A Review on Image Inpainting Techniques and Datasets

David Josué Barrientos Rojas
Escola Politécnica de Pernambuco
Universidade de Pernambuco
Recife, Brazil
Email: djbr@ecomp.poli.br

Bruno José Torres Fernandes
Escola Politécnica de Pernambuco
Universidade de Pernambuco
Recife, Brazil
Email: bjtf@ecomp.poli.br

Sergio Murilo Maciel Fernandes
Escola Politécnica de Pernambuco
Universidade de Pernambuco
Recife, Brazil
Email: smurilo@ecomp.poli.br

***Abstract*—Image inpainting is a process that allows filling in target regions with alternative contents by estimating the suitable information from auxiliary data, either from surrounding areas or external sources. Digital image inpainting techniques are classified in traditional techniques and Deep Learning techniques. Traditional techniques are able to produce accurate high-quality results when the missing areas are small, however none of them are able to generate novel objects not found in the source image neither to produce semantically consistent results. Deep Learning techniques have greatly improved the quality on image inpainting delivering promising results by generating semantic hole filling and novel objects not found in the original image. However, there is still a lot of room for improvement, specially on arbitrary image sizes, arbitrary masks, high resolution texture synthesis, reduction of computation resources and reduction of training time. This work classifies and orders chronologically the most prominent techniques, providing an overall explanation on its operation. It presents, as well, the most used datasets and evaluation metrics across all the works reviewed.**

I. Introduction

Inpainting started to be applied, as early as the Renaissance [1], in the restoration of damaged painted images, due to aging, scratching or other factors. Physical inpainting is a very time consuming process as it's manually carried by skilled art conservators or restorers to reconstruct valuable paintings and conserve its cultural heritage, using any methods that prove effective in keeping it as close to its original condition. With the arrival of photography and film, the need to reconstruct media extended, giving birth to digital inpainting. This type of inpainting addresses the same issues as physical inpainting, plus the ones added by digital image corruption. Digital inpainting is a process that focus on the application of sophisticated algorithms to reconstruct digital image data. Currently it is used for many applications such as, image editing, coding, restoration, removal or replacement of objects, film and television special effect production, robot vision, etc.

This paper presents and discusses different types of inpainting techniques. Section 2 presents the review on image inpainting, divided by categories and in chronological order. Sections 3 and 4 presents the most used datasets and quantitative metrics across the reviewed works. Section 5 discuss the current trends and challenges on the area. And finally, Section 6 presents the conclusions.

Fig. 1. Digital Inpainting example using Context Encoders [2]

II. Inpainting Techniques

Existing image inpainting techniques can be divided into two different groups: traditional and deep learning-based methods.

A. Traditional Methods

We classify the traditional methods for image inpainting into three groups: diffusion-based techniques, patch-based techniques and convolution filter-based techniques.

1) Diffusion-based techniques: these techniques fill the missing region by propagating the local image appearance around the hole.

In 2000, Bertalmio et al. [3] introduced an automatic image inpainting technique based on partial differential equations (PDE). The algorithm uses the concept of isophotes: curves of constant light intensity on a surface. This technique smoothly propagates the information from the surrounding areas in the isophotes direction. If the grid is set to zero, the inpainting technique naively resembles a third order equation. If the isophote directions are reconstructed, it is possible to obtain the gradient direction, allowing to reconstruct the gray levels.

Inspired by this, in 2001 Chan and Shen proposed in [4] the Total Variational (TV) Inpainting model. This model uses the Euler-Lagrange equation along with an isotropic diffusion on isophotes. Isotropic diffusion allows to completely maintain the isophotes directions across all the structure. Later that year they proposed the Curvature Driven Diffusion model [5]. Chan and Shen found that, as the isophotes flatten, the connectivity principle, basic to implement TV, is at risk. So, the model was updated to include on the algorithm the isophotes geometric information, required to define the strength of the diffusion

process. This change not only addressed the issue but enabled inpainting over larger areas.

In 2003 Grossauer et al. [6] proposed another PDE method based on the Ginzburg-Landau equation. This allows inpainting to be directly applied to restore higher dimensional data, improving sparsely sampled volumetric data and to fill in fragmentary surfaces, an application of importance in architectural heritage preservation. This model presented an increase in performance, compared to the previous ones, and the ability to retain the original image symmetry.

All of the above mentioned algorithms are very time consuming. In 2004 Telea [7] proposed a fast-marching method (FMM), which is a faster and simpler way to implement a PDE-based algorithm. His method works by propagating an image smoothness estimator along the image gradient. Image smoothness is estimated as a weighted average over a known image neighborhood of the pixel to inpaint. The missing regions are treated as level sets and the FMM described in [8] is used to propagate the image information.

Diffusion-based inpainting algorithms produce accurate results when the missing areas are small. As the size of the missing area increases, the diffusion process adds blur to the resulting image. On big areas, the algorithm takes too much time and is unable to generate good outcomes.

2) Patch-based techniques: these techniques fill the missing region by sampling texture patches from the existing regions of the image and pasting them in the hole region.

In 1999 Efros and Leung [9] proposed a non-parametric texture synthesis model, based on Markov random field for texture synthesis. The process grows a new image outward from an initial seed, one pixel at a time. First, a neighborhood around a damaged pixel is selected, then all known regions of the image are searched to find the most similar one to the selected neighborhood. Finally, the central pixel in found neighborhood is copied to the damaged pixel.

In 2000, Wei and Levoy implemented a fast algorithm for the search step in non-parametric synthesis [10]. This technique uses tree-structured vector quantization (TSVQ), creating two image pyramids: one for the sample texture and one for the output image. Searching in multi-resolution pyramids decreases computation time, dramatically accelerating the synthesis process, from days of computation to a few seconds.

In 2001, Ashikhmin proposed an algorithm for synthesizing natural quasi-repeating textures consisting of familiar elements [11]. This algorithm works diminishing the search space of each pixel to only four candidates based on four previously synthesized neighbor pixels. Since there are many fewer options to choose from, there is no need to make a fine discrimination and the neighborhood sufficient for this task can be significantly smaller. A candidate of a neighbor pixel is its source in sample texture, shifted into the new pixel.

These three methods are often referred as pixel-based synthesis; they provide good quality results in reasonable time. However, in 2001 Efros and Freeman proposed the first patch-based texture synthesis algorithms: image quilting [12]. Image quilting works by stitching together small patches of existing images, based on local image information. It traverses the image, searching the input texture for a set of blocks that satisfy the overlap constraints and randomly pick selecting one. Then, it computes the minimum cost path using the surface error. Finally, it pastes the block onto the texture for a complete texture transfer. Despite its simplicity, this technique produces results that equal or better than the previously listed, but with improved stability and at a fraction of the computational cost. Quilting is used as a fast and very simple texture synthesis algorithm.

In 2001, Liang et al. proposed another patch-based algorithm in [13]. This technique approximates the nearest neighbors search for a new patch that fits the overlapping boundary regions of previously synthesized patches, avoiding mismatching features across patch boundaries using color blending for handling the overlapping boundaries between patches.

In 2003, Kwatra et al. proposed a different approach to the problem of finding a seamless patch boundary in graph cut texture synthesis [14]. In contrast to other techniques, the size of the patch is not chosen previously, but instead a graph cut technique is used to determine the optimal patch region for any given offset between the input and output texture. Later this year, Criminisi et al. [15] proposed an efficient algorithm that uses the advantages of both patch-based and diffusion-based techniques. The algorithm uses [9] model as a base, replicating both texture and structure for structure propagation. The algorithm works by limiting the search space and copying the neighborhood instead of the central pixel. A priority for the pixels is introduced, where known pixels or those near an edge have higher priority.

In 2005, Cheng et al. demonstrated that Criminisi's priority function may become unreliable after several iterations. Thus, Cheng proposed a new generic priority function that integrates the structure and the texture information to facilitate the image reconstruction [16].

In 2007, Wexler et al. presented a new framework [17] for the completion of missing information based on local structures. It models the task of completion as a global optimization problem with a well-defined objective function. This iterative multi-scale optimization algorithm repeatedly searches for nearest neighbor patches for all hole pixels in parallel. In 2008, Simakov et al. enhanced Wexler et al. model [18] using a global optimization-based method that can obtain more consistent fills by optimizing the summarizing/retargeting using the bi-directional similarity measure. This reduces the coherence objective function of Wexler et al. and obtains a similar optimization algorithm.

In 2009, Wexler and Simakov techniques were accelerated by Barnes et al. [19] using a fast-approximate nearest neighbor patch search algorithm. "PatchMatch" finds good patch matches via random sampling, using the nearest-neighbor field (NNF) algorithm for computing patch correspondences. The NNF is filled with either random offsets or some prior information. Next, an iterative update process is applied in

which good patch offsets are propagated to adjacent pixels, followed by random search in the neighborhood of the best offset found, and using that natural coherence, propagating those matches quickly to surrounding areas.

In 2012, Darabi et al. [20] using a patch-based optimization demonstrated improved image completion by integrating the image gradients into the patch representation and replaced the usual color averaging with a screened Poisson equation solver.

Unlike diffusion-based techniques, patch-based techniques provide better performance in filling large hole regions. However, they depend on low-level features, filling the hole regions regardless of the visual semantics and being ineffective to inpaint complicated structures resulting in images with poor visual quality and the inability to generate novel objects not found in the source image.

3) Convolution Filter based techniques: these techniques fill the missing region by convolving the neighborhood of the damaged pixels with a proper kernel.

In 2001, Oliveira et al. introduces a fast image inpainting algorithm based on the convolution operation [21]. The algorithm repeatedly applies a convolution between the region to be inpainted with a diffusion kernel. This is repeated until none of the pixels belonging to the domain had their values changed by more than a certain defined threshold during the previous iteration.

In 2008, Hadhoud et al. introduced in [22] a modification of Oliveira's algorithm. Hadhoud et al. reduced the time of inpainting and increased the quality of the results modifying the convolution stage. In the filling step, this method applies convolution within the inpainting region and an averaging filter that has a zero weight at the bottom right corner instead of the center.

In 2010, Noori further enhanced the method in [23] by using an adaptive kernel. This algorithm is adaptive and uses gradient to calculate weights of a convolving mask at each position. A predefined function is introduced to assign low weights to a pixel near an edge. The algorithm is fast, iterative, simple to implement.

Convolution filter-based algorithms, provide an easy, fast and precise way to inpaint small regions, however suffered from the same problem, as the region to be inpainted increases blurring is added to the result. These filters too cannot generate novel objects not found in the source image.

TABLE I
TRADITIONAL TECHNIQUES

Category	Method	Feature
Diffusion-based techniques	Bertalmio et al. 2000 [3]	Isophote propagation
	Chan and Shen 2001 [4]	Total Variational (TV) model
	Chan and Shen 2001 [5]	Curvature Driven Diffusion (CDD) model
	Grossauer et al. 2003 [6]	Ginzburg-Landau equation based
	Telea 2004 [7]	Fast marching method (FMM) based
Patch-based techniques	Efros and Leung 1999 [9]	Markov random field based
	Wei and Levoy 2000 [10]	Tree-structured vector quantization (TSVQ)
	Ashikhmin 2001 [11]	Natural quasi-repeating textures synthesis
	Efros and Freeman 2001 [12]	Image quilting
	Liang et al. 2001 [13]	Color blending to handle overlapping boundaries
	Kwatra et al. 2003 [14]	Graph cut technique to determine optimal patch region
	Criminisi et al. 2003 [15]	Patch-based and diffusion-based combination. Pixel priority
	Cheng et al. 2005 [16]	Structure and the texture information in priority function
	Wexler et al. 2007 [17]	Iterative multiscale optimization algorithm
	Simakov et al. 2008 [18]	Bi-directional similarity measure
	Wexler and Simakov 2009 [19]	Random sampling using the nearest-neighbor field (NNF)
	Darabi et al. 2012 [20]	Image gradients integration and screened Poisson equation solver
Convolution Filter based techniques	Oliveira et al. 2001 [21]	Convolution operation model
	Hadhoud et al. 2008 [22]	Convolution stage modification for increased quality
	Noori 2010 [23]	Adaptative kernels

B. Deep Learning based Methods

With the prominent recent advances on deep learning, specially Convolution Neural Networks for image recognition, image inpainting received the tool that was missing.

The introduction of the Deep Learning (DL) technique by LeCun [24] allowed to in-paint images by utilizing supervised image classification. The idea is that each image has a specific label and a convolutional neural network (CNN) learns to recognize the mapping between images and their labels [25] [26].

In 2012, Xie et al. proposed in [27] a new training plan for Denoising Auto-encoder (DA) that was able to denoise images and blindly inpaint images on the same framework. DA is a two-layer neural network that tries to reconstruct the original input from a noisy version of it.

In 2013, Eigen proposed in [28] a three-layer CNN model to remove rain drops and dirt. It demonstrated the ability of CNN to blindly inpaint images where the exact shape of the missing region might be uncertain, making it completely applicable in real world challenges.

In 2014, Xu et al. presented a robust model for deconvolution in [29]. Deconvolution is an operation to recover an image that is degraded by a convolution process. This model is introduced as a deconvolution convolutional neural network (DCNN), that is completely based on separable kernels for robust deconvolution against artifacts, yielding decent performance on non-blind image deconvolution.

The same year, Köhler explored how the shape of the

mask affects performance in [30]. Concluding that by just including the shape of the mask into the input layer enhances the inpainting results. The mask specific training makes the solution more specific, with the limitation that a trained network will not perform optimally if trained on the wrong mask.

In 2015, Ren et al. found that CNN and sparse auto-encoder are inherently with translation invariant operators. This highly reduces the DL approaches performance when the task requires translation-variant interpolation (TVI). In order to fix this they proposed in [31] a new model: Shepard Convolutional Neural Networks (ShCNN). Using Shepard interpolation and a specific structure, it efficiently executes end-to-end trainable TVI operators in the network.

Although these DL methods are good, they still produce some blurriness on the image and are unable to inpaint on complex scenes due to a lack of semantic understanding of the image.

In 2016 Pathak et al. based on the work of Goodfellow on Generative Adversarial Networks (GAN) in [32], introduced Context Encoders employing a conditional GAN (CE) [2]. CEs are encoder-decoder networks that can predict the missing parts, where an adversarial loss is adopted in training to produce much sharper results. CEs need to do both: understand the content of the entire image, as well as produce a plausible hypothesis for the missing part(s). This model achieved exceptional results as it is able to capture image semantics and global structure.

In 2017, Iizuka et al. further improved this model in [33] by introducing an extra discriminator to ensure local image coherency. The global discriminator assesses if completed image is coherent as a whole, while the local discriminator focuses on a small area centered at the generated region to enforce the local consistency. In addition, it uses dilated convolutions [34] to expand the receptive field and Poisson blending [35] to refine the image.

Convolutional neural networks show to be ineffective in explicitly borrowing or copying information from distant spatial locations, something that patch-based techniques predominate in. Motivated by this, in 2018, Yu et al. proposed in [36] a deep generative model-based approach which can not only synthesize novel image structures but also explicitly utilize surrounding image features as references during network training to make better predictions. This technique uses a coarse-to-fine network with a contextual attention module (CAM), which can learn where to borrow the background features for the hole region by computing the cosine similarity between the background and foreground feature patches. The model consists of two stacked generative networks (coarse and refinement networks) to generate a first image, then the refinement network using CAM refines this intermediate image to produce the final inpainting result. This method achieves remarkable performance however, it requires considerable computational resources.

Meanwhile, Yan et al. proposed in [37] a new network named Shift-Net, which provides image inpainting via Deep Feature Rearrangement. This network has a special shift-connection layer added to the U-Net architecture [38]. The encoder feature of the known region is shifted to serve as an estimation of the missing parts. A guidance loss is introduced to enhance the explicit relation between the encoded feature in the known region and decoded feature in the missing region. By exploiting such relation, the shift operation can be efficiently performed and is effective in improving inpainting performance.

Liu et al. proposed the use of partial convolutions for inpainting in [39]. This model uses an UNet-like architecture, replacing all convolutional layers with partial convolutional layers and using nearest neighbor up-sampling in the decoding stage. This model is able to handle masks (filling regions) of any shape, size, location, or distance from the image borders without losing performance as holes increase in size.

In 2019, Zeng et al. proposed in [40] a deep generative model built upon U-Net: The Pyramid-context Encoder Network (PEN-Net). This model uses a pyramid-context encoder, which progressively learns region affinity by attention from a high-level semantic feature map and transfers the learned attention to the previous low-level feature map. As the missing content can be filled by attention transfer from deep to shallow in a pyramid fashion, both visual and semantic coherence for image inpainting can be ensured.

Sagong et al. proposed in [41] the PEPSI (Parallel Extended-decoder Path for Semantic Inpainting) model. This model enhances Yu's model, exchanging the two-stage process for feature encoding with a single shared encoding network and a parallel decoding network with coarse and inpainting paths, reducing the number of convolution operations by half. The coarse path produces a preliminary inpainting result with which the encoding network is trained to predict features for the CAM. At the same time, the inpainting path creates a higher-quality inpainting result using refined features reconstructed by the CAM.

In 2020, Jiang et al. enhanced Iizuka's model in [42] by adding a skip-connection in the generator to improve the prediction power of the model and the Wasserstein GAN loss [43], to ensure the stability of the training process.

Recently Li et al. [44] proposed a Recurrent Feature Reasoning (RFR) network, constructed using a RFR module and a Knowledge Consistent Attention (KCA) module. RFR networks works by solving the easier parts first and the using that information to solve the difficult parts, for inpainting it infers the hole boundaries of the convolutional feature maps and then uses them as clues for further inference. Exploiting the correlation between adjacent pixels and strengthens the constraints for estimating deeper pixels. The Knowledge Consistent Attention (KCA) module is used to synthesize features of higher quality by searching in the background for possible textures to replace the textures the hole.

DL techniques have greatly improved the quality on image inpainting, delivering promising results by generating semantic hole filling and novel objects that are not in the original image.

TABLE II
DEEP LEARNING TECHNIQUES

Method	Feature
Xie et al. 2012 [27]	Denoising Auto-encoder
Eigen 2013 [28]	Three-layer CNN
Xu et al. 2014 [29]	Deconvolution convolutional neural network (DCNN)
Köhler 2014 [30]	Included mask shape into the input layer
Ren et al. 2015 [31]	Shepard Convolutional Neural Networks (ShCNN)
Pathak et al. 2016 [2]	Context Encoders (CE)
Iizuka et al. 2017 [33]	CE with Local and Global discriminator. Dilated Convolution. Poisson blending
Yu et al. 2018 [36]	Two stacked generative networks (coarse and refinement networks) and a contextual attention module (CAM)
Yan et al. 2018 [37]	Shift-Net
Liu et al. 2018 [39]	Partial convolution
Zeng et al. 2019 [40]	The Pyramid-context Encoder Network (PEN-Net)
Sagong et al. 2019 [41]	PEPSI (Parallel Extended-decoder Path for Semantic Inpainting)
Jiang et al. 2020 [42]	Skip-connection. Wasserstein GAN loss
Li et al. 2020 [44]	Recurrent Feature Reasoning network with a Knowledge Consistent Attention module

III. DATASETS

Since DL techniques work by learning to recognize the mapping between images and their labels, a dataset is required. Many datasets have been used to solve some application-specific problems, however, as inpainting advanced, models started to be trained not only with one dataset, but with a dataset for each of the most common applications required.

This section presents the most used datasets across all the works to train inpainting models.

- Places2 [45]: A repository of ten million scene photographs, labeled with scene semantic categories, comprising a large and diverse list of the types of environments encountered in the world.
- CelebA [46]: A repository of ten thousand identities, each of which has twenty images. Each image is annotated with forty face attributes and five key points by a professional labeling company.
- Celeb HQ [47]: A high-quality version of the CelebA dataset, consisting of 30,000 images in 1024 by 1024 resolution.
- DTD [48]: The Describable Textures Dataset (DTD) is a repository of 5,640 real-world texture images annotated with one or more adjectives selected in a vocabulary of 47 English words.
- ImageNet [49]: The Imagenet large scale visual recognition challenge for 2012 (ILSVRC) is a repository that contains 1,000 object categories. It contains around 1,300,000 images for training, 50,000 images for validation and 100,000 images for testing.

DTD is used to train the model to reconstruct different textures and attributes, required to extend the textures present on the original image. CelebA and CelebHQ are used to train the ability of the model to reconstruct semantically correct complex structures. It works perfectly to test the model in novel object reconstruction and global consistency. Places2 trains the model to reconstruct any kind of background present on the original image, as it contains a huge list of different environments all around the world. And finally, ImageNet trains the model to reconstruct all kind of objects giving a great generalization ability to the model.

Previously, not all the works used these models, as each dataset only addressed one specific application, but as inpainting evolved it has become a general strategy to train the model on all. The two most used datasets are Places2 and CelebHQ.

Fig. 2. Places2 image samples in [45]

Fig. 3. CelebA image samples in [46].

Fig. 4. CelebHQ image samples in [47].

Fig. 5. DTD image samples in [48]

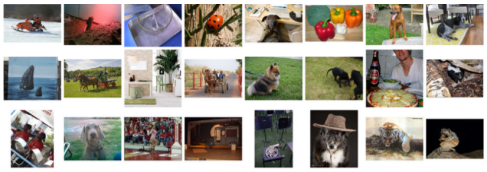

Fig. 6. ImageNet image samples in [49]

IV. EVALUATION METRICS

For a quantitative comparison, most projects use two different metrics to calculate the quality of the image reconstructions: Peak signal-to-noise ratio (PSNR) and Structural Similarity (SSIM).

- PSNR measures the relation between the maximum energy of a signal and the noise that affects its accurate representation. For inpainting it is a relation between the original image (Ground Truth) and the inpainted result. The higher the PSNR, the better the quality of the resulting image.
  To compute the PSNR of a test image (g) using a reference image (f), it is required to calculate first the mean-squared error using the following equation:

$$MSE(f,g) = \frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}(f_{ij} - g_{ij})^2 \tag{1}$$

  where, M and N are the number of rows and columns of the images.
  Then we can proceed to compute the PSNR using the following equation:

$$PSNR(f,g) = 10\log_{10}\left(\frac{R^2}{MSE(f,g)}\right) \tag{2}$$

  where, R is the maximum fluctuation in the input image data type.
- SSIM is a technique to measure the similarity between two images. It compares an image regarded as of perfect quality with another image. For inpainting it compares the original image with the resulting inpainted image. Regions with small local SSIM value correspond to areas where the resulting image differs from the reference image. Large values of local SSIM correspond to uniform regions of the reference image.
  The SSIM of a test image (g) using a reference image (f), can be calculated using the following equation:

$$SSIM(f,g) = l(f,g)c(f,g)s(f,g) \tag{3}$$

$$l(f,g) = \frac{2\mu_f\mu_g + C_1}{\mu_f^2 + \mu_g^2 + C_1} \tag{4}$$

$$c(f,g) = \frac{2\sigma_f\sigma_g + C_2}{\sigma_f^2 + \sigma_g^2 + C_2} \tag{5}$$

$$s(f,g) = \frac{\sigma_{fg} + C_3}{\sigma_f\sigma_g + C_3} \tag{6}$$

  Equation (4) represents the luminance comparison function. Equation (5) represents the contrast comparison function. Equation (6) represents the structure comparison function.

These quantitative comparisons can be applied for the local and global regions.

V. Trends and Challenges

Inpainting techniques have greatly evolved to the point where it is possible to generate novel object and semantically consistent images.

We can identify three different trends on the DL approaches: those works using Context Encoders (CE), as described in [2], [33], [40], [42]; those works using Contextual Attention Module (CAM), as showed in [36], [41]; and finally, those using U-Net architectures, as presented in [37], [39].

TABLE III
Results of PSNR and SSIM using square masks on CELEBA-HQ dataset found in [41].

	PSNR		SSIM
	Local	Global	
Pathak et al. 2016 [2]	17.7	23.7	0.872
Iizuka et al. 2017 [33]	19.4	25.0	0.896
Yu et al. 2018 [36]	19.0	24.9	0.898
Sagong et al. 2019 [41]	19.5	25.6	0.901

TABLE IV
Results of PSNR and SSIM using free form masks on CELEBA-HQ dataset found in [41].

	PSNR		SSIM
	Local	Global	
Pathak et al. 2016 [2]	9.7	16.3	0.794
Iizuka et al. 2017 [33]	15.1	21.5	0.843
Yu et al. 2018 [36]	12.4	18.9	0.798
Sagong et al. 2019 [41]	22.0	28.6	0.929

TABLE V
Results of PSNR and SSIM values on the Places2 dataset. No results found for the metric is represented with a −.

	PSNR	SSIM
Yu et al. 2018 [36]	18.91	-
Liu et al. 2018 [39]	33.75	0.946
Zeng et al. 2019 [40]	-	0.780
Sagong et al. 2019 [41]	21.2	0.832
Li et al. 2020 [44]	27.75	0.939

However, there is still a lot of room for improvement. Although, some of the challenges have already been addressed by a model, there is still no model that addresses them all.

We list the most notable challenges at the moment:

- Training time: The number of images used for the training process is proportional to the quality of the results obtained. However, training times, are still high. To address this, current models are trained using only one of the datasets, followed by a fine tuning. In order to test the model within different applications and its generalization capacity, the process is repeated for each one of the datasets. Iizuka et al. [33] model, takes roughly 2 months on a single machine equipped with four K80 GPUs, to complete the entire training procedure. Different approaches has been tested. Yu et al. [36] model reduced the training time to just a week by using spatially discounted reconstruction loss with a weighted mask.
  Training time varies within each model however. By reducing the training time, multiple dataset training can be achieved, improving the quality of the results and the generalization capabilities; allowing models to be trained only once.
- Post processing: Some models generate areas with subtle color inconsistencies with the surrounding regions. To fix this, they perform a simple post-processing by blending the completed region with the color of the surrounding

pixels. This has been addressed by Yu et al. [36] model, using a CAM. This could be applied to other models that shine in different areas, but still require post processing.

- Computational resources: Despite the promising results, some works require high computational resources and consumes considerable memory, making the model too computational resource heavy. Coarse-To-Fine models [39] uses more than 100M parameters. Partial Convolution based methods [38] uses around 33M parameters. Yu et al. [36] model uses two stacked generative networks making it one of the heaviest models. Reducing computational resources without a big loss on the quality of the reconstruction is a challenge that still needs to be addressed.
- Arbitrary masks: Models using a local discriminator suffer a drawback, being able to only deal with a single rectangular hole region. So, if any hole appears with arbitrary shapes, sizes, and locations in real-world applications, the local discriminator will fail. Liu et al. [39] shows remarkable results being able to handle masks of any shape, size, location, or distance from the image borders without losing performance as holes increase in size. Adding the mask as an extra input, is another way to address this, however this requires that the user marks the region to be filled, reducing the automation process.

VI. Conclusions

Even tough inpainting has been around for a few centuries, it's still a relevant topic today. With the introduction of digital inpainting, a complete new scheme of techniques were provided to fulfill the matter, revolutionizing the way we inpaint. This review categorizes each of the most prominent techniques on the area in the last twenty years.

Digital inpainting techniques are divided into two categories: traditional techniques and DL techniques. The traditional methods include three different groups of image inpainting algorithms: the first group consists of diffusion-based techniques, which reconstruct an image by solving a PDE; the second group consists of patch-based techniques, which use texture synthesis to inpaint missing regions of the image; and finally, the third group of convolution filter-based techniques includes all those techniques using the convolution operand to reconstruct images.

Diffusion-based techniques produce accurate results when the missing areas are small, but as the size of the missing area increases, the diffusion process adds blur to the resulting image. On large areas the algorithm takes too much time compared to the other techniques and it is unable to generate acceptable results. Patch-based techniques provide better performance in filling large hole regions. However, they depend on low-level features, filling the hole regions regardless of the visual semantics and being ineffective to inpaint complicated structures, which results in images with poor visual quality and the inability to generate novel objects not found in the source image. Convolution filter-based algorithms provide an easy, fast and precise way to inpaint small regions. However (similar to diffusion-based techniques) as the region to be inpainted increases, blurring is added to the output image.

None of the traditional techniques are able to generate novel objects not found in the source image, and they are unable to produce semantically consistent results. This greatly limits traditional techniques, leaving them unable to reconstruct complex structures. To fix this, the model would need to get the information from an external source.

DL techniques fulfill this purpose by training a model to recognize the mapping between images and their labels. The model is trained using external data (one or multiple datasets). Once the training is completed it proceeds to reconstruct the image. First approaches with DL provided faster and more precise ways for image reconstruction however, it still introduced some blurriness and completely lacked a semantic understanding of the image. With the introduction of Context Encoders, DL techniques where able to produce coherent images as a whole, reconstruct complex structures and produce novel objects, while maintaining local consistency. The blurriness on the results is produced by the models being ineffective in copying information from distant spatial locations. This was addressed by combining a DL approach with a patch based technique, providing a new purpose for patch-based techniques that were under the radar for some time.

A lot of work has been done to reduce the training process and computation resources. Some works have addressed some specific necessities like arbitrary image sizes, arbitrary masks and high resolution texture synthesis without post processing. DL techniques have greatly improved the quality on image inpainting delivering promising results by generating semantic hole filling and novel objects that are not in the original image.

Finally, inpainting evaluation metrics and datasets follows a trend. For evaluation metrics PSNR and SSIM are the default quantitative metrics to calculate the quality of the image as can be applied for the local and global regions. For datasets, Places2, DTD, CelebA, CelebHQ and ImageNet are the preferred datasets: DTD to train on different textures; CelebA and CelebHQ to train the ability of the model to reconstruct human faces and complex novel structures; Places2 trains the model to reconstruct all kinds of backgrounds and environments; and finally, ImageNet to reconstruct different types of objects.

Acknowledgment

This study was financed in part by the Coordenaçao de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001, and the Brazilian agencies FACEPE and CNPq

References

[1] G. Emile-Male, "The restorer's handbook of easel painting," 1976.

[2] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros, "Context encoders: Feature learning by inpainting," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2536–2544.

[3] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester, "Image inpainting," in *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, 2000, pp. 417–424.

[4] T. Chan, "Local inpainting models and tv inpainting," *SIAM J. Appl. Math.*, vol. 62, no. 3, pp. 1019–1043, 2001.

[5] T. F. Chan and J. Shen, "Nontexture inpainting by curvature-driven diffusions," *Journal of visual communication and image representation*, vol. 12, no. 4, pp. 436–449, 2001.

[6] H. Grossauer and O. Scherzer, "Using the complex ginzburg-landau equation for digital inpainting in 2d and 3d," in *International Conference on Scale-Space Theories in Computer Vision*. Springer, 2003, pp. 225–236.

[7] A. Telea, "An image inpainting technique based on the fast marching method," *Journal of graphics tools*, vol. 9, no. 1, pp. 23–34, 2004.

[8] J. A. Sethian, "Fast-marching level-set methods for three-dimensional photolithography development," in *Optical Microlithography IX*, vol. 2726. International Society for Optics and Photonics, 1996, pp. 262–272.

[9] A. A. Efros and T. K. Leung, "Texture synthesis by non-parametric sampling," in *Proceedings of the seventh IEEE international conference on computer vision*, vol. 2. IEEE, 1999, pp. 1033–1038.

[10] L.-Y. Wei and M. Levoy, "Fast texture synthesis using tree-structured vector quantization," in *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, 2000, pp. 479–488.

[11] M. Ashikhmin, "Synthesizing natural textures," in *Proceedings of the 2001 symposium on Interactive 3D graphics*, 2001, pp. 217–226.

[12] A. A. Efros and W. T. Freeman, "Image quilting for texture synthesis and transfer," in *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, 2001, pp. 341–346.

[13] 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 (ToG)*, vol. 20, no. 3, pp. 127–150, 2001.

[14] V. Kwatra, A. Schödl, I. Essa, G. Turk, and A. Bobick, "Graphcut textures: image and video synthesis using graph cuts," *ACM Transactions on Graphics (ToG)*, vol. 22, no. 3, pp. 277–286, 2003.

[15] A. Criminisi, P. Perez, and K. Toyama, "Object removal by exemplar-based inpainting," in *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings.*, vol. 2. IEEE, 2003, pp. II–II.

[16] W.-H. Cheng, C.-W. Hsieh, S.-K. Lin, C.-W. Wang, and J.-L. Wu, "Robust algorithm for exemplar-based image inpainting," in *Proceedings of International Conference on Computer Graphics, Imaging and Visualization*, 2005, pp. 64–69.

[17] Y. Wexler, E. Shechtman, and M. Irani, "Space-time completion of video," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 29, no. 3, pp. 463–476, 2007.

[18] D. Simakov, Y. Caspi, E. Shechtman, and M. Irani, "Summarizing visual data using bidirectional similarity," in *2008 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2008, pp. 1–8.

[19] C. Barnes, E. Shechtman, A. Finkelstein, and D. B. Goldman, "Patchmatch: A randomized correspondence algorithm for structural image editing," *ACM Trans. Graph.*, vol. 28, no. 3, p. 24, 2009.

[20] 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 (TOG)*, vol. 31, no. 4, pp. 1–10, 2012.

[21] M. M. O. B. B. Richard and M. Y.-S. Chang, "Fast digital image inpainting," in *Appeared in the Proceedings of the International Conference on Visualization, Imaging and Image Processing (VIIP 2001), Marbella, Spain*, 2001, pp. 106–107.

[22] M. M. Hadhoud, K. A. Moustafa, and S. Z. Shenoda, "Digital images inpainting using modified convolution based method," *Int. J. Signal Process. Image Process. Pattern Recogn*, pp. 1–10, 2008.

[23] H. Noori, S. Saryazdi, and H. Nezamabadi-Pour, "A convolution based image inpainting," in *1st International Conference on Communication and Engineering*, vol. 1, 2010, p. 2.

[24] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[25] K. Fukushima, "Neocognitron: A hierarchical neural network capable of visual pattern recognition," *Neural networks*, vol. 1, no. 2, pp. 119–130, 1988.

[26] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Backpropagation applied to handwritten zip code recognition," *Neural computation*, vol. 1, no. 4, pp. 541–551, 1989.

[27] J. Xie, L. Xu, and E. Chen, "Image denoising and inpainting with deep neural networks," in *Advances in neural information processing systems*, 2012, pp. 341–349.

[28] D. Eigen, D. Krishnan, and R. Fergus, "Restoring an image taken through a window covered with dirt or rain," in *Proceedings of the IEEE international conference on computer vision*, 2013, pp. 633–640.

[29] L. Xu, J. S. Ren, C. Liu, and J. Jia, "Deep convolutional neural network for image deconvolution," in *Advances in neural information processing systems*, 2014, pp. 1790–1798.

[30] R. Köhler, C. Schuler, B. Schölkopf, and S. Harmeling, "Mask-specific inpainting with deep neural networks," in *German Conference on Pattern Recognition*. Springer, 2014, pp. 523–534.

[31] J. S. Ren, L. Xu, Q. Yan, and W. Sun, "Shepard convolutional neural networks," in *Advances in Neural Information Processing Systems*, 2015, pp. 901–909.

[32] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014, pp. 2672–2680.

[33] S. Iizuka, E. Simo-Serra, and H. Ishikawa, "Globally and locally consistent image completion," *ACM Transactions on Graphics (ToG)*, vol. 36, no. 4, pp. 1–14, 2017.

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

[35] P. Pérez, M. Gangnet, and A. Blake, "Poisson image editing," in *ACM SIGGRAPH 2003 Papers*, 2003, pp. 313–318.

[36] J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu, and T. S. Huang, "Generative image inpainting with contextual attention," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 5505–5514.

[37] Z. Yan, X. Li, M. Li, W. Zuo, and S. Shan, "Shift-net: Image inpainting via deep feature rearrangement," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 1–17.

[38] 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*. Springer, 2015, pp. 234–241.

[39] 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 *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 85–100.

[40] Y. Zeng, J. Fu, H. Chao, and B. Guo, "Learning pyramid-context encoder network for high-quality image inpainting," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2019, pp. 1486–1494.

[41] M.-c. Sagong, Y.-g. Shin, S.-w. Kim, S. Park, and S.-j. Ko, "Pepsi: Fast image inpainting with parallel decoding network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 11 360–11 368.

[42] Y. Jiang, J. Xu, B. Yang, J. Xu, and J. Zhu, "Image inpainting based on generative adversarial networks," *IEEE Access*, vol. 8, pp. 22 884–22 892, 2020.

[43] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein gan," *arXiv preprint arXiv:1701.07875*, 2017.

[44] J. Li, N. Wang, L. Zhang, B. Du, and D. Tao, "Recurrent feature reasoning for image inpainting," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[45] 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*, vol. 40, no. 6, pp. 1452–1464, 2017.

[46] Z. Liu, P. Luo, X. Wang, and X. Tang, "Deep learning face attributes in the wild," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 3730–3738.

[47] T. Karras, T. Aila, S. Laine, and J. Lehtinen, "Progressive growing of gans for improved quality, stability, and variation," *arXiv preprint arXiv:1710.10196*, 2017.

[48] M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, and A. Vedaldi, "Describing textures in the wild," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 3606–3613.

[49] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein *et al.*, "Imagenet large scale visual recognition challenge," *International journal of computer vision*, vol. 115, no. 3, pp. 211–252, 2015.