Exploring Generative Methods For Smart Inpainting of Fine Art

Oscar O'Rahilly
Mathematical and Computational Science

1 Introduction, Motivation

1.1 Introduction and Motivation

The goal of this project is to explore different computational methods of inpainting on artificially damaged paintings. Unfortunately, paintings are prone to a variety of different forms of damage. Natural damage as a result of light, heat, moisture, air pollutants, dust, dirt, insects, physical vibration and impact can lead to slow deterioration of, or sudden damage, to a painting. Humans can also damage paintings. For example, in 2017 a 12-year-old Taiwanese boy tripped and accidentally punched a hole in a $1.5million Paolo Porpora oil on canvas. A more deliberate example occurred in May 27, 1993, when an explosion set off by the Italian mafia ripped through the heart of Florence, killing five and maiming the city's art and architecture.

Fixing damage caused to art, whether the damage be natural or man-made, is also a very expensive, time-consuming, and sometimes risky procedure. For some works, the cost of getting a damaged area restored may simply be too expensive or too risky. The use of smart computational methods for infilling therefore provides a way to bring back to life damaged work in a fast and almost inexpensive manner. It can also provide a very rough guide to conservators who wish to quickly see what a restored image may look like. Within the commercial world, these methods and techniques could even be used in galleries to show visitors, on their phone, what a damaged painting might have looked like in its original state.

The motivation, therefore, to develop accurate tools to infill paintings is certainly present. The task of inpainting has also been present for a long time in the domain of real-life photography. Filling in missing areas of photographs of people or places has maintained a lot of interest within the Computer Science community for many years. Recent advances in Deep learning through architectures like Generative Adversarial Networks (GANs) have propelled this task forward to a state where reliably distinguishing between photos that have been infilled or not is extremely difficult. These same techniques, however, have seen less success in the art domain notably due to the fact that these models require massive amounts of image data to perform well. Amassing datasets consisting of hundreds of thousands of photographs in the modern day is a relatively simple task; however, datasets of this scale are not as readily available for art. Furthermore, art is not just constrained to one genre. Paintings created by an impressionist artist occupy a completely different feature space to paintings created by an abstract artist. This matter of multiple different types of art makes creating these large datasets even more difficult. Thus, the challenge of creating believably infillings for paintings is still very much present.

In this paper I will explore a two main methods of performing infilling computationally. The first are two algorithmic approaches which serve as computationally inexpensive baselines. The second are Deep Learning approaches, which ultimately provide extremely impressive results.

35th Conference on Neural Information Processing Systems (NeurIPS 2021).

2 Initial Data

2.1 Target Paintings

As I mentioned, one of the more difficult challenges in creating an generalizable infilling model for art is taking into account the wide variance in style. In order to robustly test my approaches I decided to use three different pieces of art in three very different styles. Also, in order to keep everything fair amongst my approaches I kept the size of each image the same (256x256). The three paintings I used were Monet's *The Water Lily Pond*, Raphael's *The School of Athens*, and De Kooning's *Composition*.

Figure 1: *The Water Lily Pond*, *The School of Athens*, *Composition*

2.2 Artificial Damage

Another aspect of this project I was very keen to pay attention to was making sure that the "damage" I applied to these paintings resembled realistic damage that a painting might fall prey too. To do this I found three different paintings, all with varying degrees of damage. I then isolated the damaged regions to form a mask of the damage, which I could then easily apply to my target paintings. The masks, and the original paintings they are from, are shown below and are labelled small, medium and large, depending on the amount of "damage" they do to a painting.

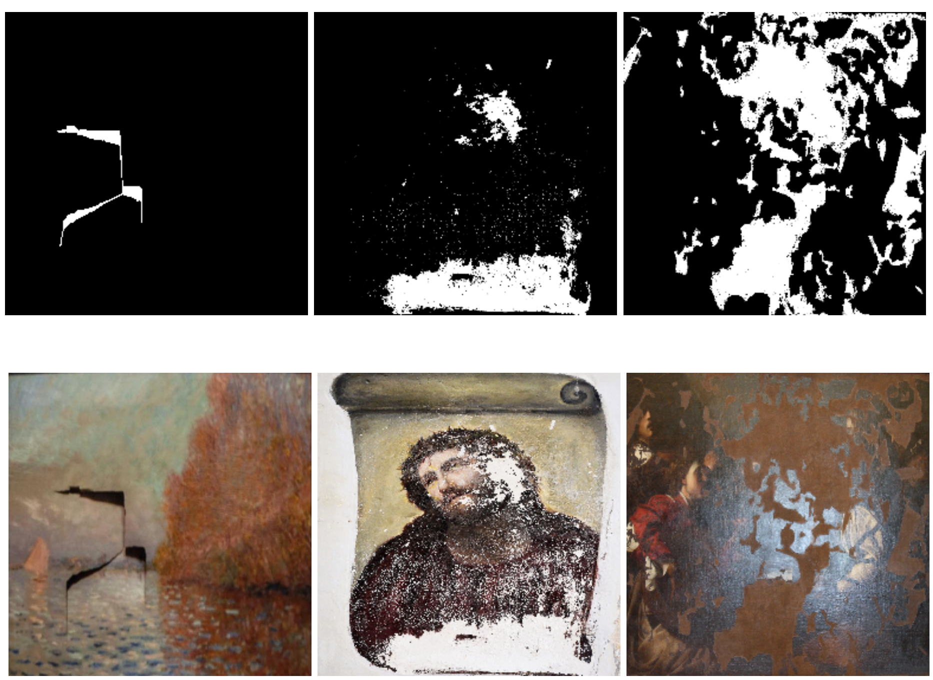

Figure 2: From left to right: small, medium, large mask and their original image

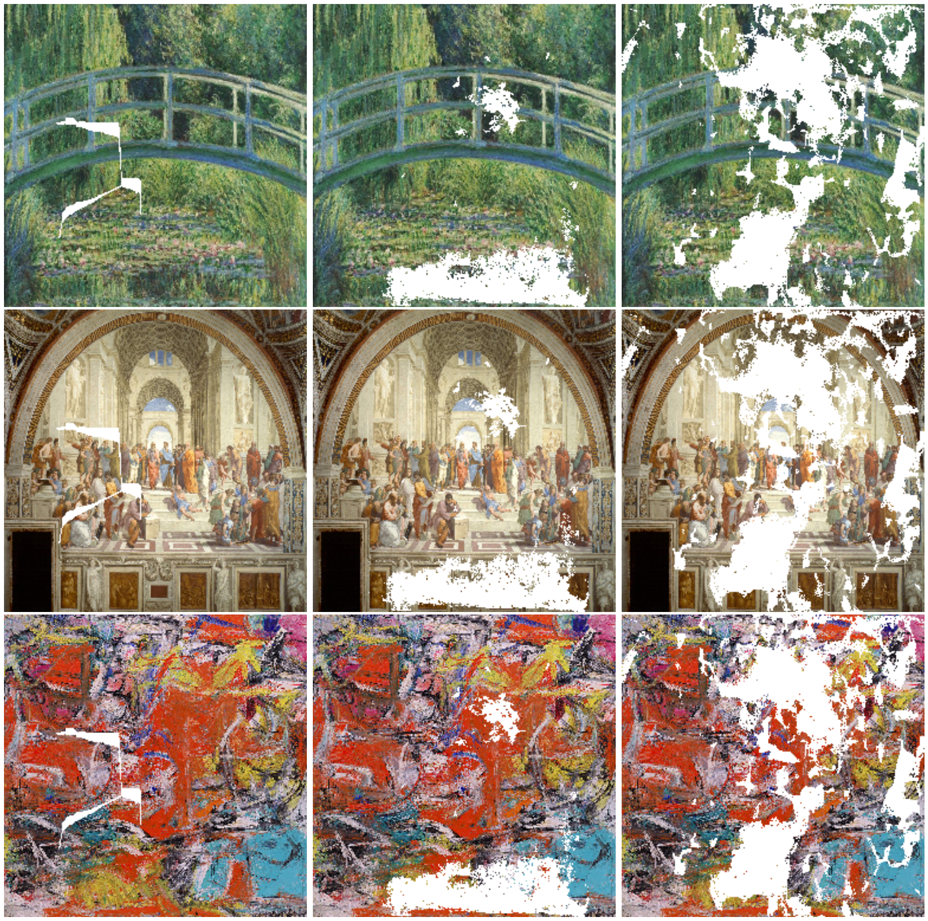

Figure 3: Damaged paintings

3 Methods

3.1 Algorithmic

The algorithmic methods discussed below serve as a baseline. They demonstrate extremely simple, computationally inexpensive ways of infilling and serve as good comparison to the GAN methods discussed later.

3.1.1 Dominant Color Infilling (baseline)

This method is extremely simple. It essentially performs K-means clustering (with $k = 1$) on the undamaged region of a painting and uses that RGB color cluster value to infill the remainder of the image.

3.1.2 Weighted Nearest Neighbor

This method is slightly more involved. For each pixel in the damaged region, a weighted average of the surrounding pixels is taken. The result of this weighted average is then used as the target's pixel value. The weights assigned to the pixels are assigned relative to the distance to the target pixel, with pixel values further away from the target being assigned a lower weight.

3.2 Deep Learning

With the rapid advancement in computational power and the wide accessibility of data, deep learning approaches for generative tasks have quickly become industry standards. For my infilling tasks I will be using the GAN architecture trained on two different datasets: Places2 & Places2+WikiArt.

3.2.1 Generative Adverserial Networks

Since their invention by Ian Goodfellow in 2014, GANs have exponentially accelerated generative modelling tasks. Shown below is an example of the advancements in results: the five pictures show a different sample generated from a generative model trained on human faces. In only three years we went from grainy black and white images to high-definition hyper realistic color images of human faces.

Figure 4: Generation of human faces using generative models by year

The technical insight into what makes GANs so powerful is their adverserial nature. GANs consist of two separate neural networks: a discriminator and a generator. Throughout training, the goal of the generator is to produce unseen images that resemble the underlying distribution of the training set. The discriminator, on the other hand, is trained to discern between samples generated from the generator and samples actually from the dataset. By optimizing both of these two networks, we force the generator to produce samples more representative of the underlying distribution. This is known as a MiniMax optimization problem. The objective function is shown below. Here D is the discriminator and G is the generator.

$$\min_G \max_D V(D, G) = \mathbb{E}_{x \sim p_{data}(x)}[\log D(x)] + \mathbb{E}_{z \sim p_z(z)}[\log(1 - D(G(z)))]$$

Figure 5: GAN optimization objective

The model I used is based off the model discussed in Guilin Liu et al's paper *Image Inpainting for Irregular Holes Using Partial Convolutions*.

3.2.2 GAN trained on Places2 - (GAN1)

The first model I experimented with was a GAN trained on the Places2 dataset. The Places2 dataset consists of over 1.8 million images, depicting 400+ unique scene categories. Examples include people, nature, buildings and objects. I decided to use Places2 due to the wide variety of images it contains and the sheer size of the dataset. Therefore, despite the fact it contains no images of artworks, the model will still hold a very strong representational capacity for colors, objects and their relationships to each other.

3.2.3 GAN trained on Places2 & WikiArt - (GAN2)

Unfortunately, however, Places2 does not contain images of art. As the goal of this project is to create an infilling model for art, I extended the dataset to include images from WikiArt. For each painting type I extended the dataset to include images from the WikiArt dataset that I believed would aid performance. Namely, I finetuned my Places2 model using pictures categorized by WikiArt that are similar to each of my target images. Each category below is one labelled by WikiArt and the corresponding number is the number of examples in the category. Note, that there are no overlaps amongst the categories.

1. Monet's *The Water Lily Pond*: finetuned the Places2 model on landscape images (15,000) and flower paintings (1,800)
2. Raphael's *The School of Athens*: finetuned on religious paintings (8,394) cityscape (6,600) and mythological paintings (2,200)
3. De Kooning's *Composition*: finetuned on abstract paintings (15,000)

4 Results

4.1 Metrics

Evaluating the quality of an inpainted image is very difficult, namely because it is a subjective task. One person may ultimately think one infilling looks better over another for purely subjective reasons. When evaluating my different methods, however, I wanted to find some quantitative metric as well as my own subjective score.

For the quantitative metric I looked at the all the pixels filled in by a particular model, and calculated the euclidean distance between the CIE chromaticity values of the infilled pixels vs their corresponding CIE chromaticity in the original image. I reported this score for each mask applied. The equation for this score metric is shown below. Lower scores are therefore indicative of a *"better"* infilling.

$$Score = \sqrt{\sum_{p \in P} (p_{\text{infilled}} - p_{\text{ground truth}})^2}$$

Where P is the set of all pixels "damaged" by the mask.

4.2 Quantitative results

The tables below show the quantitative results as discussed in the metric sections above. The bold cells indicate the lowest score across the columns: in other words the best performing model on the given mask.

	Small - Sum	Medium - Sum	Large - Sum
Monet - GAN2	90.6	**749.4**	**1957.4**
Monet - GAN1	**81.2**	940.7	2213.7
Monet - WNN	89.77	1109	2520.2
Monet - Baseline	107.6	1187.9	3019.1

	Small - Sum	Medium - Sum	Large - Sum
Raphael - GAN2	143.3	**756.5**	**2762.2**
Raphael - GAN1	**126.3**	1146	3342.7
Raphael - WNN	142.2	1243.4	3700.2
Raphael - Baseline	340.1	1940.2	5738.4

	Small - Sum	Medium - Sum	Large - Sum
De Kooning - GAN2	189.6	**1086.2**	**3344.4**
De Kooning - GAN1	**161.6**	1488.3	3610.1
De Kooning - WNN	176.7	1547.2	3983.6
De Kooning - Baseline	196.1	1779.1	4776.2

4.3 Dominant Color Infilling (baseline)

As expected this yielded the worst results; however, highlights that for very small damages this method can look somewhat believable. An interesting example is the good mask on De Kooning's *Composition* (bottom left image in Figure 6). Due to the abstract nature of the painting the mask, while not correct still looks visually appealing.

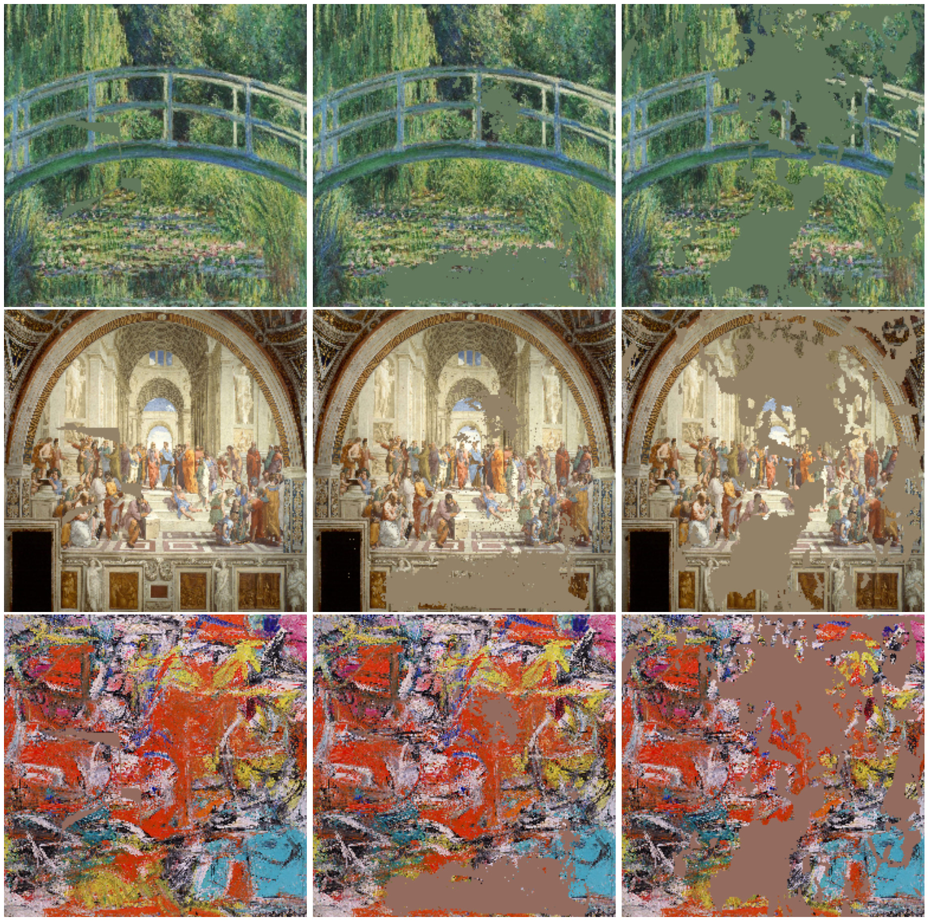

Figure 6: Baseline Results

4.4 Weighted Nearest Neighbour

The Weighted Nearest Neigbour method performed surprisingly well visually, especially on the smaller masks. In fact, on the smallest mask the infill looks almost identical to the original.

See Figure 7.

4.5 GAN1

The GAN trained on Places2 saw strong performance on all three masks. On the larger masks it becomes increasingly less like the original image; however, the results are still surprisingly good considering this was trained only on images. The reason for such a strong performance is no doubt due to the enormous size of the Places2 which even though was not trained on art, is able to learn complex relationships between colors and objects in images. Interestingly, when we look at the quantitative metric, GAN1 consistently performs better on then smallest mask. As this mask is so small the added benefit of the WikiArt data does not add anything significant. Infilling the small mask, despite painting's genre, is more of color matching than anything else.

See Figure 8.

4.6 GAN2

GAN2 is the best performing model. Subjectively, I believe it provides the best infilling, especially on the largest mask. When we look at the quantitative scores we also see that the sum of euclidean distances between the infilled and ground truth pixels are lowest in GAN2. This, of course, makes a lot of sense. By finetuning our model on datasets similar in genre to the target images, our model is able to better approximate an underlying distribution for each genre of art.

See Figure 9.

5 Conclusion

Overall, the results found are extremely promising. We see that the Deep Learning approach vastly improved upon the more computationally lightweight algorithmic methods and for both the small and medium masks, look extremely (if not near identical) to the original images. Furthermore, when comparing the the two different GAN methods, we are able to make improvements through further training on more representative datasets. The fact this is supported both subjectively and quantitatively provides strong evidence for a clear improvement in the two models. With more advance computational power, these GANs could be extended even further with more representational and finer curated train sets for each piece of art. From the findings I have discussed in this project we would expect to see significant improvements through such an extension.

Figure 7: Weighted Nearest Neighbour Results

Figure 8: GAN1 Results

Figure 9: GAN2 Results