# Restoration of Damaged Face Statues Using Deep Generative Inpainting Model

Master Thesis

Supervised by Dr.Ir. L.J. Spreeuwers ($1_{st}$) and Dr. D.V. Le Viet Duc ($2_{nd}$)

Abraham Theodorus
University of Twente, M-CS-DST
abrahamtheodorus@student.utwente.nl

## 1. Introduction

The History department of the University of Nijmegen has started a joint-research initiative with the Data Science group of the University of Twente in order to investigate an image collection of ancient face statues using computer vision techniques. Although some statues are in a perfect condition, some statues suffer from damages on face attributes, as observed in Figure 1. Therefore, one possible use case is to restore the damages, so that the image collection can still be showcased in a digital gallery.

Image restoration is one of the most active areas of research in computer vision domain. It is typically an ill-posed inverse problem where the restored image candidate is produced by approximating the original form of the degraded image. Image restoration tasks can be further specialized into image denoising [16], super-resolution [6], and inpainting [34]. The damaged face statues restoration task can be considered as an image inpainting task which aims to fill missing regions of an image with pixels which are fit semantically. In this case, the damaged area can be marked as target regions to be inpainted.

Researchers have proposed various approaches to tackle image inpainting problems. Generally, they fall into two categories, i.e., traditional algorithms and deep learning-based generative models.

Some traditional algorithms look for similar spatial patterns in the image in order to fill the missing regions [19, 35, 3]. This approach works well in texture synthesis and object removal where after the removal, the missing regions need to be blended into a repetitive background (e.g. grass, wall of bricks, and fence). However, these approaches suffer when the inpainting region is large and belongs to a rather complex scene, i.e. body parts or human face.

Figure 1. Damaged statues along with their annotated damage regions. As observed, the damaged regions are mostly located on the face attributes within certain radius.

On the other hand, deep generative models, i.e., Variational Auto Encoder (VAE) [9] and Generative Adversarial Network (GAN) [34, 25], are able to capture non-linear representations of complicated scenes by relying on Convolutional Neural Network (CNN). Coupled with proper network architectures and objective functions, they are able to generate more semantic-aware inpainting regions. GAN-based solution, especially, has become the state of the art for image inpainting task.

In this thesis, research on how to tackle the face statues inpainting problem by utilizing a GAN-based model is carried out. In addition, a comparative study with one of the conventional algorithms will also be conducted.

The proposed GAN model follows the architecture of existing GAN inpainting models which comprises an encoder-decoder network as the generator and a classi-

fier as the discriminator. For the conventional algorithm a PCA face reconstruction approach [42], which has been proposed by Wang *et al.*, is implemented as a contender. Due to the limited number of available face statue images, data augmentation strategies will be discussed further.

All in all, by the end of my thesis project, the following research questions will have been answered:

1. What kind of damages do the face statues suffer from?
2. How to construct a dataset for training the generative models?
3. How to design a recursive PCA approach for solving the image inpainting problem?
4. How to train a GAN model for solving the image inpainting problem?

   - Which loss functions are the most contributive towards the outcome?
5. How to evaluate the inpainting results?

   - How does the recursive PCA perform compared to the GAN model in restoring the missing attributes of the face statues?

The rest of this thesis will be structured as follows. The literature study results are discussed in Section 2, then the methodology and experiments will be elaborated in Section 3 and Section 4 respectively. Meanwhile, experiment results will be displayed in Section 5 and further discussed in Section 6. Finally, the conclusion would be presented in Section 7.

## 2. Related Work

### 2.1. Image Inpainting

In the early period, the proposed image inpainting solutions were based on *diffusion using partial differential equations (PDEs)*, introduced by Bertalmio *et al.* [4], which propagates image geometric information from the missing regions' border [2, 41] inward. The particular approach was able to fill in small holes but fails to fill textured regions.

*Exemplar*-based method then emerged to recover the missing region by doing sampling on other image patches that best match its surrounding. The proposed solutions are usually based on Markov Random Fields (MRF) [19, 35, 37] and nearest-neighbor algorithm [3, 10]. Although this approach certainly helps on producing realistic textures, it is unable to inpaint unique scenes, i.e., body parts, human faces, and occluded objects.

Another strategy is to rely on external references as guidance to fill the missing regions. In face occlusion removal case, Mo *et al.* [31] retrieve the closest image patch obtained from a database by measuring the similarity between the target's surrounding pixels and the references. Lee *et al.* [22] use a similar approach, but average patch is used instead.

Face images have similar structural patterns, therefore they can be modeled as a linear combination of lower-dimensional components, represented by eigenvectors. Thus, many Principal Component Analysis (PCA) reconstruction methods were also proposed [42, 33, 32, 43]. PCA approach recovers the missing regions by iteratively retrieving pixel candidates of the missing regions from the face space. The authors claimed to produce natural reconstructed human faces. Since the face statues resemble human faces and the observed damages on the face statues are comparable to the missing regions in [42], this algorithm might work on the face statues inpainting task too. Most of the reconstructed faces of these approaches look natural, but still lack contextual alignment, such as gender, race, and ethnicity.

### 2.2. Deep Generative Models

The superior performance of deep learning in image processing and computer vision tasks has an impact on the insurgence of deep generative models. While traditional approaches typically craft features manually through some rule-based methods or certain algorithms, deep learning-based models are capable of learning non-linear features from the training data which are extracted by using Convolutional Neural Networks (CNN) [21]. However, prior to CNN architecture, early deep learning adoption in image restoration task uses fully connected Multi Layer Perceptrons (MLPs) [5, 38].

Encoder-decoder architecture is one of the CNN architecture variant that is used for solving the image inpainting problem. It contains sequences of convolution layers to encode input images to latent representation, then followed by sequences of de-convolution layers to upsample the representation back into an image. Mao *et al.* [30] demonstrates an auto-encoder architecture to solve the image inpainting problem along with several other image restoration tasks with Mean Squared Error (MSE) as the reconstruction loss to be minimized between the degraded image and its ground-truth. Unfortunately, the inpainting results of an auto-encoder tend to be blurry because the typically used MSE loss smoothens the inpainted region by computing the mean of the ground-truth pixels.

Generative Adversarial Networks (GAN) [8] defines a new paradigm for generating images. GAN uses two networks, called the generator and the discriminator. Its training objective can be considered as a mini-max game, where the generator needs to generate fake realistic images out of a known prior distribution to fool the discrim-

inator, while the discriminator needs to distinguish between the real and the generated fake images. The objective function is called the adversarial loss. Since GAN was first introduced, many proposed image processing and computer vision tasks are based on the GAN framework. The mage inpainting task is no exception. Pathak *et al.* [34] is one of the first that proposes a combination of an auto-encoder architecture and the adversarial loss for image inpainting. Different with the original GAN framework, the generator of a GAN inpainting model produces an inpainted image $I_{inp}$ conditioned on a degraded image $I_{deg}$. The inclusion of the adversarial loss is the key to get more photo-realistic image inpainting results.

The subsequently proposed improvements revolve around designing better adversarial loss [1, 29, 46] and better network architectures [18], adding more optimization objectives [23, 14], and also devising new GAN training strategies [17, 39]. Wasserstein-GAN (WGAN) architecture, introduced by Arjovsky *et al.* [1], minimizes Earth Mover (EM) distance between probability distribution of the real data and the generated data, thus leading to a more stable training and more meaningful loss metric. Progressive-GAN [17]employs a curriculum learning strategy [39] where a GAN model is trained on multiple stages of the image generation task with increasing difficulty, starting from generating low to high resolution images. Moreover, as the training stage progresses, the parameters and layers are also increased incrementally.

To solve an image inpainting problem, Yu *et al.* and Song *et al.* train GAN in two stages, first they train a coarse GAN to inpaint a rough inpainting estimation, followed by a refined GAN which enhances the inpainting quality [44, 40]. Iizuka *et al* and Li *et al.* use two adversarial losses [14, 25], local and global loss, to evaluate the generated inpainted patches and the whole image consistency respectively. Li *et al.* pre-fill the inpainting target with captured symmetry of the input image [24]. Meanwhile, Li *et al.* add a segmentation loss function [25] to enforce position awareness of the generated face attributes.

Regardless of existing sophisticated GAN architectures, this thesis project adopts relatively simpler GAN architectures, i.e the original Minimax GAN as well as the Wasserstein-GAN (WGAN). Additionally, the classical recursive PCA algorithm performs adequately in face inpainting tasks, therefore this method will act as a baseline comparison.

## 3. Methodology

In this face statue inpainting task, the damages are represented by some missing regions. One way to train the inpainting models is by first constructing a dataset of 'normal - missing region' face statue pairs. Ideally, a dataset of 'normal - damaged' pairs is even more desirable, but, unfortunately, such pairs are non-existent in the dataset. Therefore, the damages are simulated on the normal statues instead. The damage simulation is elaborated in this section. Moreover, two approaches for solving the face statue inpainting task, GAN and recursive PCA reconstruction algorithm, will also be discussed further.

Figure 2. Face landmarks detected by using pre-trained MTCNN model. Then, the statue images were cropped and centered with respect to the proportion between the face landmarks and the image borders.

### 3.1. Simulating damages through binary masks

The damages are simulated by applying artificial binary masks on the normal statue images, such that, for each unique normal statue there are multiple damaged versions of the statue. This approach may also be considered as a data augmentation step. The simulated damages need to look as similar as possible to the real damaged face statues, therefore, the structural form of the damages follows an observation on the existing damaged face statues. Originally, three types of damages are identified, i.e. *missing face attributes*, *eroded textures*, and *hollow regions*. Based on visual inspection, most of the damaged statues possess some *missing face attributes*, hence, the developed solution will only tackle this damage type to limit the scope of this thesis. Some of the damaged statues are showcased in Figure 1.

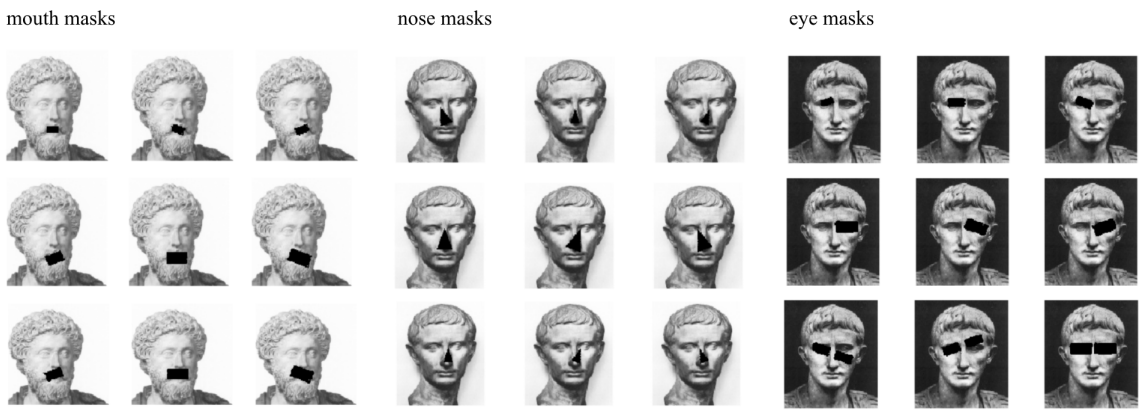


Figure 3. There are three major mask groups, i.e., mouth masks, nose masks, and eye masks. The masks are augmented by considering different mask sizes, position with respect to the attributes' centroids, as well as rotations.

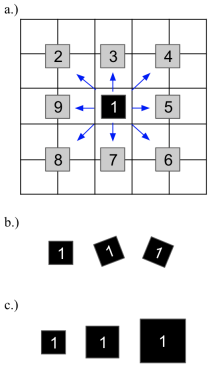


Figure 4. Binary masks data augmentation includes: a.) mask centroid translations b.) mask rotations (0°, −20°, 20°) c.) mask scaling. In a.), the square number 1 represents the initial mask centroid coodinates, meanwhile the grey squares are the mask position after being translated. In b.) and c.), the initial masks are represented by the first squares, whereas the subsequent squares represent the transformed masks.

Localization of the face attributes and face alginment are necessary prior to simulating the *missing face attributes* damage. In this case, face landmark detection is performed on the normal face statues by utilizing a pretrained MTCNN [45]. The MTCNN model is able to detect 5 face landmarks' coordinates (left eye, right eye, nose, left mouth corner, and right mouth corner) and the inferred face landmark coordinates are observed to have sufficient quality. Based on the obtained coordinates, the face statues are aligned, cropped, and centered with respect to the proportion between the coordinates and the image borders. The localization results are displayed in Figure 2.

The face landmarks are assigned into 3 major groups, i.e, eyes, nose, and mouth. Each landmark group will then have its own unique binary mask shape which is relatively able to fully cover the corresponding face attributes. The eye and mouth landmark groups have similarly rectangular binary masks, except the mouth masks have slightly thinner width to match the lips' width proportion. Meanwhile, the nose masks have an isosceles triangle shape. The dataset is augmented further by varying the masks' properties, i.e., the mask positions with respect to the face attributes centroids, mask sizes, and also rotations. The resulting binary masks are showcased in Figure 3, whereas the augmentation lists are depicted in Figure 4.

### 3.2. Recursive PCA

PCA reconstruction algorithm learns linear representation of face structures in the training data, thus, no 'damaged - normal' face statue pairs are required. The binary masks are only utilized during inference phase on the test set.

PCA reconstruction of a face statue image follows a classical PCA face reconstruction formula [42].

$$x_{train} = m_{train} + \sum_{i=1}^{K} y_i v_i \tag{1}$$

Given a train image of N x N pixels, $x$ corresponds to the unrolled $N^2$-dimensional vector of the image and $m_{train}$ denotes the mean face of the training set. Meanwhile, $v$ corresponds to the $K$ selected eigenfaces and $y$ are the coefficients for the linear combination of $v$.

Analytically, the coefficient vector $y_{test}$ of an unseen face statue image $x_{test}$ can be obtained by applying Equation 2.

$$y_{test} = v_{train}^{T} \cdot (x_{test} - m_{train}) \tag{2}$$

Then the new reconstructed image $x_{test}^{'}$ is retrieved by applying Equation 3 on the test image $x_{test}$.

$$x^{'}_{test} = m_{train} + \sum_{i=1}^{K} y_{test,i} v_{train,i} \tag{3}$$

However, in face statue inpainting problem only a portion of the image marked by the binary mask $M$ needs to be reconstructed. Therefore, Equation 3 can be refined further.

$$x^{'}_{test} = M \circ x^{'}_{test} + (1 - M) \circ x_{test} \tag{4}$$

Then, the inpainting process of image $x_{test}$ is repeated by applying Equation 2-4 again iteratively. Therefore, all of the equations can be written with respect to a time-step $t$, where the original unrolled test image is represented by $x^t$ at $t = 0$.

$$\begin{aligned} y^t &= v^T_{train} \cdot (x^t - m_{train}) \\ x^{t+1} &= m_{train} + \sum_{i=1}^{K} y^t_i v_{train,i} \\ x^{t+1} &= M \circ x^{t+1} + (1 - M) \circ x^t \end{aligned} \tag{5}$$

The iteration stops when the maximum absolute difference between coefficient vector $y^{t+1}$ and $y^t$ reaches certain threshold $\epsilon$.

$$\max(\| y^{t+1} - y^t \|) < \epsilon \tag{6}$$

## 3.3. Utilizing GAN for face statue inpainting

### 3.3.1 Minimax GAN

The original GAN [8] employs simultaneous optimization of two networks, namely the generator and the discriminator. This optimization objective is also called the adversarial loss and it can be illustrated in Equation 7.

$$\begin{aligned} L_{adv}(G, D, X, Z) = \min_G \max_D & \mathbb{E}_{x \sim p_{data}(x)}[\log(D(x))] + \\ & \mathbb{E}_{z \sim p_Z(z)}[\log(1 - D(G(z)))] \end{aligned} \tag{7}$$

The generator $G$ tries to generate fake images from a distribution of noise $p_z$ as realistic as possible to fool the discriminator, at the same time the discriminator $D$ tries to distinguish the real images sampled from ground-truth distribution $p_{data}$ and fake images induced by the generator $G(z)$ as well as possible. The model training goes on by optimizing a minimax objective function until Nash Equilibrium has been reached.

Moreover, the adversarial loss, which the Minimax GAN is optimizing on, reaches a global minimum when the Jensen-Shannon divergence (JSD) is minimized [8]. The adversarial loss is at its minimum if and only if $p_g = p_{data}$. That means, the generator is able to generate samples identical to the distribution $p_{data}$. In this case, the global minimum is equal to $-\log(4)$ when JSD reaches 0 as interpreted from Equation 8.

$$\min L_{adv}(G, D) = -\log(4) + 2.JSD(p_{data} || p_g) \tag{8}$$

**Discriminator**. Discriminator is a key component in the GAN framework. Some believe, the optimization of JSD which is approximated through an adversarial training is actually prominent to the success of the GAN framework [13]. Here, the adversarial training involves an adversarial loss and it is directly minimized by training the discriminator. In Equation 7, $\mathbb{E}_{x \sim p_{data}(x)}[\log(D(x))]$ optimizes how the discriminator is able to discern the real data, therefore the value of $D(x)$ should be close to 1. Meanwhile, $\mathbb{E}_{z \sim p_Z(z)}[\log(1 - D(G(z)))]$ optimizes how the discriminator is able to penalize fake samples induced by the generator, thus the optimized $D(G(z))$ should be close to 0. Note that the real data is labeled as 1, while the fake data is labeled as 0.

In an image inpainted task, the real data is equivalent with the ground-truth images $I_{gt}$, whereas the generated samples are the inpainted images constructed from a collection of masked face statues $(1 - M) \circ I_{gt}$ instead of random noise $z$. Here, $M$ is a binary mask where the inpainted target pixels are denoted by 1. Thus, this leads to an image inpainting adversarial loss described in Equation 9.

$$\begin{aligned} L_{Dadv}(G, D, I_{gt}, M) = \min_G \max_D & \mathbb{E}_{i \sim p_{data}(I_{gt})}[\log(D(i))] + \\ & \mathbb{E}_{i \sim p_{data}(I_{gt})}[\log(1 - D(G((1 - M) \circ i)))] \end{aligned} \tag{9}$$

**Generator**. From generator standpoint, the generator $G$ is trained to produce an inpainted image as realistic as possible given an input image $I_{gt}$ and a binary mask $M$. The adversarial loss is then also applicable for the generator. In contrast with the discriminator's adversarial loss, the generator's adversarial objective is to minimize $\mathbb{E}_{z \sim p_Z(z)}[\log(1 - D(G(z)))]$ by getting $D(G(z))$ as close as possible to 1. Hence, the inpainting generator's adversarial objective is to minimize Equation 10.

$$L_{Gadv}(G, D, I_{gt}, M) = \mathbb{E}_{i \sim p_{data}(I_{gt})}[\log(1 - D(G((1 - M) \circ i)))] \tag{10}$$

### 3.3.2 Wassterstein-GAN

Wassterstein-GAN (WGAN) was introduced by Arjovsky et al. to overcome the original GAN's training instability [1]. WGAN's core idea revolves around minimizing the Earth Mover (EM) distance between probability distributions of real data $p_{data}$ and generated data $p_g$. EM distance measures the distance between two probability distributions

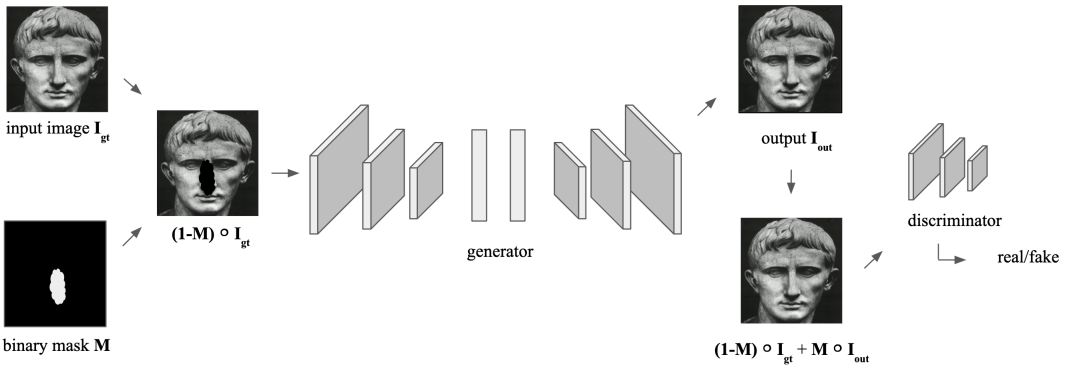


Figure 5. The proposed GAN architecture. Masked input images are fed into an encoder-decoder network with U-Net architecture to generate inpainted images. Then, only portion of the target regions are cropped and then merged with the previously masked inputs. A discriminator with PatchGAN network architecture is utilized to evaluate the final outputs.

and it represents the most optimum cost to move "piles" of distribution from one distribution to another and vice versa.

They have argued that Earth Mover (EM) distance provides a smoother optimization metric compared to Jensen-Shannon divergence (JSD) of GAN [1]. Consider a case where there is no support between $p_{data}$ and $p_g$ and the distance between the two distributions is parameterized by $\theta$. During the optimization, the JSD will constantly output log 2 no matter how far or close the two distributions are, thus it is not providing meaningful gradients. On the other hand, as the EM distance decreases, the value will linearly follow $\theta$. Given this fact, WGAN is claimed to have better training stability compared to the original GAN.

In image inpainting case, the goal of WGAN is to maximize a critic loss:

$$L_{critic} = \max_{\|f\|_L \leq 1} \mathbb{E}_{x \sim p_{data}}[f(x)] - \mathbb{E}_{x \sim p_\theta}[f(x)] \tag{11}$$

This critic loss is the objective function of the discriminator which is originated from the transformed EM distance estimate under 1-Lipschitz continuity [1]. The transformation is necessary due to the intractability of joined probability distribution $\gamma(p_{data}, p_\theta)$. Here, $p_\theta$ represents the parameter distribution of the inpainting generator $g_\theta((1-M) * I_{gt})$.

$$L_G = -\mathbb{E}_{i \sim p_{I_{gt}}}[f(g_\theta((1-M) * i))] \tag{12}$$

The generator, on the other hand, is optimized by minimizing Equation 12 and in the end, the optimized EM distance should be close to 0. Later, the WGAN architecture will be compared with the original Minimax GAN in the face statue inpainting use case.

### 3.4. Generator's objective functions

The sequence of generating inpainted images is as follows. The ground-truth images $I_{gt}$ are multiplied by binary masks $M$ resulting to some masked images $I_m$ in order to simulate damages. The masked images $I_m$ are then fed into the generator G to produce reconstructed images $I_{out} = G(I_m)$. Since only the inpainted regions are desired, the non-masked regions of $I_{out}$ are directly substituted by the ground-truths $I_{gt}$, such that the final outputs $I_{comp} = (1-M) \circ I_{gt} + M \circ I_{out}$ are obtained.

Apart from the adversarial loss, additional objective functions are also typically employed to the generator in order to infer higher fidelity results. Note that, even though only the final outputs $I_{comp}$ are considered in the end, some of the objective functions optimize both $I_{out}$ and $I_{comp}$. This way, the optimizations happen locally on the inpainted region as well as globally on the whole image. The followings are some objective functions which have been previously proven useful for image inpainting tasks [24, 26, 23, 44].

**Reconstruction Loss**. The generator typically uses an auto-encoder architecture. The auto-encoder learns to compress the masked face statues into latent representation then complete the missing pixels while reconstructing the latent representation back to an output image $I_{comp}$. Naturally, an L2 reconstruction loss (Equation 13) is employed additionally to evaluate the inpainted images. L2 reconstruction is used generally in any image reconstruction tasks [citation], even on non GAN approach [citation]. $N_M$ symbolizes the number of pixels being inpainted into masked regions of $I_{comp}$. While $N_I$ represents the total number of pixels of the full image. The denominator is useful to illustrate the reconstruction differences only at the affected pixels.

$$L_{recon} = \frac{\|I_{comp} - I_{gt}\|_2}{N_M} + \frac{\|I_{out} - I_{gt}\|_2}{N_I} \tag{13}$$

**Content Loss**. It was first introduced by Gatys et al. for neural style transfer GAN [7]. For the image inpainting case, the purpose of this loss function (Equation 14) is to reconstruct low level representations of images by minimizing the difference between the ground-truths and the inpainted results in feature space. Hence, it may enforce content similarity between the two images. Additionally, the content loss can also be considered as an identity preserving metric [12]. According to previous studies [25, 26], it improves fidelity of the generated results. Concretly, this is achievable by utilizing a pre-trained VGG-19 model to calculate feature maps of each input in a set of $K$ layers. In particular, they are the first ReLu activation outputs $\psi_k(.)$ of the first 5 convolutional blocks (*relu1_1,relu2_1,relu3_1,relu4_1,relu5_1*).

$$L_{content} = \sum_{k=1}^{K} \frac{\|\psi_k(I_{comp}) - \psi_k(I_{gt})\|_1}{N_{\psi_k(I_{gt})}} + \sum_{k=1}^{K} \frac{\|\psi_k(I_{out}) - \psi_k(I_{gt})\|_1}{N_{\psi_k(I_{gt})}} \tag{14}$$

**Style Loss**. Besides content loss, Gatys et al. [7]. also introduced style loss. This loss function promotes style similarity across generated images by minimizing the difference of feature maps distribution between the ground-truths and generated images. The distribution of feature maps is computed by utilizing feature maps autocorrelation (Gram matrix) on both images. Additionally, the Gram matrix is nothing more than the dot product of feature maps on each layer for the ground-truths and the inpainted results as detailed in Equation 15. Here, $C_k$, $W_k$, and $H_k$ act as normaliizing coefficents and they represent the number of channels, width, and height of the k-th feature maps respectively. In image inpainting use case [26], along with content loss, style loss helps removing artifacts on the inpainted regions.

$$L_{style} = \sum_{k=1}^{K} \frac{\|\psi_k(I_{comp})^T \psi_k(I_{comp}) - \psi_k(I_{gt})^T \psi_k(I_{gt})\|_1}{C_k.W_k.H_k} + \sum_{k=1}^{K} \frac{\|\psi_k(I_{out})^T \psi_k(I_{out}) - \psi_k(I_{gt})^T \psi_k(I_{gt})\|_1}{C_k.W_k.H_k} \tag{15}$$

**Face Parsing Loss**. In order to obtain properly positioned and shaped inpainted face attributes, a face parsing loss is adopted. Given face attribute segmentation outputs of the ground-truth images as well as the inpainted results, weighted cross-entropy loss between the ground-truth's segments and inferred segments of the inpainted image is computed and averaged across the number of segmented pixels. The utilized weights are [0.1,1,0.7,0.7] where each weight corresponds to a segment label [*background,eye,nose,mouth*]. The weights are necessary to reduce the dominance of some face attributes which caused a class imbalance problem. The face attribute segmentation model is trained separately and will be described further in section 4. The generated segments samples are as displayed in Figure 6.

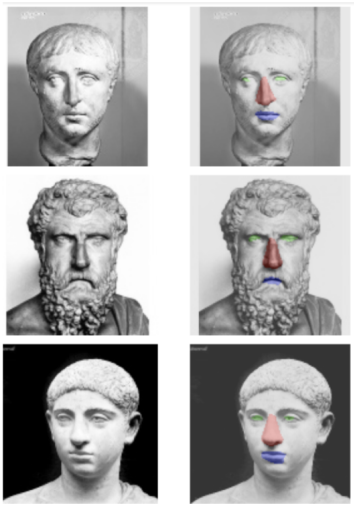

Figure 6. Generated face attribute segments for ground-truth images

$$L_{parsing} = -\sum_{i=0}^{C} y_i \log(s_i) \tag{16}$$

The loss function in Equation 16 is a typical multi-categorical cross-entropy loss where $C$ represents the number of segment classes (which is equal to 3), $y_i$ represents the current true pixel label (it will equal to 1 if the current pixel has a true label $C_i$). And lastly, $s_i$ represents the sigmoid score of the segmentation model.

**Total-Variation Loss**. Total-variation loss [28] ensures smoothness of boundaries between the inpainted regions and the ground-truth region. Here, $W$ and $H$ denote the width and the height of the one-dilated masked region respectively. Specifically, one-dilated mask region means a dilation morphological operation of a 3x3 structural element is applied once on the binary masks. Only one operation is applied because the smoothness is only desired near the inpainted regions' border.

$$L_{TV} = \sum_{i=0}^{W} \sum_{j=0}^{H} \|I_{comp}(i+1, j) - I_{comp}(i, j)\|_1 + \sum_{i=0}^{W} \sum_{j=0}^{H} \|I_{comp}(i, j+1) - I_{comp}(i, j)\|_1 \tag{17}$$

Finally, the generator's total optimization objective is illustrated on Equation 18, where there are 5 $\lambda$s as the hyperparameter coefficients that will be fine tuned. The full proposed GAN architecture can be visualized on Figure 5.

$$L_{Gtotal} = L_{adv} + \lambda_1 L_{recon} + \lambda_2 L_{content} + \lambda_3 L_{style} + \lambda_4 L_{TV} + \lambda_5 L_{parsing} \quad (18)$$

### 3.5. Evaluating face statue inpainting models

Apart from visual observation, several quantitave metrics will also be utilized to reflect the face statue inpainting quality.

**L2 Reconstruction Loss**. One of the generator objective is to minimize the L2 distance between ground-truth images $I_{gt}$ and inpainted images $I_{comp}$. Therefore, the reconstruction loss will be observed during training and testing phase. The L2 distance will be an evaluation metric to justify how well the generator can reconstruct the missing regions as close as possible to the ground-truths.

**FID score**. Image inpainting is an ill-posed problem, meaning, there are multiple outcome variations that don't necessarily need to be identical with the ground-truths. Frechlet Inception Distance (FID) score, as introduced by [11], is able to evaluate the high level structure of GAN generated images without evaluating pixel distances with respect to their ground-truths. Instead, FID picks up feature maps distribution differences by utilizing a pretrained InceptionV3 model. In addition, FID is able to tolerate different styles of the inpaintings while penalizing distortion on the inpainted regions, such as blur and artifacts, thus, this metric is useful to evaluate the inpainted face statues.

**Ablation Study** Ablation study is conducted to compare the contribution of the GAN optimization objectives to the inpainting results quality. Typical execution of ablation study is aligned with the curriculum training procedure. In a curriculum strategy, one loss function is optimized one at a time as the model is progressively finetuned from one objective function to the other. Therefore, the ablation study will inspect the FID scores at each stage of the curriculum training.

## 4. Experiments

### 4.1. Statue Identity Split

The face statue dataset is retrieved from the History Department of the University of Nijmegen. Originally, the dataset is designed for Roman Emperor classification, however, it is ignored in this thesis and the roman emperor statues are merged with the non-emperor ones. There are no available ground-truths on the damaged face statues, meaning, the 'normal - damaged' image pairs don't exist. Therefore, only the normal face statues are utilized for constructing the train and test set in the next preprocessing stage. Manual cherry-picking is involved to separate normal and damaged face statues. And finally, the normal face statues are split into train and test set with 80%:20% proportion.

### 4.2. Dataset Preprocessing

**Face Crop**. Most of the face statues have sizes more than 700 pixels and they are not squares, therefore first, they are cropped and resized based on their face landmarks. A pre-trained MTCNN model is utilized to obtain face landmarks of the face statues. As a result, a set of 5 face landmarks coordinates (left eye, right eye, nose, left mouth corner, and right mouth cirner) is produced for each MTCNN inference. The face crop is determined by performing a standard face alignment based on the landmarks as well as computing two equal paddings to compensate the image width. The final outcome would be 128 x 128 face images. The specific image dimension is chosen to limit the required resources for training the face inpainting models.

**Damage Simulation**. As mentioned in Section 3.1, this thesis is limited to recover only *missing face attributes* type of damage. Therefore, the simulated damages are based on face attributes. The damage simulation is applied on the face crops and for each input image $I_{gt}$, the simulation produces additional two sets of images, the binary masks $M$ and the masked input images $(1-M) \circ I_{gt}$.

Overall, the dataset split amount per damage simulation type is broken down in Table 1. The total amount of the training set is 12000 statue-mask pairs, whereas the test set has 2250 statue-mask pairs.

| Damage simulation | Dataset split | Number |
|---|---|---|
| Eye region | Train | 4000 |
| | Test | 750 |
| Nose region | Train | 4000 |
| | Test | 750 |
| Mouth region | Train | 4000 |
| | Test | 750 |

Table 1. Dataset split

### 4.3. Training a recursive PCA model

A recursive PCA model will act as a baseline comparison to the GAN model. Later, the inpainting outcomes of the two models will be compared quantitatively and qualitatively.

First, eigenfaces $v_{train}$ are obtained by applying Singular Value Decomposition on the 147 training ground-truth images. Second, the mean face $m_{train}$ is computed by averaging the training images. Third, the inpainted face statue is produced by iteratively reconstructing each test

image $x_{test}$ and also specifying threshold $\epsilon = 0.01$. And finally, parameter $K$ is set to 1 after tuning it. The parameter $K$ symbolizes percentile of the used eigenfaces. Note that, the eigenfaces are already sorted descendingly by the singular values.

### 4.4. Training GAN

The GAN training occurs on a single TITAN X 12GB GPU machine located in the CTIT cluster at the University of Twente. There are two GAN frameworks being trained and each of them has a slightly different training approach. Network architecture-wise, both frameworks' generators utilize a U-Net architecture [36] which is renowned in generating segments for biomedical images. Meanwhile the discriminators adopt a PatchGAN architecture [15]. Each GAN training has 250 epochs long.

**Minimax GAN**. In general, the Minimax GAN training consists of discriminator updates and generator updates. During the discriminator updates, the ground-truth images $I_{gt}$ are mapped with label 1, while inpainted images $I_{out}$ are generated by the generator, then the final inpainting outputs $I_{comp}$ are mapped with label 0. Inversely, the generator is updated by mapping the final inpainting outputs $I_{comp}$ to label 1. Due to the binary objectives, Minimax GAN's discriminator needs to output a sigmoid score. In the end, each network update is followed by a backpropagation on respective adversarial loss function. Minimax GAN's generator and discriminator are optimized by an Adam optimizer with default hyperparameter values: learning rate 0.001, $\beta_1$ 0.9, $\beta_2$ 0.999.

**WGAN**. On the contrary, WGAN training is a bit trickier. First, the discriminator outputs a critic loss as an approximation of the EM distance. The EM distance is obtained by feeding the discriminator outputs of real images $I_{gt}$ and inpainted images $I_{comp}$ to Equation 11. Since the discriminator uses a critic loss, the discriminator doesn't deal with cross-entropy between real and fake data anymore. Consequently, the discriminator doesn't output sigmoid score anymore. This is achieved by removing the sigmoid operation at the end of the discriminator forward propagation. Moreover, to satisfy 1-Lipschitz continuity, the discriminator outputs are clamped between -0.01 and 0.01. In the mean time, the generator is trained by feeding the output of real images to Equation 12.

At the beginning of training iteration, WGAN training focuses on converging the discriminator. Therefore for the first 25 epochs, the generator is only updated once every epoch, while the discriminator is updated each iteration. Afterwards, the generator update frequency is set to be equal with the discriminator's. WGAN's generator and discriminator are optimized by RMSProp with learning rate 0.00005 following the original implementation [1].

All of the training iterations employ a curriculum strategy [24] where each optimization objective is tackled once at a time while freezing the others. The training order starts by optimizing reconstruction loss, then followed by content loss, style loss, face parsing loss, and finally regularized by total-variation loss.

In addition, the hyperparameter lambdas are set to the following settings, $\lambda_1 = 1$, $\lambda_2 = 0.01$, $\lambda_3 = 0.1$, $\lambda_4 = 1$, and $\lambda_5 = 0.1$. These values are obtained by doing observations on the inpaintings in multiple trials.

### 4.5. Training a face segmentation model

The segmentation model is trained by following [27]. It is trained on a human face dataset, namely Helen dataset [20] and the dataset is split into a typical 80%:20% split. The network architecture is an encoder-decoder network with a U-Net architecture, following the inpainting generator mentioned in the previous section. The model optimizes a categorical cross-entropy loss between the pixel classes and target segment labels. It is the exact same loss function used as the face parsing loss explained in Section 3.4.

During training, the segmentation model only considers less segment labels, i.e. eye, nose, and mouth segments, since only these three face attributes are necessary to recover in the inpainting task. The segmentation model is trained in 100 epochs and it achieves 80.42% precision and 83.88% recall on test set (excluding the background class). Thus, it is deemed sufficient to generate the ground-truth's segments on the face statues.

### 4.6. Evaluation

The evaluation is focused on examining the proposed models as well as the dataset construction strategies with respect to the three types of statue damages. The aforementioned quantitative metrics, i.e. L2 reconstruction loss and FID scores, will be reported based on the evaluation against the test set for both the GAN and PCA-based models. However, the ablation study is only applicable for the GAN-based models.

## 5. Results

Here, the performance of Minimax GAN, WGAN, and recursive PCA models are displayed. Firstly, both GAN models are trained using the adversarial loss and reconstruction loss. And secondly, they are trained using the adversarial loss as well as the full list of generator objective functions mentioned in Section 3.4.

**Quantitative Evaluation**. As seen in Table 2, the FID scores and the L2 reconstruction loss suggest that the GAN-based models clearly outperform the recursive PCA model. However, overall, the FID scores of the Minimax GAN, WGAN, and recursive PCA do not differ much.

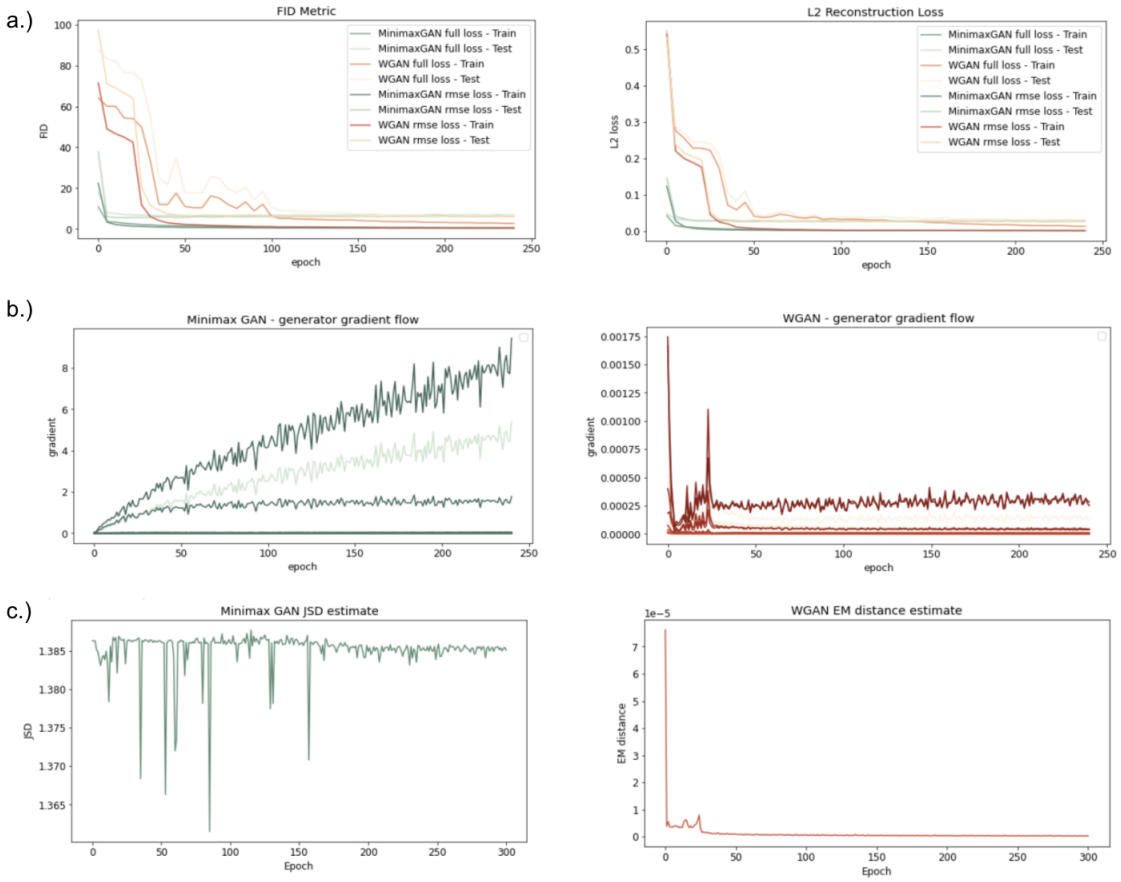


Figure 7. a.) Evaluation metrics on test set: FID and L2 reconstruction loss. b.) Gradient flow on Minimax GAN and WGAN c.) GAN's adversarial function minimizes JSD between real and inpainted image distribution, while WGAN's adversarial function minimizes EM distance between real and inpainted image distribution

| Model | Loss functions | Images optimized on | FID | L2 reconstruction loss | Best epoch |
|---|---|---|---|---|---|
| Minimax GAN | Adv+L2 | $I_{comp}$ | **5.566** | 0.0280 | Epoch 15 |
| Minimax GAN | Adv+L2+Content+Style+TV+FP | $I_{comp}$ | 6.44 | 0.0277 | Epoch 55 |
| Minimax GAN | Adv+L2+Content+Style+TV+FP | $I_{out} + I_{comp}$ | 5.97 | 0.0272 | Epoch 35 |
| WGAN | Adv+L2 | $I_{comp}$ | 6.015 | 0.0276 | Epoch 110 |
| WGAN | Adv+L2+Content+Style+TV+FP | $I_{comp}$ | 6.51 | 0.0320 | Epoch 190 |
| WGAN | Adv+L2+Content+Style+TV+FP | $I_{out} + I_{comp}$ | 6.089 | **0.0271** | Epoch 155 |
| Recursive PCA | - | - | 6.7 | 0.0514 | - |

Table 2. A list of evaluation results. Both the FID scores and L2 reconstruction loss are obtained from the test set. The abbreviations of the loss functions correspond to the followings. Adv: adversarial loss, L2: L2 reconstruction loss, Cont: content loss, Style: style loss, TV: total-variation loss, FP: face parsing loss

When considering only the GAN-based models, Minimax GANs perform the best at generating inpaintings compared to WGAN on either combination of loss functions. In addition, optimizing the loss functions on both $I_{out}$ and $I_{comp}$ helps to improve the FID score as well as the L2 reconstruction loss on models trained on the full loss functions.

When the loss functions are assessed individually, Table 3 suggests that the L2 reconstruction loss has the most impact on the inpaintings fidelity, followed by the style loss.

Next, as observed in Figure 7a, Minimax GANs tend to converge early and then begin to overfit around epoch 30-50. This fact is also supported by Figure 7b, where the

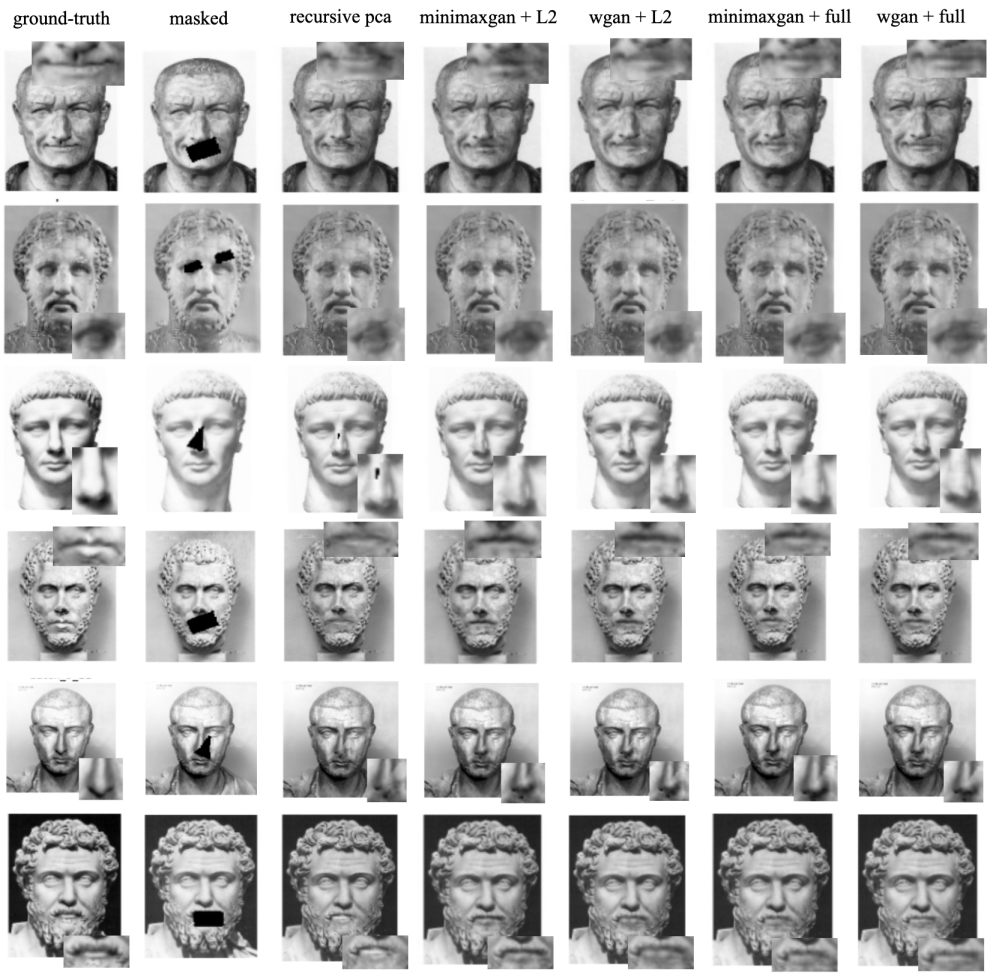


Figure 8. Inpainting results induced by the generative models that have the best test FID score during their entire training. Here, full loss represents all of the generator objective functions elaborated on section 3.

| Loss functions | FID |
|---|---|
| L2 | 6.015 |
| Style | 5.75 |
| Content | 37.56 |
| Total-variation | 51.88 |
| Face parsing | 35.60 |

Table 3. The result of an ablation study where multiple WGAN models are trained using only adversarial loss and each of the listed loss functions. L2 loss and style loss emerge as the most contributive loss functions towards inpaintings' fidelity.

gradient flow of the Minimax GAN is way more fluctuative and has an up-trend. Furthermore, the observed JSD estimate in Figure 7c doesn't tell any meaningful correlation with the inpainting results as it looks pretty stagnant with some minor fluctuations.

The WGAN is observed to have more stable gradients overtime compared to the Minimax GAN. Moreover, the estimated EM distance observed in Figure 7c correlates with the quality of the inpainting results as the epoch increases. Also, the most performing WGANs exist at much later epochs.

On the gradient flow figures, the color shades from dark to light represent the layer position in the generator from top to bottom. Note that, the gradient flows on the displayed figure belong to the models trained using only L2 reconstruction loss, but the other variants also possess similar gradient flow.

**Qualitative Evaluation**. The inpainting results are hardly distinguishable by visual observation. In a glance, most of the inpainted regions look well-restored and they blend well with the rest area of the images. But, when zoomed, the inpainted regions are noticeably blurry. Also, they have some subtle differences in a detailed level.

By looking at Figure 8, the recursive PCA model clearly leaves more artifacts and blurriness on the inpaintings compared to the GAN variants. But, there is no clear

| Iter. | Baseline | Target | FID | Best |
|---|---|---|---|---|
| 1 | L2 | Content | 5.65 | Epoch 1 |
| 1 | L2 | Content+TV | 5.65 | Epoch 40 |
| 1 | L2 | Style | 5.65 | Epoch 1 |
| 1 | L2 | Style+TV | 5.64 | Epoch 45 |
| 1 | L2 | Face parsing | **5.57** | Epoch 1 |
| 2 | Style+TV | Face parsing | 5.72 | Epoch 1 |
| 2 | Style+TV | Content | 5.67 | Epoch 1 |
| 2 | Face parsing | Style+TV | 5.6 | Epoch 10 |
| 2 | Face parsing | Content+TV | 5.72 | Epoch 5 |

Table 4. Curriculum strategy implementation results. The *baseline* represents models trained solely on the loss function while *target* represents the additional loss functions to optimize at the respective curriculum iteration. Note that, on the second iteration, the L2 loss is also included as a baseline.

winner on the GAN variants each of them seem to excel at certain face statues. For example, the Minimax GAN trained with only L2 reconstruction loss inpaints the lips of statue images on the fourth and the sixth row best compared to the others as they have less artifacts and bluriness. However, referring to the same rows, the GAN variants trained on the full loss seem to share more identical attributes of the ground-truths. The reconstructed nose size by the models trained on the full loss are also more identical with the ground truths. Moving on to the next examples, the inpainted images on the first and third row are pretty much similar. Meanwhile, the inpainted eye on images of the second row generated by the Minimax GAN trained with the full loss is slightly more detail. This qualitative evaluation is highly subjective and the judgment solely depends on personal taste.

**Curriculum Strategy**. Even though FID scores of the models trained with curriculum strategy surpass the models trained normally, most of them overfit as the FID scores stop decreasing after less than 10 epochs. Summary on the curriculum strategy implementation is provided in Table 4.

## 6. Discussion

WGAN has relatively more stable gradients accross the training epochs compared to Minimax GAN's as observed on Figure 7c. The WGAN's gradients only have minor fluctuation, meanwhile the Minimax GAN's gradients fluctuate more heavily and the uppermost and bottom-most gradients tend to increase rapidly. Uppermost layer handles the image input, whereas the bottom-most generates the final output. This implies that the generator learns the most from compressing the input images and generating final inpainting results in each respective layer. This observation is also aligned with the claimed behavior of WGAN by Arjovsky et al. regarding how EM distance is a better optimization objective compared to JSD estimate. This has already been discussed in Section 3.3.2. The stable gradients lead to higher quality results for WGAN on later epochs as the best epochs of WGAN in Table 2 are significantly greater than the best epochs of Minimax GAN. Nevertheless, this doesn't really matter as the GAN training can be stopped by considering FID value stagnancy after certain amount of epochs. Concretely, the Minimax GAN's FID score reaches the lowest value at early epochs and starts to show an overfitting behavior overtime. Here, the FID score can be utilized as an indication of early stopping mechanism.

Curriculum strategy doesn't provide value since the baseline loss has produced a good result on this experiment. Hence, adding more training epochs on subsequent curriculum iterations tend to overfit the models. Nonetheless, more challenging inpainting tasks such as larger masked regions may prove the strategy useful.

Visually, inpainting results induced by the recursive PCA model, as a simpler model, are not that far compared to the GAN variants. Specifically, for less complex face attribute i.e. nose, the generated inpaintings may be sufficient. On the other hand, the inpainting results induced by the GAN variants have their own strong points on certain images. However, the reconstructed face attributes of the models trained on the full loss function tend to match the identity of the ground truth images better. This is influenced by the style, content, and face parsing loss. While the total-variation loss helps reducing artifacts.

Although the final inpainting results correspond to the complemented images $I_{comp}$, optimizing the full loss on the direct output images $I_{out}$ additionally helps to improve the FID scores of models trained on the full loss. Spefically, the style, content, and face parsing loss are expected to learn more information from $I_{out}$ rather than $I_{comp}$ as the training progresses.

Lower FID scores don't strictly correspond to better inpaintings. One reason behind this turn of event could be due to the small area of inpainting regions. In the literatures [24, 26], the regions are relatively larger and they cover multiple face attributes. In that case, most likely the GAN variants trained on the full loss will excel more significantly. However, the WGAN framework can be considered best when gradient flow stability is acknowledged. Future study on the correlation between the FID scores, size of inpainting regions, and inpainting results could explore this matter further.

## 7. Conclusion

In this thesis, investigation of a face statue restoration task using image inpainting methods is carried out. Specifically, a deep generative model, GAN, is examined

along with a classical reconstruction algorithm, namely recursive PCA.

By visual observation, the real damaged face statues suffer from three types of damages, i.e. *missing face attributes*, *eroded textures*, and *hollow regions*. However, the damaged statues suffer from *missing face attributes* the most.

The training dataset needs to be in 'normal-damaged' in pairs. In order to overcome the limited amount of normal face statues, binary-mask augmentation is performed such that there are multiple 'normal-damaged' image pairs across three face attributes, i.e eye, nose, and mouth for each statue identity.

Recursive PCA approach aims to recover the damaged regions by iteratively projecting the masked images to the eigenspace obtained from SVD of the training data. The iteration concludes when there is not much difference observed on the restored masked regions.

Minimax GAN and WGAN models are trained to reconstruct the masked images as close as possible to the ground-truths. Based on FID score alone, Minimax GAN trained on L2 reconstruction loss is the most performant model. But, when gradient stability and identity matching are also considered, WGAN with the full loss is the most preferred GAN variant. Specifically, adding a combination of style, content, and total-variational loss helps maintaining the identity of the reconstructed images. Furthermore, when assessed individually, L2 and style loss are the most contributive objective functions.

L2 loss and FID score are feasible to evaluate the inpainting result. But, FID score can't be utilized as a strict measure in deterimining the best quality inpaintings. Instead, it can be utilized as an early stopping mechanism when training the GAN-based models. In the end, visual observation is necessary to judge the final outcomes.

In comparison with recursive PCA as a simpler model, even though it generates acceptable inpaintings for less complex attribute, i.e. nose, overall, recursive PCA model generates inpaintings with the least fidelity. More artifacts and bluriness are found on the inpainting results of the recursive PCA model compared to the GAN variants.

## References


[1] M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein GAN. 1 2017.

[2] C. Ballester, M. Bertalmio, V. Caselles, G. Sapiro, and J. Verdera. Filling-in by joint interpolation of vector fields and gray levels. *IEEE Transactions on Image Processing*, 10(8):1200–1211, 8 2001.

[3] C. Barnes, E. Shechtman, A. Finkelstein, and D. B. Goldman. PatchMatch. page 1. Association for Computing Machinery (ACM), 2009.

[4] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester. Image inpainting. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques - SIGGRAPH '00*, pages 417–424, New York, New York, USA, 2000. ACM Press.

[5] H. C. Burger, C. J. Schuler, and S. Harmeling. Image denoising: Can plain neural networks compete with BM3D? In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 2392–2399, 2012.

[6] C. Dong, C. C. Loy, K. He, and X. Tang. Image Super-Resolution Using Deep Convolutional Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 38(2):295–307, 2 2016.

[7] L. Gatys, A. Ecker, and M. Bethge. A Neural Algorithm of Artistic Style. *Journal of Vision*, 16(12):326, 9 2016.

[8] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative Adversarial Nets. Technical report, 2014.

[9] C. Ham, A. Raj, V. Cartillier, and I. Essa. Variational Image Inpainting. Technical report.

[10] K. He and J. Sun. Statistics of patch offsets for image completion. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 7573 LNCS, pages 16–29, 2012.

[11] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter. GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium. Technical report.

[12] R. Huang, S. Zhang, T. Li, and R. He. Beyond Face Rotation: Global and Local Perception GAN for Photorealistic and Identity Preserving Frontal View Synthesis. In *Proceedings of the IEEE International Conference on Computer Vision*, volume 2017-October, pages 2458–2467. Institute of Electrical and Electronics Engineers Inc., 12 2017.

[13] F. Huszár. How (not) to Train your Generative Model: Scheduled Sampling, Likelihood, Adversary? 11 2015.

[14] S. Iizuka, E. Simo-Serra, and H. Ishikawa. Globally and locally consistent image completion. In *ACM Transactions on Graphics*, volume 36. Association for Computing Machinery, 2017.

[15] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-Image Translation with Conditional Adversarial Networks. *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, 2017-January:5967–5976, 11 2016.

[16] V. Jain and H. S. Seung. Natural Image Denoising with Convolutional Networks. Technical report, 2009.

[17] T. Karras, T. Aila, S. Laine, and J. Lehtinen. Progressive growing of GANs for improved quality, stability, and variation. In *6th International Conference on Learning Representations, ICLR 2018 - Conference Track Proceedings*. International Conference on Learning Representations, ICLR, 10 2018.

[18] T. Karras, S. Laine, and T. Aila. A Style-Based Generator Architecture for Generative Adversarial Networks. pages 4396–4405, 12 2018.

[19] N. Komodakis and G. Tziritas. Image completion using global optimization. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages 442–449, 2006.

[20] V. Le, J. Brandt, Z. Lin, L. Bourdev, and T. S. Huang. Interactive facial feature localization. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 7574 LNCS, pages 679–692. Springer, Berlin, Heidelberg, 2012.

[21] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2323, 1998.

[22] S. W. Lee, H. Jeong, and J. S. Park. Recursive reconstruction of non-facial components using support vector data description. *Pattern Analysis and Applications*, 21(2):337–350, 5 2018.

[23] T. Li, S. Liu, R. Qian, Q. Yan, L. Lin, C. Dong, and W. Zhu. Beautygan: Instance-level facial makeup transfer with deep generative adversarial network. In *MM 2018 - Proceedings of the 2018 ACM Multimedia Conference*, pages 645–653. Association for Computing Machinery, Inc, 10 2018.

[24] X. Li, M. Liu, J. Zhu, W. Zuo, M. Wang, G. Hu, and L. Zhang. Learning Symmetry Consistent Deep CNNs for Face Completion. 12 2018.

[25] Y. Li, S. Liu, J. Yang, and M.-H. Yang. Generative Face Completion. 4 2017.

[26] G. Liu, F. A. Reda, K. J. Shih, T.-C. Wang, A. Tao, and B. Catanzaro. Image Inpainting for Irregular Holes Using Partial Convolutions. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 11215 LNCS:89–105, 4 2018.

[27] J. Long, E. Shelhamer, and T. Darrell. Fully Convolutional Networks for Semantic Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(4):640–651, 11 2014.

[28] A. Mahendran and A. Vedaldi. Understanding deep image representations by inverting them. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 07-12-June-2015, pages 5188–5196. IEEE Computer Society, 10 2015.

[29] X. Mao, Q. Li, H. Xie, R. Y. K. Lau, Z. Wang, and S. P. Smolley. Least Squares Generative Adversarial Networks. *Proceedings of the IEEE International Conference on Computer Vision*, 2017-October:2813–2821, 11 2016.

[30] X.-J. Mao, C. Shen, and Y.-B. Yang. Image Restoration Using Convolutional Auto-encoders with Symmetric Skip Connections. 6 2016.

[31] Z. Mo, J. P. Lewis, and U. Neumann. Face Inpainting with Local Linear Representations. Technical report, 2004.

[32] H. Park and Y. S. Moon. Automatic denoising of 2D color face images using recursive PCA reconstruction. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 4179 LNCS, pages 799–809. Springer Verlag, 2006.

[33] J. S. Park, Y. H. Oh, S. C. Ahn, and S. W. Lee. Glasses removal from facial image using recursive PCA reconstruction. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2688:369–376, 2003.

[34] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros. Context Encoders: Feature Learning by Inpainting. 4 2016.

[35] Y. Pritch, E. Kav-Venaki, and S. Peleg. Shift-map image editing. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 151–158, 2009.

[36] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 9351, pages 234–241. Springer Verlag, 2015.

[37] S. Roth and M. J. Black. Fields of experts. *International Journal of Computer Vision*, 82(2):205–229, 4 2009.

[38] C. J. Schuler, H. C. Burger, S. Harmeling, and B. Scholkopf. A machine learning approach for non-blind image deconvolution. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1067–1074, 2013.

[39] R. Sharma, S. Barratt, S. Ermon, and V. Pande. Improved Training with Curriculum GANs. 7 2018.

[40] Y. Song. Contextual-based Image Inpainting: Infer, Match, and Translate. Technical report.

[41] D. Tschumperlé. Fast anisotropic smoothing of multi-valued images using curvature-preserving PDE's. In *International Journal of Computer Vision*, volume 68, pages 65–82, 6 2006.

[42] Z. M. Wang and J. H. Tao. Reconstruction of partially occluded face by fast recursive PCA. In *Proceedings - CIS Workshops 2007, 2007 International Conference on Computational Intelligence and Security Workshops*, pages 304–307, 2007.

[43] X. Xie, W. S. Zheng, J. Lai, and C. Y. Suen. Restoration of a frontal illuminated face image based on KPCA. In *Proceedings - International Conference on Pattern Recognition*, pages 2150–2153, 2010.

[44] J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu, and T. S. Huang. Generative Image Inpainting with Contextual Attention. 1 2018.

[45] K. Zhang, Z. Zhang, Z. Li, and Y. Qiao. Joint Face Detection and Alignment using Multi-task Cascaded Convolutional Networks. *IEEE Signal Processing Letters*, 23(10):1499–1503, 4 2016.

[46] J. Zhao, M. Mathieu, and Y. LeCun. Energy-based Generative Adversarial Network. *Iclr*, (2014):1–16, 9 2016.