

**Computer Vision Project**  
**Image Denoising and Colorizing using Condition Generative Adversarial Networks**  
**Mid Progress Report**

## 1. Introduction:

This project proposes to implement a Conditional Generative Adversarial Network [1] in order to perform two fundamental components of vintage image enhancement: image denoising and image colorization. Unlike previous attempts at image colorization and denoising using Convolutional Neural Networks (CNNs) and CNN based auto-encoders respectively, this project attempts to perform both tasks using a single CGAN. The motivation for this project comes from the need of enhancing thousands of old photographs and movies to make them eligible to be displayed on large screens without any significant noise and with accurately added colors.

## 2. Methodology:

### 2.1 Network Architecture:

As mentioned before, this project employs the use of a CGAN to perform noise reduction and image colorizing to old grayscale images. A Generative Adversarial Network comprises of two separate neural networks in which one is a discriminating network and the other one is a generating network. Discriminative models are the ones most often used in classification networks and their task is to output a label of class given an example input. A generator model performs the exact opposite task of a discriminating model and therefore takes noise as input and outputs an image. The network employed in this project is based on a Pix2Pix CGAN [2] popularly used for image-to-image translations.

Both generator and discriminator use modules of the form convolution-BatchNorm-ReLu [3]. The generator model is based on an encoder-decoder with skip connections as described in the U-Net architecture [4]. A defining feature of image-to-image translation problems is that they map a high resolution input grid to a high resolution output grid. In such a network, the input is passed through a series of layers that progressively downsample, until a bottleneck layer, at which point the process is reversed. Such a network requires that all information flow pass through all the layers, including the bottleneck [2]. However, to ensure that much of the low level detail is transferred to the output image with being downsampled by the network, skip connections are employed which enable transfer of information directly from input layers to output layers. This architecture forms the shape 'U' and is therefore called U-Net as shown in figure 1.

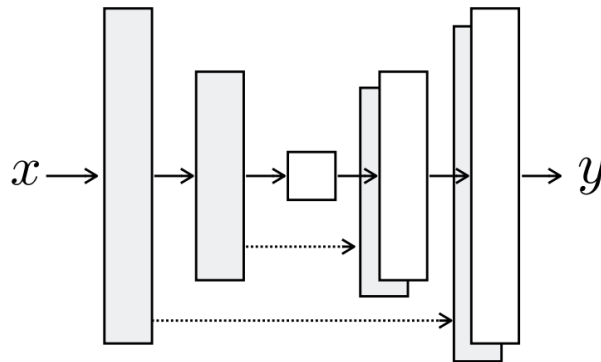


Figure 1: U-Net based Generator Architecture with Skip Connections, Source: [2]

In image-to-image translations, L1 and L2 loss is not applicable since such loss functions result in the production of blurry images. Nonetheless, L1 works very well for low frequency variations in images. In order to model high-frequencies, it is sufficient to restrict our attention to the structure in local image patches. Therefore, a PatchGAN [2] model is implemented for the discriminator architecture which only penalizes structure at the scale of patches. This discriminator tries to classify if each  $N \times N$  patch in an image is real or fake. We run this discriminator convolutionally across the image, averaging all responses to provide the ultimate output of D [2]. In this particular model, patches of  $70 \times 70$  are utilized.

## 2.2 Dataset Preparation:

In order to achieve good color accuracy, the training dataset domain was limited to images that contains humans in them like portraits, group photographs etc. to ensure that the network learns to colorize human skin tone exceptionally well. Nonetheless, the dataset also contains other images in minority to ensure that the network learn color schemes of grass, sky, wood and other background details. Such a dataset was not present anywhere due to which a custom dataset was designed to perform some tests on the network. 791 royalty free images, majority of which were human pictures, were downloaded and zipped into a file which was then uploaded to Google Drive for remote access. Using a custom written python script, the images were downsized to  $512 \times 512$ . Two directories were made on the drive, one for colored images, and the other one for noisy grayscale images. In order to produce grayscale noisy images, python was used along with OpenCV to grayscale the images and to add salt and pepper noise. Table 1 below shows some sample pairs of source and target images that were used for training the network.

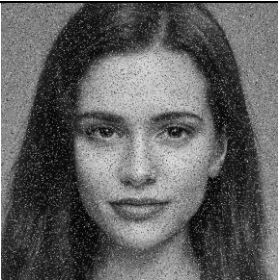





Source Image	Target Image
	
	
	

Table 1: Source to Target Image Dataset Preparation

### 3. Progress and Results:

After training the CGAN with patches of  $70 \times 70$  for the discriminator and U-Net architecture for generator for a total of 60 epochs, the results were obtained on the training set images as shown in table 2. As visible in the images in table 1, results obtained were exceptionally well for images that were a part of the training data. The colors are accurate and noise has been entirely removed. On close observation, the output image of the CGAN shows slight loss of sharpness on the edges. These results were expected since the network was actually trained on these images.









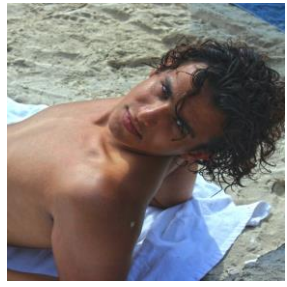
Input Image	Output Image	Ground Truth
		
		
		

Table 2: Results of the CGAN on Training Set Images

The real test of the network's accuracy would be to test the network on images that it has not seen before. The tests were performed with some popular images that were converted to grayscale and noise was added to them manually. One image, however, did not contain salt and pepper noise. This became a test of the network's ability to remove different noise types. The results are presented in table 3.

As visible, the network failed to assign accurate colors to validation images although some accuracy of skin tone is visible on close inspection. This might be due to the fact that the majority of the training set images were of human skin tones. In the first and third images, salt and pepper noise was added manually before the test, and as expected, the noise has been removed perfectly. In the second image however,



the noise was inherent to the image, and this noise is due to grainy nature of old films. Since the network couldn't identify this as noise, it did not attempt to remove it fully.

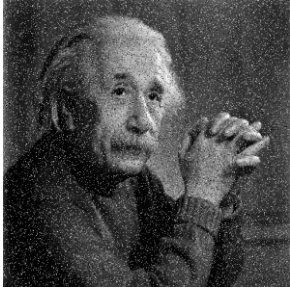
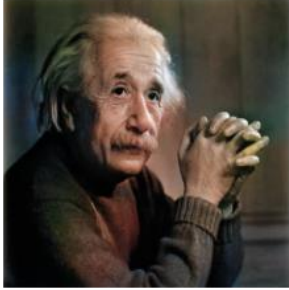
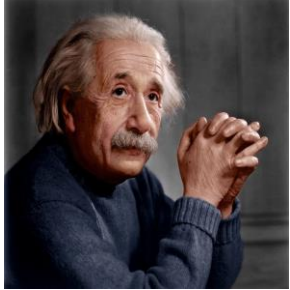






Input Image	Output Image	Ground Truth
		
		
		

Table 3: Results of the CGAN on Validation Set Images

#### 4. Future Plans:

Given the results obtained in table 3, following shortcomings and their respective solutions were identified:

- The network can only deal with salt and pepper noise since this is the kind of noise it was trained to remove from an image. To solve this issue, the dataset will be modified to include different kind of noises including Gaussian noise and Poisson noise.
- The network is unable to accurately assign colors. The obvious reason for this problem is the fact that the dataset on which it was trained was of only 791 images. The dataset size will have to be increased by a large factor to improve the color accuracy of the network.
- The output image of the generator network is of only  $256 \times 256$  dimensions, it needs to output at least  $512 \times 512$  images to improve discriminator loss as well as to enable better comparison with ground truth images.

- Apart from visual comparison, to quantify the effectiveness of the network, an image comparison metric will have to be decided to compare the output image with ground truth image.

## 5. References:

- [1] I. J. Goodfellow, J. Pouget-Abadie and M. Mirza, "Generative adversarial nets," in *27th International Conference on Neural Information Processing Systems*, Montreal, 2014.
- [2] P. Isola, J.-Y. Zhu, T. Zhou and A. A. Efros, "Image-to-Image Translation with Conditional Adversarial Networks," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.
- [3] S. Ioffe and C. Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift," in *ArXiv*, 2015.
- [4] O. Ronneberger, P. Fischer and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," in *MICCAI*, 2015.