

Photobombing Removal Benchmarking

Sai Pavan Kumar Prakya, Madamanchi Manju Venkata Sainath,
Vatsa S. Patel, Samah Saeed Baraheem, and Tam V. Nguyen*

Department of Computer Science, University of Dayton, Dayton OH 45469, USA

*Corresponding Author: tamnguyen@udayton.edu

Abstract. Photobombing occurs very often in photography. This causes inconvenience to the main person(s) in the photos. Therefore, there is a legitimate need to remove the photobombing from taken images to produce a pleasing image. In this paper, the aim is to conduct a benchmark on this aforementioned problem. To this end, we first collect a dataset of images with undesired and distracting elements which requires the removal of photobombing. Then, we annotate the photobombed regions which should be removed. Next, different image inpainting methods are leveraged to remove the photobombed regions and reconstruct the image. We further invited professional photoshoppers to remove the unwanted regions. These photoshopped images are considered as the groundtruth. In our benchmark, several performance metrics are leveraged to compare the results of different methods with the groundtruth. The experiments provide insightful results which demonstrate the effectiveness of inpainting methods in this particular problem.

Keywords: Photobombing removal, image inpainting, benchmark, performance metrics.

1. Introduction

“Images bring back forgotten memories”

Capturing any picture aims to record the moment/fact behind it. It remains to be a beautiful/good memory back in one’s life. People tend to click pictures often when they are happy, when they visit new places, or to create a memory with family and friends during family gatherings. Every picture is a unique reflection of the people and/or things present in the frame. Thus, the vital part is that it makes us comprehend more about its history. Photobombing occurs when a person or object intentionally or unintentionally comes into the camera field of view, just before the photo is taken. Thus,

they appear unexpectedly in the background and make it evident that they were not supposed to be the main "subject" of the photograph.

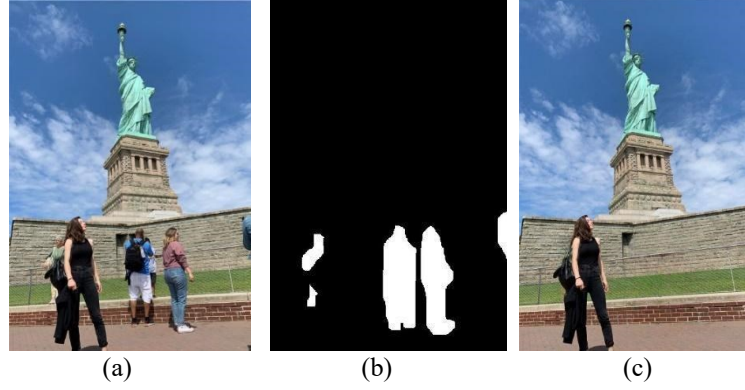


Figure 1. Inputs and output of photobombing removal process. From left to right: (a) the original photobombed image, (b) the mask of the unwanted regions, (c) the photobombing removal image.

This would eventually ruin the memories of the picture taken at an historic place, at some close friend's wedding, or during a sudden fan moment. The gravity of the situation of photobombing would transform the beautiful memories into a disaster and ruin the picturesque scene. This act of sabotage may be seen in anything from a selfie to a professional image. However, if you purposefully or inadvertently intrude yourself into a photograph that was not intended for you, you are a photo bomber and this behavior is now known as "photobombing". This has been originated back in 1853 [1], where the first suspected photobombing happened. An anonymous person who popped his head in when two women were being photographed, although he was never supposed to be in the shot.

Each one of us will have a specific reason to capture a photograph, but all we want is to capture those split seconds and store them as an everlasting memory. Indeed, a photo has the power to grab our attention and communicate with our emotions. Few moments cannot be recreated but can only be remembered and relived in our hearts through walking over a captured photo without any obstruction. In case the photos are photo bombed, then it is indispensable and essential to remove photobombing from the picture.

Removing the unwanted region from an image and reconstructing it to achieve an image devoid of a photobomb is known as image inpainting. We can remove photobombing using Adobe Photoshop tool, but it requires a good skill set to use the tool. Additionally, it is time consuming and a tedious task. To overcome this, there are few state-of-art inpainting algorithms which can be easily accessible through website and mobile applications. Figure 1 shows an example of the input and output of a photobombing removal method.

In this paper, we have benchmarked the results of image inpainting algorithms. To the best of our knowledge, we are the first ones to address this interesting yet challenging problem of benchmarking the images with the undesired region removed. Therefore, we first collect photobombed images from various online sources. Then, masks are

generated by annotating the unwanted areas in our collected images. Next, image inpainting algorithms are used, where the inputs to the algorithms are the photobombed photos and masks. Following this, the results are compared with the groundtruth images. In fact, having groundtruth images to compare with the results of the inpainting algorithms is one of the primary challenges. Therefore, we have photoshopped each and every image to eliminate the unwanted or photobombed region from it in order to create a groundtruth image. Finally, we use Frechet Inception Distance (FID) [2], Structural Similarity Index (SSIM) [3], and Peak to Noise Signal Ratio (PSNR) [4] in the evaluation stage. Furthermore, we propose the Texture-based Similarity Index (TSI) [22] into the benchmark.

The remainder of this paper is organized as follows. Section 2 briefly describes the related work. Section 3 introduces the benchmarking dataset and methods. Section 4 presents the extensive evaluation on the benchmark. Finally, Section 5 concludes the paper and paves way to the future work.

2. Related work

2.1 Traditional Methods

Criminisi et al. [5] describe a new and efficient approach that combines the benefits of "texture synthesis" and "inpainting." The key technique necessary to recreate both texture and structure is contained in exemplar-based texture synthesis, although it is very dependent on the sequence in which the filling proceeds. This work presents a best-first technique that propagates confidence in synthetic pixel values in a way comparable to information propagation in inpainting. Efficient approach is used in this paper to propagate texture and structure information at the same time. Computational efficiency is achieved by a block-based sampling process. Since a significant number of repetitions may be required for stability considerations, resulting in a computational complexity that is frequently too large for interactive image editing. Meanwhile, Bornemann et al. [6] introduce a fast non-iterative method for image inpainting based on a deep analysis of stable first order transport equations. It traverses the inpainting domain once using the fast marching method, conveying picture data in a coherence direction securely calculated by the structure tensor along the way. The approach alternates between diffusion and directional transport based on a measure of coherence strength.

Moreover, Telea [7] proposes the fast marching method-based novel algorithm for digital inpainting. The missing sections are inpainted using the Fast Marching Method after a mathematical boundary model has been obtained. Bertalmio et al. [8] present a group of automated techniques for digital inpainting. The method propagates isophote lines constantly from the exterior into the area that will be painted by applying concepts from classical fluid dynamics. The resulting algorithm's objective is to maintain isophotes while matching gradient vectors at the inpainting region's edge.

2.2 Deep Learning-based Methods

Jiahui et al. [9] introduce a generative image inpainting method for image completion using a free-form mask and guidance. The technique is constructed on gated

convolutions learned from image dataset without the need for extra labeling. The proposed gated convolution addresses the issue of vanilla convolution, which regards all input pixels as legitimate, by providing a learnable dynamic feature selection mechanism for each channel at each spatial location across all layers. Furthermore, because free-form masks can appear in images of any shape, global and local GANs created for a single rectangular mask are inapplicable. As a result, researchers present SN-PatchGAN, a patch-based GAN loss function that uses a spectral-normalized discriminator on dense image patches.



Figure 2. A sample of the inputs to image inpainting methods. While (a) shows the photobombed images, (b) represents the corresponding generated masks of undesired regions

Despite tremendous advances, modern image inpainting systems frequently struggle with vast missing portions, complicated geometric patterns, and high-resolution images. One of the key reasons for this is the lack of an efficient receptive field in both the inpainting network and the loss function. To better this issue, Roman et al. [10] propose a new technique termed huge mask inpainting (LaMa). LaMa is built on three components, i) a unique inpainting network architecture that employs Fast Fourier Convolutions (FFCs) with an image-wide receptive field, ii) a high receptive field perceptual loss, and iii) huge training masks that unlock the potential of the first two components.

In addition, photobombing inspires other research works. For example, data augmentation makes machine learning more robust by synthesizing more training data [30]. Mohamad et al. [31] proposed photobombing guided data augmentation in order to improve the performance of event recognition.

In this paper, we benchmark the performance of widely available and state-of-the-art image inpainting algorithms. Specifically, a total of six image inpainting algorithms are evaluated. First, we collect 150 photobombed images from various online sources, followed by annotating the unwanted area to generate masks. Reconstructed images without the photobomb are the final output of the algorithms. We obtained a total of 900 reconstructed images using 6 image inpainting methods and 150 images. To benchmark the outputs of algorithms, we have created 150 photoshopped images by manually removing the photobombed area. These photoshopped images are utilized as groundtruth images. Afterwards, to achieve the results, we benchmark each of the 900 images against its corresponding photoshopped image and calculate Frechet Inception Distance (FID) [2], Structural Similarity Index (SSIM) [3], Peak to Noise Signal Ratio (PSNR) [4], and Texture-based Similarity Index (TSI) [22].

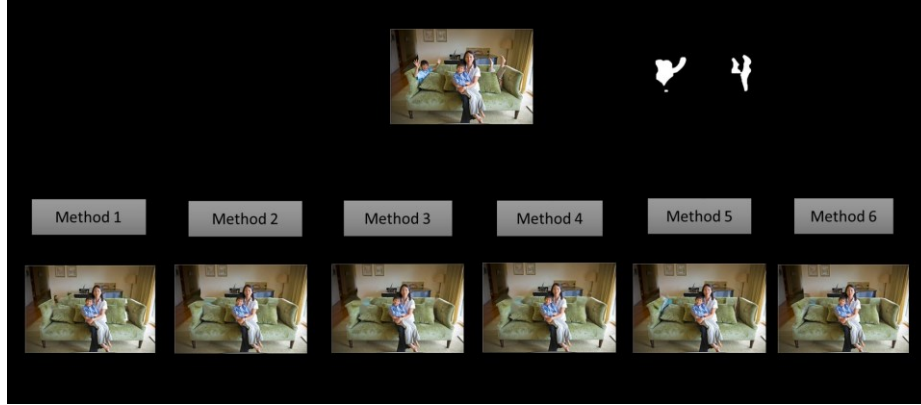


Figure 3. The flowchart of photobombing removal with image inpainting methods.

3. Photobombing Removal Benchmark

3.1 Benchmarking Dataset

To evaluate the methods using performance metrics, photobombed images, logical masks, and custom edited images are needed. In total, we collected 150 photobombed images. Then, we prepare masks of photobombed images by creating the masks of unwanted and distracting elements. Furthermore, custom edited images for each photobombed image are created as groundtruth. To accomplish this, Adobe Photoshop, Microsoft Paint, and the Cleanup Pictures tool are utilized to custom modify all of the photobombed images and remove the photobombed areas.

In particular, we first manually collect photobombed images from various online sources, such as Facebook [11], Getty Images [12], Bored Panda [13], Adobe stock [27], Shutterstock [28] and Pinterest [29]. To the collected images, we annotate the photobombed region using the Freehand object of the "Region of Interest" function present in matlab [14] to create logical masks. First, we implement a for loop to automate the process of loading images into Matlab one by one from local. Then, we use the "FreeHand" function of "Region of Interest "[14] to annotate the region. A single image is iterated depending on the number of photobombed items in the image until we obtain a mask, also known as a binary image.

The average time it took to annotate one image was approximately 50 - 60 seconds, depending on how many photobombed objects needed to be eliminated from a single image. More time spent on multiple objects and less time spent on single objects. Then, the photobombed images with their respective logical masks are used as inputs for various inpainting methods to remove the undesired regions.

3.2 Benchmarking Methods

The photobombed image and its associated mask are fed into different inpainting methods as inputs. A sample of the inputs fed into inpainting methods is illustrated in Figure 2. The output is a reconstructed or an inpainted image. For the benchmarking, different inpainting methods are used. Specifically, Exemplar-Based Image Inpainting

(EBII) [15], Coherence Transport (CT) [16], Fast Marching (FM) [17], Fluid Dynamics (FD) [18], Gated Convolution (GC) [19], and Resolution-robust Large Mask Inpainting with Fourier Convolutions (LaMa) [10] are incorporated to determine the best method. The overview of photobombing removal using various image inpainting methods is illustrated in Figure 3. As seen in Figure 3, the photobombed image alongside its corresponding mask are fed into various image inpainting methods to remove the unwanted and distracting elements from the original photobombed image and reconstruct the image. Particularly, both EBII [15] and CT [16] are an image inpainting algorithm present in Matlab [20] [21]. Meanwhile, FM [17] and FD [18] are available in OpenCV [23]. For GC [19], we utilize the author’s code provided on github [24] with a pre-trained model on Places2 dataset [25]. Meanwhile, with regards to LaMa [10], the available author’s code is used from github [26]. Finally, we run all the methods on the testing set to produce the photobombing removal images. The resulting images are compared with the groundtruth in various metrics which will be described in the following section.

4. Experiments

4.1 Performance Metrics

To benchmark the outputs of all methods, the photobombing removal images are compared with the groundtruth images which are custom edited images. Several evaluation metrics are used to assess the results of photobombing removal images. In particular, Fréchet inception distance (FID) [2], Structural Similarity Index (SSIM) [3], and Peak to Noise Signal Ratio (PSNR) [4] are used to compare the reconstructed images to custom edited images. We further propose Texture-based Similarity Index (TSI) based on Local Binary Patterns (LBP) [22].

With regards to FID [2], it computes the difference between the distribution of the reconstructed image and the distribution of the groundtruth based on the extracted features, as shown in Equation (1).

$$FID(x, y) = d^2 = ||\mu_x - \mu_y||^2 + \text{Tr}(c_x + c_y - 2 * \text{sqrt}(c_x * c_y)) \quad (1)$$

, where μ_x and μ_y are the feature-wise mean of the groundtruth and reconstructed image, and c_x and c_y refer to the covariance matrix of the feature vectors of ground truth and reconstructed image. Tr indicates the trace linear algebra operation.

For SSIM [3], it calculates the similarity between the groundtruth and reconstructed image, where three main features are extracted from the given images which are luminance, contrast, and structure. These three features are considered during the comparison between two images. The equation of SSIM is demonstrated in Equation (2).

$$SSIM(x, y) = [l(x, y)]^\alpha \cdot [c(x, y)]^\beta \cdot [s(x, y)]^\gamma \quad (2)$$

, where x and y are the ground truth and the corresponding reconstructed image. l , c , and s are the luminance, contrast, and structure, respectively. α , β , and γ refer to the relative importance of each metric.

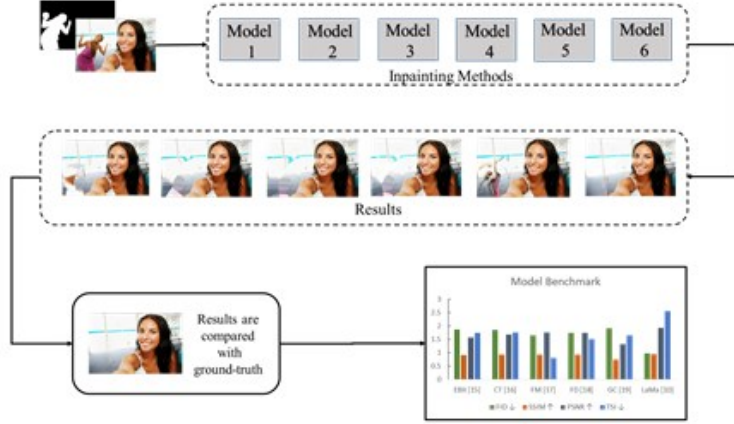


Figure 4. The performance of different inpainting methods on our benchmark dataset in terms of FID [2], SSIM [3], PSNR [4], and TSI [22].

In terms of PSNR [4], it measures the ratio between the maximum possible power of a signal and the power of distorting noise. The mathematical representation is shown in Equation (3).

$$PSNR = 20 \log_{10} \left(\frac{MAX_f}{\sqrt{MSE}} \right) \quad (3)$$

, where Mean Squared Error (MSE) is computed as in Equation (4).

$$MSE = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} ||f(i, j) - g(i, j)||^2 \quad (4), \text{ where } f$$

and g are the data matrix of the groundtruth and the corresponding reconstructed image. m indicates the number of pixels in rows of the image and i denotes the index of that row. n refers to the number of pixels in columns of the image and j is the index of that column. In all the aforementioned metrics, we observe that they have a strong correlation. Therefore, in this paper, we further propose another metric, namely, Texture-based Similarity Index (TSI). For this metric, we extract textural features, i.e., Local Binary Patterns (LBP) [22] from both images, namely, photobombing removal image and the groundtruth. We compute the Chi-Squared (χ^2) distance between the two extracted features as:

$$\chi^2(lbp_1, lbp_2) = \frac{1}{2} \sum_{m=1}^{256} \frac{[lbp_1(m) - lbp_2(m)]^2}{lbp_1(m) + lbp_2(m)} \quad (5)$$

Then, we rank the benchmarking methods for each test image. The final TSI is computed based on the average rank across the testing image set.

Table 1. The performance of different image inpainting methods in our benchmark in terms of FID [2], SSIM [3], PSNR [4], and TSI.

Benchmarking Models	FID ↓	SSIM ↑	PSNR ↑	TSI ↓
EBII [15]	56.7610	0.9146	23.0781	3
CT [16]	56.1764	0.9341	24.8162	2
FM [17]	49.8993	0.9378	25.7535	6
FD [18]	53.0694	0.9373	25.5839	5
GC [19]	58.1851	0.7502	19.5278	4
LaMa [10]	29.7018	0.9448	28.3508	1

4.2 Experimental Results

The performance assessment is conducted on our dataset to evaluate the results of different inpainting methods in terms of removing the photobombed regions while preserving the overall quality. In particular, FID [2], SSIM [3], and PSNR [4] metrics are calculated to measure the results of various inpainting techniques with the groundtruth. In addition, a new metric, namely, Texture-based Similarity Index (TSI) is introduced. The performance of various image inpainting methods on our benchmark is illustrated in Figure 4 and Table 1. As can be seen from Table 1, LaMa [10] achieves the best performance with 29.7018, 0.9448, and 28.3508 for FID [2], SSIM [3], and PSNR [4], respectively. Moreover, LaMa [10] achieves the best rank in terms of the average rank across the testing set based on Texture-based Similarity Index (TSI). This might be attributed to the efficient receptive field in the inpainting model which utilizes fast Fourier convolutions (FFCs). Furthermore, LaMa [10] uses a receptive field loss function that effectively helps in generating the missing regions. Surprisingly, GC [19] obtains the worst results among the other inpainting techniques. GC [19] model has the capability of reconstructing the masked region using the trivial pixel details present around the region. Sometimes those trivial pixels are insignificant for reconstructing, thus not providing better results in photobomb removal. The results of different inpainting techniques, which are used in this study, are shown in Figure 5. It can be observed that the outputs produced by LaMa [10] outperform all other used inpainting methods.

Figure 6 (i) shows which inpainting method performs more effectively based on FID score [2] with reference to the percentage of the region to be inpainted. As seen in Figure 6 (i), when the percentage is between 0 and 10, all algorithms performed better in terms of FID [2] since the FID value is smaller than in other percentage ranges. When compared to models CT [16], FM [17], FD [18], GC [19], the EBII [15] model performed well in the 0-10 and 40-50 ranges. However, performance dropped in the 10-20, 20-30, and 30-40 percentage levels. Model FM [17] maintained average performance across all ranges. We can clearly observe that the LaMa [10] model performed well across all percentage levels. The trend shows that the FID score is directly proportional to the percentage of the region to be inpainted.

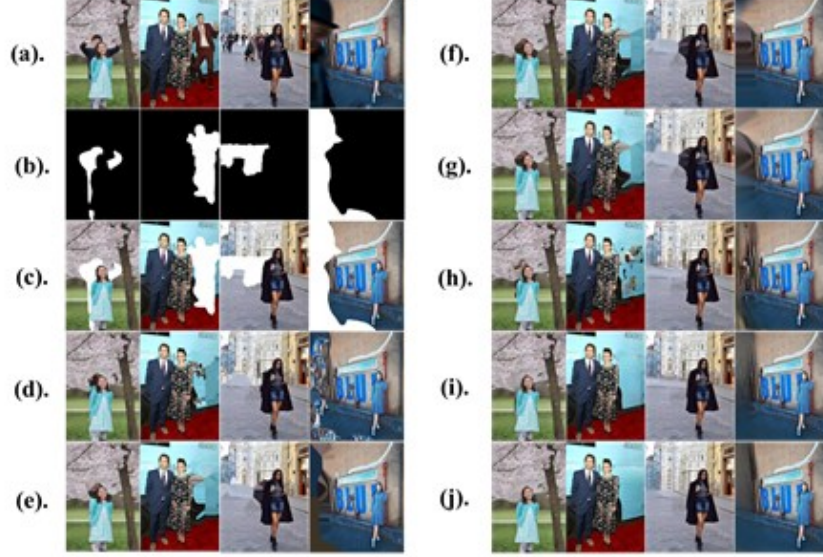


Figure 5. A sample of different inpainting methods' outputs in our proposed framework to remove undesired regions. From top to bottom, (a) Photobombed images, (b) Generated masks, (c) Image with mask, (d) EBII [15], (e) CT [16], (f) FM [17], (g) FD [18], (h) GC [19], (i) LaMa [10], and finally (j) the ground-truth images.

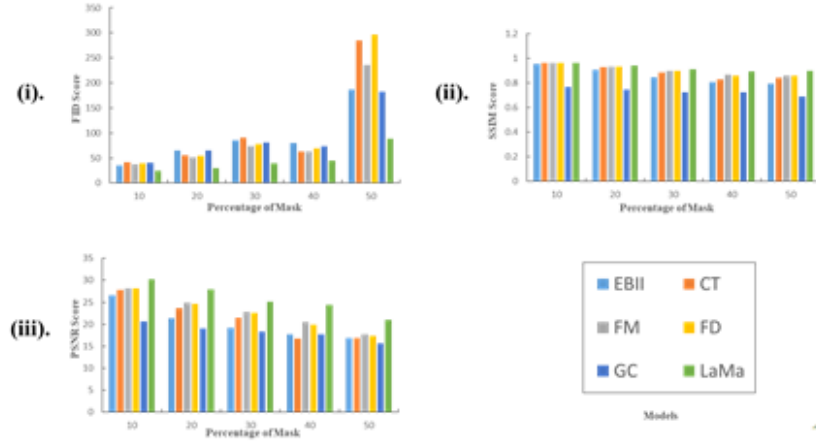


Figure 6. Performance of different inpainting methods with percentage of mask used for inpainting. (i) FID score [2] with percentage of masks, (ii) PSNR score [4] with percentage of mask and (iii) SSIM score [3] with percentage of mask.

Figure 6 (ii) illustrates the performance of methods based on the SSIM score [3] in terms of the percentage of region to be inpainted. The chart shows that when the

percentage is between 0 and 10, all algorithms perform better since the SSIM value is closer to 1. We observe that GC[19] model's performance was less when compared to all the models and maintained SSIM score less than 0.8 in all the levels. Expectedly, LaMa [10] outperformed in all percentage levels and maintained an SSIM score that is nearly greater than 0.9 in all percentage levels. The SSIM score is inversely proportional to the percentage of the region to be inpainted. Figure 6 (iii) illustrates the performance of methods based on the PSNR score[4] in terms of mask percentage. The bar chart illustrates that as the mask percentage increased, the PSNR score declined. LaMa[10] excelled at all percentage levels, as expected, by having high PSNR score. When compared to other models, GC[19] has lower PSNR at all percentage levels. CT[16] performs better than EBII[15] in the region of 0-30 percentage, however in the range of 30-50 range, EBII[15] performs better than CT[16].

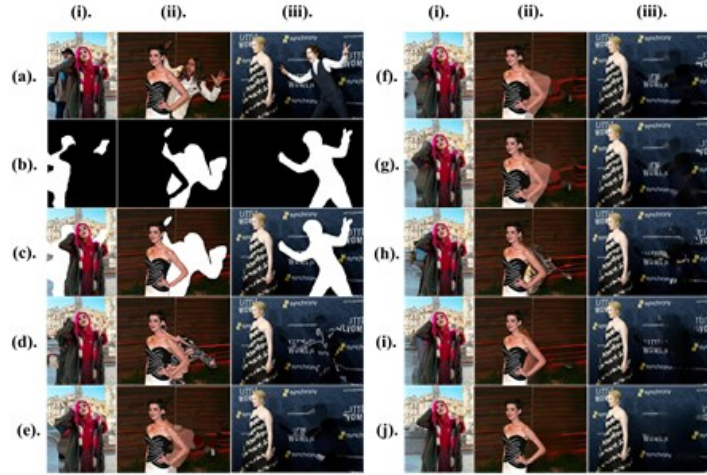


Figure 7. A sample of different inpainting methods' failure cases in our framework to remove undesired regions. (a) Photobombed images, (b) Generated masks of unwanted regions, (c) Image with mask, (d) EBII [15], (e) CT [16], (f) FM [17], (g) FD [18], (h) GC [19], (i) LaMa [10], and finally (j) the ground-truth images.

Figure 7 shows the failure cases when all inpainting methods cannot remove the photobombing regions perfectly. All methods for the image Figure 7 (a) (i) utilized the subject to generate the masked region. This is due to the fact that the percentage of subject presence in the image is more than background, and it is adjacent to the masked region. For the image in (a) (ii), LaMa [10] model, as shown in (j)(ii), performed better than all the other models, but was not able to complete the region which is inside the subject. Finally, for the image from in (a) (iii), all the models did not perform well in generating the letters texture in the image.

5. Conclusion and Future Works

Photobombing removal is the process of eliminating unwanted and distracting items in images. It is a challenging task; however, it is vital to produce aesthetic and plausible

images without undesired people or things. In this study, our dataset initially consists of photobombed images. From these images, masks of unwanted regions are generated. Then, photobombed images and their corresponding masks are fed into many inpainting models to remove the undesired regions and reconstruct the images. Following this, several evaluation metrics are calculated to validate the outcomes based on the results of the different inpainting methods and the ground truth. Our experiments demonstrate the effectiveness of using inpainting models in photobombing removal tasks.

We believe this work attracts and encourages more research works in the near future. For future work, we plan to increase the dataset size to acquire better results. Moreover, we aim to investigate more image inpainting models in camouflage images inspired by [32, 33].

Acknowledgment

This research was supported by the National Science Foundation (NSF) under Grant 2025234.

References

1. P. Edwards. "This 1853 image might show the first photobomb". 2015 <https://www.vox.com/2015/9/25/9397733/first-photobomb>.
2. M. Keeling. Frechet Inception Distance. <https://www.mathworks.com/matlabcentral/fileexchange/79071-frechet-inception-distance>
3. Zhou, W., A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. "Image Quality Assessment: From Error Visibility to Structural Similarity." *IEEE Transactions on Image Processing*. Vol. 13, Issue 4, April 2004, pp. 600–612.
4. A. Hore and D. Ziou, "Image Quality Metrics: PSNR vs. SSIM," in 2010 20th International Conference on Pattern Recognition, 2010, pp. 2366–2369.
5. Criminisi, A., P. Perez, and K. Toyama. "Region Filling and Object Removal by Exemplar-Based Image Inpainting." *IEEE Transactions on Image Processing*. Vol. 13, No. 9, 2004, pp. 1200–1212.
6. F. Bornemann and T. März. "Fast Image Inpainting Based on Coherence Transport." *Journal of Mathematical Imaging and Vision*. Vol. 28, 2007, pp. 259–278.
7. Telea, Alexandru. "An image inpainting technique based on the fast marching method." *Journal of graphics tools* 9.1 (2004): 23-34.fm
8. M. Bertalmio, A. L. Bertozzi and G. Sapiro. "Navier-Stokes, Fluid Dynamics, and Image and Video Inpainting", in Proceedings of the 2001 *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*.
9. J. Yu, Z. Lin, J. Yang, X. Shen, Xin Lu, T. Huang "Free-Form Image Inpainting with Gated Convolution", 2019 *IEEE/CVF International Conference on Computer Vision (ICCV)*.
10. R. Suvorov, E. Logacheva, A. Mashikhin, A. Remizova, A. Ashukha, A. Silvestrov, Naejin Kong, H. Goka, K. Park, V. Lempitsky "Resolution-robust Large Mask Inpainting with Fourier Convolutions", *IEEE/CVF Winter Conference on Applications of Computer Vision* 2022, pp. 3172-3182.
11. J. Fridman, Facebook Page, <https://www.facebook.com/jamesfridmanpage/photos>.
12. Getty Images, Photobombing Images, <https://www.gettyimages.com/photos/photobombing>

13. Bored Panda, Photobomb Images,
https://www.boredpanda.com/?s=photobomb&utm_source=google&utm_medium=organic&utm_campaign=organic
14. Matlab, Region of Interest, <https://www.mathworks.com/help/images/roi-creation-overview.html>
15. M. Shroff and M. S. R. Bombaywala, "A qualitative study of Exemplar based Image Inpainting," *SN Appl. Sci.*, vol. 1, no. 12, 2019.
16. F. Bornemann and T. März, "Fast image inpainting based on coherence transport," *J. Math. Imaging Vis.*, vol. 28, no. 3, pp. 259–278, 2007.
17. A. Telea, "An image inpainting technique based on the fast marching method," *J. Graph. Tools*, vol. 9, no. 1, pp. 23–34, 2004.
18. M. Bertalmio, A. L. Bertozzi, and G. Sapiro, "Navier-stokes, fluid dynamics, and image and video inpainting," in *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*.
19. Yu, Z. Lin, J. Yang, X. Shen, X. Lu, and T. Huang, "Free-form image inpainting with gated convolution," in *International Conference on Computer Vision*, 2019, pp. 4470–4479.
20. inpaintExemplar, <https://www.mathworks.com/help/images/ref/inpaintexemplar.html>
21. inpaintCoherent, <https://www.mathworks.com/help/images/ref/inpaintcoherent.html>.
22. T. Ojala, M. Pietikainen and T. Maenpää, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 971–987, 2002.
23. cv.INPAINT_TELEA https://docs.opencv.org/3.4/df/d3d/tutorial_py_inpainting.html
24. Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, Thomas S. Huang: Generative Image Inpainting With Contextual Attention. *CVPR 2018*: 5505–5514
https://github.com/JiahuiYu/generative_inpainting.git.
25. B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, and A. Torralba, "Places: A 10 million image database for scene recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 6, pp. 1452–1464, 2018.
26. LaMa: Resolution-robust Large Mask Inpainting with Fourier Convolutions.
<https://github.com/saic-mdal/lama.git>
27. Adobe stock, photobomb, <https://stock.adobe.com/search?k=photobomb>
28. Shutterstock, photobombing, <https://www.shutterstock.com/search/photobombing>.
29. pinterest, photobomb <https://www.pinterest.com/agasca11/photobomb/>
30. Q. Chung, T. Le, T. Dang, N. Vo, T. V. Nguyen, K. Nguyen, "Data Augmentation Analysis in Vehicle Detection from Aerial Videos". *RIVF 2020*, pp. 1–3
31. Y. I. Mohamad, S. Baraheem, T. V. Nguyen, "Olympic Games Event Recognition via Transfer Learning with Photobombing Guided Data Augmentation". *Journal of Imaging*, Vol. 7(2): 12, 2021.
32. T-N. Le, T. V. Nguyen, Z. Nie, M-T. Tran, A. Sugimoto, "Anabran network for camouflaged object segmentation", *Journal of Computer Vision and Image Understanding*. Vol. 184, pp. 45–56, 2019.
33. T-N. Le, Y. Cao, T-C. Nguyen, M-Q. Le, K-D. Nguyen, T-T. Do, M-T. Tran, T. V. Nguyen, "Camouflaged Instance Segmentation In-the-Wild: Dataset, Method, and Benchmark Suite", *IEEE Transactions on Image Processing*, Vol. 31, pp. 287–300, 2022.