Evaluation of Image Inpainting for Classification and Retrieval

Samuel Black Somayeh Keshavarz Richard Souvenir
Department of Computer and Information Sciences, Temple University
{sam.black,somayeh.keshavarz,souvenir}@temple.edu

Abstract

A common approach to censoring digital image content is masking the region(s) of interest with a solid color or pattern. In the case where the masked image will be used as input for classification or matching, the mask itself may impact the results. Recent work in image inpainting provides an alternative to masking by replacing the foreground with predicted background. In this paper, we perform an extensive evaluation of inpainting approaches to understand how well inpainted images can serve as proxies for the original in classification and retrieval. Results indicate that the metrics typically used to evaluate inpainting performance (e.g., reconstruction accuracy) do not necessarily correspond to improved classification or retrieval, especially in the case of person-shaped masked regions.

1. Introduction

Censoring content is a common pre-processing step with images containing sensitive information. For a variety of reasons, users may wish to hide portions of an image prior to uploading to a cloud-based service. For example, a special-purpose image search engine was developed to identify hotel rooms from images to aid in the fight against human trafficking [28]. In this scenario, users (i.e., law enforcement) obscure the victims in the images, particularly in the case of minors, often using off-the-shelf photo editing software. There are a variety of readily-available image processing tools that include "painting" over regions using a solid color or pattern, blurring or pixelating these areas, and, more recently, employing deep learning to predict the value of "missing" pixels (Figure 1).

Image inpainting is the process of recovering missing information from an image. In many cases, inpainting is used to restore images corrupted in some way. The focus of this paper is the case where inpainting is used extend the background of an image into a region where a foreground object may have been explicitly removed to conceal the presence of an object or person, usually for privacy preservation. We conduct an evaluation of image inpainting methods for image classification and retrieval. While previous surveys have evaluated the reconstruction performance of inpainting methods qualitatively [7, 11, 33], the goal of this work is to evaluate how well inpainted images can be used as proxies for the original in modern image classification and retrieval scenarios.

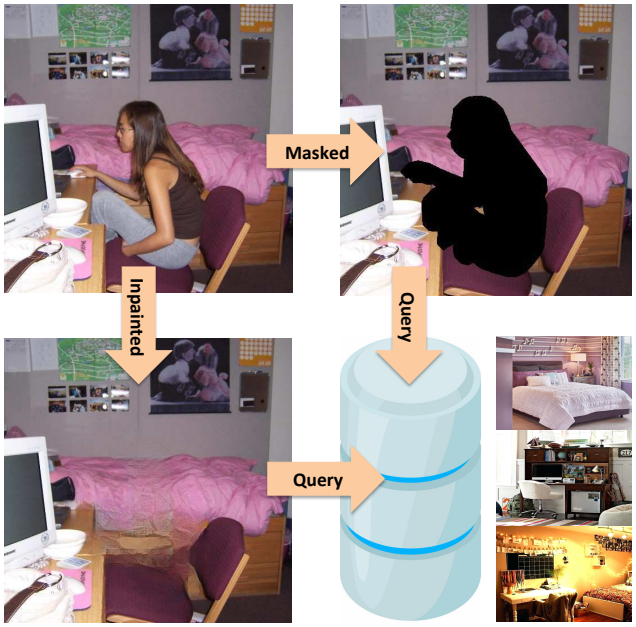

Figure 1. Image content can be censored by masking (top right) or inpainting (bottom left). In this paper, we evaluate how masking or inpainting images affects image classification and retrieval

In this paper, we (1) present a comprehensive evaluation of both recent and classic methods for image inpainting, (2) compare the accuracy of downstream tasks with inpainting reconstruction accuracy, and (3) investigate whether inpainting and/or retrieval performance are impacted by the shape of the masked region. Rather than applying inpainting for aesthetic purposes, we seek to to understand how these tools can be applied to privacy-preserving image search. To the best of our knowledge, this work represents the first quantitative evaluation of inpainting approaches for image classification and search.

2. Related Work

Inpainting methods can be broadly categorized as traditional or learning-based. In this section, we review methods from both classes, providing additional details for the methods used in the comparative evaluation.

2.1. Traditional Methods

The tools typically found in commercial photo-editing software are mainly based on traditional inpainting methods. These traditional methods compute the value of missing pixels using the values from neighboring (spatially and/or visually) image regions. This category of inpainting methods can be further subdivided into two subcategories: (1) diffusion and (2) patch-based methods that differ mainly in definition of "neighboring" image patches.

Diffusion Methods Diffusion methods extrapolate the values of adjacent image regions to the missing portion; approaches differ mainly in the extrapolation technique. In [4], the image isophotes are calculated by computing the direction of least change at each pixel in the known region. The values of the pixels in the missing region are computed such that the image Laplacian is constant in the isophote direction, which ensures a smooth transition from the known region to the inpainted area. A successor method analogizes the task with fluid dynamics and applies the Navier-Stokes equations to solve for the missing pixel values [3]. To improve isophote estimation, other methods minimize the total variation in the infilled region [5, 6]. Telea [29] introduced an approach based on the fast marching algorithm [24] where the pixels along the border of the missing regions are inpainted using weighted averages of the neighborhood pixel values such that both low and high frequency information is maintained; the image is infilled in one pass. Overall, diffusion approaches work reasonably well for small regions, but typically fail to reproduce textured regions, especially for large infills.

Patch-Based Methods Patch-based methods rely on other visually-similar, rather than spatially adjacent, known image regions as support for the infilling process. Patch-based methods take inspiration from an algorithm initially designed for texture synthesis [9]. At a high level, the process involves searching for (and replicating) the most similar image patch in the known region to an image patch at the border of the unknown region. Methods in this class vary in the size and shape of the patches, similarity measures, and search process. One method optimized the sampling strategy for finding similar patches [19]. PatchMatch [2] uses randomized search and, for efficiency, exploits the assumption that the best matches for neighboring unknown patches are likely nearby. Enhancements include improving how patches are blended together [8] and using structural sparsity to improve match quality [35]. While patch-based methods outperform diffusion methods for inpainting highly textured regions, they are limited to only replicating visual content from known regions of the given image.

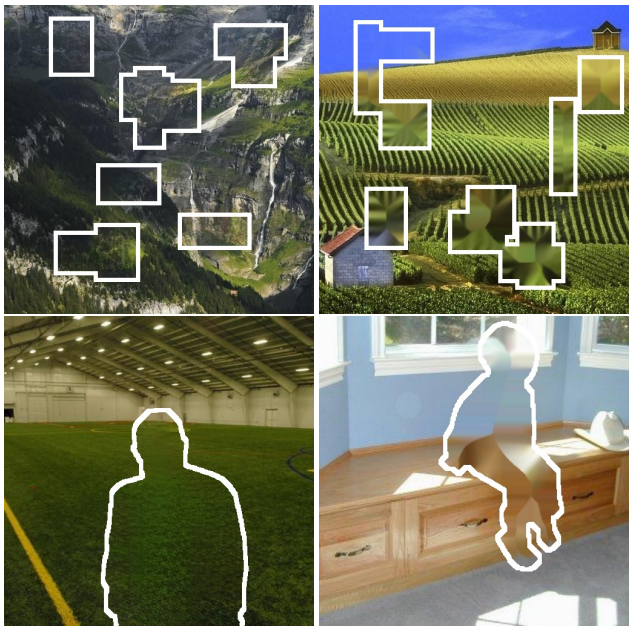

Figure 2. The outlined regions were infilled using learning-based approaches (left) and traditional methods (right).

2.2. Learning-Based Methods

Recently, learning-based methods have been developed to overcome the limitations of traditional methods, as shown in Figure 2. These approaches learn from large image collections to predict the values of missing pixels. The early learning-based methods relied on relatively simple multilayer perceptrons to inpaint images in regions with superimposed text by minimizing reconstruction loss [18, 34]. Modern approaches have involved using convolutional networks in conjunction with adversarial training to improve results with larger masked regions.

Context Encoder Pathak et al. [22] trained a convolutional model using an objective function that combines l_2 reconstruction and adversarial loss. This infilling network, called the Context Encoder, follows the typical encoder-decoder paradigm. The encoder module, based on the architecture of AlexNet, takes as input a masked image and outputs a feature vector to the decoder through a channel-wise fully connected layer. The decoder, through a series of deconvolutional layers, then outputs the infilled image. During training, a separate discriminator network is used to produce more realistic looking images. Many modern infilling methods incorporate this adversarial training strategy.

GAN-based Methods A number of methods derive from the architecture of the Context Encoder [22], with variations in loss functions, regularization, backbone architectures, and connectivity.

Globally and Locally Consistent Image Completion (GLCIC) [16] uses two discriminator networks, one for the entire image and another for image patches. Rather than outputting a flattened feature vector, the encoder reduces the input to quarter-sized feature maps. Dilated convolutional layers [37], where the kernels cover a larger image region without adding additional parameters, are also used within the network. This allows for the decoder to cover a larger region around each pixel without increasing the number of weights, which is beneficial when the size of the mask is large in proportion to the rest of the image.

The generator in Contextual Attention (CtxAttn) [38], is comprised of two subnetworks. The first is the coarse network, which is trained with l_1 reconstruction loss and outputs a rough prediction of the masked region. This output is then passed to a refinement network that produces the infilled image using two parallel encoder modules, one of which includes a contextual attention layer, which generates an attention score for each non-masked pixel that is based on the similarity to the unknown patches. These attention scores are incorporated during deconvolution. The refinement network is trained using reconstruction and adversarial loss with separate local and global discriminators. The adversarial objective combines Wasserstein loss [1] and a gradient penalty term [12] applied to missing pixels. Other inpainting methods follow a similar structure [27].

Generative Multi-column Convolutional Neural Networks (GMCNN) [31] uses a generator constructed of three parallel encoder-decoder modules, each with a different filter size. The output feature maps of each module are concatenated and passed to a shared two-layer convolution network, which produces the infilled image. In addition to reconstruction and adversarial losses, GMCNN also uses an implicit diversified Markov random fields (ID-MRF) loss, which uses patches extracted randomly from both the infilled and known regions. For each possible pair of generated and ground truth patches, a relative similarity metric is computed between the feature maps produced from specified layers of a pretrained VGG network [25].

EdgeConnect [21] is comprised of two networks. The edge generator takes as input the grey-scaled version of a masked image and an edge map to produce a prediction of the edges for the masked region of the image. The masked image and generated edge map are passed to the infilling network to produce the completed image. The generators are trained with separate discriminators. The loss function for the edge generator combines adversarial loss with feature-matching loss [30], which is computed using the difference between the feature maps of the predicted and ground truth edges generated by its discriminator. For the infilling network, its objective function includes l_1, adversarial, perceptual [10], and style loss [17]. Perceptual and style loss are similar to ID-MRF loss in that they are derived by comparing the feature maps of the infilled and ground truth images that are generated from a separate, pretrained network.

PIC-Net [39] differs from the other GAN-based methods by sampling an encoding vector from a learned probability distribution of the latent space in a manner similar to Conditional Variational Autoencoders [26]. This model generates multiple inpaintings for the same input. Additionally, PIC-Net uses two generators with shared weights during training. One takes as input a masked image and samples the encoding vector, which is then decoded to produce the output image. The other uses the masked portions of the image in conjunction with feature vectors from the first generator to reconstruct the groundtruth image. This second generator helps facilitate training and only the first is used during inference.

U-Net-based Methods Recent inpainting approaches are based on a U-Net [23]-like architecture, such as PConv [20] and DF-Net [14]. DF-Net uses fusion blocks in the decoding layers. Each fusion block is essentially a shallow convolutional network that produces an inpainted image by generating a "raw completion" and alpha composition map, using the decoder's feature maps and the scaled masked image as input. These are then blended together to produce the inpainted output. The network generates multiple images of varying resolutions by placing fusion blocks at different levels within the decoder (only the output of the top block is used at test time). Each output layer of the network is trained to minimize an objective function suited to the output resolution. l_1 loss is used to reconstruct large-scale features at all levels, while the higher levels incorporate a combination of perceptual [10] and style [17] losses to better achieve finer textures.

2.3. Summary

Previous surveys on image inpainting and completion describe and taxonomize the many different algorithms that have been developed for this task [11, 7, 33]. While these surveys often include qualitative evaluations, they seldom include a direct comparison between methods. Part of the challenge is that quantitatively evaluating infilled images is difficult due to the ill-posed nature of the task; there are multiple plausible outputs for a given masked image. Metrics that estimate reconstruction accuracy do not fully account for the visual aesthetics of the reconstruction. This work presents a quantitative evaluation of inpainting methods using the proxy measures of classification and retrieval performance on the generated images.

Figure 3. Sample images used in the experiments.

Model	Platform	Framework	Infill (s)	RAM (GB)
NS	CPU	OpenCV	.154	.065
FM	CPU	OpenCV	.138	.065
PIC-Net	GPU	PyTorch	.544	2.59
GMCNN	GPU	Tensorflow	.262	3.28
GLCIC	CPU	Torch	5.97	4.00
CtxAttn	GPU	Tensorflow	.610	2.62
EdgeCon	GPU	PyTorch	.207	3.64
DF-Net	GPU	PyTorch	.044	2.75

Table 1. Properties of the evaluated inpainting algorithms. Infill time (s) is the average per 512x512 image resolution with 40% pixels masked.

3. Evaluation

The experiments are meant to evaluate the efficacy of inpainted images for image classification and retrieval in order to understand how censoring images using methods designed for inpainting impacts the results of an image search. We evaluated each method on datasets commonly used for evaluating inpainting, scene classification, and image retrieval. Figure 3 shows example images from these datasets.

Evaluated Methods We selected a representative sample of inpainting methods. Serving as exemplars for traditional methods, we evaluated Navier-Stokes (NS) [3] and Fast Marching (FM) [29]. For learning-based methods, we considered Globally and Locally Consistent Image Completion (GLCIC) [16], Generative Image Inpainting with Contextual Attention (CtxAttn) [38], EdgeConnect (EdgeCon) [21], Generative Multi-column CNN (GMCNN) [31], Pluralistic Image Completion (PIC-Net) [39], and Deep Fusion Network for Image Completion (DF-Net) [14].

Implementation Our experiments were carried out on high-performance compute nodes with Intel Xeon CPUs, 96GB+ RAM, and Nvidia Tesla P100 GPUs. Where available, we use the implementation provided by the original authors. The learning-based approaches were pre-trained on the Places2 dataset [40], a large-scale collection of indoor and outdoor scenes. Two of the methods require square images as input, so the images were scaled such that the smaller dimension was 512 pixels and then center-cropped. PIC-Net worked best with smaller input, so the images were reduced to 256x256.

3.1. Performance

Table 1 presents the platform, framework, and computational costs (e.g., compute time, memory) for the methods used in the evaluation. Each of the methods is built on commonly used platforms for computer vision (e.g., OpenCV, Tensorflow). In general, the traditional methods are CPU-based, while the deep learning methods offer implementations that take advantage of the GPU.[1] On average, the deep learning methods take 2-5x time to infill, even with GPU acceleration. Without GPU acceleration, these methods can be an order of magnitude slower. The memory usage is negligible for the traditional approaches and dominated by the CNN weights for the learning based versions. For the learning-based versions, the timing does not include loading the pre-trained weights into memory.

3.2. Classifying Inpainted Images

The first experiment follows the most common evaluation protocol for inpainting, which uses irregular holes, or randomly positioned patches, as the missing image regions, as depicted in Figure 2 (top). For this experiment, we use Places2 [40], a widely-used dataset for scene recognition with over 2 million images from 365 different classes.

The query images include 5,000 randomly-selected images from the Places2 validation set. For each image, we generated 7 different masks occupying $10\%, 20\%, \ldots, 70\%$ of the image area. Each mask is generated by randomly placing small rectangles ($\sim$2.5% image area) until the coverage threshold is met. We provided each masked version to the set of inpainting algorithms. To evaluate the inpainting performance, we compared the infilled output to the original image and computed 3 reconstruction metrics: normalized root mean square error (NRMSE), peak signal to noise ratio (PSNR), and structural similarity index (SSIM) [32] (computed with window size = 11). For classification, we used two models, ResNet-18 [13] and DenseNet-161 [15], pre-trained to Places2. For the original, masked, and infilled images, we compute the top-5 classification accuracy for each network. For retrieval, our dataset consists of 100,000 images from the Places 2 training set. For the ResNet-18 model, we take the output after the last pooling layer normalized to the unit hypersphere as our image feature representation. We query the database using the features generated for the original, masked, and infilled images and sort the results based on cosine similarity to the input. To assess

[1] Due to library conflicts with the GPU version of GLCIC, we used the CPU version in testing.

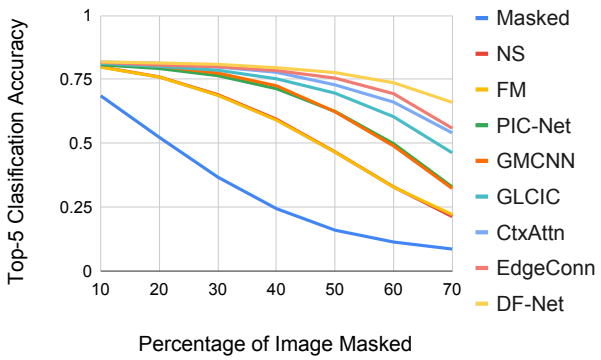

Figure 4. Top-5 classification accuracy vs. % masked.

retrieval performance, we consider the retrieval results on the original image to be the ground truth and compute the normalized cumulative gain @10 and @50 of the matches to the altered queries. The results are presented in Table 2.

Inpainting In general, the learning-based methods outperform the traditional methods and performance is consistent across all 3 metrics. As the masks increase in size, reconstruction performance drops across all methods. For example, SSIM for the NS method drops from .872 at 20% masking to .614 at 60% and for DF-Net from .878 to .605. In addition, the margin between the best and worst performing methods increases with the size of masked region, growing from 4.7% at 20% to 40.5% at 60%.

Classification Figure 4 shows that classification accuracy also degrades as a function of the masked area. At 20% masked, all of the methods achieve a top-5 classification accuracy within 6% of that of the unaltered original image. The differences between methods are more pronounced at the higher masking levels. At 60% masking, the best inpainting method, DF-Net, shows drops of 7% and 8% for ResNet and DenseNet, respectively, while GLCIC underperforms by 20% and 22%. By far, the worst option is simply masking; classification accuracy drops by 68% and 71% for the two networks. Figure 5 shows example classification results for masking, NS, PIC-Net, and DF-Net. For each image, the top 5 predicted labels are shown in the inset with the correct label (if present) checked.

Figure 6 shows a plot of the classification accuracy using ResNet-18 at 40% masking versus the reconstruction performance, as measured using the structural similarity index (SSIM). For the learning-based methods, there is a strong correlation ($r^2 = .737$) between classification accuracy and inpainting reconstruction. The traditional methods, NS and FM, do not follow this trend; both methods score highly using reconstruction metrics, but the resulting infilled images are not well classified.

Retrieval The retrieval results follow the general pattern of classification. DF-Net generates images which result in the highest performance for both classification and retrieval while the traditional methods are the worst, doing only slightly better than masking. However, while the relative ordering is mostly similar between the performance in the classification and retrieval tasks, the impact of masking and inpainting is much more evident in the retrieval results. Consider the traditional NS method. At the small masking level (20%) the DenseNet classification accuracy differs from the baseline by only 5%. However, nDCG_{10}, a weighted measure of the similarity between rankings, is only .524 suggesting a very different ordering of the top 10 returned matches. The same is true for the best-performing method at 20% masking, where the classification accuracy of the infilled images is the same as the original, yet nDCG_{10} is .837. This issue is exacerbated at the higher masking levels, with most of the methods exhibiting quite low retrieval performance when compared to the unaltered image. This suggests that the feature vectors used for image similarity searches can be sensitive to the alterations of infilling methods. A complicating factor is that this "irregular hole" pattern affects multiple regions of the image. In the next experiment, we consider a real-world use case where the masked regions tend to be contiguous.

3.3. Image Censoring

While the irregular hole mask pattern is commonly used for evaluating inpainting, it is uncommon in real-world settings. In this experiment, we consider the image editing task of censoring person-shaped regions from real-world images. This experiment is motivated by a special-purpose image search engine developed to identify hotel rooms from images to aid in the fight against human trafficking. Users obscure the victims in the images, often using basic masking. Hotels50k [28] is designed to evaluate this task. Hotels50k contains over one million images of hotel rooms from 50,000 different hotels. The test images include human-shaped masks designed to simulate censored image queries. Compared to the generic classification and retrieval tasks of the previous experiments, the goal is fine-grained hotel room identification.

The test set consists of 17,150 images. We evaluate the 'medium' setting where the masked regions occupy roughly 20% of the image area. The database of training images consists of 1,027,871 images. Our backbone model is a ResNet-50 embedding network trained on Hotels50k [36]. Following the experimental protocol in [28], we report the top-k accuracy for $k = 1, 10, 100$.

Table 3 shows the top-k classification accuracy on the Hotels50k dataset. Stylianou et al. [28] take a different approach to dealing with censored images. They assumed the query images will contain solid color masks and, dur-

		Inpainting			Classification (top-5)		Retrieval (ResNet)	
		NRMSE ↓	PSNR ↑	SSIM ↑	DenseNet	ResNet	$nDCG_{10}$	$nDCG_{50}$
Original (0% Masked)		–	–	–	.848	.821	–	–
20% Masked	Masked	–	–	–	.639	.523	.141	.233
	NS	.147	23.4	.872	.796	.759	.524	.652
	FM	.145	23.5	*.874*	.796	.759	.525	.653
	PIC-Net	.151	23.21	.805	.827	.792	.663	.775
	GMCNN	.152	23.1	.871	.834	.801	.748	.840
	GLCIC	.143	23.6	.852	.835	.800	.765	.854
	CtxAttn	.143	23.7	*.874*	*.841*	*.813*	.793	.874
	EdgeCon	*.126*	*24.8*	.868	.839	.806	*.804*	*.882*
	DF-Net	**.125**	**24.9**	**.877**	**.843**	**.821**	**.837**	**.905**
40% Masked	Masked	–	–	–	.359	.244	.022	.050
	NS	.226	19.9	.747	.664	.601	.175	.286
	FM	.215	20.0	*.750*	.661	.590	.174	.284
	PIC-Net	.2217	19.8	.660	.754	.713	.358	.495
	GMCNN	.253	18.5	.724	.771	.725	.392	.527
	GLCIC	.215	20.0	.712	.798	.752	.482	.616
	CtxAttn	.215	20.1	.741	.810	.778	.555	.682
	EdgeCon	**.187**	**21.3**	*.750*	*.815*	*.784*	*.601*	*.725*
	DF-Net	*.189*	*21.2*	**.751**	**.826**	**.795**	**.661**	**.772**
60% Masked	Masked	–	–	–	.164	.113	.005	.014
	NS	.284	17.5	*.614*	.420	.329	.034	.075
	FM	.284	17.5	**.617**	.418	.329	.034	.076
	PIC-Net	.308	16.9	.481	.534	.498	.096	.176
	GMCNN	.389	14.7	.532	.563	.489	.100	.174
	GLCIC	.285	17.4	.548	.644	.604	.181	.292
	CtxAttn	.289	17.5	.584	.689	.661	.269	.390
	EdgeCon	**.249**	**18.7**	.611	*.732*	*.694*	*.315*	*.449*
	DF-Net	*.256*	*18.5*	.609	**.765**	**.737**	**.420**	**.556**

Table 2. Reconstruction, classification, and retrieval results for the irregular holes infill experiment using the Places2 dataset. In each grouping, the first and second best results are in **bold** and *italics*, respectively.

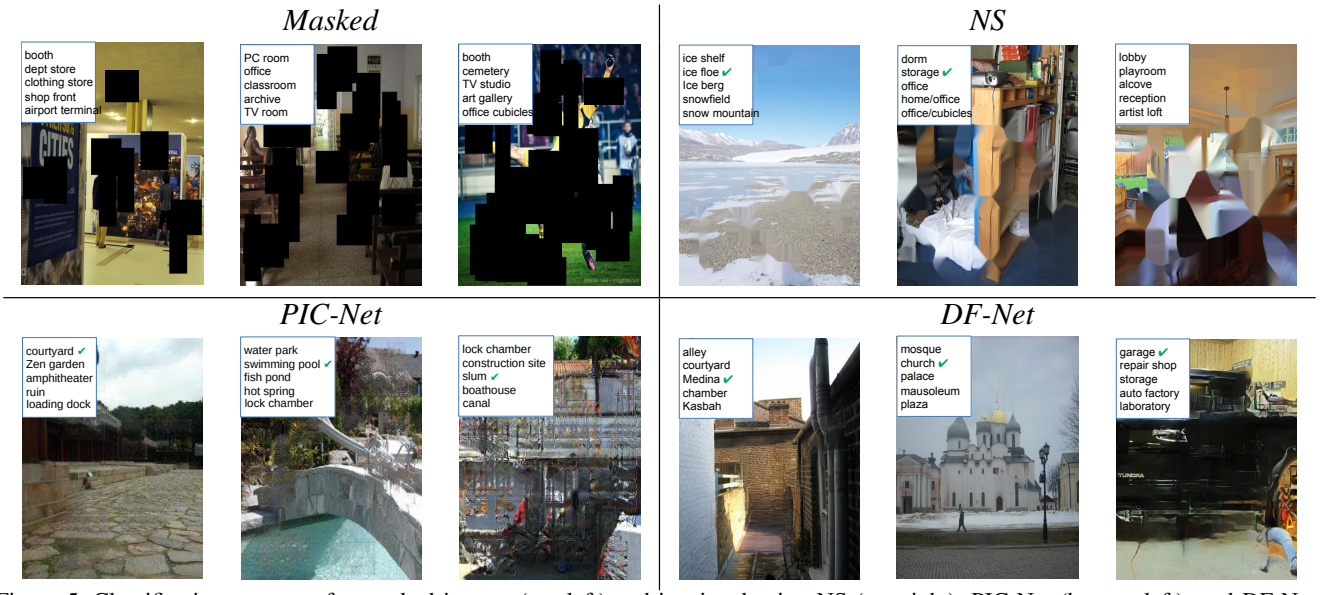

Figure 5. Classification accuracy for masked images (top left) and inpainted using NS (top right), PIC-Net (bottom left), and DF-Net (bottom right). For each image, the inset lists the top 5 predicted labels from DenseNet-161, with the correct label (if present) checked. For each group, from L-R, the images were 20%, 40%, and 60% masked.

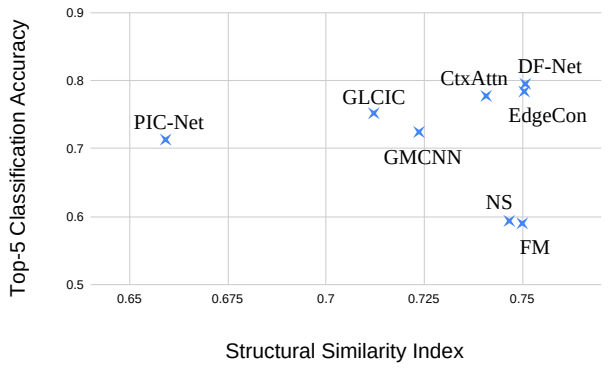

Figure 6. Plot of classification accuracy (ResNet-18) versus reconstruction performance for each of the methods (40% mask.)

	$k = 1$	$k = 10$	$k = 100$
[28]	0.059	0.141	0.299
Original	.164	.300	.495
Masked	.110	.225	.407
NS	.132	*.255*	*.443*
FM	.131	.252	.442
PIC-Net	.082	.178	.339
GMCNN	.127	.246	.430
GLCIC	.117	.231	.416
CtxAttn	.128	.248	.440
EdgeCon	*.133*	.250	.437
DF-Net	**.141**	**.270**	**.460**

Table 3. Top-k retrieval results using the Hotels50k dataset. The first and second best results are in **bold** and *italics*, respectively.

ing training, randomly apply masks to the training images as data augmentation. The infilled results are not directly comparable with [28] as both the backbone network and training regime differ. In this case, we observe that the top-performing infill approaches perform on par with the unaltered image and much better than the masked images. While the overall top performing method is the same as the previous experiment (DF-Net), the next best performing methods are the traditional methods, NS and FM. Moreover, unlike the synthetic setting of the previous experiment, we did not observe drastic decreases in retrieval performance using inpainted images for this task. Figure 7 shows the top 5 retrieval results for 3 queries using masking, a traditional method (NS), and learning-based method (DF-Net).

3.4. Discussion

This work provides an alternative quantitative framework for evaluating inpainting methods. Of the methods evaluated, DF-Net was consistently the best performing method across metrics for reconstruction, classification, and retrieval and for masked regions of various sizes. Visual inspection of the inpainted images aligns with the quantitative results; DF-Net produced the most natural-looking inpainted regions. The generated portions blended well with the rest of the image. For larger masks, the learning-based methods were prone to producing noticeable artifacts in the inpainted area, such as faint, repeated textures. However, DF-Net produced the least noticeable visual artifacts. It may be noteworthy that DF-Net was the only learning-based method evaluated that did not employ adversarial loss.

For each method, we used the settings recommended by the original authors. Some of the methods consistently produced visual artifacts in the output images. It was not clear if this was by design or an unfortunate combination of method parameters and our experimental setting. For instance, for GMCNN, the inpainted areas often did not blend well with the boundary regions and faint, vertical black lines were noticeable when the infill was lightly colored. PIC-Net appeared to produce repeating textural patterns across different images, especially when the inputs had large, contiguous masked regions. These noticeable artifacts help explain the performance of these two methods.

It is evident that the size, shape, and distribution of the masked region plays a role in the performance of downstream algorithm. While the masking had a large impact on retrieval performance for the synthetic experiment with multiple irregularly-shaped masked regions per image, the effect was not as pronounced for the real-world setting with the localized person-shaped masks. Also, the traditional methods, which were the worst performing for classification task were near the top for the real-world retrieval task. The best censoring approach for image search may not necessarily correspond with the most visually-appealing inpainting method; it may also depend on the problem domain.

4. Conclusion

This paper presented an evaluation of inpainting approaches as a pre-processing step for image classification and search. The results show the modern-learning based approaches outperform traditional methods even when the difference is not reflected in reconstruction metrics. For the types of queries where image censoring is necessary, inpainting provides an alternative to masking for privacy-preserving image search. In addition to the improved classification and retrieval performance, inpainting can conceal the fact that the query image was altered.

Acknowledgements

This research includes calculations carried out on Temple University's HPC resources supported in part by NSF Grant No. 1625061 and ARL contract W911NF-16-2-0189. Thanks to NSF REU Site students, Chiara Maalouf and Bella Hovis, who were involved in the early stages of this work and supported by NSF Grant No. 1757533.

Figure 7. The top 5 retrieval result for masked image, inpainted image using NS method and inpainted image using DF-Net method. Image from correct hotel instance are highlighted in green.

References

[1] M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein GAN. *arXiv preprint arXiv:1701.07875*, 2017. 3

[2] C. Barnes, E. Shechtman, A. Finkelstein, and D. B. Goldman. Patchmatch: A randomized correspondence algorithm

for structural image editing. *ACM Transactions on Graphics*, 2009. 2

[3] M. Bertalmio, A. L. Bertozzi, and G. Sapiro. Navier-stokes, fluid dynamics, and image and video inpainting. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2001. 2, 4

[4] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester. Image inpainting. In *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*, pages 417–424, 2000. 2

[5] T. Chan and J. Shen. Mathematical models for local deterministic inpainting. *UCLA Computational and Applied Mathematics Reports*, 2000. 2

[6] T. F. Chan and J. Shen. Nontexture inpainting by curvature-driven diffusions. *Journal of Visual Communication and Image Representation*, pages 436–449, 2001. 2

[7] J. K. Chhabra and M. V. Birchha. Detailed survey on exemplar based image inpainting techniques. *International Journal of Computer Science and Information Technologies*, 5(5), 2014. 1, 3

[8] S. Darabi, E. Shechtman, C. Barnes, D. B. Goldman, and P. Sen. Image melding: Combining inconsistent images using patch-based synthesis. *ACM Transactions on Graphics*, pages 82–1, 2012. 2

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

[10] L. A. Gatys, A. S. Ecker, and M. Bethge. Image style transfer using convolutional neural networks. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 2414–2423, 2016. 3

[11] C. Guillemot and O. Le Meur. Image inpainting: Overview and recent advances. *IEEE Signal Processing Magazine*, 2013. 1, 3

[12] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville. Improved training of wasserstein GANs. In *Advances in Neural Information Processing Systems*, pages 5767–5777, 2017. 3

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

[14] X. Hong, P. Xiong, R. Ji, and H. Fan. Deep fusion network for image completion. In *ACM Multimedia*, 2019. 3, 4

[15] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger. Densely connected convolutional networks. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 4700–4708, 2017. 4

[16] S. Iizuka, E. Simo-Serra, and H. Ishikawa. Globally and locally consistent image completion. *ACM Transactions on Graphics*, page 107, 2017. 3, 4

[17] J. Johnson, A. Alahi, and L. Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *Proc. European Conference on Computer Vision*, pages 694–711, 2016. 3

[18] R. Köhler, C. Schuler, B. Schölkopf, and S. Harmeling. Mask-specific inpainting with deep neural networks. In *German Conference on Pattern Recognition*, pages 523–534, 2014. 2

[19] L. Liang, C. Liu, Y.-Q. Xu, B. Guo, and H.-Y. Shum. Real-time texture synthesis by patch-based sampling. *ACM Transactions on Graphics*, pages 127–150, 2001. 2

[20] G. Liu, F. A. Reda, K. J. Shih, T.-C. Wang, A. Tao, and B. Catanzaro. Image inpainting for irregular holes using partial convolutions. In *Proc. European Conference on Computer Vision*, 2018. 3

[21] K. Nazeri, E. Ng, T. Joseph, F. Qureshi, and M. Ebrahimi. Edgeconnect: Generative image inpainting with adversarial edge learning. *IEEE International Conference on Computer Vision Workshop*, 2019. 3, 4

[22] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros. Context encoders: Feature learning by inpainting. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 2536–2544, 2016. 2, 3

[23] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 234–241, 2015. 3

[24] J. A. Sethian. A fast marching level set method for monotonically advancing fronts. *Proceedings of the National Academy of Sciences*, pages 1591–1595, 1996. 2

[25] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 3

[26] K. Sohn, H. Lee, and X. Yan. Learning structured output representation using deep conditional generative models. In *Advances in Neural Information Processing Systems*, pages 3483–3491, 2015. 3

[27] Y. Song, C. Yang, Z. Lin, X. Liu, Q. Huang, H. Li, and C.-C. Jay Kuo. Contextual-based image inpainting: Infer, match, and translate. In *Proc. European Conference on Computer Vision*, pages 3–19, 2018. 3

[28] A. Stylianou, H. Xuan, M. Shende, J. Brandt, R. Souvenir, and R. Pless. Hotels-50k: A global hotel recognition dataset. In *Proc. National Conference on Artificial Intelligence*, 2019. 1, 5, 7

[29] A. Telea. An image inpainting technique based on the fast marching method. *Journal of Graphics Tools*, pages 23–34, 2004. 2, 4

[30] T.-C. Wang, M.-Y. Liu, J.-Y. Zhu, A. Tao, J. Kautz, and B. Catanzaro. High-resolution image synthesis and semantic manipulation with conditional GANs. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 8798–8807, 2018. 3

[31] Y. Wang, X. Tao, X. Qi, X. Shen, and J. Jia. Image inpainting via generative multi-column convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 331–340, 2018. 3, 4

[32] Z. Wang, A. C. Bovik, H. R. Sheikh, E. P. Simoncelli, et al. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, pages 600–612, 2004. 4

[33] X. Wu, K. Xu, and P. Hall. A survey of image synthesis and editing with generative adversarial networks. *Tsinghua Science and Technology*, pages 660–674, 2017. 1, 3

[34] J. Xie, L. Xu, and E. Chen. Image denoising and inpainting with deep neural networks. In *Advances in Neural Information Processing Systems*, pages 341–349, 2012. 2

[35] Z. Xu and J. Sun. Image inpainting by patch propagation using patch sparsity. *IEEE Transactions on Image Processing*, pages 1153–1165, 2010. 2

[36] H. Xuan, A. Stylianou, and R. Pless. Improved embeddings with easy positive triplet mining. *arXiv preprint arXiv*, 2019. 5

[37] F. Yu and V. Koltun. Multi-scale context aggregation by dilated convolutions. *arXiv preprint arXiv:1511.07122*, 2015. 3

[38] J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu, and T. S. Huang. Generative image inpainting with contextual attention. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 5505–5514, 2018. 3, 4

[39] C. Zheng, T.-J. Cham, and J. Cai. Pluralistic image completion. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 1438–1447, 2019. 3, 4

[40] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, and A. Torralba. Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017. 4