Exploring CycleGAN and potential improvements for Unpaired Image-to-Image Translation

Karan Pardasani
kp955@scarletmail.rutgers.edu
Rutgers University
Department of Computer Science

Animesh Sharma
animesh.sharma@rutgers.edu
Rutgers University
Department of Computer Science

Jaini Patel
jp1891@scarletmail.rutgers.edu
Rutgers University
Department of Computer Science

Harish Udhay
harish.udhayakumar@rutgers.edu
Rutgers University
Department of Computer Science

Abstract

The class of algorithms in computer vision where the aim is to learn the mapping between an input image and output image is called image to image translation. There are two class of algorithms - paired and unpaired image to image translation. The CycleGAN architecture is used to train the model to learn the mapping using unpaired image to image translation. The goal for this training is to learn the mapping $G : X \rightarrow Y$ such that the transformation of the input image, $G(X)$ are very close to the images present in the target domain. Quantitative and Qualitative results are presented to evaluate the performance of the CycleGAN to learn the transformation from real to prerecorded synthetic images and vice versa, and also from real to live synthetic images and vice versa. After completing the training on standard CycleGAN model, we introduce some enhancements in the CycleGAN model. We train the enhanced model on same pairs of datasets that we train in case of standard CycleGAN model.

1. Introduction

The goal of Image-to-image translation (I2I) is to transfer images from a one domain to a another while preserving the image representations. I2I has grabbed increasing attention and made massive progress in recent years due to its wide range of applications in many problems, such as image synthesis, segmentation, style transfer, restoration, and pose estimation. In this project, we experiment and present an overview of the I2I works developed in recent years. We will analyze the existing key machine learning frame works (GANs). Additionally, we will experiment novel approach to enhance existing techniques.

2. Related Work

2.1. Recollection: Task 1

In Task 1, we implemented the basic GAN model with Generator architecture as described in [3] and implemented 70x70 PatchGAN for the Discriminator architecture as described in [2]. The Discriminator identifies whether the generated image belongs to the target distribution. The Generator uses Adversarial Loss to incorporate the effect of Discriminator on Generator weights. For this task, we performed the Qualitative Evaluation and Quantitative Evaluation by calculating Frechet Inception Distance [1, 5] and Inception Score [4]. Fig. 1 displays the output images generated from the model using real pizza dataset. From the images, we can conclude that in the generation of real images, the GAN is able to generate the correct shape and place the topping in the good locations. Also, the size of the toppings are good in relative to the size of the pizza. The model is not able to perform well in synthetic pizza generation. The shape is not proper and also the number of toppings are not consistent with the input image. Fig. 2 displays the output images generated from the model using pre-recorded synthetic pizza datasets.

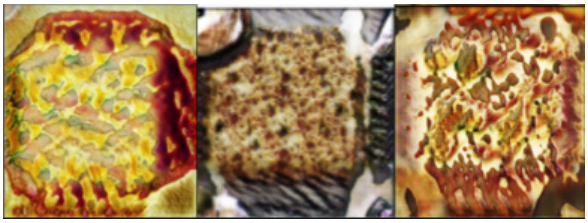

Figure 1. Sample generated real pizza images by GAN

Figure 2. Sample generated synthetic pizza images by GAN

2.2. Recollection: Task 2

In Task 2, we focused on paired image-to-image translation and trained Pix2PixGAN model as described in [2] on Dayton dataset. The Generator uses Adverserial Loss to incorporate the effect of PatchGAN Discriminator on Generator weights. For this task, we performed the Qualitative Evaluation and Quantitative Evaluation by calculating IS and FID scores. Fig. 3 and Fig. 4 display the generated output images for overhead to street view translation and street view to overhead translation, respectively. We can conclude that in the generation of street view images, the GAN is able to generate different shapes of house, trees, roads and pavements. The visual acuity can distinguish between different objects but the distinction is not very clear. The model also has decent performance in generating overhead images. The model is able to correctly generate trees, roads and buildings. But, the distinction of different objects is not as apparent as the distinction in generated street view images.

Figure 3. Sample generated images of Street view

Figure 4. Sample generated images of Overhead view

3. Dataset

For the purpose of training CycleGAN and enhanced CycleGAN models, we have used datasets from three domains; pre-recorded synthetic pizza dataset, real pizza dataset and live synthetic pizza dataset. Our goal with unsupervised image-to-image translation is to translate the images from one domain to another, and so, we perform two types of translations; real to synthetic and synthetic to real, using two different synthetic datasets and one real pizza dataset. The total images in real pizza dataset train–test-val sets are 6688-836-824. The total images in pre-recorded synthetic dataset train-test-val sets are 13321-1538-1481.

To generate the live synthetic dataset, we have used 13 different types of toppings with each having approximately 5 to 6 images, 5 different types of pizza base images, and 6 different background images. We have applied alpha channel to the toppings and base images. Total number of images in the live synthetic dataset train-test-val sets are 13321-1538-1481.

We have pre-processed the images by up-scaling the image, then using random crop and finally normalizing it, for the train dataset. The final dimensions of all input images are 256×256 pixels.

Figure 5. Example of components of Live Synthetic images

Figure 6. Sample images from live synthetic pizza dataset

Figure 7. Sample images from real pizza dataset

Figure 8. Sample images from prerecorded Synthetic pizza dataset

4. Cycle GAN

We have implemented unpaired image-to-image translation using CycleGAN model as in the paper [6], on pizza datasets.

4.1. Methodology

The aim of CycleGAN is to learn a mapping $G : X \rightarrow Y$as such the distribution of images from $G(X)$ is indistinguishable from the distribution Y using an adversarial loss. As this mapping is highly under-constrained, we couple it with an inverse mapping $F : Y \rightarrow X$ and establish a cycle consistency loss to push $F(G(X)) \approx X$ (and vice versa).

4.2. Formulation

Adversarial Loss:
This loss is used to learn the distribution of the target domain so that the Generator can generate images from source domain to the target domain.

$$Loss\_advers(G, D_y, X, Y) = \frac{1}{m}\Sigma(1 - D_y(G(x)))^2$$

$$Loss\_advers(F, D_x, X, Y) = \frac{1}{m}\Sigma(1 - D_x(F(y)))^2$$

Cycle Consistency loss:
This loss is used to make consistent information when we translate image from the source domain to the target domain.

$$Loss\_cyc(G, F, X, Y) = \frac{1}{m}[(F(G(x_i))-x_i)+(G(F(y_i))-y_i)$$

Identity loss:
This loss function is used to preserve information for the images that are already present in the target domain.

$$\begin{aligned} Loss\_identity(G, F) = {} & E_{y\sim p_data}(y)[||F(y) - y||_1] \\ & + E_{x\sim p_data}(x)[||F(x) - x||_1] \end{aligned}$$

Overall Loss:

$$\begin{aligned} Overall\ Loss = {} & Loss\_advers(G, D_y, X, Y) \\ & + Loss\_advers(F, D_x, X, Y) \\ & + Loss\_cyc(G, F, X, Y) \\ & + Loss\_identity(G, F) \end{aligned}$$

4.3. Architecture

The architecture of CycleGAN has 2 Generator networks $(GandF)$ and two discriminator networks, (D_XandD_Y). (fig. 9)

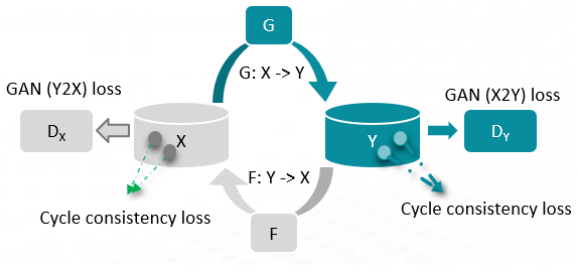

Figure 9. Cycle GAN architecture

Generator: c7s1-64, d128, d256, R256, R256, R256, R256,

R256, R256, R256, R256, R256, u128, u64, c7s1-3.
Also, for the discriminator, we use 70 × 70 patchGAN discriminator and the architecture is as follows:
C64-C128-C256-C512

4.4. Training

The CycleGAN was trained for two combinations of datasets as follows:

1. Live synthetic pizza dataset as one Domain and real pizza as the another. The live synthetic pizza dataset corresponds to images that were generated with toppings, pizza bases and background that were manually collected.
2. Pre-recorded Synthetic pizza dataset as first Domain and real pizza dataset as another.

We have trained the model for 50 epochs and the FID scores are shown in the Fig 14, 15, 16 and 17.

4.5. Qualitative Evaluation

We perform Qualitative Evaluation of the models by manually checking the quality of the generated images. Here are few of the generated images for both the training combinations as mentioned above. As we can see from Fig. 10, 11, 12 and 13, the quality of translated images from one domain to another is decent for all the combinations. There are some visible differences in the background texture, toppings texture, size and shape of the image objects in the translated images from the original images. The reconstructed images are also similar to the original input images, but with some difference in proportion of the toppings and pizza sizes.

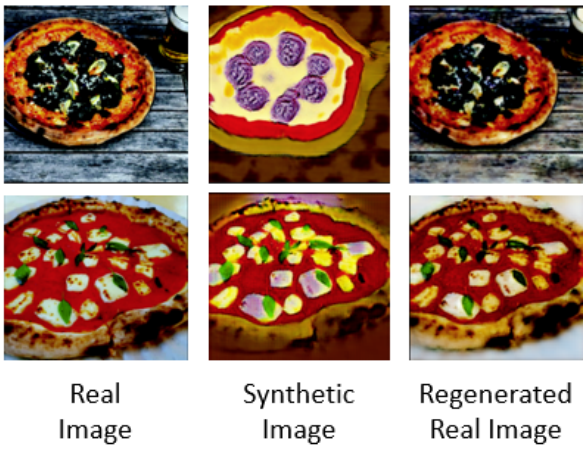

Figure 10. Translated Real images to synthetic pizza images and reconstructed it back to real Pizza from pre-recorded dataset

4.6. Quantitative Evaluation

For quantitative evaluation we use the following evaluation metric:

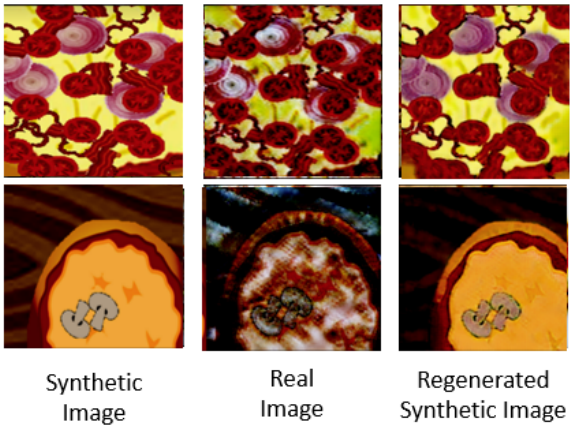

Figure 11. Translated Synthetic images to Real pizza images and reconstructed it back to Synthetic Pizza from pre-recorded dataset

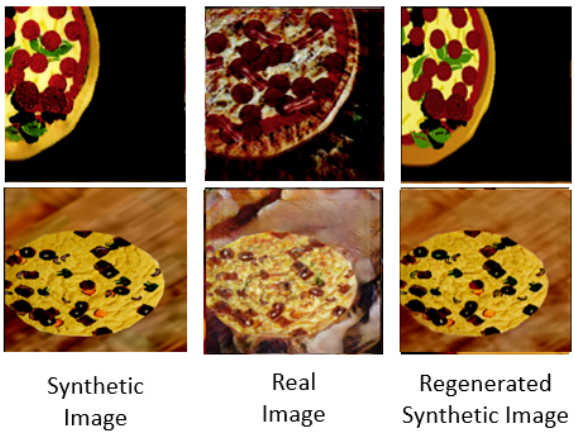

Figure 12. Translated Syhthetic images to real pizza images and reconstructed it back to Syhthetic Pizza from live dataset

Figure 13. Translated real images to synthetic pizza images and reconstructed it back to real Pizza from Live dataset

FID - The Frechet Inception Distance (FID) , is an evaluation metric that is used for evaluating the quality of generated images and specifically developed to evaluate the performance of generative adversarial networks. For the Base CycleGAN, we observed that there was decrease in FID scores with increase in number of epochs, but after certain

number of epochs, the FID levels increased a bit, showing the possible effects of mode collapse. Fig. 14, 15, 16 and 17 show the FID scores for the base CycleGAN model.

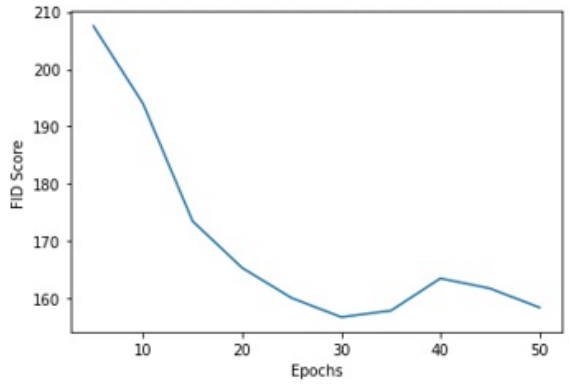

Figure 14. FID score for real images versus reconstructed real images from live pizza dataset

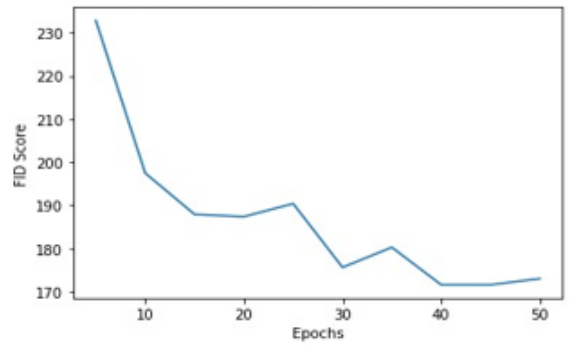

Figure 15. FID score for real images versus reconstructed real images from recorded pizza dataset

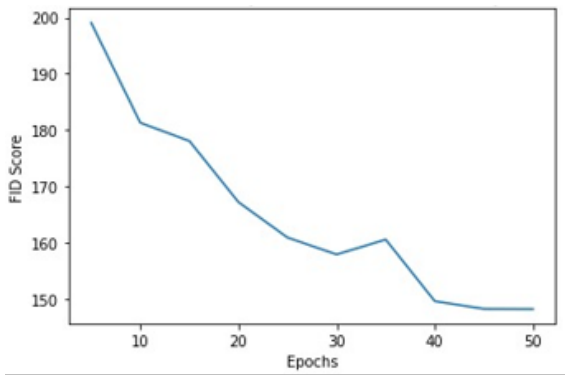

Figure 16. FID score for synthetic images versus reconstructed synthetic images from live pizza dataset

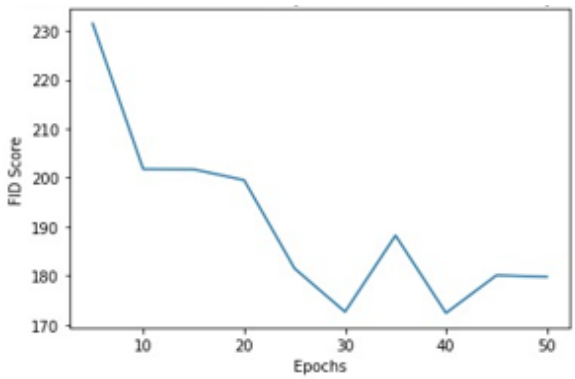

Figure 17. FID score for synthetic images versus reconstructed synthetic images from recorded pizza dataset

5. Improvements to Cycle GAN

5.1. Need for Improvements

While training the base Cycle GAN we noticed the following drawbacks:

- Cycle GAN does not produce good results if the difference between the two domains is too large. We believe this is partly due to the fact that the cycle consistency loss is very rigid and does a pixel-by-pixel comparison, which limits it's flexibility.

- The model suffered from mode collapse after some epochs. This happens when there is vanishing gradient in the discriminator. So, in this case, the weights of the discriminator stop updating and the generator keeps on getting better. So, generator will continue generating a specific type of images that will always fool the discriminator.

5.2. Introducing Global Discriminator

The standard architecture of Cycle GAN uses two PatchGAN discriminators. In addition to the two PatchGAN discriminators, we are introducing two Global Discriminators to the model. The PatchGAN Discriminator classifies whether each patch of the image belongs to the target domain distribution or not. The Global Discriminator classifies by looking at the whole image as a whole and outputs the probability of the image being in the distribution or not.

The image Fig. 18 shows the architecture of the enhanced CycleGAN model.

PatchGAN discriminator is used because it focuses on local features (finer details) and independently looks at local patches to decide whether the image is real or fake. Also, using only PatchGAN discriminator is insufficient since the patches of the images are dependant on each other. So, we are using the Global Discriminator to consider the relation

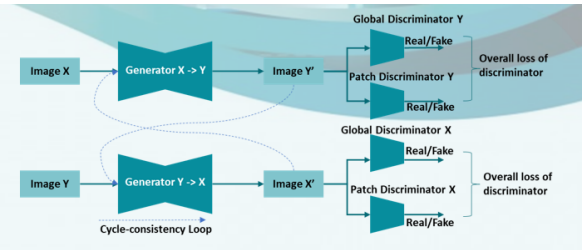

Figure 18. Cycle GAN architecture

between the patches of images. Also, PatchGAN is important because in considering the global features, the local information is sometimes overlooked. Hence, including both PatchGAN and Global Discriminator should allow the model to benefit from both types of Discriminators and make the model more robust against mode collapse.

5.3. Considering FID score in Generator Loss function

FID score is the similarity score used to measure the closeness of two datasets of images. Rather than directly comparing images pixel by pixel, the FID compares the mean and standard deviation of one of the deeper layers in the Inception v3 network. These layers are closer to output nodes that correspond to real-world objects and further from the shallow layers near the input image. As a result, they tend to mimic human perception of similarity in images. FID scores are inversely proportional to the similarity of the image sets. So, we consider FID score in loss function. Since the model tries to navigate towards minimising loss function, the model will have low values of FID. Now, there are three loss functions in the standard GAN model Adversarial Loss, Identity Loss and Cycle Consistency Loss and we add two more for the Enhanced GAN model:

- **Adversarial Loss**: This loss penalises if the Discriminator is able to differentiate between the generated images and the images from the target domain.
- **Identity Loss**: To calculate this loss, we take an image from the target domain and generate the image using the generator G. Since the input is in the target domain, the output image should ideally be the same as input image. So, this loss penalises if the generated image is very far from the input image, if input image is from the target domain. We consider L1 loss to penalise the divergence of input image from the target domain and the output image.
- **Cycle Consistency Loss**: To calculate this loss, we take the image from the source domain and generate the output image that should be in target domain. Also, we use this output image to reconstruct the input image using another Generator that maps the output domain to input domain. The reconstructed image should be very close to output image. Hence, this loss function penalises if the input image is very far from the reconstructed image. We consider L1 loss to penalise the divergence of input image and the reconstructed image.
- **FID for Identity Loss**: The idea is that in addition to the L1 Loss from the Base CycleGAN model, using the FID between the input and output images will produce a more robust identity loss to penalise the divergence of the input image from the target domain and the output image. We scale the FID score down by a user-tunable parameter and then take the weighted mean of the L1 Loss and the FID score as the net identity loss.
- **FID for Cycle Consistency Loss**: Now, in the Cycle Consistency Loss, it was discussed that we use L1 loss to penalise the divergence of the input image from the source domain and the reconstructed image. In this case, we are considering the FID between the two images. Penalising the Generator on large FID scores between the input from source domain and the corresponding reconstructed image will make the images similar to each other.

The Final Loss function of the Enhanced Generator Model is:

$$\begin{aligned} Updated\,Generator\,Loss = {} & L1\,Identity\,Loss \\ & + FID\,Identity\,Loss \\ & + Adversarial\,Loss \\ & + L1\,Cycle\,Consistency\,Loss \\ & + FID\,Cycle\,Consistency\,Loss \end{aligned} \tag{1}$$

5.4. Generator Architecture

We adopted the Generator architecture from the Baseline CycleGAN model. We have two generators $G : X \rightarrow Y$ and $F : Y \rightarrow X$. The loss function of the generator is described in equation (1). The weights of the two generators are updated according to the equation (1).

5.5. Discriminator Architecture

For the discriminator network two 70 × 70 PatchGANs were used, which aim to classify whether 70 × 70 overlapping image patches are real or fake. Such a patch-level discriminator architecture has fewer parameters than a full-image discriminator and can work on arbitrarily sized images in a fully convolutional fashion. We have adopted the PatchGAN Discriminator Architecture as described in [2].

Likewise, we also added two global discriminators - classifying if entire image is fake or real. The global and patch

discriminator work independently and the net Generator Adversarial Loss includes components from all the discriminators to make the model robust (or immune) to mode collapse.

5.6. Qualitative Evaluation

For the training of Enhanced model, we use the batch size of 8.

We perform Qualitative Evaluation of the enhanced model by manually checking the quality of the generated images. Here are few of the generated images for both the training combinations as mentioned above. As we can see from Fig. 19, 20, 21 and 22, the quality of translated images from one domain to another is better than from the base CycleGAN model for all the combinations. The new model is able to capture the proportions, size, texture and shape of toppings of well compared to the Base model in the synthetic as well as reconstructed images.

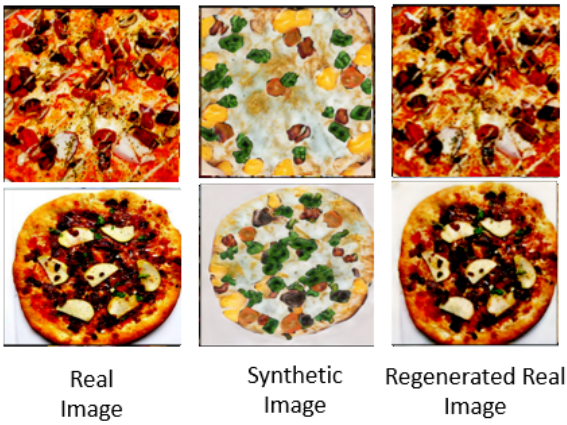

Figure 19. Sample output images of live dataset from Enhanced Cycle GAN. Real → generated Synthetic → reconstructed Real

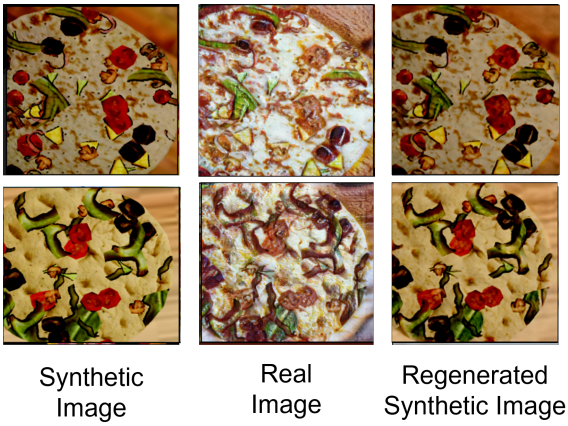

Figure 20. Sample output images of live dataset from Enhanced Cycle. Synthetic → generated Real → reconstructed Synthetic

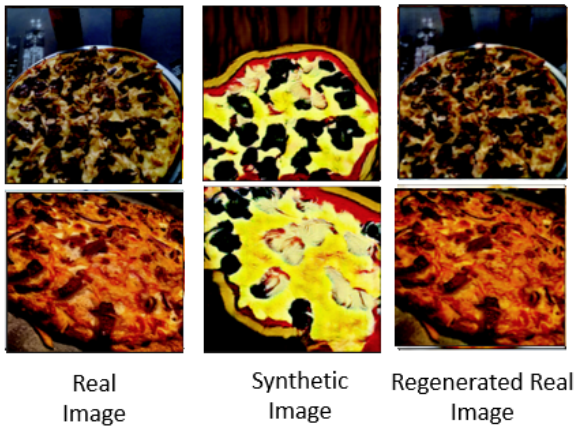

Figure 21. Sample output images of pre-recorded dataset from Enhanced Cycle. Real → generated Synthetic → reconstructed Real

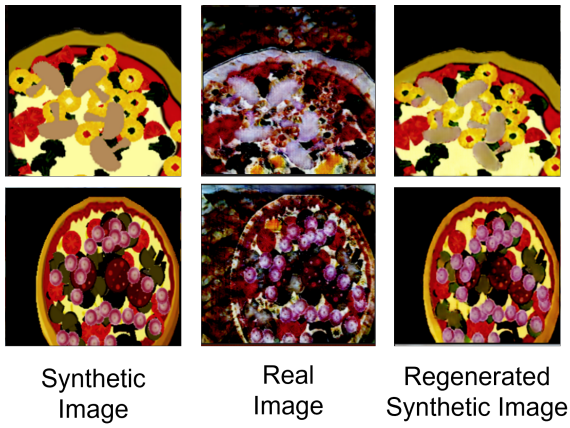

Figure 22. Sample output images of pre-recorded dataset from Enhanced Cycle. Real → generated Real → reconstructed Synthetic

5.7. Quantitative Evaluation

Below graphs show FID scores from the enhanced model evaluation. (Fig. 23, 24, 25 and 26).

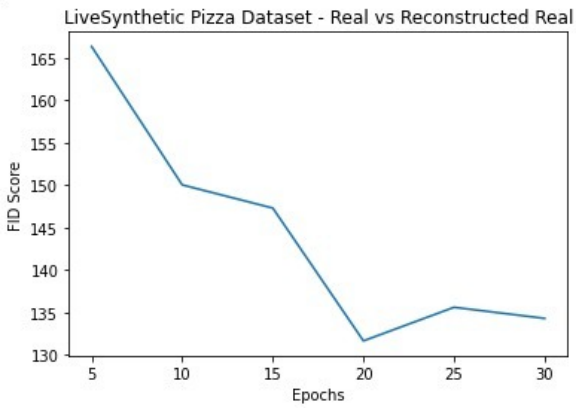

Figure 23. Enhanced CycleGAN - FID Score for Real images versus reconstructed real images for Live Synthetic pizza dataset

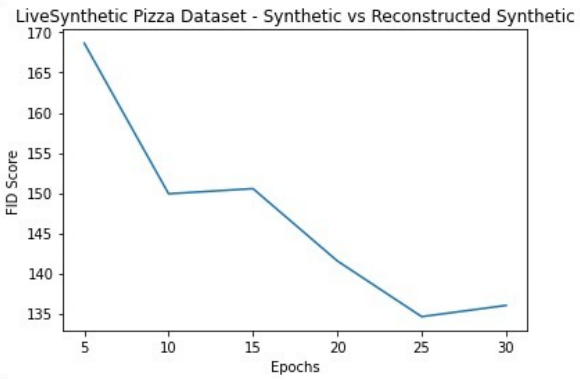

Figure 24. Enhanced CycleGAN - FID Score for synthetic images versus reconstructed synthetic images Live Synthetic pizza dataset

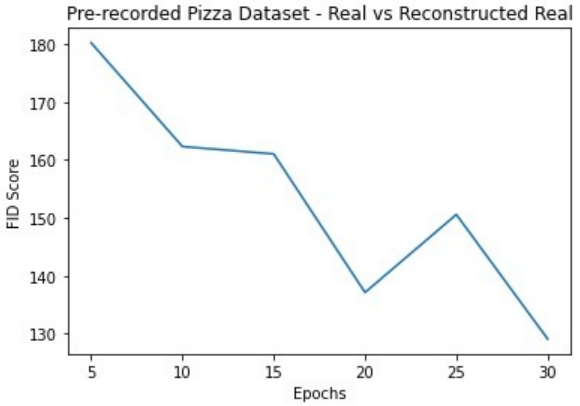

Figure 25. Enhanced CycleGAN - FID Score for real images versus reconstructed real images for pre-recorded pizza dataset

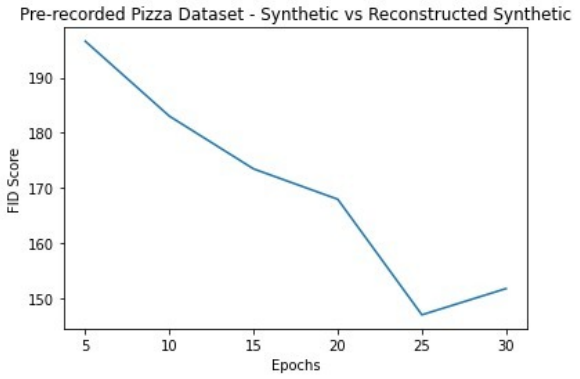

Figure 26. Enhanced CycleGAN - FID Score for synthetic images versus reconstructed synthetic images for pre-recorded pizza dataset

From the above visualizations of FID scores of two datasets, we infer that the range of FID scores for Live dataset are lower, compared to the pre-recorded dataset. Hence, the model is better with live dataset than synthetic dataset.

6. Discussion and Limitations

With our enhancements, we observed that the identity and Cyclic Consistency losses are more robust when FID is included, especially for larger batch sizes. Also, the addition of the Global Discriminator helps make the model more immune to mode collapse resulting from stagnation of learning in discriminator. So, overall the quality of the images have improved. Although the model performed better than the Base CycleGAN model, it has slower training time due to additional calculations that have to be performed for FID scores calculations and also for training the two global discriminators.

7. Conclusion

To conclude, the results of this paper suggest that, although CycleGAN is considered as one of the powerful unpaired image-to-image translation techniques, it fails to perform well when the domain gap between the source and target domains is large. The enhanced model proposed in this paper, tries to overcome some of the limitations of the CycleGAN and achieves better results, both qualitatively and quantitatively but it takes increases the training time and also requires more resources than the baseline model(CycleGAN).

8. Contribution

The rough outline of how we worked in Task 3 and 4 is as follows:

1. Task 3: Training and Evaluation of Cycle GAN
   (a) Pre Recorded to Real Pizza and vice versa: Animesh and Harish
   (b) Real Pizza to Pre Recorded Pizza and vice versa: Karan and Jaini
2. Task 4: Enhancement of Cycle GAN
   (a) Generation of Live Dataset: Jaini, Animesh, Karan and Harish
   (b) Enhancement of FID: Harish and Animesh
   (c) Global and PatchGAN Discriminator: Karan and Jaini
3. Presentation : Animesh, Harish, Karan and Jaini
4. Report: Animesh, Harish, Karan and Jaini

Everyone gave equal contribution and everyone is satisfied with each other's participation and support.

9. Code Repository and Supplementary Results

The code for our Step 3 and Step 4 can be found at: GitHub Repository. We have added another file containing Inception Scores which further supports the better performance of our improved model.

References

[1] A. Borji. Pros and cons of GAN evaluation measures: New developments. *CoRR*, abs/2103.09396, 2021. 1

[2] P. Isola, J. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. *CoRR*, abs/1611.07004, 2016. 1, 2, 6

[3] J. Johnson, A. Alahi, and L. Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. volume abs/1603.08155, 2016. 1

[4] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen. Improved techniques for training gans. June 2016. 1

[5] M. Seitzer. pytorch-fid: FID Score for PyTorch. https://github.com/mseitzer/pytorch-fid, August 2020. Version 0.2.1. 1

[6] Y. Zhao, R. Wu, and H. Dong. Unpaired image-to-image translation using adversarial consistency loss. *CoRR*, abs/2003.04858, 2020. 3