Completing Face Pictures: a Study on Image and Facial Inpainting Methods

PMR3550 – Trabalho de Conclusão de Curso II em Engenharia Mecatrônica — Artigo

Pedro Orii Antonacio – Nº USP 10333504
Orientadora: Prof.ª Dr.ª Larissa Driemeier
Departamento de Engenharia Mecatrônica e de Sistemas Mecânicos (PMR)
Escola Politécnica da Universidade de São Paulo
São Paulo, Brasil – 2021

***Abstract*—Since the early 2000's, the research on digital image inpainting has been very active and encountered numerous applications in image processing and computer vision. With the increase in computer processing power and the development of high-quality image datasets, many new Deep Learning inpainting methods were proposed in recent years, achieving outstanding results with no precedents in traditional methods. For instance, DeepFillv2 is a complete and robust GAN-based model regarded as one of the most practical and well-established image inpainting methods available to this day. However, for the use case of facial inpainting, the DeepFillv2 was initially trained on CelebA-HQ, a dataset composed exclusively of celebrities' face images, which makes it strongly biased towards white and attractive faces mostly from adult western personalities. As an alternative, this project proposes to train the DeepFillv2 model from scratch on FFHQ, a dataset of faces of normal people from all over the world, which offers greater variation in terms of age, ethnicity and image background, besides being more than twice as large as CelebA-HQ. Therefore, this project aims to to improve the state-of-the-art DeepFillv2 method for the use case of facial inpainting through its training data. Detailed qualitative and quantitative evaluations suggest that our DeepFillv2 model trained on the FFHQ dataset was able to produce better results than the DeepFillv2 model originally trained on CelebA-HQ.**

***Keywords*—Image Inpainting, Facial Inpainting, Deep Learning, Generative Adversarial Network (GAN), DeepFillv2**

I. Introduction

Image inpainting—or image completion—refers to the process of reconstructing damaged or missing parts of an image in such a way that the inpainted image appears seamless, physically plausible and visually pleasing to the casual observer. Nowadays, image inpainting is an important field in computer vision and lays the functional foundations in many imaging and graphics applications, from image denoising to object removal.

This research project focuses on facial inpainting, which is a subfield of image inpainting. Also referred as facial image inpainting and face completion, facial inpainting is the task of completing damaged or missing parts on facial images using image inpainting techniques and taking advantage of the similarities between different human faces.

In the last couple of decades, with the advancements in digital image processing and in computer vision, the research fields of image and facial inpainting have been very active, boosted by numerous application other than image restoration, such as removal of objects [1] and text overlays [2], image-based rendering [3] and image denoising [4], compression [5], retargeting [6] and compositing [7]. More recently, Faria Silva *et al.* [8] proposed the use of inpainting techniques to optimize the production process of nasal prostheses.

Today, digital image and facial inpainting methods are able to achieve exceptional results, maintaining high levels of accuracy and consistency across different types of scenarios and applications.

II. State of the Art

State-of-the-art digital image inpainting methods can be classified into two major groups: Traditional Methods and Deep Learning Methods, each with their corresponding sub-categories. Figure 1 presents a classification of the digital image inpainting methods that are going to be discussed in this paper.

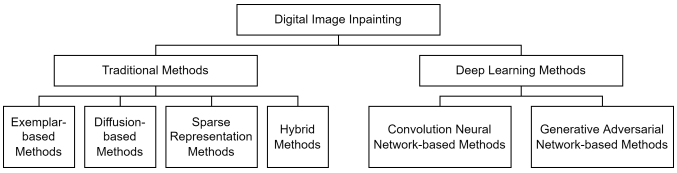

Fig. 1. Classification of digital image inpainting methods

A. Traditional Methods

This subsection highlights a few of the most prominent traditional image inpainting methods proposed in the last couple of decades, which can be categorized into four groups: exemplar-based methods, diffusion-based methods, sparse representation methods, and hybrid methods.

1) Exemplar-based Methods: Also known as patch-based methods, exemplar-based methods complete the missing regions of the image by sampling patches from the existing parts, and then generating new textures and structures that are visually similar to the sampled patches whilst not being exact copies of them.

a) Exemplar-based Texture Synthesis: Exemplar-based texture synthesis methods are able to successfully generate textures similar to the ones present on the image while preserving recurrent details, even when they are not continuous. On the other hand, these methods can also produce meaningless and unreasonable results when the input image presents texture variations or more complex structures.

b) Exemplar-based Structure Synthesis: Exemplar-based structure synthesis methods use similar patches from the known parts of the image to restore texture in the missing region while maintaining structural consistency. However, these methods struggle to restore areas with little similarity with the rest of the image and are unable complete curved or more complex structures.

2) Diffusion-based Methods: Diffusion-based methods perform the inpainting task by smoothly propagating image content from boundary areas into the interior of the missing regions in the image. Diffusion-based methods produce accurate and satisfactory results when inpainting small areas like scratches, straight lines, curves, and edges. However, as the size of the missing regions increases, the resulting inpainted regions get more and more blurred and lack texture information, also significantly increasing the computational cost of those methods.

3) Sparse Representation Methods: Sparse representation inpainting methods assume that images contain natural signals that admit a sparse decomposition over a redundant dictionary, which can be regarded as a compressed or encoded version of the original image vector. Despite having the same number of elements as the original image, the sparse representation has mostly zero entries, reducing noise and focusing on the more relevant information from the image. The inpainting task is then conducted under the assumption that the existing and missing regions share similar sparse representations.

4) Hybrid Methods: Hybrid inpainting methods use a combination of exemplar-based and diffusion-based techniques—the ones that produced the best results among traditional methods—, in an attempt to take advantage of their strengths and overcome their main limitations in a complementary manner. Hybrid methods are capable of completing structures and texture with considerable local coherence and visual quality, while avoiding generating blurred results. However, these methods perform poorly and have a high computational cost when inpainting regions increase in size. Also, they offer no guarantee of convergence during the inpainting task.

Summary of Traditional Methods: Traditional digital inpainting methods are able to accurately complete structure and texture and thus produce satisfactory and high-quality results when the missing areas are small and narrow. However, these methods cannot capture high-level semantic features from the image and therefore fail to perform the inpainting task as missing areas get larger and structures and textures become more complex, resulting in non-realistic images with repetitive patterns. Table I summarizes the discussed traditional image inpainting methods.

Moreover, because traditional methods use only the existing information on the input image to complete the missing regions, they cannot generate novel structures or textures besides those previously available in the input image, which can be a major limitation in applications that depend on high-level semantics and abstractions. In Facial Inpainting, for example, if entire facial elements were masked, such as the whole nose, both eyes or the mouth, traditional methods would not be able to perform the inpainting task, as they would not be able to generate such facial elements since the necessary information is not present in the input image.

TABLE I
SUMMARY OF THE STATE-OF-THE-ART TRADITIONAL METHODS FOR IMAGE INPAINTING

Category	Methods	Advantages	Disadvantages
Exemplar-based Texture Synthesis	Efros and Leung (1999) [9]; Efros and Freeman (2001) [10]; Le Meur *et al.* (2012) [11]	No occurrence of blur. Preserves textural information.	Fails to reconstruct large textured regions or images with multiple damaged areas. Unable to propagate structural consistency. Can lead to repetitive and meaningless patterns.
Exemplar-based Structure Synthesis	Criminisi *et al.* (2004) [1]; Barnes *et al.* (2009) [12]	Restores texture, structure and colour. Performs well in the completion of large textured regions.	The process of finding candidate patches is costly and time consuming. Can lead to repetitive patterns. Unable to complete curved or complex structures.
Diffusion-based Methods	Bertalmio *et al.* (2000) [13]; Bertalmio *et al.* (2001) [14]; Chan and Shen (2001) [15]; Shen *et al.* (2003) [16]; Telea (2004) [17]	Produces satisfactory results when missing areas are small and narrow (suitable for completing lines and curves). Does not generate exact copies from know regions of the image. Preserves structure of inpainted region.	Unable to inpaint large missing areas. Unable to preserve texture information. Can generate overly blurred results. Computational cost increases significantly with the size of missing areas.
Sparse Representation Methods	Mairal *et al.* (2007) [18]; Shen *et al.* (2009) [19]	Preserves color and simple textures and structures. Able to handle change in light intensity.	Unable to inpaint complex textures or structures. Can lead to blurred results.
Hybrid Methods	Bertalmio *et al.* (2003) [20]; Le Meur *et al.* (2011) [21]	Able to preserve structures and texture in a satisfactory way. Performs well when restoring linear structure.	Computationally complex and costly with no guarantee of convergence.

B. Deep Learning Methods

In the last decade, the increase in computer graphics processing power and the development of high-quality and reliable image datasets enabled significant advancements in Deep Learning, specially in the domain of computer vision. As a consequence, new data-driven learning-based image inpainting methods were proposed, achieving remarkable results with greater generalization capabilities than traditional methods. The Deep Learning-based techniques can be classified into two major groups: Convolutional Neural Network-based methods and Generative Adversarial Network-based methods.

1) Convolution Neural Network-based Methods: Convolutional Neural Networks (CNNs) have been widely employed in state-of-the-art computer vision tasks and were quickly adopted for image inpainting tasks as well. In this context, CNNs are used as image feature extractors to capture high dimensional data abstractions.

2) Generative Adversarial Network-based Methods: Generative Adversarial Network (GAN) is a two-model framework (generator and discriminator) used for estimating and training generative models through an adversarial process. This framework was proposed by Goodfellow *et al.* [31] and represented a major breakthrough in Deep Learning due to its exceptional capacity of approximating data distributions and generating plausible new data from the learned distribution, which can be images, audios clips, videos, etc. that do not exist in reality.

The intuition behind GANs is quite simple yet very powerful. In image applications like inpainting, the generative model (generator) starts with a random input noise and transforms it into a fake image, intending it to look like a real image from the training set. The discriminative model (discriminator) then receives either this generated image or a real sample from the training set and decides whether the received image is real or not, by estimating the probability that it came from the training set rather than from the generative model. With enough training in a properly built model, the generator should become increasingly better at creating images that look like real images, whereas the discriminator should get increasingly better at distinguishing generated images from real ones.

Despite originally designed as an unsupervised learning framework, GANs were proven useful for semi-supervised learning, fully supervised learning, and reinforcement learning as well, achieving promising results in the most various Deep Learning applications, including image and facial inpainting. The discriminative and generative models from GAN-based image and facial inpainting methods are built with CNNs, combining CNN's image feature extraction and pixel synthesis capabilities with the enhanced coherency between generated and original pixels that can be obtained through the adversarial training. Finally, because image inpainting GAN models' learning is unsupervised, any image dataset can be used for training, eliminating the need for manually masking and labeling pictures for an inpainting training set.

TABLE II
SUMMARY OF THE STATE-OF-THE-ART DEEP LEARNING METHODS FOR IMAGE INPAINTING

Method	Year	Model	Description	Loss Function	Dataset
Jain *et al.* [22]	2008	CNN	Autoencoder formulated for image denoising and extended to image inpainting.	Reconstruction loss	In-house 100 grayscale images
Xie *et al.* [4]	2012	CNN	Sparse coding and deep neural networks as a denoising autoencoder. Extended to image inpainting.	Reconstruction loss	In-house images
Pathak *et al.* [23]	2016	GAN	Context Encoder autoencoder network as the generative model.	*L2* reconstruction loss and adversarial loss	Paris StreetView, ImageNet, PASCAL VOC 2007
Yang *et al.* [24]	2017	GAN	Style transfer-based texture network and encoder-decoder content network for multi-scale neural patch synthesis with high-frequency details.	*L2*-based texture loss and adversarial loss computed with VGG19 features.	Paris StreetView, ImageNet
Iizuka *et al.* [25]	2017	GAN	Global and local discriminators and encoder-decoder generator with dilated convolutions	Weighted *L2*, adversarial loss	Places2, ImageNet, CMP Facade
Yeh *et al.* [26]	2017	GAN	Search for closest encodings on latent space assisted by spatial attention mechanism to reconstruct the original image.	Weighted *L1*-based context loss and adversarial loss.	CelebA, SVHN, Stanford Cars.
Yu *et al.* [27]	2018	GAN	DeepFillv1: local and global discriminators and a two-stage feed-forward coarse-to-fine generative model with Contextual Attention layers and dilated convolutions.	Reconstruction loss and global and local WGAN-GP adversarial losses	Places2, CelebA, CelebA-HQ, DTD, ImageNet
Liu *et al.* [28]	2018	CNN	U-Net-based network with only partial convolutions in order to disregard invalid masked pixels	Total variation, *L1*, perceptual and style losses	Places2, CelebA, CelebA-HQ, ImageNet
Nazeri *et al.* [29]	2019	GAN	EdgeConnect: a two-stage adversarial model. First stage predicts an edge map and the second stage fills the predicted map with color and texture.	Feature matching, style, perceptual, *L1* and adversarial losses	Paris StreetView, Places2, CelebA
Yu *et al.* [30]	2019	GAN	DeepFillv2: DeepFillv1 with Gated Convolutions (learnable Partial Convolutions), optional user-guided inpainting and SN-PatchGAN discriminator.	Pixel-wise *L1* reconstruction loss and SN-PatchGAN loss	Places2, CelebA-HQ

In conclusion, despite being around for no more than years, GAN-based image and facial inpainting methods have greatly evolved and established themselves as the best methods for digital image inpainting, achieving outstanding results with no precedents in traditional or CNN-based methods.

Summary of Deep Learning Methods: Deep Learning image inpainting methods are able to produce remarkable results and solved the main limitations from traditional methods, such as the lack of semantic understanding of the image, the inability to generate novel and previously unseen information, and the blurriness and repetitive patterns of larger inpainted regions. Table II summarizes the reviewed Deep Learning image inpainting methods.

C. Datasets

Whether the learning task consists on mapping images to labels (supervised) or generating new plausible images through adversarial training (unsupervised), Deep Learning image inpainting methods need datasets in order to be developed, trained and evaluated. Table III summarizes the main aspects of the most relevant datasets used to develop, train and evaluate state-of-the-art Deep Learning image inpainting methods.

TABLE III
SUMMARY OF THE MAIN DATASETS USED IN STATE-OF-THE-ART DEEP LEARNING IMAGE INPAINTING METHODS

Dataset	Year	Content	Total Images	Image Type	Resolution
PASCAL VOC 2007	2007	Miscellaneous	19,737	RGB	Variable
ImageNet	2009	Miscellaneous	14 million	RGB	Variable
Paris Street View	2015	Street images	10,000	RGB	936 × 537
CelebA	2015	Facial images	202,599	RGB	Variable
CelebA-HQ	2017	Facial images	30,000	RGB	1024 × 1024
Places2	2017	Scenery images	10 million	RGB	512 × 512
NVidia Irregular Mask Dataset	2018	Masks	79,982	Binary	512 × 512
Quick Draw Irregular Mask Dataset	2018	Masks	60,000	Binary	512 × 512
Flickr-Faces-HQ (FFHQ)	2019	Facial images	70,000	RGB	1024 × 1024

III. MATERIALS AND METHODS

A. Project Objective

The main goal of this project is to attempt to improve an established state-of-the-art Deep Learning image inpainting method for the specific use case of facial inpainting. To do that, the selected method is the DeepFillv2 model proposed by Yu *et al.* [30] in 2019, a complete and robust GAN-based method that achieved unprecedented inpainting results on the evaluated datasets when compared to previous state-of-the-art methods, and which is regarded as one of the most practical and well-established image inpainting methods available to this day.

According to the authors, DeepFillv2 was trained and evaluated on CelebA-HQ and Places2 datasets separately. The dataset used to train a Deep Learning model has a major influence on its performance on the inpainting task and thus has to be carefully selected in accordance with the targeted application. The DeepFillv2 model trained on Places2 would be suitable for reconstructing image backgrounds and scenery pictures, but would perform poorly in other scenarios such as completing animal pictures, since images of animals are extremely rare in the Places2 dataset. On the other hand, for the specific application of facial inpainting at which this project is targeted, the DeepFillv2 model trained on CelebA-HQ is much more appropriate, since it was trained exclusively on facial images and hence was able to learn to understand and extract high-level semantics and features present in the different elements of the human face.

Nonetheless, there is still room for improvement on the employment of DeepFillv2 for facial inpainting in real-world applications. Because CelebA-HQ is exclusively composed of face images from celebrities, it is a dataset strongly biased towards white and what would be considered attractive and good-looking faces, with a significant prevalence of adult western personalities. As an alternative, the FFHQ dataset would be more suitable for facial inpainting in real-world applications. Besides being more than twice as large as CelebA-HQ, the FFHQ dataset is composed of facial pictures of normal people from all over the world and therefore offers significantly greater variation in terms of age, ethnicity and image background when compared to CelebA-HQ.

For those reasons, this project's goals can be summarized as follows:

- Train the DeepFillv2 model from scratch on the FFHQ dataset.
- Evaluate the DeepFillv2 model trained on the FFHQ dataset.
- Compare the DeepFillv2 model trained on the FFHQ dataset with the DeepFillv2 models trained on the CelebA-HQ and Places2 datasets, whose pretrained weights were made available by Yu *et al.* [30].
- Finally, discuss the obtained results.

B. Evaluation Methods for Image Inpainting

Since the introduction of the first digital inpainting algorithms, different performance evaluation methods were proposed and used in order to assess how well those algorithms were able to complete masked images. This section presents and discusses the most relevant evaluation methods, which can be divided into two groups: quantitative metrics and qualitative evaluation.

1) Quantitative Metrics: Quantitative metrics measure the quality of the generated image based on the deviation of the pixels from the reconstructed image in relation to those from the original ground-truth reference image. Although several quantitative metrics have been proposed in the last 20 years [32]—such as universal quality index [33], multi-scale SSIM [34], visual information fidelity [35], inception score [36], Fréchet inception distance [37] and LPIPS [38]—, two of them stand out as the most commonly used quantitative

metrics in the literature: Peak Signal to Noise Ratio (PSNR) and Structure Similarity Index Measure (SSIM).

2) Qualitative Evaluation: As previously discussed, the main goal of the inpainting task is to produce images that appear seamless, physically plausible and visually pleasing to the casual observer. In other words, the overall quality of an inpainted image is inevitably conditioned to at least some degree of human subjectivity.

In this way, the inpainting task and the process of artwork creation are somewhat similar. Given a masked image or a blank canvas, there are multiple—if not infinite—ways in which these objects could be completed such that the result would be considered satisfactory and of high quality. It is not for nothing that, for centuries, artists were the main responsible for executing the inpainting tasks.

For those reasons, quantitative metrics based on pixel-wise reconstruction errors like PSNR or structural similarity like SSIM are not necessarily the best ways to evaluate inpainted images. As an alternative, a qualitative evaluation can be employed to provide additional insights into the inpainting results.

A typical qualitative evaluation consists of visually inspecting specific parts and the entire inpainted image, and then comparing it to the ground-truth image and to the results by other inpainting methods. Despite being simple and lacking objective metrics, this type of evaluation can quickly highlight the main flaws and strengths of each algorithm and thus has been proven useful when comparing different inpainting methods, being widely adopted in many of the previously mentioned papers, including those describing the DeepFillv1 and DeepFillv2 methods [27], [30].

C. The DeepFillv2 Model

As discussed in the previous chapter, DeepFillv2 [30] incorporated key concepts from previous GAN-based method to produce a complete, robust and well-performing model. This section describes the main features and characteristics of the DeepFillv2 model.

1) Gated Convolutions: In a standard convolutional layer, considering an input channel C, each output pixel O located at (x, y) in the output channel C' is calculated as:

$$O_{y,x} = \sum_{i=-k'_h}^{k'_h} \sum_{j=-k'_w}^{k'_w} W_{k'_h+i,k'_w+j} \cdot I_{y+i,x+j} \tag{1}$$

where x, y represent the x- and y-axis of the output map, k_h and k_w denote the kernel's height and width (*e.g.* 3×3), $k'_h = \frac{k_h-1}{2}$, $k'_w = \frac{k_w-1}{2}$, $W \in \mathbb{R}^{k_h \times k_w \times C' \times C}$ represents the kernels, and $I_{y+i,x+j} \in \mathbb{R}^C$ and $O_{y,x} \in \mathbb{R}^{C'}$ are the input and output pixel values.

In such convolutions—also referred as vanilla convolutions—, the same kernels, or convolutional filters, are applied to all input pixels through a sliding window. This approach works well for most computer vision tasks like image classification or object detection, but falls short in image inpainting tasks, where the input of each layer is composed of both valid and invalid/masked pixels. The invalid/masked pixels cause ambiguity in the training process, which leads to visual flaws such as color discrepancies, blurriness and inconsistencies in the image texture and structure.

In 2018, Liu *et al.* [28] introduced Partial Convolutions, which update and normalize the binary mask at each layer to make the convolutions depend only on valid pixels. The output values are computed according to the following rule:

$$O_{y,x} = \begin{cases} \sum\sum W \cdot \left(I \odot \frac{M}{\mathrm{sum}(M)}\right), & \text{if } \mathrm{sum}(M) > 0 \\ 0\,, & \text{otherwise} \end{cases} \tag{2}$$

where M is the corresponding binary mask for each kernel window (with 1 for valid pixels and 0 for invalid ones), and $\odot$ represents element-wise multiplication. Then, after each partial convolution is performed, a new binary mask M' is updated with the following rule: $m'_{y,x} = 1$ if $\mathrm{sum}(M) > 0$.

By making the convolutions conditioned only on valid pixels, Partial Convolutions produce significantly better inpainting results than vanilla convolutions while also supporting masks of any shape and size. However, Partial Convolutions still present some limitations. First, after each convolution, the binary mask update rule sets $m'_{y,x}$ to 1 no matter how many valid pixels from the previous layer are covered by the kernel window. For example, in a 3×3 window, $m'_{y,x}$ is set to 1 both when there is only 1 valid pixel and also when all 9 pixels are valid, which can propagate unwanted information from invalid pixels throughout the convolutional layers. Also, with such mask update rule, invalid pixels in the binary masks are guaranteed to disappear in deep layers, after a sufficient number of convolutions, which prevents the mapping of larger invalid regions from being propagated to the deeper layers of the model. Finally, all channels in a single layer must share the same binary mask, limiting the flexibility of the model.

To address those issues, DeepFillv2 introduced Gated Convolutions. Instead of hard-gating masks based on fixed and arbitrary update rules, Gated Convolutions introduce an extra standard convolutional layer followed by a sigmoid function at each step, which provides a learnable dynamic feature selection mechanism for every channel at each spatial location across all layers, while still supporting masks of any shape and size. It is formulated as:

$$\begin{aligned} \text{Gating}_{y,x} &= \sum\sum W_g \cdot I \\ \text{Feature}_{y,x} &= \sum\sum W_f \cdot I \\ \Rightarrow \quad O_{y,x} &= \phi(\text{Feature}_{y,x}) \odot \sigma(\text{Gating}_{y,x}) \end{aligned} \tag{3}$$

where σ is the sigmoid function, ϕ can be any activation function (*e.g.* ReLU, ELU, LeakyReLU), and W_g and W_f are two distinct convolutional kernels.

In this way, the validness/importance ($\text{Gating}_{y,x}$) of each pixel or feature in the image becomes a learnable parameter through the model training process. Also, due to the sigmoid activation function, such pixel validness/importance can be

any number in the $[0, 1]$ interval, as opposed to either 0 or 1 in the hard-gating approach from Partial Convolutions, which leads to a more accurate estimation of the relative weight of each pixel at each Gated Convolution layer. Finally, because Gated Convolutions introduce a dynamic feature selection mechanism for every channel, the model can learn different masks for each image channel, and the inputs are not limited to the standard RGB channels anymore, allowing the DeepFillv2 model to support an additional user-provided sketch channel for edge guidance. Figure 2 illustrates the main differences between Partial and Gated Convolutions.

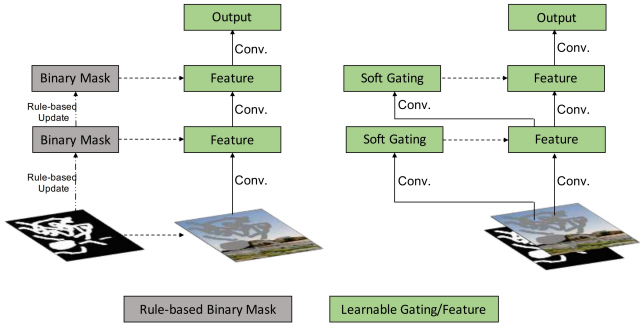

Fig. 2. Comparison between Partial Convolutions (left) and Gated Convolutions (right). Source: Yu *et al.* [30]

2) Network Architecture: The DeepFillv2 generative model architecture is the same two-stage feed-forward coarse-to-fine model from its predecessor DeepFillv1, except that all standard convolutions are replaced by Gated Convolutions (the dilated convolution and Contextual Attention layers remain unchanged). Just as in the DeepFillv1 model, the Contextual Attention mechanism is introduced to explicitly borrow features from regions that are spatially distant from the masked area. The first stage of the generator—the Coarse Network—is an autoencoder responsible for producing a rough estimation of the masked region, while the second stage—the Refinement Network—is a two-branch network that uses the Contextual Attention mechanism alongside dilated convolutions to polish this rough estimation and generate the final inpainted result.

Differently from previous GAN-based methods, DeepFillv2 does not employ an additional local discriminator focused on a fixed-size rectangular masked area, as it was designed to support masks of any shape and size. Instead, motivated by global and local GANs [25], MarkovianGANs [39], perceptual loss [40] and spectral-normalized GANs [41], the DeepFillv2 model introduces a single spectrally normalized patch-based GAN discriminator, named SN-PatchGAN, which consists of a convolutional network that takes the generated image as input and outputs a 3D feature with shape $\mathbb{R}^{h \times w \times c}$ (h, w, c being the height, width and number of channels, respectively). Finally, to discriminate if the input is real or fake, the hinge loss is directly applied to each element of this output 3D feature map, contemplating all different locations and multiple semantics—represented in the different channels—of the input image. As Yu *et al.* [30] describe, "SN-PatchGAN is simple in formulation, fast and stable in training and produces high-quality inpainting results". Figure 3 presents the complete DeepFillv2 model architecture.

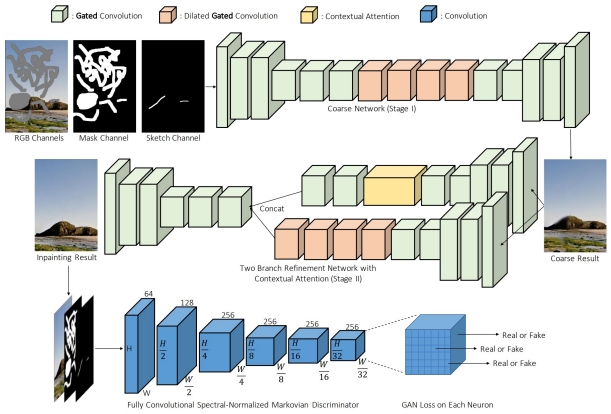

Fig. 3. Overview of DeepFillv2 framework with Gated Convolutions and SN-PatchGAN discriminator for free-form image inpainting. Source: Yu *et al.* [30]

3) Loss Function: Because of the SN-PatchGAN discriminator, the loss function of the DeepFillv2 is significantly simplified, consisting of only two terms: the pixel-wise *L1* reconstruction loss and the SN-PatchGAN adversarial loss, with default loss balancing hyperparameter set to 1:1—as opposed to the 6 different loss terms used in the Partial Convolutions method, for example.

To discriminate if the input is real or fake, DeepFillv2 model uses the hinge loss as objective function for the generator $\mathcal{L}_G = -\mathbb{E}_{z \sim \mathbb{P}_z(z)} \left[D^{sn}(G(z)) \right]$ and for the discriminator $\mathcal{L}_{D^{sn}} = \mathbb{E}_{x \sim \mathbb{P}_{\text{data}}(x)} \left[\text{ReLU} \left(\mathbb{1} - D^{sn}(x) \right) \right] + \mathbb{E}_{z \sim \mathbb{P}_z(z)} \left[\text{ReLU} \left(\mathbb{1} + D^{sn}(G(z)) \right) \right]$, where D^{sn} represents the SN-PatchGAN discriminator, and G is the generative network that takes the incomplete image z as input.

4) Free-Form Mask Generation Algorithm: Yu *et al.* [30] also proposed an algorithm to automatically generate free-form masks on-the-fly during the training procedure of the DeepFillv2 model. This algorithm was designed in a way that the generated masks are "(1) similar to masks drawn in real use-cases, (2) diverse to avoid over-fitting, (3) efficient in computation and storage, and (4) controllable and flexible". To do that, the proposed algorithm generates masks by combining rectangular shapes with randomized connected lines that simulate human's back-and-forth brushing behaviour when using a digital eraser. Figure 4 presents two examples of masks that were automatically generated during the DeepFillv2 model training process on the the FFHQ dataset.

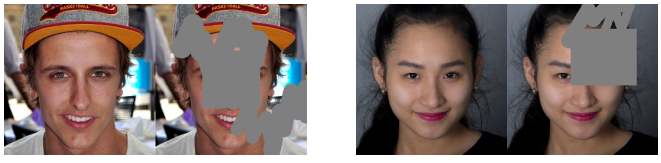

Fig. 4. Examples of masks automatically generated during the DeepFillv2 model training process on the FFHQ dataset.

5) Extension to User-Guided Image Inpainting: The DeepFillv2 model supports an additional and optional input channel for user-provided sketches that guide the completion of structures during the inpainting task. Sketches, or edges, are intuitive for users to draw and such guidance can produce more meaningful and satisfactory results once they are closer to the user's expectations, which can e useful in practical applications.

For faces, the model extracts facial landmarks and connects related landmarks according to the sketch, whereas for natural scene images, it directly extracts edge maps using the HED edge detector [42] before connecting them accordingly. By using conditional channels as input to the discriminator, the model is able to learn a conditional generative network in which the generated results respect user guidance faithfully, so it is not necessary to add an additional term to the loss function to enable the model's training procedure. The user-guided inpainting model is separately trained with a 5-channel input (RGB color channels, and mask and sketch channels).

D. Implementation

1) Dataset: The dataset used in this project implementation is the FFHQ dataset resized to 256×256 pixels in order to speed up the training procedure. This dataset was downloaded from Kaggle's "Flickr-Faces-HQ Dataset (Nvidia) - Resized 256px" repository [43], has a size of approximately 2GB and consists of 70,000 face images, which were split into two groups: 60,000 for the training process and 10,000 for the model validation.

The DeepFillv2 models trained on the CelebA-HQ and Places2 datasets whose pretrained weights were made available by Yu *et al.* [30] also used resized CelebA-HQ and Places2 datasets at 256×256 resolution. For that reason, utilizing the resized 256×256 FFHQ dataset in this project's implementation allows for the comparison of those three models' outputs by using the same input image, which standardizes the evaluation process.

2) Model Training: The training process was carried out in the Google Colab platform, with the Pro+ subscription. The DeepFillv2 model was trained for a total of 280 hours, which corresponded to 1,048,000 iterations or 262 epochs, as 1 epoch consisted of 4,000 iterations. The utilized servers were equipped with NVIDIA's Tesla P100 PCIe GPU with 16GB of memory, and either 13.6GB or 54.8GB of RAM, depending on the assigned runtime. The average GPU memory usage was 10.5GB and the average RAM usage was 10.0GB when using a runtime with 13.6GB of RAM, and 6.24GB when using a runtime with 54.8GB of RAM. On average, the training procedure computed 1.04 batch per second, with batch size of 16 images. The utilized Tensorflow version was 1.15.0.

Figure 5 presents a graph of DeepFillv2 loss function computed on the validation set during the training iterations and the corresponding smoothed curve, calculated through an exponentially weighted moving average (EWMA). The average value of the loss function decreases over the training iterations, indicating that the model learns and gets better at performing the inpainting task on unseen data from the validation set.

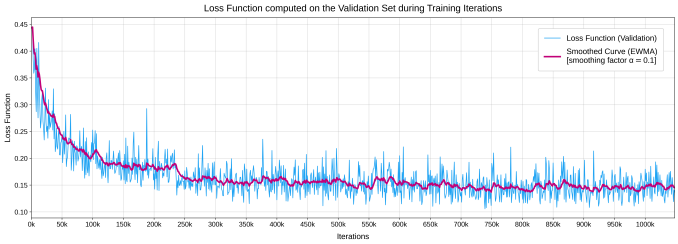

Fig. 5. DeepFillv2 loss function computed on the validation set during training iterations and the corresponding smoothed curved.

This project's implementation details are publicly available in a GitHub repository at: https://github.com/PedroAntonaci o/facial-inpainting [44]. The next chapter presents this results of the DeepFillv2 model trained on the FFHQ dataset.

IV. Results

This section presents the inpainting results of our DeepFillv2 model trained from scratch on the FFHQ dataset, and compares them with the results from the DeepFillv2 models trained on the CelebA-HQ and Places2 datasets, whose pretrained weights were made available by Yu *et al.* [30]:

- Subsection IV-A "Model Progress During Training" displays how the performance of our DeepFillv2 model developed and progressed during the 1,048,000 iterations—or 280 hours—of the training process.
- Subsection IV-B "Models Comparison" compares the results from the DeepFillv2 models trained on CelebA-HQ and Places2 by Yu *et al.* [30] with the results from our DeepFillv2 model trained on the FFHQ dataset, for the case of regular facial inpainting tasks.
- Subsection IV-C "Edge Cases" compares the results from the DeepFillv2 model trained on CelebA-HQ by Yu *et al.* [30] with the results from our DeepFillv2 model trained on FFHQ, for the case of more challenging and difficult facial inpainting tasks.
- Subsection IV-D "Personal Pictures" shows the same model comparison from section IV-C "Edge Cases", but the inpainting tasks use face pictures from the author's friends and family, obtained and utilized with their consent.

With the exception of subsection IV-D "Personal Pictures"—which does not use images from the FFHQ dataset—, all the images displayed in this chapter were taken from our FFHQ validation set rather than the training set. Therefore, the selected images in the following subsections were never seen by any of the models during their respective training processes.

V. Discussion

FFHQ Model Progress During Training

Even though the average validation loss function value remains relatively constant and does not improve significantly

A. Model Progress During Training

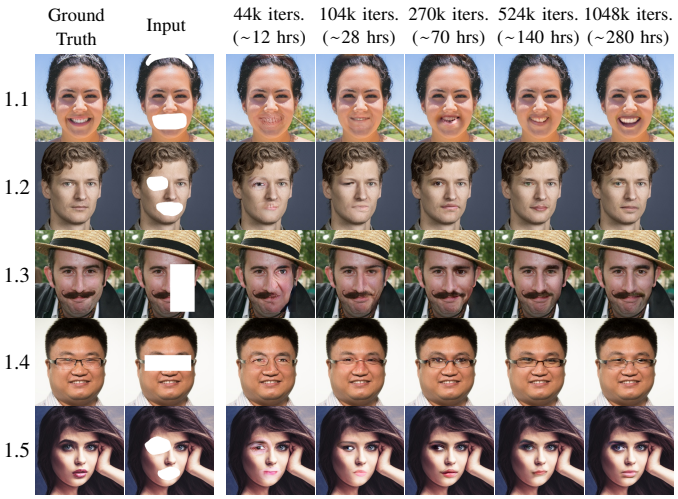

Fig. 6. Results: model progress during the training process.

B. Models Comparison

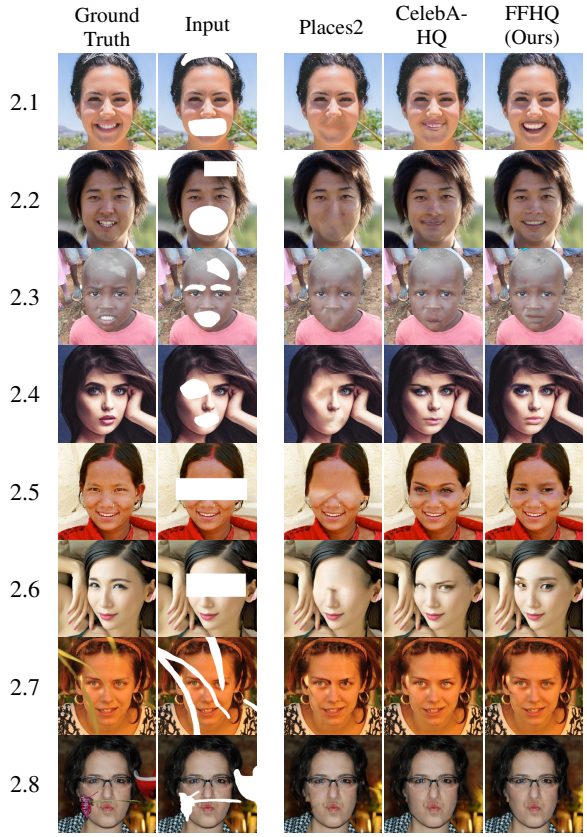

Fig. 7. Results: models comparison.

from training iteration 250,000 onwards as shown in figure 5, the snapshots of the model training process in figure 6 from subsection IV-A "Model Progress During Training" clearly show that the model does get better at performing the facial inpainting task with each additional group of training iterations and hours. The longer the model is trained, the better it becomes at propagating textures and colors and generating

C. Edge Cases

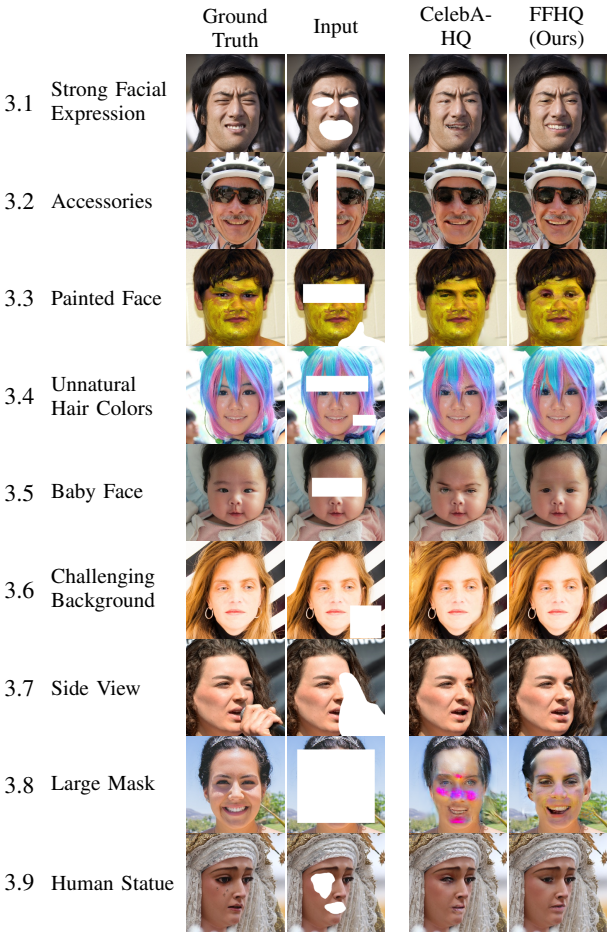

Fig. 8. Results: edge cases.

D. Personal Pictures

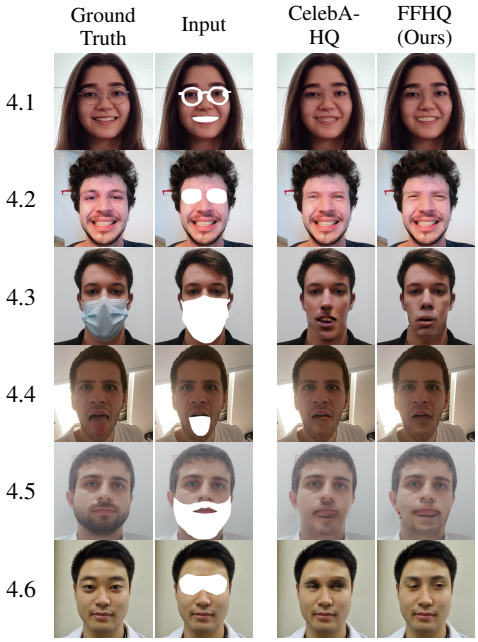

Fig. 9. Results: personal pictures.

facial structures that are more realistic and coherent with the rest of the original image.

FFHQ, CelebA-HQ and Places2 Models Evaluation

Subsection IV-B "Models Comparison" brings to light some of the key differences in the way each of the three models completes the masked images. The model trained on Places2, for instance, is not able to complete the faces in a satisfactory manner. Places2 is a dataset composed exclusively of scenery images, hence it does not contain any aligned close-up face images. As the DeepFillv2 model trained on Places2 has never encountered aligned faces during its training process, it has never learned to extract and generate facial features. So even though this model is able to smoothly propagate colors, textures and major structural outlines into the masked areas—which can be useful for reconstructing image backgrounds and scenery pictures—, it cannot understand and extract high-level semantics and features from human faces, nor is it capable of generating new facial elements such as eyes or noses. Therefore, it cannot perform the facial inpainting task successfully.

On the other hand, both DeepFillv2 models trained on CelebA-HQ and FFHQ produce satisfactory results when performing the facial inpainting task, successfully extracting high-level features and semantics from the input images and generating new facial elements to complete the masked areas. However, upon a more meticulous qualitative evaluation of the results from both models, our DeepFillv2 trained on FFHQ tends to produce better results for the images from subsection IV-B "Models Comparison", which can be clearly observed in images 2.1 to 2.4 from figure 7. Also, the FFHQ model is significantly better than the CelebA-HQ model at generating facial elements that maintain the ethnic characteristics of the original image, which can be observed in the generated eyes in images 2.5 and 2.6 from figure 7. For the use case of object removal, where masks tend to be narrower and smaller, as seen in images 2.7 and 2.8 from figure 7, both CelebA-HQ and FFHQ models produced high-quality results, without major differences between their results.

In subsection IV-C "Edge Cases", the CelebA-HQ and FFHQ models were subjected to perform significantly more challenging inpainting tasks. Although the results from both models were not perfect and most of the inpainted images would quickly be perceived as fake by casual observers—specially for the large mask and side view in images 3.7 and 3.8 from figure 8—, the models displayed a few promising capabilities in the resulting images. For example, both models were able to recognize and reconstruct the helmet and sunglasses on image 3.2, adequately propagate texture and color for the painted face and unnatural hair colors in images 3.3 and 3.4, and also deal with a challenging background in a satisfactory way in image 3.6 from figure 8. In general, both models performed similarly for the images from subsection IV-C "Edge Cases", with no model clearly and consistently outperforming the other throughout the test cases.

Real-World Facial Inpainting Applications

For subsection IV-D "Personal Pictures", I selected pictures of myself and asked my family and friends for face pictures in order to simulate a real-world inpainting application where there is little control over the input images. Although these collected images present less variation in terms of age, ethnicity, background and accessories when compared to the FFHQ images from the previous subsections, they cover a wider spectrum of picture lighting, contrast and image resolution and pixel density (in the original files). Also, these images are not subject to any of the standards and alignment procedures used to build CelebA-HQ and FFHQ datasets, forming an unbiased set of images to assess the two models.

In general, due to the generally poorer quality of the input images, the results from subsection IV-D "Personal Pictures" were the worst among all test cases, for both CelebA-HQ and FFHQ models. The models often produced unrealistic facial structures—such as in images 4.5 and 4.6 from figure 9—, or excessively blurred results—as seen in images 4.4 from figure 9. However, results improved when the input image had clear lighting and good color contrast, without strong shadows in the face, which can be observed in images 4.1 to 4.3 from figures 9. In other words, the resulting images were significantly better when the input image lighting, alignment and colors were similar to those from the CelebA-HQ and FFHQ datasets.

Therefore, for a real-world inpainting application, when using a model trained on a standardized dataset of aligned face images such as CelebA-HQ or FFHQ, the creation of the input image needs to be a controlled process so that results can be satisfactory. This can be done, for example, by building a photo booth to take the input pictures, where the camera model and lenses, the camera distance to the person, the environment lighting and the face alignment are all parameters that can be precisely controlled in such a way that the pictures taken in the photo booth follow similar standards as those from the dataset on which the model was trained, which tends to lead to better results.

Conclusion

In conclusion, upon the detailed evaluation presented in this section, the main objective of this project was successfully achieved, as our DeepFillv2 model trained on the FFHQ dataset was able to produce better results than the existing CelebA-HQ and Places2 DeepFillv2 models. Finally, by improving the state-of-the-art DeepFillv2 method using the FFHQ dataset, I hope that the work from this project can serve as a first step towards the development of a real-world application that employs facial inpainting to help people and promote their inclusion and well-being.

REFERENCES

[1] A. Criminisi, P. Pérez, and K. Toyama, "Region filling and object removal by exemplar-based image inpainting," *IEEE Transactions on image processing*, vol. 13, no. 9, pp. 1200–1212, 2004.

[2] M. Khodadadi and A. Behrad, "Text localization, extraction and inpainting in color images," in *20th Iranian Conference on Electrical Engineering (ICEE2012)*. IEEE, 2012, pp. 1035–1040.

[3] E. Park, J. Yang, E. Yumer, D. Ceylan, and A. C. Berg, "Transformation-grounded image generation network for novel 3d view synthesis," in *Proceedings of the ieee conference on computer vision and pattern recognition*, 2017, pp. 3500–3509.

[4] J. Xie, L. Xu, and E. Chen, "Image denoising and inpainting with deep neural networks," *Advances in neural information processing systems*, vol. 25, pp. 341–349, 2012.

[5] Y. Chen, R. Ranftl, and T. Pock, "A bi-level view of inpainting-based image compression," *arXiv preprint arXiv:1401.4112*, 2014.

[6] V. Setlur, S. Takagi, R. Raskar, M. Gleicher, and B. Gooch, "Automatic image retargeting," in *Proceedings of the 4th international conference on Mobile and ubiquitous multimedia*, 2005, pp. 59–68.

[7] A. Levin, A. Zomet, S. Peleg, and Y. Weiss, "Seamless image stitching in the gradient domain," in *European Conference on Computer Vision*. Springer, 2004, pp. 377–389.

[8] B. Faria Silva and G. Sugahara Faustino, "Algoritmo para confecção de próteses nasais," São Paulo, Brazil, 2019.

[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] 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.

[11] O. Le Meur and C. Guillemot, "Super-resolution-based inpainting," in *European Conference on Computer Vision*. Springer, 2012, pp. 554–567.

[12] 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.

[13] 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.

[14] 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. CVPR 2001*, vol. 1. IEEE, 2001, pp. I–I.

[15] 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.

[16] J. Shen, S. H. Kang, and T. F. Chan, "Euler's elastica and curvature-based inpainting," *SIAM journal on Applied Mathematics*, vol. 63, no. 2, pp. 564–592, 2003.

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

[18] J. Mairal, M. Elad, and G. Sapiro, "Sparse representation for color image restoration," *IEEE Transactions on image processing*, vol. 17, no. 1, pp. 53–69, 2007.

[19] B. Shen, W. Hu, Y. Zhang, and Y.-J. Zhang, "Image inpainting via sparse representation," in *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2009, pp. 697–700.

[20] M. Bertalmio, L. Vese, G. Sapiro, and S. Osher, "Simultaneous structure and texture image inpainting," *IEEE transactions on image processing*, vol. 12, no. 8, pp. 882–889, 2003.

[21] O. Le Meur, J. Gautier, and C. Guillemot, "Examplar-based inpainting based on local geometry," in *2011 18th IEEE international conference on image processing*. IEEE, 2011, pp. 3401–3404.

[22] V. Jain and S. Seung, "Natural image denoising with convolutional networks," *Advances in neural information processing systems*, vol. 21, pp. 769–776, 2008.

[23] 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.

[24] C. Yang, X. Lu, Z. Lin, E. Shechtman, O. Wang, and H. Li, "High-resolution image inpainting using multi-scale neural patch synthesis," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 6721–6729.

[25] 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.

[26] R. A. Yeh, C. Chen, T. Yian Lim, A. G. Schwing, M. Hasegawa-Johnson, and M. N. Do, "Semantic image inpainting with deep generative models," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 5485–5493.

[27] 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.

[28] 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.

[29] K. Nazeri, E. Ng, T. Joseph, F. Z. Qureshi, and M. Ebrahimi, "Edgeconnect: Generative image inpainting with adversarial edge learning," *arXiv preprint arXiv:1901.00212*, 2019.

[30] J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu, and T. S. Huang, "Free-form image inpainting with gated convolution," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 4471–4480.

[31] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," *arXiv preprint arXiv:1406.2661*, 2014.

[32] J. Jam, C. Kendrick, K. Walker, V. Drouard, J. G.-S. Hsu, and M. H. Yap, "A comprehensive review of past and present image inpainting methods," *Computer Vision and Image Understanding*, p. 103147, 2020.

[33] Z. Wang and A. C. Bovik, "A universal image quality index," *IEEE signal processing letters*, vol. 9, no. 3, pp. 81–84, 2002.

[34] Z. Wang, E. P. Simoncelli, and A. C. Bovik, "Multiscale structural similarity for image quality assessment," in *The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003*, vol. 2. Ieee, 2003, pp. 1398–1402.

[35] H. R. Sheikh and A. C. Bovik, "Image information and visual quality," *IEEE Transactions on image processing*, vol. 15, no. 2, pp. 430–444, 2006.

[36] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training gans," *Advances in neural information processing systems*, vol. 29, pp. 2234–2242, 2016.

[37] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "Gans trained by a two time-scale update rule converge to a local nash equilibrium," *Advances in neural information processing systems*, vol. 30, 2017.

[38] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 586–595.

[39] C. Li and M. Wand, "Precomputed real-time texture synthesis with markovian generative adversarial networks," in *European conference on computer vision*. Springer, 2016, pp. 702–716.

[40] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in *European conference on computer vision*. Springer, 2016, pp. 694–711.

[41] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida, "Spectral normalization for generative adversarial networks," *arXiv preprint arXiv:1802.05957*, 2018.

[42] S. Xie and Z. Tu, "Holistically-nested edge detection," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1395–1403.

[43] @xhlulu, "Flickr-faces-hq dataset (nvidia) - resized 256px," Feb 2020. [Online]. Available: https://www.kaggle.com/xhlulu/flickrfaceshq-dataset-nvidia-resized-256px

[44] P. O. Antonacio, "Completing face pictures: a study on image and facial inpainting methods," Dec 2021. [Online]. Available: https://github.com/PedroAntonacio/facial-inpainting