

QATAR UNIVERSITY

COLLEGE OF ENGINEERING

GENERATIVE ADVERSARIAL NETWORKS BASED
RECONSTRUCTION AND RESTORATION OF CULTURAL
HERITAGE

BY

Nesreen Hamadallah Jboor

A Thesis Submitted to
the Collage of Engineering
in Partial Fulfillment of the Requirements for the Degree of
Masters of Science in Computing

June 2019

© 2019 Nesreen. All Rights Reserved.

COMMITTEE PAGE

The members of the Committee approve the Thesis of
Nesreen Hamadallah Jboor defended on 18/04/2019.

Prof. Abdelaziz Bouras
Thesis/Dissertation Supervisor

Dr. Abdulaziz Al-Ali
Thesis Co-Supervisor

Dr. Khaled Shaban
Committee Member

Prof. Omar Boussaid
Committee Member

Approved:

Abdel Magid Hamouda , Dean, College of Engineering

ABSTRACT

Jboor Nesreen, Masters : January : 2020, Masters of Science in Computing

Title: Generative Adversarial Networks Based Reconstruction and Restoration of Cultural Heritage

Supervisors of Thesis: Prof. Abdelaziz Bouras, Dr. Abdulaziz Al-Ali.

Cultural heritage takes an important part in defining the identity and the history of a civilization or a nation. Valuing and preserving this heritage is thus a top priority for governments and heritage institutions. Through this paper, we present an image completion (inpainting) approach adapted for the curation and the completion of damaged artwork. Our approach uses a set of machine learning techniques such as Generative Adversarial Networks which are among the most powerful generative models that can be trained to generate realistic data samples. As we are focusing mostly on visual cultural heritage, the pipeline of our framework has many optimizations such as the use of clustering to optimize the training of the generative part to ensure a better performance across a variety of cultural data categories. The experimental results of our framework were validated on cultural dataset of paintings collected from Wiki-Art and the Rijksmuseum. We used the divide-and-conquer strategy by clustering the training data into different small clusters containing similarly looking images to train smaller Specialized DCGANs. The training has been made on five painting categories containing 2000 paintings each, which took an average of 6.1 training hours. Training the Specialized DCGAN on 1200 paintings from one of the clusters took 3.4 training hours. The inpainting results of the Specialized DCGANs are clearly better in quality than the results of a DCGAN trained on mixture of paintings or on painting category.

DEDICATION

This dissertation is dedicated to my beloved parents Amal and Hamadallah who have continuously encouraged and surrounded me with their endless love, prayers and support to get through the tough times. Also, I would like to dedicate my success to my siblings, Diana, Tamara, Sami, Mohammad and Reem, and thank them for their guidance, advices and support they have provided me during this journey. This accomplishment would not have been possible without them being in my life.

ACKNOWLEDGMENTS

First and foremost, praises and thanks to **Allah** the Almighty, for his showers of blessings throughout this thesis to complete the research successfully.

I would like to acknowledge and express my sincere gratitude to my supervisor and co-supervisor **Prof. Abdelaziz Bouras** and **Dr. Abdulaziz Al-Ali** for their guidance, support, insight and engagement through the learning process of this master thesis. My appreciation is also extended to **Abdelhak Belhi** for introducing me to this interesting topic and for his continued support and advices he has provided. His encouragement and belief in what he do has inspired me.

This thesis is part of the CEPROQHA NPRP project (9-181-1-036) funded by the Qatar National Research Fund (a member of Qatar Foundation), in collaboration with the multimedia team of the Museum of Islamic Art (MIA).

Contents

DEDICATION	iv
ACKNOWLEDGMENTS	v
LIST OF TABLES	viii
LIST OF FIGURES	ix
LIST OF ABBREVIATIONS	x
Chapter 1: introduction	1
1.1. Problem Statement	3
1.2. Objectives.....	4
Chapter 2: literature review	7
2.1 Image Inpainting	7
2.2 Image Inpainting in Conventional Programming.....	8
2.3 Image Inpainting Deep Learning Based Solutions.....	10
2.3.1 Generative Adversarial Networks.....	11
2.3.2 Image Inpainting using DCGANs.....	14
Chapter 3: Data Collection and Pre-Processing.....	20
3.1 Data Collection.....	20
3.2 Data Pre-Processing	24
Chapter 4: Methodology and Implementation	26
4.1 Painting completion framework	26

4.1.1 Motivation	26
4.1.2 Framework Design and Implementation	28
4.2 Approach Evaluation.....	30
Chapter 5: Experimental Results, Validation and Evaluation	37
Chapter 6: Conclusion and Future work	50
References.....	53
Appendix A : COMPLETION EVALUATION SURVEY.....	56

LIST OF TABLES

Table 1: Training and Inpainting Hyperparameters.....	38
Table 2: General DCGAN Training and Inpainting Hyperparameters.....	38
Table 3: Categorized DCGAN Inpainting Results on Five Paintings Categories	40
Table 4: Specialized DCGAN Vs General DCGAN Inpainting Results	45
Table 5: Training time for each DCGAN type.	46
Table 6: AutoEncoder Vs Specialized DCGAN Inpainting Results.....	48

LIST OF FIGURES

Figure 1: Generative Adversarial Networks Architecture.	12
Figure 2: Selected Artworks from Wiki-Art.....	21
Figure 3: Selected Artworks from the MET	22
Figure 4: Selected Artworks from the Rijksmuseum.....	24
Figure 5: The Training Step of the Cultural Inpainting Framework.....	27
Figure 6: The Completion Step of the Cultural Inpainting Framework.....	28
Figure 7: Our Framework DCGAN Architecture	29
Figure 8: AutoEncoder Architecture.....	34
Figure 9: Z Vector Dimension Tuning for GoogleImages DCGAN Inpainting	35
Figure 10: Learning Rate Tuning for Specialized DCGAN Inpainting.....	36
Figure 11: General Image Inpainting DCGAN Trained on Mixture of Images	39

LIST OF ABBREVIATIONS

GANs	Generative Adversarial Networks
DCGANs	Deep Convolutional Generative Adversarial Networks
CNNs	Convolutional Neural Networks
MSE	Mean Squared Error
EBI	Exemplar-based Inpainting
VRAM	Video Random Access Memory
GPU	Graphical Processing Unit
PGGAN	Patch Global GAN
ResNet	Residual Network
MIA	Museum of Islamic Art
MET	The Metropolitan Museum
CEPROQHA	Cost-Effective High-Quality Preservation and Restoration of Qatar Cultural Heritage
ReLU	Rectified Linear Unit
CSV	comma-separated values
OAI-PMH	Open Archives Initiative Protocol for Metadata Harvesting
API	Application Programming Interface
XML	eXtensible Markup Language
SSD	Sum Square Difference
BOVW	Bag of Visual Words
SIFT	Scale Invariant Feature Transform
SURF	Speeded Up Robust Features

VGG	Visual Geometry Group
DoG	Difference of Gaussian
NLP	Natural Language Processing

CHAPTER 1: INTRODUCTION

Cultural Heritage represents the identity of societies as it is a bridge that transfers the history from previous generations to the current and future generations. It strengthens the ties among nations and civilizations and enables humans to learn from the past as it is the most reliable medium of history transfer. Cultural assets or artifacts are very precious, valuable and important because they cannot be recreated or replicated easily. These items are very fragile and prone to physical degradation due to multiple reasons such as their degradation over time or due to environmental impacts such as earthquakes, hurricanes, air pollution, temperature, humidity, etc. [1-3]. The preservation, restoration, and reconstruction of cultural assets is the work of curators and skilled art conservators in art institutions and museums. Their work is performed manually and consists of preserving valuable assets in order to maintain their physical state or to restore damaged assets to a better state. Those professionals reconstruct and restore the assets using special treatments and techniques to minimize and stop any further damage. This overall process is time-consuming and risky as these assets are very fragile and require very careful handling. In addition, the risk is extremely high when dealing with those valuable assets since the restoration process could cause more damage and loss of value. Not to mention the high cost of restoration process which adds a financial burden on art galleries and museums when bringing professional curators to perform this task.

The sustainable preservation of cultural heritage plays nowadays a primordial and important role in maintaining this legacy for future generations. However, physical preservation alone is insufficient. These assets usually suffer from information loss which makes them undervalued compared to fully annotated assets. Thanks to the advances in computer science and information technology, modern

digital technologies are used broadly because they are very reliable, cheaper and sustainable for information capture, sharing and archival. Digital preservation of cultural heritage is thus a top priority for governments and heritage institutions because it solves many problems related to physical damage and information loss of different cultural assets [4-6].

Data science computer-based tools have proved to be very successful in analyzing and extracting useful knowledge from data to improve decision making. Those tools are used in different applications to find powerful solutions to problems in different areas such as healthcare [7], fraud detection [8], warning of natural disasters [9], etc. Data science was successful in different fields, but not really used in cultural heritage. Thanks to its success, many research teams around the world are currently trying to apply data science technologies to enrich and add more value to cultural assets. The current approaches are mostly related to data classification and semantic annotation. However, very little work seems to target the completion of degraded and damaged cultural assets. The completion process is to curate the digital copy of the artwork by training a deep learning model to complete the damaged or missing areas. The term image inpainting that is used frequently in this thesis, describes the process of visually completing an image that has some missing regions. The most important contribution in deep learning is without a doubt Generative Adversarial networks (GANs) which are nowadays among the best performing generative models for multiple tasks related to computer vision such as Super-resolution Images [10], Unsupervised Image Generation [11], Speech Enhancement [12], Image to Image Transilation [13], Face Aging [14], Text to Photo Synthesis [15], CT Generation from MR Image [16] etc. GANs are also used for unsupervised image completion tasks where their performance is considered as state of the art [17].

However, due to the intricacies in addition to the diversity of cultural artwork, it is clear that trying to visually complete any incomplete asset is a tough challenge even for long term human experts. Solving this challenge using computer-based tools is even harder.

In this dissertation, we mainly focus on approaches based on generative adversarial networks which are known for their very good performance for this type of challenge. Most of the approaches focus on completing images from specific visual categories, such as completing faces, building facades, etc. [17]. However, cultural heritage assets commonly span multiple categories which make the existing solutions not viable. Through our analysis and experiments, we found that it is rather inefficient to design an inpainting approach based on a single generative model to address several contexts.

We propose a new image inpainting framework inspired by the semantic image inpainting approach proposed in [17]. This framework is based on Deep Convolutional Generative Adversarial Networks (DCGANs) for inpainting visual cultural data using a divide-and-conquer strategy based on clustering. The principle consists of clustering similarly looking cultural images and then training a generative model for each category. When presented with an incomplete image, the system identifies the category of that image and use the associated GAN for the visual completion process and produce plausible completed images. My research work is part of the CEPROQHA NPRP project (9-181-1-036) funded by the Qatar National Research Fund (a member of Qatar Foundation), in collaboration with the multimedia team of the Museum of Islamic Art (MIA).

1.1. Problem Statement

Both MIA and other cultural heritage local partners (Sheikh Faisal Museum

and Qatar National Library), expressed the need for completing damaged art using digital tools to provide art curators with inspiration on what content should be filled in the missing areas and to effectively help them in their work.

In this thesis, we introduce a new semantic image inpainting framework based on Deep Generative Adversarial Networks (DCGANs) to solve the problem of reconstructing and restoring damaged and missing regions in cultural artwork. This framework is inspired by the divide-and-conquer strategy and it is based on clustering. It clusters similarly looking cultural artworks and then trains separate DCGAN for each cluster. The idea is that we train multiple Specialized DCGANs on similarly looking images instead of training one General DCGAN on a variety of images as it is a common approach in many types of research in the field of image inpainting. Each trained DCGAN will be used to realistically complete a sample of artwork that has missing or damaged areas.

The success of the developed framework can be measured by answering the following basic research questions of this thesis:

- Can our image inpainting framework digitally repair the damaged and missing areas in cultural artwork?
- Does training a DCGAN on similarly looking data generates more plausible and realistic output than training it on a mixture of data?
- Does the completion process using our image inpainting framework produce realistic completed artwork?
- Can the framework efficiently identify the category of the incomplete image and choose the correct associated DCGAN for the visual completion process?

1.2. Objectives

The aim of this thesis study is to achieve reliable, efficient and sustainable

completion of missing and damaged regions of cultural data by designing, building and evaluating a deep learning generative based framework. This framework will help curators in art institutions, galleries and museums to reconstruct damaged artwork with less time and effort. Moreover, it will reduce the risk and cost of performing the restoration process manually on the valuable assets.

The aim of this research can be achieved through the following objectives:

- Propose a solution for reconstructing damaged cultural heritage assets.
- Build the framework based on an existing DCGAN image inpainting implementation and adjust it to support our case study.
- Improve the data that is fed to the DCGANs in order to improve its performance in terms of the training time and the realism of the completed image.
- Train DCGANs on cultural artwork and use the trained DCGAN for completing artwork that has missing or damaged regions.
- Experiment the image inpainting behavior when training DCGANs on artwork categories and on similarly looking artwork.
- Cluster artwork to separate similarly looking data using pre-trained deep learning models and use its features extraction part.
- Extract the features of the incomplete artwork to be redirected to the associated trained DCGAN for image completion.
- Present experiment results and findings in detail.
- Provide recommendations for future research work.

This document is organized as follows. Chapter 1 is an introduction reflecting general background information about the topic, problem statement, and objectives of

this thesis. Chapter 2 contains a literature review about the state-of-the-art in the field of image inpainting and focusing on their strengths and weaknesses. In Chapter 3, we explore the different cultural datasets we have obtained to be used for training and image inpainting tasks. Also, this chapter illustrates the data pre-processing step. To discuss the methodology and implementation in details, Chapter 4 demonstrates in depth the approaches, methodology, algorithm, and implementation of our framework. Chapter 5 evaluates and validates the proposed Cultural Heritage inpainting framework by discussing the experimental results and findings. The conclusion and future work are discussed in Chapter 6 to present the important research findings, limitations and possible future work.

CHAPTER 2: LITERATURE REVIEW

This chapter provides background information by looking at research studies that focus on solving the problem of image inpainting. Also, it includes some discussion about the developed image inpainting applications with different advanced architectures and techniques.

2.1 Image Inpainting

In computer vision, the digital process of filling and reconstructing damaged and missing regions in images is known as image inpainting or image interpolation, as often referred to in the literature [18, 19]. This process tries to replicate the real basic techniques used by professional restorers when manually restoring valuable cultural assets to their consistent state in order to maintain its quality and value. Image inpainting is an active topic in computer vision research that is used in numerous applications like image or scene restoration or object removal etc. Various image completion algorithms have been proposed that use different approaches to reconstruct the missing areas in images with information that is semantically valid and properly textured [19-22]. Efficient image inpainting techniques should generate images that cannot be identified by the human eye as distorted samples and appear as realistic as possible. Thanks to the recent progress of machine learning and with data sources becoming available for researchers, tackling such a challenge was never this possible. In fact, many research efforts are dedicated to techniques related to data completion and more specifically the ones used to complete visual data. Multiple machine learning based techniques were used to address the image inpainting challenge, but with the rise of deep learning, these approaches saw a big leap forward mostly due to the superior performance observed on image reconstruction tasks using autoencoders and restricted Boltzmann machines. Deep learning techniques are

increasingly adopted in many image inpainting and editing tasks and successfully proved their ability in realistic content generation compared to traditional techniques.

Image inpainting can be performed in two ways, either by using an external source of data like other images that have similar context or using the available uncorrupted data in the input image to help in reconstructing it.

2.2 Image Inpainting in Conventional Programming

Hays et al. [23] propose an approach for scene completion that samples the best matching patch by leveraging a large visual database to fill the incomplete image. The authors are using millions of images with a variety of scenes to perform the image completion. Looking for the perfect patch among millions of images is considered a time-consuming process. Therefore, to speed up the search process, semantically similar scenes that have very small distance are grouped together. The grouping is done by computing the gist descriptor for all images in the database and for the source image excluding the whole region. Then, they compute the Sum Square Difference (SSD) that calculates the difference in the shape of the source image gist descriptor and every gist descriptor in the database. In addition, the color difference between the source and the database images is computed in the $\mathcal{L} \times a \times b$ color space. After getting the SSD, the top 200 best matching scenes that has the minimum weighted SSD error are selected to extract the similar patches. The local context of the image is characterized by making the nearby context to be all pixels inside 80-pixel radius of the hole boundaries. Beside computing the SSD, a texture similarity score is also important to measure the compatibility of the filled content to the source image within the local context. The selected regions to fill the holes are composite at its best matching scene using a graph cut seam finding approach and Poisson blending. The seam finding operation removes pixels from the source image which

are undesirable while having the remaining pixels not changed. The seam finding operation is restricted to remove only small number of valid pixels surrounding the hole by applying a small cost for removing each pixel that increases with distance from the hole. The researchers chose to minimize the gradient of the image difference along with the seam to make the seam pass through regions of the image which either match or are both constant colors. Afterwards, the Poisson blending is applied on the entire

image to hide the color difference at low frequencies. Finally, each filled region is assigned a score which is the sum of the scene matching distance, the local context matching distance, the local texture similarity distance and the cost of the graph cut. Those four scores contribute roughly and equally. The user is given 20 composites with the lowest scores. In order for this algorithm to succeed and produce plausible images, it requires a large amount of data. The chance of finding the best matching patch from the input image increases as the database grows. However, gathering a large number of images will not ever be enough to cover all types of images in the world. The available number of similar image sets gathered for image completion tasks will not cope with the huge number of images produced over the time that varies in color, structure, and texture. This will limit the performance of the algorithm in finding the

best matching scenes which will lead the algorithm to complete images with content that is not perfectly similar to the source image and fill it with incompatible texture. Also, the fact that this technique relies on a large database is a major drawback.

Another image inpainting algorithm called Exemplar-based inpainting, known as EBI uses iterative solution to generate the missing region based on the available data in the source image [24]. The process of filling the missing regions starts at the

edges between the corrupted and uncorrupted region and gradually moves inwards to complete the missing area. The filling rule is to extend the isophotes, or linear structures while matching gradient vectors at the neighboring edge of the fill-region. EBI consists of Laplacian-based edge detection, followed by iterations of two major filling steps: determining pixel filling priority and calculating the weighted pixel value. The Confidence Term is used to prioritize the filling of pixels locating closest to the source region (known pixels). It evaluates each edge pixel with its surrounding pixels and gives a ratio of pixel location in the fill versus source region. For example, if the pixel is located at fill-front and has 2 out of 9 surrounding pixels located within the source region, then it will obtain a lower fill priority than the pixel with 5 out of 9 surrounding pixels located within the source region. An advanced version of this algorithm by adding a similarity term based on Non-Local-Mean method, which measures how similar the current pixel patch is to the rest of the regions within the image. To evaluate the EBI algorithm, the priority term is defined with both the confidence term and the similarity term. In each iteration, the pixel with the highest priority enters the filling stage and has its value assigned by a normalized weighted sum of its surrounding source region pixels. The pixel weighted estimation with L2-norm gives emphasis to pixels closed to the inpainting pixel and the boundary. With every pixel update, the fill-front pixel priority is re-evaluated and the new pixel with the highest priority proceeds to the filling stage. The algorithm iterates until the entire fill region is complete. The drawback of this algorithm is when the missing regions get larger, the filled content tends to get blurry because it uses diffusion process to fill the image.

2.3 Image Inpainting Deep Learning Based Solutions

Deep learning [25] algorithms are very promising solutions in research since they are used in automatic feature extraction for complex datasets such as images, at a high level of abstraction. Those algorithms are developed in a hierarchical architecture containing very deep layers that can deal with a large amount of data in unsupervised settings. We are encouraged to use DCGANs as a deep learning solution in our proposed framework because through the use of deep learning techniques, our framework can learn the representation of the large cultural dataset that we will use and be able to extract global features and detected patterns without any human interference.

2.3.1 Generative Adversarial Networks

Generative Adversarial Networks (GANs) is a new class of unsupervised machine learning models. GANs have proved that it can achieve far better performance in different image applications compared to traditional networks. The concept of GANs was introduced in research by Ian Goodfellow in 2014 [26] comprised of two networks, pitting one against the other (thus the “adversarial”). The two networks in this architecture are, namely, the Generator and the Discriminator. The Generator network can be described as a counterfeiter that generates fake data that looks as realistic as possible and the Discriminator network acts as the police that is trying to detect if the provided data from the Generator is real or fake. As demonstrated in Figure 1 below, the first network is the Generator which takes as an input a latent vector (also called noise vector \mathcal{Z}) initialized with random values. The main purpose for the Generator is to generate samples that looks like the hidden distribution of the training dataset without seeing any samples or creating copies from it. The second network is the Discriminator, it takes as an input a mixture of data from the real dataset as well as generated data from the Generator. The Discriminator

distinguish if the data produced by the Generator is real or fake and output a probability of the generated image being real or fake. This value represents the loss of the GAN which is back propagated to the Generator to update the noise vector and generate improved samples. Concurrently, the same loss value is back propagated to the Discriminator to improve its performance and capturing the weakness points of the Generator.

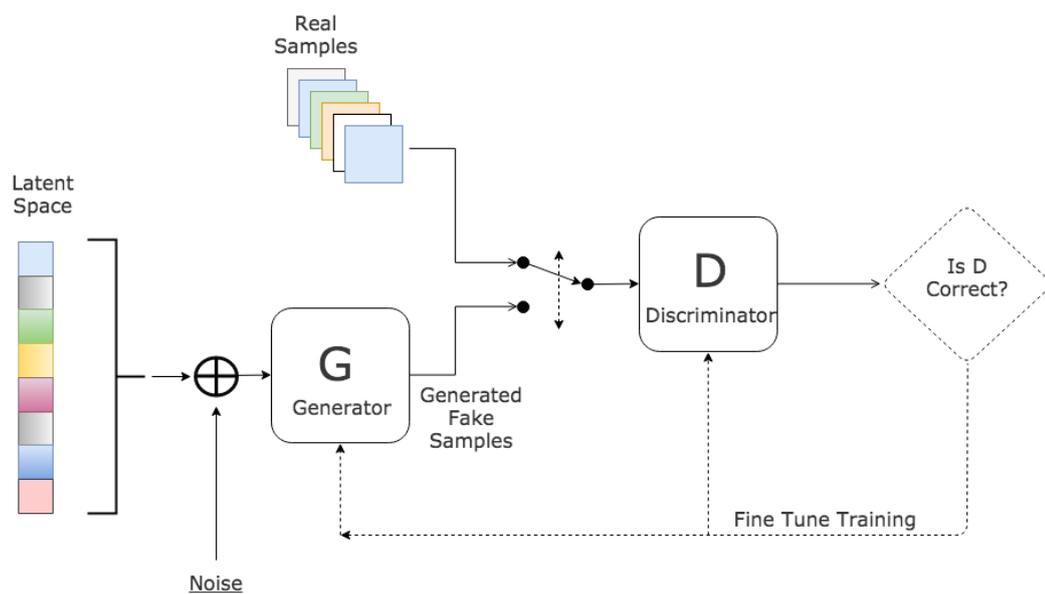


Figure 1: Generative Adversarial Networks Architecture¹.

Both networks are trained simultaneously in an adversarial setting. The Discriminator is trained to maximize the probability of assigning the correct label to both real images from the training dataset and generated images from the Generator. Simultaneously, the Generator is trained to minimize the loss value to challenge the Discriminator into accepting the generated samples as real. Eventually, we hope that

¹ Generative Adversarial Networks Architecture : <https://www.kdnuggets.com/2017/01/generative-adversarial-networks-hot-topic-machine-learning.html>

this network (GAN) would reach a *Nash Equilibrium State*, meaning that the Generator captured the hidden distribution of the training data and can generate samples that look realistic and the Discriminator is smart enough to distinguish between generated or real samples. According to *equation* (1) from [4], the Discriminator and the Generator play minimax two-player game with value function $V(\mathcal{D}, \mathcal{G})$ where $P_{data}(\mathcal{X})$ denotes the true data distribution and $P_Z(\mathcal{Z})$ denotes the noise distribution. Training the DCGAN consists in optimizing the following loss equation using the backpropagation algorithm:

$$\min_{\mathcal{G}} \max_{\mathcal{D}} V(\mathcal{D}, \mathcal{G}) = E_{\mathcal{X} \in P_{data}(\mathcal{X})} [\log(\mathcal{D}(\mathcal{X}))] + E_{\mathcal{Z} \in P_Z(\mathcal{Z})} [\log(1 - \mathcal{D}(\mathcal{G}(\mathcal{Z})))] \quad (1)$$

Deep Convolutional Generative Adversarial Networks (DCGANs)

architecture proposed by [27] has set some constraints on the architecture of the convolutional networks for the Discriminator and the Generator. The constraints of the architecture include:

1. Replacing all pooling layers with strided convolutions and fractional-strided convolutions.
2. Using batch normalization layers to make the training more stable.
3. Discarding fully connected hidden layers.
4. In the Generator, using \tanh as the activation function and using *ReLU* activation for the remaining layers.
5. In the Discriminator, using *LeakyReLU* activation for all layers [28].

Adding convolution architecture made the training phase more stable for the generative adversarial networks and improved the quality of the generated data. GANs in general suffer from model collapse in the training process. Model collapse refers to a situation where the Generator generates only a few similar data samples

that successfully fools the discriminator instead of generating a diversity of samples by learning the underlying distribution of data. Furthermore, GANs do not have any clear stopping criteria and no explicit evaluation metric. The main evaluation criterion is the perceptual quality of the completion. GANs are not suited for image completion as their output has high chances to be unrelated to what we want to complete. In the following section, we present methodologies used to constraint the GANs output.

2.3.2 Image Inpainting using DCGANs

Even though GANs are mainly developed to generate data, they can be used for semantic image inpainting tasks. The visual completion using GANs consists of constraining the output of the generator in order to generate an image that has the same visual characteristics as in the damaged one. The damaged area will then be replaced with the associated area from the generated image. This section review approaches that implement GANs based solution for image inpainting. The reviewed literature discusses different architectures of image inpainting techniques.

- Context Encoders

Authors in [29] introduce Context Encoders as the first parametric image inpainting algorithm producing realistic results for semantic hole filling for large missing regions. This CNN network is trained in unsupervised settings. The architecture consists of Encoder and Decoder pipeline. The Encoder is initialized with random weights. It takes as an input a masked image and try to extract its context and compact it into a latent feature representation. A channel wise fully-connected layer is placed between the Encoder and Decoder to pass the latent feature representation from the Encoder to the Decoder. The Decoder generates the missing content of the image relying on the feature representation received from the Encoder. The network is constrained by joint loss function contains reconstruction loss (L_{rec}) and generative

adversarial loss (L_{adv}). The reconstruction loss is L2 distance. Using only L_{rec} tends to make the generated content in the output image blurry. They alleviate this problem by adding L_{adv} that is based on GANs. L_{adv} forces the network to generate realistic and acceptable images. It conditions only the generator on the context of the input and unconditions it on using the noise vector. The overall joint loss is defined as $L=L_{rec}+L_{adv}$. Using the joint loss improves the inpainted images significantly. The Context Encoder still needs more improvements because the model is over-fitting and that tends to make the output images unrealistic.

- Semantic Image Inpainting with Deep Generative Models

Semantic image inpainting introduced by Yeh et al. [17], is a novel method for generating the missing parts of an image by conditioning it to the available data in the input image. The DCGAN is implemented with a Generator and a Discriminator which are trained on complete images. After the training is completed, the Generator now has the ability to generate images mimicking the training data distribution P_{data} by taking the random vector Z from prior distribution P_z . The Z vector is iteratively updated through back-propagation to find the closest encoding \hat{Z} of the corrupted image in the latent space. The closest encoding \hat{Z} is fed to the trained DCGAN to generate the missing regions by being constrained to the manifold. The surrounding pixels of the missing regions are given higher importance weight than the far pixels. This technique defines the closest encoding \hat{Z} by combining a contextual loss in addition to the perceptual loss (evaluated by the discriminator). This loss combination is used to perform a backpropagation on the input of the generator (Z vector). The goal of the backpropagation optimization is to lower as much as possible this combined loss. As the Z vector is the only parameter controlling the output after training the DCGAN, the goal is to generate an output (image) that minimizes the loss

combination which will theoretically result in an image that looks similar to the one that we want to complete. The context loss defines the weighted L1-norm difference between the completed image and the original image. On the other hand, the prior loss forces the generator to generate images having similar features to the training dataset by having some penalties based on high-level image features instead of pixel-wise difference. The proposed approach shows significantly realistic completed images when the completion is conditioned by the combined loss.

- Globally and Locally Consistent Image Completion

Iizuka et al. [11] introduce an image inpainting approach with an advanced architecture that ensures global and local consistencies of the filled image using CNNs. The architecture consists of a Generator and a Context Discriminators comprised of Global and Local Discriminator. The Generator network is a fully convolutional network that has two types of layers: Convolutional and Dilated layers. Convolutional layers conserve the spatial structure of the input and the Dilated layers allow to compute each output pixel of the generated region with a much larger input area using the same parameters and computational power to improve the realism of the generated content. Therefore, the network can see a larger area of input at a low resolution when computing each output pixel (generated pixel) than with standard convolutional layers. For the Discriminator, the Global Discriminator evaluates the completed input image (256×256 pixels) as a whole and the Local discriminator evaluates the completed masked region individually (128×128 pixels). The output of both Discriminators is concatenated into a single fully connected layer which predicts a value corresponds to the probability of the image is real or fake. Sometimes the generated region has subtle color inconsistencies with the surrounding regions. To overcome this issue, simple post-processing is performed by blending the completed

region with the color of the surrounding pixels. Those steps are done by applying a fast method followed by Poisson image blending. The loss functions used in the Generator network are Mean Squared Error (MSE) for training stability and GANs loss (adversarial loss) to improve the realism of the results. The mixture of those two losses has been used in previous problems like in the image to image translation and image completion to stable the training of high-performance network mode. The training process was executed first to the Generator network using MSE loss. After the Generator is trained for T_g iterations, the network is fixed and the Context Discriminators starts training for T_d iterations. Finally, both the Generator and the Context Discriminators are trained jointly until the end of the training. The authors have proved the importance of the Context Discriminators and experimented the quality of the results by removing either of them. It was observed, removing either of them will result in blurry and unreal results. Therefore, using both Context Discriminators, the model can achieve more realistic completion that has local and global consistencies. The researchers in [11] have faced some limitations when the corrupted region is larger than the surrounding area, it cannot be filled due to the special support of the model. The model can be changed to have more dilated convolutions to push the limit. The limitation refers strictly to square masks, wide hole can be completed as long as they are not tall because the information above and below the hole will be needed to complete the image. In addition, when the object is heavily structured e.g., the object is partially masked, the model fails to generate a realistic image. Not to mention the entire training procedure of the model took 2 months to be completed which is a very long and difficult to monitor as if the discriminator fails to detect a generated sample, the training process will be stuck. This major issue is what we are trying to highlight and improve in this research.

- Patch-Based Image Inpainting with Generative Adversarial Networks

Recently, a study was conducted discussing the completion problem of high-resolution images by paying attention to local details along with the global structure of completed images using PGGAN [30]. PGGAN is a network that consists of a Generative Residual Network and two Discriminators, Global and PatchGAN discriminator. The Generative ResNet layers are down-sampling, residual blocks and up-sampling. In this research, they are also using Dilated Convolutional layers in the ResNet to increase the receptive field size and spread out the convolution weights over a wider area to expand the receptive field size without increasing the number of parameters. To avoid training separate networks like in [11], they have designed a weight sharing architecture (Shared Layers) to learn common low-level visual values. The path after the shared layers is split into two paths, namely, the PatchGAN path and the Global Discriminator path. Global Discriminator evaluates the global structure of the completed image while the PatchGAN Discriminator evaluates the local completed patches in the masked input and explores every possible local region as well as dependencies among them to exploit local information to the fullest. PatchGAN shows improved quality of the generated images compared to Local GAN used in [11] that forces the network to produce independent textures that do not blend well with the whole image semantics. At the end of PatchGAN path, a fully connected layer is added to reveal full dependency across the local patches. A combination of three loss functions is used in the training stage which is: Reconstruction loss, Adversarial loss, and Joint loss. The Reconstruction loss (L_{rec}) finds the L1 difference between the completed image and the ground truth. The Adversarial loss is computed by both paths of PGGAN discriminator networks. Finally, the Joint loss is

calculated by summing Global GAN Discriminator loss (L_g), PatchGAN Discriminator loss (L_p) and Reconstruction loss (L_{rec}). The generator parameters are updated by the Joint loss to improve the quality of the generated images. The GGAN Layers, PatchGAN Layers, and Shared Layers are updated respectively by L_g , L_p and $L_g + L_p$.

The DCGAN based semantic image inpainting approach using a generative model in [17] will be the base of our framework for performing image inpainting on cultural heritage assets. This approach will be explained further in details in chapter 4.

CHAPTER 3: DATA COLLECTION AND PRE-PROCESSING

This section states detailed information about the cultural datasets we used to validate our approach. Furthermore, this section demonstrates how the cultural dataset is extracted, collected, pre-processed and prepared for DCGAN training and inpainting. We are leveraging clustering to cluster the cultural dataset into similarly looking images in order to train specialized inpainting DCGANs that will save us time and produce plausible completed images.

3.1 Data Collection

The dataset used for training and inpainting is an important factor of our cultural inpainting framework. The datasets we have used to validate our approach on cultural data are Wiki-Art Dataset, the Metropolitan Museum (MET) Dataset and the Rijksmuseum Dataset.

Wiki-Art is a visual art encyclopedia containing a variety of visual artworks for different artists including the well-known ones such as Vincent van Gogh, Pablo Picasso, and Salvador Dali. The artworks are gathered from all around the world and from across a very wide time span. This encyclopedia gathers a huge number of historical artworks available on the planet and it is considered to be an important source of digital historical artworks that is accessed and used by the public. It is worth noting that we mostly used paintings from Wiki-Art dataset for evaluating and validating our framework as paintings were the biggest cultural type regarding data samples. The Wiki-Art dataset we have used in our research was obtained by Belhi et al. [31]. It is a collection of more than 140,000 data samples. Wiki-Art website does not provide any public APIs to fetch the desired visual artworks which makes it hard for interested individuals to obtain a group of digital artworks easily. Therefore, Belhi et al. [31] have designed custom harvesting scripts based on the python library

beautiful soup to crawl important data from the Wiki-Art website which mainly consists of the visual artworks with their metadata. The metadata obtained contains the art style, year, artist, media, category and other important metadata that is available on the website. Figure 2 shows some selected artworks from Wiki-Art dataset from different artworks categories. The information fetched from the website by [31] is maintained in a MySQL database for fast retrieval of information and for any future usage.

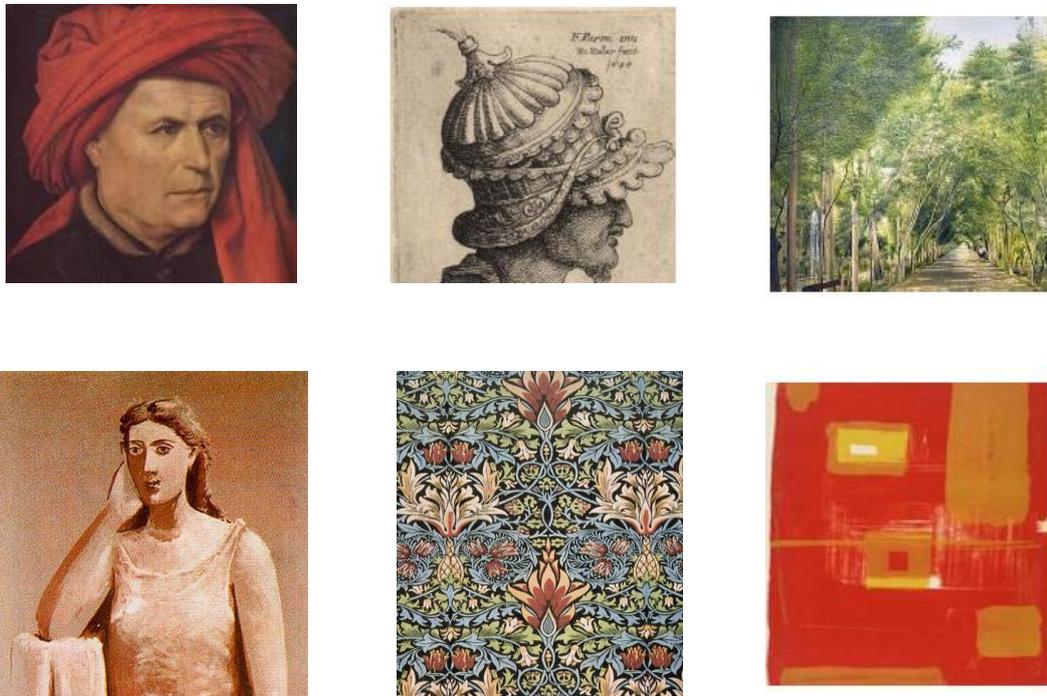


Figure 2: Selected Artworks from Wiki-Art

The dataset² set of The Metropolitan Museum (MET) [32] of New York is collected from the MET museum which is one of the largest in the US. It is also considered as the most visited art museum in the world. It exhibits more than 5,000

² The Metropolitan Museum (MET) <https://www.metmuseum.org/art/collection>

years of art from all around the world. The art data available on the website has more than 400,000 high-resolution art samples that are published under the Creative Commons open access license [33]. The images in this institution are of two categories: images with free unrestricted use for public and images under restrictions and copyright that require additional fees to get the digital image captured by the MET staff. Same as in the Wiki-Art dataset, the MET dataset is mostly a collection of visual art including basic metadata like title, date, artist, dimensions, and medium. However, the majority of the presented artworks on the website have metadata that is not fully available. The data collection is provided by the MET website in a CSV file. However, to be able to harvest digital images from the museum's website, some custom-made Python scripts are required to do the job. Figure 3 shows some special selected samples from the MET dataset.

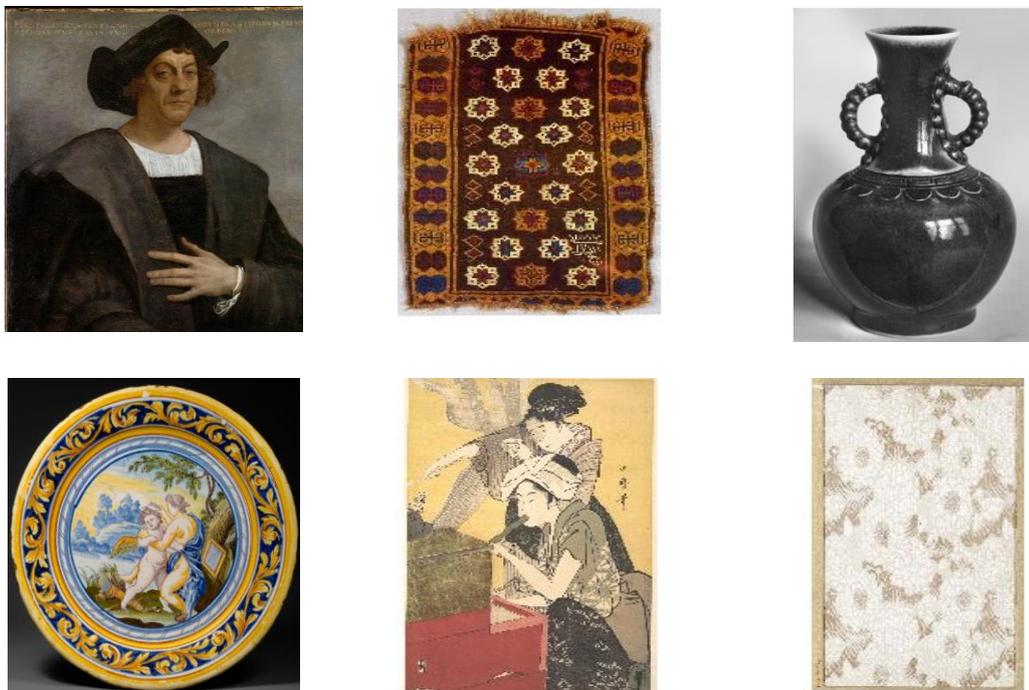


Figure 3: Selected Artworks from the MET

The Rijksmuseum of Amsterdam often referred to as the Rembrandt museum, have recently published a cultural ³dataset [34]. The museum has a very wide variety of artifacts and artworks representing more than 800 years of Dutch and global cultural heritage of the golden age. The data collections contain more than 100,000 assets accessible by the public via an API developed by the museum featuring the OAI-PMH⁴ protocol (Open Archives Initiative Protocol for Metadata Harvesting). The API web service offers the opportunity to the audience to get the data easily and contribute to sharing it to a larger audience. We have noticed that most of the data in the dataset are more related to the Dutch history. Also, the API provides the data in an XML file containing inconsistent tags structure because some images have missing metadata. The Rijksmuseum dataset consists mainly of paintings, pottery, glass art, etc. and each image is associated with its metadata. The collection on the website is constantly changing. It is being updated with new acquisitions of historical assets. Figure 4 illustrates the collected assets from the dataset.

³ The Rijks Museum website <https://www.rijksmuseum.nl/>

⁴ The Rijks Museum API <https://www.rijksmuseum.nl/en/api>

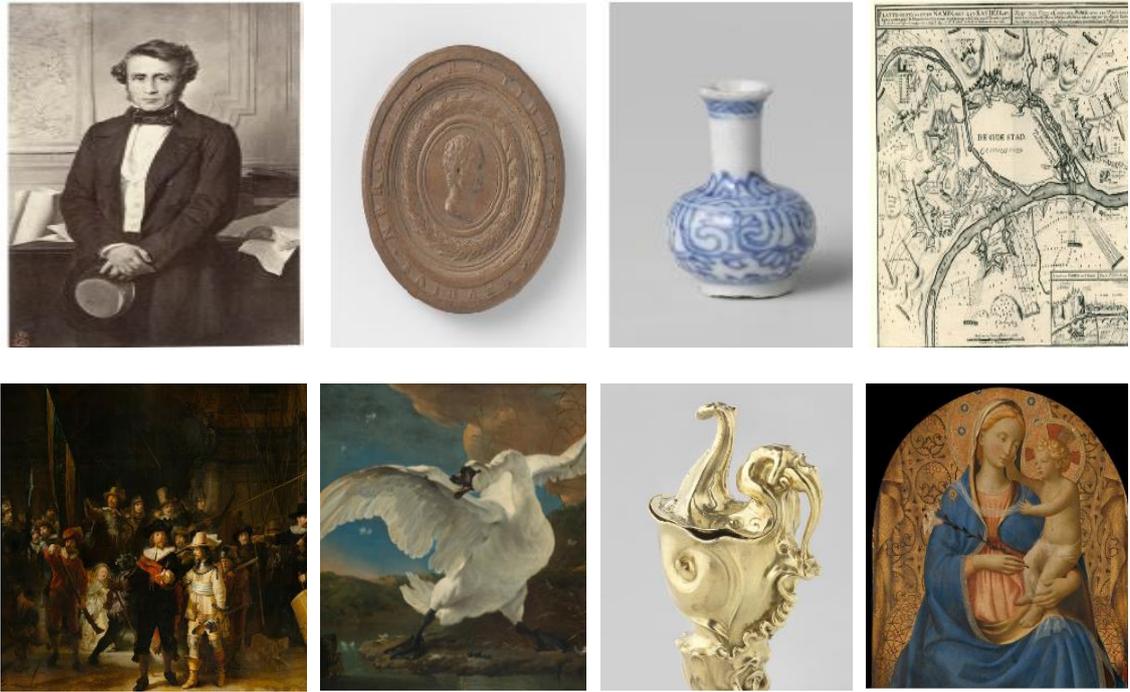


Figure 4: Selected Artworks from the Rijksmuseum

3.2 Data Pre-Processing

After analyzing the collected datasets, we have found that the samples collected from Wiki-Art are better categorized, well-structured and its metadata is fully available compared to the MET and Rijksmuseum datasets that have missing and inconsistent metadata. Moreover, as we are mostly focusing on cultural paintings and our first experiment was based on paintings categories, the Wiki-Art dataset contains more paintings from diverse categories than the other datasets and they are mostly used in our experiments.

Before working with cultural datasets, we have cleaned the data and removed all duplicated images. We leverage paintings categorization to examine the image inpainting results in one of the experiments by training multiple DCGANs based on paintings categories. Therefore, five paintings categories are chosen from the Wiki-Art dataset, each category has 2000 paintings, to have a total of 10,000 paintings from all categories. The categories selected are Realism, Surrealism, Impressionism,

Expressionism, and Baroque. To prepare for the training process, the 10,000 Wiki-Art images are re-sized from their original size to 128×128 pixel sized images without reducing the original image's quality. The image size was chosen to be of this size so that the network can detect more details from the artworks to improve the generation results and complete images with realistic information during the image inpainting process. Also, due to the limited VRAM size in our machine, unfortunately, we cannot handle images of a size larger than 128×128 pixels which forced us to stop at this size.

CHAPTER 4: METHODOLOGY AND IMPLEMENTATION

To address the visual data completion problem in the cultural context, we will discuss in this chapter our designed and implemented cultural inpainting framework that combines visual clustering and multiple DCGANs to efficiently and effectively perform an accurate visual completion.

4.1 Painting completion framework

4.1.1 Motivation

Instead of relying on a single DCGAN for the completion, our semantic image inpainting approach is motivated by the divide-and-conquer strategy to train several "specialized GANs" by splitting the task of completing several cultural categories using one GAN trained on sparse dataset, to only a single category per GAN. The question now is how to split the training data across the categories efficiently. To ensure an effective grouping of similarly looking images, our solution leverages global visual features to perform unsupervised clustering using K-Means. The training scenario of our framework is as follows: we select the dataset of cultural art-work that we want to use for training. Then we compute visual global features of each image using either CNN features, SIFT [35] or SURF [36] features with Bag of Visual Words [37], etc. Once computed, these global features are clustered using the K-Means algorithm with an estimated number of cultural categories as the number of centroids (K). Once all the images have been clustered, a DCGAN is trained for each cluster. The training principle is illustrated in Figure 5.

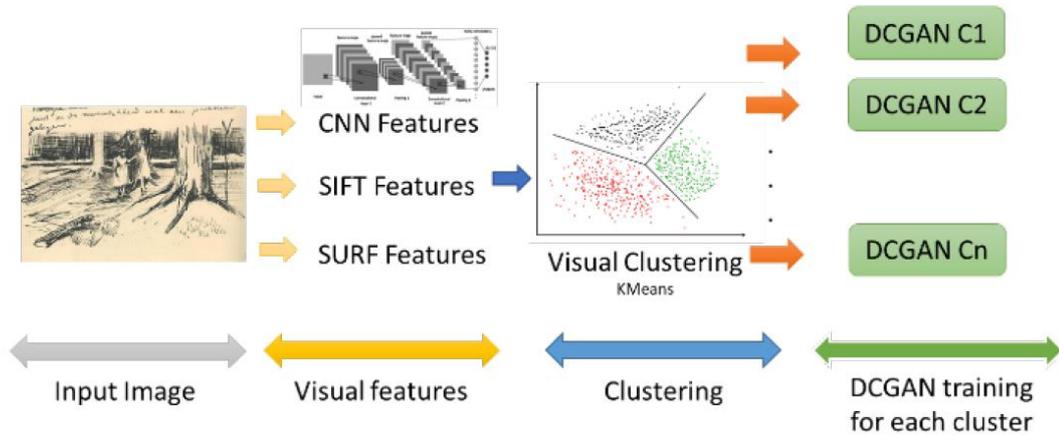


Figure 5: The Training Step of the Cultural Inpainting Framework

The completion scenario of our framework is as follows. We take an incomplete image of cultural artwork, and based on what visual information is available, we select the best matching cluster as per the last step. Once selected, the generator associated with this cluster is used to generate samples following a semantically constrained generation. The quality of these samples is evaluated using two losses as in the technique proposed in [17]. In the inpainting step, the damaged area will be replaced with the associated area from the generated image. When presented with incomplete image, our framework tries to identify its closest cluster and assigns the completion task to the DCGAN related to that cluster. Figure 6 summarizes the completion stage of our framework.

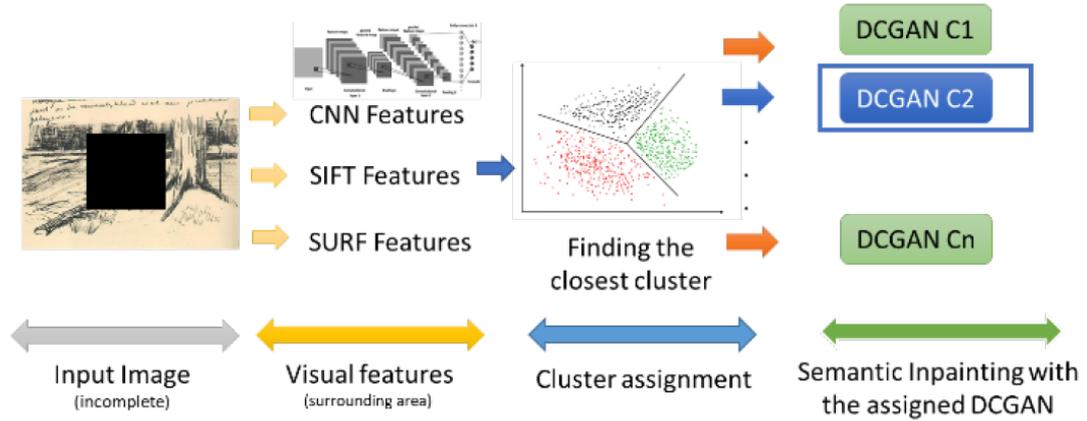


Figure 6: The Completion Step of the Cultural Inpainting Framework

4.1.2 Framework Design and Implementation

The architecture of the framework is based on semantic image inpainting with deep generative models implementation⁵ proposed by Yeh et al. [17]. We have tested the image inpainting algorithm on different image sizes and found that 128×128 is the most suitable size since it shows clear details in artworks compared to smaller image sizes like 64 or 32. Additionally, due to GPU memory limitation, this is the largest size that can be handled by the available VRAM for DCGANs training and inpainting. The network architecture obtained from [17] is modified based on the size of images that are fed to the network. One more deconvolutional layer is added in the Generator network and a convolutional layer in the Discriminator network to adapt the network structure to be able to handle input of size 128×128 . Also, the hyperparameters were tuned to improve the realism of the completed images. The trained DCGAN is following the architecture outlined in Figure 7.

⁵ Semantic image inpainting: https://github.com/bamos/dcgan-completion.tensorflow_

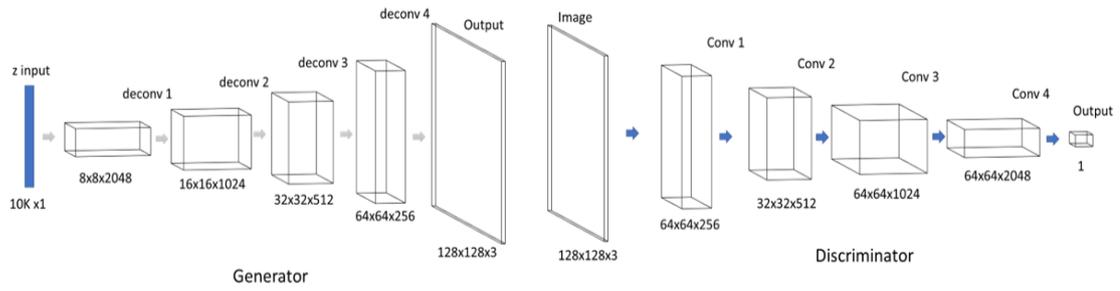


Figure 7: Our Framework DCGAN Architecture

Vanilla DCGANs are mainly used for content generation but not for completion. If a vanilla GAN is used for completion, the completed image will have a high chance to be filled with content that is not related to the remaining of the image. Therefore, we constraint the output of the DCGAN to have generated content with similar characteristics to the available content of the image. How the output of the generator will be constrained since the generator is fed by a random vector?. The authors in [17] propose an inpainting methodology that combines two types of losses, the contextual in addition to the perceptual loss (evaluated by the discriminator). This loss combination is used to perform a backpropagation on the input of the generator (z vector). The goal of the backpropagation optimization is to lower as much as possible this combined loss. As the z vector is the only parameter controlling the output after training the DCGAN, the goal is to generate an output (image) that minimizes the loss combination which will theoretically result in an image that looks similar to the one that we want to complete. Regarding the contextual loss, the authors use the L1 Norm as a distance measure between the generated content and the existing content removing the missing areas in both images. On the other hand, the prior loss forces the generator to generate images having similar features to the training dataset by having some penalties based on high-level image features instead of pixel-wise

difference. The authors stress the fact that this measure has to be weighted in order to ensure effective training. The weighting consists of giving high importance to pixels close to the missing regions and less importance to pixels far from those regions.

We have reached to the final design of our framework by executing multiple experiments to test the performance of DCGANs on cultural inpainting. The number of experiments is four and they are as follows:

4.2 Approach Evaluation

In the following, we present the experiments that we have designed and implemented to compare our approach with.

- **First Experiment: General DCGAN based Inpainting**

In the first experiment, a General DCGAN is trained on a mixture of 10,000 paintings collected from five categories, namely, Realism, Surrealism, Impressionism, Expressionism, and Baroque. After training the General DCGAN is used to complete images collected from the five categories.

- **Second Experiment: Categorized DCGAN based Inpainting**

In the second experiment, we train five DCGANs, each is trained on data from each category to build five Categorized DCGANs. Each Categorized DCGAN is trained using 2000 paintings. The question now, can a Categorized DCGAN trained on painting from one category be able to produce better-completed paintings than the General DCGAN that is trained on a variety of paintings? We have observed that the cultural inpainting using Categorized DCGAN produced better and sharper completed images compared to the results of the General DCGAN. The results of this experiment will be highlighted with further details in chapter 5.

- **Third Experiment: Specialized DCGAN based Inpainting**

Instead of using categorized paintings which are from the same painting category but

has different contexts, we will use similarly looking images to train DCGANs which are from the same context. Therefore, we have collected 1500 natural scene paintings from Google that has similarly looking images to train a GoogleImages DCGAN. After the GoogleImages DCGAN is trained, we use it for image inpainting on a group of paintings that are similar to the training data. It was observed that DCGANs trained on similarly looking data will complete the missing region with content that is highly related to the available data in the input. We conclude from this experiment that the 10,000 paintings from the five painting categories should be clustered where each cluster has similar looking paintings. For this reason, we want a method that would successfully separate similarly looking images into different groups based on the content. Afterward, we will use each group to train multiple DCGANs. We have found that K-Means clustering algorithm works perfectly in clustering data since it accepts unlabeled data just like in our case. The painting's features must be detected and extracted accurately so that K-Means algorithm can produce good clustering results. Nevertheless, the question remains, how the paintings features will be extracted to be fed to K-Means clustering algorithm?. To cluster the training data, we compute visual global features of each image using multiple feature extraction and detection methods to compare their performance in clustering. We chose SIFT (Scale Invariant Feature Transform) and SURF (Speeded Up Robust Features) features with Bag of Visual Words (BOVW) [37] and CNN features using VGG16 [38] and ResNet50.

The following list explains each feature extraction method we have used:

1. SIFT: Detects and identifies interesting keypoints in images using DoG method (Difference of Gaussian). Each keypoint represents scale, orientation, and location. Then, it computes descriptor for each image keypoint for local image

region by scanning the keypoints in different scales and orientations [35].

2. SURF: It also detects and identifies interesting keypoints in images just like SIFT.

Compare to SIFT, SURF is optimized such that it improves the speed of scale invariant feature detector by using Hessian matrix approximation instead of DoG method that is used in SIFT [36].

Both SIFT and SURF feature detection algorithms extract a list of descriptors from each image. Images cannot be clustered using only the descriptors because every image has a different number of descriptors and they cannot be compared using them directly. Here comes the idea of Bag of Visual Words (BOVW) [37] to cluster our dataset using image descriptors. BOVW is used for image classification and it is inspired by the NLP Bag of Words [39]. This idea creates a dictionary for the visual words that appear in the dataset. The dictionary is built by clustering images descriptors using K-Means algorithm. Then the resulting dictionary is used to compute a histogram for each image from the dataset. The histogram is a global features vector that counts the occurrence of each visual word in an image based on the number of visual words we have specified when creating the dictionary. Now, we feed the histograms to the K-Means algorithm to cluster the images. Our images now are visually clustered where each cluster has a group of similarly looking images.

3. VGG16: Visual Geometry Group 16 is a CNN pre-trained model with 16 weighted layers, trained on imageNet dataset for classifying images into 1000 classes [38]. We have discarded the classification part of the model and used the feature extraction part to extract artwork features.

4. ResNet50: Residual Networks are a type of very deep CNNs that uses skip connections between layers to solve the problem of vanishing gradient that occurs when training very deep networks [40]. ResNet50 is also a pre-trained CNN

model with 50 weighted layers, trained on imageNet dataset. ResNet50 extracts more hidden features from images than VGG16 since it is a deeper network. Similarly to VGG16, we only use the feature extraction part to extract artwork features and dispose the classification part.

The produced CNN features vector from VGG16 and ResNet50 is passed to K-Means clustering algorithm to cluster the artworks.

We have visually compared the feature extraction methods performance based on the clustering results. VGG16 outperformed the other methods in extracting better features by having more related artworks in each cluster. After the paintings are successfully clustered, a Specialized DCGAN is trained on each cluster. To evaluate our model, we save 15 paintings not used in training from each cluster for image inpainting. For our 10,000 paintings, we cluster them into 6 clusters and we chose one cluster for training a DCGAN. The inpainting results of the chosen cluster will be demonstrated in chapter 5.

- **Fourth Experiment: AutoEncoder based Image Inpainting**

In this last experiment, we investigate another image inpainting technique based on Convolutional AutoEncoders. We compare its results against the other approaches in terms of quality and realism. Similarly to GANs, Autoencoders combine two neural networks in their architecture: an encoder and a decoder [41]. Our implementation comprises 5 convolutional layers in both the encoder and the decoder as shown in Figure 8: AutoEncoder Architecture. AutoEncoders are capable of discovering structures and patterns within the data by learning a compressed representation of the input data in the shared middle layer. In our context, Autoencoders are used to reconstruct the missing region in the image. The input is the damaged image and the output is the recovered one. The architecture of the encoder relies on convolutional,

batch normalization and max-pooling layers. The decoder’s architecture includes convolutional, batch normalization and upsampling layers. In our implementation, the encoder input is a 3-channel image of 128×128 pixels with a missing central region. The output of the encoder represents the bottleneck for the network, also known as the latent space which is a compressed representation of the input: the network maps the input to the latent space by training the AutoEncoder (encoder-decoder) on masked images in an unsupervised setting. The decoder learns how to map the compressed representation into a restored visual output.

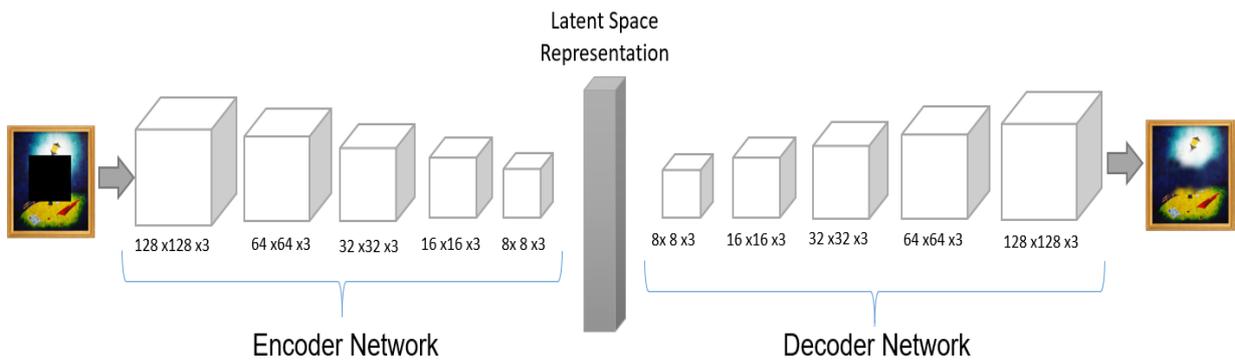


Figure 8: AutoEncoder Architecture

In our implementation, only the missing region is recovered from the output of the autoencoder (similar to the GAN implementation). The AutoEncoder network reduces the reconstruction loss by measuring the difference between the original image and the completed one. For evaluation, the same clusters from the *Third experiment* are used for comparison purposes. We trained 3 AutoEncoders, each one completing one type of paintings from a single cluster. The results of this experiment are highlighted with further details in chapter 5.

- **Hyperparameters Tuning**

During the GoogleImages and Specialized DCGAN experiments, we wanted to push the DCGAN to produce more plausible content by tuning two hyperparameters. The tuned hyperparameters are as follows:

1. Z Random Vector Dimension: in GoogleImages DCGAN experiment the Z dimension was tested by tuning the value between 100, 1000 and 10,000. Figure 9 shows the results of tuning the Z dimension. The Z dimension has shown improved results when it is assigned to 10,000 values.



Figure 9: Z Vector Dimension Tuning for GoogleImages DCGAN Inpainting

2. Learning Rate: in Specialized DCGAN inpainting the learning rate was tuned using multiple values starting from 1 to 0.007. Figure 10 shows the

results of tuning the learning rate. Setting the learning rate to 0.007 have shown more stable and accepted completed images and significantly decreased the loss. Reducing the learning rate more will make the training process last longer and produce the same output as if the learning rate is set to 0.007. For that reason, we stopped at 0.007 and chose it as the learning rate for training the inpainting DCGANs.



Figure 10: Learning Rate Tuning for Specialized DCGAN Inpainting

CHAPTER 5: EXPERIMENTAL RESULTS, VALIDATION AND EVALUATION

In this chapter, we will provide details about the experimental setup including the features of the machine used for training and inpainting, software libraries and the framework's hyperparameters. We will also investigate in depth the results of our experiments and their evaluation.

Experimental Setup

Our approach is implemented in Python version 3.6.3 using TensorFlow deep learning library (1.9.0). For experimental tests, we used a machine running the Ubuntu operating system (16.04 LTS) with an Intel Core i5-7600K CPU, 16 GB of RAM and an Nvidia Titan XP GPU. The DCGANs used in our experiments has the same architecture as the one outlined previously in Figure 7. Table 1 shows the training and inpainting hyperparameters that we used to train and validate our final approach. Since our approach is based on clustering, we used K-means clustering algorithm and set the number of clusters (K) to 6 for the 10,000 artwork samples collected from Wiki-Art dataset. Our selection was made for the purpose of having a reasonable number of samples in each cluster to effectively train a DCGAN for each cluster. The K value should reflect the diversity and the visual contexts found in the data. The selection of K remains ambiguous and requires further investigations to select the best number of clusters. Therefore, we will explore in the future the effect of varying the number of clusters on the performance of the framework, as well as on the quality of the produced images. All images that go through the inpainting process are fed the DCGAN with a centered mask.

Table 1. Training and Inpainting Hyperparameters.

Train LR	Epochs	Completion LR	Completion Iterations	Input Size	Z Dim	Optimizer
0.001	400	0.007	40,000	128×128	10,000	Adam

Results

The following section presents the three experiments we have conducted in this research and it discusses the obtained results of our image inpainting framework.

(a) Results of General DCGAN

General DCGAN was initially trained on a mixture of 10,000 images using the hyperparameters outlined in Table 2.

Table 2. General DCGAN Training and Inpainting Hyperparameters.

Train LR	Epochs	Completion LR	Completion Iterations	Input Size	Z Dim	Optimizer
0.001	400	0.001	40,000	128×128	100	Adam

Figure 11 shows image inpainting results for a selected number of paintings that were not seen by the General DCGAN during the training step.

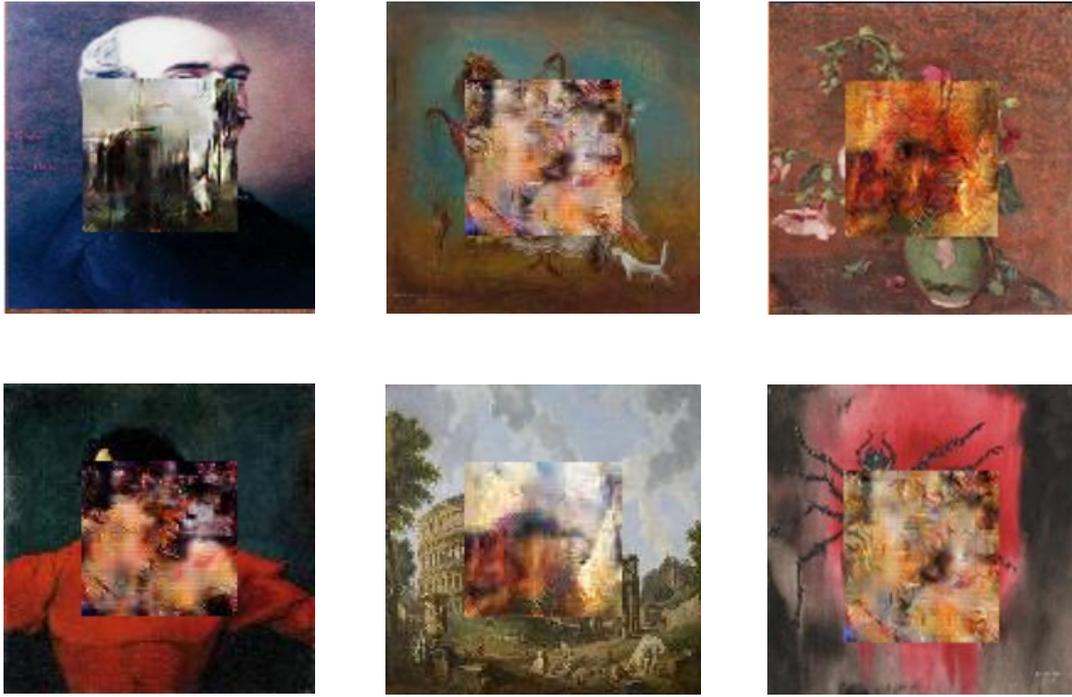


Figure 11: General Image Inpainting DCGAN Trained on Mixture of Images

It was observed that the General image inpainting DCGAN is filling the masked region in paintings with random and completely unrealistic content that in some cases has a repeated pattern in multiple filled images. The reason for this behavior is because the General DCGAN was trained on a variety of paintings that make the search space very fuzzy and prevents the DCGAN from producing realistic completed paintings. It indicates that training DCGANs on a mixture of paintings is an inefficient approach for image inpainting. Here comes the idea of experimenting the completion behavior of DCGANs when the training is based on paintings categories.

(b) Results of Categorized DCGANs

To train Categorized DCGANs for image inpainting, we choose 5 paintings categories each has 2000 images to have a total of 10,000 paintings. The chosen categories are

Realism, Surrealism, Impressionism, Expressionism, and Baroque.

Table 3. Categorized DCGAN Inpainting Results on Five Paintings Categories

Category	Original Image	Categorized DGAN	General DCGAN
Realism			
			
			
			

Surrealism



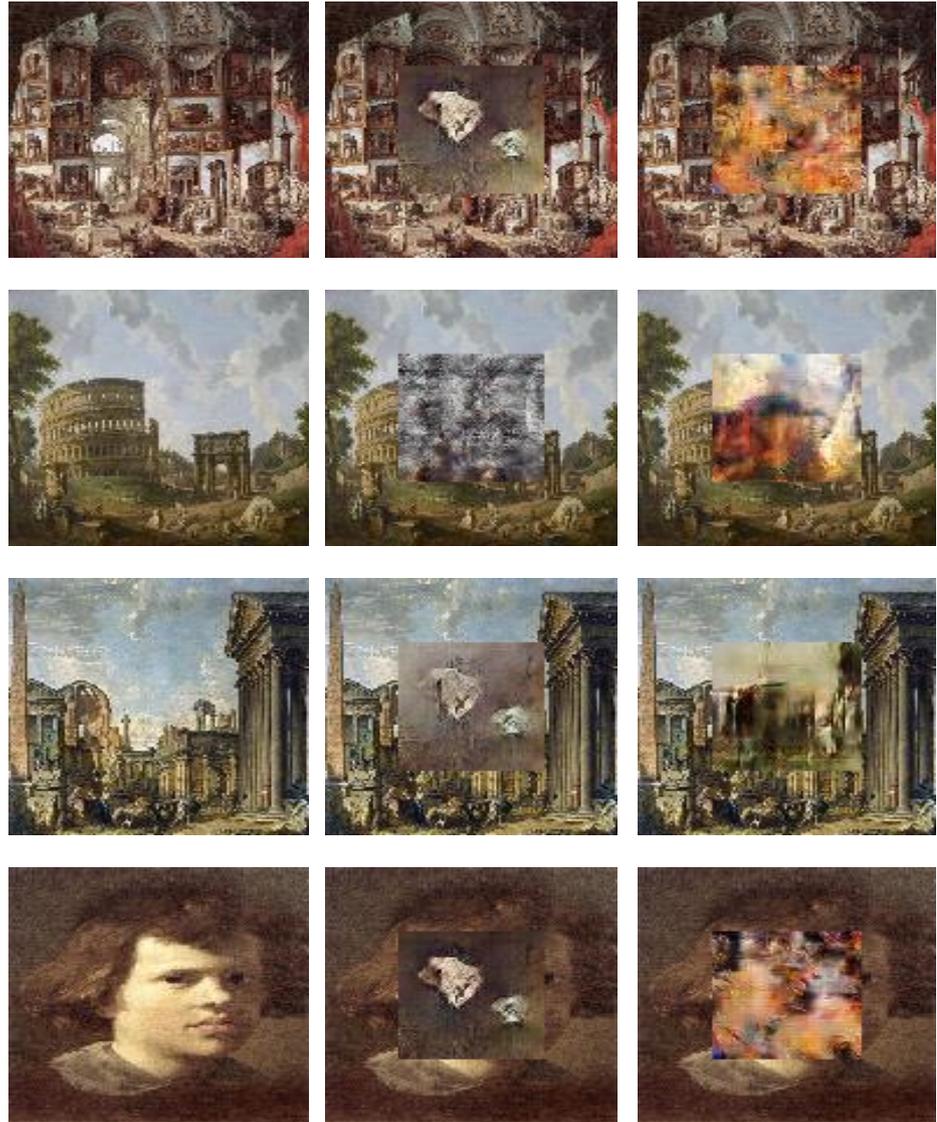
Impressionism



Expressionis
m



Baroque



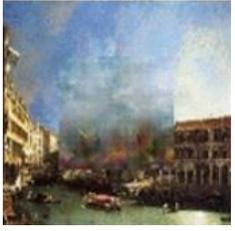
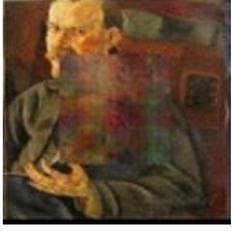
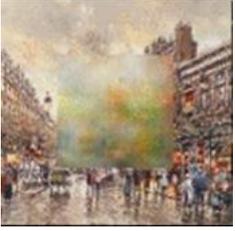
Comparing the inpainting results of the Categorized DCGANs and the General DCGAN, we have found that the Categorized DCGANs and the General DCGAN are producing unsatisfying results as its shown in Table 3. In both types of DCGANs, there is a repetitive texture appearing in different images of different content. The reason for this behavior is because the Categorized DCGANs are unable to detect the pattern of paintings in each category and could not find the hidden distribution behind the sparse dataset since each category has different patterns and styles in its paintings. Table 3 reflects 15 inpainting results from both DCGANs having 4 paintings from

each Category.

(c) Results of Specialized DCGANs

To investigate more about the cause of failing to complete paintings with realistic content based on paintings categories, we use a clustering approach to cluster our cultural dataset contextually. The clustering is based on global visual features of the dataset. We have chosen multiple feature extraction methods that will help in clusterings to choose the best performing one. The visual feature extraction methods are SIFT [35], SURF [36], VGG16 [38] and ResNet50 [40]. We have found that the visual features used for clustering have resulted in the best perceptual quality were CNN features based on the VGG16 CNN. We have used the same 10,000 paintings for training General DCGAN and Specialized DCGANs. Both types of DCGANs in this experiment are using the hyperparameters outlined in Table 1. For the Specialized DCGAN training, we have clustered the 10,000 paintings into 6 clusters and we chose 3 clusters to test the DCGAN training and inpainting. A number of paintings from those clusters are used for inpainting. The images used for inpainting are not used in training. Table 4 outlines some selected paintings that were completed using our framework (with clustering) compared to the General DCGAN that was trained with the whole dataset.

Table 4. Specialized DCGAN Vs General DCGAN Inpainting Results

Specialized DCGANs (clustering)	General DCGAN	Specialized DCGANs (clustering)	General DCGAN
			
			
			
			
			

From the inpainting results obtained from different experiments, it can be clearly noticed that a DCGAN trained on similarly looking images (like in Specialized

DCGAN) produces better inpainting results compared to a DCGAN trained on a mixture of images (like in General DCGAN) in terms of quality and realism of the output. By training a DCGAN on visually similar data, we have significantly restricted the visual output context of the DCGAN. The impact of adding clustering can be easily perceived on the image inpainting results because the content generated in the missing part of the image highly relates to the uncorrupted content of the masked input. The results discussed in this section have been communicated in the 2019 IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology (JEEIT) [42].

The training time for the Specialized DCGAN based on clustering is greatly decreased since we are using the divide-and-conquer strategy by clustering the training data into different small clusters to train smaller sized DCGANs. The General DCGAN trained on 10,000 paintings from five categories took approximately 32 training hours, while the Categorized DCGAN trained on 2000 paintings from each category took an average of 6.1 training hours per category. Training the Specialized DCGAN on 1200 paintings from one of the clusters took 3.4 training hours. Table 5 summarizes the training time and the number of samples used for training for the three types of DCGANs.

Table 5. Training time for each DCGAN type.

DCGAN Type	Number of Training Images	Training Time
General DCGAN	10,000	32
Categorized DCGAN	2000	6.1
Specialized DCGAN (one selected cluster)	1200	3.4

To further evaluate the performance of our Specialized DCGAN based on clustering, we have implemented a convolutional AutoEncoder for the purpose of image inpainting. The AutoEncoder architecture consists of an encoder-decoder pipeline that is trained on completing images with missing regions. The encoder takes an input with a central missing region and captures its context into a latent feature representation at the bottleneck layer by compressing the input. Then, the decoder decompresses the representation to have an image with content filled in the missing region of the input. We have trained 3 AutoEncoders, each is trained on paintings from the same three clusters that are used in the third experiment. Just like the Specialized DCGANs, each AutoEncoder is trained for 400 epochs using Adam optimizer. The loss function used is Cross Entropy that measures the distance between the completed output and the original input. Table 6: AutoEncoder Vs Specialized DCGAN Inpainting Results presents the image inpainting results of both AutoEncoders and Specialized DCGANs. The completed regions in the results obtained from the AutoEncoder are obviously blurry and visually implausible in all completed paintings. Additionally, in some paintings, the completed part does not relate to the remaining content of the paintings. Comparing the AutoEncoder and Specialized DCGANs results, our Specialized DCGANs have outperformed the AutoEncoder solution in image inpainting task in terms of quality and realism of the completed paintings.

Table 6: AutoEncoder Vs Specialized DCGAN Inpainting Results

Specialized DCGANs (clustering)	AutoEncoder	Specialized DCGANs (clustering)	AutoEncoder
			
			
			
			
			

For the purpose of evaluating the completed paintings using our proposed framework in comparison with General DCGAN and AutoEncoders, a survey has been conducted using Google Forms among eighteen participants from different backgrounds. The participants are objectively evaluating the realism and quality of completed paintings obtained from the three approaches. Five completed paintings from each approach are selected and viewed to the participants to evaluate them on a scale from one to five. The survey results showed that our framework has an improvement of 18.22% and 13.18% on the completed results compared to the ones obtained from the General DCGAN and AutoEncoders respectively. Also, our outcomes in this research have been viewed and discussed with the multimedia team of MIA to have their feedback on the obtained results. Their expressed interest opened additional possibilities of extension of our approach to their own collections. A meeting is under preparation for further details.

CHAPTER 6: CONCLUSION AND FUTURE WORK

In this thesis, we have presented a cultural inpainting framework adapted for completing visual cultural assets containing damaged areas. We have simulated damaged cultural assets by removing the central part of the image and predicting realistic content to be filled in the missing region with the help of our framework. The framework relies on deep convolutional generative adversarial networks (DCGAN), which are nowadays considered among the most powerful generative models. These models can be used to perform semantic image inpainting by generating realistic content, that is related to the available data in the image. However, through our analysis, we saw that using a single model with a dataset containing several visual contexts is ineffective. Therefore, we have designed a cultural inpainting framework, which has been validated on cultural data and can effectively perform visual completion of different contexts. Our framework is inspired by the divide and conquer strategy. Instead of training a single DCGAN to complete images from different visual contexts, we have clustered our training data using K-Means clustering algorithm, where each cluster contains contextually similar data to allow us to train a DCGAN for each cluster. Each DCGAN that is trained on the same visual context have in fact resulted in a better-quality completion.

By examining and comparing the completion results of the General DCGAN, AutoEncoders and the Specialized DCGAN, it is observed that the results of the Specialized DCGANs are sharper, more plausible and highly relates to the available data in the corrupted image. Our framework replaces the manual work of curators for reconstructing damaged areas in valuable cultural assets by automating it to reduce cost and time. We have targeted this topic to open the door for more research in the field of cultural heritage restoration because it is a very important research problem

and fewer researchers are paying attention to it.

During the literature review, it was observed that most of the research work related to image inpainting is focusing on producing models that are trained on huge datasets. Additionally, we have found that there is no research related to the completion of damaged cultural artworks which is a very important topic to cultural institutions and museums.

Our approach also presents some limitations related to DCGANs training and inpainting processes. Indeed, those processes are very time-consuming because they require hyperparameters and architecture adjustments to improve the quality and realism of the completed outputs after long training and inpainting iterations. Adding to that, DCGAN training and inpainting do not have clear stop criteria and there is no explicit evaluation criteria since the completed image depends on perceptual evaluation. The input size was limited to the size of 128×128 due to GPU memory limitation and this is the largest size that can be handled by the available VRAM for DCGAN training and inpainting.

Future Work

Although our proposed framework shows promising results, we target several improvements regarding the quality of the output. In the following, we discuss such potential enhancements for our work.

- Since each Specialized DCGAN needs to address homogenous set of similarly looking images, the expressive power of the model needs to be decreased because it is not handling a variety of images. Therefore, the requirements of the processing power can be optimized by reducing the number of parameters and optimizing the network's architecture for the Specialized DCGANs.

- In the case of a large number of clusters, we expect fewer samples in each cluster and it will affect the performance of the framework since there is not enough data. Therefore, it is important to explore the effect of varying the number of clusters on the performance of the framework and its impact on the quality of the produced images.
- As there are multiple cultural categories, we aim at using our approach with other categories of cultural data such as pottery, carpets, swords, or statues to train different Specialized DCGANs that will be used for image inpainting.
- Our approach relayed on specific cultural data which is paintings. However, understanding how it performs on other domains such as missing audio signals, missing videos clips, etc., remains to be discovered and tested.
- A collaboration is currently under investigation with Chengdu University research team led by Prof. Xi Yu to apply CNNs and GANs in the medical field for cancer detection.

REFERENCES

- [1] D. De la Fuente, J. M. Vega, F. Viejo, I. Díaz, and M. J. J. o. c. h. Morcillo, "Mapping air pollution effects on atmospheric degradation of cultural heritage," vol. 14, no. 2, pp. 138-145, 2013.
- [2] C. Cardell *et al.*, "Risks of atmospheric aerosol for cultural heritage assets in Granada (Spain)," p. 45, 2013.
- [3] M. A. Rogerio-Candelera, M. Lazzari, and E. Cano, *Science and technology for the conservation of cultural heritage*. CRC Press, 2013.
- [4] K. Ikeuchi *et al.*, "The great buddha project: Digitally archiving, restoring, and analyzing cultural heritage objects," vol. 75, no. 1, pp. 189-208, 2007.
- [5] F. Stanco, S. Battiato, and G. Gallo, *Digital imaging for cultural heritage preservation: Analysis, restoration, and reconstruction of ancient artworks*. CRC Press, 2011.
- [6] K. Ikeuchi and D. Miyazaki, *Digitally archiving cultural objects*. Springer Science & Business Media, 2008.
- [7] W. Raghupathi, V. J. H. i. s. Raghupathi, and systems, "Big data analytics in healthcare: promise and potential," vol. 2, no. 1, p. 3, 2014.
- [8] B. Baesens, W. Verbeke, and V. r. v. Vlasselaer, "Fraud analytics using descriptive, predictive, and social network techniques: a guide to data science for fraud detection," 2015.
- [9] T. J. F. E.-G. S. C. o. I. I. Kumar, "Implementation of the indian national tsunami early warning system," pp. 380-391, 2009.
- [10] C. Ledig *et al.*, "Photo-realistic single image super-resolution using a generative adversarial network," *arXiv preprint*, 2017.
- [11] S. Iizuka, E. Simo-Serra, and H. Ishikawa, "Globally and locally consistent image completion," *ACM Transactions on Graphics (TOG)*, vol. 36, no. 4, pp. 107-107, 2017.
- [12] S. Pascual, A. Bonafonte, and J. J. a. p. a. Serrà, "SEGAN: Speech enhancement generative adversarial network," 2017.
- [13] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1125-1134.
- [14] G. Antipov, M. Baccouche, and J.-L. Dugelay, "Face aging with conditional generative adversarial networks," in *2017 IEEE International Conference on Image Processing (ICIP)*, 2017, pp. 2089-2093: IEEE.
- [15] H. Zhang *et al.*, "Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 5907-5915.
- [16] D. Nie *et al.*, "Medical image synthesis with context-aware generative adversarial networks," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 2017, pp. 417-425: Springer.
- [17] R. A. Yeh, C. Chen, T.-Y. Lim, A. G. Schwing, M. Hasegawa-Johnson, and M. N. Do, "Semantic Image Inpainting with Deep Generative Models," in *CVPR*, 2017, vol. 2, pp. 4-4.
- [18] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester, "Image inpainting," in *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, 2000, pp. 417-424.

- [19] M. M. O. B. B. Richard and M. Y.-S. Chang, "Fast digital image inpainting," in *Appeared in the Proceedings of the International Conference on Visualization, Imaging and Image Processing (VIIP 2001), Marbella, Spain, 2001*, pp. 106-107.
- [20] Z. Xu and J. J. I. t. o. i. p. Sun, "Image inpainting by patch propagation using patch sparsity," vol. 19, no. 5, pp. 1153-1165, 2010.
- [21] M. Bertalmio, L. Vese, G. Sapiro, and S. J. I. t. o. i. p. Osher, "Simultaneous structure and texture image inpainting," vol. 12, no. 8, pp. 882-889, 2003.
- [22] H.-y. Zhang, Q.-c. J. J. o. i. Peng, and graphics, "A Survey on Digital Image Inpainting [J]," vol. 1, 2007.
- [23] J. Hays and A. A. Efros, "Scene completion using millions of photographs," in *ACM Transactions on Graphics (TOG)*, 2007, vol. 26, pp. 4-4.
- [24] C. Huang and K. Yoshida, "Evaluations of Image Completion Algorithms: Exemplar-Based Inpainting vs. Deep Convolutional GAN," 2017.
- [25] Y. LeCun, Y. Bengio, and G. J. n. Hinton, "Deep learning," vol. 521, no. 7553, p. 436, 2015.
- [26] I. Goodfellow *et al.*, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014, pp. 2672-2680.
- [27] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *arXiv preprint arXiv:1511.06434*, 2015.
- [28] X. Wu, K. Xu, and P. Hall, "A survey of image synthesis and editing with generative adversarial networks," *Tsinghua Science and Technology*, vol. 22, no. 6, pp. 660-674, 2017.
- [29] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros, "Context encoders: Feature learning by inpainting," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2536-2544.
- [30] U. Demir and G. Unal, "Patch-Based Image Inpainting with Generative Adversarial Networks," *arXiv preprint arXiv:1803.07422*, 2018.
- [31] A. Belhi, A. Bouras, and S. Foufou, "Leveraging known data for missing label prediction in cultural heritage context," *Applied Sciences*, vol. 8, no. 10, pp. 1768-1768, 2018.
- [32] C. Tomkins, "Merchants and masterpieces: the story of the Metropolitan Museum of Art," 1991.
- [33] Z. Katz, "Pitfalls of open licensing: An analysis of Creative Commons licensing," vol. 46, p. 391, 2005.
- [34] T. Mensink and J. Van Gemert, "The rijksmuseum challenge: Museum-centered visual recognition," in *Proceedings of International Conference on Multimedia Retrieval*, 2014, p. 451: ACM.
- [35] G. Lowe, "SIFT-The Scale Invariant Feature Transform," *J. Comput. Vision*, vol. 60, pp. 91--110, 2004.
- [36] H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: Speeded up robust features," in *European conference on computer vision*, 2006, pp. 404-417: Springer.
- [37] Y. Yang and S. Newsam, "Bag-of-visual-words and spatial extensions for land-use classification," in *Proceedings of the 18th SIGSPATIAL international conference on advances in geographic information systems*, 2010, pp. 270-279: ACM.
- [38] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

- [39] E. Cambria and B. J. I. C. i. m. White, "Jumping NLP curves: A review of natural language processing research," vol. 9, no. 2, pp. 48-57, 2014.
- [40] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770-778.
- [41] J. Xie, L. Xu, and E. Chen, "Image denoising and inpainting with deep neural networks," in *Advances in neural information processing systems*, 2012, pp. 341-349.
- [42] N. Jboor, A. Belhi, A.-A. Abdulaziz, A. Bouras, and A. Joaua, "Towards an Inpainting Framework for Visual Cultural Heritage," *IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology (JEEIT)* 2019.

APPENDIX A : COMPLETION EVALUATION SURVEY

Images Reality and Quality Evaluation

Form description

The following images will be viewed to evaluate their quality on scale from 1 to 5
(1 is least real, 5 is most real)

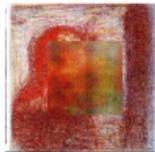


1-Rate the quality of the image on the scale from 1 to 5 *



- 1
- 2
- 3
- 4
- 5

2-Rate the quality of the image on the scale from 1 to 5 *



- 1
- 2
- 3
- 4
- 5

3-Rate the quality of the image on the scale from 1 to 5 *



- 1
- 2
- 3
- 4
- 5

4-Rate the quality of the image on the scale from 1 to 5 *



- 1
- 2
- 3
- 4
- 5

5-Rate the quality of the image on the scale from 1 to 5 *



- 1
- 2
- 3
- 4
- 5

6-Rate the quality of the image on the scale from 1 to 5 *



- 1
- 2
- 3
- 4
- 5

7-Rate the quality of the image on the scale from 1 to 5 *



- 1
- 2
- 3
- 4
- 5

8-Rate the quality of the image on the scale from 1 to 5 *



- 1
- 2
- 3
- 4
- 5

9-Rate the quality of the image on the scale from 1 to 5 *



- 1
- 2
- 3
- 4
- 5

10-Rate the quality of the image on the scale from 1 to 5 *



- 1
- 2
- 3
- 4
- 5

11-Rate the quality of the image on the scale from 1 to 5 *



- 1
- 2
- 3
- 4
- 5

12-Rate the quality of the image on the scale from 1 to 5 *



- 1
- 2
- 3
- 4
- 5

13-Rate the quality of the image on the scale from 1 to 5 *



- 1
- 2
- 3
- 4
- 5

14-Rate the quality of the image on the scale from 1 to 5 *



- 1
- 2
- 3
- 4
- 5

15-Rate the quality of the image on the scale from 1 to 5 *



- 1
- 2
- 3
- 4
- 5