MULTIPLE EDITOR OPINION MODELLING USING GENERATIVE ADVERSARIAL NETWORK FOR STYLE TRANSFER

PRATEEK KUKREJA

MULTIPLE EDITOR OPINION MODELLING USING GENERATIVE ADVERSARIAL NETWORK FOR STYLE TRANSFER

A thesis submitted in partial

fulfillment of the requirement for the award of the degree of

Master of Technology

in

Visual Information and Embedded Systems

by

Prateek Kukreja
[Roll No: 19EC65R15]

Under the supervision of

Prof. Prabir Kumar Biswas

Department of Electronics and Electrical Communication Engineering
Indian Institute of Technology Kharagpur
Kharagpur, West Bengal, India – 721 302
April 2021

CERTIFICATE

This is to certify that the thesis entitled **Multiple Editor Opinion Modelling using Generative Adversarial Network for Style Transfer**, submitted by **Prateek Kukreja** to the Department of Electronics and Electrical Communication Engineering, IIT Kharagpur, in partial fulfilment for the award of the degree of Master of Technology, is a bona fide record of work carried out by him under my supervision and guidance. The thesis has fulfilled all the requirements as per the regulations of this Institute and, in my opinion, has reached the standard needed for submission.

Place: KHARAGPUR
Date: 16/04/2021

Prof. Prabir Kumar Biswas
Project Guide
(E&ECE Dept.)
IIT Kharagpur.

DECLARATION

I declare that

a. The work contained in this thesis is original and has been done by me under the guidance of my supervisor.

b. The work has not been submitted to any other Institute for any degree or diploma.

c. I have followed the guidelines provided by the Institute in preparing the thesis.

d. I have conformed to the norms and guidelines given in the Ethical Code of Conduct of the Institute.

e. Whenever I have used materials (data, theoretical analysis, figures, and text) from other sources, I have given due credit to them by citing them in the text of the thesis and giving their details in the references. Further, I have taken permission from the copyright owners of the sources, whenever necessary.

Date: Prateek Kukreja

Acknowledgments

Firstly, I express my gratitude to the almighty God who I think has always been by my side and guided me through all the crests and troughs of life including stay at IIT Kharagpur. The realization that God is with me has helped me keep cheerful among the toughest of times.

I would like to thank my faculty advisor and project guide, Prof. Prabir Kumar Biswas, without whose insightful advice and guidance this work would not have been possible. He is one of the most approachable faculties in the institute. He has been very patient, always encouraged questioning and gave me full freedom to chart my own course during this work.

I am thankful to Mr. Aupendu Kar, research scholar in the under Prof. Prabir Kumar Biswas, for his help throughout this work. He was the one who initiated me to work on Generative modelling networks. His immense help in all aspects of this work related to machine learning is invaluable.

I am thankful to Mr. Arumoy for his orderly and dedicated supervision of the Computer Vision Lab. His meticulous upkeep always kept all the resources in the lab in a fine running state.

I will always remain indebted to my family for their constant encouragement, support and faith in me. Their faith is my biggest asset and strength.

Finally, I would like to thank all my classmates who made my stay wonderful at IIT Kharagpur. The incredibly humorous online group chats at odd times were a welcome distraction from the hectic schedule. All the worldly conversations with innumerable cups of tea at the department canteen will always have a special place in me. I would like to name them all: Yash Gupta, Pranoy Mukherjee, Abhishek Gupta, and Shubham Mandloi.

Prateek Kukreja

List of Acronyms

NST	Neural Style Transfer
I2I	Image to Image
GAN	Generating Adversarial Networks
CNN	Convolutional Neural Networks
PSNR	Peak-signal to noise-Ratio
FID	Frechet Inception Distance
SSIM	Structural Similarity Index Measure

Abstract

Image-to-image(I2I) translation is a class of computer vision and image processing related issues where the objective is to learn the transformation between an input image from source domain and an output image in the target domain. It has large number of applications, such as artistic style transfer, object transformation, season transfer and image enhancement.

Neural Style transfer(NST)[11] is also a way for I2I translation, it builds on the idea that it is possible to separate the style representation and content representation in a CNN, during the learning task. NST employs a pretrained CNN to transfer style from given image to another. This is done by defining a loss function that has objective of minimize the distance between content image, a generated input image and input style image.

A lot of significant advances have been made in I2I translation with Generative Adversarial Networks(GANs) as core Neural Network architecture[5], [6]. But it is still challenging to effectively translate an image to set of diverse images in multiple target domain using a pair of generator and discriminator. Earlier works on I2I translation methods involve pixel regression or classification using Encoder-Decoder pair based on Convolution Neural Networks(CNNs) if pair of labeled data is available[4]. More recently, GANs are used as they give diversity in output, as they learn probability distribution of target domain as compared to one to one mapping from one domain to another domain.

Existing multi-modal I2I translation methods that work well utilize many number of domain-specific content encoder for different domains, where during training each domain specific encoder is trained with images from same domain. But, this scheme fails due to latent code condition injection method to condition the generator network leads to mode collapse, resulting in giving single domain output for all latent code conditions.

We explore different methods for two domain mapping, multi-modal and multi-domain I2I. We also try to generate artistic images by interpolating the transformation from one domain to another domain.

Contents

List of Figures

Chapter 1

Introduction

1.1 Problem Statement

Our work is based on I2I translation with respect to application of style transfer. I have divided work under three Objectives. They are:

- Two Domain mapping
- Multi-Modal Mapping
- Multi-Domain Mapping

This work can also extent to interpolation of transformation space.

Imagine if you take a selfie and want to make it more artistic as a drawing from a cartoonist, how can you automatically achieve that with a computer? This type of research work can be broadly deemed the image2image translation (I2I) problem. In general, the goal of I2I is to transform an input image x_A from a input domain A to a output domain B with the intrinsic source content information preserved and the extrinsic target style transferred. To this end, we need to train a mapping G: A→B that generates image $x_{AB} \in B$ indistinguishable from target domain image $x_B \in B$ given the input source image $x_A \in A$. Mathematically, we can model this translation process as $x_{AB} \in B : x_{AB} = G: A \rightarrow B(x_A)$.

1.1.1 Two Domain Mapping

Two-Domain Mapping is I2I at very basic level, here we are going to explore supervised learning. For this I2I translation, we should have paired images i.e. (X_A, X_B) where x_A is in source domain and x_B is in target domain. ($x_A \in A$, and $x_B \in B$). We use encoder and decoder based Convolutional Neural Network and MSE Loss function is used to update weights of network. Its works in the sense that it first learns low-dimension latent code and then builds up on that to final image.

Another method to explore is GAN based network i.e. pix2pix[3] where U-Net[4] is used in generator and PatchGAN style discriminator. It uses Adversarial Loss along with L1 loss to train and learn probability distribution of the target domain. These approaches generally produce deterministic outputs i.e. same output for same input(one to one mapping). Two domain mapping is represented in figure:

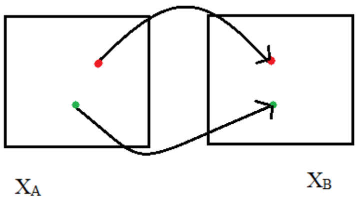


Figure 1.1: Two-Domain Mapping

1.1.2 Multi-Modal I2I

A basic approach of implementing a multi-modal I2I is learning a low-dimensional code, which should represent information of the possible outputs not present in the input image. At testing time, a generator uses the input image, along with latent codes sampled from stochastic distribution, to generate randomly sampled outputs. The most common problem in existing multi-modal I2I methods is mode collapse, where only few real samples get generated in the output.

I use BicycleGAN[2] architecture to solve above commonly observed issue in GANs. Considering the input domain $X_A \subset R^{H\times W\times 3}$, which we want to map to an output domain $X_B \subset R^{H\times W\times 3}$. During training, we use a dataset of paired images from these domains, {(A∈A, B∈B)}, that is given by a joint probability distribution p(AB). Here it is observed that there are many set of images domain X_B that can be mapped from X_A, but the dataset available to us only has one2one mapping pairs. Now, during testing with a new image instance X_A, this BicycleGAN network is able to generate a diverse set of output X_B's, according to different modes in the conditional probability distribution p(B|A).

Here z ~N(0, σ) is concatenated with input feature map of 4D. z $\subset R^{H\times W\times Z}$ is concatenated to input feature map B $\subset R^{H\times W\times Z}$, this is classical latent code injection method in the Neural Network.

Multi-Modal I2I translation is represented in Figure:

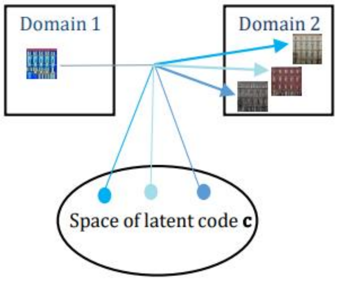


Figure 1.2: Multi-Modal I2I

1.1.3 Multi-Domain I2I

Multi-Domain refers to different domains having completely different attributes, e.g. pencil sketch images, cartoon images, art/painting images etc. Here we want to translate image from one domain to multiple other domains where no or very less attributes are shared among these target domains.

We use unsupervised learning to through to achieve results in this section as paired images are not easily available that are aligned in multiple domains ie same image stylized in multiple domains. In case of availability of paired images, we use Conditional GAN[8] to achieve Multi-Domain I2I translation but it leads to mode collapse, which is a common issue observed in GAN[6] training procedure where only few real samples are generated in output.

Here we use SimpleGAN[1] Network architecture which is an extension of BicycleGAN[2], here it uses multiple discriminators corresponding to multiple domains to classify real/fake. Moreover, it also introduces different domain code/latent code injection technique in Neural network. More of this will be explained in Methodology chapter. Multi-Domain I2I translation is represented in Figure:

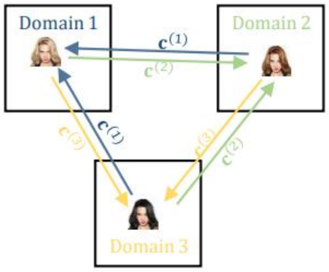


Figure 1.3: Multi-Domain I2I

1.2 Neural Style Transfer

Neural style Transfer(NST) is a technique that was developed on Neural Network that generates good stylized images[11]. It is a very good idea to achieve I2I translation, it builds on the idea that it is possible to separate the style representation and content representation in CNN during the training of the Neural Network.

Convolutional Neural Networks(CNN) trained on dataset for goal of object classification, this network understands the feature map of the image which allows us to classify object increasingly in the later layers of CNN. Therefore, at the later stage/layers of the network, the input image is converted into feature maps that has more information about the structural component in the image and not about its texture or style details. This can be directly represented by the information contained in each layer by reconstructing the image only from the feature maps in that layer. Last few layers in the network learn more detailed structural information in terms of object shapes and location in the input image instead learn the color/texture/style information. We thereby call the feature maps in last few layers of the neural network as the structural information and use it to calculate content loss.

To obtain style/textural information of an image, a feature map is used which was originally designed to learn texture/style information. This feature maps are built on after the filter responses from every layer in the neural network. Correlations is calculated between the separate filter responses over the corresponding feature maps. This feature correlation calculated over the layers, provide us with correlated, multi-scale information of the image, this process provides us with texture/style information but not the content information.

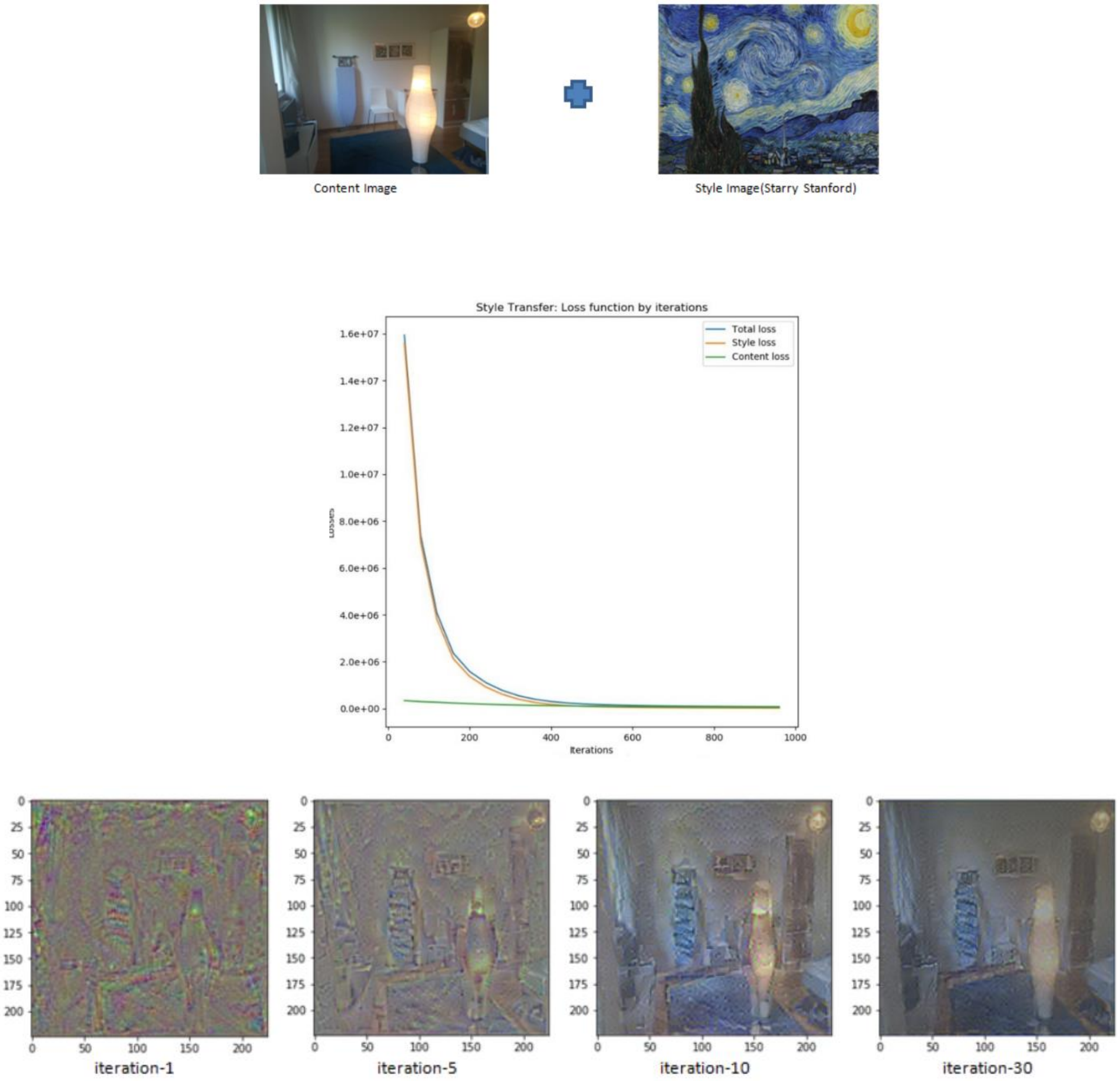


Figure 1.4: NST process over the iterations

1.3 U-Net

The U-Net was originally designed by Olaf Ronneberger et al. for Bio Medical Image Segmentation[4]. Its primary task is semantic segmentation. They are many levels of complexity in image and U-net helps to understand the image by first classification for capturing the semantics of image and then localization for knowing the position of semantics.

The U-net network has two paths, the first path is the contraction path which is used to capture the context in the image or to know what are semantics of image i.e. objects in image(car, water, grass, flower). The encoder or contacting is a convolutional neural network(CNN) with max pooling layers, which used for knowing 'WHAT' are the semantics of image(Classification) but in the process of finding what are semantics we lose the information of position of those semantics, i.e. 'WHERE' are those semantics(Localization) ,for this an expanding path is used and as max pool is used for contracting in encoder network similarly up sampling with transposed convolution is done in expanding path which tries to replicate inverse of max pool layer.

Actually idea behind transposed convolution(also known as deconvolution) is reversing the convolution operation but here we want to reverse the max-pool operation, and as in ideal transposed convolution the weights of convolution convolution is known but here max pool layer dont have any weights and the network tries to predict the weights of convolution matrix ,which is approximation transpose of max-pool layer used for up sampling ,which when applied to a matrix of reduced size resulted after max-pool, will approximate the matrix of bigger size on which max-pool is applied at first place.

The reason behind U-net segments & localizes with such finer details is skip connections. They play an important role and solve two problems, first they transfer the information lost during contraction which is the position of semantics and as it is a very deep network it has a problem of vanishing gradients, which is solved by these skip connections as it transfers the gradients from end layers to starting layers as shown in architecture.

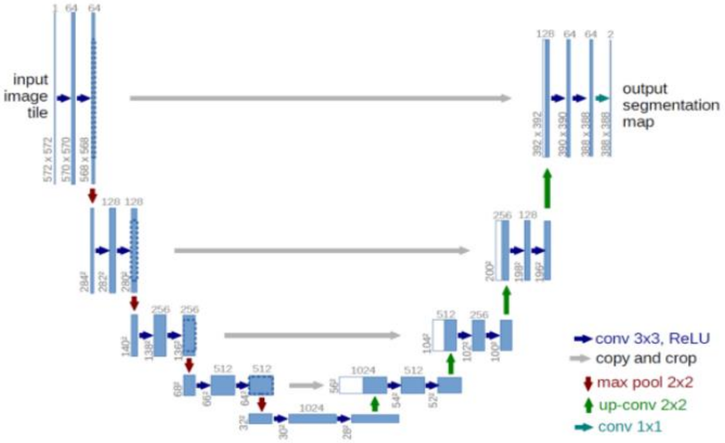


Figure 1.5: U-Net Architecture

1.4 Generative Adversarial Networks

Generative Adversarial Network (GAN) neural network architecture was proposed by Goodfellow et al. [5, 6]. It was new type of generative model that overcame several limitations of existing generative models. The proposed model of GAN comprises of two neural networks called the generator and discriminator. It works on the premise of Game theoretic situation in which there are two networks that are opponents of each other and compete with each other, this in turn makes them better during the training phase. The generator's Objective is to generate sample images that are indistinguishable from real image data. Its opponent, the discriminator's objective is to classify the a image is generated from the generator's model learned probability distribution or the original image data probability distribution. As training proceeds, generator learns to make its image data distribution very close to real image dataset distribution and both of these networks are trained at same time till the generator has learned to produce samples consistently that the discriminator fails to classify as real or fake (i.e. Give a value of 0.5 for both real image and fake image).

Both generator and discriminator are constructed from multilayer perceptron with many layers so as to model complex information. But the problem of style transfer belongs from the class of I2I translation which is why the generator and discriminator are both constructed using convolutional neural networks (CNNs). The generator is function of random sampled noise is given by G(z; θG), where z is a noise variable sampled from normal distribution that acts as the input. Similarly, the discriminator is function of real and fake images given by D(y; θD) and gives a value between 0 & 1 which is classification probability. The output of the discriminator is between 0 to 1 which can be explained as the probability of the input is from a real image if its value is 1 and probability of the input is from a fake image if its value is 0. These implementations of Generator network and Discriminator network are such that the optimization objective: Generator is trained in a way that we want to decrease the classification probability of discriminator of classifying generator output as real i.e. predict generated sample as original one, while Discriminator is trained in such a way that we want to maximize the probability of making correct decision i.e. not to be fooled by generator. This loss function of minmax is very different from any distance based loss function (e.g. Euclidean loss) hence averaging effect is not there.

$$min_{\theta G} J_G(\theta D, \theta G) = min_{\theta G} E_z[log(1 - D(G(z))]$$
$$max_{\theta D} J_D(\theta D, \theta G) = max_{\theta D}(E_x[log(D(x)] + E_z[log(1 - D(G(z))])$$

We can also represent above two Objective functions in a single optimization objective equation V(G, D) as:

$$min_G max_D V(G, D) = E_x[logD(x)] + E_z[log(1 - D(G(z)))]$$

This optimization objective proposed has few drawbacks as observed during training stages. This Objective is monotonically decreasing function which is not bounded in reverse direction. This sometimes make objective to diverge to negative infinity during the training stage. Another issue is that during early stages of training process, generator output is very bad, which makes its classification as fake very easy for discriminator network. This causes generator network parameters to not being updated during training. This results in very slow convergence.

Above mentioned issues are resolved by modifying the generator's Objective function to maximizing the probability of the generator output being classified incorrectly, instead of minimizing the probability of the discriminator classifying generator output correctly.

$$max_{\theta G} J_G(\theta D, \theta G) = max_{\theta G} E_z[log(D(G(z))]$$

Another way to write above Objective is as follows:

$$min_{\theta G} J_G(\theta D, \theta G) = min_{\theta G} - E_z[log(D(G(z))]$$

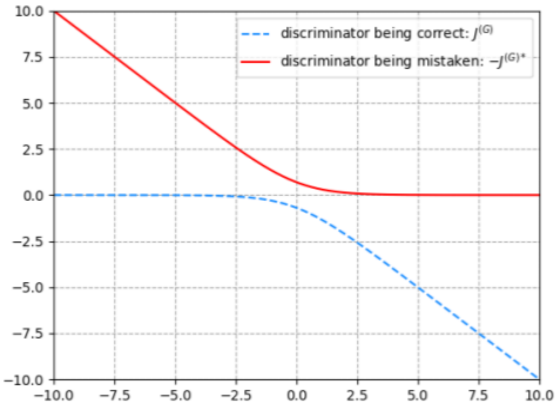


Figure 1.6: Objective function variation

This cost function is modified a little bit by using ℓ^1-norm to regularization the neural network. This had an effect on generator that it is forced to generate images that are very much same as ground truth.

$$min_{\theta G} J_G(\theta D, \theta G) = min_{\theta G} - E_z[log(D(G(z))] + \lambda ||G(z) - y||_1$$

where λ is a hyperparameter that controls amount of regularization.

1.5 Conditional GAN

In the originally proposed GAN architecture, a randomly sampled noise data z from some standard distribution is given to generator as input[8]. But this method cannot be applied to the problem of underexposed image enhancement because grayscale target images act as input for our problem, not noise. Conditional generative adversarial networks addressed this problem by using a modified version of GAN. This makes the generator being conditioned by the input, or in terms of mathematics, G($\mathbf{0}_z|x$).Discriminator input is also changed according to condition selected for conditioning generator network. Finally, our Objective functions become:

$$min_{\theta G} J_G(\theta D, \theta G) = min_{\theta G} - E_z[log(D(G(0_z/x))] + \lambda||G(0_z/x) - y||_1$$
$$max_{\theta D} J_D(\theta D, \theta G) = max_{\theta D}(E_y[log(D(y/x))] + E_z[log(1 - D(G(0_z/x)/x))])$$

The discriminator will get Expert images from both the original dataset and generator and has to decide which has been generated by Generator and which is actual image from dataset.

1.6 Instance Normalizarion

IN is also knows as contrast normalization, it is the normalization method that has been proposed in 2017[9] specifically for image to image translation for style transfer. It suggests that this improved normalization method gives better quality results over more common batch normalization in case for images in style transfer application.

The results are show cased in case of NST. The original work proposed in 2016 showed that it is possible to train a network g(x, z) that can be used to change style of a input content image x from the style of another style input image x0, this is similar to original NST work. In this work, the style image x0 is fixed and the neural network learns to transfer the style to any input image x. The variable z is a randomly sampled seed that is used to obtain diverse stylization results. The neural network is a convolutional neural network. Here an example is just a content image x_t, t = 1, . . . , n

$$y_{tijk} = \frac{x_{tijk} - \mu_{ti}}{\sqrt{\sigma_{ti}^2 + \epsilon}}, \quad \mu_{ti} = \frac{1}{HW}\sum_{l=1}^{W}\sum_{m=1}^{H} x_{tilm}, \quad \sigma_{ti}^2 = \frac{1}{HW}\sum_{l=1}^{W}\sum_{m=1}^{H}(x_{tilm} - mu_{ti})^2.$$

Result Comparison:

Content Image Style Image BN Result IN Result

Figure 1.7: Comparison between BN and IN in NST

1.7 ADIN

Adaptive Instance Normalization is a normalization method is used to combine content features with style features by using their statistical properties such as mean and variance.
Adaptive Instance Normalization is an improvement over IN. In AdaIN, we have two input images A and B, and we just have to calculate the channel-wise mean and variance of A and match it with those of B. AdIN accordingly learns the affine parameters from the one of the inputs, it does not have its own learnable affine parameters like other normalization techniques.

$$\mathrm{AdaIN}(x, y) = \sigma(y)\left(\frac{x - \mu(x)}{\sigma(x)}\right) + \mu(y)$$

1.8 CBIN

CBIN is an Domain code injection scheme in Neural Network for the Multi-Domain I2I translation task. It was proposed in 2020[10], paper suggest that due to existing normalization techniques, mode collapse occurs in multi-domain modelling is used classical domain code injection method. These normalization methods either causes the discrepancy in the feature maps distribution and eliminate the effect of the domain latent code. To rectify these issues, they proposed the consistency within diversity criteria for designing the multi-domain mapping model.

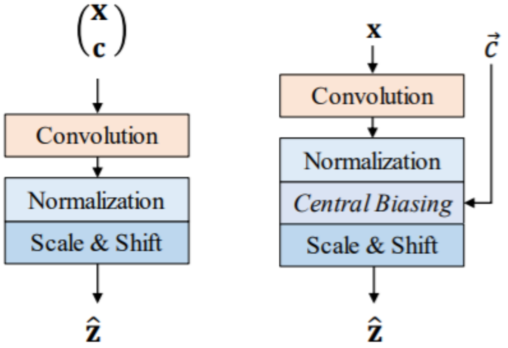


Figure 1.8: left: classical domain code injection right: proposed method

It proposes that IN leads to impaired diversity in the generator output, and BN leads to impaired consistency. CBIN effectively achieves consistency within diversity i.e.:
Consistency criteria: To reduce mapping inconsistency, different inputs should produce similar outputs when conditioned on same domain label code.

$$\mathrm{mean}(g(x_A, c)) = \mathrm{mean}(g(x_B, c)).$$

This show that statistical measure mean of the mapping function g(x, c) should not be relate to the input image.

Diversity Criteria: For maintaining diversity, same input image should produce diverse outputs when given different domain latent code c_1 and c_2. This means that the mean value of the outputs should not be similar: mean(g(x, c_1)) = mean(g(x, c_2)).
This is same as saying that the statistical mean of the learned mapping function g(x, c) should be related to the domain label code c .

$$\mathrm{CBIN}(x_i) = \frac{x_i - \mathrm{E}[x_i]}{\sqrt{\mathrm{Var}[x_i]}} + \tanh(f_i(z)),$$

1.9 Contribution Of This Thesis

- Did a comparison based study of different I2I models using Generative neural network architectures, and also explored NST with application to style transfer
- Show cased various style transfer applications such as under-exposed images enhancement, photo2art, photo2sketch.
- Objective has been divided into two-domain mapping, multi-modal mapping and multi-domain mapping. All these objectives were met with various GAN based architectures.
- Future work has been proposed with regards to extending this work to interpolating the transformation achieved by these implemented GAN based architectures.

1.10 Organization Of This Thesis

Different parts of this work are summarized below:

- Chapter 2 is Literature Review of the methods proposed for I2I translation. All these methods are widely known and contributed extensively to the image based neural network applications.
- Chapter 3 describes Datasets used throughout this work and Evaluation metrices used to evaluate the working models.
- Chapter 4 presents the Methodology of all the models that were developed and utilized to achieve I2I and how they met out criteria of proposed Objectives
- Chapter 5 is Future Works proposed by us as an extension to this work.
- Chapter 6 Concludes this thesis by presenting short summary of techniques and objectives met in this work.

Chapter 2

Literature Review

2.1 Pix2Pix

Pix2Pix is one of the most popular GAN based I2I translation architecture that uses paired image data to learn probability distribution of target domain images[3].
Conditional GAN is improved version of originally proposed GAN architecture, it enabled to control the output of Generator network by conditioning the generator during its training phase. In pix2pix conditioning is done by the input image given along the randomly sampled noise.

The objective Loss function of a cGAN can be represented as

$$L_{cGAN}(G,D) = E_{x,y\sim pdata(x,y)}[logD(x,y)] + E_{x\sim pdata(x),z\sim p_z(z)}[log(1-D(x,G(x,z)))]$$

here objective is minimized with respect to G and objective is maximized with respect to D i.e.

$$G^* = argmin_G max_D L_{cGAN}(G,D)$$

It also introduces L1 loss, it was observed that L1 loss causes smoothening effect to decrease as compared to L2 loss.
Full Objective of Pix2pix is:

$$G^* = argmin_G max_D L_{cGAN}(G,D) + \lambda L_{L1}(G)$$

Without randomly sampled noise from standard distribution z, the neural network can learn a transformation of image from source domain to target domain, but it will generate outputs that have no variation which is evident from distribution it learns, and hence, it will fail to learn the distribution of input images, instead it learns an impulse function.Earlier works on cGAN have experimented regarding this and proved that giving noise sampled from standard source should be given to generator along with input.

Network Architecture:
Generator network is made using U-Net[4] model, image is encoded to a low-dimension representation, then it is upscaled using decoder to get actual output image, to enable reconstruction of image and preserve information across layers, connections are made from ith layer to (n-i) layer. This also help in preventing vanishing gradient problem encountered in deep neural networks(DNN).
Discriminator is modelled after on PatchGAN(Markovian Discriminator), this is used as it is observed that L2 and L1 loss generally produce blurry results. In order to retain high frequencie information, it restrict attention to the local image patches and classify pathes as real/fake instead of one classification for whole image.

Paper showcased applications like image colorization, labels2street scene, Ariel2Map view, day2night scenes etc.

2.2 CycleGAN

This generative model uses unsupervised learning to achieve I2I translation [17]. It is very difficult to come by paired image data, and most data available is in unlabeled form. So this method is I2I has easier time working compared to methods that require paired data. This proposed method is built upon GAN architecture and uses 2 generators and 2 discriminator networks as building blocks.

First GAN model is used to translate image from domain A to domain B. Other GAN model is used to translate image from domain B to domain A.

Above mentioned process is not sufficient to model the translation effectively, an additional constrain is used called cycle consistency loss. To understand this loss, take example of two different languages A and B. If we are able to convert some word from A to B and we should be able to convert word from B to A. This is also called bijection.

This cycle loss add additional constrain between output of GAN1 and output of GAN2. This essentially makes sure that one to one mapping is effectively learned by the network.

Loss Function: we are having image data from 2 domains A and B, we want to learn a transformation G:A→B and F:B→A. We want to enforce the intuition that these mappings should be reverses of each other and that both mappings should be bijections. Cycle Consistency Loss encourages F(G(x))≈x and G(Y(y))≈y. It decreases the possible one to one mapping learned by network by removing irrelevant mappings. This is achieved as a result of using cycle loss to achieve bijection.

$$\mathcal{L}_{cyc}(G,F) = \mathbb{E}_{x\sim p_{data}(x)}[||F(G(x)) - x||_1] + \mathbb{E}_{y\sim p_{data}(y)}[||G(F(y)) - y||_1]$$

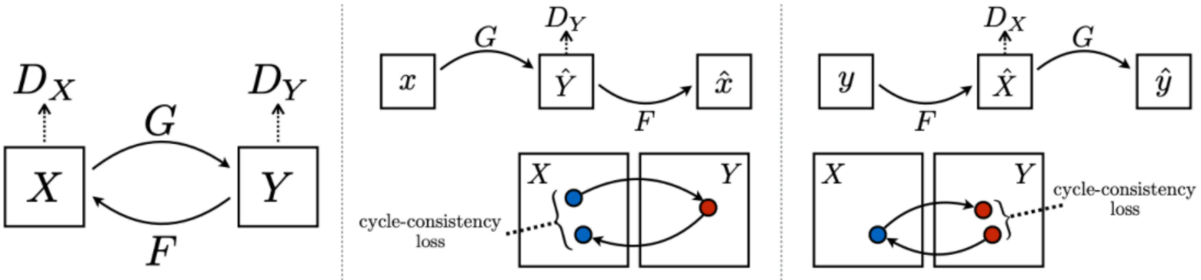


Figure 2.1: High-level overview of CycleGAN

2.3 UNIT(Unsupervised Image to Image Translation)

In this method of unsupervised I2I translation, an assumption is taken that there exist a point in shared latent space which represent an image from source and one image from target domain that can be mapped back to a same point in latent representation in shared latent space. I2I translation is performed through that shared latent space[15].

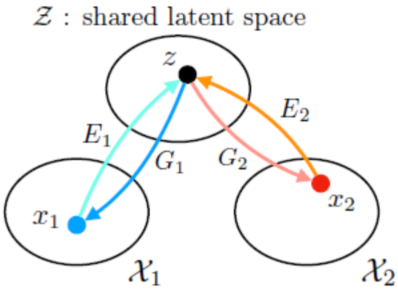


Figure 2.2: Representation of shared latent code space

If we take any 2 images from X1 and X2, then we can show that a latent code z shared among images from different domains in a shared latent space.

This model proposes use to 2 Generating Networks: GAN1(D1, G1) and GAN2(D2, G2). Images from X1 and classified by discriminator of GAN1 and image from X2 and classified using discriminator of GAN2The images generated by *G*1 can come from 2 sources, one is the reconstruction (VAE), one is the translation (GAN). For image-to-image translation, i.e., *X*1 → *X*2 and *X*2 → *X*1 through *Z*, it is called the image translation stream. In contrast to the self-reconstruction, this image translation stream is adversary trained**.**

2.4 MUNIT(Multi-Modal unsupervised Image to Image Translation)

This is extension of UNIT paper, it proposes method to generate diverse target domain output from given source domain image[16]. An assumption has been taken into consideration while designing the method that an image can be separated into domain invariant information and domain variant information. This domain invariant in information is shared among all the images and domain variant information is represented separately in different spaces by network. Below figure show the above assumption.

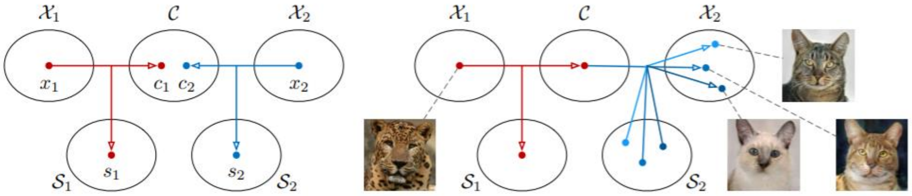
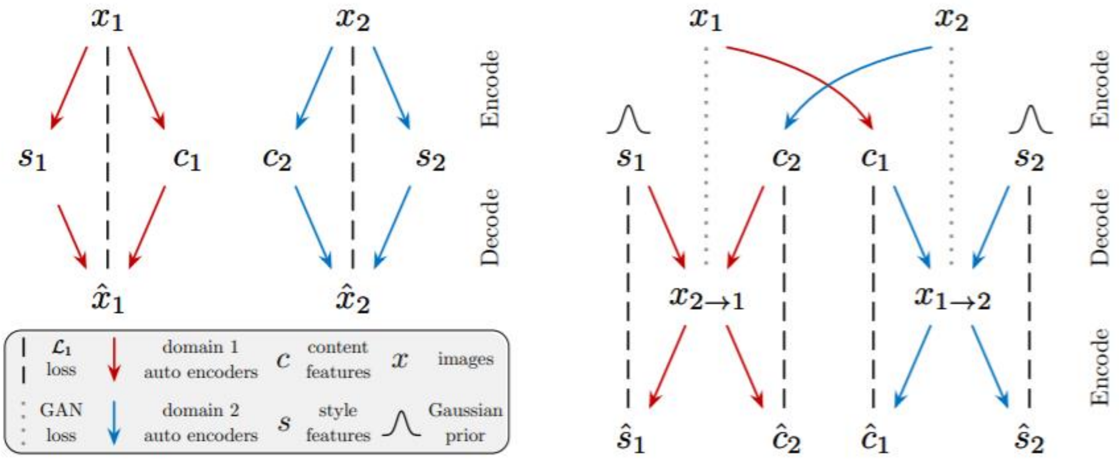


Figure 2.3: Representation of MUNIT

Let x1 ∈ X1 and x2 ∈ X2 be the unpaired images sampled from different domain dataset. In unsupervised I2I method, we try to are given access to data distributions of X1 and X2. Our objecting is to estimate conditional distribution to achieve I2I tasks. Below Figure gives detailed flow of image data from different domains to learn multi-modal mapping between domains.



(a) Within-domain reconstruction

(b) Cross-domain translation

Figure 2.4: High level overview of MUNIT proposed menthod

Loss Functions:

Total Objective = Image Reconstruction + Latent Reconstruction + Adversarial Loss

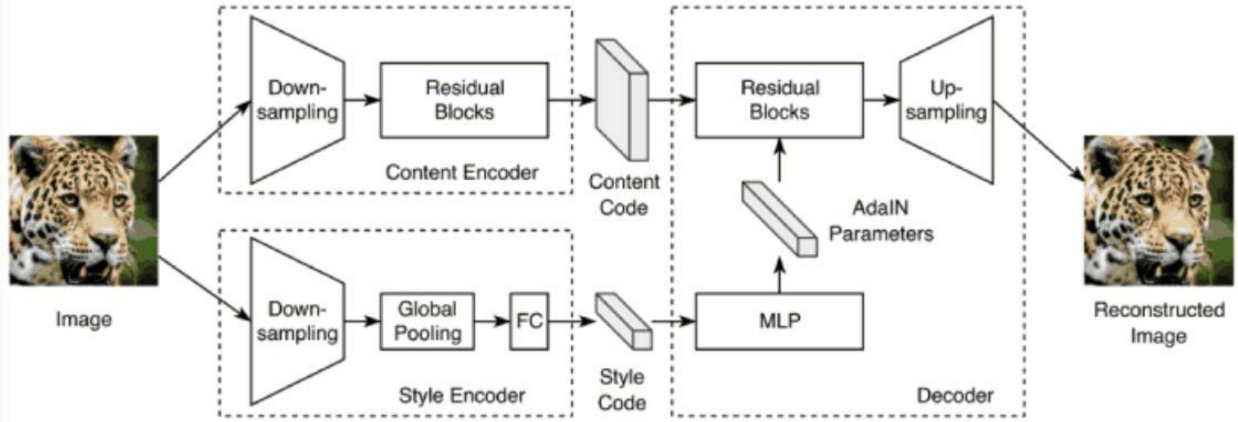


Figure 2.5: Architecture proposed

Chapter 3

Datasets and Evaluation Metrices

3.1 Datasets

MIT fiveK dataset: This is collection of images where an image of low exposure was given to 5 photo editing experts and their goal was to make this image as good as possible. So we have paired data. Let original image is x then we have 5 paired set{(x, Expert A), (x, Expert B), (x, Expert C), (x, Expert D), (x, Expert E)}.

Photo2art, monet2photo, cezanne2photo, vangogh2photo: These dataset have paired data set of natural image and corresponding different art version.

night2day: This is collection of image of night scene and corresponding day scene.

pencil Sketch images downloaded from google images

3.2 Evaluation Metrices:

Peak signal to noise ratio (PSNR): It is one of the most widely used full-reference quality metrics. It calculates the intensity difference between translated image and its ground truth. A higher PSNR score means that the intensity of two images is closer.

Structural similarity index (SSIM): I2I uses SSIM to compute the perceptual distance between the translated image and its ground truth. The higher the SSIM is, the greater the similarity between translated image and the ground truth image. It uses luminance, contrast and structure as a measure of similarity.

Classification accuracy: This metric adapts a classifier pretrained on target domain images to classify the translated images. The intuition behind this metric is that a well-trained I2I network will generate outputs images that will easily classify this image from the target domain. A high accuracy indicates that our generative network has learned more deterministic patterns from the target domain.

Frechet inception distance (FID): The FID measures the distance of the distributions of synthesized images with the distribution of real images. A lesser FID score means a good performance by the generative network.

Chapter 4

Multiple Opinion Generation

4.1 Two Domain Mapping

4.1.1 U-Net Regression

As our first objective to achieve two domain I2I translation. I have used original U-Net neural network architecture with 2 modifications.

They are including Batch Normalization in every layer and using stride of 1 for convolution layers, otherwise architecture is same as proposed in original paper.

Here I have used 1) Expert A from MIT Adobe 5k dataset as paired supervised learning where input image is underexposed and ground truth is Expert A. I have employed Mean Square Error as loss function as it is observed to give good results. 2) Flower Recognition dataset that I converted to grayscale to get paired data, this pair of (grayscale, color) is used for supervised learning to showcase image colorization.

U-Net architecture is composed of encoder layers that learn the low-dimension representation of the input source domain image, it is then reconstructed to target domain output image. To improve and retain content information, connections are made from ith layer to (n-i)th layer. This allows information to pass from encoder layers to decoder layers. It also solves the issue of vanishing gradient problem which is faced by many DNNs. These are also helpful as they help in generating same structure outputs.

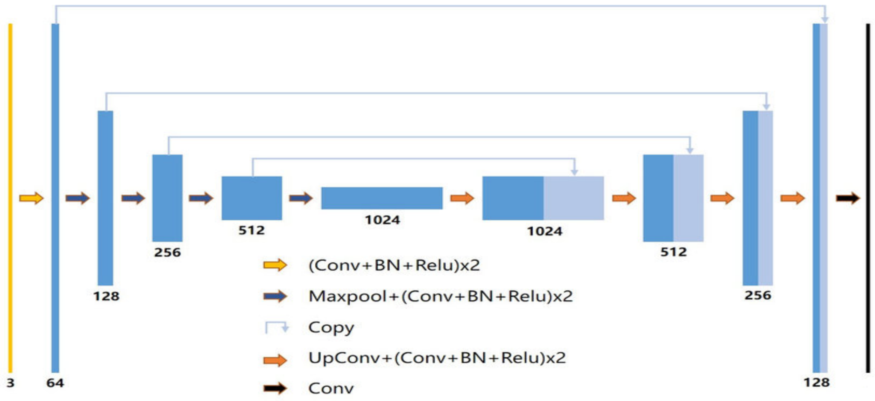
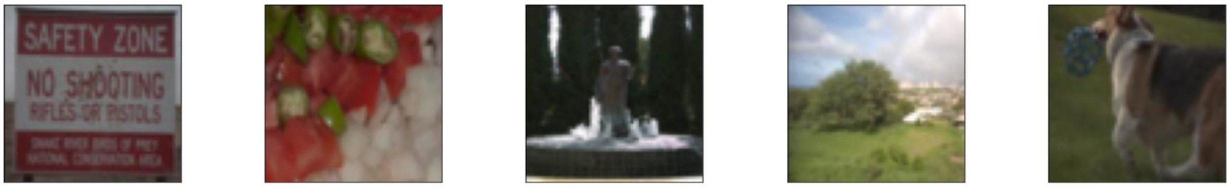


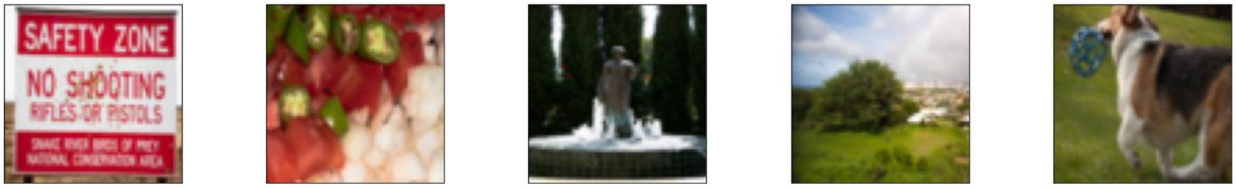
Figure 4.1: U-Net architecture

Results

Input:



Output (with corresponding PSNR):



31.26dB 30.58dB 29.46dB 28.63dB 31.82dB

Figure 4.2: Two-Domain mapping output and inputs

4.1.2 Neural Style Transfer

NST is an techniques where input are 3 images, content image, style image and Generated image. Here I am using pretrained VGG19 network to extract content information and style image and content images are passed through simple CNN. Content loss is calculated at last layer using MSE and style loss is calculated using correlation between features maps over all layers.

$$\mathcal{L}_{content}(\vec{p}, \vec{x}, l) = \frac{1}{2} \sum_{i,j} \left(F^l_{ij} - P^l_{ij}\right)^2 .$$

$$\mathcal{L}_{style}(\vec{a}, \vec{x}) = \sum_{l=0}^{L} w_l E_l \qquad E_l = \frac{1}{4 N_l^2 M_l^2} \sum_{i,j} \left(G^l_{ij} - A^l_{ij}\right)^2 \qquad G^l_{ij} = \sum_k F^l_{ik} F^l_{jk}.$$

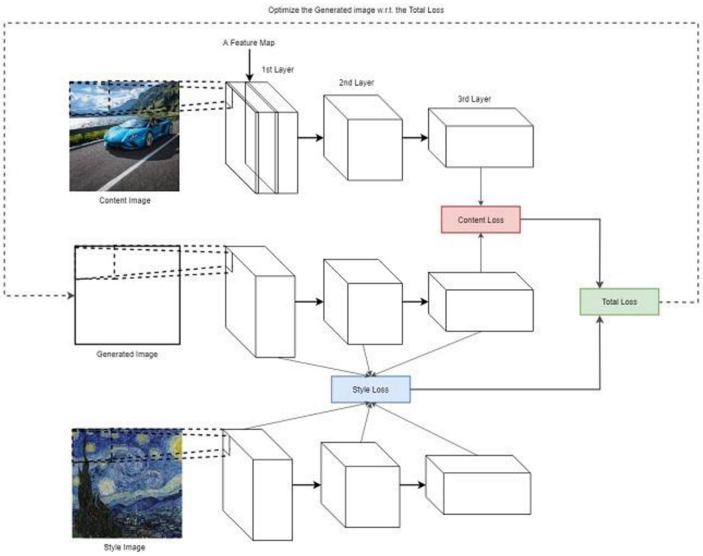


Figure 4.3: High-level overview of NST

$$\mathcal{L}_{total}(\vec{p}, \vec{a}, \vec{x}) = \alpha \mathcal{L}_{content}(\vec{p}, \vec{x}) + \beta \mathcal{L}_{style}(\vec{a}, \vec{x})$$

Total Loss is calculated by weighted adding Content Loss and Style Loss and this loss is used to updated network and generated image using Adam optimizer. Here we observe that using instance normalization instead of Batch normalization gives better result.

Results:

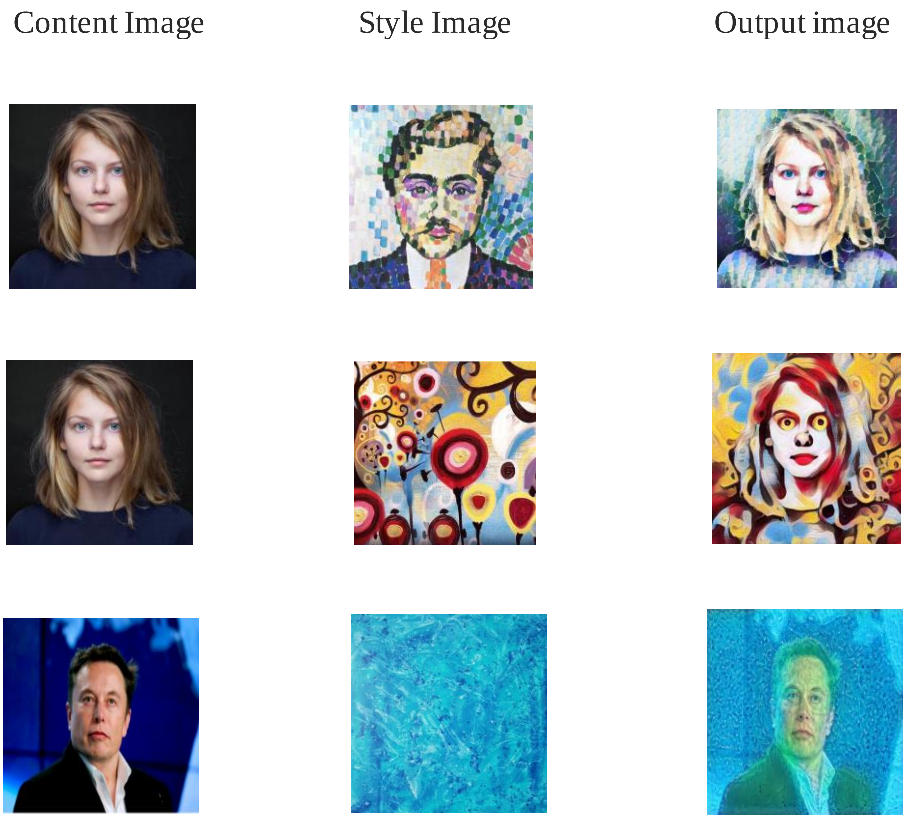


Figure 4.4: Results of NST

Comparison between NST and U-Net Comparison

1) NST network has to be trained every time we need to take output for new content image or style image whereas U-Net Regression Network has to be trained only once.

2) U-Net requires large dataset to train and generate good results, whereas NST requires only 2 images(content and style image) to train and generate good result.

4.2 Multi-Modal I2I Translation

4.2.1 BiCycleGAN

BiCycleGAN aims to generate a diverse set of output images given an input image. It is an extension to I2I translation model such as Conditional GAN based models such as pix2pix. Note that this uses 2 different neural networks: GAN and Encoder.

This model uses a variational auto-encoder. This is to learn low-dimensional representation of the images using an encoder network i.e., a probability distribution which has generated all the target domain images and then try to make this distribution as close as possible to gaussian distribution so that samples are easy to sample during testing. Input source image is mapped to output target image using encoded latent low dimension representation z using the generator network.

In next stage, Conditional GAN is used with image and z sampled from normal distribution N(z), this is fed to Generator to get output image. Output image to fed to encoder to get z' which is required to be as close to N(z). Loss Function is:

$$\begin{aligned} G^*, E^* = \arg\min_{G,E}\max_{D} \quad & \mathcal{L}_{\text{GAN}}^{\text{VAE}}(G,D,E) + \lambda\mathcal{L}_1^{\text{VAE}}(G,E) \\ & + \mathcal{L}_{\text{GAN}}(G,D) + \lambda_{\text{latent}}\mathcal{L}_1^{\text{latent}}(G,E) + \lambda_{\text{KL}}\mathcal{L}_{\text{KL}}(E), \end{aligned}$$

Where contribution is controlled by respective weight hyper-parameters. Bijection is achieved by latent code to output image and output image to latent code.

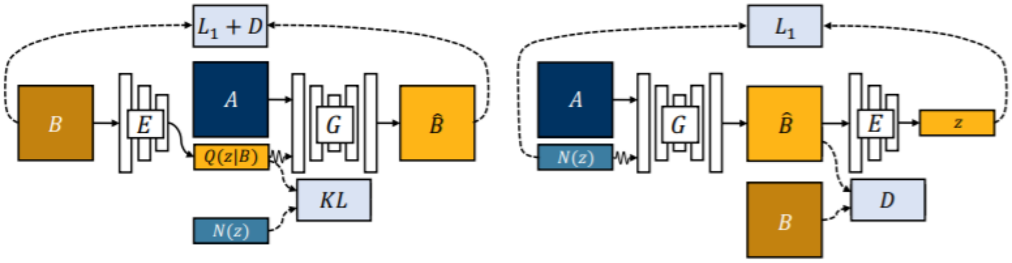
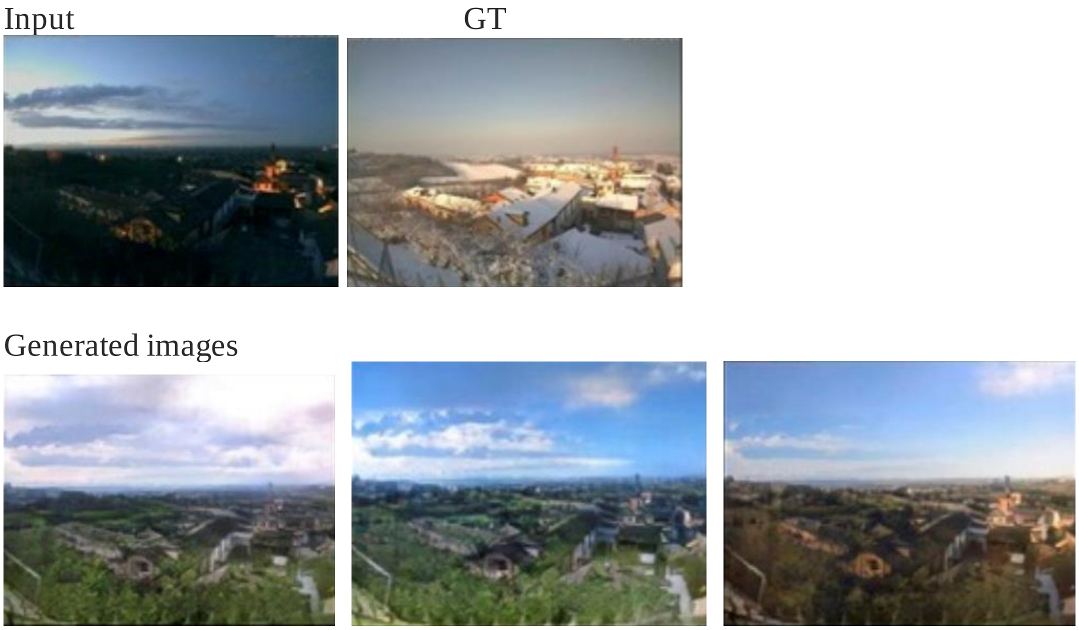


Figure 4.5: BicycleGAN training flow

Encoder usage : input image to latent code to reconstructed input image.
GAN usage: latent code to fake image to reconstructed latent code
Combining both above gives us BicycleGAN.

Results:

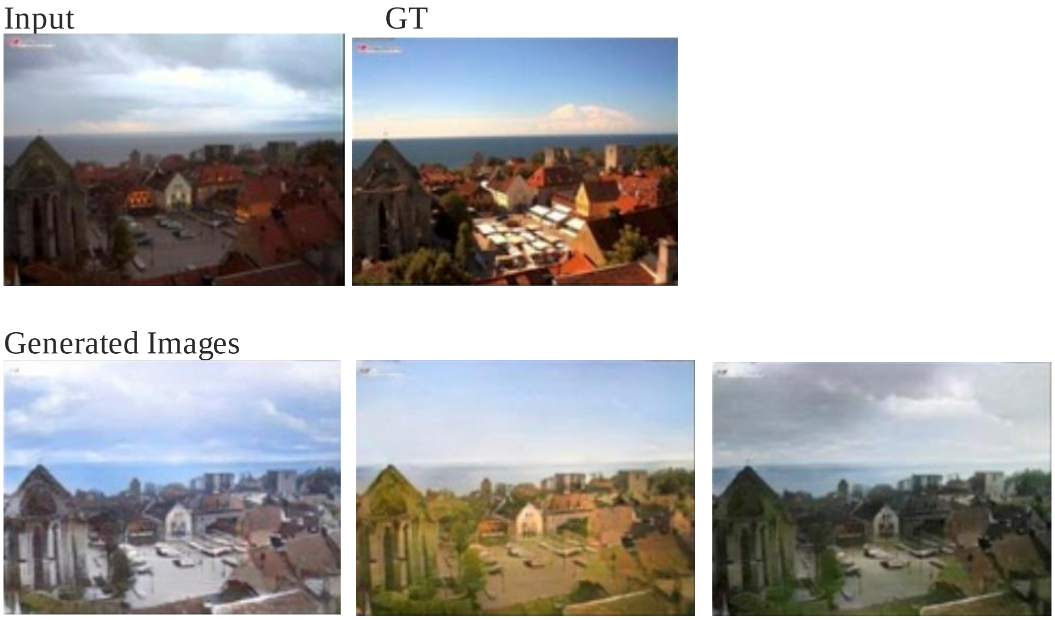
Night2day dataset


Input

GT

Generated images

Figure 4.6: Results of BycycleGAN


Input

GT

Generated Images

Figure 4.7: Results of BycycleGAN

4.3 Multi-Domain I2I Translation

4.3.1 U-Net Regression Single Encoder-Multiple Decoders

As discussed above, most important role of model in I2I translation issue is generating high quality output images from high quality input images

For multiple editor opinion generation for underexposed image enhancement, I am using U-Net like neural network architecture where connections are made from ith layer to (n-i) th layer, where n is total number of layers.

For generating multiple opinion, I have adopted multi decoder style architecture where single encoder and 5 decoders that generate 5 different good outputs for single underexposed image input.

I have used combined MSE loss function where I added MSE loss from every branch and used that to backpropagate and update weights of the network.

Objective Function: $L(\theta) = \sum(MSE_i)$ where i = {A, B, …till number of domains}

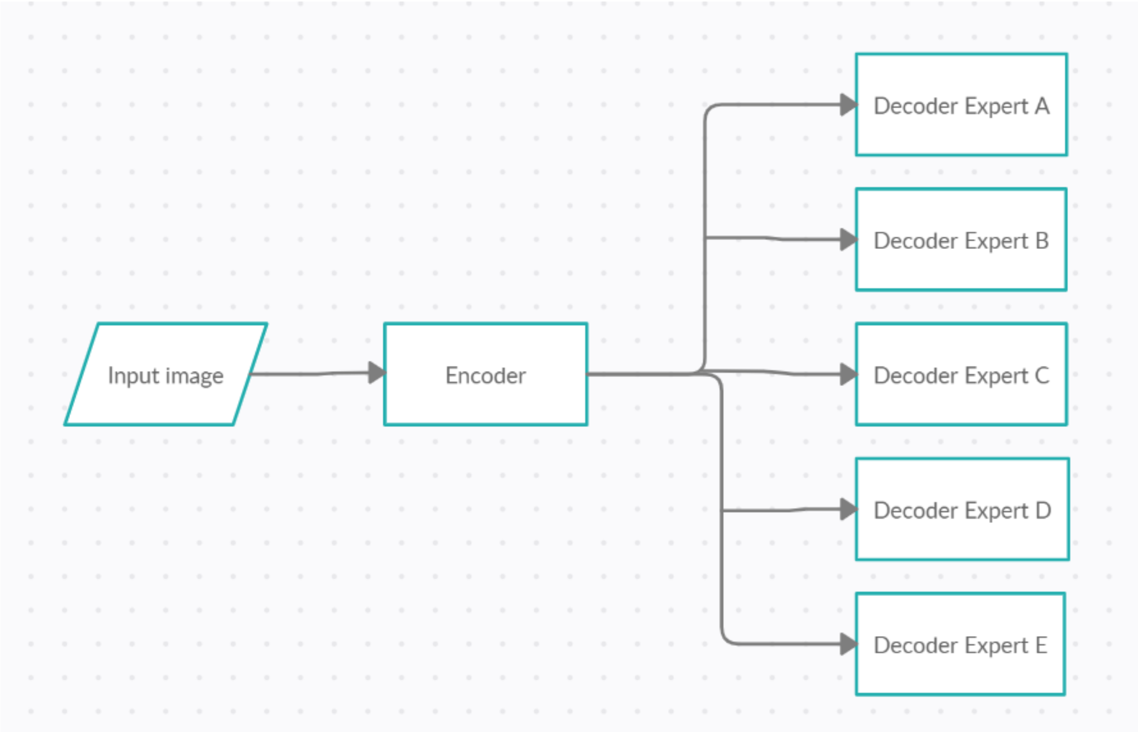


Figure 4.8**:** Single encoder-multi decoder U-Net style

4.3.2 U-Net Regression Single Encoder-Single Decoder(Improvement)

Above method has an obvious problem that it is using multiple decoders which increases the network size. To solve this, we have modified above architecture to single encoder and single decoder with conditioning being done with gaussian noise with different means. Below explains the process of training:

Output for Expert A is conditioned using gaussian noise of mean 1

Output for Expert B is conditioned using gaussian noise of mean 2

Output for Expert C is conditioned using gaussian noise of mean 3

Output for Expert D is conditioned using gaussian noise of mean 4

Output for Expert E is conditioned using gaussian noise of mean 5

Gaussian noise of size similar to image channel length is added to the latent code before giving input to the decoder. I have used combined MSE loss function where I added MSE loss from every Expert output and used that to backpropagate and update weights of the neural network.

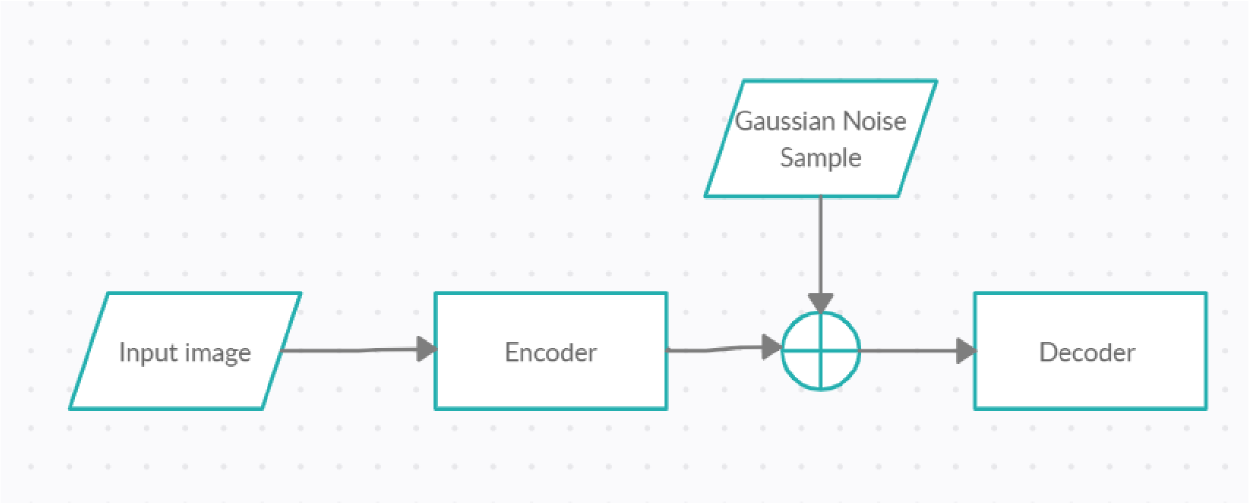
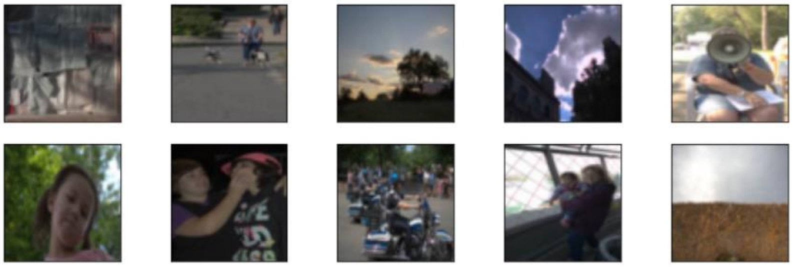


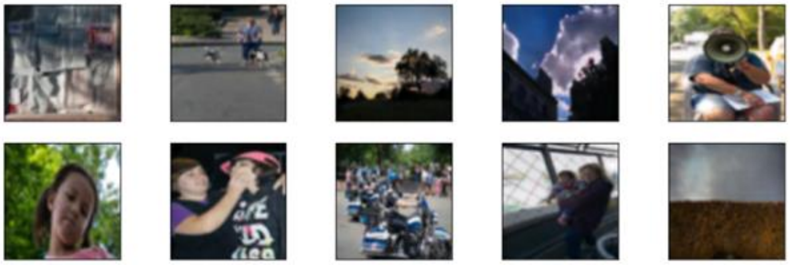
Figure 4.9: Single Encoder-single decoder conditioning with noise

Experimental Results

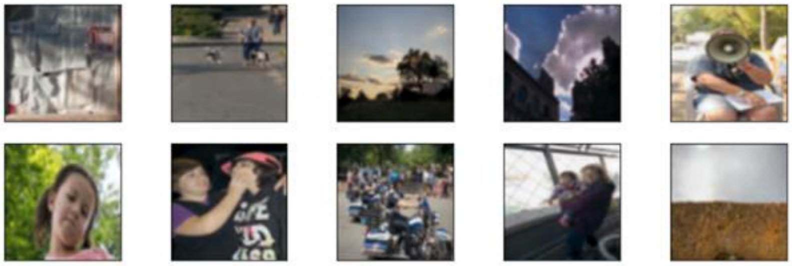
Input:

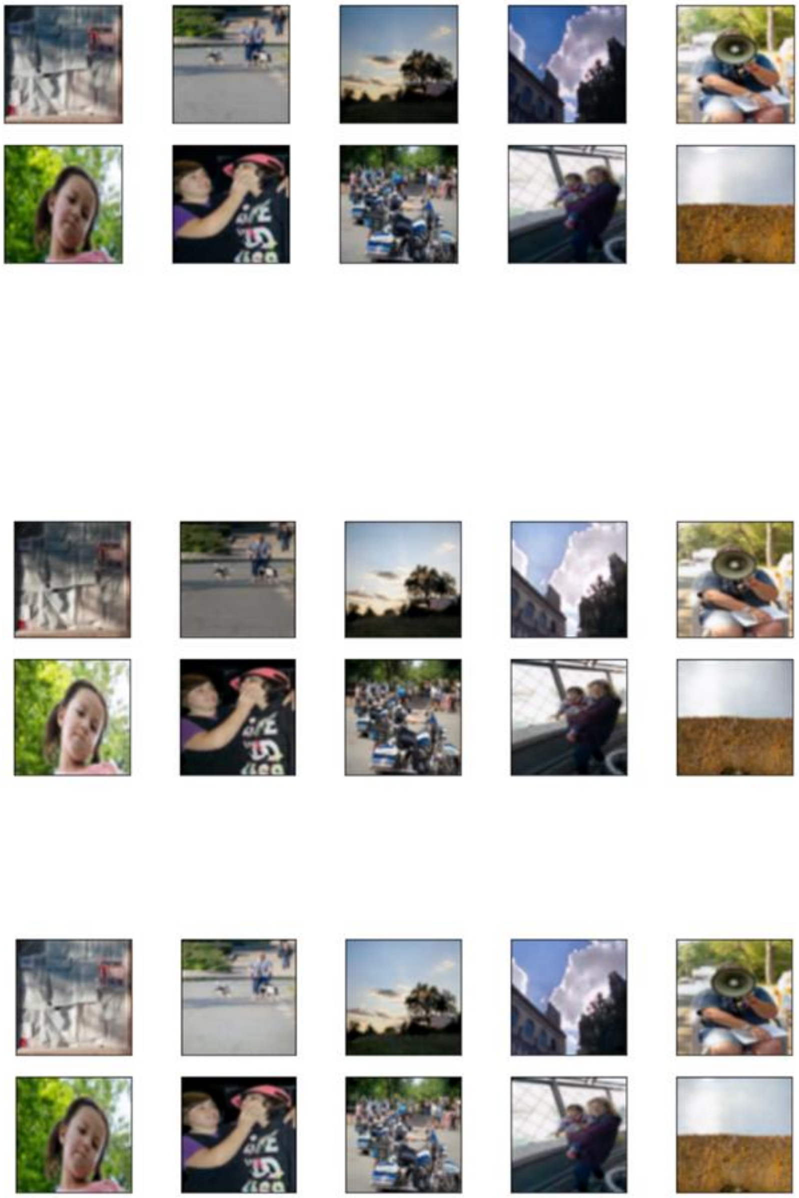


Expert A:



Expert B:





Expert C:

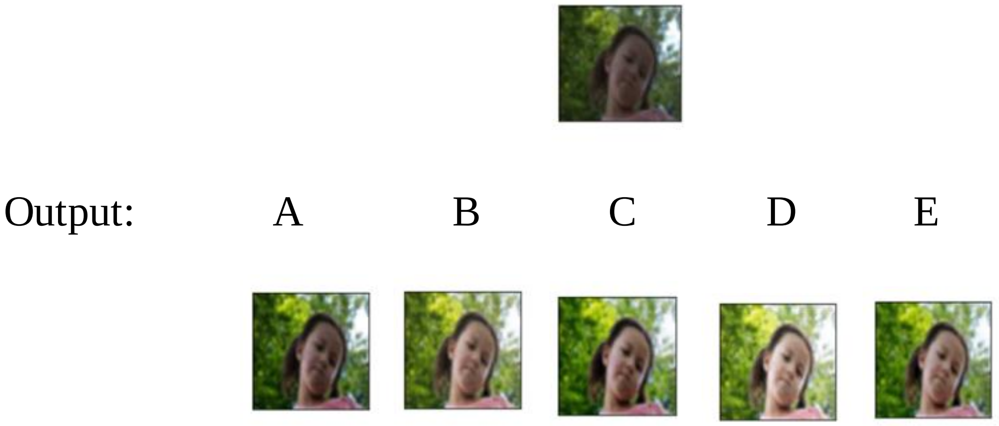
Expert D:

Expert E:

Figure 4.10: Objective-2 input and outputs

Comparison:

Input:



PSNR: 30.12dB 31.63dB 30.76dB 30.32dB 31.45dB

Figure 4.11: Comparison between different Expert Outputs

4.3.3 Conditional GAN

I have employed Conditional GAN architecture for generation of multiple opinion generation.

In an unconditioned generative model, there is no control on modes of the data being generated. However, by conditioning the model on additional information it is possible to direct the data generation process. Such conditioning could be based on class labels, or even on data from different modality.

I have employed training methodology from SRGAN architecture[9], traditional GAN architecture aim to minimize MSE loss. Resulting in an image having reduced high-frequency structural details and are visually not good to look at. By adding perceptual loss, high-frequency content are retained in more efficient manner in output generated image.

This perceptual loss is calculated as distance between feature maps generated by pretrained VGG network and our generator model. By introducing this loss, we try to minimize this distance and as a result high-frequency details are retained in output generated image.

Loss function: weighted sum of perceptual loss and adversarial loss

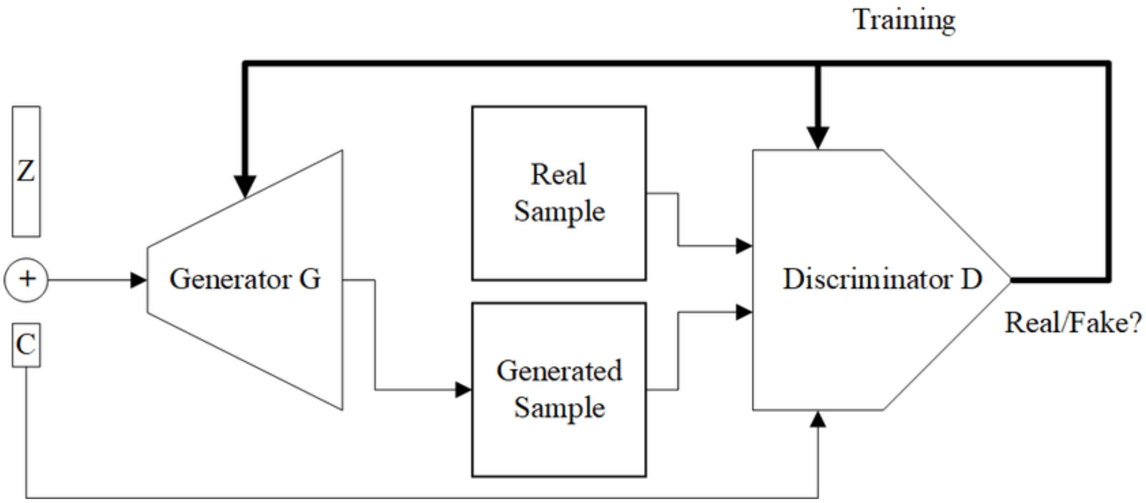
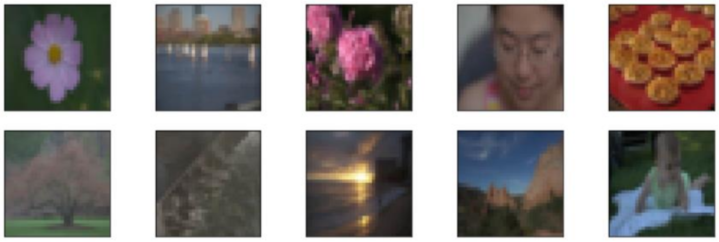


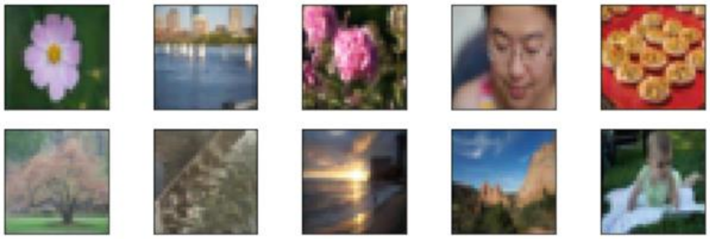
Figure 4.12: Conditional GAN

Experimental Results

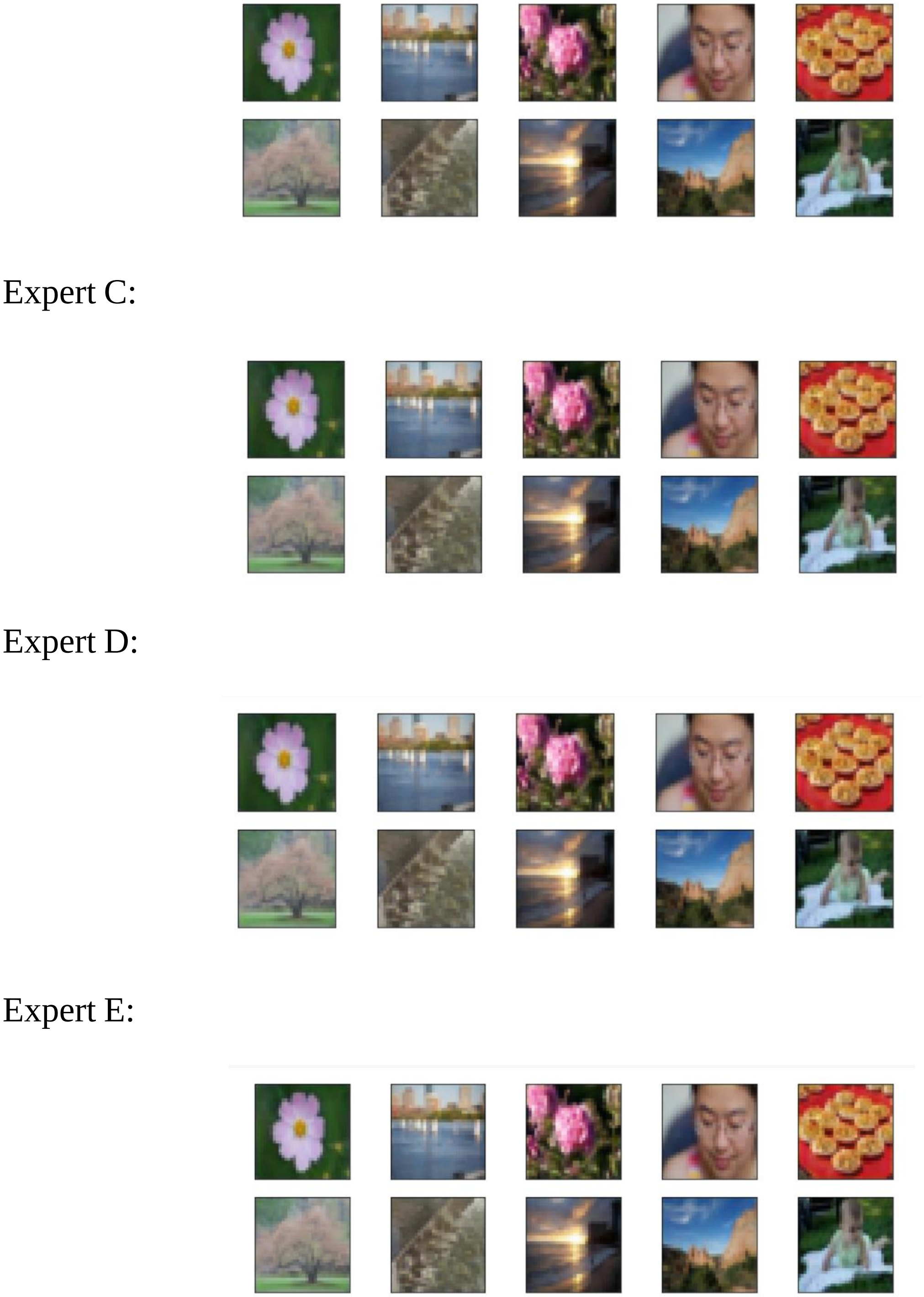
Input:



Expert A:



Expert B:



Expert C:

Expert D:

Expert E:

Figure 4.13: Multi-Domain Output

As we can observe, we fail to produce satisfactory results for multimodal Image to Image translation task as all outputs are same visually for different conditions.

4.3.4 SingleGAN

This is GAN architecture that uses single Generator network and multiple Discriminator network with domain code injection using Central Biasing Instance Normalization [9], [10] to achieve multi-modal I2I translation. This GAN architecture has a only one generator and multiple number of discriminators to achieve unsupervised I2I translation.

$$\mathrm{CBIN}(x_i) = \frac{x_i - \mathrm{E}[x_i]}{\sqrt{\mathrm{Var}[x_i]}} + \tanh(f_i(z)),$$

In above CBIN formulation xi is the feature map, and z is the domain latent code that need to be injected in neural network. Here, fi is the affine transformation /fully connected layer. The goal of CBIN is to change the distributions of the input feature maps according to the learnable parameters and the domain latent code, this enables the domain latent code to condition generative network for generating different domain outputs.

Loss Function: as we have only one generator network that has to generate multi-domain images, we compute adversarial loss with respect to each domain using different discriminator. Each discriminator classify image as real or fake from its own domain.

$$\begin{aligned}\mathcal{L}_{adv}(G, D_{\mathrm{A}}) =& \mathbb{E}_{\chi_{\mathrm{A}}}[\log(D_{\mathrm{A}}(x_{\mathrm{A}}))] + \mathbb{E}_{\chi_{\mathrm{B}}}[\log(1 - D_{\mathrm{A}}(G(x_{\mathrm{B}}, z_{\mathrm{A}})))],\\ \mathcal{L}_{adv}(G, D_{\mathrm{B}}) =& \mathbb{E}_{\chi_{\mathrm{B}}}[\log(D_{\mathrm{B}}(x_{\mathrm{B}}))] + \mathbb{E}_{\chi_{\mathrm{A}}}[\log(1 - D_{\mathrm{B}}(G(x_{\mathrm{A}}, z_{\mathrm{B}})))].\end{aligned}$$

Above adversarial loss is not sufficient to I2I translation as it is highly under-constrained mapping which leads to a very common issue in GANs called mode collapse. As there is no paired information available, it can lead to many mappings. To reduce the irrelevant mappings, the cycle-consistency loss is used in addition to adversarial loss in the training stage. Cycle consistency loss forces the generator to shrink mapping space by forcefully learning relevant mappings.

$$\begin{aligned}\mathcal{L}_{cyc}(G) =& \mathbb{E}_{\chi_{\mathrm{A}}}\left[\, \|x_{\mathrm{A}} - G(G(x_{\mathrm{A}}, z_{\mathrm{B}}), z_{\mathrm{A}})\|_1 \,\right] +\\ & \mathbb{E}_{\chi_{\mathrm{B}}}\left[\, \|x_{\mathrm{B}} - G(G(x_{\mathrm{B}}, z_{\mathrm{A}}), z_{\mathrm{B}})\|_1 \,\right],\end{aligned}$$

where ||.|| denotes L1 norm. The final Loss function is given by:

$$G^* = arg \min_{G} \max_{D_A, D_B} \sum_{i \in \{\mathrm{A},\mathrm{B}\}} \mathcal{L}_{adv}(G, D_i) + \lambda_{cyc} \cdot \mathcal{L}_{cyc}(G)$$

where λ_{cyc} is hyperparameter that controls the importance of cycle consistency loss in Total loss function.

Experimental Results:

One2one mapping:

Under-Exposed Image Enhancement: Dataset used: MIT fiveK dataset

Input Output

Figure 4.14: Result of SingleGAN for under-exposed image enhancement

Photo2sketch Dataset used: self-downloaded pencil sketch files

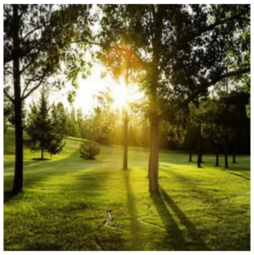
Input Output

Figure 4.15: Result of SingleGAN for photo2sketch

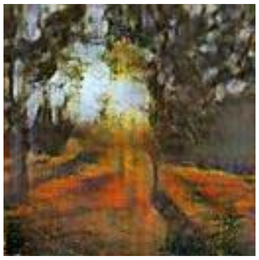
Multi-Domain:

Dataset used: photo2art, monet2photo, cezanne2photo, vangogh2photo

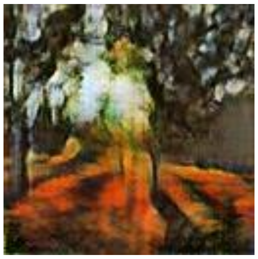
Input



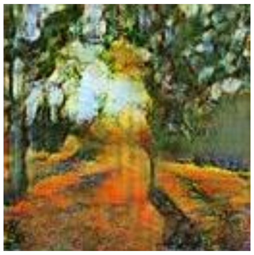
Monet



Cezanne



Vangogh



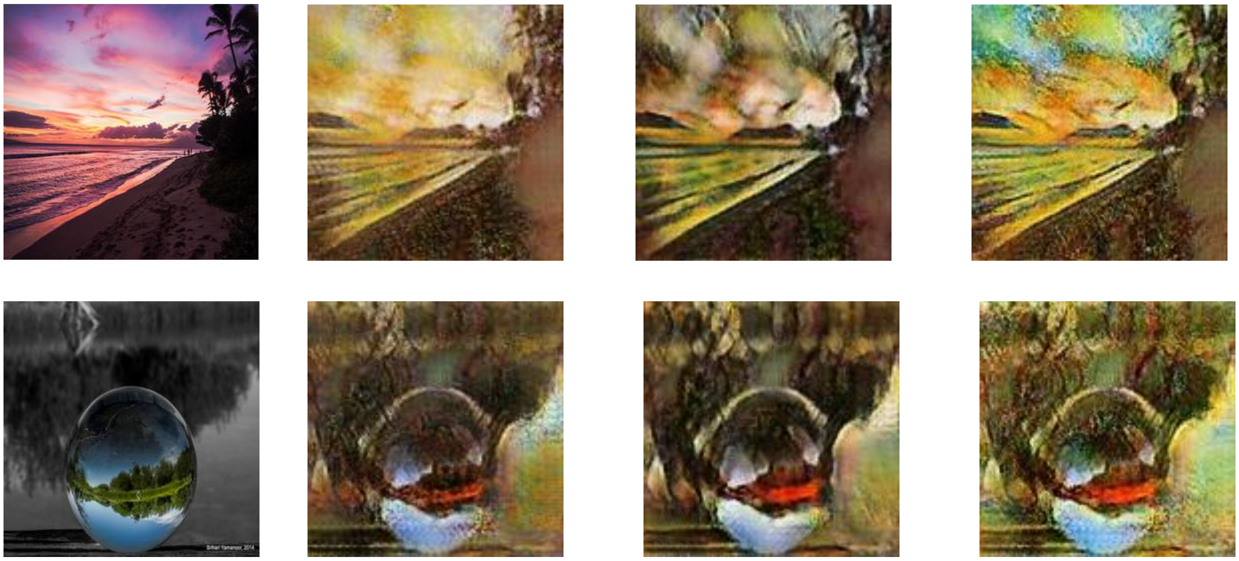


Figure 4.16: Results of SingleGAN for photo2art

Dataset used: MIT fiveK dataset(Under-Exposed Image Enhancement)

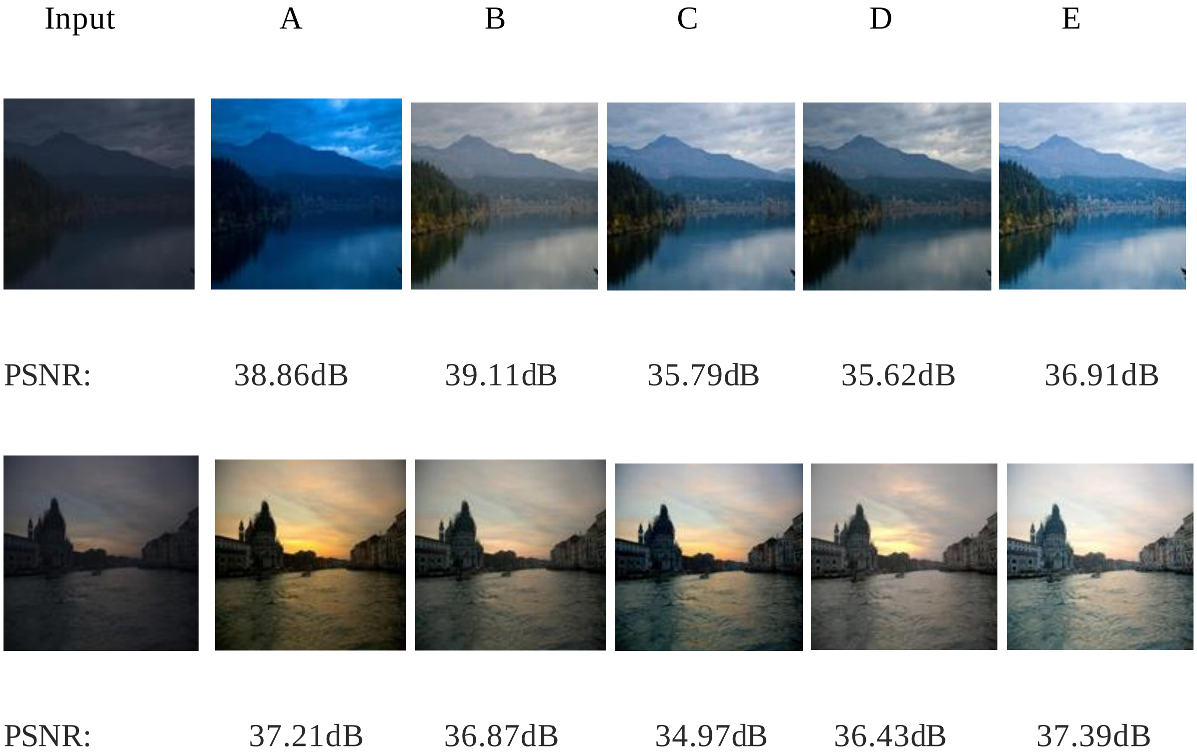


Figure 4.17: Results of SingleGAN for multi-domain under-exposed image enhancement

Chapter 5

Future Works

As we observed SingleGAN neural network architecture, number of Discriminators is equal to number of target domains we are want to map from source domain. This is one of big disadvantages of that network. As number of Domains increase, more memory is utilized, training time will increase.

To improve upon this architecture, I am working on GAN architecture called SoloGAN. In SoloGAN, content and style are separately learned by content encoder and style encoder. And Single Generator and Single Discriminator is used. To eliminate the need of multiple Discriminator, it is modified to output two results ie real/fake of input image and Domain label of input image. GAN will be conditioned on style vector concatenated with domain latent code. Central Biased Instance Normalization scheme will be followed to inject Domain/Style label in the GAN networks.

Finally, last objective is to interpolate the transformation space so as to generate distinct intermediate results along the transformation from source domain image to target domain image. For this TrGAN has been proposed for two domain mapping. My aim to extent it for multi-domain.

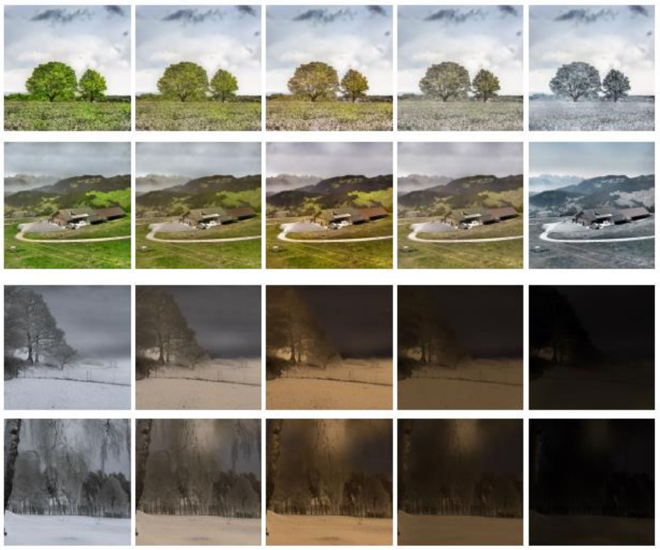


Figure 5.1: Some examples of interpolation of transformations

Chapter 6

Conclusion

The results shown in the report are indicative of two-Domain mapping achieved through U-Negt regression network, multi-modal mapping achieved through BicycleGAN architecture. And finally multi-Domain mapping achieved through U-Net Regression Network and SingleGAN architecture. Although Conditional GAN results were not good due to inherent issue with GAN called as mode collapse. It was resolved by introducing cycle consistency loss and multiple discriminators in SingleGAN architecture.

This showed some wide applications in above mentioned Objectives such as under-exposed image enhancement, photo2art, photo2sketch. Neural Style Transfer was also developed and comparison was done wrt to U-Net Regressor Network looking at its advantages and disadvantages.

In future works, I will try to improve upon SingleGAN architecture and develop and integrate network capable to interpolating Transformation space.

References

[1] Yu, X., Cai, X., Ying, Z., Li, T., & Li, G. (2018). SingleGAN: Image-to-Image Translation by a Single-Generator Network using Multiple Generative Adversarial Learning. In *Asian Conference on Computer Vision*.

[2] Zhu, J.Y., Park, T., Isola, P., & Efros, A. (2017). Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networkss. In *Computer Vision (ICCV), 2017 IEEE International Conference on*.

[3] Isola, P., Zhu, J.Y., Zhou, T., & Efros, A. (2017). Image-to-Image Translation with Conditional Adversarial Networks. In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*.

[4] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In International Conference on Medical.

[5] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Advances in neural information processing systems, pages 2672–2680, 2014.

[6] Ian Goodfellow. Nips 2016 tutorial: Generative adversarial networks. 2016.

[7] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In International Conference on Machine Learning, 2015.

[8] M. Mirza and S. Osindero. (2018). ‘‘Conditional generative adversarial nets.’’ [Online]. Available: https://arxiv.org/abs/1805.08657

[9] Dmitry Ulyanov and Andrea Vedaldi and Victor S. Lempitsky (2016). Instance Normalization: The Missing Ingredient for Fast Stylization. *CoRR, abs/1607.08022.*

[10] Xiaoming Yu and Zhenqiang Ying and Ge Li (2018). Multi-Mapping Image-to-Image Translation with Central Biasing Normalization. *CoRR, abs/1806.10050.*

[11] Leon A. Gatys and Alexander S. Ecker and Matthias Bethge (2015). A Neural Algorithm of Artistic Style. *CoRR, abs/1508.06576.*

[12] T. Park, M.-Y. Liu, T.-C. Wang, and J.-Y. Zhu, “Semantic image synthesis with spatially-adaptive normalization,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), June 2019.

[13] T. Kim, M. Cha, H. Kim, J. K. Lee, and J. Kim, “Learning to discover cross-domain relations with generative adversarial networks,” arXiv preprint arXiv:1703.05192, 2017.

[14] Z. Yi, H. Zhang, P. Tan, and M. Gong, “Dualgan: Unsupervised dual learning for image-to-image translation,” in Proceedings of the IEEE international conference on computer vision, 2017, pp. 2849–2857.

[15] Ming-Yu Liu and Thomas Breuel and Jan Kautz (2017). Unsupervised Image-to-Image Translation Networks*. CoRR, abs/1703.00848.*

[16] Xun Huang and Ming-Yu Liu and Serge J. Belongie and Jan Kautz (2018). Multimodal Unsupervised Image-to-Image Translation. *CoRR, abs/1804.04732*.

[17] Z. Zheng, C. Wang, Z. Yu, N. Wang, H. Zheng, and B. Zheng, “Unpaired photo-to-caricature translation on faces in the wild,” Neurocomputing, vol. 355, pp. 71–81, 2019

[17] Jun-Yan Zhu and Taesung Park and Phillip Isola and Alexei A. Efros (2017). Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks. *CoRR, abs/1703.10593*.

[18] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” arXiv preprint arXiv:1412.6980, 2014.

[19] H.-Y. Lee, H.-Y. Tseng, J.-B. Huang, M. Singh, and M.-H. Yang, “Diverse image-to-image translation via disentangled representations,” in Proceedings of the European conference on computer vision (ECCV), 2018, pp. 35–51.

[20] Y. Choi, M. Choi, M. Kim, J.-W. Ha, S. Kim, and J. Choo, “Stargan: Unified generative adversarial networks for multi-domain image-toimage translation,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2018.

[21] L. Hui, X. Li, J. Chen, H. He, and J. Yang, “Unsupervised multidomain image translation with domain-specific encoders/decoders,” in 2018 24th International Conference on Pattern Recognition (ICPR), 2018, pp. 2044–2049.

[22] P. Esser, E. Sutter, and B. Ommer, “A variational u-net for conditional appearance and shape generation,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2018