

Generative Adversarial Network (GANs) for Image Translation

Dr. Anala M R¹, Dr. Hemavathy R², Krishna Kulkarni³ and Samhitha A⁴

¹⁻⁴R V College of Engineering, Bangalore, India

Email: analamr@rvce.edu.in, hemavathyr@rvce.edu.in, krishnak.scs20@rvce.edu.in, samhithaa.sit20@rvce.edu.in

Abstract—Image to Image translation deals with converting images from one form of representation to another automatically, while keeping the original structure or semantics. For example, the conversion of images represented by semantic label maps to color images, or the conversion of edge maps to full images. Most image transformation tasks require significant human involvement and artistic skill. Colorization of black and white images, is currently done by hand, a process that is both time consuming and expensive. However, due to the multitudes of coloring schema that exist for a single black and white image, the automatic colorization of images is not a trivial problem, as the mapping is not one-to-one. This paper deals with image to image translation using Generative Adversarial Networks (GANs), a technique for image synthesis and prediction which is time and cost effective. This work proposes a comparison between the proposed model (GAN) and a CNN, both were made to run on one image translation problem. The chosen problem was grayscale to color translation and the dataset was a subset of the Imagenet dataset. It was observed that the GAN took around eight hours lesser than the CNN model and the quality of images generated by the GAN were much superior than the one generated by the CNN.

I. INTRODUCTION

In Computer Vision (CV) applications a single image can be represented in a multitude of ways, and there is a need to be able to use the information inherent in one form to transform it to its corresponding image in another format of representation. Image translation is technique to map an input image to an output image. For example, given a semantically labelled image, it should be possible to obtain a corresponding realistic image of the same by image translation. Most image transformation tasks require significant human involvement and artistic skill. Colorization of black and white images, is currently done by hand, a process that is both time consuming and expensive. However, due to the multitudes of coloring schema that exist for a single black and white image; the automatic colorization of images is not a trivial problem, as the mapping is not one-to-one.

The translation process is made complex due to the fact that the mapping between various image representations is rarely one-to-one, given that most problems involve the generation of photo-realistic images from sparse input representations such as semantic labels, edge maps etc. However, each translation problem is essentially the same, the prediction of a set of pixel corresponding to a given set. The goal is to develop a single generic architecture that can perform this function, regardless of the specific format of the input and output images.

The term image translation covers a wide array of problems under the domains of computer vision and computer graphics. These are the problems that can be best described as the conversion of an image from one representation to another corresponding representation, such as conversion of an image with semantic labels to

realistic images. Previously implemented machine learning solutions to image translation have been problem specific, involving a hand-engineered loss. No common framework exists that can solve any image translation problem. Also, proposed solutions to image prediction problems largely use CNNs, which also involves human effort to design effective losses. Thus, most frameworks, even if general enough to address all image classes, require precise hand-tuning of loss functions in order to produce realistic output. A major impediment in using deep learning method is, due to lack of dataset for solving real world practical application. GANs can find solution to this problem, as it has the capacity to generate artificial data. The images generated are also of high quality and are photo-realistic. In GAN dataset training process is carried on two networks namely generator and discriminator. GANs networks are able to learn which loss function to minimize and how to minimize it.

II. LITERATURE

Without the use of a GAN, it is near impossible to address all image translation problems simultaneously. However, many authors have investigated separately a solution to each individual image-image translation problem as described in the following passages. The following section presents a detailed discussion of the presentsituation as well as the recent developments in solving similar, but narrower, problems.

The Fully Convolutional Networks (FCNs)[1]architecture is trained pixel to pixel to perform semantic segmentation. Pixel-wise predictions are used in order to assign each pixel to a class representing the enclosing object or region. Global information is needed to resolve 'what it is' while local information resolves the question of where. The skip architecture is used for obtaining deep, coarse, fine, semanticand appearance information. The issue of edge detection is addressed [2], using holistically-nested edge detection. The model performs image-to-image prediction, resolves ambiguity in edge and boundary detection by learning rich hierarchical representations, which allow the model to make decisions as human to resolve the same ambiguity. The problem is approached as an image to image problem, and the output is not explicitly modeled.

The semantic style transfer [3] claims that neural style transfer techniques often lead to unpredictable results. They find two major flaws in existing CNN architectures. When CNNs are applied for labeling, it can be reused to extract features like strokes, textures and grainbut not designed or trained for acceptable synthesis. The GAN architecture was first conceived by the authors of [4]. Until this point, discriminative models were very successful in deep learning. These models typically map a high-dimensional input to a class label. However, deep generative models were more difficult to work with as training them involved complex probabilistic computations, for example those that would arise in maximum likelihood estimation applied to generative networks. In a GAN, the generative model is trained against an opponent, the discriminator, which learns to differentiate a sample as either model distribution or data distribution. Hence it approximates the probability that a sample came from.

The effect of conditioning the input of a GAN is examined in[5]. In a normal GAN, the user has very little control over the actual output generated, or the modes of data generated. The model conditioning with additional information guides the output towards the desiredform. The conditioning can be done on class labels or on some part of data. The various architectural constraints on GANs were imposed [6]. These constraints were found to make the GAN more stable during training. They called their class of architectures Deep Convolutional GANs (DCGAN). However, even the DCGANs are prone to some forms of model instability.

The discriminator is treated as an energy function [7]. Rather than the probabilistic estimates given by conventional discriminators, the discriminator tries to minimize an energy function. The absence of normalization in this energy- based form of GAN (an EBGAN) provides greater flexibility in the choice of architecture of the discriminator and the training procedure. The realistic image manipulation is a challenge. Many images when manipulated look unnatural. An attempt is made in [8] to make realistic manipulation of images possible by attempting to keep the images on the manifold of natural images. A GAN learns this natural manifold from the training data. It then attempts to keep images looking realistic by ensuring they do not 'fall off' this natural image manifold.

A locally affine model in order to convert images between different times of day is proposed [9]. This was one of the first data driven approaches to this problem. Previous models were generally based on physical, 3D structure data. However, the problem is complicated by the presence of different materials in the scene and the fact that many parts in theimage undergo different color changes as an effect of reflectance, nearby geometry, shadows, etc. The authors of [10] elaborated on the need for a method to deal with the blurry results produced due to the use of a Mean Squared Error (MSE) as a loss function. The proposed concept supports three feature learning approaches: a multi-scale architecture, an adversarial training method, and an image gradient difference loss function. All these methods were individually shown to be free of the blurring effect associated with the MSE

loss function. The adversarial training method was inspired by the work in [4]. The aim of the paper is to develop a generic architecture using GANs that will be independent of the specific image translation problem it is meant to handle.

The multi-target semantic segmentation is applied on remote sensing images using a newer version of CNN called SegNet and U-net. Using SegNet structure Encoder-Decoder CNN with pooling index is designed. U-net is integrated with CNN to obtain the segmentation. The results of the two models are fed to an algorithm that can combine the semantic segmentation results of these two diverse CNN models segNet and U-Net. The proposed integrated algorithm exploits the advantages of these two models and improves the segmentation. The proposed method provides improved results that further validate our method [15]. Artificial Neural Network (ANN) model to restore image obtained by sensor exposure, saturation, and underexposure, is addressed [16]. This method has improved images damaged by an amount of 3%. Architecture used in this project is convolution layers with dilatation for context aggregation down and up scaling and uses a deterministic objective functions. The model constraints are tuned using likeness measures among the output and the reference image. Limitation is in filling large areas with severe saturation and intensification of block in JPEG compression, this is due to data interpolation problem and high-frequency noise at the edges of each compression block

Robust Multi-classification of brain tumors at the early stage is detected using CNN models using grid search. CNN model with different layers, activation functions link maxpooling, softmax are used to identify multi classification labeling of the tumor with fully connected layers. In the first phase, model is used to detect presence of tumor or not. In the second phase the model classifies brain tumor into normal, glioma, meningioma, pituitary and metastatic. In the third phase, model labels glioma tumors into different classes with an accuracy of 98%. This helps in initial screening of brain tumor [17]. Action recognition in video is performed using fully connected CNN model designed with convolution layer and rectified linear unit. This learns discriminative features efficiently and provides good recognition with reduced computational time and robust to action variations [18].

III. SYSTEM ARCHITECTURE

The process of image translation involves manipulating images from one type to the corresponding image of the second type. For example, converting gray scale image to a color image or converting a day scene into a night scene. This can be achieved by applying machine learning models. The input to the model, therefore, will be images of one type. The output of the model will be the corresponding images of the second type. The proposed model used here is called a Generative Adversarial Network (GAN). GANs basically consist of two neural nets called the Generator and the Discriminator. Generator imitate to create realistic images that attempts to deceive Discriminator and in the other hand a Discriminator tries to label images as existent or fake and is therefore trained not to be fooled by the Generator. In doing so, the Discriminator learns a loss function that classifies an image as real or fake, while the Generator tries to minimize this loss. Since the learnt loss function accustomed to data, the same GAN is capable for enormous tasks that would require diverse architectures. Since the loss function is not customized, it also results in negligible blurring of images and produces output quite indistinguishable from reality.

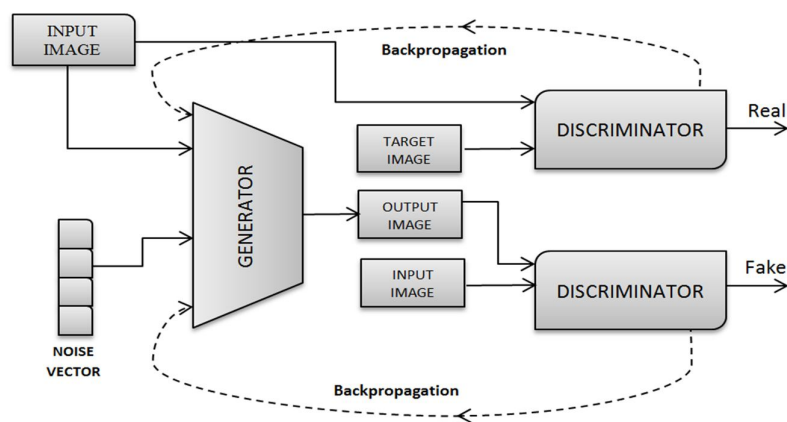


Figure 1: System Architecture

Another important feature of GAN is that it learns a structured loss. Most of the image conversion applications is developed as per-pixel classification or regression problems. Which treat the output space as unstructured in the sense that each pixel in the output is considered conditionally independent from all others given in the input image. The models are different in that they learn a loss which penalizes any possible difference in the structure of the output and the target. In this paper, a variation of the GAN called a conditional GANs it is conditioned on data. This conditioning makes it possible to direct generation of data process and helps to reduce the overall time taken to complete the training. The complete system architecture is illustrated in Figure1.

A. Generator Architecture

The architecture that has been used for the generator is called an encoder-decoder. The encoder is basically a normal convolution network which involves convolutions, pooling and activations as shown in Figure 2. The input to the generator is a 256 x 256 image with three color channels. As the image moves through the multiple layers of the encoder, it gets progressively down sampled.

The encoder therefore, reduces input image to smaller representation to achieve higher level representation of data. Batch normalization has been carried out which is used to shift the parameters and weights to zero mean and unit variance. This helps in improving the speed of learning and also the overall accuracy. The batch normalization layer has been added right after the convolution layer and before the non-linearities.

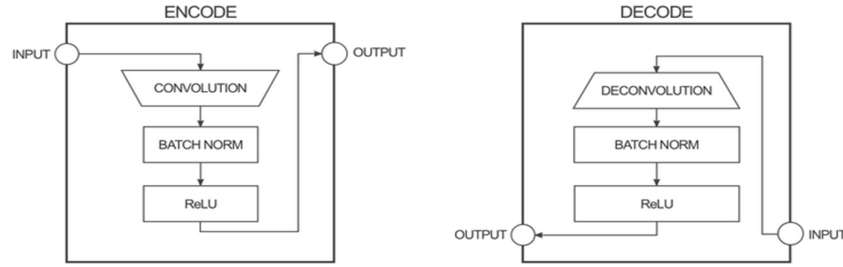


Figure 2: Operations of Encoder and Decoder Units

The activation that is used in the encoder is Leaky ReLU (Rectified Linear Unit) with 0.2 slope. Leaky ReLUs are an effort to fix the “dying ReLU” problem.

The decoder is a deconvolution network with activation function, as shown in Figure 2. The objective of the decoder is to reverse the operations of the encoder. The input to the first layer of the decoder is the down sampled input obtained at the last layer of encoder. The output of the last layer of the decoder and hence the generator network is an image which is of the same dimensions as the original input image. Another addition to this architecture is skip connections. Skip connections, as the name suggests, skips layers in the encoder and directly feeds the input of one layer to the next as well as some other layer. The main purpose of these skip connections is to provide the decoder layers information from the initial layers of the encoder so that the images can be properly reconstructed during up-sampling. Encoder-decoder along with the skip connections is termed as U-net. In our architecture, there is a skip connection between every i -th layer of encoder to the $(n-i)$ -th layer of the decoder. The complete architecture of the Generator is illustrated in Figure 3.

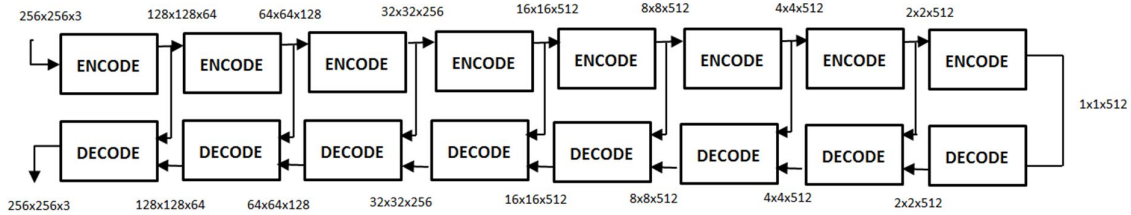


Figure 3: Complete architecture of the Generator

B. Discriminator Architecture

The discriminator is a regular convolutional neural network whose objective is to classify the input image as either real or fake. Real images are from the actual dataset and fake images are generated by the generator module. Since we are using a conditional GAN, we feed the target image along with the input by concatenating both to the discriminator. This is followed by a series of convolutions, batch normalization and activations as

shown in Figure 4 If the input image is from the real dataset, then the output of the classification should be “1”. If the input image is an image synthesized by the generator, then the output of the classification should be “0”. The discriminator is trained based on cross-entropy loss function.

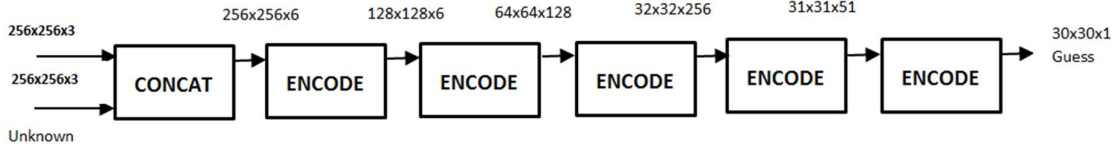


Figure 4 Discriminator architecture

C. Loss Function

The objective of the conditional GAN can be expressed as Eq(1). The Generator G tries to minimize this objective against the adversarial network D , which tries to maximize this objective. According to previous approaches to conditional GANs, it has been found that combining the GAN objective with traditional losses like L2 distance or L1 distance is more beneficial. This additional loss is used only for the Generator since it the images generated by the Generator should be very close to the target image and this additional loss provides for that. The Discriminator loss remains unchanged. The implemented GAN uses L1 distance rather than L2 since L1 reduces blurring of images. The final objective function is represented in Eq(2).

$$L_{GAN}(G, D) = (E_{Y \sim P_{data}(y)})[\log D(y)] + E_{x \sim P_{data}(x), z \sim P_Z(z)}[\log(1 - D(G(x, y)))] \quad (1)$$

$$L_{L1}(G) = (E_{x, y \sim P_{data}(x, y), z \sim P_Z(z)})[||y - G(x, z)||_1] \quad (2)$$

$$G^* = \arg \min_G \max_D L_{CGAN}(G, D) + \alpha L_{L1}(G) \quad (3)$$

IV. EXPERIMENTAL ANALYSIS AND RESULTS

Experimental analysis determines how well the system performs. The various metrics used for evaluating the performance of the implementation and the results have been discussed in this section. An analysis on effectiveness of the working model is quantitatively measured using metric and datasets. For performance evaluation of a GAN model it is important to consider the type of dataset to choose. Testing cannot be performed on the same training dataset, as the output will be prejudiced. Since, while training, the model would have tuned to the predicted results to the training dataset.

To estimate the generalization error, the model is required to test on new dataset. The main metrics used for testing performance of the implementation are the L1 loss of the Generator, Discriminator loss and the aesthetic quality of the generated images. The L1 loss determines how much near the generated image is with the target image for a particular dataset. The Discriminator loss determines how well the network can classify the generated images and real images. A multitude of datasets were used to find out how the GAN performs on different classes of image translation problems. The datasets on which the GAN was trained and tested are Cityscapes dataset (Semantic labels to photo), CMP facades dataset (Architectural labels to photo) and ImageNet (Black and white to color).

Two directions were used in the implementation namely, AtoB direction converts an image of a building to a labeled image and BtoA direction converts a labeled image to an image of a building. Figure 5 shows the format of dataset used for the Facades dataset. The same format has been used for Cityscapes as well as ImageNet datasets.



Figure 5 Dataset Format

The GAN was trained on Facades dataset with 400 images in both directions (AtoB and BtoA). The results obtained after 100 and 200 epochs for BtoA and after 200 iterations for AtoB direction are shown in Figure 6 and Figure 7.

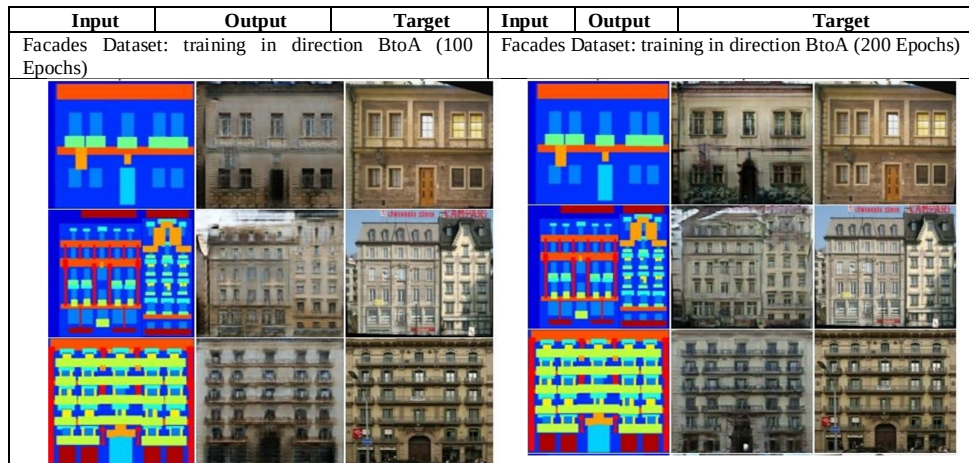


Figure 6 Facades Dataset: 100 epochs and 200 epochs of training in direction BtoA

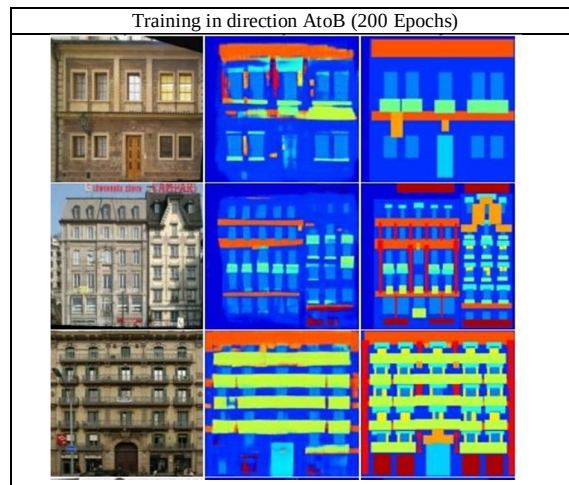


Figure 7 200 epochs of training in direction AtoB

The GAN was also trained on the Cityscapes dataset which had 2975 images in the direction AtoB. The results for 50 and 100 epochs are shown in Figure 8. The results indicate that the model is able to segment the images much more accurately at 100 epochs than 50 epochs.

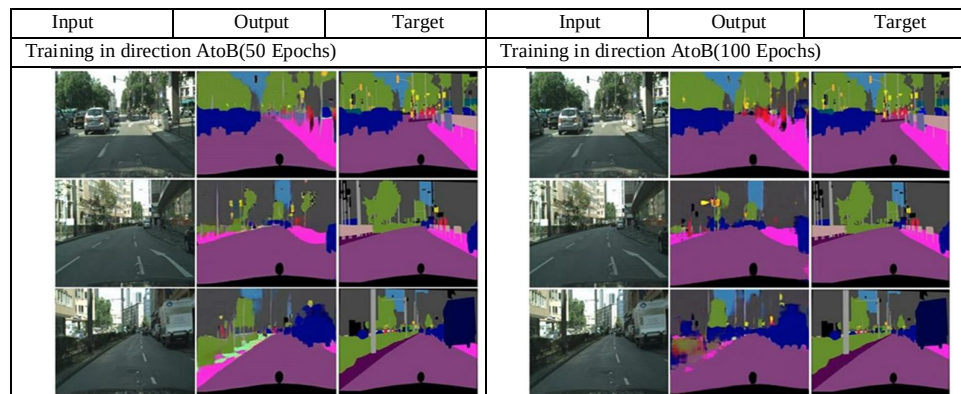


Figure 8 Cityscapes Dataset: (a) 50 epochs and (b) 100 epochs of training in direction AtoB

The GAN was finally trained and tested on a subset of the ImageNet dataset consisting of 3064 images for converting black and white images to color images. The results after 40 epochs and 160 epochs are shown in Figure 9.

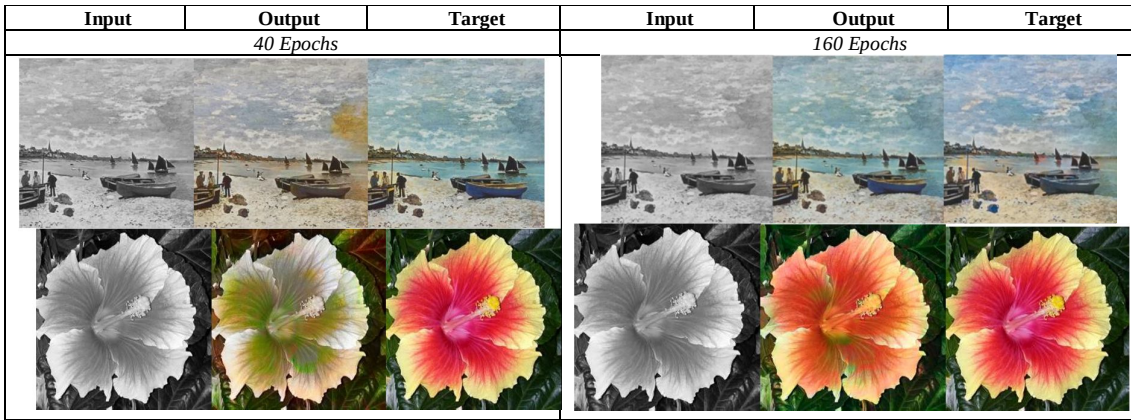


Figure 9a) Training on ImageNet dataset: Results after 40b) ImageNet dataset: 160 epochs training epochs

The performance of the GAN and CNN model depends on the loss metric as shown in Figure 10. The x axis of the graph specifies the total iterations of the training phase and the y axis specifies the measure of loss. L1 loss of the GAN slowly decreases from an initial value of 0.1(epoch 1) to 0.06(epoch 160) while the CNN loss does not show any trend assuch as shown in Figure 10b).

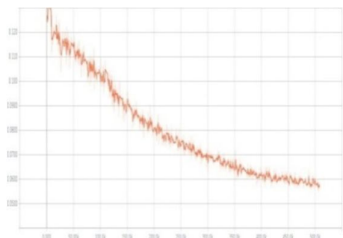


Figure10 a) Generator L1 loss

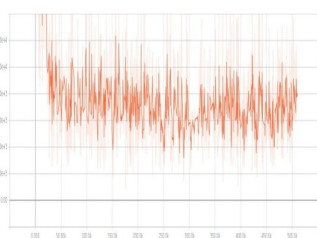


Figure10 b)CNN loss

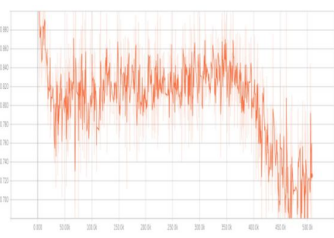


Figure10 c) Discriminator loss

The Discriminator loss graph is shown in Figure 10c). This graph indicates that as the amount of training iterations improves , the discriminator is able to label the images better as either real or fake.

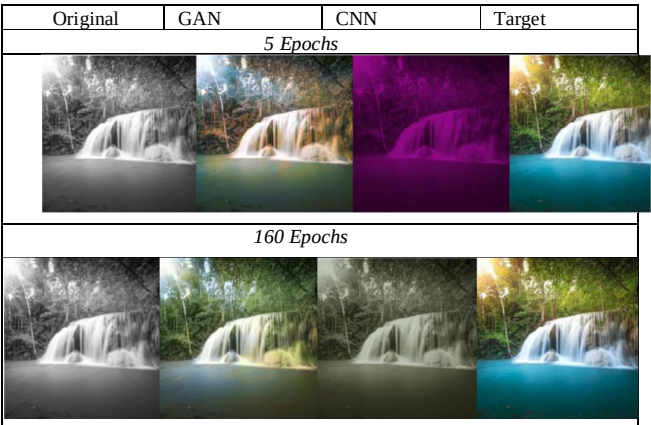


Figure 11 Imagenet Dataset: (a) 5 epochs and (b) 160 epochs

In order to assess the GAN model with the available CNN models, both were trained on a subset of ImageNet dataset which had 3064 images. This dataset was used to convert the gray scale image to a color image. The

output image obtained at various epochs of the training were compared. Figure 11 shows the result at 5 epochs and the result obtained at 160 epochs of both the GAN and CNN models.

V. CONCLUSION

The results obtained indicate that as the number of epochs increases, the Generator L1 loss decreases and the output images obtained are closer to the desired target image. This is observed for all the three datasets - Cityscapes dataset, CMP facades dataset and ImageNet on which the GAN was trained. The comparison between the GAN and CNN on the Imagenet dataset yielded that GAN produced better results in terms of the output image and the time taken. While the GAN took 26 hours to train on the Imagenet dataset for 200 epochs, the CNN took 34 hours for the same. The images obtained from the GAN are much closer to the expected target image than the ones generated by the CNN model. This shows that the CNN model requires a much larger dataset than the one used and the time it takes to complete the training phase is also more. In order to assess the GAN model with the available CNN models, both were trained on a subset of ImageNet dataset which had 3064 images. This dataset was used to convert the gray scale image to a color image. The output image obtained at various epochs of the training were compared. Figure 11 shows the result at 5 epochs and the result obtained at 160 epochs of both the GAN and CNN models.

REFERENCES

- [1] J. Long, E. Shelhamer, and T. Darrell, "Fully Convolutional Networks for Semantic Segmentation", 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp 3431–3440, [Jun2015]
- [2] S. Xie and Z. Tu, "Holistically-Nested Edge Detection", International Conference on Computer Vision (ICCV), [2015]
- [3] Alex J. Champandard, "Semantic Style Transfer and Turning Two-Bit Doodles into Fine Artworks", arXiv:1603.01768, [Mar2016]
- [4] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, Yoshua Bengio, 'Generative Adversarial Networks', NIPS, [Jun2014]
- [5] Mehdi Mirza, Simon Osindero, "Conditional Generative Adversarial Nets", arXiv:1411.1784, [Nov2014]
- [6] Alec Radford, Luke Metz, Soumith Chintala, "Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks", International Conference on Learning Representations (ICLR), [Jan 2016]
- [7] Junbo Zhao, Michael Mathieu and Yann LeCun, "Energy-Based Generative Adversarial Networks", International Conference on Learning Representations (ICLR), [Mar2017]
- [8] Jun-Yan Zhu, Philipp Krähenbühl, Eli Shechtman and Alexei A. Efros. "Generative Visual Manipulation on the Natural Image Manifold", European Conference on Computer Vision (ECCV). [2016]
- [9] Shih, Yichang, Sylvain Paris, Frédo Durand, and William T. Freeman. "Data-Driven Hallucination of Different Times of Day from a Single Outdoor Photo." ACM Transactions on Graphics 32, No. 6, pp 1–1, [Nov2013]
- [10] Michael Mathieu, Camille Couprie, Yann LeCun, "Deep multi-scale video prediction beyond mean square error", arXiv:1511.05440, [Feb2016]
- [11] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, Alexei A. Efros, 'Image-to-Image Translation with Conditional Adversarial Networks', arXiv:1611.07004 [cs.CV], November 2016
- [12] Jun-Yan Zhu, Philipp Krähenbühl, Eli Shechtman and Alexei A. Efros. "Generative Visual Manipulation on the Natural Image Manifold", European Conference on Computer Vision (ECCV). [2016]
- [13] Richard Zhang, Phillip Isola, Alexei A. Efros, 'Colorful Image Colorization', arXiv:1603.08511 [cs.CV], October 2016
- [14] Jun-Yan Zhu, Taesung Park, Phillip Isola, Alexei A. Efros, 'Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks', arXiv:1703.10593 [cs.CV], March 2017
- [15] Alam, M., Wang, J.F., Guangpei, C. et al. 'Convolutional Neural Network for the Semantic Segmentation of Remote Sensing Images'. Mobile Netw Appl 26, 200–215 (2021), <https://doi.org/10.1007/s11036-020-01703-3>. February 2021
- [16] Steffens, C.R., Messias, L.R.V., Drews-Jr, P.J.L. et al. 'CNN Based Image Restoration.' J Intell Robot Syst 99,. <https://doi.org/10.1007/s10846-019-01124-9> 609–627 (2020).
- [17] Irmak, E. Multi-Classification of Brain Tumor MRI Images Using Deep Convolutional Neural Network with Fully Optimized Framework. Iran J Sci Technol Trans Electr Eng 45, 1015–1036 (2021). <https://doi.org/10.1007/s40998-021-00426-9>, 10 April 2021.
- [18] Russel, N., Selvaraj, A. 'Fusion of spatial and dynamic CNN streams for action recognition'. Multimedia Systems (2021). <https://doi.org/10.1007/s00530-021-00773-x> 23 March 2021.