

---

# Applications of Generative Image Models

**Akshat Sharma**  
ashar312@asu.edu

**Aradhita Sharma**  
ashar314@asu.edu

**Apoorva Uplap**  
auplap@asu.edu

## Abstract

This paper describes the applications of generative models to generate new images. Generative models are a class of statistical models which involve using a model to generate new examples that plausibly come from an existing distribution of samples. Our project focuses mainly on three applications of generative image models that are 1) image restoration, which is performed to restore the old deteriorated image 2) image upscaling, which is performed to enlarge the restored image 3) neural style transfer, which is performed to give an artistic touch to the images.

## 1 Introduction

Generative models have lately made news for applications like creating fake images and swapping faces in celebrity images, but this application possesses a serious social challenge to discriminate between real and fake images. Generative models are created from unsupervised learning (analyze structure with unlabeled data), and once the structure is learnt, a new set of data can be created that does not exist. The two popular kinds of generative models are Generative Adversarial Network (GAN) and Variational Autoencoder (VAE). This project focuses on explaining three key applications of generative modeling for images :

1. Image Restoration : Restoring old degraded photos by using variational autoencoders.
2. Image Upscaling : Enlarging and enhancing a small image using enhanced super resolution GAN.
3. Neural Style Transfer : Generates a digital image which adopts the style of a different image.

## 2 Implementation

We implemented the project to test the state of art generative networks for image restoration, image upscaling using super-resolution and then using style transfer to give artistic touch to the images. Further details for each of the methods above is provided now :

### 2.1 Microsoft Image Restoration

Research has shown that looking back at old photos triggers the feelings of happiness and love emotions, and reminiscing the special moments turns out to be more relaxing than

meditating [5]. But old printed photos before the digital era are deteriorating because of aging and improper handling. There are methods to restore these photos using image processing, but it is difficult to construct a signal dependent noise filtering model or a generalized image restoration model. Hence, an attempt is made to restore these old images using deep learning approach to construct a generalized model.

### 2.1.1 Previous image restoration methods

Image degradation is classified into structured and unstructured degradation. Blur, camera misfocus, color fading, noise and low resolution are examples of unstructured degradation, while patches, holes, marks and scratches are examples of structured degradation, which is more challenging to deal with than the former. Image denoising, deblurring, and local smoothness can be used to repair both organized and unstructured degradation in old images. However, there is little concern for repairing color fading or poor resolution issues, and thus restored photos appear outdated.

Real old photos are a mixture of unknown degradation (which can be any combination of structured or unstructured degradation), which is difficult to be characterized accurately, making it more difficult to create a degradation model which can realistically render the old image artifact. Hence, there is a need to construct a degradation model, which includes real degraded images and synthetic generated data.

### 2.1.2 Latent space translation

Previous deep learning methods used supervised learning which did not give good results for real old photos because the degradation model consisted of synthetic generated data of degraded photos, which were nowhere similar to the real old photos. As a result of this, a domain gap is created between real old photos and the photos synthesized for training the model. To reduce this domain gap, triplet domain translation is used to bridge between domains of real old photos ( $R$ ), synthetic photos ( $X$ ) constructed for training, and ground truth domain ( $Y$ ) consisting of images without degradation.  $r, x, y$  are denoted as images where  $r \in R, x \in X$  and  $y \in Y$  domains and mapping is done to corresponding latent spaces through

$$E_R: R \rightarrow Z_R, E_X: X \rightarrow Z_X, E_Y: Y \rightarrow Z_Y$$

The latent space of synthetic photos and real old photos are oriented in the shared domain such that  $Z_R, Z_X$  are close to each other ( $Z_R \approx Z_X$ ) using variational autoencoders (VAEs). This shared latent space is used for performing image restoration. After latent space translation, real old photos “ $r$ ” can be restored by sequentially performing the mappings,

$$r_{R \rightarrow Y} = G_Y \circ T_Z \circ E_R(r)$$

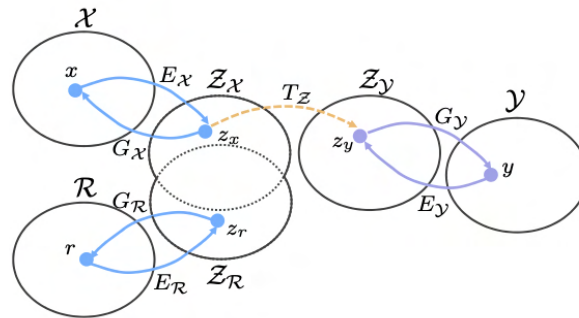


Figure 1 : Latent translation method for three domains [6]

### 2.1.3 Variational Autoencoders (VAEs)

Autoencoders take high dimensional input data, compress it by passing it through encoder to create a smaller representation (less dimension than the input) known as a "(bottleneck) latent space representation" and is given as input to decoders to reconstruct the high dimensional data. This reconstructed data and input data are compared to obtain the error function, and an iterative optimisation method is used for training the neural network in order to generate compressed output images. But this case produces unrealistic images when expected variation because of the discontinuities in its latent space representation.

The variational autoencoder has a continuous latent space representation, because in this case, the encoded distribution is regularized during training. Hence, VAEs can generate new data with different variations. In VAEs, rather than a single point, the input is encoded as a distribution (near to a standard normal distribution) throughout the latent space. Here, the bottleneck vector is replaced by two separate independent vectors representing standard deviation ( $\sigma$ ) and mean ( $\mu$ ) of the encoded distribution. Samples from this distribution (sampled latent vector) are fed to the decoder network, which acts as a generator. Difference between the learned distribution and standard normal distribution is the loss function which is minimized along with the reconstruction loss function. In order to train this network, reparameterization is performed which includes a fixed space standard distribution  $\epsilon$  which is randomly sampled and not learnt during back propagation, whereas mean and standard deviations are updated in each iteration. The update is denoted as  $z = \mu + \sigma \odot \epsilon$ .

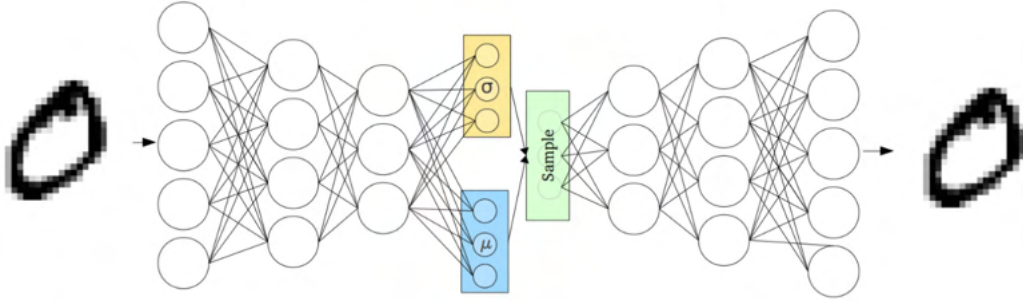


Figure 2 : Variational Autoencoder network diagram [7]

### 2.1.4 Image restoration through latent mapping

The given figure describes the architecture of the proposed network. Here, two VAEs are used to generate variational data for two cases. One VAE consists of old photos "R" and synthetic photos "X" where  $E_{R,X}$  is the encoder and  $G_{R,X}$  is the generator, which share old photos and synthetic photos such that both the degraded images can be mapped into this shared latent space. The other VAE is used for the output image "Y" where the encoder and generator are  $E_Y$ ,  $G_Y$ . VAEs are used because it learns the mapping of old photos and synthetic photos and generalizes well to real photos by reducing the domain gap. Afterwards, image restoration is performed for the synthetic pair  $\{X,Y\}$  using mapping  $T$  which include "ResBlocks" and "Partial Nonlocal Blocks". Nonlocal blocks deal with structured degradation (patches, holes, scratches) and ResBlocks deal with unstructured degradation (blurr, color fading, noise, low resolution). The combination of these blocks

enhances the capability of latent space.. Given the latent space  $Z_R \approx Z_X$ , the generator  $G_Y$  always generates a completely clean image without degradation.

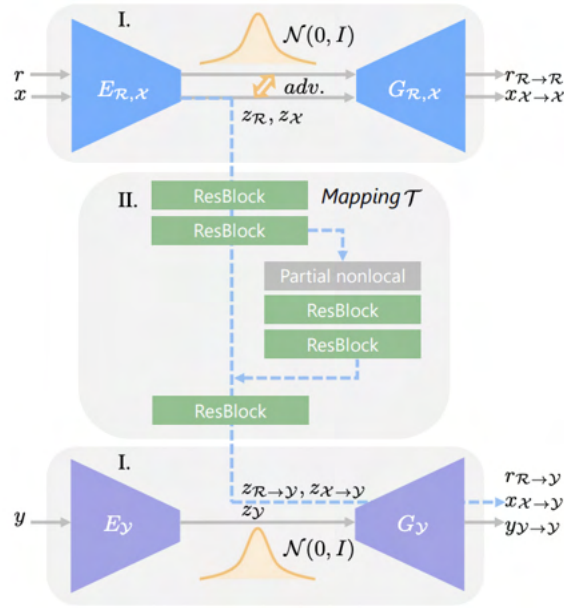


Figure 3 : Image restoration network [6]

### 2.1.5 Face refinement network

It is assumed that the old photos reminisce about the special moments, which include the faces of the loved ones. When generating synthetic images, sometimes unwanted textures are observed on generated faces. Therefore, a face refinement network is included to retrieve fine details of faces present in the old photos in the latent space “z”. As a result, the perceptual quality of the faces is greatly enhanced.

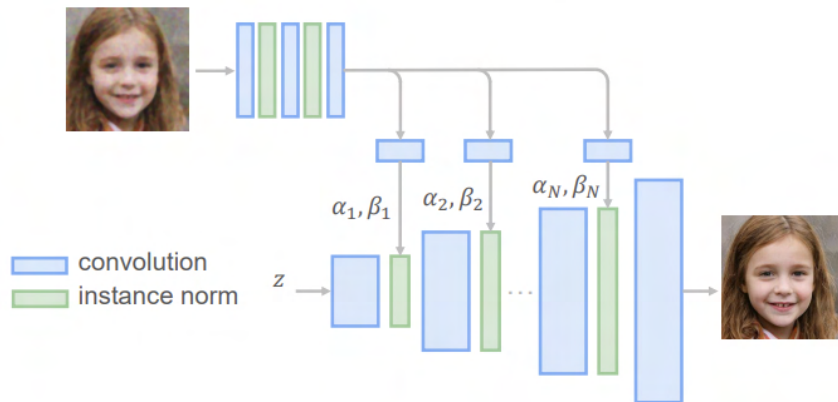


Figure 4 : Face refinement network to enhance the quality of faces [8]

## 2.2 Image Up-Scaling

In many CSI movies, there's that scene where someone finds a small and obscured image, and they get a clear picture out of it by zooming and enhancing it. Is this really possible? Mostly no, those movies are nowhere near technically accurate. But, to some extent, yes. It is indeed possible to enlarge and enhance images. The process of upscaling and enhancing an image is called super-resolution.

### 2.2.1 Initial Ideas for Image Up-Scaling (Using SR-CNNs)

In information theory, there's a concept called data processing inequality. It states that whatever way you process data, we cannot add information that is not already there. This implies that missing data cannot be recovered by further processing. Does that mean super-resolution is theoretically impossible? Not if we have an additional source of information.

A neural network can learn to hallucinate details based on some prior information it collects from a large set of images. The details added to an image this way would still not violate the data processing inequality. Because the information is there, somewhere in the training set, even if it's not in the input image. First, we can create a dataset by collecting high-resolution images and downscaling them, or we can simply use one of the existing super-resolution datasets, such as the DIV2K dataset. Then, we can build a convolutional neural network that would input only the low-resolution images, and we can train it to produce higher resolution images that match the original ones the best. As shown in figure 2, The SRCNN[1] paper simply minimized the squared difference between the pixel values to produce images that are as close as possible to the original high-resolution images.

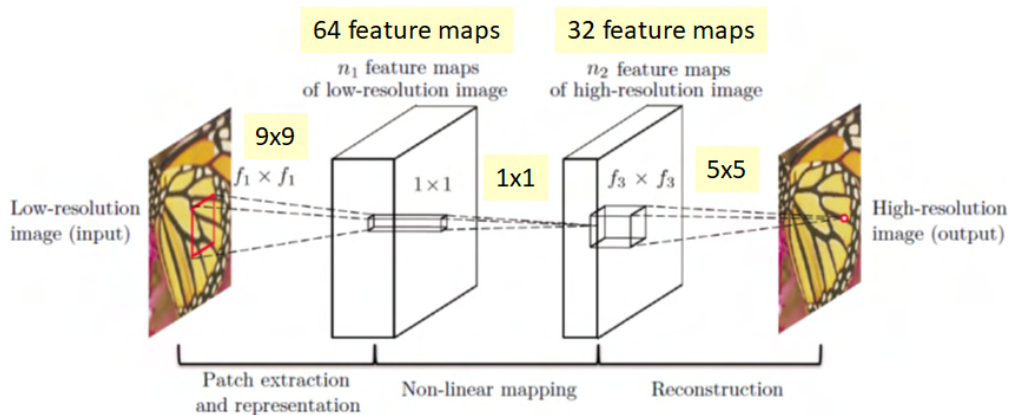


Figure 5 : Super Resolution-CNN

Before understanding super-resolution using GANs it will be good to know more about how Generative Adversarial Networks work in general.

### 2.2.2 Generative Adversarial Networks (General Idea)

In GANs [2], we have two neural networks, one is Generator and other is a Discriminator. The generator tries to generate a high resolution image and the

discriminator tries to determine whether or not it's real or not. Imagine that there is a counterfeiter and he wants to create an image that looks identical to the real image but obviously it's fake, so he takes it over to a pawn shop to try to get some money for it. The store owner then tries to critique that artwork to determine whether or not it's real. This is exactly how GANs work. The counterfeiter in this case is the generator and the critic is the discriminator. We feed in low resolution images to the generator and it creates a high resolution image or the artwork and then our discriminator tries to tell if it is fake or real. As it can be seen in figure-1, there are two models (both are neural networks). Generator receives the input  $Z$  (which can be a low res image) and then outputs  $\hat{X}$ .  $\hat{X}$  is fed to the discriminator network where it calculates the distance between  $\hat{X}$  and  $X$  where  $X$  is the real high-res image, thus regarding  $\hat{X}$  as fake or real.

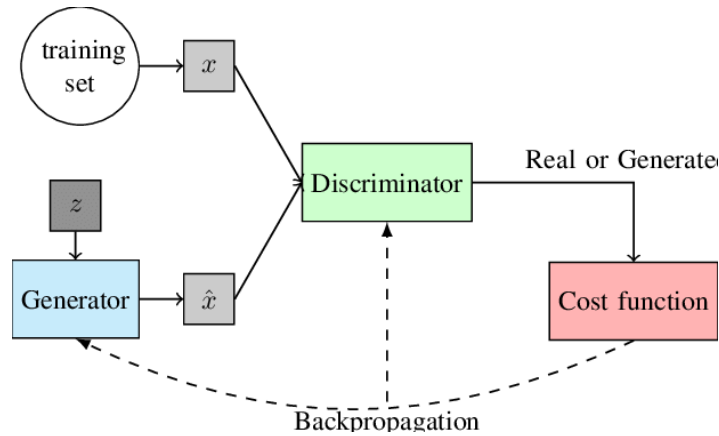


Figure 6 : Generative Adversarial Networks

Loss is such that the generator is incentivized to generate  $\hat{X}$  such that  $\hat{X}=X$  and the discriminator is incentivized to be able to differentiate between  $\hat{X}$  and  $X$ . Theoretically, the generator will become so good that it will be able to generate  $\hat{X}$  such that it is the same as  $X$  and the discriminator will say  $\hat{X}$  is real every time.

### 2.2.3 Super Resolution-GAN

There's a GAN-based super-resolution system called SRGAN [3]. It uses a generator network that inputs low-resolution images and tries to produce their high-resolution versions. It also uses a discriminator network that tries to tell whether this is a real high-resolution image or an image upscaled by the generator. Both networks are trained simultaneously, and they both get better over time. Once the training is done, all we need is the generator part to upscale low-resolution images. In addition to this adversarial training setup, SRGAN also used a VGG-based loss function.



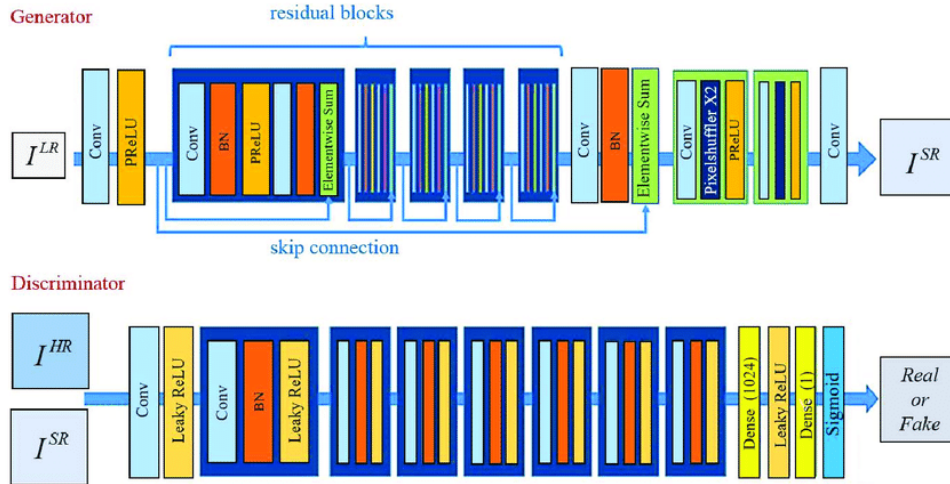


Figure 7 : SRGAN architecture

### 2.2.3 Enhanced Super Resolution-GAN

There's another paper called Enhanced SRGAN [4], which proposes a few tricks to improve the results further. Enhanced-SRGAN, or ESRGAN for short, somehow got popular in the gaming community. It was used for upscaling vintage games, and it worked pretty well. It's surprising how well it worked on video game graphics despite being trained only on natural images.

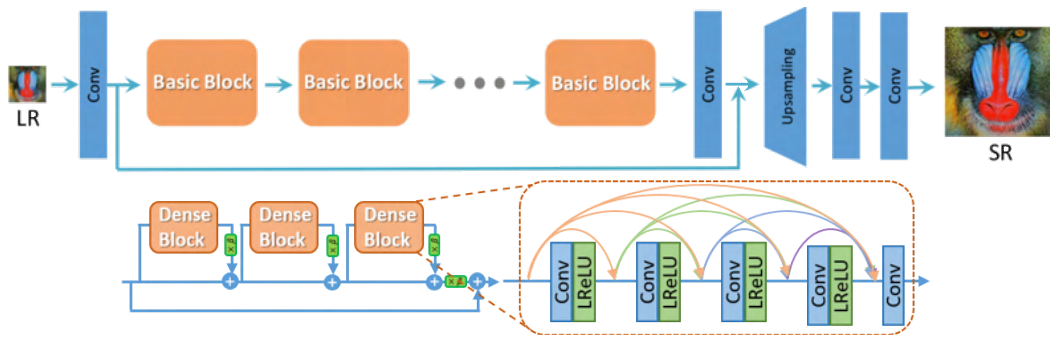


Figure 8 : ESR-GAN architecture

One of the enhancements made was the removal of batch normalization layers in their network architecture. Batch normalization does help a lot for many computer vision tasks. But for image-processing related tasks, such as super-resolution or image restoration in general, batch normalization can create some artifacts. Researchers also added more layers and connections to this model architecture. It's not surprising that a more sophisticated model resulted in better images, but deeper models can be trickier to train, especially if they are not using batch normalization layers. So, the authors of ESRGAN used some tricks like residual scaling to stabilize the training of such a network. In addition to the changes in the model architecture, they also modified the loss functions. We have used ESR-GAN for implementation of super-resolution.

## 2.3 Neural Style Transfer

In this section, we take an artistic image (style) such as a Van Gogh painting or a psychedelic image and capture the features from it. The style is then applied to a seemingly normal photograph (content) and we can visualize the artistic results. The motivation to obtain such a style transfer image is to imagine how a person would be painted by Van Gogh or for purely artistic/curiosity purposes.

### 2.3.1 Fast Style Transfer with TF-Hub

The model available in TF-Hub was built by the team at Google Brain [9], which was trained on the ImageNet dataset [10] for content images and the Kaggle Painter by Numbers dataset [11] along with the Describable Textures Dataset [12] for style images. The models consist of two networks, one for style prediction and another for style transfer. The Style Prediction Network is loosely based on the Inception-v3 architecture [13] which predicts and embedding vector  $\bar{s}$  which is the input for the Style transfer network along with the content image. The Style Transfer Network largely follows [14]. The objective for style transfer model is to minimize:

$$\mathcal{L}_c(x, c) + \lambda s \mathcal{L}_s(x, s)$$

$\mathcal{L}_c$  is the content loss and  $\mathcal{L}_s$  is the style loss while  $\lambda s$  is a lagrangian multiplier that weights the relative strength of the style loss. The content and style losses are defined as

$$\mathcal{L}_c = \sum_{j \in C} \frac{1}{n_j} \left\| f_j(x) - f_j(c) \right\|_2^2$$
$$\mathcal{L}_s = \sum_{i \in S} \frac{1}{n_i} \left\| G[f_i(x)] - G[f_i(s)] \right\|_F^2$$

where  $f_l(x)$  are the network activations in  $l^{\text{th}}$  layer,  $n_l$  is the number of units in  $l^{\text{th}}$  layer and  $G[f_l(x)]$  is the square and symmetric gram matrix that measures the spatially averaged correlation structure across the filters for the  $l^{\text{th}}$  layer activations.

### 2.3.2 Style Transfer with VGG19

Here, we implemented style transfer using the pretrained VGG-19 [15] network. First we load the content image and test the VGG19 network to check whether the correct label is predicted by the image classification model. We then load the VGG19 network without the classification head, take the intermediate layers of it and use them to represent the content and style images which is equivalent to the latent space representation of generative networks. We can do this as somewhere between the model before the classification label is predicted, the model acts as a feature extractor. By using the intermediate layers we describe the content and style of the input images. The content of an image is given by the intermediate feature map values, the style of the image by the means and correlations across various feature maps. After building the model for content and style tensor extractor, we run gradient descent with Adam optimizer by setting style and content weights. We also regularize the high frequency terms of the image which is also called total variation loss which is basically an edge detector. We use the inbuilt function for the total variation loss in TensorFlow for this.



### 3 Results









#### 3.1 Image Restoration

In this project, we give the old degraded photo as the initial input and the system removes the unstructured degradation and gives the clean restored image as output. In order to restore the structured degraded images, we need to specify to the system that the image contains scratches such that the system deals with both the unstructured and structured degradation and gives a clean output image.

##### 3.1.1 Restoration results for unstructured degraded images

It is seen in the resulting images that color of image is restored, noise and blur is removed, face is enhanced, and we obtain a clean image.







**Table 1 : Results of image restoration for unstructured degradation**

Degraded Image	Restored Image	Degraded Image	Restored Image
			
			

##### 3.1.2 Restoration results for structured degraded images (including scratches)

It is seen in the resulting images that scratches and rough patches present in the degraded image are fixed, color is restored, face is enhanced, and we obtain a clean image.

**Table 2 : Results of image restoration for structured degradation**

Degraded Image	Restored Image
	
	
	

### 3.2 Image Upscaling using ESR-GAN

We took the restored image and passed it through the ESR-GAN generator. The results were as follows -





Figure 9 : Low Resolution Image input to ESRGAN



Figure 10 : High Resolution Output from ESRGAN

### 3.3 Style Transfer

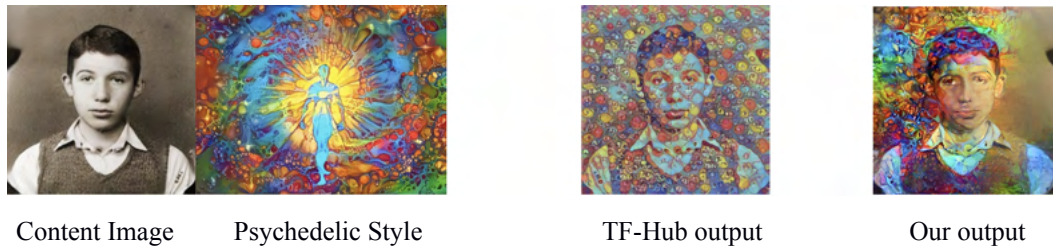


Figure 11 : Style transfer on content image with TF-Hub and our model

We used the content-style image pairs as given in figure 12.

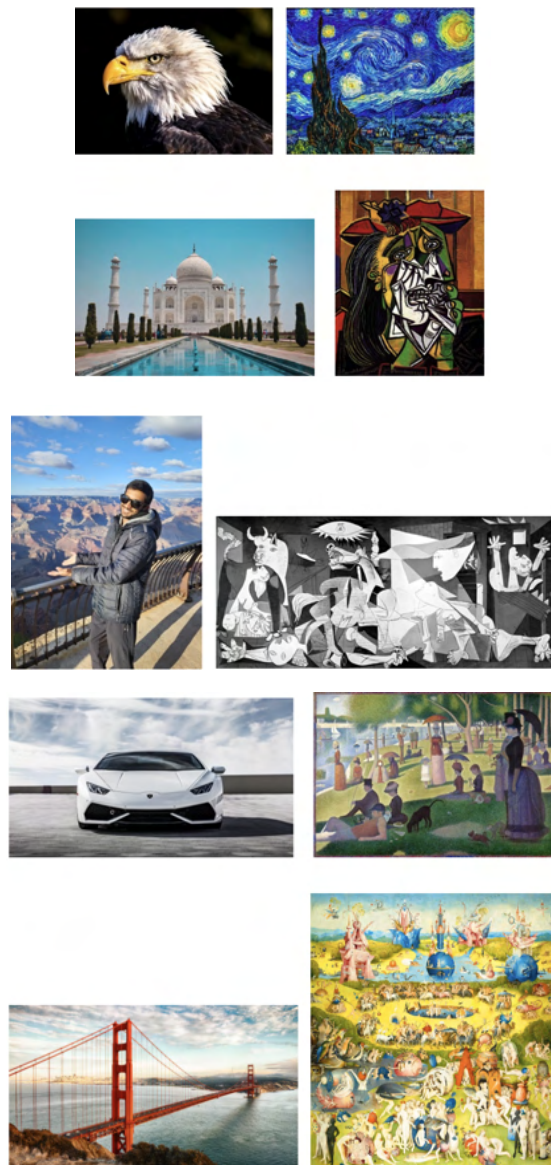


Figure 12 : Content-style image pairs



The content images in order are of an eagle, the Taj Mahal, a self-clicked photograph, a car and the Golden State Bridge obtained from [16].

The Style images in order are of a psychedelic poster and famous paintings such as the Starry Night by Van Gogh, the Guernica by Picasso, the Weeping Woman by Picasso, A Sunday Afternoon on the Island of La Grande Jatte by Georges Seurat and the Garden of Earthly Delights by Heironymous Bosch obtained from [16].

We can see the gradual stylization of each content image by printing it at different epochs. The following results showcase the progress of stylization:

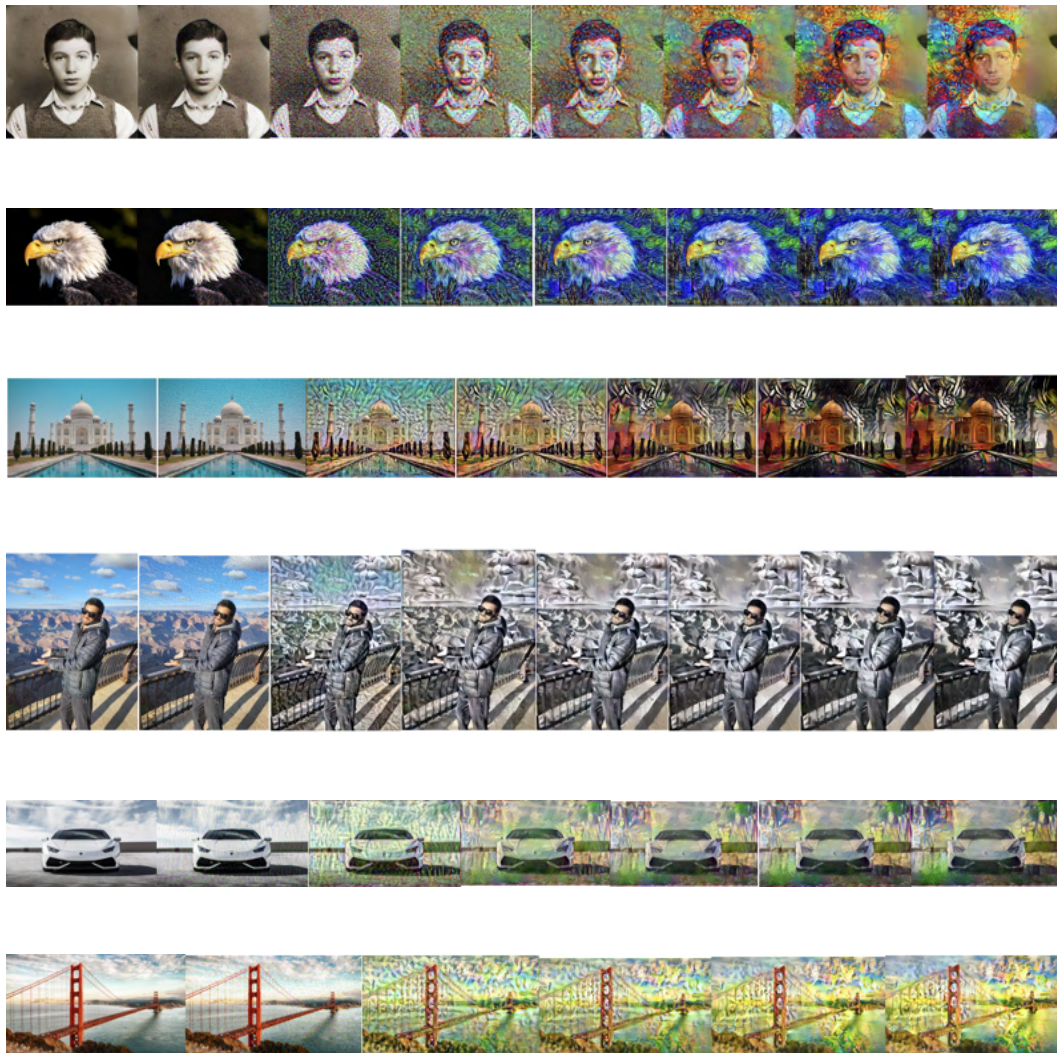


Figure 13 : Gradual stylization of each content image at different epochs

## 4 Experiments

We conducted some experiments to test our model and find some empirical properties of the generative models that we have used for image restoration, image up-scaling and style transfer.

### 4.1 Image Restoration

Apart from the old images available in Microsoft's dataset, we tested this model with our old printed images to observe the model's efficiency. It is observed in figure 14 that faces present in the images are enhanced, color is fixed however, scratches, and unwanted patches are not completely removed (as observed in the first restored image top right corner, and in the left center of the second restored image). Datasets including degraded images with such types of patches can help improve the model.

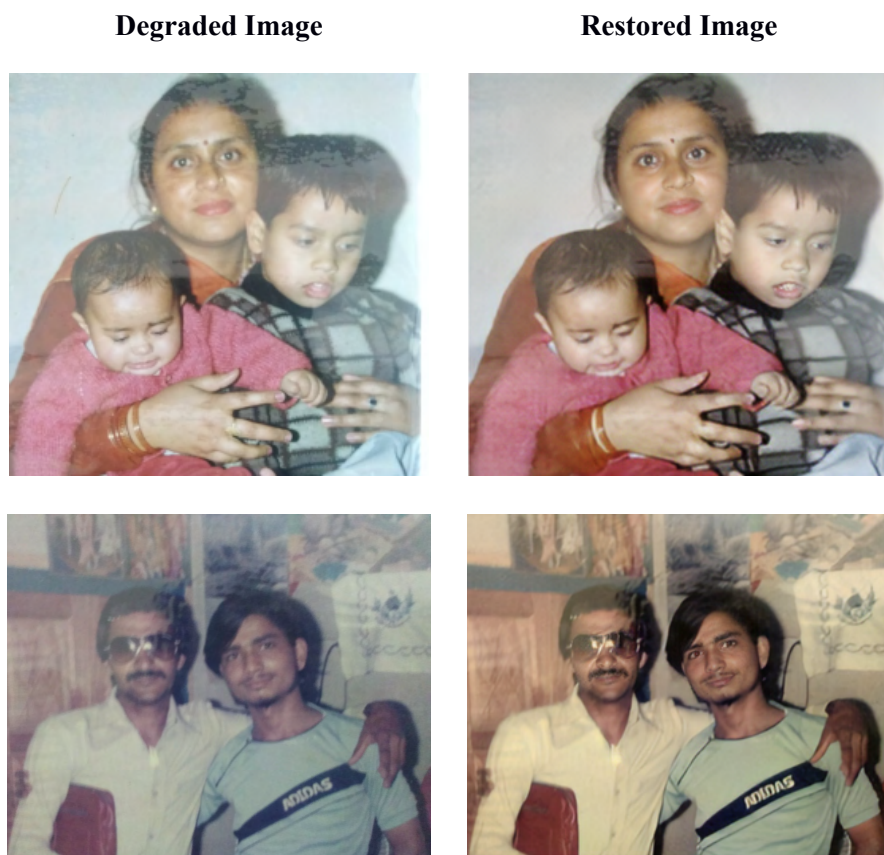


Figure 14 : Image restoration results for our old degraded photos

### 4.2 Image Up-Scaling

We conducted an experiment to see how the upscaled image compares to the original image. We fed the image as given in figure 15 as the input to the network. The upscaled image obtained is shown in figure 16.





Figure 15 : Input image given to the model



Figure 16 : Upscaled output image



If we take a close look at the face in figure 17, we have interesting observations -



Figure 17 : Side by side comparison of ESRGAN output and input

The ESRGAN applies some smoothening and paints in some of the details. We can clearly see that the teeth are not visible in the input image. The generative model has painted in the details (in this case teeth), which is undesirable. Also taking a closer look at the right ear, we can see that the model has failed to draw the ear properly. This can give us some insights about the validity of our model and what kind of training data should be used for further training the model. ESRGAN was trained for anime upscaling. Training the ESR-GAN model more on real life dataset rather than on anime dataset can help alleviate these problems.

### 4.3 Style Transfer

We conducted experiments to see how different content images are stylized using just one style image. Following are the content images used from self-photographs and screenshots [18] [19] while the style is Psychedelic as shown in figure 18. The style vs content loss for each is also plotted.



Figure 18 : Style image



Figure 19 : Psychedelic style transfer on different input images.

As we can see from using the same style for different content images, the features of the style towards the outer layers (bubbles) are applied to the outer parts of the content image in different areas. The style transfer is not uniform for all content images and it is not a simple superimposition of the content and style images. The style features are extracted and transferred in varying degrees and orientations according to the content image features.

## 5 Discussion and Conclusion

### 5.1 Image Restoration

From our study, it can be concluded that the use of VAEs have helped in reducing the domain gap to generate realistic clean images out of old degraded images. This model works efficiently for restoring old degraded images which consist of unstructured degradations. However, it is not efficient for some types of structured degradations (patches). We are using the Google Colab platform that entirely runs in the cloud and provides access to GPU. It is seen that CUDA runs out of memory when large file size images are given input to this model. Hence, this model can be improved to deal with large file size images and function well on local machines.

### 5.2 Image Up-Scaling

After the experiments, it can be concluded that although the results of ESR-GAN are pretty good, there is still some room for improvement, especially relating to the datasets on which the model is being trained. Also, since the training of GAN's is extremely hard, the issues stated in the experiments can also arise due to overfitting. To conclude, the results are pretty good right now and will only get better with more research.

### 5.3 Style Transfer

After experimenting with different content and style images, we conclude that style transfer is a very tool for artists, designers and the curious minds. Elmyr de Hory made millions by forging artwork and selling it to art dealers and museums. The skill that is needed by a forger can now be emulated by computers to capture the artistic style of a painting and apply it to different photographs. One can visualize the artwork of long lost painters and need not go to museums or galleries and can simply try out style transfer on their own images. This convenient tool can be included in the various filters and layouts that a modern smartphone camera or any social media platform has. We observed that style transfer can be best visualized for a content image that has contrast colors than the applied style. Also, sometimes only the predominant features in the art may be captured and the transfer of style may also be done only partially when the image and style have similar features.

## 6 References

- [1] Image Super-Resolution Using Deep Convolutional Networks, <https://arxiv.org/abs/1501.00092>
- [2] Generative Adversarial Networks, <https://arxiv.org/abs/1406.2661>
- [3] Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network, <https://arxiv.org/abs/1609.04802>
- [4] ESRGAN: Enhanced Super-Resolution Generative Adversarial Networks, <https://arxiv.org/abs/1809.00219>
- [5] Digitalcameraworld.com, "Its official looking at old photos is more relaxing than meditating", <https://www.digitalcameraworld.com/news/its-official-looking-at-old-photos-is-more-relaxing-than-meditating>
- [6] Wan, Ziyu & Zhang, Bo & Chen, Dongdong & Zhang, Pan & Chen, Dong & Liao, Jing & Wen, Fang. (2020). Bringing Old Photos Back to Life. 2744-2754. 10.1109/CVPR42600.2020.00282.

- [7] towardsdatascience.com, "Intuitively understanding variational autoencoders " <https://towardsdatascience.com/intuitively-understanding-variational-autoencoders-1bfe67eb5daf>
- [8] Z. Wan et al., "Old Photo Restoration via Deep Latent Space Translation," in IEEE Transactions on Pattern Analysis and Machine Intelligence, doi: 10.1109/TPAMI.2022.3163183.
- [9] Golnaz Ghiasi, Honglak Lee, Manjunath Kudlur, Vincent Dumoulin, Jonathon Shlens. [Exploring the structure of a real-time, arbitrary neural artistic stylization network](#). Proceedings of the British Machine Vision Conference (BMVC), 2017.
- [10] ImageNet dataset <https://www.image-net.org/>
- [11] Kaggle Painter by Numbers <https://www.kaggle.com/competitions/painter-by-numbers/data>
- [12] Describable Textures Dataset <https://www.robots.ox.ac.uk/~vgg/data/dtd/>
- [13] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision. IEEE Computer Vision and Pattern Recognition (CVPR), 2015.
- [14] V. Dumoulin, J. Shlens, and M. Kudlur. A learned representation for artistic style. International Conference of Learned Representations (ICLR), 2016.
- [15] Simonyan, Karen, and Andrew Zisserman. "Very deep convolutional networks for large-scale image recognition." arXiv preprint arXiv:1409.1556 (2014).
- [16] <https://www.freeimages.com/>
- [17] C. Irving. Fake: the story of Elmyr de Hory: the greatest art forger of our time. McGraw-Hill, 1969.
- [18] Eichiro Oda, TOEI, FUNimation Entertainment (Firm),. (2022). *One piece: Episode 1015*.
- [19] Hajime Isayama, Wit Studios, (Firm),. (2013). *Shingeki no Kyojin: Episode 12*.