PEDRO ORII ANTONACIO

COMPLETING FACE PICTURES: A STUDY ON IMAGE AND FACIAL INPAINTING METHODS

São Paulo
2021

PEDRO ORII ANTONACIO

COMPLETING FACE PICTURES: A STUDY ON IMAGE AND FACIAL INPAINTING METHODS

Trabalho de Conclusão de Curso apresentado à Escola Politécnica da Universidade de São Paulo para obtenção do Título de Engenheiro Mecatrônico.

São Paulo
2021

PEDRO ORII ANTONACIO

COMPLETING FACE PICTURES: A STUDY ON IMAGE AND FACIAL INPAINTING METHODS

Trabalho de Conclusão de Curso apresentado à Escola Politécnica da Universidade de São Paulo para obtenção do Título de Engenheiro Mecatrônico.

Área de Concentração:

Engenharia Mecatrônica e de Sistemas Mecânicos

Orientadora:

Prof.ª Dr.ª Larissa Driemeier

São Paulo
2021

For Mari and Paulo

ACKNOWLEDGMENTS

Firstly, I would like to thank Prof.ª Dr.ª Larissa Driemeier for all the guidance, support and time dedicated to me both throughout the course of this project and also over the past years as one the finest professors I had during my engineering studies.

I would also like to express my deepest appreciation for my parents Mari and Paulo, who early on taught me to do what is right, and for 22 years relentlessly strived to offer me the best education and opportunities possible. You are undoubtedly the main responsible for the man I am today, and I dedicate this project to you.

I wish to thank my sister Maria Paula for always standing by my side and making me a better person as we grow together.

I would like to thank my nearest and dearest friends from Baita and Oshiman for sending me the face pictures I needed for this project in the last minute and, most importantly, for accompanying me for all those years. Alongside my dear classmates, you made this arduous journey feel much lighter, more enjoyable and worthwhile.

Finally, I wish to extend my special thanks to all professors, employees and fellow students from the Department of Mechatronics Engineering and Mechanical Systems and the Polytechnic School of the University of São Paulo who in any way contributed, directly or indirectly, to my education and development during my years as an engineering student.

Thank you all very much.

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, the main contributions of this project are: to provide a historic overview on image and facial inpainting techniques, to present a thorough review of the most relevant digital inpainting methods proposed in the last couple decades, and 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.

RESUMO

Desde o início dos anos 2000, muitas pesquisas foram conduzidas sobre métodos digitais de *image inpainting*, que encontraram diversas aplicações nos campos de processamento de imagem e visão computacional. Com o aumento no poder de processamento dos computadores e o desenvolvimento de *datasets* de imagens de alta qualidade, diversos novos métodos de *inpainting* com Deep Learning foram propostos nos últimos anos, obtendo resultados extraordinários, sem precedentes nos métodos tradicionais. Por exemplo, o DeepFillv2 é um completo e robusto modelo baseado em GAN, considerado um dos mais práticos e bem estabelecidos métodos de *image inpainting* desenvolvidos até os dias atuais. Contudo, para o caso de uso de *facial inpainting*, o DeepFillv2 foi inicialmente treinado com o CelebA-HQ, um *dataset* de imagem composto exclusivamente de rostos de celebridades, sendo assim fortemente enviesado para rostos brancos e atraentes, majoritariamente de personalidades adultas do mundo ocidental. Como alternativa, este projeto propõe treinar o modelo DeepFillv2 do zero usando o FFHQ, um *dataset* de rostos de pessoas normais do mundo todo, que oferece uma maior variabilidade de idade, etnia e plano de fundo nas imagens, sendo mais de duas vezes maior que o CelebA-HQ. Portanto, as principais contribuições deste projeto são: fornecer um panorama geral da história das técnicas de *image* e *facial inpainting*, apresentar uma revisão dos métodos digitais de *image inpainting* mais relevantes propostos nas duas últimas décadas e melhorar o método do estado da arte DeepFillv2 para o caso de uso de *facial inpainting* através dos seus dados de treinamento. Avaliações qualitativas e quantitativas sugerem que nosso modelo DeepFillv2 treinado no *dataset* FFHQ obteve melhores resultados quando comparado ao modelo DeepFilllv2 originalmente treinado no CelebA-HQ.

Palavras-Chave – Image Inpainting; Facial Inpainting; Deep Learning; Generative Adversarial Network (GAN); DeepFillv2.

LIST OF FIGURES

1.1 Artistically reconstructed portrait photographs, performed by illustrator Michelle Spalding. Source: Stewart [1] 14

1.2 From left to right: Arnolfini Portrait, by Jan van Eyck (1434); Mona Lisa, by Leonardo da Vinci (1503); portrait miniature of George Washington, by Robert Field (1800). Sources: Wikimedia Foundation and The National Gallery [2–4] 15

1.3 Hand inpainting performed by a professional art conservator. Source: Thottam [5] 16

1.4 The original negative and print (left) followed by a retouched negative and print (right). Source: Schriever [6] 17

1.5 Demonstration of neck retouching technique. Source: Schriever [6] 17

1.6 On the left, Kliment Voroshilov, Vyacheslav Molotov, Joseph Stalin, and Nikolai Yezhov walking along the banks of the Moscow-Volga Canal, in April, 1937. On the right, Nikolai Yezhov has been removed from the original image. Source: Gessen [7] 18

1.7 Common types of distortion tackled by image and facial inpainting algorithms. Source: Elharrouss *et al.* [8] 19

1.8 State-of-the-art image inpainting algorithm for object removal. Source: adapted from Zeng *et al.* [9] 19

1.9 The Night Watch with the AI-reconstructed edges attached to it, on display at the Rijksmuseum in Amsterdam. Source: [10] 20

2.1 Classification of digital image inpainting methods. Source: created by the author 21

2.2 Texture synthesis through Image Quilting. (a) Square blocks from the input texture are randomly chosen and patched together. (b) Some overlap is introduced to the blocks and they are rearranged so as to better "agree" with their neighbours. (c) Finally, the boundary between blocks is determined by finding a minimum cost path through the error surface at the overlap. Source: Efros and Freeman [11] 23

2.3 Structure propagation through exemplar-based texture synthesis proposed by Criminisi *et al.* [12]. (a) Input image with target (masked) region. (b) First chosen patch based on its priority level. (c) Most likely candidates to fill the chosen patch. (d) Most likely candidate copied to the filling area. Source: Criminisi *et al.* [12] 24

2.4 PatchMatch structure synthesis. (a) Original image. (b) User marks a hole (magenta) and line constraints (red/green/blue) to elucidate the continuity of the roofline; (c) The hole is filled in. (d) User inputs line constraints for retargeting, (e) which eliminates two columns automatically. (f) Reshuffling: user translates the roof upward. Source: Barnes *et al.* [13] 25

2.5 PatchMatch guided image inpainting. (a-c) User applies a mask to the bird and marks constraints on the input image for completion. Pedestal and gateway removal with (d-f) and without (g-i) user-supplied constraints. Source: Barnes *et al.* [13] 25

2.6 Inpainting results with the proposed FMM-based method. On the right, a close-up of the inpainted eye is presented. Source: Telea [14] 26

2.7 Results comparison between methods by Bertalmio *et al.* [15] (center) and Telea [14] (right and bottom). Source: Telea [14] 26

2.8 Inpaiting results by the proposed patch-wise sparse representation method. Source: Shen *et al.* [16] 27

2.9 (a) Masked images. (b) Inpainting results by the proposed hybrid method, and the results by the original methods (c) by Tschumperlé *et al.* [17] and (d) by Criminisi *et al.* [12]. Source: Le Meur *et al.* [18] 28

2.10 The basic structure of a Generative Adversarial Network (GAN). Source: Hitawala [19] 31

2.11 Context encoder trained with joint reconstruction and adversarial loss for semantic inpainting. Source: Pathak *et al.* [20] 32

2.12 Semantic Inpainting results for Context Encoder trained using reconstruction and adversarial loss. Source: Pathak *et al.* [20] 33

2.13 Proposed framework overview. Source: Yang *et al.* [21] 34

2.14 Results from the proposed method, Context Encoder and PatchMatch. Image size is 512×512 with a 256×256 missing hole. Source: adapted from Yang *et al.* [21] 34

2.15 Graphical illustration of standard and dilated convolution. Source: Li [22] 35

2.16 Overview of the proposed architecture with two context discriminator networks. Source: Iizuka *et al.* [23] 35

2.17 Image completion results by the method proposed by Iizuka *et al.*. Source: Iizuka *et al.* [23] 35

2.18 Inpainting results for the proposed model trained separately on scenery images (b), and on face images (c). Source: adapted from Iizuka *et al.* [23] 36

2.19 Proposed inpainting framework. The Coarse Network is explicitly trained with reconstruction loss, while the Refinement Network is trained with reconstruction loss and global and local WGAN-GP adversarial loss. Source: Yu *et al.* [24] 37

2.20 On the left, an illustration of the Contextual Attention layer. On the right, the Refinement Network architecture with two parallel encoders. In the attention map, colors indicate relative locations of the most interested background patch for each pixel in foreground. Source: adapted from Yu *et al.* [24] 37

2.21 Results by the proposed method on natural scene, face, and texture images. Source: Yu *et al.* [24] 38

2.22 Results comparison between some of the discussed inpainting methods. Source: adapted from Liu *et al.* [25] 38

2.23 Inpainting results by the proposed partial convolution-based method. Source: Liu *et al.* [25] 39

2.24 Proposed method, in which the masked grayscale image, the mask and a prior edge map are the inputs of *G₁* to predict the full edge map. Predicted edge map and incomplete color image are passed to *G₂* to perform the inpainting task. Source: Nazeri *et al.* [26] 39

2.25 Ground Truth (left). Masked input images (center left). Edge map: black edges are previously computed using Canny edge detector and blue edges are predicted in the generator first stage for the missing regions (center right). Inpainting results of the proposed method (right). Source: Nazeri *et al.* [26] 40

2.26 Comparison between Partial Convolutions (left) and Gated Convolutions (right). Source: Yu *et al.* [27] 41

2.27 Overview of DeepFillv2 framework with Gated Convolutions and SN-PatchGAN discriminator for free-form image inpainting. Source: Yu *et al.* [27] 41

2.28 Examples of inpainting results by DeepFillv2. Source: Yu *et al.* [27] 42

2.29 User-guided inpainting of faces and natural scenes. Source: Yu *et al.* [27] . 42

2.30 Qualitative comparison between results from DeepFillv2 and other state-of-the-art inpainting methods. Source: Yu *et al.* [27] 42

2.31 Example images from PASCAL VOC 2007. Source: Everingham *et al.* [28] 45

2.32 A snapshot of the Vehicle Subtree from ImageNet. Source: Deng *et al.* [29] 46

2.33 Example images from Paris Street View. Source: Doersch *et al.* [30] 46

2.34 Example images from the CelebA dataset. Source: Liu *et al.* [31] 46

2.35 Example images from the CelebA-HQ dataset. Source: Karras *et al.* [32] . 47

2.36 Example images from the Places2 dataset. Source: Zhou *et al.* [33] 47

2.37 Example masks from the testing set of NVidia Irregular Mask Dataset, with different hole-to-image area ratios. Source: Liu *et al.* [25] 48

2.38 Example masks from Quick Draw Irregular Mask Dataset. Source: Iskakov *et al.* [34] 48

2.39 Example images from the Flickr-Faces-HQ (FFHQ) dataset. Source: Karras *et al.* [35] 49

3.1 Examples of masks automatically generated during the DeepFillv2 model training process on the FFHQ dataset. Source: created by the author . . . 58

3.2 DeepFillv2 loss function computed on the validation set during training iterations and the corresponding smoothed curved. Source: created by the author 59

4.1 Model progress during the training process. Source: created by the author 61

4.2 Models comparison: images 2.1 to 2.10. Source: created by the author . . . 62

4.3 Models comparison: images 2.11 to 2.21. Source: created by the author . . 63

4.4 Models comparison: images 2.22 to 2.32. Source: created by the author . . 64

4.5 Edge case: strong facial expressions. Source: created by the author 65

4.6 Edge case: accessories. Source: created by the author 66

4.7 Edge case: painted faces. Source: created by the author 66

4.8 Edge case: unnatural hair colors. Source: created by the author 67

4.9 Edge case: babies. Source: created by the author 67

4.10 Edge case: challenging backgrounds. Source: created by the author 68

4.11 Edge case: side view of faces. Source: created by the author 68

4.12 Edge case: large masks. Source: created by the author 69

4.13 Edge case: human statues. Source: created by the author 69

4.14 Personal pictures: images 4.1 to 4.10. Source: created by the author 70

4.15 Personal pictures: images 4.11 to 4.20. Source: created by the author . . . 71

LIST OF TABLES

2.1 Summary of the reviewed traditional methods for image inpainting. Source: adapted from Jam *et al.* [36] 29

2.2 Summary of the reviewed Deep Learning methods for image inpainting. Source: adapted from Jam *et al.* [36] 44

2.3 Summary of the discussed datasets. Source: adapted from Jam *et al.* [36] and Elharrouss *et al.* [8] 49

4.1 Average Structural Similarity Index Measure (SSIM) values calculated for the resulting images from each test case. Source: created by the author . . 72

4.2 Average Peak Signal-to-Noise Ratio (PSNR) values calculated for the resulting images from each test case. Source: created by the author 72

CONTENTS

1 Introduction 14

A Brief History of Image and Facial Inpainting 15

2 State of the Art 21

2.1 Traditional Methods 21

2.1.1 Exemplar-based Methods 22

2.1.2 Diffusion-based Methods 25

2.1.3 Sparse Representation Methods 27

2.1.4 Hybrid Methods 27

Summary: Traditional Methods 28

2.2 Deep Learning Methods 30

2.2.1 Convolution Neural Network-based Methods 30

2.2.2 Generative Adversarial Network-based Methods 31

Summary: Deep Learning Methods 43

2.3 Datasets 45

Summary: Datasets 49

3 Materials and Methods 50

3.1 Project Objective 50

3.2 Evaluation Methods for Image Inpainting 51

3.2.1 Quantitative Metrics 51

3.2.2 Qualitative Evaluation 53

3.3 The DeepFillv2 Model 54

3.3.1 Gated Convolutions 54

3.3.2 Network Architecture 56

3.3.3 Loss Function 57

3.3.4 Free-Form Mask Generation Algorithm 57

3.3.5 Extension to User-Guided Image Inpainting 58

3.4 Implementation 58

3.4.1 Dataset 58

3.4.2 Model Training 59

4 Results 60

4.1 Model Progress During Training 61

4.2 Models Comparison 62

4.3 Edge Cases 65

4.3.1 Strong Facial Expressions 65

4.3.2 Accessories 66

4.3.3 Painted Faces 66

4.3.4 Unnatural Hair Colors 67

4.3.5 Babies 67

4.3.6 Challenging Backgrounds 68

4.3.7 Side View of Faces 68

4.3.8 Large Masks 69

4.3.9 Human Statues 69

4.4 Personal Pictures 70

4.5 Quantitative Metrics 72

5 Discussion 73

References 77

1 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. Figure 1.1 shows examples of damaged portrait photographs reconstructed with facial inpainting techniques.

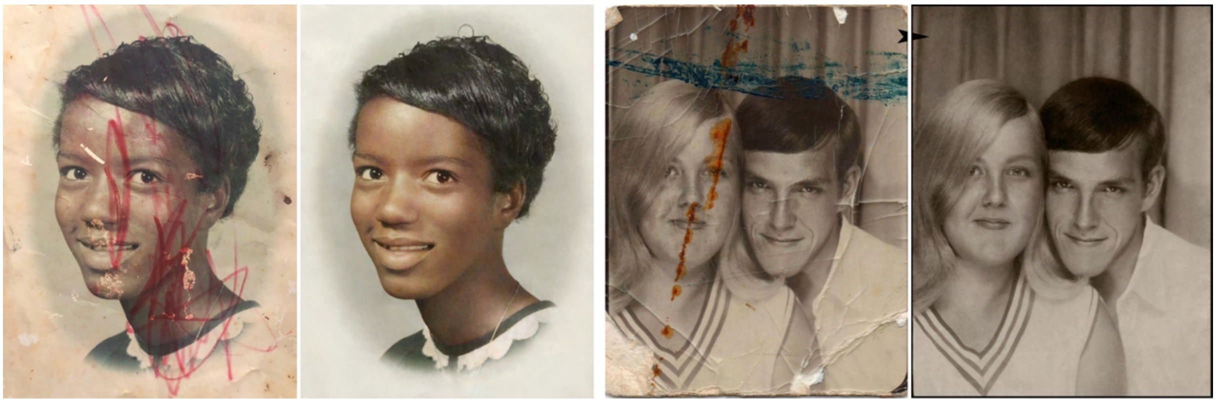


Figure 1.1: Artistically reconstructed portrait photographs, performed by illustrator Michelle Spalding. Source: Stewart [1]

In summary, the main contributions intended with this project are the following:

- Provide a historic overview on image and facial inpainting techniques, their origins and development until this day.
- Present a thorough review of the most prominent and relevant digital image and facial inpainting methods proposed in the last decades.

- Attempt to improve a well-established state-of-the-art Deep Learning image inpainting method for the use case of facial inpainting through its training data, and then present and discuss the obtained results.

A Brief History of Image and Facial Inpainting

Even though facial and image inpainting are currently active fields of research with numerous recent applications in image processing and computer vision, these concepts have long been explored by artists and scientists.

During most of Modern History, paintings have been one of the primary forms of artistic expression and historical record. In Renaissance, with the rise of Individualism and the greater focus on individuality, paintings depicting humans as the central subject—or portrait paintings—became increasingly popular. Although initially restricted to the rich and powerful, over time portrait paintings gradually spread across middle-class people mainly in the form of portrait miniatures [37]. Until this day, many of those portrait paintings remain as important historical sources. Figure 1.2 shows examples of famous portrait paintings in Art History.

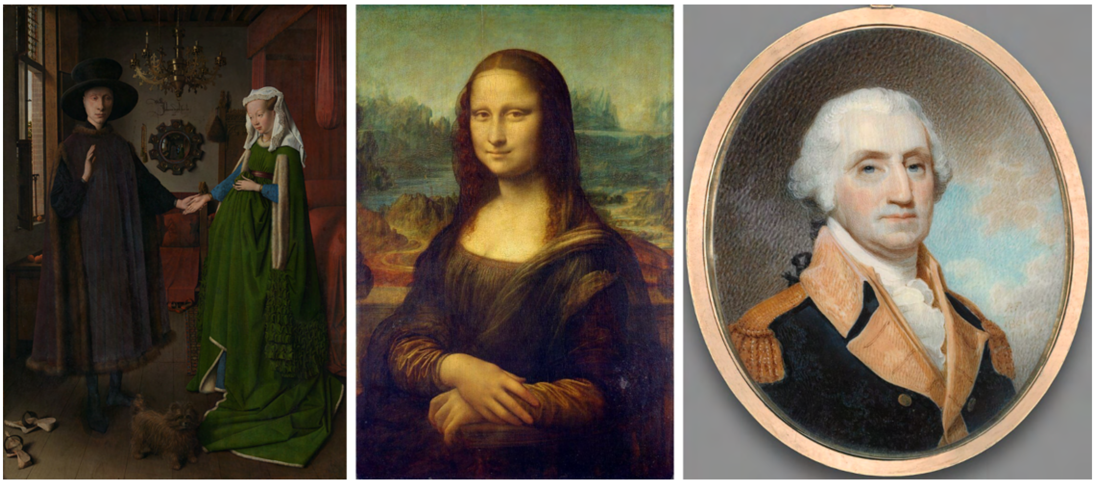


Figure 1.2: From left to right: Arnolfini Portrait, by Jan van Eyck (1434); Mona Lisa, by Leonardo da Vinci (1503); portrait miniature of George Washington, by Robert Field (1800). Sources: Wikimedia Foundation and The National Gallery [2–4]

With the invention and the development of film photography in the 19th and 20th centuries [38], portrait photographs became widely accessible to the population and quickly turned into one of the most common ways of registering individual and family moments or social and professional events. Nowadays, portrait photographs not only serve as fundamental records of human history, but also became indispensable personal items in almost every household.

However, portrait paintings and photographs exist on physical mediums that inevitably get damaged over time due to environmental conditions such as relative humidity, temperature, light, pollutants and even pests like rodents and insects. In this way, photography prints and paintings' canvases and oil paints deteriorate and degrade with time, accumulating dirt and dust in their surface and causing defects such as spots and cracks. Furthermore, human-induced factors like storage in improper environments, poor handling practices and physical bumps and scratches can also cause permanent damages to paintings and photographs.

For those reasons, introducing adjustments to artworks in a way that they remain unnoticed to the observer unaware of the original image is a practice that has been around since the origins of art creation itself. Medieval paintings started to be restored during Renaissance with the objective to bring them "up to date", in a process that was called *retouching* or *inpainting* [39, 40]. Since the first documented artwork restoration occurrence in the early 16th century [41], the disciplines of conservation and restoration of works of art—and, later, photographs as well—have evolved significantly [42,43] and today most major museums have dedicated scientific laboratories with advanced equipment like X-ray machines and infrared cameras to perform tasks like image inpainting [5].

Nevertheless, artwork inpainting is a time-consuming and labour-intensive activity mainly comprised of manual work and that requires specialized professional conservators. Smaller paintings usually take two to three weeks to restore and cost U$800 to U$1,000, whereas large paintings could cost U$10,000 to U$15,000 and take months or, in some cases, years to complete [5]. As a result, inpainting of artworks is an expensive and inaccessible process. Figure 1.3 illustrates an artwork inpainting process.

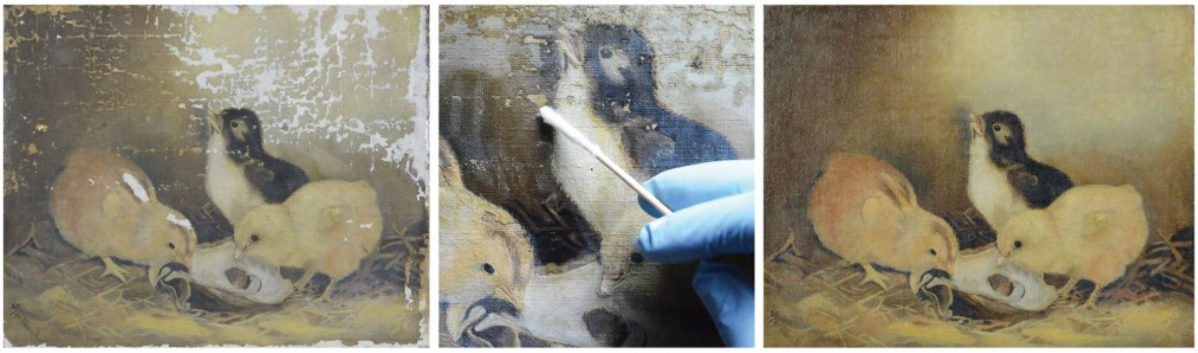


Figure 1.3: Hand inpainting performed by a professional art conservator. Source: Thottam [5]

In addition to artwork conservation and restoration, image manipulation has also employed image inpainting techniques in a process called photographic retouching, in which photographs' negatives were physically altered to produce desired changes on the original image. The first known act of photographic retouching occurred in 1846 [44], just five years after the patent of the calotype—the first practical photographic process capable of generating multiple prints from a single negative. Since then, photographic retouching became increasingly common, specially in portrait photographs. In 1909, Schriever published a complete guide [6] detailing the best tools and practices for retouching photographs, from the etching knife usage to the retouching desk positioning and the chemicals' preparation. Figures 1.4 and 1.5 show instructive retouching examples from Schriever's guide.

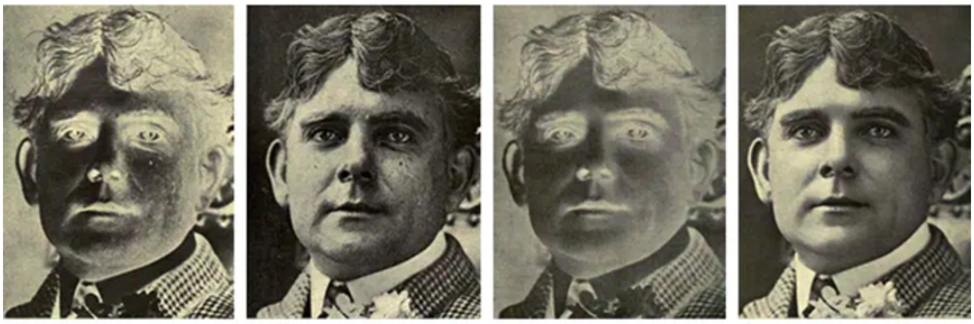


Figure 1.4: The original negative and print (left) followed by a retouched negative and print (right). Source: Schriever [6]

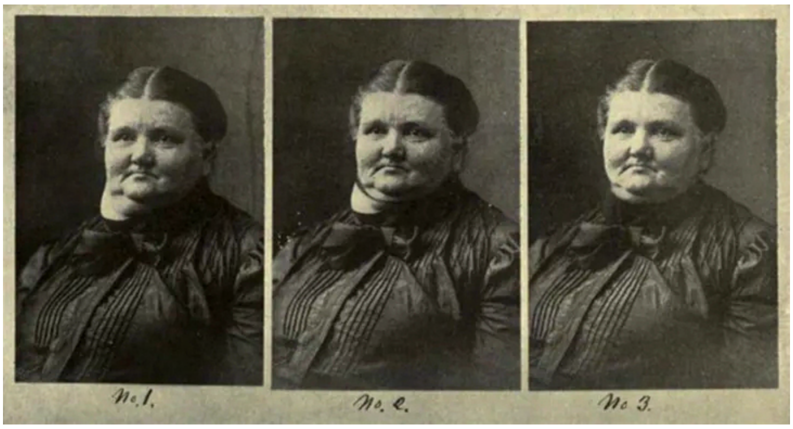


Figure 1.5: Demonstration of neck retouching technique. Source: Schriever [6]

In the first half of the 20th century, from Hollywood celebrity pictures to ordinary family portraits, photographic retouching was a widespread practice. To fit into the prevailing beauty standards, noses, jaws, ears, shoulders and waists were often reshaped while wrinkles, blotches and freckles were constantly smoothed out, consolidating a photo editing culture that prevails until this day.

At the same time, photographic retouching was also put to use in document falsification and political propaganda. In the late 1930s, for example, photographic retouchers in Soviet Russia spent long hours physically removing Stalin's political opponents from official photographs, which helped to produce a false and manipulated perception of reality among the Russian population. Figure 1.6 shows an example of retouched photograph from Stalin. As King [45] describes:

> Photographs for publication were retouched and restructured with airbrush and scalpel to make once famous personalities vanish. Paintings, too, were often withdrawn from museums and art galleries so that compromising faces could be blocked out of group portraits. [...] A parallel industry came into full swing, glorifying Stalin as the ”great leader and teacher of the Soviet people” through socialist realist paintings, monumental sculpture, and falsified photographs representing him as the only true friend, comrade, and successor to Lenin, the leader of the Bolshevik Revolution and founder of the USSR. The whole country was subjected to this charade of Stalin-worship.

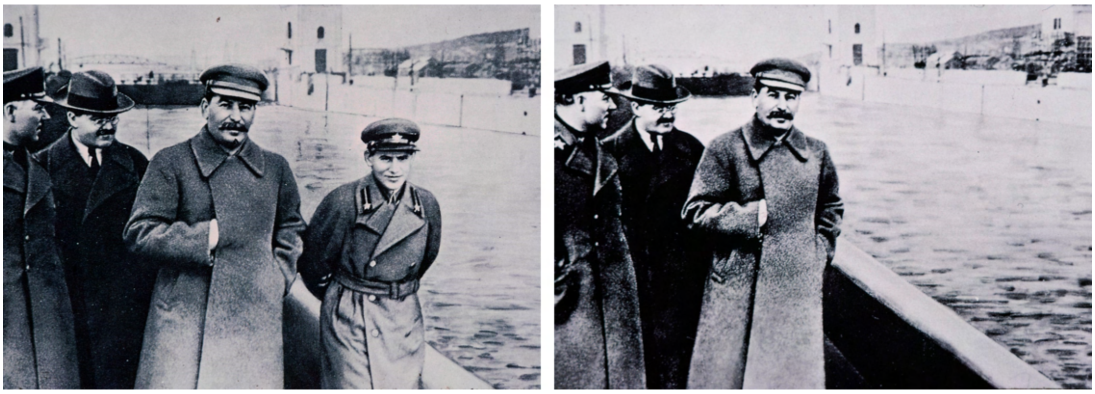


Figure 1.6: On the left, Kliment Voroshilov, Vyacheslav Molotov, Joseph Stalin, and Nikolai Yezhov walking along the banks of the Moscow-Volga Canal, in April, 1937. On the right, Nikolai Yezhov has been removed from the original image. Source: Gessen [7]

With the Digital Revolution in the second half of the 20th century and the subsequent mass availability of digital computers and smartphones in the last decades, images left the physical mediums of painting canvasses and photography prints and started to be stored digitally as long sequences of 0's and 1's. This shift not only solved the conservation problem but also paved the way for numerous new applications that were previously unthinkable.

Nowadays, image restoration and manipulation processes can be easily carried out by a much less specialized public through photo-editing softwares, at a fraction of the cost and time in relation to the physical inpainting process. However, these processes still require dedicated human interaction, which makes them hard to scale and prone to human errors. For those reasons, an interest on the development of automated digital image inpainting techniques emerged.

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 [12] and text overlays [46], image-based rendering [47] and image denoising [48], compression [49], retargeting [50] and compositing [51]. More recently, Faria Silva *et al.* [52] proposed the use of inpainting techniques to optimize the production process of nasal prostheses. Figures 1.7 and 1.8 show common use cases in which digital image and facial inpainting algorithms can be applied.

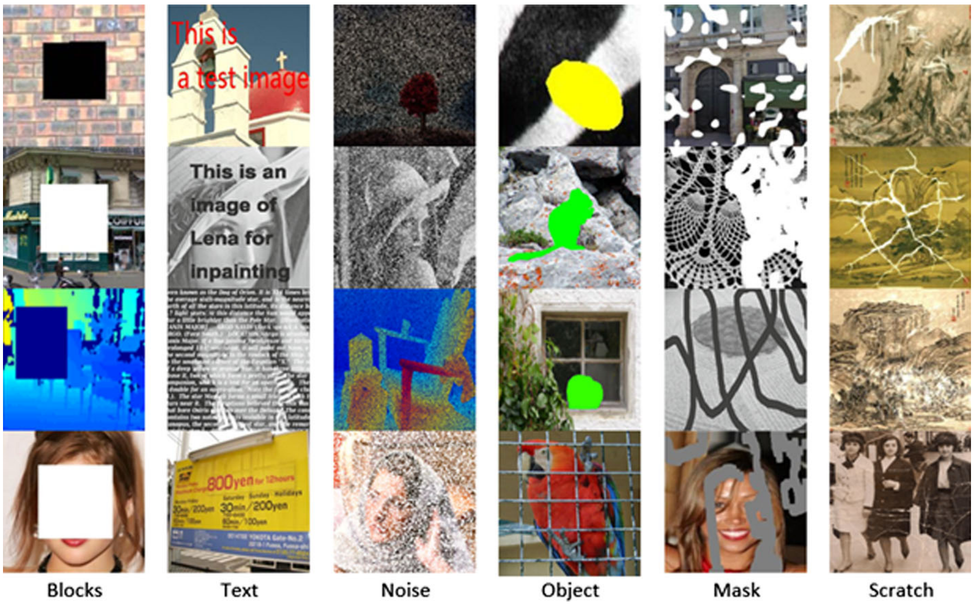


Figure 1.7: Common types of distortion tackled by image and facial inpainting algorithms. Source: Elharrouss *et al.* [8]

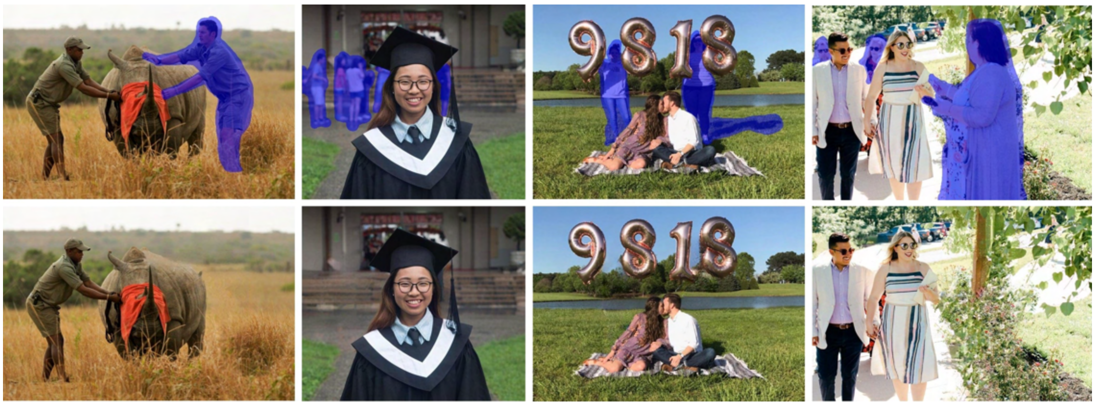


Figure 1.8: State-of-the-art image inpainting algorithm for object removal. Source: adapted from Zeng *et al.* [9]

In June 2021, an Artificial Intelligence (AI) algorithm was employed to restore the missing parts of Rembrandt's famous painting The Night Watch [53, 54]. The canvas, painted in 1642 with a size of 363 cm × 437 cm, had its edges trimmed in 1715 to fit between two doors at Amsterdam's city hall. For over 300 years, the painting has been missing 60 cm from the left, 7 cm from the right, 22 cm from the top and 12 cm from the bottom. Using a 17$^{\text{th}}$-century small-scale copy of the original canvas painted by Gerrit Lundens as a basis, the AI was able to restore the missing edges on The Night Watch while preserving Rembrandt's original painting style and colors—which would not be possible if the restoration process was conducted by a human artist. Figure 1.9 shows Rembrandt's The Night Watch with the edges restored by the AI algorithm.

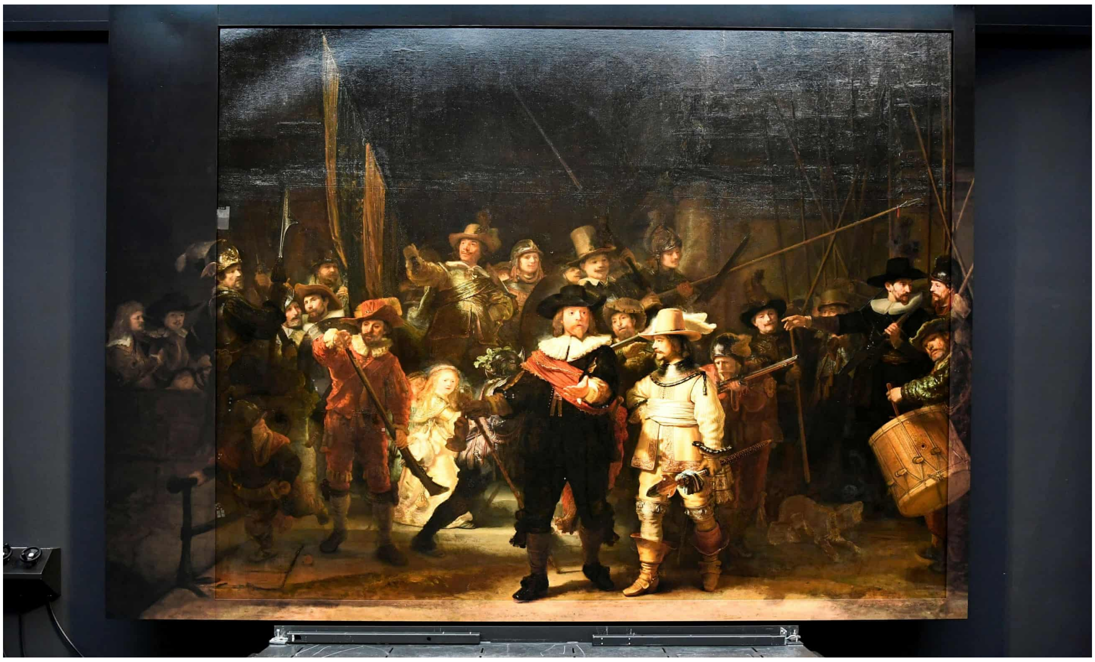


Figure 1.9: The Night Watch with the AI-reconstructed edges attached to it, on display at the Rijksmuseum in Amsterdam. Source: [10]

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. The next chapter will present and discuss the current state-of-the-art image and facial inpainting methods.

2 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 2.1 presents a classification of the digital image inpainting methods that are going to be discussed in the following sections.

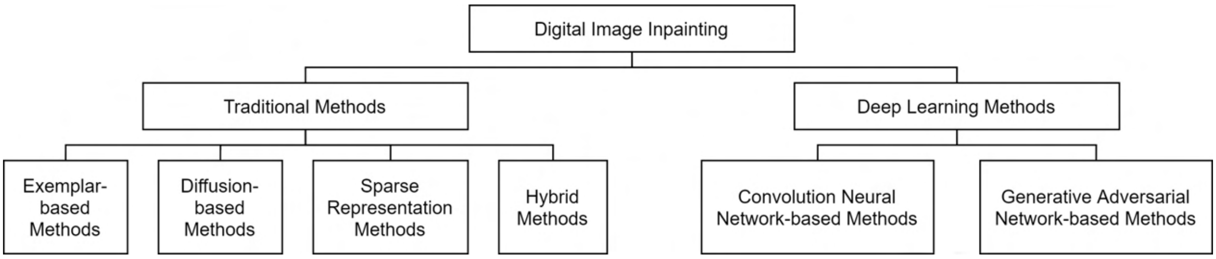


Figure 2.1: Classification of digital image inpainting methods. Source: created by the author

2.1 Traditional Methods

The term ”image inpainting” in reference to a digital image processing algorithm was first used in 2000 by Bertalmio *et al.* [15], who pioneered diffusion-based inpainting methods. But before reviewing the above inpainting methods in further detail, the concepts of texture and structure should be defined more precisely. As Jam *et al.* [36] define:

> A **texture** is a visual pattern on an infinite 2-D plane with a stationary distribution at some scale [55]. This pattern refers to the feel (smooth, rough) of the image surface. Textures are either regular (repeated texels, or texture elements) or stochastic (imprecise texels) and can be synthesised based on the assumption that the sample is large and uniform with known statistics of regular patterns [56]. A geometric texture of an image is the entire representation as a texture based on statistical details of which a small patch is sufficiently a representative [57]. In textural inpainting, the available data considered for the inpainting task are exemplar textures. Textural inpainting uses statistical knowledge of patterns due to its stationary distribution of missing regions and known parts of the image, commonly modelled by Markov Random Fields [55].

> The **structure** of an image is a visual object constructed by distinct parts (global contour information) of the image texture [58]. The geometric structure of an image is a representation of composition and structure. During inpainting, the geometric structure has a low dimensionality representation in subspace. That is, the coordinates of the inpainted region are exact representations of the subspace and do not exceed its dimension. This is because it must satisfy the coordinate vertices of the image representation before decomposition to yield an approximate representation of the parent structure. With this technique, the target region does not exceed the parent structure, and the outcome is a good representation of the global context. In structural inpainting, taking account the nature of the smoothness in the missing regions and the boundary conditions is a precondition which uses either isotropic diffusion or anisotropic diffusion to propagate boundary data in the isotropic direction [15].

Effective image inpainting methods must be able to synthesize both texture and structures successfully based on the known pixels of the masked image. The following sections highlight 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.

2.1.1 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.

Exemplar-based Texture Synthesis

Efros and Leung [55] pioneered exemplar-based texture synthesis methods in 1999 through the use of Markov Random Fields (MRF) to model the image texture—that is, the probability distribution of brightness values for each pixel within a neighbourhood. The proposed method is iterative and completes the missing regions one pixel at a time, by selecting a neighbourhood around a missing pixel and then querying the know parts of the image to find one that is the most similar the the selected neighbourhood. Finally, the central pixel from the found part is then copied to the location of the missing pixel. This method produces effective results, but it is sensible to the window size of the selected neighborhood and can generate discontinuous and undesirable patterns.

In 2001, Efros and Freeman [11] proposed Image Quilting, a fast, stable and simple method for texture synthesis and transfer. This method synthesises textures in blocks—instead of one pixel at a time—by overlapping and merging patches sampled from the input texture that satisfy the overlap constraints within some error tolerance. However, because patch sizes do not always match the frequency of the details in the input texture and because patches are randomly selected from the set of blocks that satisfy the constraints, this method can generate discontinuous and misaligned results. Figure 2.2 illustrates the proposed Image Quilting method.

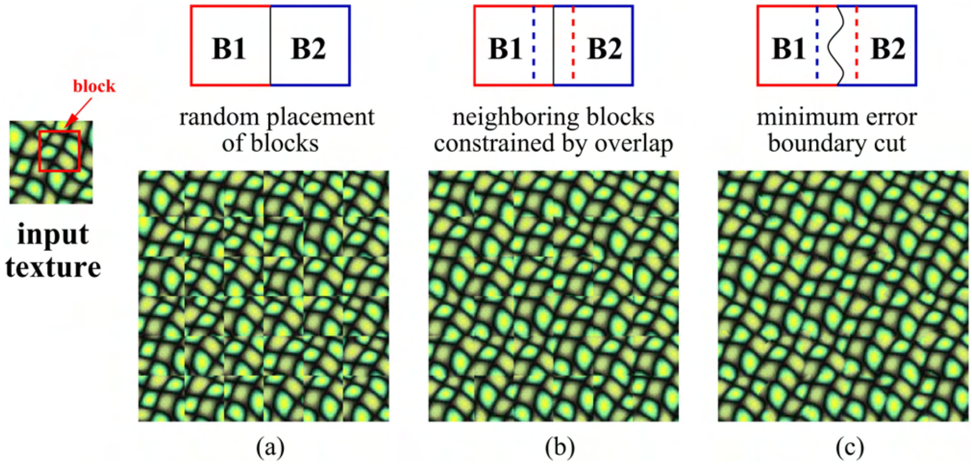

B1
B2
random placement of blocks
neighboring blocks constrained by overlap
minimum error boundary cut
block
input texture
(a)
(b)
(c)

Figure 2.2: Texture synthesis through Image Quilting. (a) Square blocks from the input texture are randomly chosen and patched together. (b) Some overlap is introduced to the blocks and they are rearranged so as to better ”agree” with their neighbours. (c) Finally, the boundary between blocks is determined by finding a minimum cost path through the error surface at the overlap. Source: Efros and Freeman [11]

In 2012, Le Meur *et al.* [59] improved upon Efros and Leung's [55] method by introducing K-Nearest Neighbour (KNN), K-coherence candidate SSD and the Battacharya distance [60] to the selection of the most similar patch. Such improvements reduce computational complexity and make the method less sensitive to noise, allowing it to focus on the dominant orientations of the image structures. Finally, a super-resolution algorithm is applied to enhance the details in the inpainted image. However, this method depends on proper parameter tuning to produce high-quality results and the super-resolution algorithm increases its overall computational cost.

In essence, 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.

Exemplar-based Structure Synthesis

In 2004, Criminisi *et al.* [12] used existing patch-wise exemplar-based texture synthesis methods in combination with a confidence mechanism which prioritizes the filling of regions in the direction of propagation of linear structures. In that way, linear structures are first propagated into the target masked area and then texture is synthesised according to the structural constraints. Figure 2.3 shows the proposed structure propagation method. This method is able to preserve both texture and structure, but cannot handle curved structures and is dependant on accurate priority pixel values to produce satisfactory results.

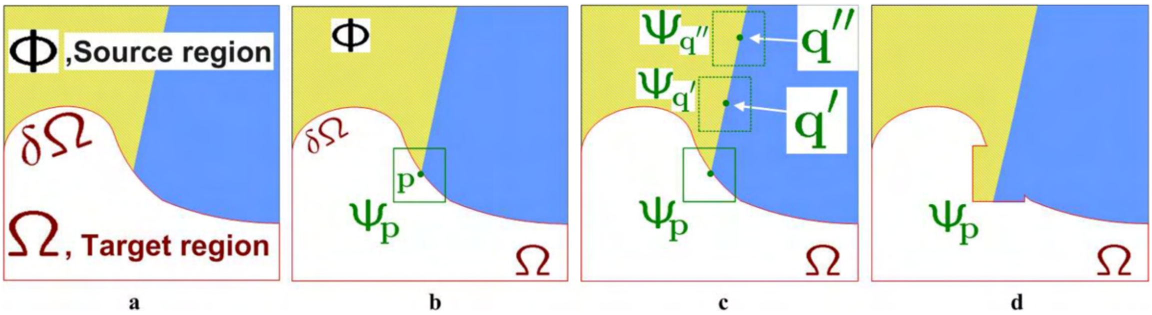
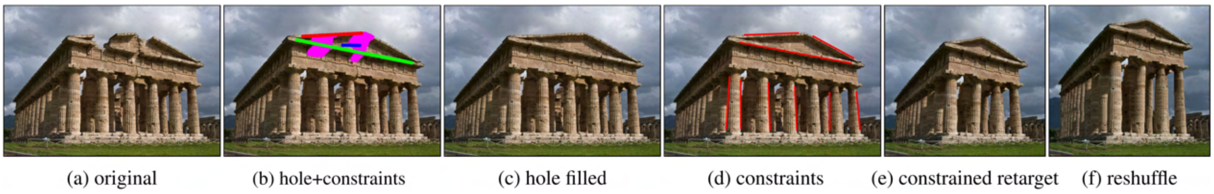


Figure 2.3: Structure propagation through exemplar-based texture synthesis proposed by Criminisi *et al.* [12]. (a) Input image with target (masked) region. (b) First chosen patch based on its priority level. (c) Most likely candidates to fill the chosen patch. (d) Most likely candidate copied to the filling area. Source: Criminisi *et al.* [12]

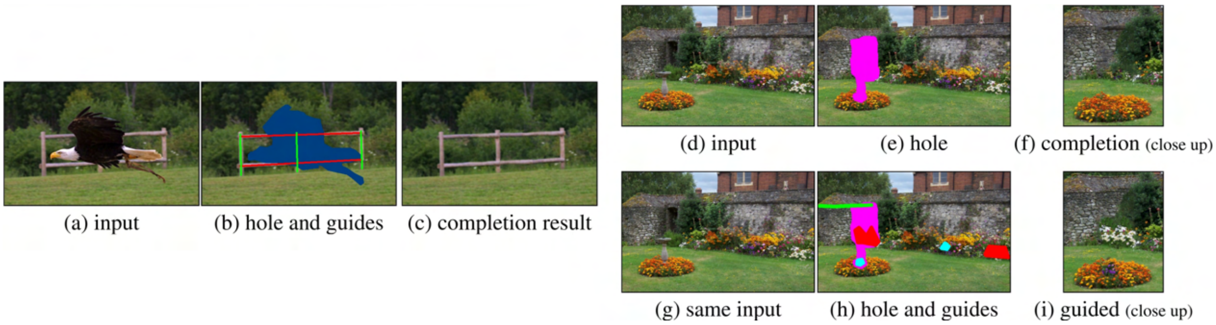
In 2009, Barnes *et al.* [13] introduced PatchMatch, an interactive image editing tool for image retargeting, completion and reshuffling, which uses Nearest Neighbour Fields (NNF) and a fast randomised sampling algorithm to quickly find approximate nearest-neighbor matches between image patches. This method is significantly faster than previous approaches and successfully restores structures and textures even with large missing areas, while also allowing for user inputs to produce more accurate results. By the time it was introduced, PatchMatch produced unprecedented results and represented a major breakthrough in image inpainting methods. It was implemented as an editing tool in Adobe Photoshop CS5 and until this it is regarded as the state-of-the-art technique for traditional non-learning methods. Figures 2.4 and 2.5 show some of PatchMatch results.

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.



(a) original (b) hole+constraints (c) hole filled (d) constraints (e) constrained retarget (f) reshuffle

Figure 2.4: PatchMatch structure synthesis. (a) Original image. (b) User marks a hole (magenta) and line constraints (red/green/blue) to elucidate the continuity of the roofline; (c) The hole is filled in. (d) User inputs line constraints for retargeting, (e) which eliminates two columns automatically. (f) Reshuffling: user translates the roof upward. Source: Barnes *et al.* [13]



(a) input (b) hole and guides (c) completion result

(d) input (e) hole (f) completion (close up)

(g) same input (h) hole and guides (i) guided (close up)

Figure 2.5: PatchMatch guided image inpainting. (a-c) User applies a mask to the bird and marks constraints on the input image for completion. Pedestal and gateway removal with (d-f) and without (g-i) user-supplied constraints. Source: Barnes *et al.* [13]

2.1.2 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.

In 2000, Bertalmio *et al.* [15] proposed an inpainting method that used the concept of isophotes (curves of constant brightness in the image) and a set of Partial Differential Equations (PDE) to automatically fill the missing regions with the surrounding information along the isophotes directions, regardless of the shape or size of the mask. Bertalmio *et al.* improved this method in the following year by utilizing the Navier-Stokes equations for fluid dynamics to propagate image information [61]. The proposed algorithms perform well when inpainting small and thin masks, but lead to blurred results in large missing areas and struggle to complete images with multiple missing parts.

In 2001, Chan and Shen [62] proposed the PDE-based Curvature-Driven Diffusion (CDD) inpainting model that was able to consider the geometric information (or curvature) of isophotes for the diffusion process, fixing a limitation of the previous Total Variational (TV) method by the same authors [63] and allowing for the inpainting of larger missing areas. However, this method cannot inpaint textured images, once the statistical fluctuations in textures are smoothed out by the employed PDE's. The proposed method was further developed by Shen *et al.* [64] with their Euler's Elastica and Curvature-based inpainting model.

In 2004, Telea [14] combined the Fast Marching Method (FMM) from Sethian [65] with the PDE-based propagation of the image smoothness estimator along the image gradient from Bertalmio *et al.* [15] and proposed an inpainting method that was simpler and considerably faster than previous diffusion-based techniques. This method estimates image smoothness as a weighted average over a known neighborhood of the pixel to inpaint. Finally, image information is then propagated into the missing areas, which are treated as level sets for the FMM. Figure 2.6 shows some results by the proposed method and figure 2.7 presents a comparison between the proposed method and the one by Bertalmio *et al.* [15].

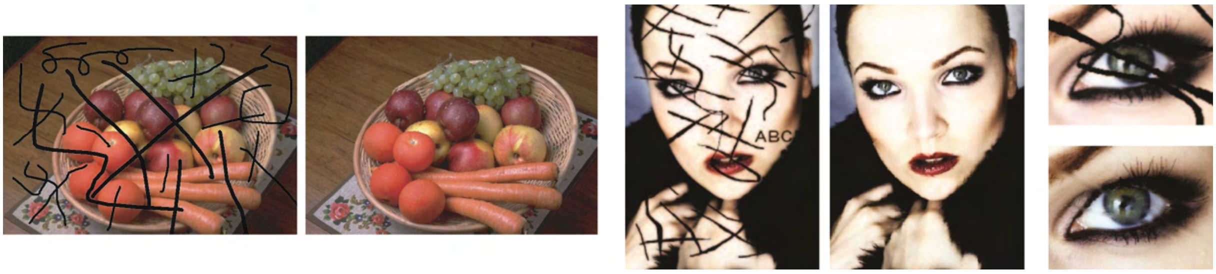


Figure 2.6: Inpainting results with the proposed FMM-based method. On the right, a close-up of the inpainted eye is presented. Source: Telea [14]

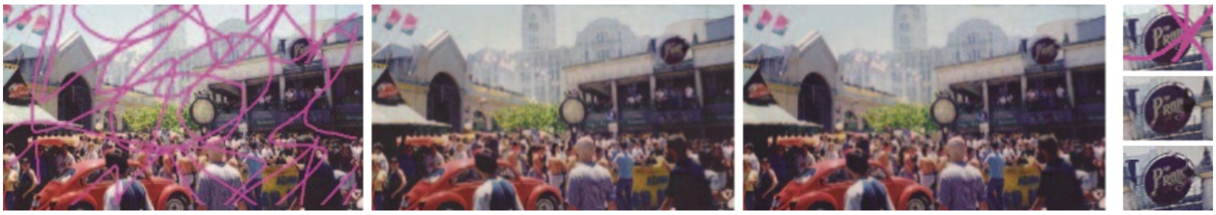


Figure 2.7: Results comparison between methods by Bertalmio *et al.* [15] (center) and Telea [14] (right and bottom). Source: Telea [14]

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.

2.1.3 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.

In 2007, Mairal *et al.* [66] introduced a dictionary learning algorithm for sparse decomposition of colored images with missing information, mainly focused on image denoising. In 2009, Shen *et al.* [16] improved upon that algorithm and proposed a patch-wise sparse representation method specifically targeted at image inpainting, allowing for user-defined masks and capable of preserving both texture and noise consistency in the inpainted result. Figure 2.8 presents a few inpainting results by the proposed method.

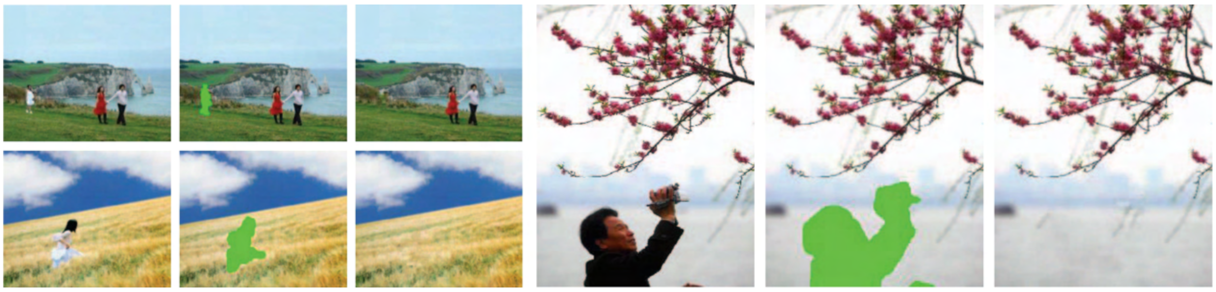


Figure 2.8: Inpaiting results by the proposed patch-wise sparse representation method. Source: Shen *et al.* [16]

2.1.4 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.

In 2003, Bertalmio *et al.* [58] proposed a method that used Efros and Leung [55] exemplar-based texture synthesis in association with their previous PDE-based inpainting algorithm from 2000 [15]. This method decomposes the masked image into texture and structure layers and then conducts the inpainting task on each layer separately, employing their diffusion-based technique on the structure layer and the exemplar-based method on the texture layer. Despite producing satisfactory results, this method is computationally expensive, can only inpaint black-and-white images, and struggles to complete large missing areas.

In 2011, Le Meur *et al.* [18] introduced a new inpainting method by combining the difusion-based image regularization PDE framework by Tschumperlé *et al.* [17] with the exemplar-based structure synthesis method by Criminisi *et al.* [12]. The proposed method is conducted in two steps: first, structure tensors are used to define the filling order priority to favor the structure propagation in the isophote direction. Then, a template matching is performed in order to find the best candidates to fill in the hole based on a KNN algorithm. Figure 2.9 shows a comparison between the inpainting results of the proposed hybrid methods and the original ones.

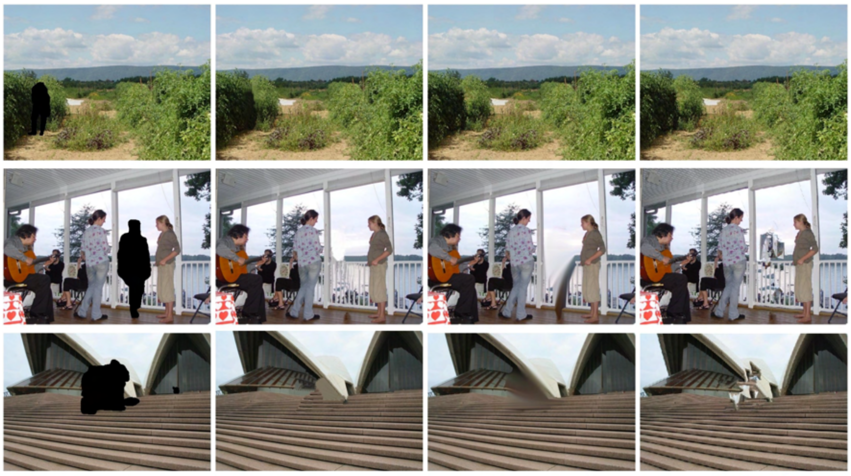

(a) Mask (b) Proposed (c) Tschumperlé (d) Criminisi

Figure 2.9: (a) Masked images. (b) Inpainting results by the proposed hybrid method, and the results by the original methods (c) by Tschumperlé *et al.* [17] and (d) by Criminisi *et al.* [12]. Source: Le Meur *et al.* [18]

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: Traditional Methods

In summary, 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 2.1 summarizes the discussed traditional image inpainting methods.

Category	**Methods**	**Advantages**	**Disadvantages**
Exemplar-based Texture Synthesis	Efros and Leung (1999) [55]; Efros and Freeman (2001) [11]; Le Meur *et al.* (2012) [59]	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) [12]; Barnes *et al.* (2009) [13]	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) [15]; Bertalmio *et al.* (2001) [61]; Chan and Shen (2001) [62]; Shen *et al.* (2003) [64]; Telea (2004) [14]	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) [66]; Shen *et al.* (2009) [16]	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) [58]; Le Meur *et al.* (2011) [18]	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.

Table 2.1: Summary of the reviewed traditional methods for image inpainting. Source: adapted from Jam *et al.* [36]

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.

2.2 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.

2.2.1 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.

CNNs were first used in image inpainting in 2008 by Jain *et al.* [67], who framed the image denoising computational task within the statistical framework of regression rather than density estimation. In that way, image denoising becomes a supervised learning problem that can be tackled with CNNs, using a training set composed of clean images and their corresponding copies with introduced Gaussian noise. However, this method is restricted to grayscale images and requires inpainting regions to be informed to the algorithm a priori (non-blind inpainting), besides having a high computational cost.

In 2012, Xie *et al.* [48] improved this method by approaching the inpainting task with a combination of sparse coding and deep neural networks pre-trained with a denoising auto-encoder. This approach reduced the computational cost of the algorithm and could automatically identify the pixels that needed to be inpainted (blind inpainting). However, this method still relies on supervised training, assuming access to a database of clean, noiseless images and their corrupted counterparts, and can only solve relatively small denoising/inpainting tasks in images with a controlled procedure of pixel corruption.

Other CNN-based blind inpainting approaches were proposed by Eigen *et al.* [68] in 2013 and Köhler *et al.* [69] in 2014, but they also suffer from the same problems mentioned above. In 2018, Liu *et al.* [25] introduced the concept of Partial Convolutions in a U-Net-based [70] network, a milestone method in image inpainting techniques. This CNN-based method is presented and discussed in further details in the next section.

2.2.2 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.* [71] 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. As stated in 2016 by Yann LeCunn, Facebook VP and Chief AI Scientist and Professor at NYU, adversarial training "is the most interesting idea in the last 10 years in Machine Learning" [72]. Figure 2.10 illustrates the basic structure of a GAN.

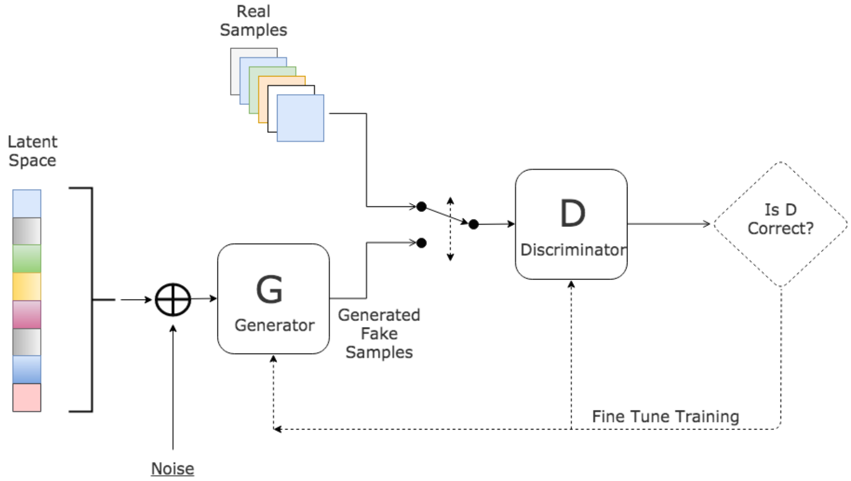


Figure 2.10: The basic structure of a Generative Adversarial Network (GAN). Source: Hitawala [19]

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.

In 2016, Pathak *et al.* [20] pioneered the use of GANs in image inpainting, by employing adversarial loss associated with *L2* reconstruction loss to train an autoencoder network for the task of image completion. Figure 2.11 shows the proposed network architecture. An autoencoder (or encoder-decoder) is a network architecture in which the input and the output are the same size, while intermediate layers have less dimensions. Autoencoders are typically aimed at learning a lower-dimensional latent feature representation of the data (encoder) and then reconstruct the input data based on the learned latent feature representation (decoder), thus reducing noise and focusing on the more relevant attributes of the data. In this case, the autoencoder is used to encode the main semantic features of the masked image—what the authors refer as ”context”—and then use those features to perform the inpainting task on the masked region.

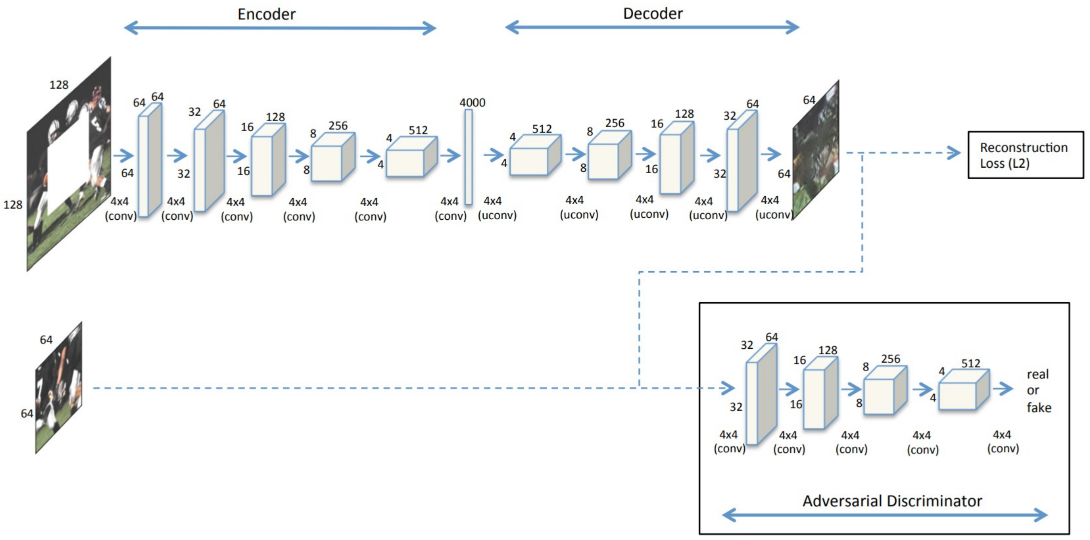


Figure 2.11: Context encoder trained with joint reconstruction and adversarial loss for semantic inpainting. Source: Pathak *et al.* [20]

The Context Encoder trained with joint adversarial and *L2* loss was able to achieve unprecedented results, successfully extracting semantic features and using them in the inpainting task. However, this method is limited to low-resolution images and often generates images with implausible structures or overly-smooth inpaintings. Moreover, if the position or size of the masks (that must be square shaped) changed, the autoencoder would have to be retrained and, for masks of different sizes, also redesigned. Lastly, when encoding the input image, the autoencoder also takes into consideration its masked white pixels, which can interfere in the feature extraction process and raise the method's computational cost. Figure 2.12 shows some of the results obtained by Pathak *et al.* [20].

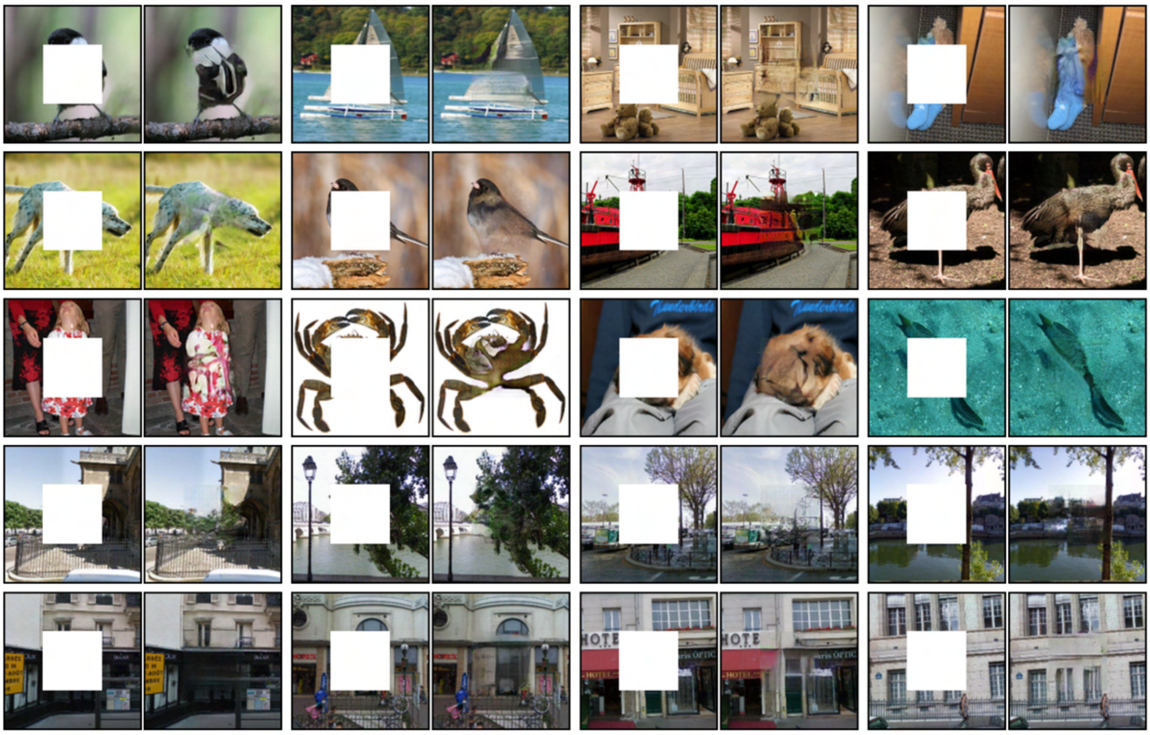


Figure 2.12: Semantic Inpainting results for Context Encoder trained using reconstruction and adversarial loss. Source: Pathak *et al.* [20]

In 2017, Yang *et al.* [21] improved upon the Context Encoder method by combining the Context Encoder architecture with style transfer techniques proposed in the prior year [73–75] as a way to introduce a "multi-scale neural patch synthesis approach based on joint optimization of image content and texture constraints". To do that, Yang *et al.* [21] proposed an inpainting framework with two networks: the Content Network—a slightly modified Context Encoder (also trained with a combination of *L2* and adversarial loss) to predict the global content and generate structurally consistent results—, and the Texture Network—a pre-trained image classification VGG-19 network to produce sharper results with high-frequency fine details. Figure 2.13 shows the proposed framework and figure 2.14 displays a results comparison between the proposed and other methods.

To work with high-resolution 512×512 images, this method downsizes the input image to 128×128 before the inpainting (size that the Context Encoder can handle) and then

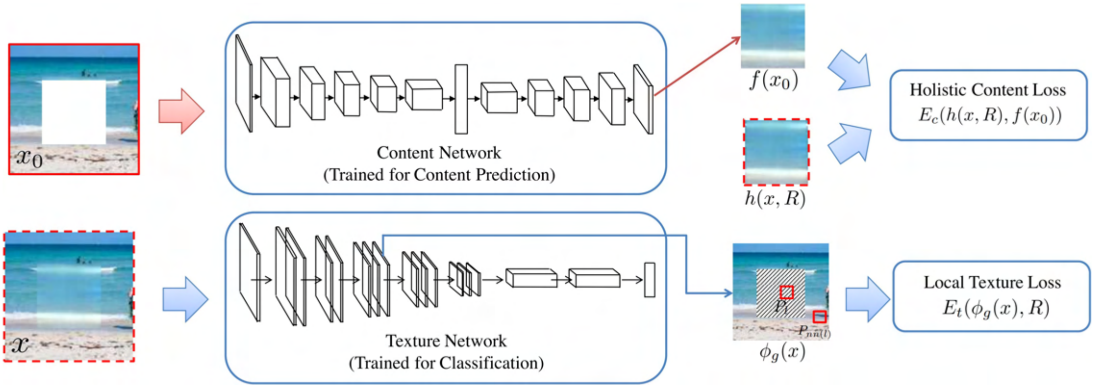


Figure 2.13: Proposed framework overview. Source: Yang *et al.* [21]

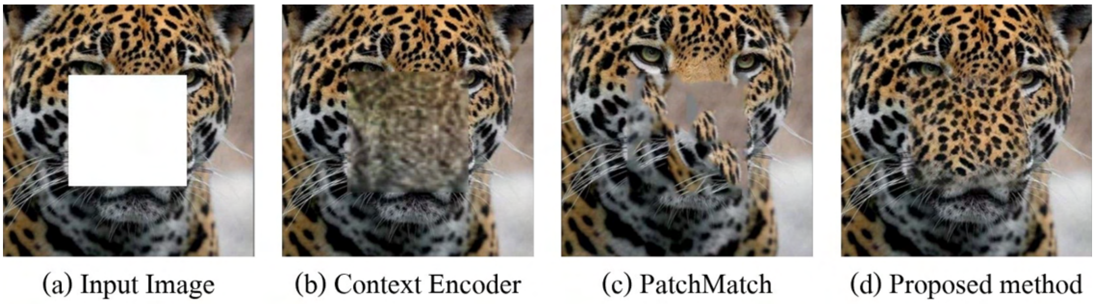

(a) Input Image (b) Context Encoder (c) PatchMatch (d) Proposed method

Figure 2.14: Results from the proposed method, Context Encoder and PatchMatch. Image size is 512×512 with a 256×256 missing hole. Source: adapted from Yang *et al.* [21]

upsamples the inpainted result back to 512×512 using bilinear interpolation. Although this method represents an improvement over Pathak *et al.* [20] in terms of the inpainted image texture, it often produces implausible structures and is computationally expensive for high-resolution images. Also, this method is still limited to rectangular shaped holes.

Also in 2017, Iizuka *et al.* [23] proposed the use of a fully convolutional network with dilated convolutions to extract image features instead of using fully-connected layers, which allows input images of arbitrary size and reduces the computational cost of the method. To ensure consistent structural and texture inpainting, the proposed method uses two discriminative models at different scales: one global and one local discriminator focused on the inpainted area. Figure 2.15 shows a graphical representation of the differences between dilated and standard convolution and figure 2.16 presents the proposed model architecture.

To increase visual coherency between original and generated pixels, Iizuka *et al.* [23] applied Fast Marching and Poisson image blending post-processing techniques on the inpainted images. Despite failing to accurately inpaint images with complex structures and textures, the proposed method achieved remarkable results in most inpainting cases when compared to other state-of-the-art techniques. Also, by supporting images of arbitrary

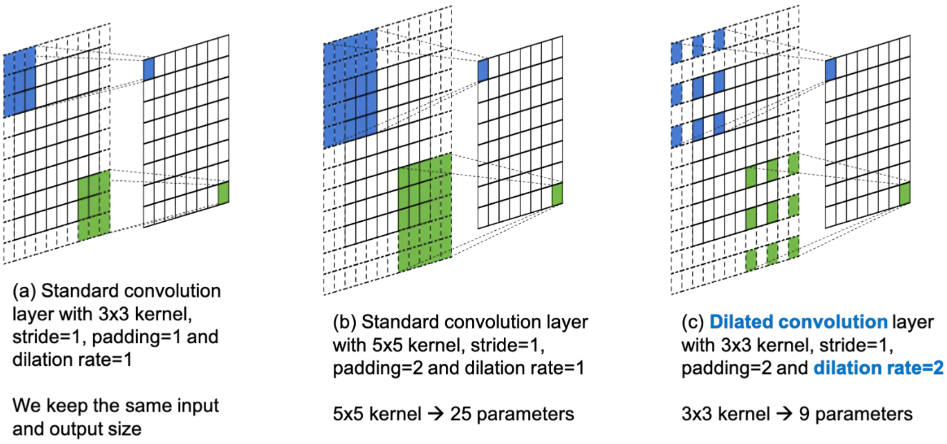

(a) Standard convolution layer with 3x3 kernel, stride=1, padding=1 and dilation rate=1
We keep the same input and output size
(b) Standard convolution layer with 5x5 kernel, stride=1, padding=2 and dilation rate=1
5x5 kernel → 25 parameters
(c) Dilated convolution layer with 3x3 kernel, stride=1, padding=2 and dilation rate=2
3x3 kernel → 9 parameters

Figure 2.15: Graphical illustration of standard and dilated convolution. Source: Li [22]

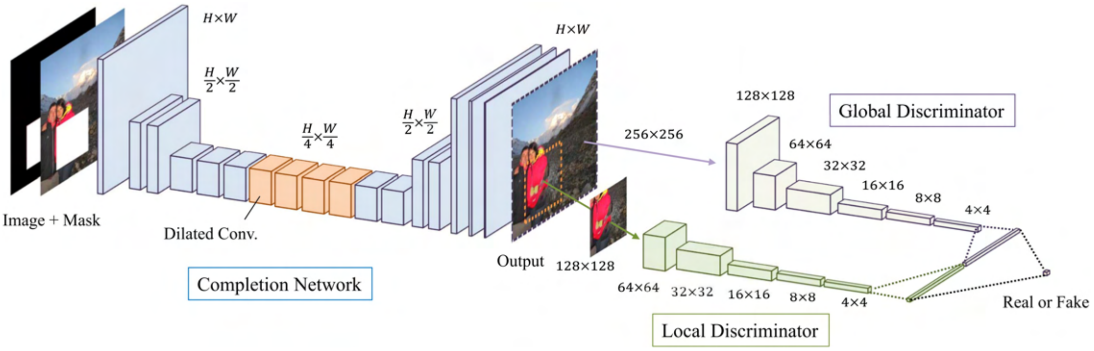


Figure 2.16: Overview of the proposed architecture with two context discriminator networks. Source: Iizuka *et al.* [23]

resolutions and missing regions of any shape, this method represented a milestone in deep image inpainting and formed the basis for the main GAN-based image inpainting approaches that were proposed afterwards. Figure 2.17 shows some of the results obtained by the proposed method.

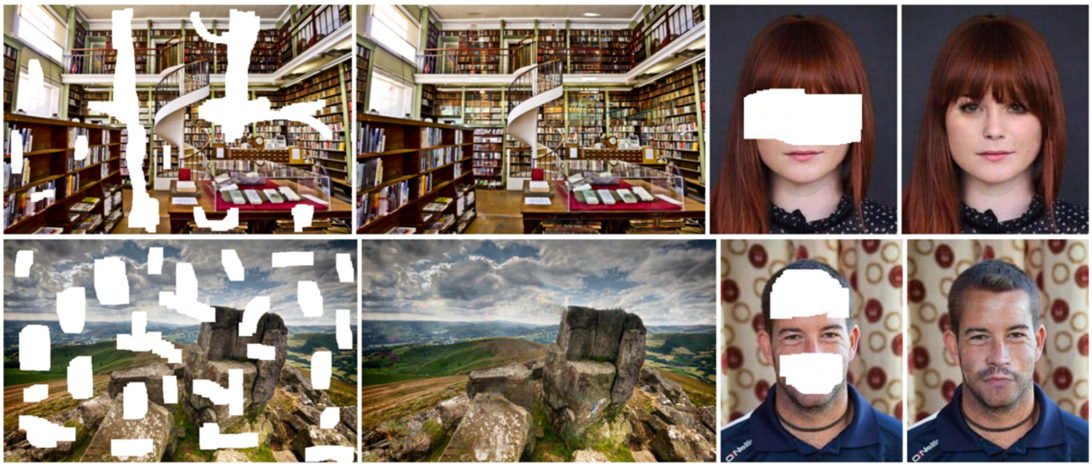


Figure 2.17: Image completion results by the method proposed by Iizuka *et al.*. Source: Iizuka *et al.* [23]

It is worth noting the influence that the training dataset has on the inpainted results. In figure 2.18, rows (b) and (c) show the inpainted results obtained by the proposed method when trained on different datasets. Model (b) was trained on Places2, a dataset of scenery pictures which does not contain any aligned face images. As model (b) never encountered aligned faces in its training process, it never learned to extract facial features and complete face images. Even though model (b) is the same as (c), the results are significantly more accurate and plausible with model (c) because it was trained on CelebA, a dataset comprised exclusively of aligned face images.

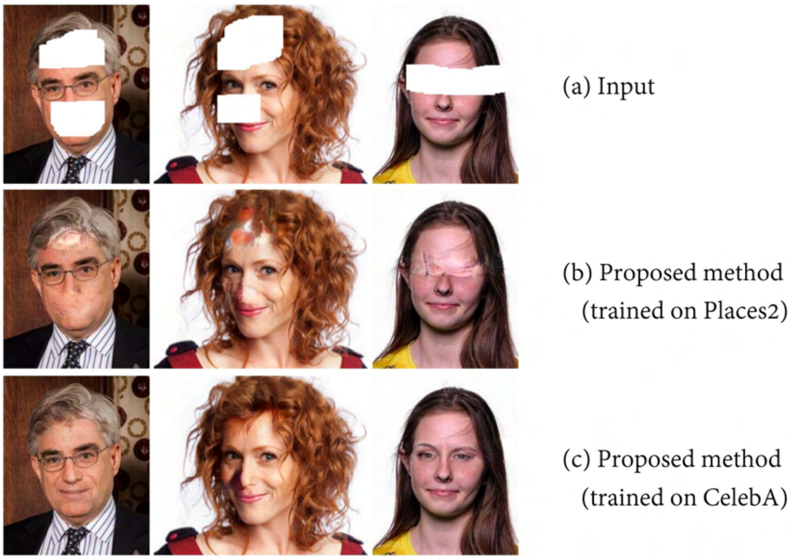


Figure 2.18: Inpainting results for the proposed model trained separately on scenery images (b), and on face images (c). Source: adapted from Iizuka *et al.* [23]

Still in 2017, Yeh *et al.* [76] introduced a new concept for GAN-based image inpainting. Instead of extracting information only from the masked image to be inpainted, the proposed a method also exploits the latent space representation of the data. After a deep generative model is trained through adversarial loss, this method searches for an encoding of the input image that is the "closest" to the image in the latent manifold—using the proposed context and prior losses. The searched encoding is then passed through the generative model to infer the missing content, regardless of the shape or size of the mask. Despite obtaining a good performance on the tested datasets, this method struggles when faced with image misalignment, complex scenes or higher resolution images.

In 2018, Yu *et al.* [24] proposed a method known as DeepFillv1, which uses a Contextual Attention mechanism to effectively borrow features from image regions that are spatially distant from the masked area for the inpainting task. To do that, the authors used a local and a global discriminator like Iizuka *et al.* [23], and introduced a two-stage feed-forward coarse-to-fine generative model. The first stage of the generator—the Coarse

Network—is responsible for a rough estimation of the missing region, while the second stage—the Refinement Network—uses the Contextual Attention to polish the rough estimation accordingly. Figure 2.19 shows the proposed model architecture and figure 2.20 details the Contextual Attention mechanism.

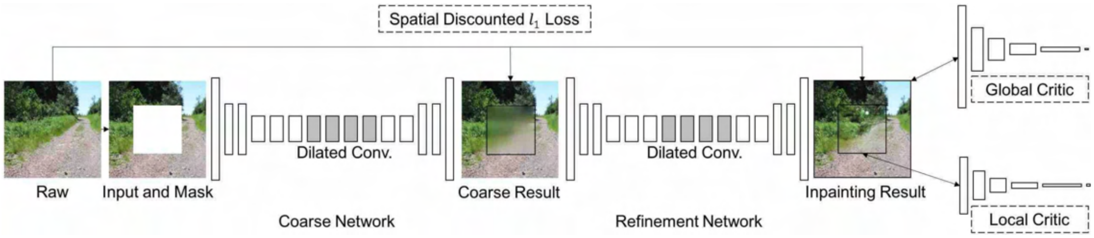


Figure 2.19: Proposed inpainting framework. The Coarse Network is explicitly trained with reconstruction loss, while the Refinement Network is trained with reconstruction loss and global and local WGAN-GP adversarial loss. Source: Yu *et al.* [24]

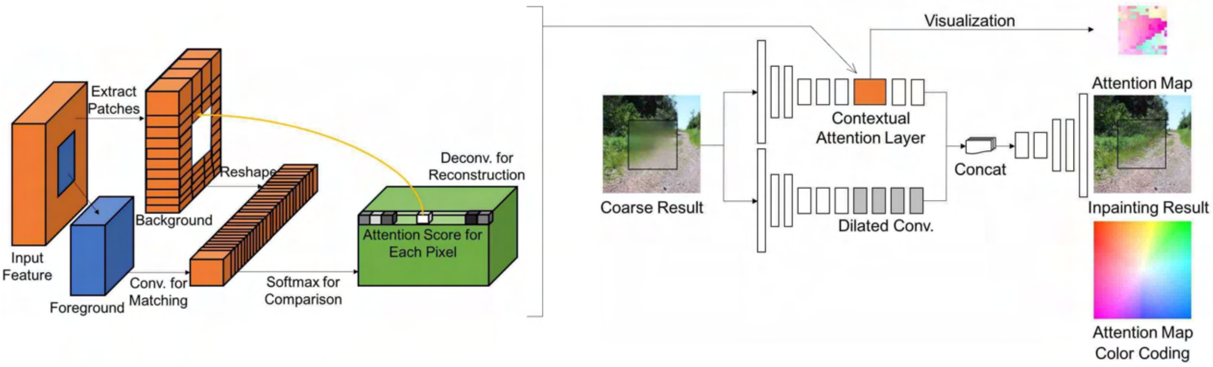


Figure 2.20: On the left, an illustration of the Contextual Attention layer. On the right, the Refinement Network architecture with two parallel encoders. In the attention map, colors indicate relative locations of the most interested background patch for each pixel in foreground. Source: adapted from Yu *et al.* [24]

The proposed model is able to inpaint multiple masks of arbitrary shapes and sizes, at any location on the image and generate more accurate structures and sharper textures that are more consistent with surrounding areas, in relation to previous methods. However, this method lacks fine texture details and generates some background inconsistencies on high resolution images. Some examples of inpainting results by the proposed model can be viewed on figure 2.21.

Also in 2018, Liu *et al.* [25] introduced Partial Convolutions to solve a major limitation from previous approaches. Up until this point, all GAN-based image inpainting methods treated valid pixels and masked pixels the same way, that is, the generative model computed convolutions on input masked image without making any distinction between valid original pixels and masked meaningless ones. To address this issue, Liu *et*

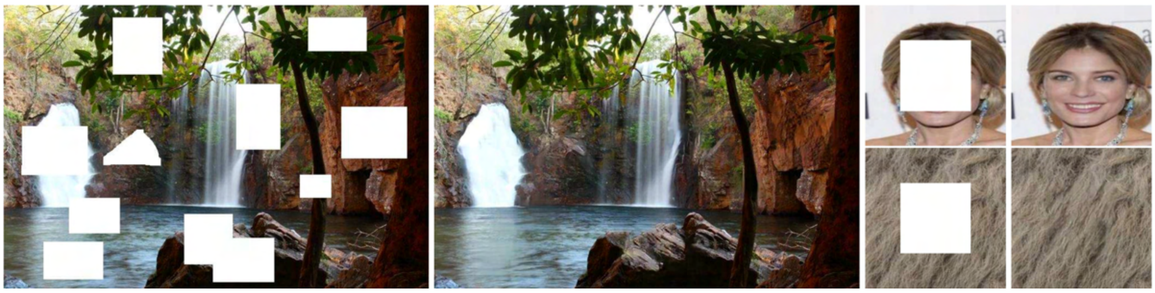


Figure 2.21: Results by the proposed method on natural scene, face, and texture images. Source: Yu *et al.* [24]

al. [25] used a U-Net-based [70] network replacing all the standard convolutions with the proposed Partial Convolutions—in which the convolution is masked and renormalized to be conditioned only on valid pixels. The authors also proposed a procedure to automatically generate an updated mask for the following layers as part of the forward pass, that is, to distinguish valid and invalid pixels at each Partial Convolution layer. Additionally, this procedure ensures that the input image undergoes a sufficient number of updates to eventually get rid of any masking on its encoded representation, regardless of the size, shape and relative position of the masks. However, this mask updating procedure between Partial Convolution layers is based on fixed hand-crafted rules, so it cannot be learned like model parameters during the training procedure.

Interestingly, this method does not use a discriminative model nor is it trained with adversarial loss. Liu *et al.* [25] used *L1* and total variation losses combined with to two high-level feature losses: a perceptual loss to reduce grid-shaped artifacts in the inpainted regions, and a style loss to generate fine local textures. Despite not being a GAN-based method and struggling with sparsely structured images and larger masks, this method produced exceptional inpainting results and introduced a way of disregarding masked pixels and focusing only on the valid ones, an important concept that was later explored by other inpainting techniques. Figure 2.22 presents a comparison with other discussed methods and figure 2.23 shows some of the results by the proposed method.

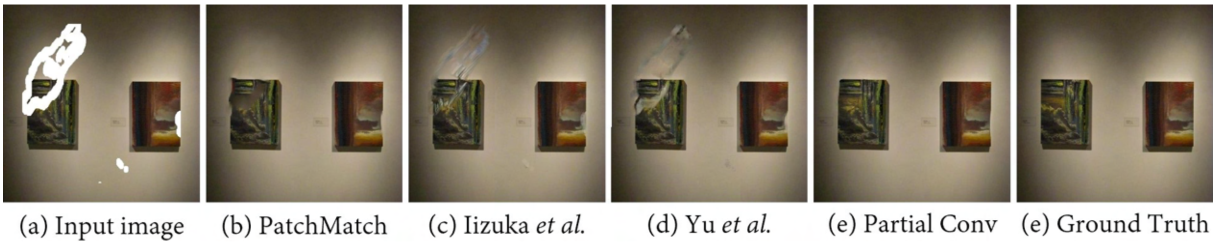


Figure 2.22: Results comparison between some of the discussed inpainting methods. Source: adapted from Liu *et al.* [25]

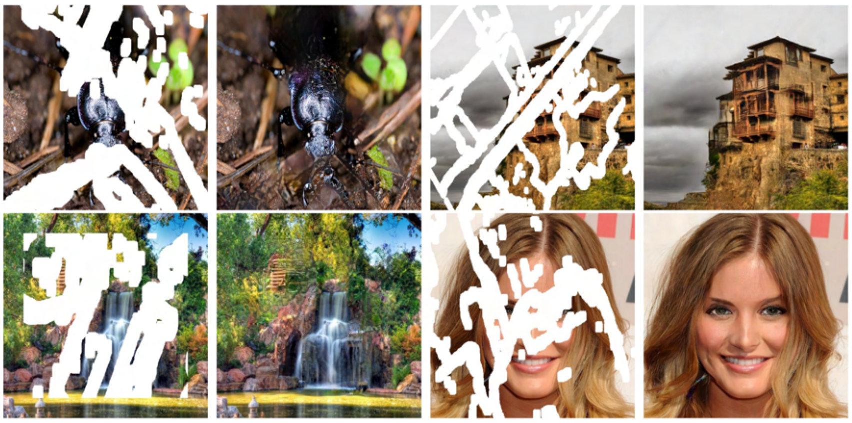


Figure 2.23: Inpainting results by the proposed partial convolution-based method. Source: Liu *et al.* [25]

In 2019, Nazeri *et al.* [26] introduced EdgeConnect, a two-stage adversarial model for image inpainting with similar generative architecture to the coarse-to-fine network from Yu *et al.* [24]. However, instead of global and local discriminators placed at the end of the generator, Nazeri *et al.* [26] used one discriminator for each of the generative stages. The first stage—trained with adversarial and feature matching losses—predicts an edge map of the missing regions, while the second stage—trained with style, perceptual, *L1* and adversarial losses—completes the image filling the predicted edges with texture and color. Figure 2.24 illustrates the proposed model architecture.

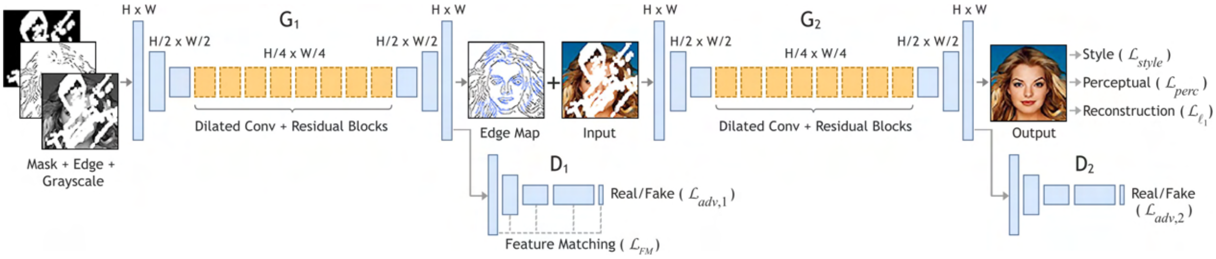

H x W
H/2 x W/2
G_1
H/4 x W/4
H/2 x W/2
H x W
Dilated Conv + Residual Blocks
Mask + Edge + Grayscale
Edge Map
+
Input
D_1
Real/Fake ($\mathcal{L}_{adv,1}$)
Feature Matching ($\mathcal{L}_{FM}$)
H x W
H/2 x W/2
G_2
H/4 x W/4
H/2 x W/2
H x W
Dilated Conv + Residual Blocks
Output
Style ($\mathcal{L}_{style}$)
Perceptual ($\mathcal{L}_{perc}$)
Reconstruction ($\mathcal{L}_{\ell_1}$)
D_2
Real/Fake ($\mathcal{L}_{adv,2}$)

Figure 2.24: Proposed method, in which the masked grayscale image, the mask and a prior edge map are the inputs of G_1 to predict the full edge map. Predicted edge map and incomplete color image are passed to G_2 to perform the inpainting task. Source: Nazeri *et al.* [26]

The proposed EdgeConnect method was able to produce sharp texture fillings and consistent structures on the inpainted images, when compared to other state-of-the-art techniques. Figure 2.25 shows some of the obtained inpainting results. Additionally, this method also proposes a way of enabling user interaction on the inpainting task, by using drawn sketches over the masked region as edge maps to guide the subsequent texture and color filling processes. However, like previous methods, EdgeConnect struggles to inpaint complex missing structures and large masks. Also, the edge generator network is sensible to hyperparameter tuning and relies on multiple inputs—the masked image (grayscale), the mask binary map and an edge map of the masked image previously calculated through the Canny edge detector [77] algorithm—, which tends to be unpractical in most applications.

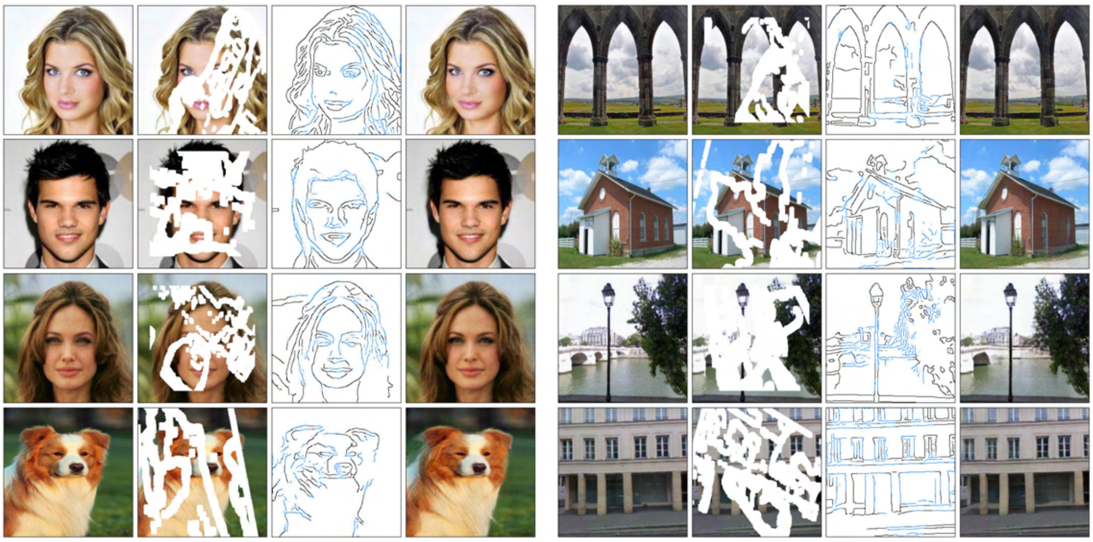


Figure 2.25: Ground Truth (left). Masked input images (center left). Edge map: black edges are previously computed using Canny edge detector and blue edges are predicted in the generator first stage for the missing regions (center right). Inpainting results of the proposed method (right). Source: Nazeri *et al.* [26]

Still in 2019, Yu *et al.* introduced DeepFillv2 [27], which incorporated key concepts from Partial Convolutions and EdgeConnect to the Contextual Attention mechanism from DeepFillv1, resulting in a more complete and robust model. This method uses Gated Convolutions, which can be regarded as a learnable version of Partial Convolutions. As discussed previously, the Partial Convolutions from Liu *et al.* [25] employ fixed—and thus non-learnable—rules to distinguish valid and invalid (masked) pixels at each convolution layer. Gated Convolutions, on the other hand, 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. In that way, the validness/importance 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 importance of each pixel at each Gated Convolution layer. Figure 2.26 illustrates the main differences between Partial and Gated Convolutions.

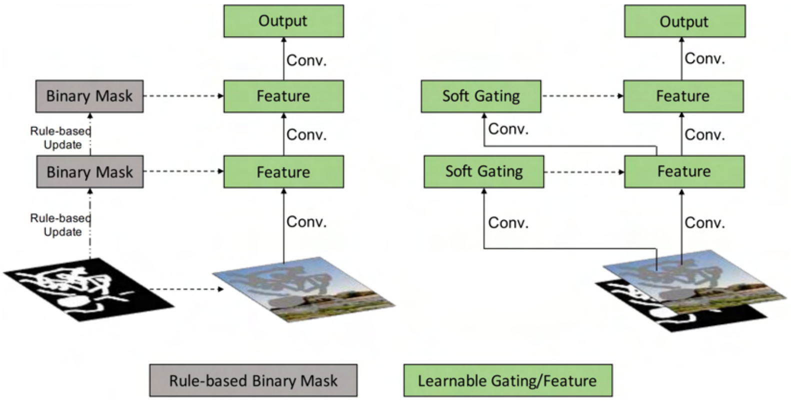


Figure 2.26: Comparison between Partial Convolutions (left) and Gated Convolutions (right). Source: Yu *et al.* [27]

The DeepFillv2 generative model architecture is the same two-stage feed-forward coarse-to-fine model from DeepFillv1, except that all standard convolutions are replaced by Gated Convolutions (the dilated convolution and Contextual Attention layers remain unchanged). For the discriminative model, Yu *et al.* [27] used PatchGAN, a patch-based GAN discriminator architecture proposed by Isola *et al.* [78]. Also, as in Edgeconnect, Spectral Normalization [79] is applied to each standard convolutional layer of the PatchGAN discriminator to stabilize the training process—which is carried with SN-PatchGAN loss and pixel-wise *L1* reconstruction loss. Lastly, DeepFillv2 borrows the concept of user interactivity from EdgeConnect by adding an optional input channel for user sketches to guide the inpainting task towards the user-desired features. Figure 2.27 shows the proposed model architecture.

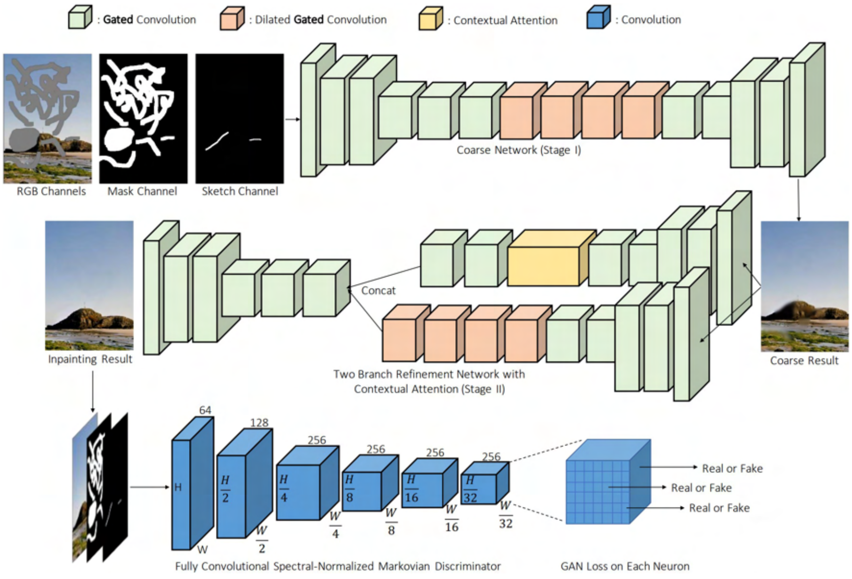


Figure 2.27: Overview of DeepFillv2 framework with Gated Convolutions and SN-PatchGAN discriminator for free-form image inpainting. Source: Yu *et al.* [27]

When compared to previous state-of-the-art methods, the proposed DeepFillv2 was able to significantly improve visual quality and color consistency of the inpainted results, specially with free-form masks. Additionally, by also supporting user inputs, this method allows for more meaningful and satisfactory results, which can be useful in practical applications. For those reasons, DeepFillv2 is regarded as one of the most practical and well-established image inpainting methods to this day. However, this method is computationally expensive due to the soft gating within convolutions and the three-stage adversarial network which has to be trained end-to-end. Figures 2.28 and 2.29 show some of DeepFillv2 inpainting results and figure 2.30 presents a comparison with other methods.

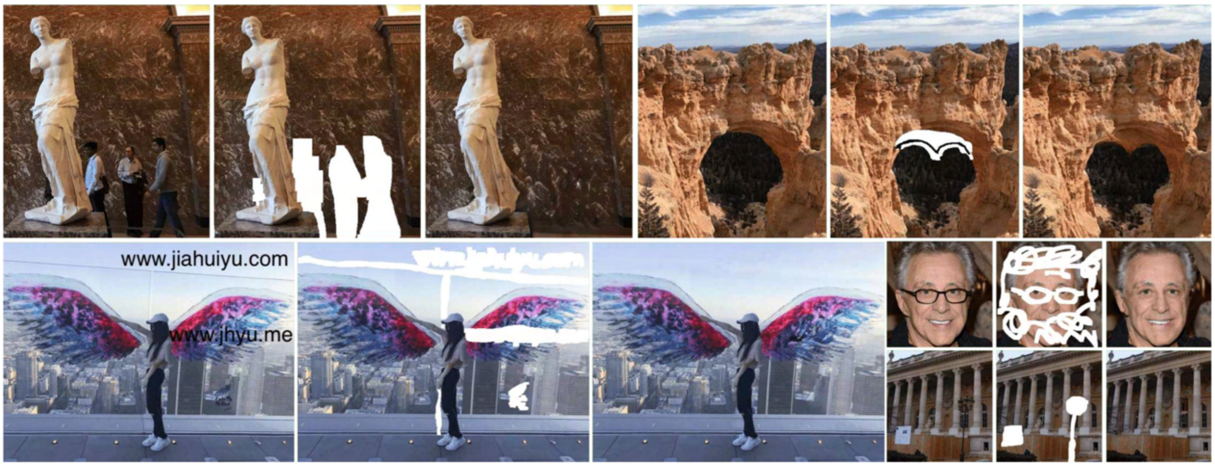


Figure 2.28: Examples of inpainting results by DeepFillv2. Source: Yu *et al.* [27]

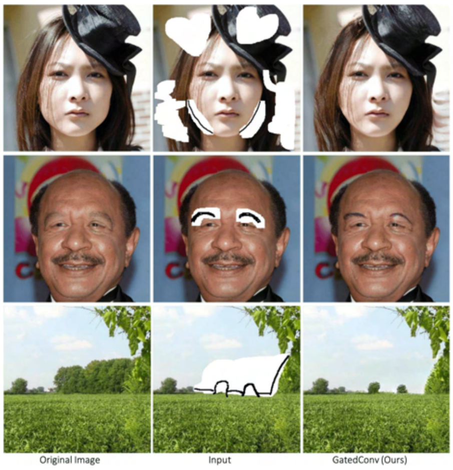


Figure 2.29: User-guided inpainting of faces and natural scenes. Source: Yu *et al.* [27]

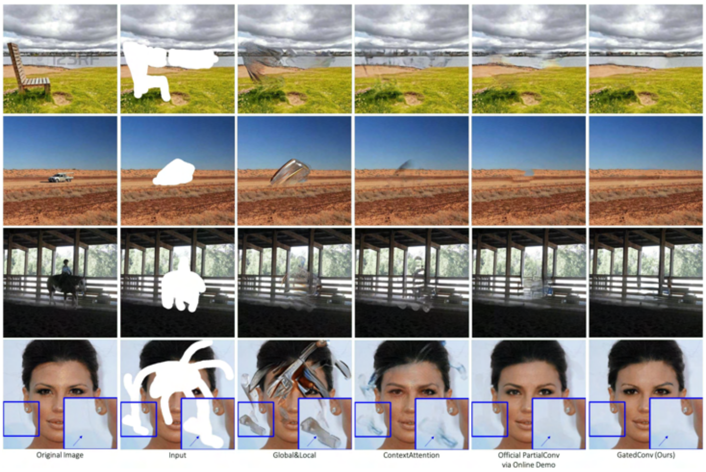


Figure 2.30: Qualitative comparison between results from DeepFillv2 and other state-of-the-art inpainting methods. Source: Yu *et al.* [27]

Although not covered in details in this section, many other interesting GAN-based approaches were proposed in recent years with promising results. Some examples are: Patch-based image inpainting with GANs [80], Shift-net for deep feature rearrangement [81], generative multi-column CNNs [82], GAN-based image completion with new loss function [83], pyramid-context encoder network for high-quality image inpainting [84], foreground-aware image inpainting [85], progressive reconstruction of visual structure for image inpainting [86], coherent semantic attention for image inpainting [87], recurrent feature reasoning [88] and ultra high-resolution image inpainting [89].

Also, since 2017 when Li *et al.* [90] introduced the first GAN-based inpainting method specifically geared towards face images, many other GAN-based facial inpainting methods were proposed, taking advantage of the unique and defining features and components in human faces to improve upon existing methods. For example, Li *et al.* [91] explore the symmetry in human faces to produce more accurate results; Zhou *et al.* [92] and Wang *et al.* [93] introduce new models to enhance texture consistency and high-frequency details among different generated facial components; Portenier *et al.* [94] and Jo *et al.* [95] use user-drawn free-form sketches with colors to guide the inpainting task and produce high-quality face images; Chen *et al.* [96] introduce controllable parameters for the inpainted face such as ”male” or ”female” and ”smiling” or ”not smiling”; and Hui *et al.* [97] achieve sharp fine-grained texture and consistent structures in the facial inpainting results, even on high-resolution images.

In conclusion, despite being around for no more than five 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. Nonetheless, these methods still face several challenges in applications such as Video Inpainting [98, 99], Extreme Image Inpainting [100]—that is, completing complex scene structures and large missing areas in high-quality images—or Pluralistic Image Completion—the task of generating multiple and diverse plausible solutions for a single masked image—, a topic first introduced in 2019 [101] with promising methods like PiiGAN [102] and UCTGAN [103]. Therefore, GAN-based Image Inpainting is currently a very active field of research with new innovative methods and approaches being proposed every year. As an enthusiast myself, I can only be excited for the next advancements and innovations those methods will present in the near future.

Summary: 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 2.2 summarizes the reviewed Deep Learning image inpainting methods.

Method	Year	Model	Description	Loss Function	Dataset
Jain *et al.* [67]	2008	CNN	Autoencoder formulated for image denoising and extended to image inpainting.	Reconstruction loss	In-house 100 grayscale images
Xie *et al.* [48]	2012	CNN	Sparse coding and deep neural networks as a denoising autoencoder. Extended to image inpainting.	Reconstruction loss	In-house images
Pathak *et al.* [20]	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.* [21]	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.* [23]	2017	GAN	Global and local discriminators and encoder-decoder generator with dilated convolutions	Weighted *L2*, adversarial loss	Places2, ImageNet, CMP Facade
Yeh *et al.* [76]	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.* [24]	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.* [25]	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.* [26]	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.* [27]	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

Table 2.2: Summary of the reviewed Deep Learning methods for image inpainting. Source: adapted from Jam *et al.* [36]

2.3 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. In this section, we present and discuss some of the most established and commonly used datasets for Deep Learning image inpainting methods.

PASCAL VOC 2007

PASCAL VOC 2007 is an image dataset built for the PASCAL Visual Object Classes (VOC) Challenge 2007 targeted at applications like object detection, image classification, object segmentation, and person layout. It is comprised of a total of 19,737 images grouped into twenty annotaded object classes in four major categories: person, animals, vehicles, and indoor objects [28]. Figure 2.31 presents images from the PASCAL VOC 2007 dataset.

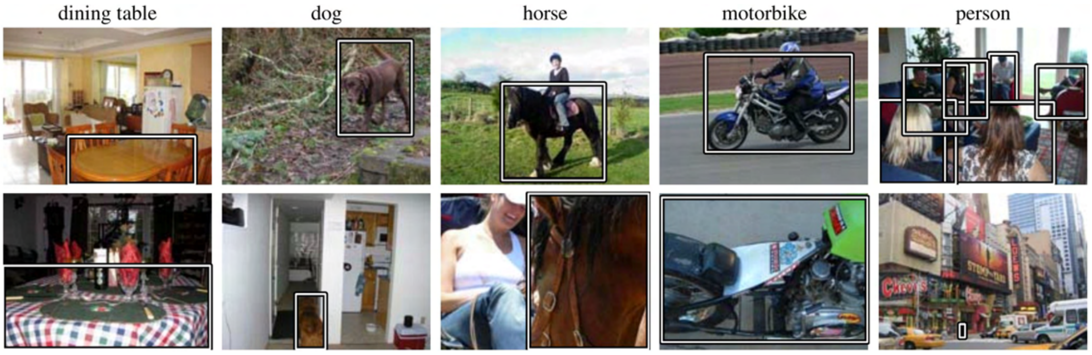


Figure 2.31: Example images from PASCAL VOC 2007. Source: Everingham *et al.* [28]

ImageNet

ImageNet is a large-scale general-purpose image dataset currently with 14,197,122 hand-annotated images classified into more than 20,000 categories. It was introduced in 2009 by Deng *et al.* [29] and, since 2010, the ImageNet project runs the annual ImageNet Large Scale Visual Recognition Challenge (ILSVRC) for objects and scenes detection and classification. Due to its size and content diversity, ImageNet is considered one of the most important datasets for general-purpose learning-based image applications with high generalization requisites, and thus has received many updates and improvements over the past years. Figure 2.32 shows images from the dataset.

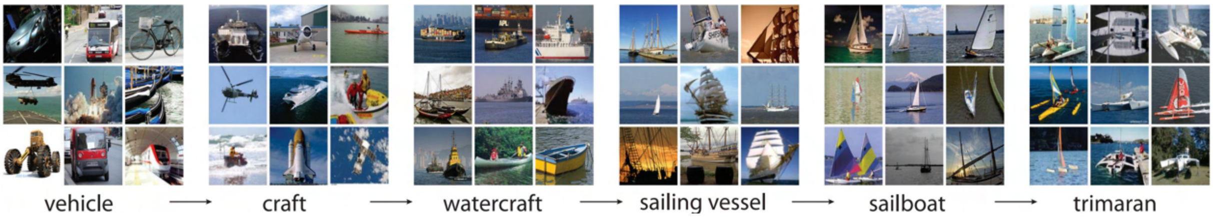


Figure 2.32: A snapshot of the Vehicle Subtree from ImageNet. Source: Deng *et al.* [29]

Paris Street View

Developed in 2015 by Doersch *et al.* [30], Paris Street View is a dataset containing approximately 10,000 geo-tagged images of the streets of Paris and its suburban areas, which were extracted from Google Street View. Figure 2.33 shows images from the dataset.

Figure 2.33: Example images from Paris Street View. Source: Doersch *et al.* [30]

CelebA

CelebFaces Attributes Dataset—or CelebA—, is a large-scale face attributes dataset composed of 202,599 images of more than 10,000 celebrity faces, each with 40 attribute annotations. Introduced in 2015 by Liu *et al.* [31], the CelebA dataset presents considerable pose variations and background clutter, which makes it suitable for facial synthesis and inpainting applications. Figure 2.34 shows some of the images from the CelebA dataset.

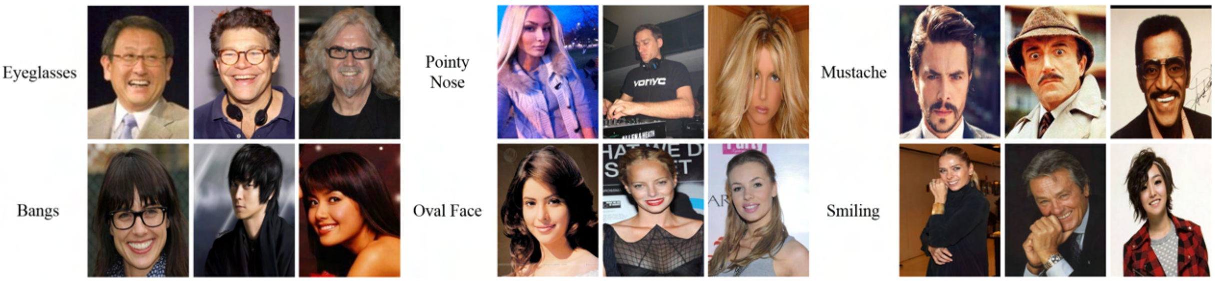


Figure 2.34: Example images from the CelebA dataset. Source: Liu *et al.* [31]

CelebA-HQ

CelebA-HQ is a high-quality version of the CelebA dataset developed in 2017 by Karras *et al.* [32], who analysed and used different image processing techniques to standardize the 202,599 original images from CelebA—whose resolutions vary from 43×55 to

6732×8984—, resulting in the 30,000 aligned and cropped 1024×1024 images of celebrity faces that constitute the CelebA-HQ dataset. Figure 2.35 presents some images from the CelebA-HQ dataset.

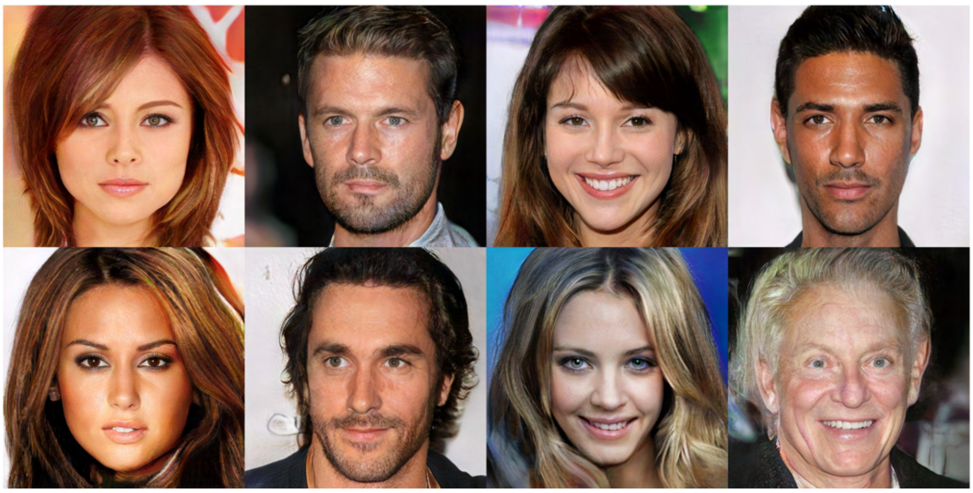


Figure 2.35: Example images from the CelebA-HQ dataset. Source: Karras *et al.* [32]

Places2

Places2 is a large-scale diverse scenery dataset used for scene recognition and high-level visual understanding tasks. Introduced in 2017 by Zhou *et al.* [33], it was designed following principles of human visual cognition and has more than 10 million scene images split into more than 400 unique categories. Figure 2.36 presents images from the Places2 dataset.

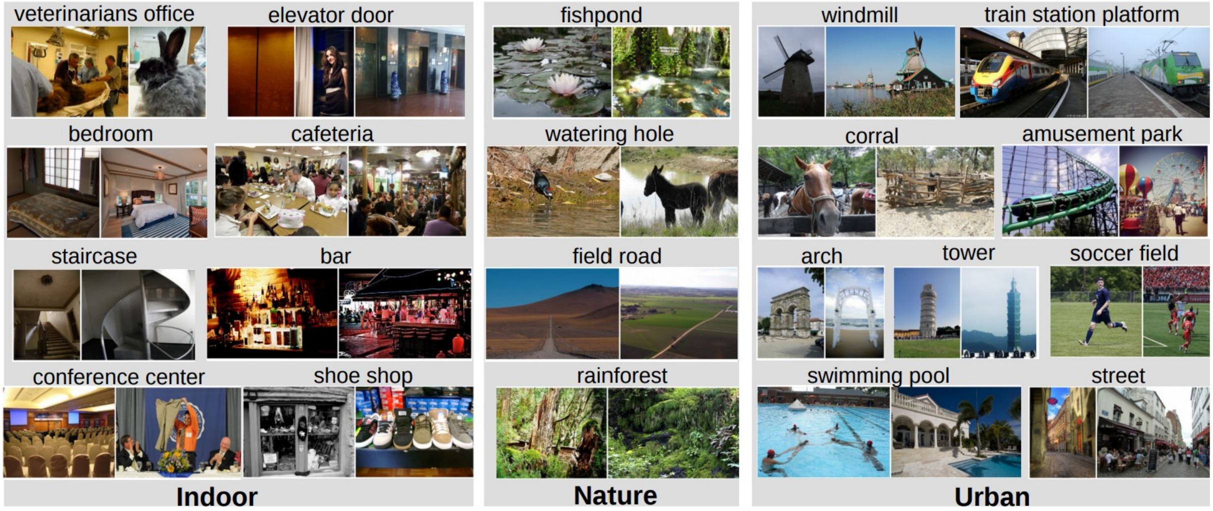


Figure 2.36: Example images from the Places2 dataset. Source: Zhou *et al.* [33]

NVidia Irregular Mask Dataset

NVidia Irregular Mask Dataset is a dataset composed of binary masks of random streaks and holes of arbitrary shapes, with and without border constraints. This dataset was developed in 2018 by Liu *et al.* [25] to train and evaluate their proposed partial convolution-based inpainting method, and contains 55,116 masks for the training set and additional 24,866 masks for the testing set, all at 512×512 resolution. Figure 2.37 shows some of the masks from NVidia Irregular Mask Dataset.

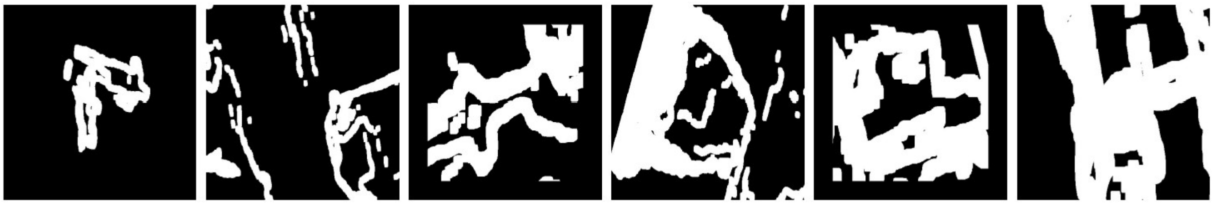


Figure 2.37: Example masks from the testing set of NVidia Irregular Mask Dataset, with different hole-to-image area ratios. Source: Liu *et al.* [25]

Quick Draw Irregular Mask Dataset

Quick Draw Irregular Mask Dataset (QD-IMD) is also a mask dataset and was introduced by Iskakov *et al.* [34] in 2018 in order to solve two limitations from NVidia Irregular Mask Dataset: the lack of human features in the masks due to the randomness in their generation process and the excessive occurrences of sharp edges in the masks due to the rough crops close to borders. The QD-IMD dataset is built with human-drawn strokes taken from the Quick Draw dataset (a collection of 50 million human drawings) and contains 60,000 irregular masks. Figure 2.38 shows some of the masks from Quick Draw Irregular Mask Dataset.

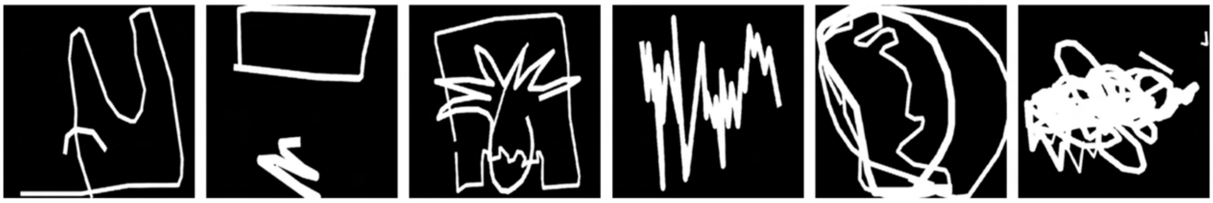


Figure 2.38: Example masks from Quick Draw Irregular Mask Dataset. Source: Iskakov *et al.* [34]

Flickr-Faces-HQ

Flickr-Faces-HQ (FFHQ) is a high-quality dataset of human faces originally created by Karras *et al.*in 2019 as a benchmark for the famous StyleGAN model [35]. The images from the FFHQ dataset were crawled from Flickr—an image and video hosting website—and automatically aligned and cropped afterwards. The dataset is composed of 70,000 images of human faces at 1024×1024 resolution, contains considerable variation in terms of age, ethnicity and image background and offers substantial coverage of accessories such as eyeglasses and hats.

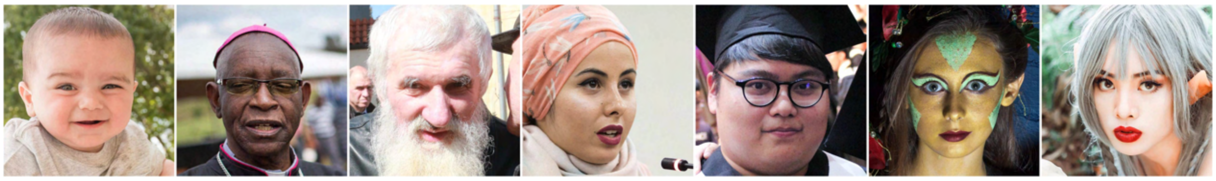


Figure 2.39: Example images from the Flickr-Faces-HQ (FFHQ) dataset. Source: Karras *et al.* [35]

Summary: Datasets

Table 2.3 summarizes the main characteristics of the reviewed datasets, used to develop, train and evaluate 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

Table 2.3: Summary of the discussed datasets. Source: adapted from Jam *et al.* [36] and Elharrouss *et al.* [8]

3 MATERIALS AND METHODS

3.1 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.* [27] 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. As previously discussed and illustrated in figure 2.18, 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.* [27].
- Finally, discuss the obtained results.

3.2 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.

3.2.1 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 [36]—such as universal quality index [104], multiscale SSIM [105], visual information fidelity [106], inception score [107], Fréchet inception distance [108] and LPIPS [109]—, 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).

Peak Signal to Noise Ratio (PSNR)

To calculate the PSRN, it is necessary to first compute the maximum possible pixel value ($\mathrm{MAX_T}$) of the ground-truth reference image T and the Mean Squared Error (MSE) between the ground truth and the inpainted image.

The maximum possible pixel value $\mathrm{MAX_T}$ can be determined as follows:

$$\mathrm{MAX_T} = 2^B - 1$$

where B is the number of bits with which the image pixels are represented. For example, if the image pixel values are represented using 8 bits per sample, $\mathrm{MAX_T}$—the maximum possible value any pixel in the image can have—would be equal to 255.

Then, given the ground-truth $m \times n$ image T and the respective same-sized inpainted image I, the Mean Squared Error (MSE) can be computed using the following equation:

$$\mathrm{MSE} = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} [T(i,j) - I(i,j)]^2$$

where $T(i,j)$ and $I(i,j)$ represent the images' corresponding pixel values in grayscale or in each RGB channel.

Finally, the PSNR (in dB) is defined as:

$$\mathrm{PSNR} = 10 \cdot \log_{10}\left(\frac{\mathrm{MAX_T}^2}{\mathrm{MSE}}\right) = 20 \cdot \log_{10}\left(\mathrm{MAX_T}\right) - 10 \cdot \log_{10}(\mathrm{MSE})$$

The PSNR measures how close the inpainted image pixels are to the corresponding pixels in the ground-truth reference image. Therefore, the higher the PSNR value, the better the quality of the reconstructed image.

Structure Similarity Index Measure (SSIM)

SSIM is a perceptual metric for image quality degradation focused on image aspects that are relevant to the Human Visual System (HVS) model, that is, it is focused on image attributes that more heavily impact the human visual perception. The SSIM equation is based on the comparison measurements of three attributes of the two images: luminance $l(T,I)$, contrast $c(T,I)$, and structure $s(T,I)$.

Given the ground-truth reference image T and the respective inpainted image I, the SSIM can be computed with the following equation:

$$\mathrm{SSIM}(T,I) = \left[l(T,I)^{\alpha} \cdot c(T,I)^{\beta} \cdot s(T,I)^{\gamma}\right], \text{ with:}$$

$$
\begin{aligned}
l(T, I) &= \frac{2\mu_T\mu_I + c_1}{\mu_T^2 + \mu_I^2 + c_1} \\
c(T, I) &= \frac{2\sigma_T\sigma_I + c_2}{\sigma_T^2 + \sigma_I^2 + c_2} \\
s(T, I) &= \frac{\sigma_{TI} + c_3}{\sigma_T\sigma_I + c_3}
\end{aligned}
$$

where:

- μ_T, μ_I are the average pixel values of T and I;
- σ_T^2, σ_I^2 are the variance of pixel values of T and I;
- σ_{TI} is the covariance of T and I; and
- c_1, c_2 and c_3 are constants used to stabilize the division with weak denominator, defined as:
  - ⋆ $c_1 = (k_1 \cdot \mathrm{MAX_T})^2$, with $k_1 = 0.01$ by default;
  - ⋆ $c_2 = (k_2 \cdot \mathrm{MAX_T})^2$, with $k_2 = 0.03$ by default; and
  - ⋆ $c_3 = c_2/2$.

Generally, the weights α, β and γ are set to 1, and the SSIM formula can be reduced to the following form:

$$
\mathrm{SSIM}(T, I) = \frac{(2\mu_T\mu_I + c_1)(2\sigma_{TI} + c_2)}{(\mu_T^2 + \mu_I^2 + c_1)(\sigma_T^2 + \sigma_I^2 + c_2)}
$$

The value of SSIM extends between -1 and $+1$. The closer it is to $+1$, the more similar the two images are in terms of luminance, contrast and structure.

3.2.2 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 [24, 27].

3.3 The DeepFillv2 Model

As discussed in the previous chapter, DeepFillv2 [27] incorporated key concepts from previous GAN-based methods—such as Partial Convolutions [25], DeepFillv1's Contextual Attention mechanism [24] and EdgeConnect's user-guided inpainting [26]—to produce a complete, robust and well-performing model. This section describes the main features and characteristics of the DeepFillv2 model.

3.3.1 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:

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

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.* [25] 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:

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

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. As described by Yu *et al.* [27], ”Partial Convolutions can be viewed as non-learnable single-channel feature hard-gating”.

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:

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

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 2.26 illustrates the main differences between Partial and Gated Convolutions.

3.3.2 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 [23], MarkovianGANs [110], perceptual loss [73] and spectral-normalized GANs [79], 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.* [27] describe, "SN-PatchGAN is simple in formulation, fast and stable in training and produces high-quality inpainting results". Figure 2.27 presents the complete DeepFillv2 model architecture.

3.3.3 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.

3.3.4 Free-Form Mask Generation Algorithm

Yu *et al.* [27] 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 3.1 presents three examples of masks that were automatically generated during the DeepFillv2 model training process on the the FFHQ dataset.

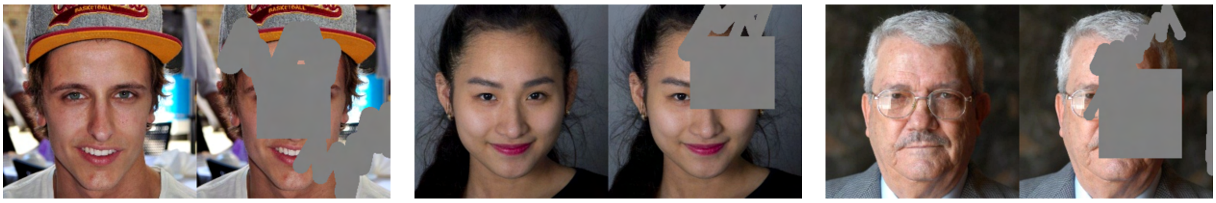


Figure 3.1: Examples of masks automatically generated during the DeepFillv2 model training process on the FFHQ dataset. Source: created by the author

3.3.5 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 [111] 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). Figure 2.29 shows examples of inpaintings guided by user-provided sketches.

3.4 Implementation

3.4.1 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 [112], 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 pre-trained weights were made available by Yu *et al.* [27] 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 also allows for the comparison of those three models' outputs by using the same input image, which standardizes the evaluation process.

3.4.2 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 3.2 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.

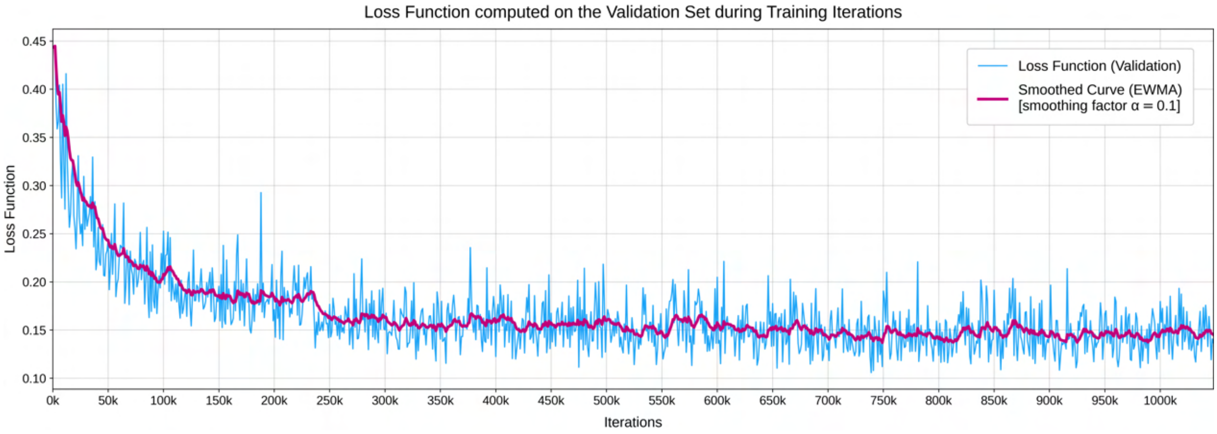


Figure 3.2: DeepFillv2 loss function computed on the validation set during training iterations and the corresponding smoothed curved. Source: created by the author

This project's implementation details are publicly available in a GitHub repository at: ⟨https://github.com/PedroAntonacio/facial-inpainting⟩ [113]. The next chapter presents this results of the DeepFillv2 model trained on the FFHQ dataset.

4 RESULTS

This chapter 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.* [27]. This chapter is divided into 5 sections:

- Section 4.1 "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.
- Section 4.2 "Models Comparison" compares the results from the DeepFillv2 models trained on CelebA-HQ and Places2 by Yu *et al.* [27] with the results from our DeepFillv2 model trained on the FFHQ dataset, for the case of regular facial inpainting tasks.
- Section 4.3 "Edge Cases" compares the results from the DeepFillv2 model trained on CelebA-HQ by Yu *et al.* [27] with the results from our DeepFillv2 model trained on FFHQ, for the case of more challenging and difficult facial inpainting tasks.
- Section 4.4 "Personal Pictures" shows the same model comparison from section 4.3 "Edge Cases", but the inpainting tasks use face pictures from the author's friends and family, obtained and utilized with their consent.
- Finally, section 4.5 "Quantitative Metrics" presents tables with the Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index Measure (SSIM) average values calculated for the resulting images from each of the test cases discussed above.

With the exception of Section 4.4 "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 sections were never seen by any of the models during their respective training processes.

4.1 Model Progress During Training

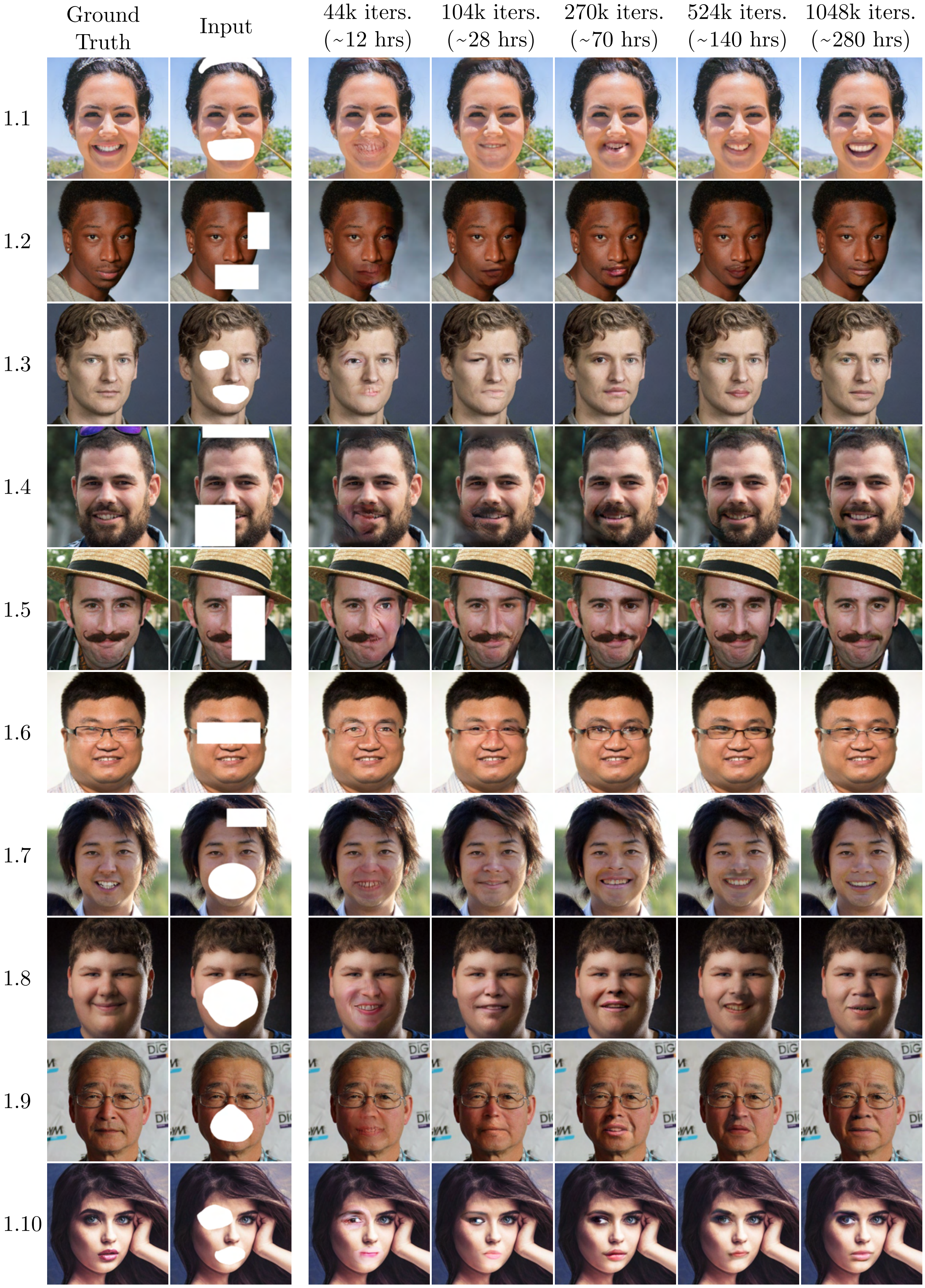


Figure 4.1: Model progress during the training process. Source: created by the author

4.2 Models Comparison

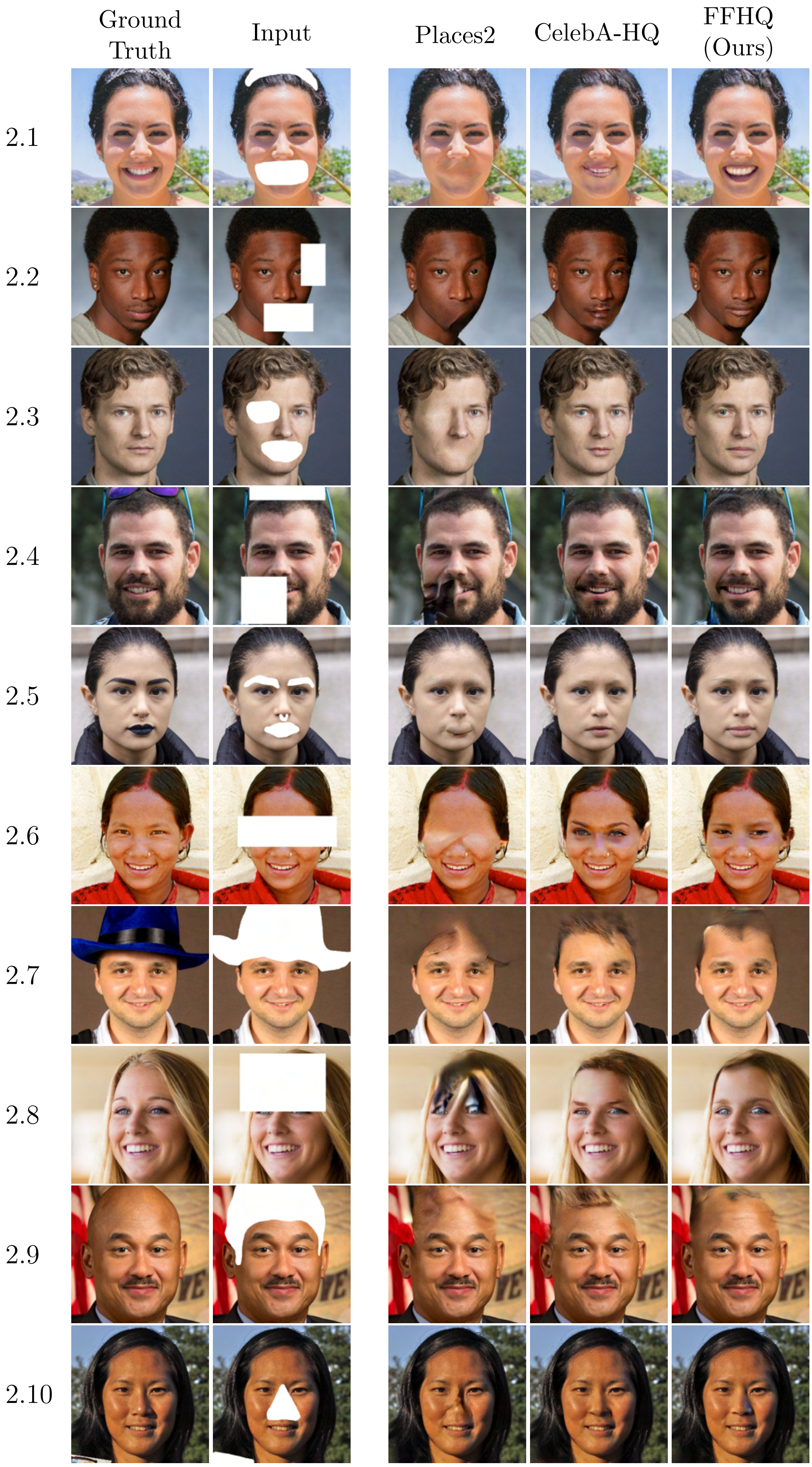


Figure 4.2: Models comparison: images 2.1 to 2.10. Source: created by the author

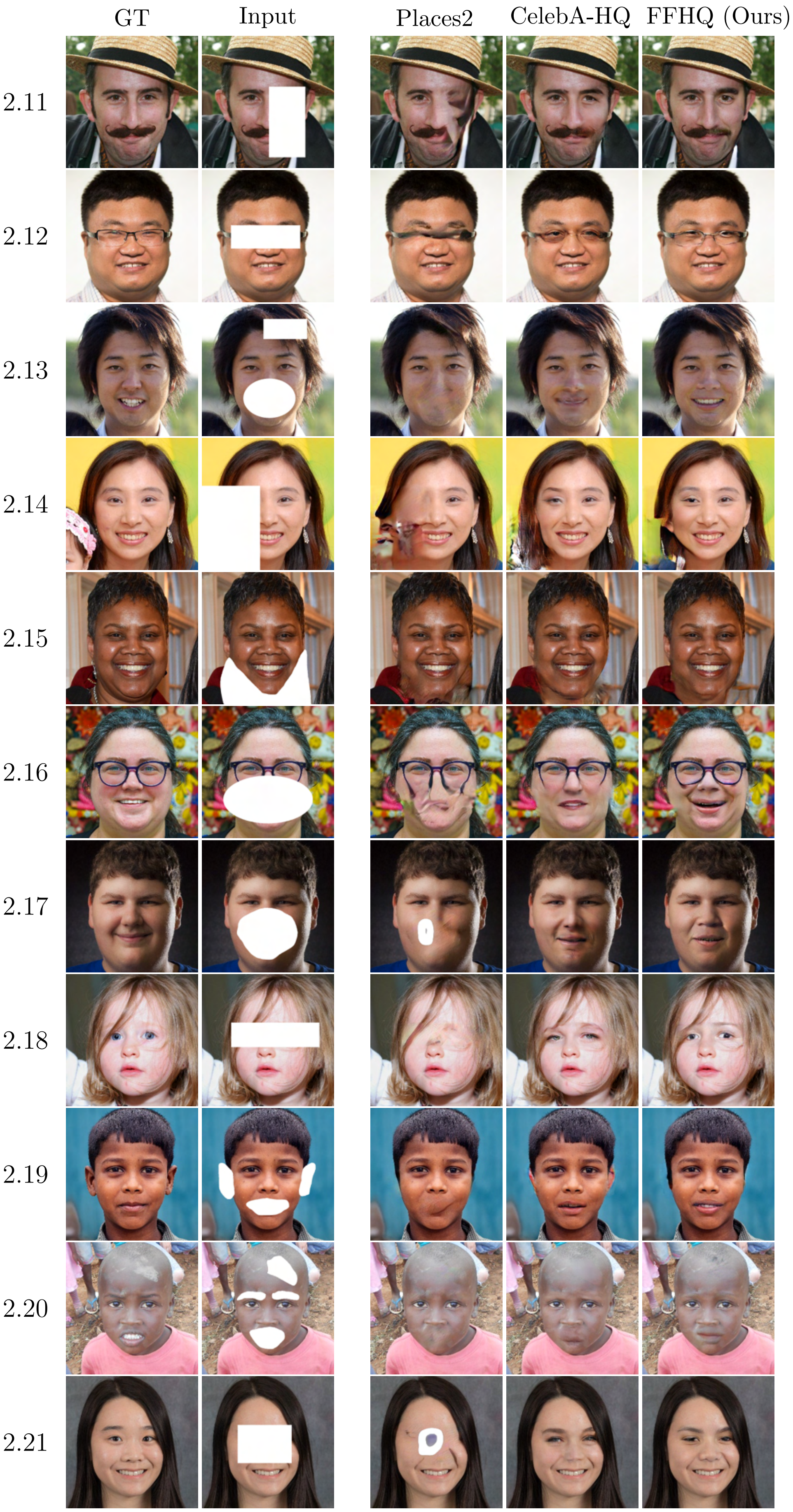


Figure 4.3: Models comparison: images 2.11 to 2.21. Source: created by the author

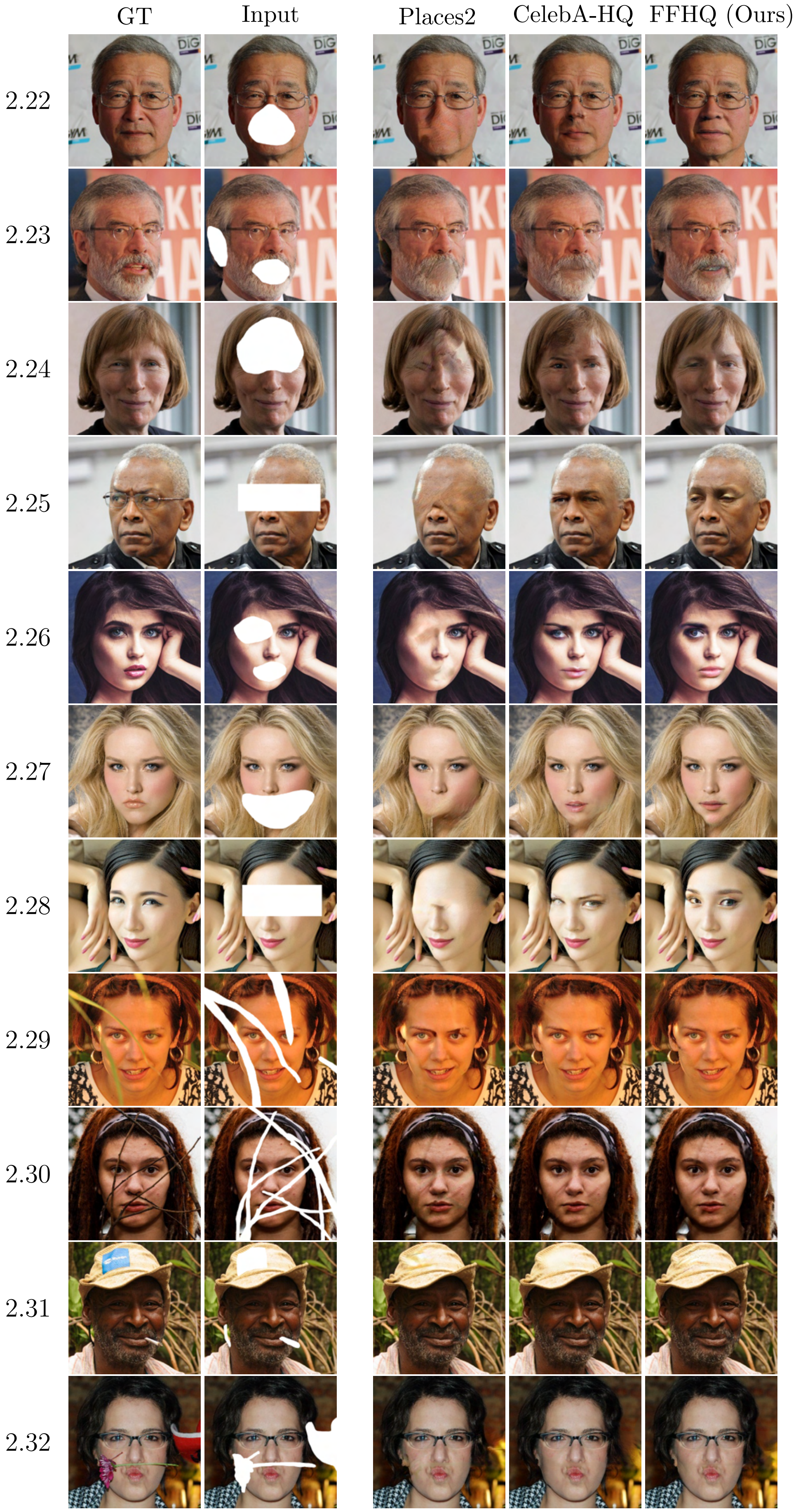


Figure 4.4: Models comparison: images 2.22 to 2.32. Source: created by the author

4.3 Edge Cases

4.3.1 Strong Facial Expressions

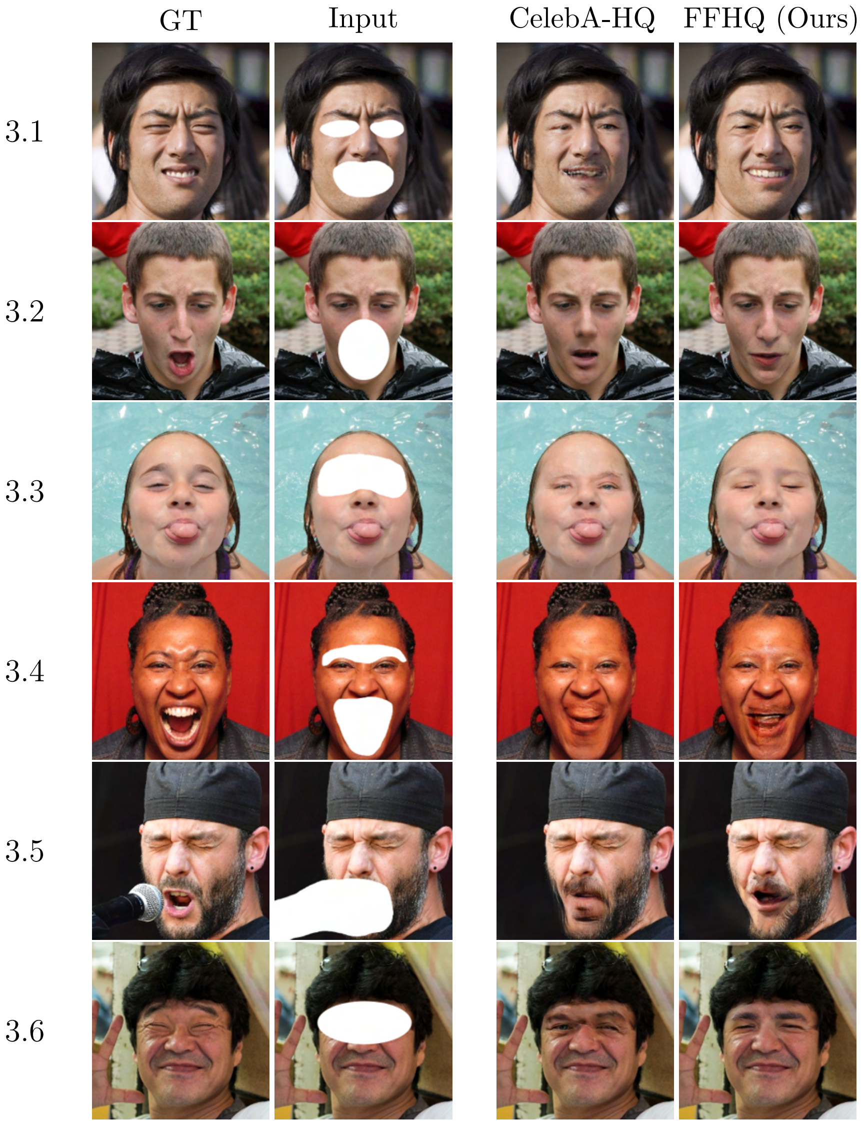


Figure 4.5: Edge case: strong facial expressions. Source: created by the author

4.3.2 Accessories

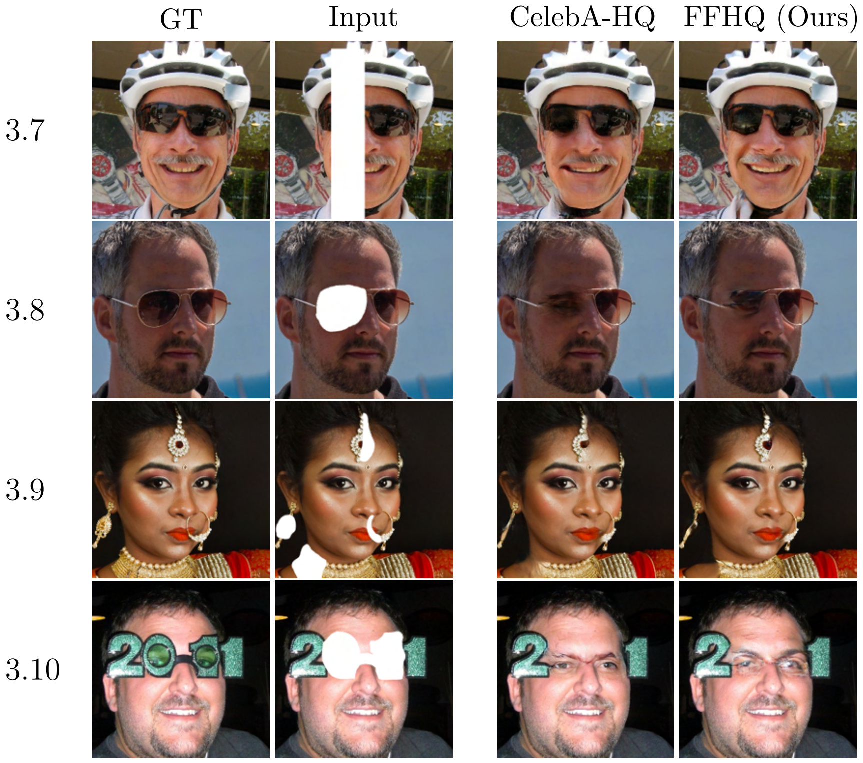


Figure 4.6: Edge case: accessories. Source: created by the author

4.3.3 Painted Faces

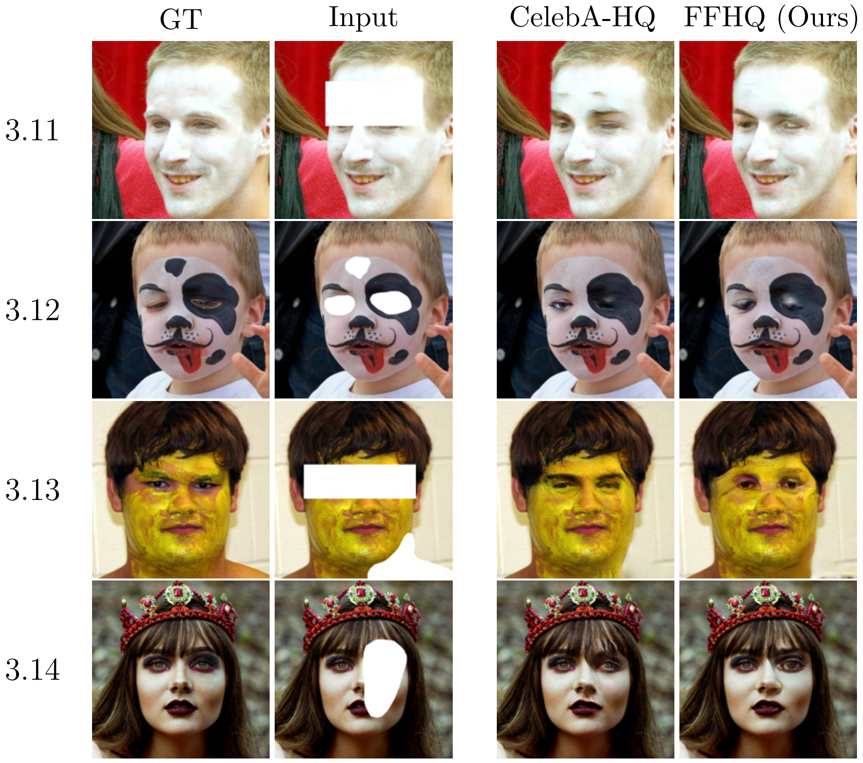


Figure 4.7: Edge case: painted faces. Source: created by the author

4.3.4 Unnatural Hair Colors

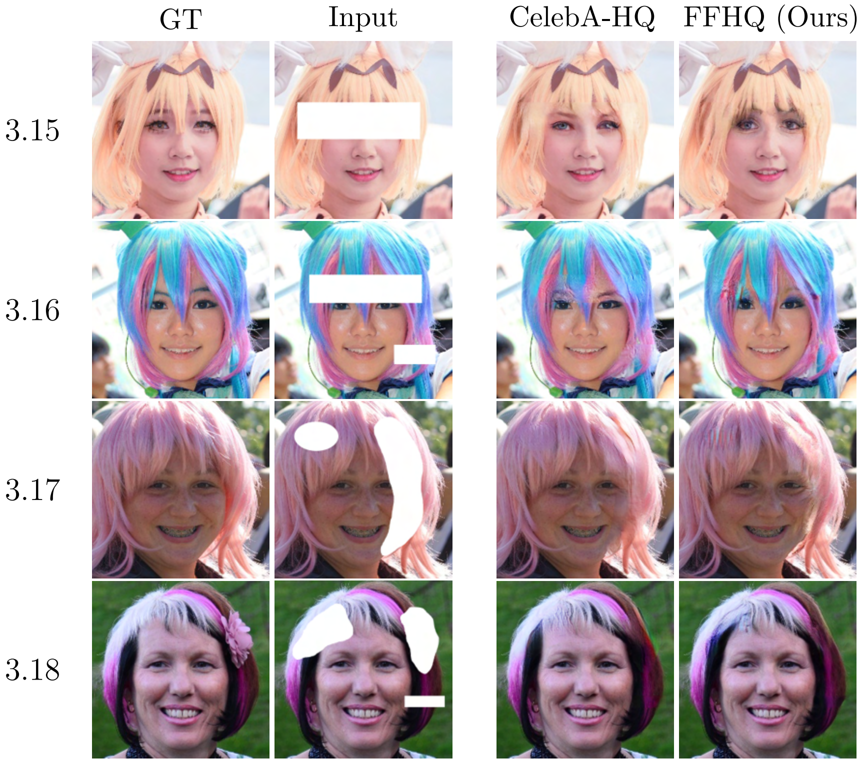


Figure 4.8: Edge case: unnatural hair colors. Source: created by the author

4.3.5 Babies

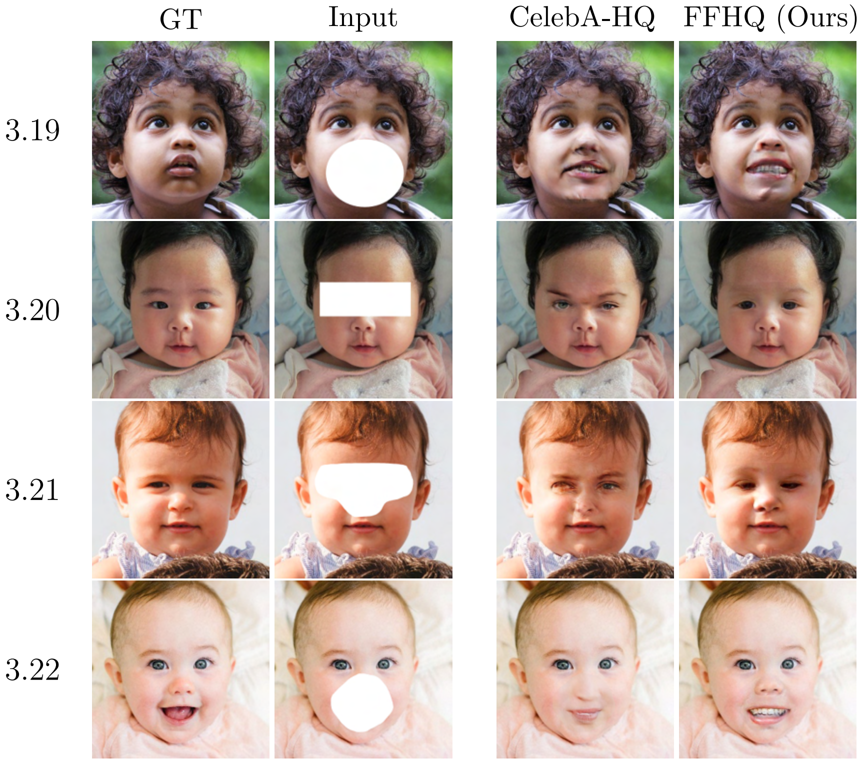


Figure 4.9: Edge case: babies. Source: created by the author

4.3.6 Challenging Backgrounds

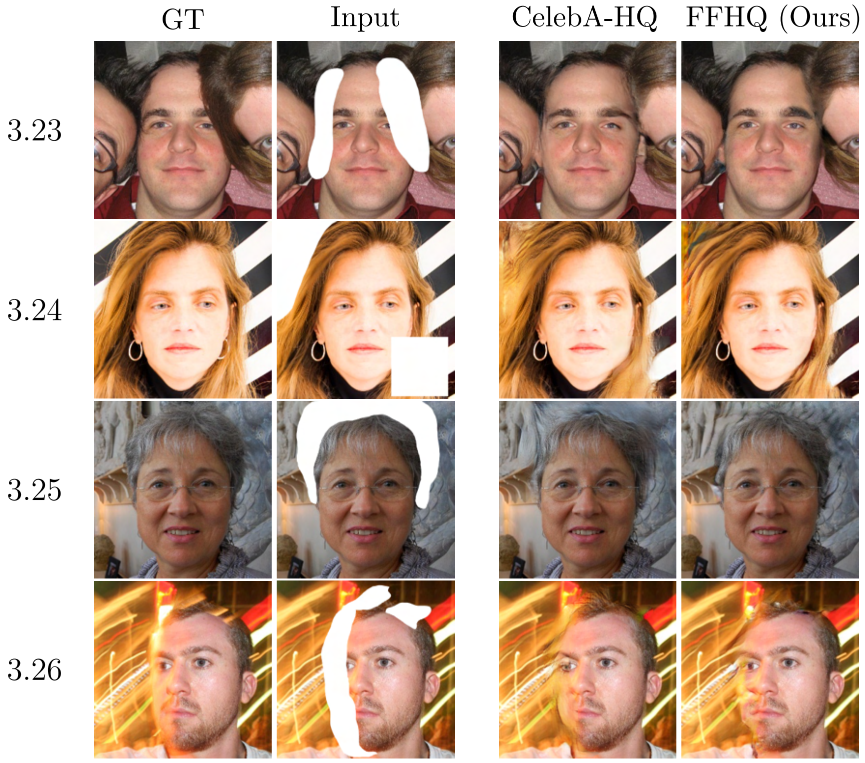


Figure 4.10: Edge case: challenging backgrounds. Source: created by the author

4.3.7 Side View of Faces

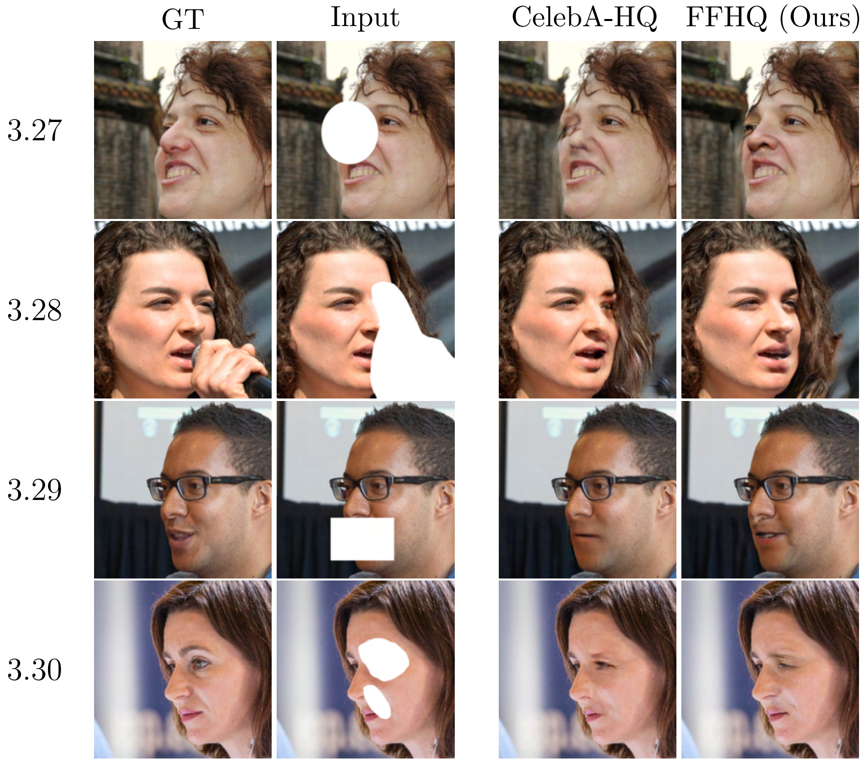


Figure 4.11: Edge case: side view of faces. Source: created by the author

4.3.8 Large Masks

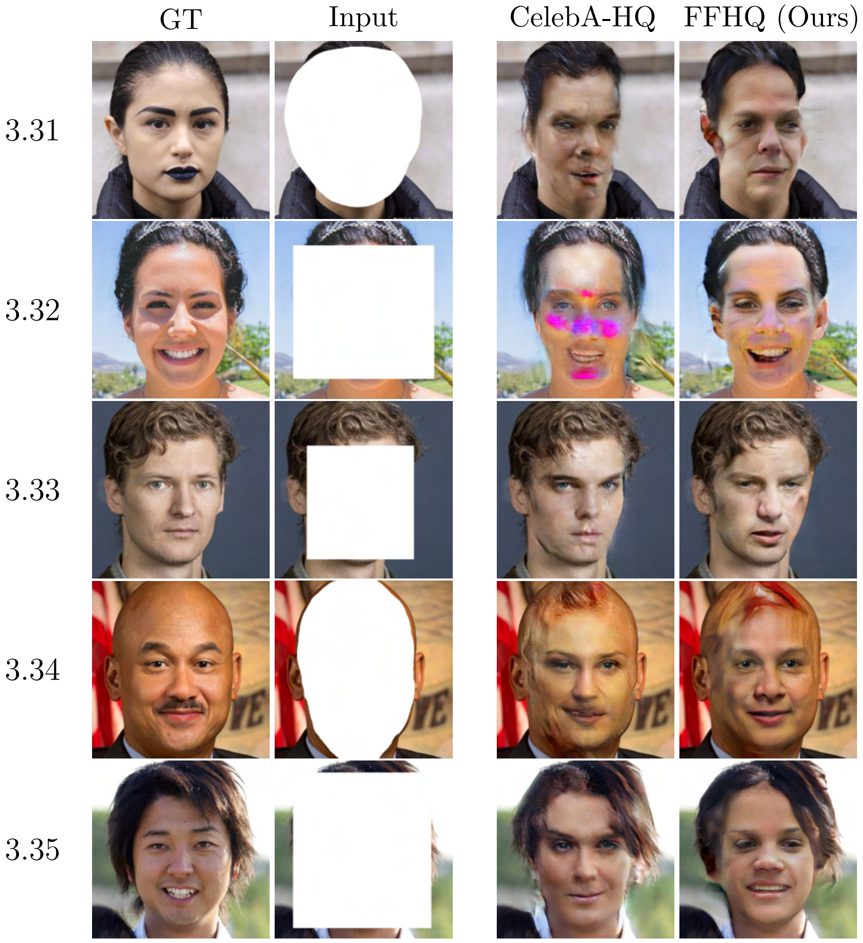


Figure 4.12: Edge case: large masks. Source: created by the author

4.3.9 Human Statues

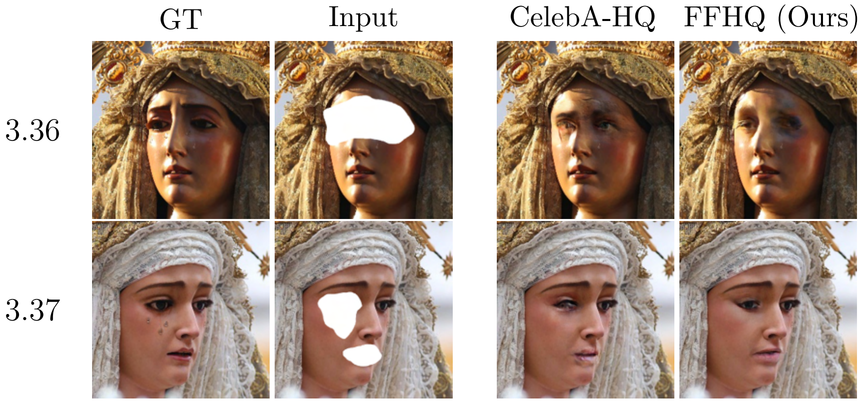


Figure 4.13: Edge case: human statues. Source: created by the author

4.4 Personal Pictures

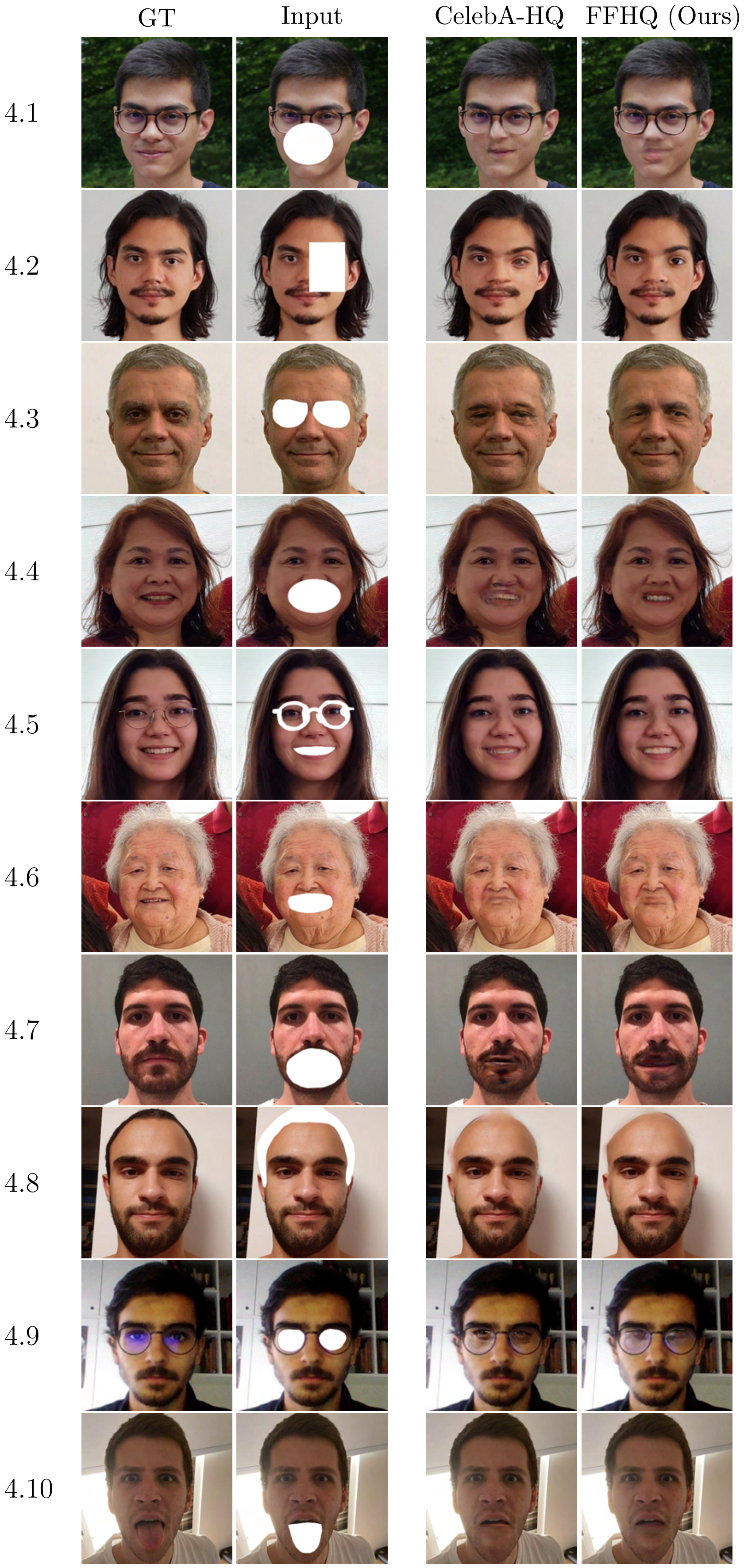


Figure 4.14: Personal pictures: images 4.1 to 4.10. Source: created by the author

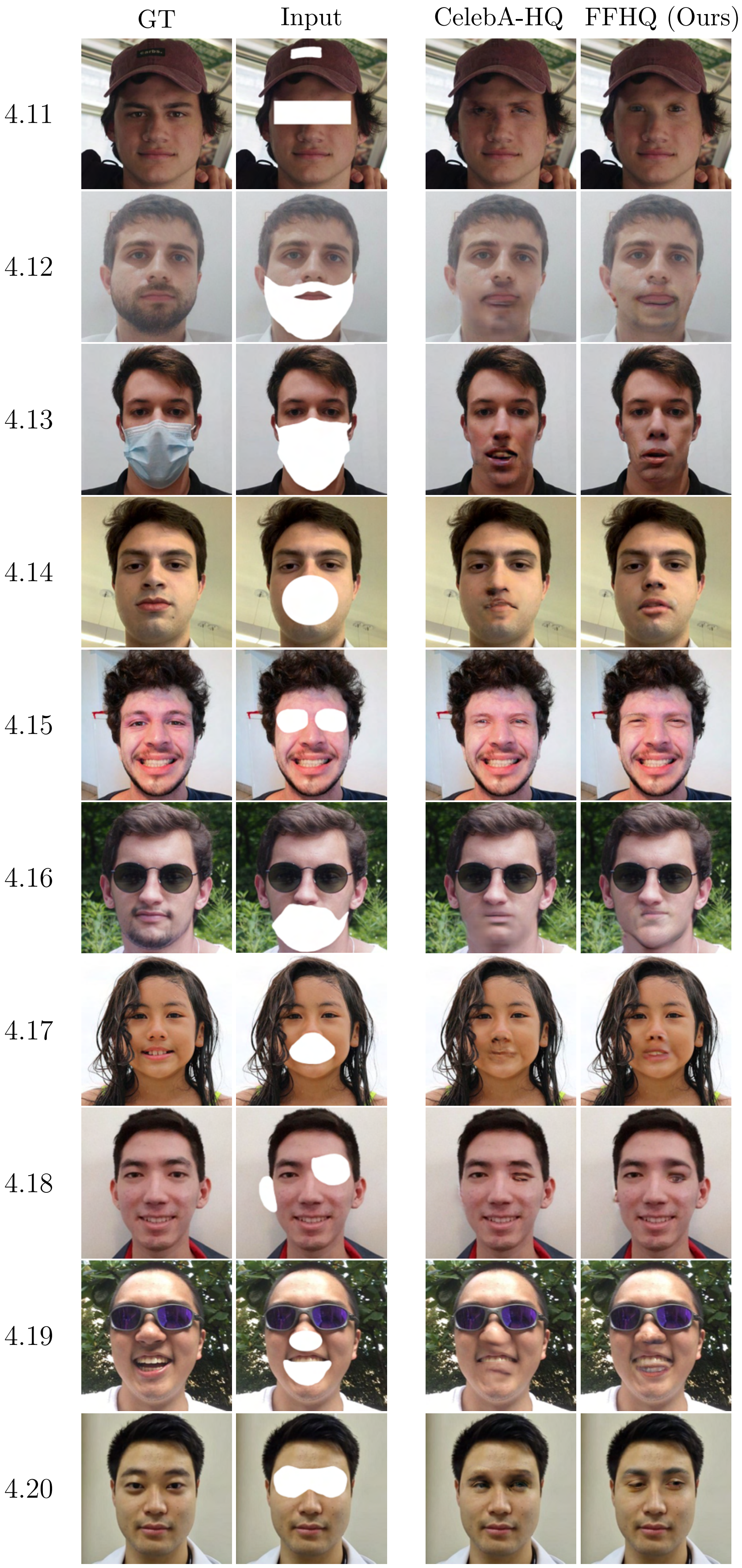


Figure 4.15: Personal pictures: images 4.11 to 4.20. Source: created by the author

4.5 Quantitative Metrics

Case	**Number of Images**	**SSIM (Mean ± SD)**		
		Places2	**CelebA-HQ**	**FFHQ (Ours)**
Regular Cases	32	0.898 ± 0.051	0.909 ± 0.050	$\mathbf{0.911 \pm 0.048}$
Edge Cases	37	—	0.878 ± 0.107	$\mathbf{0.882 \pm 0.103}$
Personal Pictures	20	—	0.909 ± 0.052	$\mathbf{0.910 \pm 0.053}$

Table 4.1: Average Structural Similarity Index Measure (SSIM) values calculated for the resulting images from each test case. Source: created by the author

Case	**Number of Images**	**PSNR (Mean ± SD)**		
		Places2	**CelebA-HQ**	**FFHQ (Ours)**
Regular Cases	32	23.52 ± 3.31	$\mathbf{25.45 \pm 3.67}$	25.44 ± 3.65
Edge Cases	37	—	24.62 ± 4.79	$\mathbf{24.90 \pm 5.01}$
Personal Pictures	20	—	26.36 ± 5.22	$\mathbf{26.62 \pm 5.32}$

Table 4.2: Average Peak Signal-to-Noise Ratio (PSNR) values calculated for the resulting images from each test case. Source: created by the author

5 DISCUSSION

FFHQ Model Progress During Training

Even though the average validation loss function value remains relatively constant and does not improve significantly from training iteration 250,000 onwards as shown in figure 3.2, the snapshots of the model training process in figure 4.1 from section 4.1 ”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 facial structures that are more realistic and coherent with the rest of the original image.

FFHQ, CelebA-HQ and Places2 Models Evaluation

Section 4.2 ”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 section 4.2 ”Models Comparison”, which can be clearly observed in images 2.1 and 2.2 from figure 4.2, images 2.11 to 2.13 and 2.20 from figure 4.3, and images 2.22, 2.26 and 2.28 from figure 4.4. 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, for example, in the generated eyes in images 2.6 and 2.14 from figures 4.2 and 4.3, respectively. For the use case of object removal, where masks tend to be narrower and smaller, as seen in images 2.29 to 2.32 from figure 4.4, both CelebA-HQ and FFHQ models produced high-quality results, without major differences between their results.

In section 4.3 ”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 masks and side views from figures 4.12 and 4.11—, the models displayed a few promising capabilities in the resulting images. For example, both models were able to recognize and partially reconstruct the sunglasses on images 3.7 and 3.8 from figure 4.6, adequately propagate texture and color for the cases of painted faces and unnatural hair colors in figures 4.7 and 4.8, and also deal with challenging backgrounds in a satisfactory way in figure 4.10. In general, both models performed similarly for the images from section 4.3 ”Edge Cases”, with no model clearly and consistently outperforming the other throughout the test cases.

Real-World Facial Inpainting Applications

For section 4.4 ”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 sections, 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 section 4.4 ”Personal Pictures” were the worst among all sections, for both CelebA-HQ and FFHQ models. The models often produced unrealistic facial structures—such as in images 4.2, 4.11 and 4.18 from figures 4.14 and 4.15—, or excessively blurred results—as seen in images 4.1, 4.6, 4.7 and 4.10 from figure 4.14. 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 4.5, 4.13, 4.14 and 4.15 from figures 4.14 and 4.15. 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.

PSNR and SSIM Metrics

Finally, section 4.5 ”Quantitative Metrics” presents the average calculated Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index Measure (SSIM) values for the resulting images from each of the test cases discussed above. As shown in tables 4.1 and 4.2, the FFHQ model indeed achieved the best SSIM and PSNR values for almost all test cases. However, the SSIM and PSNR values for both the CelebA-HQ and FFHQ models are significantly closer and fall within one standard deviation from each other in all of the test cases. This could possibly be solved by increasing the number of images with which such metrics are calculated, but for the purpose of this project, the qualitative evaluation presented in this chapter up until this point is sufficiently effective to assess the presented models.

Additionally, although the Places2 model completely failed to perform the facial inpainting tasks, its SSIM and PSNR values were only slightly lower than those of the CelebA-HQ and FFHQ models, also falling within the range of their standard deviations, which demonstrates the limitations of quantitative metrics for evaluating inpainting results, as previously discussed in subsection 3.2.2 ”Qualitative Evaluation”. Nonetheless, the SSIM and PSNR values obtained by the CelebA-HQ and FFHQ models are quite satisfactory when compared to the range of SSIM and PSNR values commonly found in the literature for other deep learning inpainting methods.

Conclusion

In conclusion, upon the detailed evaluation presented in this chapter, 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 providing a historic overview on image and facial inpainting techniques, their origins and development until this day, by presenting a thorough review of the most prominent and relevant digital image and facial inpainting methods proposed in the last decades, and 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 STEWART, J. *Interview: Artist Dedicates Her Spare Time to Restoring People's Precious Photos.* My Modern Met, 2019. Disponível em: ⟨https://mymodernmet.com/michelle-spalding-photo-restoration/⟩. Acesso em: 2021-06-04.

2 EYCK, J. van. *Arnolfini Portrait, painting by Jan van Eyck (1434).* London, United Kingdom: The National Gallery, 1434. Disponível em: ⟨https://www.nationalgallery.org.uk/paintings/jan-van-eyck-the-arnolfini-portrait⟩. Acesso em: 2021-06-04.

3 VINCI, L. da. *Mona Lisa, painting by Leonardo da Vinci (1503).* San Francisco, California: Wikimedia Foundation, 1503. Disponível em: ⟨https://upload.wikimedia.org/wikipedia/commons/thumb/e/ec/Mona_Lisa,_by_Leonardo_da_Vinci,_from_C2RMF_retouched.jpg/687px-Mona_Lisa,_by_Leonardo_da_Vinci,_from_C2RMF_retouched.jpg⟩. Acesso em: 2021-06-04.

4 FIELD, R. *Portrait miniature of George Washington, by Robert Field (1800).* San Francisco, California: Wikimedia Foundation, 1800. Disponível em: ⟨https://upload.wikimedia.org/wikipedia/commons/thumb/2/27/GeorgeWashingtonByRobertField.jpg/884px-GeorgeWashingtonByRobertField.jpg⟩. Acesso em: 2021-06-04.

5 THOTTAM, I. *The Cost of Conservation and Restoration.* Art Business News, 2015. Disponível em: ⟨https://artbusinessnews.com/2015/12/the-cost-of-conservation-and-restoration/⟩. Acesso em: 2021-06-03.

6 SCHRIEVER, J.; ART, A. S. of; (SCRANTON, P. P. *Complete Self-instructing Library of Practical Photography.* American School of Art and Photography, 1909. (Complete Self-instructing Library of Practical Photography, v. 4). Disponível em: ⟨https://archive.org/details/completeselfinst07schriala/mode/2up⟩.

7 GESSEN, M. *The Photo Book That Captured How the Soviet Regime Made the Truth Disappear.* The New Yorker, 2018. Disponível em: ⟨https://www.newyorker.com/culture/photo-booth/the-photo-book-that-captured-how-the-soviet-regime-made-the-truth-disappear⟩. Acesso em: 2021-06-05.

8 ELHARROUSS, O. et al. Image inpainting: A review. *Neural Processing Letters*, Springer, v. 51, n. 2, p. 2007–2028, 2020.

9 ZENG, Y. et al. High-resolution image inpainting with iterative confidence feedback and guided upsampling. In: SPRINGER. *European Conference on Computer Vision.* [S.l.], 2020. p. 1–17.

10 BOFFEY, D. *AI helps return Rembrandt's The Night Watch to original size.* Guardian News and Media, 2021. Disponível em: ⟨https://www.theguardian.com/artanddesign/2021/jun/23/ai-helps-return-rembrandts-the-night-watch-to-original-size⟩. Acesso em: 2021-06-30.

11 EFROS, A. A.; FREEMAN, W. T. Image quilting for texture synthesis and transfer. In: *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*. [S.l.: s.n.], 2001. p. 341–346.

12 CRIMINISI, A.; PÉREZ, P.; TOYAMA, K. Region filling and object removal by exemplar-based image inpainting. *IEEE Transactions on image processing*, IEEE, v. 13, n. 9, p. 1200–1212, 2004.

13 BARNES, C. et al. Patchmatch: A randomized correspondence algorithm for structural image editing. *ACM Trans. Graph.*, v. 28, n. 3, p. 24, 2009.

14 TELEA, A. An image inpainting technique based on the fast marching method. *Journal of graphics tools*, Taylor & Francis, v. 9, n. 1, p. 23–34, 2004.

15 BERTALMIO, M. et al. Image inpainting. In: *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*. [S.l.: s.n.], 2000. p. 417–424.

16 SHEN, B. et al. Image inpainting via sparse representation. In: IEEE. *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*. [S.l.], 2009. p. 697–700.

17 TSCHUMPERLÉ, D.; DERICHE, R. Vector-valued image regularization with pdes: A common framework for different applications. *IEEE transactions on pattern analysis and machine intelligence*, IEEE, v. 27, n. 4, p. 506–517, 2005.

18 MEUR, O. L.; GAUTIER, J.; GUILLEMOT, C. Examplar-based inpainting based on local geometry. In: IEEE. *2011 18th IEEE international conference on image processing*. [S.l.], 2011. p. 3401–3404.

19 HITAWALA, S. Comparative study on generative adversarial networks. *arXiv preprint arXiv:1801.04271*, 2018.

20 PATHAK, D. et al. Context encoders: Feature learning by inpainting. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. [S.l.: s.n.], 2016. p. 2536–2544.

21 YANG, C. et al. High-resolution image inpainting using multi-scale neural patch synthesis. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. [S.l.: s.n.], 2017. p. 6721–6729.

22 LI, C.-T. *A Milestone in Deep Image Inpainting - Review: Globally and Locally Consistent Image Completion*. Towards Data Science, 2020. Disponível em: ⟨https://towardsdatascience.com/a-milestone-in-deep-image-inpainting-review-globally-and-locally-consistent-image-completion-505413c300df⟩. Acesso em: 2021-06-08.

23 IIZUKA, S.; SIMO-SERRA, E.; ISHIKAWA, H. Globally and locally consistent image completion. *ACM Transactions on Graphics (ToG)*, ACM New York, NY, USA, v. 36, n. 4, p. 1–14, 2017.

24 YU, J. et al. Generative image inpainting with contextual attention. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. [S.l.: s.n.], 2018. p. 5505–5514.

25 LIU, G. et al. Image inpainting for irregular holes using partial convolutions. In: *Proceedings of the European Conference on Computer Vision (ECCV)*. [S.l.: s.n.], 2018. p. 85–100.

26 NAZERI, K. et al. Edgeconnect: Generative image inpainting with adversarial edge learning. *arXiv preprint arXiv:1901.00212*, 2019.

27 YU, J. et al. Free-form image inpainting with gated convolution. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. [S.l.: s.n.], 2019. p. 4471–4480.

28 EVERINGHAM, M. et al. The pascal visual object classes (voc) challenge. *International journal of computer vision*, Springer, v. 88, n. 2, p. 303–338, 2010.

29 DENG, J. et al. Imagenet: A large-scale hierarchical image database. In: IEEE. *2009 IEEE conference on computer vision and pattern recognition*. [S.l.], 2009. p. 248–255.

30 DOERSCH, C. et al. What makes paris look like paris? *Communications of the ACM*, ACM New York, NY, USA, v. 58, n. 12, p. 103–110, 2015.

31 LIU, Z. et al. Deep learning face attributes in the wild. In: *Proceedings of the IEEE international conference on computer vision*. [S.l.: s.n.], 2015. p. 3730–3738.

32 KARRAS, T. et al. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017.

33 ZHOU, B. et al. Places: A 10 million image database for scene recognition. *IEEE transactions on pattern analysis and machine intelligence*, IEEE, v. 40, n. 6, p. 1452–1464, 2017.

34 ISKAKOV, K. Semi-parametric image inpainting. *arXiv preprint arXiv:1807.02855*, 2018.

35 KARRAS, T.; LAINE, S.; AILA, T. A style-based generator architecture for generative adversarial networks. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2019. p. 4401–4410.

36 JAM, J. et al. A comprehensive review of past and present image inpainting methods. *Computer Vision and Image Understanding*, Elsevier, p. 103147, 2020.

37 COOMBS, K. *The portrait miniature in England*. London, United Kingdom: V& A Publications, 1998. ISBN 1-85177-207-3.

38 GERNSHEIM, H. *A concise history of photography*. [S.l.]: Courier Corporation, 1986. ISBN 978-0486251288.

39 WALDEN, S. *The ravished image: or how to ruin masterpieces by restoration*. [S.l.]: Weidenfeld and Nicholson; St. Martins, 1985. ISBN 0-297-78407-2.

40 EMILE-MALE, G. The restorer's bandbook of easel painting. Van Nostrand Reinhold Co., 1976.

41 CONTI, A.; GLANVILLE, H. *History of the Restoration and Conservation of Works of Art*. [S.l.]: Elsevier; Butterworth-Heinemann, 2007. ISBN 0-7506-6953-5.

42 MUIR, K. Approaches to the reintegration of paint loss: theory and practice in the conservation of easel paintings. *Studies in Conservation*, Taylor & Francis, v. 54, n. sup1, p. 19–28, 2009.

43 OFFICE, I. M.; CO-OPERATION, I. I. of I.; MUSEUMS, I. C. of. *Manual on the Conservation of Paintings.* [S.l.]: Archetype Publications, 1997. ISBN 9781873132418.

44 FINEMAN, M. et al. *Faking it: Manipulated Photography Before Photoshop.* [S.l.]: Metropolitan Museum of Art, 2012. (Metropolitan Museum of Art). ISBN 9781588394736.

45 KING, D. *The Commissar Vanishes: The Falsification of Photographs and Art in Stalin's Russia.* [S.l.]: Henry Holt and Company, 1997. ISBN 9780805052947.

46 KHODADADI, M.; BEHRAD, A. Text localization, extraction and inpainting in color images. In: IEEE. *20th Iranian Conference on Electrical Engineering (ICEE2012).* [S.l.], 2012. p. 1035–1040.

47 PARK, E. et al. Transformation-grounded image generation network for novel 3d view synthesis. In: *Proceedings of the ieee conference on computer vision and pattern recognition.* [S.l.: s.n.], 2017. p. 3500–3509.

48 XIE, J.; XU, L.; CHEN, E. Image denoising and inpainting with deep neural networks. *Advances in neural information processing systems*, Citeseer, v. 25, p. 341–349, 2012.

49 CHEN, Y.; RANFTL, R.; POCK, T. A bi-level view of inpainting-based image compression. *arXiv preprint arXiv:1401.4112*, 2014.

50 SETLUR, V. et al. Automatic image retargeting. In: *Proceedings of the 4th international conference on Mobile and ubiquitous multimedia.* [S.l.: s.n.], 2005. p. 59–68.

51 LEVIN, A. et al. Seamless image stitching in the gradient domain. In: SPRINGER. *European Conference on Computer Vision.* [S.l.], 2004. p. 377–389.

52 SILVA, B. F.; FAUSTINO, G. S. *Algoritmo para Confecção de Próteses Nasais.* São Paulo, Brazil: Escola Politécnica da Universidade de São Paulo, Departamento de Engenharia Mecatrônica e de Sistemas Mecânicos, 2019.

53 OPERATION Night Watch - Rijksmuseum. Rijksmuseum, 2021. Disponível em: ⟨https://www.rijksmuseum.nl/en/stories/operation-night-watch⟩. Acesso em: 2021-06-30.

54 CRIDDLE, C. *Rembrandt's The Night Watch painting restored by AI.* BBC, 2021. Disponível em: ⟨https://www.bbc.com/news/technology-57588270⟩. Acesso em: 2021-06-30.

55 EFROS, A. A.; LEUNG, T. K. Texture synthesis by non-parametric sampling. In: IEEE. *Proceedings of the seventh IEEE international conference on computer vision.* [S.l.], 1999. v. 2, p. 1033–1038.

56 RAAD, L.; GALERNE, B. Efros and freeman image quilting algorithm for texture synthesis. *Image Processing On Line*, v. 7, p. 1–22, 2017.

57 LAI, Y.-K. et al. Geometric texture synthesis and transfer via geometry images. In: *Proceedings of the 2005 ACM symposium on Solid and physical modeling.* [S.l.: s.n.], 2005. p. 15–26.

58 BERTALMIO, M. et al. Simultaneous structure and texture image inpainting. *IEEE transactions on image processing*, IEEE, v. 12, n. 8, p. 882–889, 2003.

59 MEUR, O. L.; GUILLEMOT, C. Super-resolution-based inpainting. In: SPRINGER. *European Conference on Computer Vision.* [S.l.], 2012. p. 554–567.

60 BUGEAU, A. et al. A comprehensive framework for image inpainting. *IEEE transactions on image processing*, IEEE, v. 19, n. 10, p. 2634–2645, 2010.

61 BERTALMIO, M.; BERTOZZI, A. L.; SAPIRO, G. Navier-stokes, fluid dynamics, and image and video inpainting. In: IEEE. *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001.* [S.l.], 2001. v. 1, p. I–I.

62 CHAN, T. F.; SHEN, J. Nontexture inpainting by curvature-driven diffusions. *Journal of visual communication and image representation*, Elsevier, v. 12, n. 4, p. 436–449, 2001.

63 CHAN, T.; SHEN, J. Mathematical models for local deterministic inpaintings. *UCLA CAM TR*, p. 00–11, 2000.

64 SHEN, J.; KANG, S. H.; CHAN, T. F. Euler's elastica and curvature-based inpainting. *SIAM journal on Applied Mathematics*, SIAM, v. 63, n. 2, p. 564–592, 2003.

65 SETHIAN, J. A. A fast marching level set method for monotonically advancing fronts. *Proceedings of the National Academy of Sciences*, National Acad Sciences, v. 93, n. 4, p. 1591–1595, 1996.

66 MAIRAL, J.; ELAD, M.; SAPIRO, G. Sparse representation for color image restoration. *IEEE Transactions on image processing*, IEEE, v. 17, n. 1, p. 53–69, 2007.

67 JAIN, V.; SEUNG, S. Natural image denoising with convolutional networks. *Advances in neural information processing systems*, v. 21, p. 769–776, 2008.

68 EIGEN, D.; KRISHNAN, D.; FERGUS, R. Restoring an image taken through a window covered with dirt or rain. In: *Proceedings of the IEEE international conference on computer vision.* [S.l.: s.n.], 2013. p. 633–640.

69 KÖHLER, R. et al. Mask-specific inpainting with deep neural networks. In: SPRINGER. *German conference on pattern recognition.* [S.l.], 2014. p. 523–534.

70 RONNEBERGER, O.; FISCHER, P.; BROX, T. U-net: Convolutional networks for biomedical image segmentation. In: SPRINGER. *International Conference on Medical image computing and computer-assisted intervention.* [S.l.], 2015. p. 234–241.

71 GOODFELLOW, I. J. et al. Generative adversarial networks. *arXiv preprint arXiv:1406.2661*, 2014.

72 LECUN, Y. *What are some recent and potentially upcoming breakthroughs in deep learning?* 2016. Disponível em: ⟨https://www.quora.com/What-are-some-recent-and-potentially-upcoming-breakthroughs-in-deep-learning/answer/Yann-LeCun⟩. Acesso em: 2021-06-06.

73 JOHNSON, J.; ALAHI, A.; FEI-FEI, L. Perceptual losses for real-time style transfer and super-resolution. In: SPRINGER. *European conference on computer vision.* [S.l.], 2016. p. 694–711.

74 ULYANOV, D. et al. Texture networks: Feed-forward synthesis of textures and stylized images. In: *ICML*. [S.l.: s.n.], 2016. v. 1, n. 2, p. 4.

75 LI, C.; WAND, M. Combining markov random fields and convolutional neural networks for image synthesis. In: *Proceedings of the IEEE conference on computer vision and pattern recognition.* [S.l.: s.n.], 2016. p. 2479–2486.

76 YEH, R. A. et al. Semantic image inpainting with deep generative models. In: *Proceedings of the IEEE conference on computer vision and pattern recognition.* [S.l.: s.n.], 2017. p. 5485–5493.

77 CANNY, J. A computational approach to edge detection. *IEEE Transactions on pattern analysis and machine intelligence*, Ieee, n. 6, p. 679–698, 1986.

78 ISOLA, P. et al. Image-to-image translation with conditional adversarial networks. In: *Proceedings of the IEEE conference on computer vision and pattern recognition.* [S.l.: s.n.], 2017. p. 1125–1134.

79 MIYATO, T. et al. Spectral normalization for generative adversarial networks. *arXiv preprint arXiv:1802.05957*, 2018.

80 DEMIR, U.; UNAL, G. Patch-based image inpainting with generative adversarial networks. *arXiv preprint arXiv:1803.07422*, 2018.

81 YAN, Z. et al. Shift-net: Image inpainting via deep feature rearrangement. In: *Proceedings of the European conference on computer vision (ECCV).* [S.l.: s.n.], 2018. p. 1–17.

82 WANG, Y. et al. Image inpainting via generative multi-column convolutional neural networks. *arXiv preprint arXiv:1810.08771*, 2018.

83 HUANG, Y. et al. Image completion based on gans with a new loss function. In: IOP PUBLISHING. *Journal of Physics: Conference Series.* [S.l.], 2019. v. 1229, n. 1, p. 012030.

84 ZENG, Y. et al. Learning pyramid-context encoder network for high-quality image inpainting. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.* [S.l.: s.n.], 2019. p. 1486–1494.

85 XIONG, W. et al. Foreground-aware image inpainting. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.* [S.l.: s.n.], 2019. p. 5840–5848.

86 LI, J. et al. Progressive reconstruction of visual structure for image inpainting. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision.* [S.l.: s.n.], 2019. p. 5962–5971.

87 LIU, H. et al. Coherent semantic attention for image inpainting. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision.* [S.l.: s.n.], 2019. p. 4170–4179.

88 LI, J. et al. Recurrent feature reasoning for image inpainting. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.* [S.l.: s.n.], 2020. p. 7760–7768.

89 YI, Z. et al. Contextual residual aggregation for ultra high-resolution image inpainting. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.* [S.l.: s.n.], 2020. p. 7508–7517.

90 LI, Y. et al. Generative face completion. In: *Proceedings of the IEEE conference on computer vision and pattern recognition.* [S.l.: s.n.], 2017. p. 3911–3919.

91 LI, X. et al. Learning symmetry consistent deep cnns for face completion. *IEEE Transactions on Image Processing*, IEEE, v. 29, p. 7641–7655, 2020.

92 ZHOU, T. et al. Learning oracle attention for high-fidelity face completion. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.* [S.l.: s.n.], 2020. p. 7680–7689.

93 WANG, Q. et al. Laplacian pyramid adversarial network for face completion. *Pattern Recognition*, Elsevier, v. 88, p. 493–505, 2019.

94 PORTENIER, T. et al. Faceshop: Deep sketch-based face image editing. *arXiv preprint arXiv:1804.08972*, 2018.

95 JO, Y.; PARK, J. Sc-fegan: Face editing generative adversarial network with user's sketch and color. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision.* [S.l.: s.n.], 2019. p. 1745–1753.

96 CHEN, Z. et al. High resolution face completion with multiple controllable attributes via fully end-to-end progressive generative adversarial networks. *arXiv preprint arXiv:1801.07632*, 2018.

97 HUI, Z. et al. Image fine-grained inpainting. *arXiv preprint arXiv:2002.02609*, 2020.

98 CHANG, Y.-L. et al. Free-form video inpainting with 3d gated convolution and temporal patchgan. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision.* [S.l.: s.n.], 2019. p. 9066–9075.

99 ZENG, Y.; FU, J.; CHAO, H. Learning joint spatial-temporal transformations for video inpainting. In: SPRINGER. *European Conference on Computer Vision.* [S.l.], 2020. p. 528–543.

100 LI, C.-T. et al. Deepgin: Deep generative inpainting network for extreme image inpainting. In: SPRINGER. *European Conference on Computer Vision.* [S.l.], 2020. p. 5–22.

101 ZHENG, C.; CHAM, T.-J.; CAI, J. Pluralistic image completion. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2019. p. 1438–1447.

102 CAI, W.; WEI, Z. Piigan: Generative adversarial networks for pluralistic image inpainting. *IEEE Access*, IEEE, v. 8, p. 48451–48463, 2020.

103 ZHAO, L. et al. Uctgan: Diverse image inpainting based on unsupervised cross-space translation. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2020. p. 5741–5750.

104 WANG, Z.; BOVIK, A. C. A universal image quality index. *IEEE signal processing letters*, IEEE, v. 9, n. 3, p. 81–84, 2002.

105 WANG, Z.; SIMONCELLI, E. P.; BOVIK, A. C. Multiscale structural similarity for image quality assessment. In: IEEE. *The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003*. [S.l.], 2003. v. 2, p. 1398–1402.

106 SHEIKH, H. R.; BOVIK, A. C. Image information and visual quality. *IEEE Transactions on image processing*, IEEE, v. 15, n. 2, p. 430–444, 2006.

107 SALIMANS, T. et al. Improved techniques for training gans. *Advances in neural information processing systems*, v. 29, p. 2234–2242, 2016.

108 HEUSEL, M. et al. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems*, v. 30, 2017.

109 ZHANG, R. et al. The unreasonable effectiveness of deep features as a perceptual metric. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. [S.l.: s.n.], 2018. p. 586–595.

110 LI, C.; WAND, M. Precomputed real-time texture synthesis with markovian generative adversarial networks. In: SPRINGER. *European conference on computer vision*. [S.l.], 2016. p. 702–716.

111 XIE, S.; TU, Z. Holistically-nested edge detection. In: *Proceedings of the IEEE international conference on computer vision*. [S.l.: s.n.], 2015. p. 1395–1403.

112 @XHLULU. *Flickr-Faces-HQ Dataset (Nvidia) - Resized 256px*. Kaggle, 2020. Disponível em: ⟨https://www.kaggle.com/xhlulu/flickrfaceshq-dataset-nvidia-resized-256px⟩. Acesso em: 2021-10-12.

113 ANTONACIO, P. O. *Completing Face Pictures: a Study on Image and Facial Inpainting Methods*. [S.l.]: GitHub, 2021. ⟨https://github.com/PedroAntonacio/facial-inpainting⟩.