13th International Conference on Advanced Computational Intelligence (ICACI)
Wanzhou, China, May 14-16, 2021

2021 13th International Conference on Advanced Computational Intelligence (ICACI) | 978-1-6654-1254-4/20/$31.00 ©2021 IEEE | DOI: 10.1109/ICACI52617.2021.9435888

ArtGAN: Artwork Restoration using Generative Adversarial Networks

Abhijit Adhikary*, Namas Bhandari*, Evan Markou*, and Siddharth Sachan*
Research School of Computer Science
The Australian National University
Canberra, ACT, Australia
emails:{abhijit.adhikary, namas.bhandari, evan.markou, siddharth.sachan}@anu.edu.au

***Abstract*—We propose a method to recover and restore artwork that has been damaged over time due to several factors. Our method produces great results by completely removing damages in most of the images and perfectly estimating the damaged region. We achieved accurate results due to (i) a custom data augmentation technique which depicts realistic damages rather just blobs (ii) novel CResNetBlocks that subsequently upsample and downsample features to restore the image with efficient backpropagation measures, and (iii) the choice of using patch-discriminators to achieve sharpness and colorfulness. Our network architecture is a conditional Generative Adversarial Network where the generator uses a combination of adversarial loss, L_1 loss and the discriminator uses binary cross-entropy loss for optimization. While the expressiveness of existing comparison methods is limited, we present our results with several metrics for future comparison and showcase some visuals of recovered artwork. PyTorch implementation is available at: https://github.com/namas191297/artgan.**

***Keywords*—Generative Adversarial Network, Image Restoration, Artwork, Neural Network, Image Inpainting.**

I. Introduction

Paintings and other similar works of art have always been a vital part of society and culture throughout history. As the years pass by, however, such artworks tend to deteriorate due to several reasons. Artwork restoration can be an extremely delicate and tedious process which requires meticulous detail by the experienced conservators. Deep learning techniques can be used to digitally restore defects (such as scratches, blurs, cracks, halftones, paint, paper damages, rips, tears etc) in images [1]. Although a lot of research has been done in the image inpainting field, most of it focuses on images rather than paintings or sketches. Moreover, previous works using deep leaning for digital artwork restoration [2], [3] do not use the state of the art adversarial approach which is common in the image inpainting domain.

Bringing these together, in this work, we propose a novel generator architecture and use adversarial training to predict original images given the damaged ones. First, we combine the artwork from three artists Paul Cezanne, Claude Monet, and Vincent Van Gogh, as well as from the Japanese art named Ukiyo-e to make our base dataset, which we called ArtNet. We then apply a variety of damage masks to create a damaged image which is used for training with the original image. We achieve qualitative scores on par with previous works, but the absence of a benchmark datasets makes it a difficult comparison. Qualitative results show the adaptability of our model to different styles of artwork. Our contributions are summarized as follows:

- We compose a dataset that consists of artwork from different artists, facilitating better generalization across images.
- We use an innovative technique of appending realistic damages to the input image so that it resembles damaged artwork in the real world.
- We propose a network architecture inspired by pix2pix, with a new CResNetBlock component that makes use of dense connectivity, residual connections and efficient connectivity for backpropagation.
- We use a patch-discriminator of different patch-sizes that allows the network to generate sharp and colorful representations/outputs.

II. Related Work

A. Image inpainting by deep generative models

Image inpainting has been a research topic in computer vision for a long time. Traditional methods use local edge information to find the direction for diffusion and spread the known information to the edges [4]–[6]. However, these methods are limited to small patches and predict unrealistic results when complex features are involved. Recently, the usage of deep generative models to inpaint images has yielded exciting results. Early works used transpose convolution layers after convolutions to get back to the same dimensions as the input. This architecture was used for image denoising and restoration [7], [8]. After [9] proposed a framework for estimating generative models via an adversarial process, Generative Adversarial Networks (GANs) became the go-to model for image inpainting. [10] proposed an encoder-decoder generation network along with a CNN to classify the generated image as real or fake. They used a combination of L_2 and adversarial loss for end-to-end training of the network. Although they reported state of the art results, the predicted images could not reproduce perceptually consistent features and worked only for rectangle-shaped masks. In our work, we use L_1 loss to get crispier results and variety of damage

*All authors contributed equally.

masks, not limited to some specific shape. [11] improved this by using local and global discriminators. The global discriminator looks at the entire image to assess plausibility, while the local discriminator looks at the area around each reproduced patch to ensure consistency. They also used dilated convolutions to exponentially increase the perceptive fields. We have used something similar to the local discriminator, which takes patches of the images to classify them as real or fake. Several works have also used attention mechanisms to improve the results [12], [13]. However, these methods require coarse predictions before applying the attention, hence increases the computation complexity. To address this, [14] proposed parallel decoders with shared weights, one for the coarse reconstruction and another for the refined image. In future work, we would use context attention along with our generator architecture to improve the results.

In parallel, there has been a lot of work in guided image inpainting using conditional GANs. [15] proposed *pix2pix*, a conditional GAN based image to image translation approach applicable for multiple purposes. This includes image generation from edges, image inpainting and segmentation. Our discriminator is inspired by this work. [16] proposed a network which first estimates the missing edges and then combines them with the input image to generate the image. They reported better results for complex structures i.e. faces as the network incorporates edge information to produce more realistic results. Extending this to the field of image editing, [17] presented a system to complete an image given the masks and auxiliary guidance. An ideal system for artwork restoration will take some extra input (like lines, increase brightness, change style etc.) from the user (a domain expert who knows what is missing) and combine it with the damaged image to predict the final reconstructed result. This is a possible direction for future work.

B. *Digital artwork restoration*

Some works have tried to assist domain experts by providing a virtual restoration estimate of the deteriorated artwork by utilising computer vision and deep learning techniques [18]. [2] proposed a hybrid model to jointly generate masks and inpaint images of defected artworks. They used a Mask R-CNN [19], an extended version of the classical Faster R-CNN [20], to automatically generate the mask for the irregular regions in each painting. Then, the masked image is passed through the second model, a U-Net [21] architecture with partial convolutions, intending to restore the damaged artwork to its original form. [3] proposed a multi-scale CNN framework that combines deep CNN networks and multi-resolution image analysis for pixel-wise prediction. However, this technique heavily relies on the availability of existing real-world drawing reproductions (pair of current faded drawing, and the drawing's less-faded reproduction). They also emphasised the need for a benchmark dataset to evaluate reconstruction performance. Although at the time of each publication, their approach was novel, GANs surpassed the classical CNN architectures as they tend to have a better generalisation of results.

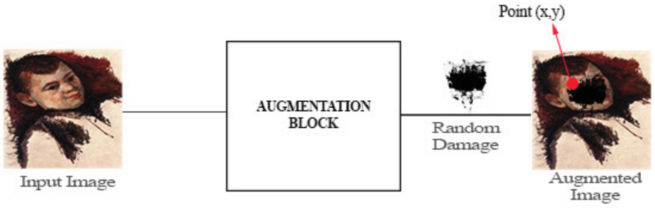


Fig. 1. An illustration of the data augmentation block. The randomly selected mask is scaled, rotated and pasted on top of the input image. The paste location is selected randomly in the centre area to avoid cropping of the damage in transforms.

III. Method

Our underlying approach consists of two major components: data augmentation and network architecture. The data augmentation component is very important and is used to augment the images in the dataset with damages to resemble damaged artworks. This is essential due to the unavailability of datasets specific to the problem that we are trying to solve. Secondly, we propose a novel generator architecture that takes the augmented image as an input and tries to generate an image similar to the original input image. This architecture is inspired by the intuitions behind ResNet [22], DenseNet [23] and the Stacked Hour Glass network [24]. Moreover, we use a standard discriminator architecture used by [15], which consists of convolutional layers, batch normalization [25], leaky ReLU [26] and a Sigmoid activation function to output the probabilities of whether the output is fake or real. The details about the components are addressed in the following sections.

A. *Data Augmentation*

We adapt a similar approach to that of Restoration of artwork using deep neural networks [2], and generate synthetically damaged images using a variety of damages i.e. scratches, blurs, cracks, halftones, paint, paper damages, rips, tears and water-colours [27], [28]. We have chosen a total of 47 different damages that we believe cover most of the types of damages that occur. To augment the input images with these damages, we randomly select a size from the set of damage sizes (100×100, 128×128, 156×156 and 200×200), which have been chosen to cover an area of approximately 40%-80% of the input image size (224×224×3). We randomly sample a point on the input image and augment the image by pasting the top left corner of the damage on that point to form the damaged input image. To ensure that the damage is not pasted at the edges of the input image, we only sample points from a central square region of the input image, which ensures that more than 60% of the damage is visible on the input image. Fig. 1 shows an example of how the input images are augmented.

B. *Generator*

The generator architecture that we have selected is inspired by existing works [22]–[24]. The process of artwork restoration is very complex and requires a very deep network to learn the mapping from damaged input images to restored images.

To facilitate the use of a deep generator, we make use of residual connections as proposed in [22] to form the blocks of our generator. Furthermore, the generator is inspired by [23] and each in decoder layer in the generator receives features from encoder layers with similar height and width, concatenates them and applies a 1x1 convolution to reduce the number of feature maps. This facilitates dense connectivity, very strong gradient flow during back-propagation, and is computationally efficient due to reduced number of parameters compared to the ResNet. Moreover, our generator network stacks generator blocks (referred as CResNetBlocks) and performs subsequent downsampling and upsampling in each block to obtain information at each scale, allowing the network to also understand damages at lower scales. This is similar to the technique proposed in Stacked Hour Glass network [24].

A single block of our generator network is called the CResNetBlock (Concatenated ResNet Block). As mentioned previously, our architecture is inspired by the approach used in DenseNets. However, [29] highlights that the excessive connections in DenseNets may not be entirely useful and could possibly decrease the computational efficiency, while making it prone to overfitting. To tackle this problem and to preserve the strong gradient flow of DenseNets, we use residual connections from the preceding CResNetBlock to facilitate a strong gradient flow, while significantly reducing the number of intra-block residual connections. This allows us to build a very deep network consisting of multiple encoder-decoder blocks, while keeping the width (number of feature maps) of each block and total number of trainable parameters limited. Fig. 2 shows how a single CResNetBlock functions.

In Fig. 2, D1, D2, and D3 denote downsampling layers and U1, U2, and U3 denote the upsampling layers. C-LR indicates Strided Convolution with Leaky-ReLU activation, BTL indicates bottlenecks, DC indicates de-convolution (Transpose Convolution) [30]. All downsampling layers receive connections from previous CResNetBlocks and are concatenated with these connections to form succeeding downsampling layers. The upsampling layers receive intra-block connections from the downsampling layers, and are formed by concatenating the downsampling layers with the output of the previous upsampling layer (except U1). U1, with these intermediate concatenated outputs (U2_CONCAT, U3_CONCAT) form inter-block connections to the next CResNetBlock. Each concatenation operation is followed by a 1x1 convolution to keep the number of features constant. Finally, the output of each block is concatenated and passed through a convolution layer with a tanh activation to produce the output of the generator. The application of tanh ensures that the input and output images' pixel values are in the same range. The overall structure of the generator network is shown in Fig. 4. We stack up to six CResNetBlocks to form the generator.

C. Discriminator

[31] highlight that L_1 and L_2 loss produce undesirable results as they incorporate coarseness in image generation. Moreover, [2] state that these losses fail to capture the crisp features from a high-resolution input image but perform efficiently on lower scales. Due to this reason, we make use of PatchGAN classifier [2], [32] which convolutionally computes losses between image patches of generated fakes and real inputs. By modelling the image as a Markov random field, it produces very good results and ensures lower parameter count, faster inference, and it does not depend on the scale of the input images. We make use of 1×1, 16×16, 70×70 and 224×224 sized patch discriminators that can be convolutionally applied to an image and the computed average across all patches is the final output of the discriminator. Based on the work of [2], the 1×1 discriminator emphasizes on colourfulness, the 16×16 discriminator focuses on spatial sharpness and the 70×70 discriminator ensures sharpness in both color and resolution. Fig. 3 demonstrates how a discriminator works on patches of the generated fakes and real images.

IV. Experiments

In this section we present the dataset that was created and used for training the model, our implementation and experimental procedure, the evaluation metrics, and finally both our qualitative and quantitative results.

A. ArtNet - Dataset

The process of artwork restoration cannot be subjective to one type of art or just an individual artist. The diversity in artworks is a challenge that artwork restoration techniques face due to the varying patterns in different styles of art. Some artwork may include patterns and styles that may be depicted as damage by a model that has only been trained on a single style of artwork. Due to this reason, we decided to combine artwork from three different artists and one Japanese art genre to achieve diversity in the dataset, whereas techniques generally make use of only a single dataset. We evaluate our method on a combination of four artwork datasets that forms our dataset called ArtNet. The dataset contains artwork from Paul Cezanne, Claude Monet, Vincent Van Gogh, and Ukiyo-e (a Japanese art form). The dimensions of each image in this dataset are constrained to 224×224 pixels. To feed the images into our model, we normalize the tensor in the range of [-1, 1] so as to match the output of the generator (which has a *tanh* activation function).

B. Implementation Details

The network that we have proposed does not use any pre-trained networks as backbones. We use 2429 images as training data from the combined ArtNet dataset. We perform validation on 130 images and our test set contains 842 images which are used to calculate the metrics. The network takes an input of size $224\times224\times6$ as we concatenate the augmented input image and a random noise tensor. The random noise tensor ensures that the generator does not produce deterministic outputs. We implement subsequent downsampling and upsampling in the CResNetBlocks with LeakyReLU in the downsampling part and ReLU in the upsampling part of the block. Several modifications can be made to the existing network. The activation functions

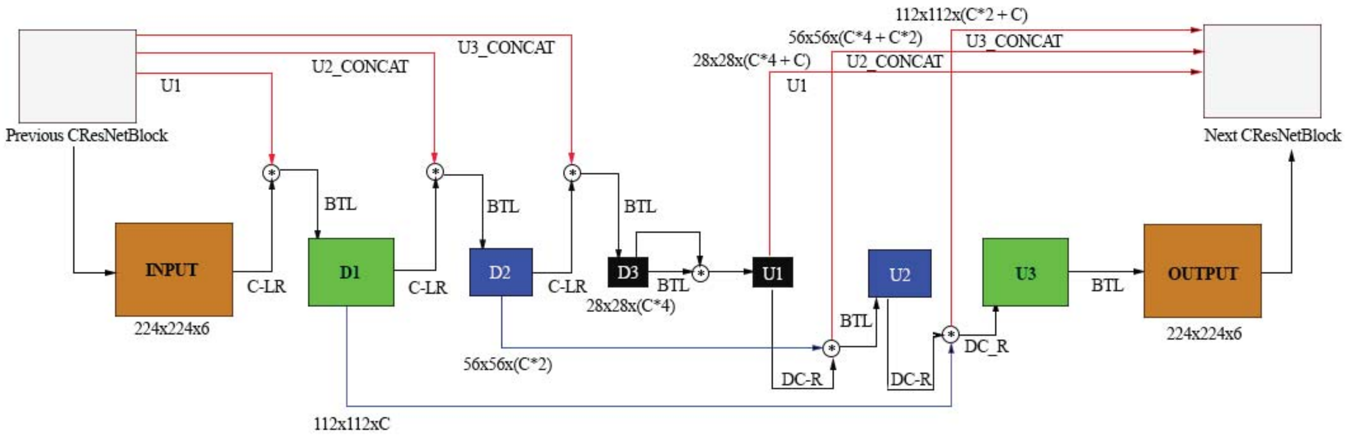


Fig. 2. The structure of one block of the generator (referred to as CResNetBlock). The input is downsampled (D1, D2 and D3) using strided convolutions accompanied by leaky ReLU activation (C-LR). Also, the results from the previous block are concatenated. Then it is upsampled (U1, U2 and U3) using transpose convolution with ReLU activation (DC-R). The outputs of the upsampling are sent to the next block as well. We use bottlenecks (BTL) to limit the number of channels from increasing.

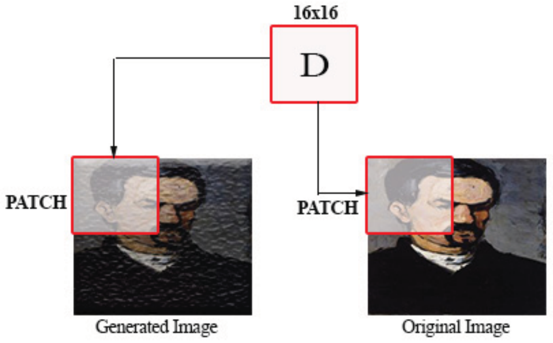


Fig. 3. An illustration of the 16×16 patch discriminator. The discriminator takes patches from both the images and tries to classify them as fake or real. Using smaller patches makes the network to predict more crisp results.

used in the CResNetBlocks can be replaced by ReLU and CReLU. Moreover, it is also possible to add regular convolution layers before the strided convolutions to enrich intermediate features. For data augmentation, we augment the damage on the image only at points that are sampled from a central square region of the image. We do this by taking a square with length reduced by a quarter from both ends of the original input image width.

We use a composition of the Adversarial loss and L_1 loss for the generator and binary cross-entropy loss for the discriminator to accurately classify reals and fakes. The L_1 loss has a weighting coefficient, which is set to 1 for the first 100 epochs and set to 100 afterward. We use the Adam optimizer to update the weights with beta values β_1=0.5 and β_2=0.999, with a learning rate of 0.001 for the first 100 epochs and 0.0002 for the next 100 epochs. Moreover, we use a patch size of 16 for the discriminator. The number of features for the discriminator and generator is 64. During the whole training and evaluation process, we used a batch size of 16. Our model was trained on one Nvidia RTX 2080Ti, with a training time of approximately 9 hours.

C. Metrics

We present below the metrics that we used throughout our experiments.

1) Mean Square Loss - MSE: We define MSE in Equation 1, with Y being the vector of the observed values, and $\hat{Y}$ the predicted values.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2 \tag{1}$$

2) Structural Similarity Loss - SSIM: We define SSIM in Equation 2, where μ_x, μ_x are the average intensity in respective windows, σ_x, σ_y are the variance of the intensity in the windows and σ_{xy} is the covariance between the two windows. We set $c_1 = (0.01)^2$, $c_2 = (0.03)^2$ and use window size = 11. The range of SSIM values extends between the range [-1, 1] and only equals 1 if the two images are identical.

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \tag{2}$$

3) Fréchet Inception Distance - FID: Fréchet Inception Distance (FID) [33] is a common metric used to assess the quality of generators in a GAN architecture. It is a comparison between the distributions of generated images and the real images. Two image sets (real and generated) are passed into an Inception V3 [34] network and instead of comparing the final outputs, we used the mean and standard deviation of one of the deep intermediate layers for comparison. Note that lower FID values translates to better image quality and diversity. We define FID in Equation 3, where $\mathcal{N}(\mu, \sigma)$ and $\mathcal{N}(\mu_w, \sigma_w)$

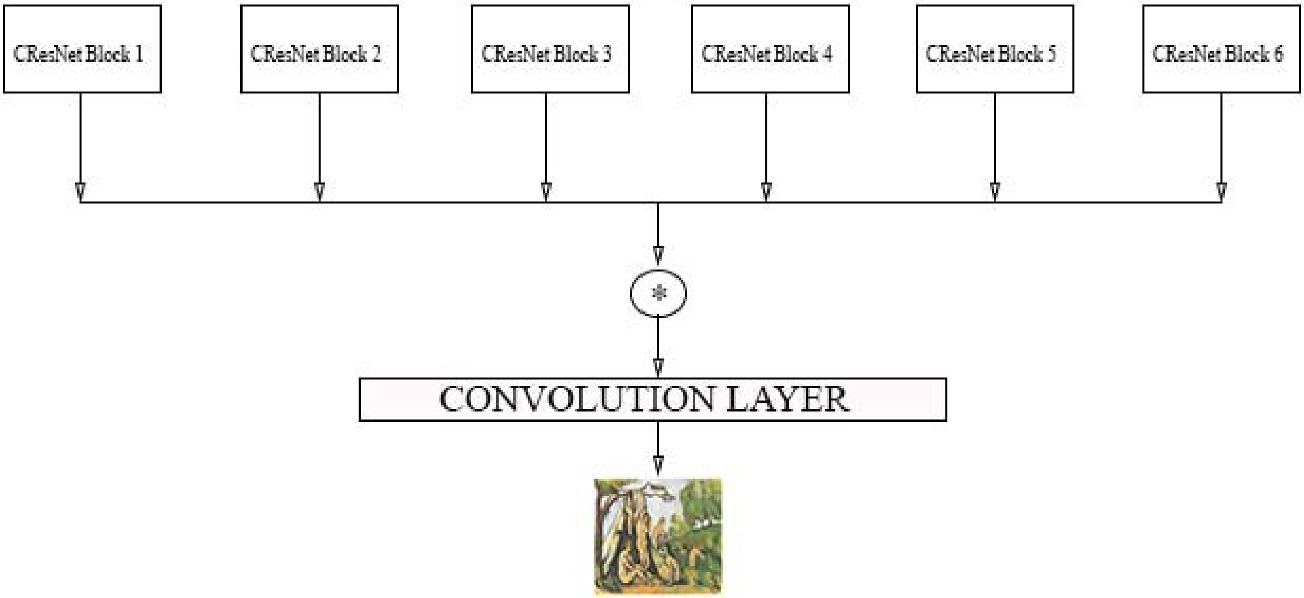


Fig. 4. The overall architecture of the generator. A number of generator blocks are stacked on top of each other and output from each block are concatenated (* operation). This is passed through a convolution layer to produce the final image. The detailed structure of CResNetBlock is given in Fig. 2.

TABLE I
QUANTITATIVE RESULTS ON THE TEST SET. UPWARD [↑] (OR DOWNWARD [↓]) ARROW INDICATES THAT A HIGHER (OR LOWER) SCORE RESULTS IN A GOOD IMAGE RESTORATION.

Metric	Score
SSIM ↑	0.795
MSE ↓	0.014
FID ↓	131.91

represent the distributions of the real and generated images respectively.

$$FID = |\mu - \mu_w|^2 + tr\left(\Sigma + \Sigma_w - 2(\Sigma\Sigma_w)^{1/2}\right) \quad (3)$$

D. Evaluation

Although there is no universally accepted metric used to evaluate the performance of GANs, previous works in image inpainting domain report Structural Similarity Index Measure (SSIM) and Mean Square Error (MSE) as quantitative measures. [35] introduced SSIM in 2004 as a method for predicting image quality given a reference. It compares luminescence, contrast, and structure information between the two images using mean, variance, and covariance and hence, expected to give more realistic performance measure than MSE. Recently, [36] introduced Frechet Inception Distance (FID), a very efficient evaluation metric for GANs. It measures the distance between curves or probability distributions. The intuition behind this metric is not entirely useful for our technique because it also involves measuring the diversity in the generated data, whereas we only emphasize on the fidelity and quality of the generated image, based on the input image. Furthermore, a shortcoming of FID is that it may not capture all the features. Table I show the scores we achieve on our test set. It's worth mentioning that all our metrics (SSIM, MSE, FID) indicate that our solution produced images of good quality with almost optimal restoration in most cases. Our SSIM score (with SSIM ranging between [-1, 1]) indicates that the original and restored versions are almost identical. For the FID score, given the difficulty of the task, our score is satisfactory, and even surpassing works that performed a similar image restoration task using a less complex and diverse dataset [37]. Our results are visually comparable to what [2] achieved for their dataset. As there is no benchmark dataset for image inpainting, it is difficult to quantitatively compare with previous works. They would not produce fair results since our dataset is unique, and to the best of our knowledge, no previous work has addressed the classical artwork restoration problem with a diverse dataset such as ours.

E. Results

Quantitative results for artwork restoration techniques are limited as there are no benchmark datasets on which the

TABLE II
QUALITATIVE RESULTS FOR DIFFERENT ARTWORK RESTORATIONS. OUR MODEL IS ABLE TO PREDICT THE MISSING PARTS REASONABLY WELL IN MOST CASES. IN THE THIRD AND FOURTH ROW, WE CAN SEE THAT THE MODEL LEARNS SOME CONTEXT FROM THE IMAGE AS IT PREDICTS OBJECT BOUNDARIES PRETTY WELL. HOWEVER, REPRODUCING COMPLEX STRUCTURES FROM SCRATCH IS MUCH MORE DIFFICULT. HENCE, THE RIGHT EYE IN THE FIRST ROW AND THE FACES IN THE LAST ROW ARE LOST.

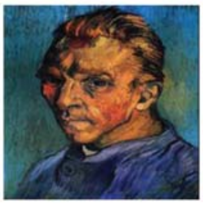
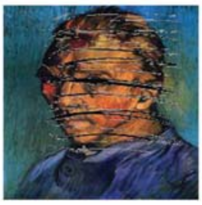
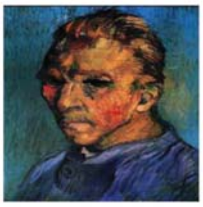
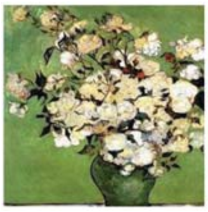
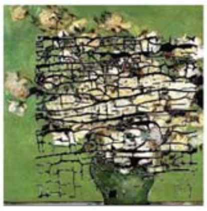
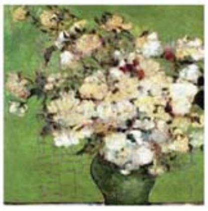
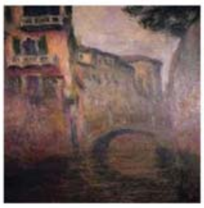
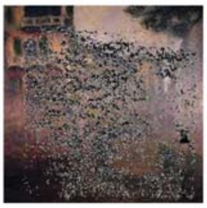
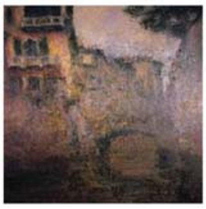
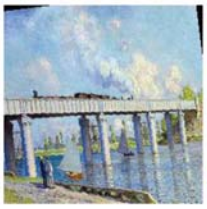
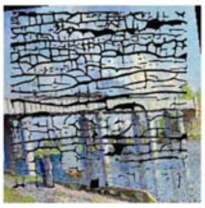
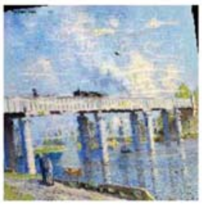
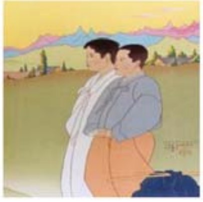
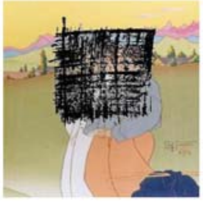
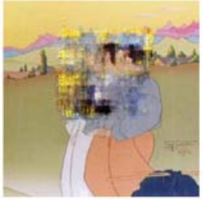
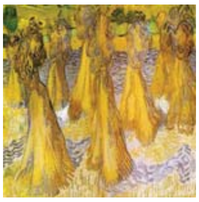
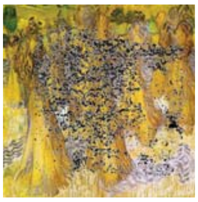
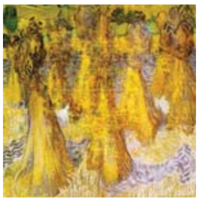
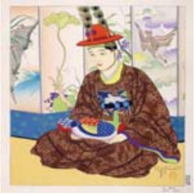
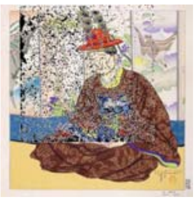
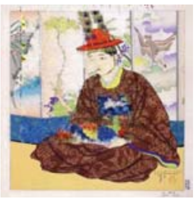
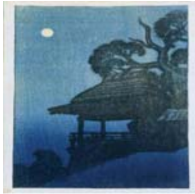
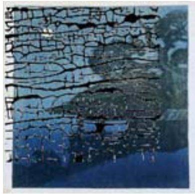
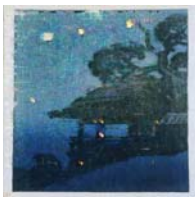
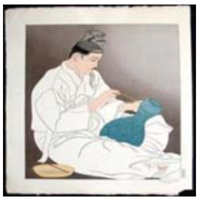
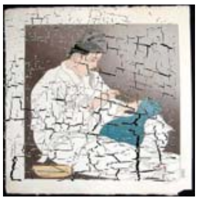
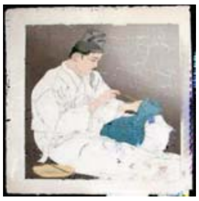
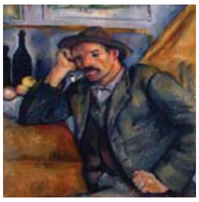
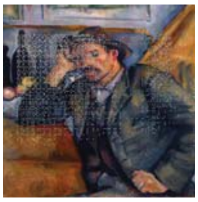
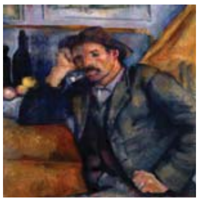
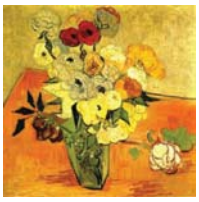
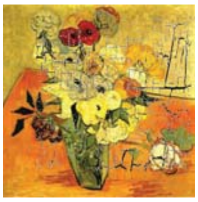
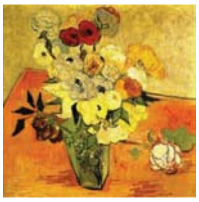
Original	Masked	Reconstructed
		
		
		
		
		

TABLE III
ADDITIONAL QUALITATIVE RESULTS OF OUR MODEL.

Original	Masked	Reconstructed
		
		
		
		
		
		

existing techniques can be evaluated. Due to this reason, [2] also adapts an innovative approach of evaluating their results through domain experts. [3] makes use of SSIM, L_2 Loss, and PSNR, however, they only augment the data with fades and blurs. On the contrary, instead of fading or blurring an image, we augment a part of the image with a damage and hence, makes it a difficult task as the neighbouring pixels in the image provide no information about the replaced region. Table II highlights some of the results of our work. It can be seen in the first row that our model is able to efficiently recover the damaged image by removing the scratches from the face. Similarly, in the 2nd and the 3rd row, our model can remove cracks and halftones from the damaged image while also reconstructing building structures in the background of the image in the 3rd row, although the background is fairly occluded by the damage. Moreover, it is interesting to observe that our technique is able to correctly recover the bridge and the boat below the bridge even though it is barely visible. This means that the model not only detects and removes the damage but also understands the context of the painting and fills the damaged region pixels with appropriate values. However, the model fails to give good results when the image is severely damaged, as seen in the last row of the table. The technique fails to capture enough information to be able to reconstruct the image, although it manages to somewhat capture the head structure. Additional results are displayed in Table III.

V. Discussion and Conclusion

We have developed a generative adversarial approach that addresses the problem of artwork restoration involving artwork that has been damaged over time. Although relevant measures are considered, this is a problem faced by many, such as art galleries and museums that house old artwork. Our method achieves convincing results, by not only removing damages from the artwork but correctly approximating the region that was damaged, just like a domain expert. This approach is not limited to artwork restoration and can be applied to several other tasks like super-resolution in distorted low-resolution images, and can be extended to make use of style transfer to not only remove damages, but to also observe novel art images with a different artistic style. We are also considering to expand the ArtNet dataset, with the help of specific domain experts, to include extra artworks from different artistic styles and therefore place ArtNet to the standard dataset for artwork restoration and image style transfer.

VI. Acknowledgements

We would like to thank the ANU College of Engineering & Computer Science for the GPU Cluster (Nvidia RTX 2080Ti graphics cards), as well as Dr. Dylan Campbell for the useful comments and feedback throughout this work.

References

[1] G. Liu, F. A. Reda, K. J. Shih, T. Wang, A. Tao, and B. Catanzaro, "Image inpainting for irregular holes using partial convolutions," ECCV, vol. 11215, pp. 89–105, 2018. [Online]. Available: https://doi.org/10.1007/978-3-030-01252-6_6

[2] V. Gupta, N. Sambyal, A. Sharma, and P. Kumar, "Restoration of artwork using deep neural networks," Evolving Systems, pp. 1–8, 10 2019. [Online]. Available: https://doi.org/10.1007/s12530-019-09303-7

[3] Y. Zeng, J. C. A. van der Lubbe, and M. Loog, "Multi-scale convolutional neural network for pixel-wise reconstruction of van goghś drawings," Mach. Vis. Appl., vol. 30, no. 7-8, pp. 1229–1241, 2019. [Online]. Available: https://doi.org/10.1007/s00138-019-01047-3

[4] A. C. Telea, "An image inpainting technique based on the fast marching method," J. Graphics, GPU, & Game Tools, vol. 9, no. 1, pp. 23–34, 2004. [Online]. Available: https://doi.org/10.1080/10867651.2004.10487596

[5] A. Levin, A. Zomet, and Y. Weiss, "Learning how to inpaint from global image statistics," ICCV, pp. 305–312, 2003. [Online]. Available: https://doi.org/10.1109/ICCV.2003.1238360

[6] C. Ballester, M. Bertalmío, V. Caselles, G. Sapiro, and J. Verdera, "Filling-in by joint interpolation of vector fields and gray levels," IEEE Trans. Image Process., vol. 10, no. 8, pp. 1200–1211, 2001. [Online]. Available: https://doi.org/10.1109/83.935036

[7] J. Xie, L. Xu, and E. Chen, "Image denoising and inpainting with deep neural networks," NIPS'12, vol. 25, pp. 350–358, 2012. [Online]. Available: https://proceedings.neurips.cc/paper/2012/hash/6cdd60ea0045eb7a6ec44c54d29ed402-Abstract.html

[8] L. Xu, J. S. J. Ren, C. Liu, and J. Jia, "Deep convolutional neural network for image deconvolution," NIPS'14, vol. 27, pp. 1790–1798, 2014. [Online]. Available: https://proceedings.neurips.cc/paper/2014/hash/1c1d4df596d01da60385f0bb17a4a9e0-Abstract.html

[9] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. C. Courville, and Y. Bengio, "Generative adversarial nets," NIPS'14, vol. 27, pp. 2672–2680, 2014. [Online]. Available: https://proceedings.neurips.cc/paper/2014/hash/5ca3e9b122f61f8f06494c97b1afccf3-Abstract.html

[10] D. Pathak, P. Krähenbühl, J. Donahue, T. Darrell, and A. A. Efros, "Context encoders: Feature learning by inpainting," CVPR, pp. 2536–2544, 2016. [Online]. Available: https://doi.org/10.1109/CVPR.2016.278

[11] S. Iizuka, E. Simo-Serra, and H. Ishikawa, "Globally and locally consistent image completion," ACM Trans. Graph., vol. 36, no. 4, pp. 107:1–107:14, 2017. [Online]. Available: https://doi.org/10.1145/3072959.3073659

[12] J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu, and T. S. Huang, "Generative image inpainting with contextual attention," CVPR, pp. 5505–5514, 2018. [Online]. Available: http://openaccess.thecvf.com/content_cvpr_2018/html/Yu_Generative_Image_Inpainting_CVPR_2018_paper.html

[13] Y. Song, C. Yang, Z. Lin, H. Li, Q. Huang, and C. J. Kuo, "Image inpainting using multi-scale feature image translation," CoRR, vol. abs/1711.08590, 2017. [Online]. Available: http://arxiv.org/abs/1711.08590

[14] M. Sagong, Y. Shin, S. Kim, S. Park, and S. Ko, "PEPSI : Fast image inpainting with parallel decoding network," CVPR, pp. 11 360–11 368, 2019. [Online]. Available: http://openaccess.thecvf.com/content_CVPR_2019/html/Sagong_PEPSI__Fast_Image_Inpainting_With_Parallel_Decoding_Network_CVPR_2019_paper.html

[15] P. Isola, J. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," CVPR, pp. 5967–5976, 2017. [Online]. Available: https://doi.org/10.1109/CVPR.2017.632

[16] K. Nazeri, E. Ng, T. Joseph, F. Z. Qureshi, and M. Ebrahimi, "Edgeconnect: Generative image inpainting with adversarial edge learning," CoRR, vol. abs/1901.00212, 2019. [Online]. Available: http://arxiv.org/abs/1901.00212

[17] J. Yu, Z. Lin, J. Yang, X. Shen, X. Lu, and T. S. Huang, "Free-form image inpainting with gated convolution," ICCV, pp. 4470–4479, 2019. [Online]. Available: https://doi.org/10.1109/ICCV.2019.00457

[18] F. Stanco, S. Battiato, and G. Gallo, Digital Imaging for Cultural Heritage Preservation: Analysis, Restoration, and Reconstruction of Ancient Artworks, 1st ed. USA: CRC Press, Inc., 2011.

[19] K. He, G. Gkioxari, P. Dollár, and R. B. Girshick, "Mask r-cnn," ICCV, pp. 2980–2988, 2017. [Online]. Available: https://doi.org/10.1109/ICCV.2017.322

[20] S. Ren, K. He, R. B. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," NIPS'15, vol. 28, pp. 91–99, 2015. [Online]. Available: https://proceedings.neurips.cc/paper/2015/hash/14bfa6bb14875e45bba028a21ed38046-Abstract.html

[21] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," MICCAI, vol. 9351, pp. 234–241, 2015. [Online]. Available: https://doi.org/10.1007/978-3-319-24574-4_28

[22] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," CVPR, pp. 770–778, 2016. [Online]. Available: https://doi.org/10.1109/CVPR.2016.90

[23] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," CVPR, pp. 2261–2269, 2017. [Online]. Available: https://doi.org/10.1109/CVPR.2017.243

[24] A. Newell, K. Yang, and J. Deng, "Stacked hourglass networks for human pose estimation," ECCV, vol. 9912, pp. 483–499, 2016. [Online]. Available: https://doi.org/10.1007/978-3-319-46484-8_29

[25] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," ICML, vol. 37, pp. 448–456, 2015. [Online]. Available: http://proceedings.mlr.press/v37/ioffe15.html

[26] B. Xu, N. Wang, T. Chen, and M. Li, "Empirical evaluation of rectified activations in convolutional network," CoRR, vol. abs/1505.00853, 2015. [Online]. Available: http://arxiv.org/abs/1505.00853

[27] Zolee and Zolee, "20 grunge scratch overlay texture (png transparent)," Aug 2017. [Online]. Available: https://www.onlygfx.com/20-grunge-scratch-overlay-texture-png-transparent

[28] "Enjoy these torn images for free," Sep 2020. [Online]. Available: https://www.freepik.com/free-photos-vectors/torn

[29] L. Zhu, R. Deng, M. Maire, Z. Deng, G. Mori, and P. Tan, "Sparsely aggregated convolutional networks," ECCV, vol. 12, pp. 192–208, 2018.

[30] V. Dumoulin and F. Visin, "A guide to convolution arithmetic for deep learning," CoRR, vol. abs/1603.07285, 2016. [Online]. Available: http://arxiv.org/abs/1603.07285

[31] A. B. L. Larsen, S. K. Sønderby, H. Larochelle, and O. Winther, "Autoencoding beyond pixels using a learned similarity metric," ICML, vol. 48, pp. 1558–1566, 2016. [Online]. Available: http://proceedings.mlr.press/v48/larsen16.html

[32] C. Li and M. Wand, "Precomputed real-time texture synthesis with markovian generative adversarial networks," ECCV, vol. 9907, pp. 702–716, 2016. [Online]. Available: https://doi.org/10.1007/978-3-319-46487-9_43

[33] A. Mathiasen and F. Hvilshøj, "Fast fréchet inception distance," CoRR, vol. abs/2009.14075, 2020. [Online]. Available: https://arxiv.org/abs/2009.14075

[34] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," CVPR, pp. 2818–2826, 2016. [Online]. Available: https://doi.org/10.1109/CVPR.2016.308

[35] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," IEEE Trans. Image Process., vol. 13, no. 4, pp. 600–612, 2004. [Online]. Available: https://doi.org/10.1109/TIP.2003.819861

[36] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "Gans trained by a two time-scale update rule converge to a local nash equilibrium," NIPS'17, vol. 30, pp. 6626–6637, 2017. [Online]. Available: https://proceedings.neurips.cc/paper/2017/hash/8a1d694707eb0fefe65871369074926d-Abstract.html

[37] Z. Wan, B. Zhang, D. Chen, P. Zhang, D. Chen, J. Liao, and F. Wen, "Old photo restoration via deep latent space translation," CoRR, vol. abs/2009.07047, 2020. [Online]. Available: https://arxiv.org/abs/2009.07047