Enhancement of Unpaired Image-to-Image Translation Model

VAMSHI SAGGURTHI
Department of Computer Science
Rutgers University
vamshi.saggurthi@rutgers.edu

VIGNESH RAJESH SINGH
Rutgers University
Department of Computer Science
vrs60@scarletmail.rutgers.edu

Qingyang Yu
Rutgers University
Department of Computer Science
qingyang.yu@rutgers.edu

Chaitanya Vallabhaneni
jt1039
Rutgers University
Department of Computer Science
cv346@scarletmail.rutgers.edu

Abstract

Image-to-image translation (I2I) is designed to transfer the images from the input domain to a output domain while preserving the main features and representations.Generative Adversarial Networks (GANs) has been used widely as a common model for in I2I tasks. GAN's come along with their defects like hard training, mode collapse etc.Many researches are undertaken to improve the reliability and performance of GAN's ,we have tried to experiment with new variations of CycleGAN trying to perform better than the existing models.

1. Introduction

Generative Adversarial Networks (GANs) are one of the most promising unsupervised generative models. Its main idea is to train a generator and discriminator such that the generator could produce fake images to confuse the discriminator after a number of adversarial training. Training GANs is a hard work since the loss function of GAN model is min-max loss function which requires much computing resources.

One main feature of regular GAN model is that training model is non-trivial, which means that the model could achieve extra help from labels or tags for better performance. As a result, conditional GAN's like Pix2Pix [4] arise as a improvement method compared to the traditional GAN's. For condition GAN models, labels are extensions of latent space that could control the category of generated data while regular GAN models would not restrict the mode of generator output. In this case, the Pix2Pix GAN changes the loss function so that the generator output is both plausible in the content of the target domain, and a plausible translation of the given input data sets.

In real world scenario ,we lack paired images/labelled data . As a result, conditional models like pixel2pixel would not work under this setting . CycleGAN [5] has been introduced to deal with such unpaired data sets. The approach frequently succeeds on translation jobs involving color and texture changes, such as many of those mentioned above. The model has little success with geometric changes with often generated images degenerate with minimal changes to input. It also struggles in texture mapping with new objects involved in the input which was not seen during training.

Motivated by the aforementioned challenges, our target is to improve the existing CycleGAN models by experimenting with new loss functions for better geometric translations of the input domains and also preserving textures when unseen inputs are present in the input . We have also tried to understand the internal latent space of each layers to see how the models learns the mappings and are able to produce results better than the baseline cycle gan model.

2. Prior Work

2.1. Vanilla GAN

Generative Adversarial Network (GAN) was introduced by Ian Goodfellow in 2014. It is used for unsupervised learning model, which consists of a generative model and a discriminative model. As is shown in Fig. 1, GAN would train the models through an adversarial framework. We have implemented such a regular GAN model to understand the basic working of GAN.

In Vanilla GAN,the Discriminator wants to drive the

likelihood of D(G(z)) to 0. Hence it wants to maximize (1-D(G(z))) whereas the Generator wants to force the likelihood of D(G(z)) to 1 so that Discriminator makes a mistake in calling out a generated sample as real. Hence Generator wants to minimize (1-D(G(z)).

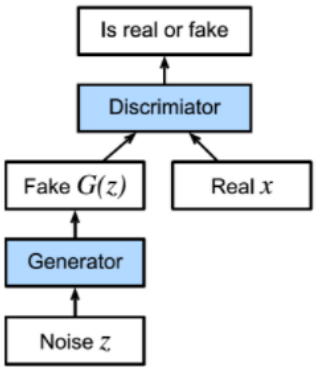

Figure 1: Vanilla GAN architecture

The generator architecture for GANs implementation 080 was adopted from [13] and the discriminator used was a 081 70 x 70 PatchGAN[12].The GAN was trained(individually for real pizza domain and synthetic pizza) over 100 epochs each.

2.1.1 Results

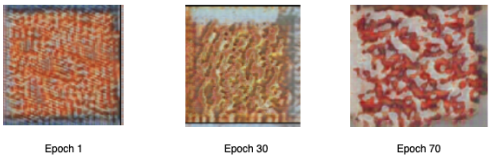

Figure 2: Real pizza Qualitative Results

Metric	Value
FID (Average)	295
IS (Average)	1.38

Table 1:
Real Pizza Quantitative Results

The basic model tries to understand the distribution of the data domain and produce images based on it , we observed the qualitative metrics are low and images are not sharp. These models does a decent job to understand and translate features although the results are not so pleasing to human eye (which is being reflected in the low IS score).

Figure 3: Synthetic pizza Qualitative Results

Metric	Value
FID (Average)	70
IS (Average)	1.35

Table 2: Synthetic pizza Qualitative Results

2.2. Pixel2Pixel

Generative models has obtained dramatically improvement through years. However, traditional GAN models are sensitive to its initial settings such as layer structures, loss functions, and other related factors. As a result, training a GAN model suffers from hyper parameter sensitivity and optimization progress. Thus, people need a new model to deal with these disadvantages.

In 2016, P Isola and his team introduced a conditional GAN model named pixel2pixel. Pixel2Pixel is a model that a generated target image is conditional on a given input image. Conditional GAN (cGAN) plays as an extension of traditional GAN models, which is able to control the classification or modes of the generated images.

As is shown in Fig. 2, the generator model is provided with a given image as input and generates a translated version of the image. The discriminator model is given an input image and a real or generated paired image and must determine whether the paired image is real or fake. Finally, the generator model is trained to both fool the discriminator model and to minimize the loss between the generated image and the expected target image

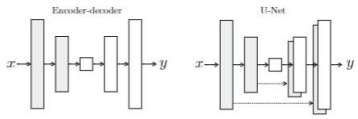

Figure 4: Two architectures of the generator. The 'U-Net' is an encoder-decoder with skip connections between mirrored layers in the decoder and encoder stacks

2.2.1 Results

The model generated better results in aerial to street (avg Fid 241 & IS of 1.9) translation then street to aerial(avg Fid 375 & IS of 1.8)

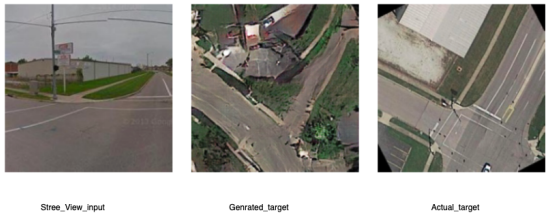

Figure 5: Street to Aerial

Figure 6: Aerial to Street

3. CycleGAN

CycleGAN is unpaired image-to-image translation task from A to B and represented by two generative networks G and F:

$$\hat{y} = G(x) \in B, \text{ for } x \in A$$
$$\hat{x} = F(y) \in A, \text{ for } y \in B$$

In a general GAN, we have a Generator generating fake images & a Discriminator detecting real & fake images given one domain. Now, as we have images coming from 2 domains (be it Real → Synthetic or Synthetic →Real), we have a pair of Generator & Discriminator for each domain so that the translation can happen both ways

3.1. Working steps

As the name suggests, CycleGAN consists of a cyclic structure formed between these multiple generators & discriminators.

Assume A=Synthetic Pizza and B=Real Pizza. The cyclic flow now looks somewhat like this.

- Domain A (Synthetic) random samples are given to generator A B (Synthetic Real). This generator takes fake photos and turns them into real pizza. Rather than using random noise (as in conventional GAN), we use images from one domain to train the generator to translate them to another.
- The generated picture for Domain B (Real Pizza) from generator A B is given to generator B A (Real Synthetic), which reproduces the input image from Domain A.

The same flow then goes vice-versa for Real → Synthetic.

- Image_A → Generator_A_B → Image_B → Generator_B_A →
- Image_B → Generator_B_A → Image_A → Generator_A_B → Image_B

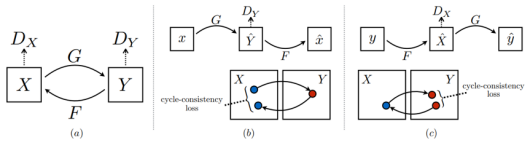

Figure 7: Cycle GAN

3.2. Dataset

We have used two different data sets for training the CycleGAN models Pre-Recorded data-set & Live Pizza data set Pre-Recorded image set consisted of about 8000+ training images for both synthetic and live domain There are 13321 train images and 1538 test images in the datasets. Live pizza dataset of synthetic images , are created by manipulating various factors related to the distribution of ingredients in a pizza image using code. We used 12 different toppings, multiple templates, multiple different backdrops, and two different pizza bases to create live pizza images. We first resize the images to 256*256 and also apply transformations like random crop,Random Flip,adding Jitters in the images while loading

3.3. Architecture

3.3.1 Generator

An encoder, a transformer, and a decoder make up every CycleGAN generator. The encoder receives the input image directly, shrinking the representation size while increasing the number of channels. Three convolution layers make up the encoder. The activation is then sent to the transformer, which is made up of nine residual blocks. The decoder then expands it again, using two transpose convolutions to increase the representation size and one output layer to produce the final RGB image.

3.3.2 Discriminator

PatchGANs, fully convolutional neural networks that look at a "patch" of an input image and output the probability of the patch being "actual," are the discriminators. This is both more computationally efficient and more effective than looking at the complete input image since it allows the discriminator to focus on surface-level properties like texture, which are frequently modified in image translation tasks.

3.4. Loss functions

- Discriminator must approve all the original images of the corresponding categories.
- Discriminator must reject all the images which are generated by corresponding Generators to fool them.
- Generators must make the discriminators approve all the generated images, so as to fool them.
- The generated image must retain the property of original image, so if we generate a fake image using a generator say $Generator_A \rightarrow Generator_B$ then we must be able to get back to original image using the another generator $Generator_B \rightarrow Generator_A$ - it must satisfy cyclic-consistency

  Adversarial loss: Loss due to misclassification between real & fake images.

  Identity loss: It is a loss term calculated at the pixel level. What would you wish your generator_A_B(i.e Synthetic →Real) to produce if a Synthetic image is fed in We would wish the generator should not tweak anything as the input image is already in the desired domain. It's assumed this term helps in preserving colors in the output images avoiding unnecessary changes.

  Cyclic loss: As we observed the above cyclic structure that exists in CycleGAN, where we pass an image from one of the domains to both the generators sequentially producing the same image as output. This term is the MSE loss between the input & output image as ideally, input & output should remain the same.

  MSE is used for the Adversarial terms & MAE for the rest of the terms.

Final Objective Loss is

$$\mathcal{L}_{total} = \mathcal{L}_{GAN}(F, D_x, X, Y) + \mathcal{L}_{GAN}(G, Dy, X, Y) + \lambda_{cyclic}\mathcal{L}_{cyclic}(G, F) + \lambda_{id}\mathcal{L}_{id}(G, F) \quad (1)$$

3.5. Training

For an image-to-image translation problem, the training procedure was quite standard. To improve training stability and efficiency, the Adam optimizer, a standard variation of gradient descent, was utilized. For the first half of the training, the learning rate was set at 0.0002, and then linearly lowered to zero throughout the remaining rounds. Because the batch size was set to 1, we refer to instance normalization instead of batch normalization.

3.6. Evaluation

Upon preliminary observation we can see that the quality of the generated image is better than vanilla GAN which can also be verified by the high inception score values. In generating synthetic images from real, the model does a good job of detecting the shape of pizza, and remove background objects from the real images. The model tries to replicate the toppings of input image into generated image. We find that as we train the model beyond **78** epochs it starts to overfit and generated images seem to loose structural similarity and looses perceived realness. In the task of generating real images from synthetic images, we see that the model is not able to detect structures and the results are not good.

Figure 8: Cycle-GAN Real to Synthetic Epoch 65

Figure 9: Cycle-GAN Real to Synthetic epoch 93

Figure 10: Cycle-GAN Synthetic to Real

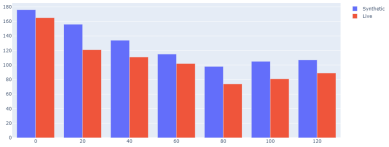

Figure 11: CycleGAN FID Scores

Metric	Value
IS (Lowest)	2.36
IS (Average)	3.78
IS (Highest)	4.98 height

Table 3: CycleGAN IS

3.7. Drawbacks & observations

- The main drawback of cycleGan is mode collapse, where the generator couldn't produce images close to that of required target domain . This was observed in the higher epochs(122) where objects like bottles are present in the input images they were clearly visible in the generated image which was being mapped to the target domain.

- We also observed that unwanted artifacts which was not seen during the training are being translated to target images

- We also noticed background color being reversed as for certain new backgrounds and color was being translated to the toppings of the pizza .

- One of the major drawback observed was the geometric shape of the pizza is being forced to be circular in most cases even though it is different in the input, this was mainly because of the cycle consistency loss which was weighted heavily in the optmization

4. Novel Approaches

Motivation : We mainly observed in our qualitative evaluation the geometric and style inconsistency being the major factors in the domain gap of the generated and target images ,so to overcome these we tried to decrease the cycle consitency loss factor and add additional loss functions which would try to incorporate the domain gap

4.1. Experiment 1 : MSE-SSIM Loss funciton

4.1.1 Backgorund

The Structural Similarity Index (SSIM)[6] is a perceptual metric that quantifies image quality degradation caused by processing such as data compression or by losses in data transmission. We have tried to compute this between real image and the generated image attempting to force the generators to produce images which are more sharp in perception .

4.1.2 Loss function

The loss function of generator has been changed as below

$$\mathcal{L}_{total} = \mathcal{L}_{GAN}(F, D_x, X, Y) + \mathcal{L}_{GAN}(G, Dy, X, Y) + \lambda_{cyclic}\mathcal{L}_{cyclic}(G, F) + \lambda_{id}\mathcal{L}_{id}(G, F) - SSIM\_loss + 1 \quad (2)$$

4.1.3 Training

The training is similar as of the cycle gan , to speed up the process and we have utilised the pre trained generator of the previous cycle gan at epoch 40

4.1.4 Results

We have majorly observe within a very few epochs the model is producing better images

We observed the results are more robust than that the cycle gan , unwanted artifacts like cardboard of the pizza in one input are not being translated,preserving the learned background

The geometric shape of the input pizza say if hexagon is being translated without being forced to be circular always and also when slices of the pizza is given in the input , the translated synthetic image has slight preserving of such cuts.

The overall sharpness of the image has been increased which is reflected in the Higher inception scores(5.6).
Comparison with baseline

Steps	CycleGAN		MSE-SSIM	
	FID	IS	FID	IS
47	134	2.86	146	3.2
56	115	3.9	109	4.7
77	98	4.98	97	5.6

Table 4: Comparison with Cycle-GAN

Figure 12: SSIM Real to Synthetic Epoch 7

Figure 13: SSIM Real to Synthetic Epoch 16

Figure 14: SSIM Real to Synthetic Epoch 37

Figure 15: SSIM Synthetic to Real Epoch 7

Figure 16: SSIM Synthetic to Real Epoch 16

Figure 17: SSIM Synthetic to Real Epoch 37

4.2. Experiment 2 : Weighted Multi Scale Discriminative Feature Loss function

4.2.1 Background

We have tried to incorporate the novelties of other gan's trained to help cycle gan imporve on the style translations . SinGAN [9], an unconditional generative model that can be learned from a single natural image. Our model is trained to capture the internal distribution of patches within the image, and is then able to generate high quality, diverse samples that carry the same visual content as the image. SinGAN contains a pyramid of fully convolutional GANs, each re-

sponsible for learning the patch distribution at a different scale of the image. This allows generating new samples of arbitrary size and aspect ratio, that have significant variability, yet maintain both the global structure and the fine textures of the training image. We have used a loss funciton computed based on a pretrained model and have given

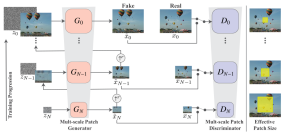

Figure 18: SinGAN Artitechture

4.2.2 Loss function

where lambda-style is set to 0.01 to not over weigh the mdf loss over cycle consistency loss

$$\mathcal{L}_{total} = \mathcal{L}_{GAN}(F, D_x, X, Y) + \mathcal{L}_{GAN}(G, Dy, X, Y) + \lambda_{cyclic}\mathcal{L}_{cyclic}(G, F) + \lambda_{id}\mathcal{L}_{id}(G, F) - \lambda_{style} * MDF\_loss \quad (3)$$

4.2.3 Training

Training is similar to that of cycle gan , only advantage is using a pre trained generator of epoch 40 from original cycle gan The training time for mdf is about 3hrs/epoch

4.2.4 Results comparison

Due to the high training time we are able to run the experiment for few epochs only

We observed high quality images from the initial epochs, The outputs generated have been capturing geometric shapes of input pizza which is objective We have also observed the results capturing the required input space , for sample even though there are humans in input the generated synthetic image has no such replications

Steps	CycleGAN		MDF	
	FID	IS	FID	IS
42	133	2.86	132	2.95
45	128	3.25	133	3.36

Table 5: Comparison with Cycle-GAN

Figure 19: MDF Real to Synthetic Epoch 2

Figure 20: MDF Real to Synthetic Epoch 5

5. Latent Space Exploaration

5.1. Dimension Reduction

We have visualized the relations between original image data and latent space vectors of model to interpret and validate the results of our experiments. To visualize the model we need to reduce the high dimensional input image(3x256x256) and output latent vector of the generator res-net blocks(256x64x64) in to 2-dimensions.

Our dataset comprised of images of that are combination of various toppings, bases and backgrounds that have non-linear relationship we found that T-SNE is useful in visualising such high dimensional data.We used T-SNE reduction individually on a batch of 100 input images and on latent vectors of these images and plot the result to find similarities between their distributions in 2-dimensions.

After running multiple tests, we found that the parameters that produced best output were perplexity = 300,learning_rate = 250, number of iterations = 300.

As is shown in Fig. 7, we can see that the spread of input images and latent space vectors is very distinct for basic Cycle GAN. This is expected as the activations for Cycle GAN are not sharp and the generated output images have less similarity with the original image.

The overlap between latent vectors and input image is

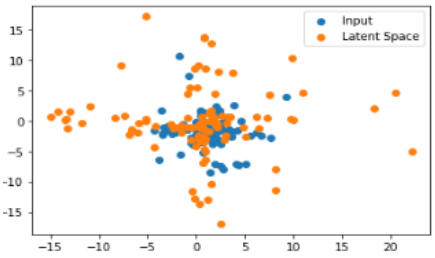

Figure 21: T-SNE reduction Cycle-GAN

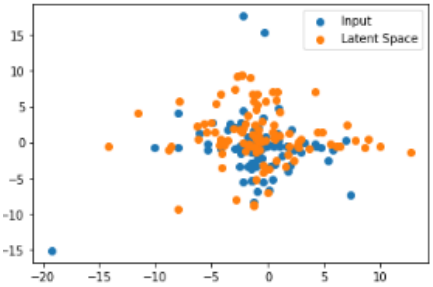

Figure 22: T-SNE reduction MSE-SSIM

highest among the MSE-SSIM models(Fig. 8). We can verify this by inspecting the structural similarities between generated output images and the original image.The activations at each latent layer were also sharper.

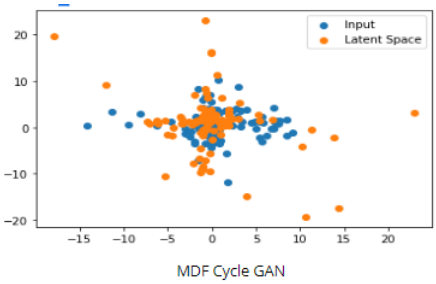

Figure 23: T-SNE reduction MDF

The MDF model(Fig. 9) had a good overlap between the points with few outliers, given that it was trained for only 30 epochs this result was expected.

5.2. Activations

To get a deeper insight into the intrinsic mechanism of GANs we explored the activations at each layer. We use these layer by layer activations to understand how each model detected geometric components and pizza topping from input image.

5.2.1 Cycle GAN :-

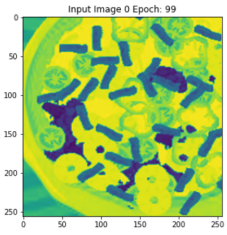

Figure 24: Cycle GAN Input (Epoch 99)

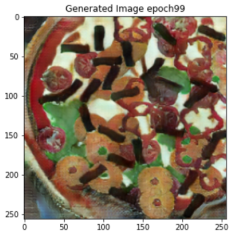

Figure 25: Cycle GAN Generated (Epoch 99)

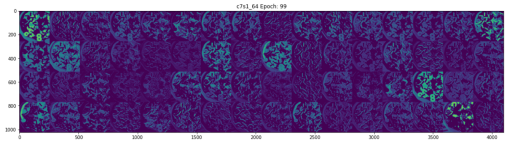

Figure 26: Cycle GAN Layer c7s1_64 (Epoch 99)

- On qualitative inspection of the generated image and latent images, it is evident that the model is recognizing features and maintaining geometric similarities between both domains.
- Looking at the c7s1_64 layer, there are multiple activations of feature and shapes but the activations are not sharp. Thus the generated image is of round shape and does not preserve the hexagonal shape of input image.

5.2.2 SSIM Based GAN:-

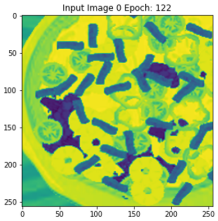

Figure 27: SSIM Input (Epoch 122)

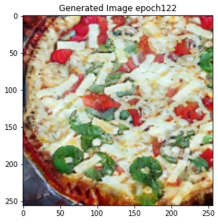

Figure 28: SSIM Generated (Epoch 122)

- The activations of toppings and geometry are sharpest and most evident in this model. This can explain the relatively better generated images of this model even at lower(12) epochs.

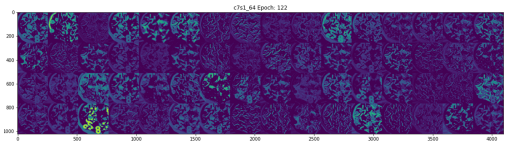

Figure 29: SSIM Layer c7s1_64 (Epoch 122)

- The generated image preserves the hexagonal shape of input, the positions of toppings. The image is less realistic than SSIM, this can be attributed to the lower trained epoch of this model.

5.2.3 MDF Based GAN:-

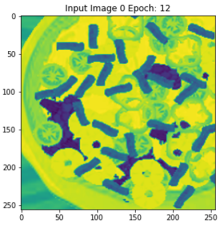

Figure 30: MDF Input (Epoch 12)

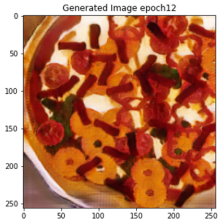

Figure 31: MDF Generated (Epoch 12)

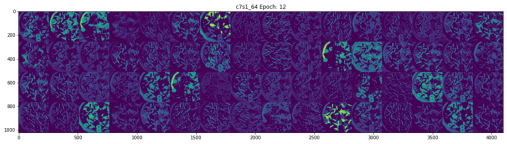

Figure 32: MDF Layer c7s1_64 (Epoch 12)

- The activations of toppings and geometry are sharper at all the latent layers compared to CycleGAN.
- The generated image preserves the hexagonal shape of input, the positions of toppings and can be visually perceived to be of excellent quality.

6. Conclusion

We have extensively learned the workings and nuances of building and applying GANs for solving image to image translations. The basic vanilla GAN introduced us to understand the effects of different parameters, training methodologies and different implementation libraries(Pytorch, Tensorflow). The Pix2Pix (a GAN implementation for paired Image to Image Translation), helped us interpret the conditional inputs rather than a random noise for style transfer from the source to the target domain. The results were quite pleasing which shows supervised learning of GANs has the best results, butin a real world setting the non-availability of paired pictures of source and target domain is a key drawback of paired image to image translation approach.

The CycleGAN(unpaired image to image translation GAN) helped us to overcome the limitations of Pix2Pix . But, this approach is also prone to some limitation such as unwanted style transfer where it tries to fit to input domain instead of target domain, mode collapse [3], and failure to transfer geometric translations.

Finally, in attempt to improve on CycleGAN, we used two novel approaches,one based on Structural Similarity Index and other approach based on Multiscale Discriminative Feature Loss [7]. Both these approaches focus on capturing the style lost like geometric transformations, we also learnt to speed up the training processes for these approaches by using pre-trained generators of Cycle GAN. We investigated the activations at each layer to gain a better understanding of GANs' intrinsic process. To evaluate and validate the outcomes of our experiments, we performed dimensional reduction to also visualize the relationships between original image data and model latent space vectors.

7. Future work

- We have not been able to train and find the right parameters for MDF based loss , we expect it to outperform the baseline much better
- Latent space exploration is done quite effectivetly ,with further understanding building diffusion based models to train the generator actively helps better converge
- We could add attention based to cycle gan combined with ssim loss to learn about the required parameters more.

8. Code and Contributions

Code

Net ID	Report	Code+training	Research
(vrs60)	40%	30%	35%
(vs734)	30%	55%	50%
(qy116)	20%	5%	15%
(cv346)	10%	10%	0%

Table 6: Contributions

References

[1] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial networks, 2014.

[2] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. 2017.

[3] T. T. Hoang Thanh-Tung. Perceptual losses for deep image restoration. 9

[4] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. "image-to-image translation with conditional adversarial networks" in *Proceedings of the IEEE conference on computer vision and pattern recognition*. pages 1125–1134, 2011. 1

[5] P. I. J.-Y. Zhu, T. Park and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. *in Proceedings of the IEEE international conference on computer vision*, page 2223– 2232, 2017. 1

[6] A. Mikhailiuk. Perceptual losses for deep image restoration. 5

[7] A. Mustafa, A. Mikhailiuk, D. A. Iliescu, V. Babbar, and R. K. Mantiuk. Training a task-specific image reconstruction loss. pages 2319–2328, 2022. 9

[8] D. P. Papadopoulos, Y. Tamaazousti, F. Ofli, I. Weber, and A. Torralba. How to make a pizza: Learning a compositional layer-based gan model. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[9] T. Rott Shaham, T. Dekel, and T. Michaeli. Singan: Learning a generative model from a single natural image. 2019. 6

[10] H. Tang, H. Liu, D. Xu, P. H. S. Torr, and N. Sebe. Attention-gan: Unpaired image-to-image translation using attention-guided generative adversarial networks, 2019.

[11] N. N. Vo and J. Hays. Localizing and orienting street views using overhead imagery. *European conference on computer vision*, page 494–509, 2016.