Addison-Wesley Publishing Co., 2000.

[3] M. Bertalmio, G. Sapiro, C. Ballester and V. Caselles, *Computer Graphics*, SIGGRAPH 2000, pp. 417-424, July 2000.

[4] M. Bertalmio, A. L. Bertozzi, and G. Sapiro, *Navier-stokes, fluid dynamics, and image and video inpainting*, CVPR, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on (Vol. 1, pp. I-I).

[5] Radford, Alec, Luke Metz, and Soumith Chintala. *Unsupervised representation learning with deep convolutional generative adversarial networks.* arXiv preprint arXiv:1511.06434 (2015).

[6] Arjovsky, Martin, Soumith Chintala, and Lon Bottou. *Wasserstein gan* arXiv preprint arXiv:1701.07875 (2017).

[7] Sun, Jian, Lu Yuan, Jiaya Jia, and Heung-Yeung Shum. *Image completion with structure propagation.* ACM Transactions on Graphics (ToG) 24, no. 3 (2005): 861-868.

[8] Avidan, Shai, and Ariel Shamir. *Seam carving for content-aware image resizing.* In ACM Transactions on graphics (TOG), vol. 26, no. 3, p. 10. ACM, 2007.

[9] Yan, Bo, Yiqi Gao, Kairan Sun, and Bo Yang. *Efficient seam carving for object removal.* In Image Processing (ICIP), 2013 20th IEEE International Conference on, pp. 1331-1335. IEEE, 2013.

[10] Goodfellow, Ian, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. *Generative adversarial nets.* In Advances in neural information processing systems, pp. 2672-2680. 2014.

[11] Fawzi, Alhussein, Horst Samulowitz, Deepak Turaga, and Pascal Frossard. *Image inpainting through neural networks hallucinations.* In Image, Video, and Multidimensional Signal Processing Workshop (IVMSP), 2016 IEEE 12th, pp. 1-5. IEEE, 2016.

[12] Khler, Rolf, Christian Schuler, Bernhard Schlkopf, and Stefan Harmeling. *Mask-specific inpainting with deep neural networks.* In German Conference on Pattern Recognition, pp. 523-534. Springer International Publishing, 2014.

[13] Yeh, Raymond, Chen Chen, Teck Yian Lim, Mark Hasegawa-Johnson, and Minh N. Do. *Semantic Image Inpainting with Perceptual and Contextual Losses.* arXiv preprint arXiv:1607.07539 (2016).

[14] Patel, Pritika, Ankit Prajapati, and Shailendra Mishra. *Review of different inpainting algorithms.* International Journal of Computer Applications 59, no. 18 (2012).

[15] Sangeetha, K., P. Sengottuvelan, and E. Balamurugan. *A Concert EVALUATION OF EXEMPLAR BASED IMAGE INPAINTING ALGORITHMS FOR NATURAL SCENE IMAGE COMPLETION.*

[16] Tiefenbacher, Philipp, Viktor Bogischef, Daniel Merget, and Gerhard Rigoll. *Subjective and objective evaluation of image inpainting quality.* In Image Processing (ICIP), 2015 IEEE International Conference on, pp. 447-451. IEEE, 2015.

[17] Liu, Ziwei, Ping Luo, Xiaogang Wang, and Xiaoou Tang. *Deep learning face attributes in the wild.* In Proceedings of the IEEE International Conference on Computer Vision, pp. 3730-3738. 2015.

[18] Gary B. Huang, Manu Ramesh, Tamara Berg, and Erik Learned-Miller. *Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environments.* University of Massachusetts, Amherst, Technical Report 07-49, October, 2007.

[19] Amos, Brandon, Bartosz Ludwiczuk, and Mahadev Satyanarayanan. *OpenFace: A general-purpose face recognition library with mobile applications.* Technical report, CMU-CS-16-118, CMU School of Computer Science, 2016.

[20] Haihong Li, Joanna Xu, Chen Zhu *Object/Defect Removal via Single-image Super-resolution on NLM-priority-based Inpainting and Sparse Coding*

[21] A. Telea, *An image inpainting technique based on the fast marching method*, Journal of graphics tools, 9(1), 23-34.

[22] Brandon Amos. *Image Completion with Deep Learning in TensorFlow* http://bamos.github.io/2016/08/09/deep-completion