



Recovering Old or Damaged Images using GAN

Harshith J L¹, Chandan Kumar M², Sanjana N B³, Dhanush B L⁴, K S Mahesh⁵

B E Student, Department of computer science and engineering, JSSSTU, Mysuru, India^{1, 2, 3, 4}

Assistant Professor, Department of computer science and engineering, JSSSTU, Mysuru, India⁵

Abstract: This paper discusses recovering old or damaged images using generative adversarial networks. Researchers are using generative adversarial networks as a general-purpose solution to image to image translation problems. We demonstrate that this approach is effective in restoring lost parts in an image and colorizing images, but these networks can be used to solve a wide range of image to image translation problems.

Keywords: Restoration, Colorization, Generative Adversarial Network, Generator, Discriminator, Deep learning.

I. INTRODUCTION

Image restoration and colorization is an essential computer vision problem. Recently there are increasing interests and significant progress in this area. The whole concept of automatically restoring and colorizing images is an interesting problem, but it never seemed to be done well, even with the existing deep learning models. The reason why this didn't work was because there was still a human involved in hand-coding a key step, which is the loss function. The most immediate way to evaluate if a neural network creates a good image is by comparing the pixels directly and penalizing according to how different they are. This encourages the neural network to be very conservative in its prediction. For example, green for grass, blue for skies and so on. Generative Adversarial Networks effectively replaces the hand coded loss functions with a network called discriminator. This enables the network to learn on its own and do the work appropriately [4].

Also, there has been the existence of various tools like Adobe Photoshop as a solution to this problem. This requires the user to have very good knowledge about the tool and the result would be totally dependent on the skills of the user. But our approach helps the user achieve rather more satisfying or equivalent results at the cost of just providing the image to be restored or colorized as an input to the model. There is no demand of individual skills here. Later, we will be deploying a web application where these models will be available as services and anyone with the knowledge of internet can access it.

II. RELATED WORK

1. Image to Image Translation by Conditional Adversarial Network by Philip Isola, Jun-Yan Zhu, Tinghui Zhou, Alexei A Efros [1]

The authors point out Generative adversarial networks as a general-purpose solution to image-to-image translation problems. They mention that adversarial networks not only learn the mapping from input image to output image, but also learn a loss function to accomplish this mapping. This makes it possible to use an equivalent generic approach to problems that traditionally would require very different loss formulations. They demonstrate that this approach is effective in generating photos from label maps, reconstructing objects from edge maps, and colorizing images, among other tasks.

2. Old Photo Restoration via Deep Latent Space Translation by Ziyu Wan, BO Zhang, Dongdong Chen, Pan Zhang, Dong Chen, Jing Liao, Fang Wen [2]

In this paper, the authors propose an approach restore old photos that is deteriorated from severe degradation using a deep learning approach. They suggest that downgrading in real photos is complicated and the domain gap between synthetic images and real old photos makes the network fail to generalize. Therefore, they proposed a novel triplet domain translation network by leveraging real images along with massive fabricated image pairs. Specifically, they train two variational auto encoders to respectively transform old images and clean images into two latent spaces. Then the translation between two latent spaces is learned with synthetic paired data. This translation generalizes well to real images because the domain gap is closed in the compact latent space. Besides, to address multiple degradations mixed



in one old image, they design a global branch with partial non-local block targeting the structured defects, such as scratches and dust spots, and a local branch targeting the unstructured defects, such as noise and the two branches are merged in the latent space, leading to improved capability to restore old photos from multiple defects.

III. PROPOSED WORK

We have developed two models namely, the restoration model and the colorization model.



Fig. 1 Restoration model

The restoration model is a deep learning model that takes the input damaged image and produces a restored image. We used a very small proportion of training images from the ImageNet dataset. Around 1000 training images of size 256 * 256 were used to train the model [3].

Training: The generator G is a U-Net. The discriminator D is a classifier. The network was trained for 75 epochs. The training set includes both masked image and the corresponding real image. The discriminator D was trained initially by freezing the generator G. Now the discriminator D can classify between real and fake image. The generator G is trained by freezing the discriminator D. The input to G is masked image and the generator G is trained until D fails to classify the images generated by G as fake. At this point G is able to generate fake images that looks similar to the ones in actual dataset. Hence training is stopped at this stage [4].

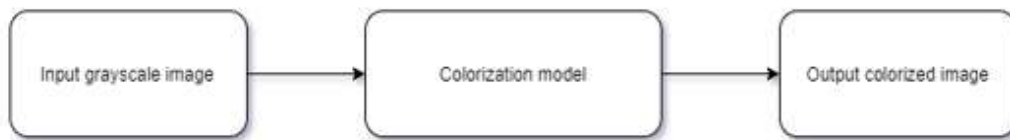


Fig. 2 Colorization model

The colorization Model is a deep learning model that takes a grey scale image as input and produces a coloured image. We used COCO dataset for training this model. Around 8000 images of size 256 * 256 were used for training.

Training: The generator model was pre-trained separately to avoid the blind leading the blind problem where neither the generator nor the discriminator knows anything about the task at the start of training. Pre-training is done in two stages. The backbone of the generator is pre-trained for classification and then the whole generator will be trained for colorization using L1 loss. To accomplish the first stage of training, we used pre-trained ResNet18 as backbone of our U-Net and for the second stage we train the U-Net on our training set with L1 loss only. After pre-training the U-Net for desired number of epochs, then we move to combined adversarial and L1 loss to complete the whole process of training. As a result, the output produced by this model is more satisfying than those produced by the model trained on normal training procedure.

Both the models have the following architecture for generator and discriminator [1].

The generator architecture

The encoder-decoder architecture is depicted as follows:

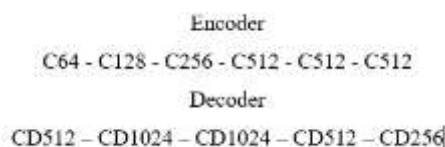


Fig. 3 Encoder-decoder architecture for generator



Each block in the encoder except the first block contains convolution layer followed by batch-normalization layer and RELU activation function. Batch-normalization is absent in the first C64 layer. The RELUs in the encoder are leaky with a slope of 0.2, while the RELUs in the decoder are not leaky. The U-Net architecture has a skip connection between layer i in the encoder and $n - i$ in the decoder, where n is the total number of layers. After the last decoder layer, a convolution is applied followed by Tanh activation function [7].

The discriminator architecture

The discriminator is a Patch-discriminator. This discriminator tries to classify each $n*n$ patch in an image as real or fake, where n is the size of the kernel. We try to run the discriminator convolutionally across the image averaging all the response to provide the ultimate output. The architecture is as follows,

C64 – C128 – C256 - C512 – C512 – C512

Fig. 4 Discriminator architecture

After the last layer convolution is applied followed by sigmoid activation function. Batch-Norm is not applied to C64 layer. RELUs are leaky with a slope of 0.2 [7].

IV. RESULTS

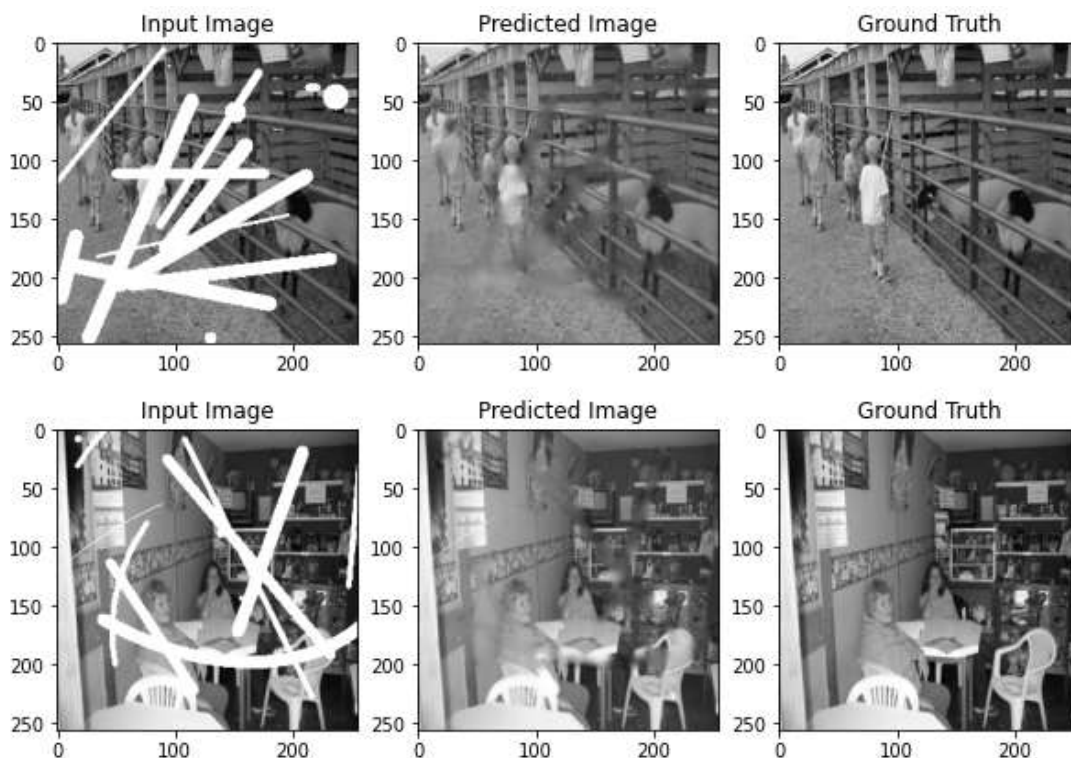
The models were evaluated using various metrics. Metrics like precision, recall, f1 score and Fréchet inception distance (FID) were used for evaluation.

We achieved a precision of 0.40 for the restoration model and a precision of 0.44 for the colorization model [6].

The recall score for restoration model is 0.40 and 0.44 for the colorization model [6].

The f1 score for restoration model is 0.40 and 0.44 for colorization model [6].

The FID score for restoration model is 0.866 and for colorization model is 0.036 [5].



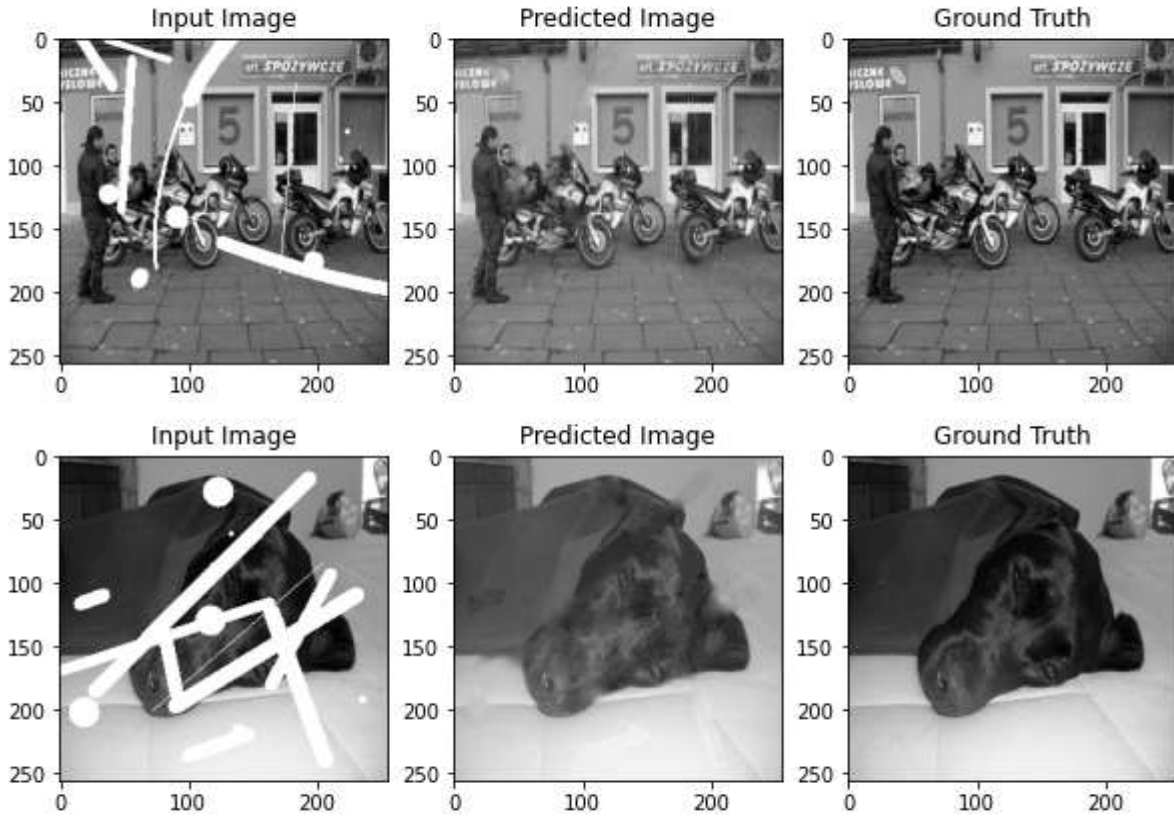


Fig. 5 Screenshots of sample outputs from restoration model

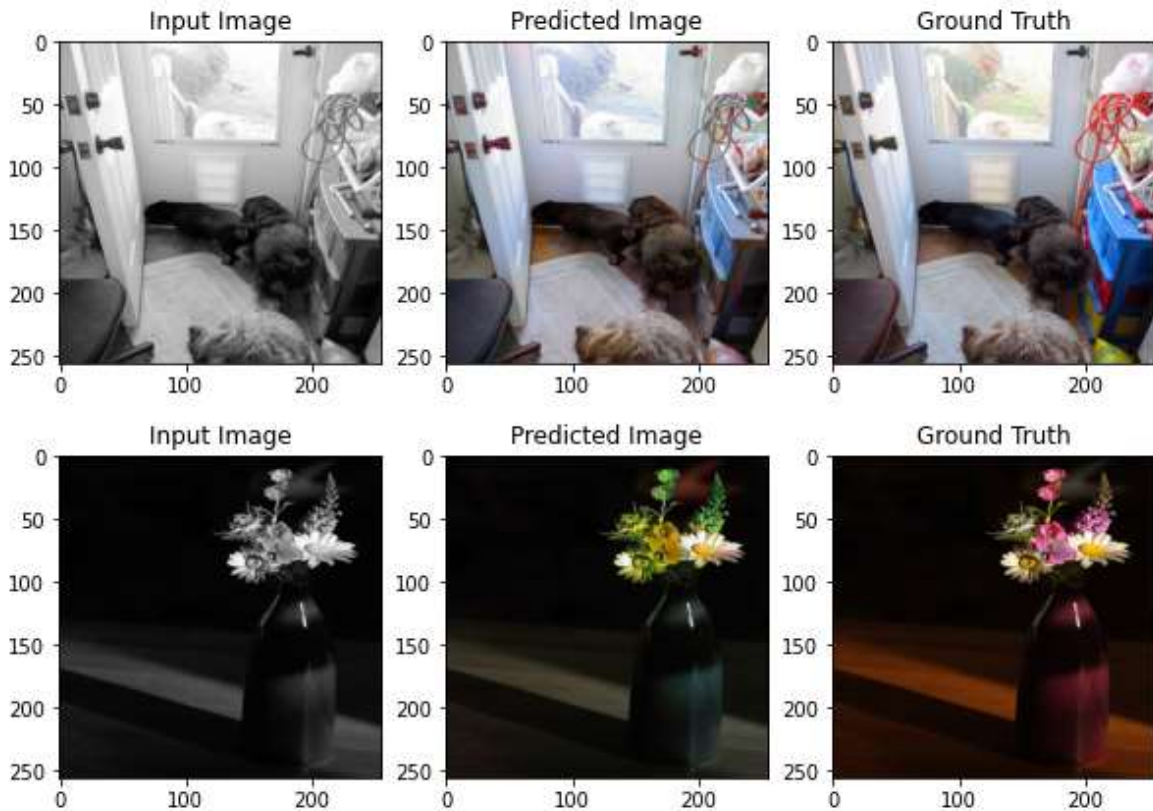




Fig. 6 Screenshots of sample outputs from colorization model.

V. CONCLUSION AND FUTURE WORK

In this paper we have demonstrated how generative adversarial networks outperforms conventional methods. Although, we have demonstrated only for image restoration and colorization, their application can be extended to solve a wide variety of image to image translation problems. The performance can be significantly improved upon training the models with a larger training set and for larger number of epochs.

We will be developing a web-based application to which we shall integrating these two models as services. This will enable the user to access our models on the web and the user will be able to recover images by uploading them on to our website.

ACKNOWLEDGMENT

We extend our gratitude to Dr. M P Pushpalatha, HOD of Dept. of Computer Science and Engineering, JSSSTU for providing an excellent environment for our education and its encouragement throughout our stay in college. We extend our heartfelt gratitude to our guide Prof K S Mahesh, Assistant Professor of Dept. of Computer Science and Engineering, JSSSTU who has supported us throughout our project with his patience and knowledge whilst allowing us the room to work in our own way.

REFERENCES

- [1] Image to image translation with conditional adversarial networks. <https://arxiv.org/pdf/1611.07004v1.pdf>
- [2] Old Photo Restoration via Deep Latent Space Translation. <https://arxiv.org/abs/2009.07047>
- [3] GAN implementation using keras. <https://machinelearningmastery.com/how-to-implement-pix2pix-gan-models-from-scratch-with-keras>
- [4] Generative adversarial networks by Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, Yoshua Bengio. <https://arxiv.org/abs/1406.2661>
- [5] How to implement FID to measure GAN performance. <https://machinelearningmastery.com/how-to-implement-the-frechet-inception-distance-fid-from-scratch/>
- [6] How to Measure GAN Performance - <https://jonathan-hui.medium.com/gan-how-to-measure-gan-performance-64b988c47732>
- [7] Tips for training Stable Generative adversarial networks - <https://machinelearningmastery.com/how-to-train-stable-generative-adversarial-networks/>