Image to Image Translation

Arvind A S*
Georgia Tech
arvind_s@gatech.edu

Aastha Agrawal*
Georgia Tech
aagrawal319@gatech.edu

Sreehari S*
Georgia Tech
ssreejith3@gatech.edu

Abstract

We study the problem of image to image translation by exploring state of the art techniques that propose general models to solving image to image translation. We specifically explore two main approaches - (1) supervised img2img translation proposed by Isola et al. *[5] and (2) unpaired img2img translation proposed by Zhu* et al. *[17] - and evaluate their training and performance qualitatively on the CMP Facades dataset [15]. Our experiments attempt to capture some of the nuances involved in training GANs by showing the effect that various GAN objective functions and generator network architectures can have when training is done under a limited budget. We also implemented a Variational Autoencoder conditional GAN (VAE-cGAN) for the image to image translation task. Link to img2img experiments:* https://github.com/mon95/img2img-experiments *Link to VAE model additions:* https://github.gatech.edu/asrinivasan87/Pix2Pix_CVAE

1. Introduction

Image to image translation is a popular computer vision and image processing problem. It aims at learning a mapping between the characteristics of a set of input images and output images provided in the training data. Similar to machine translation of languages, image to image translation generates another representation of the same data. It is useful in many applications like photo enhancement, style transfer and object transfiguration. An example of this task would be season transfer in a set of landscape images.

1.1. Objective

Our main objective is to evaluate state-of-the-art image to image translation techniques. In this project, we aim to experiment with two image to image translation models - (1) the Pix2Pix model proposed by Isola *et al.* [5] where the models are trained on pairs of corresponding input (class A images) and output images (class B images) and (2) the CycleGAN model which is an unsupervised method that allows for unpaired image to image translation (figure 1 shows the difference between paired and unpaired images). We analyze the results from these models for the image to image translation task and explore how a few specific aspects of training GANs such as objective functions and generator architectures affect the training and performance on test data, when training needs to be performed under a limited budget. Further, we also experiment with using a Variational Autoencoder in place of a generator in the Pix2Pix model to explore and potentially investigate if this helps improve how the model generates images.

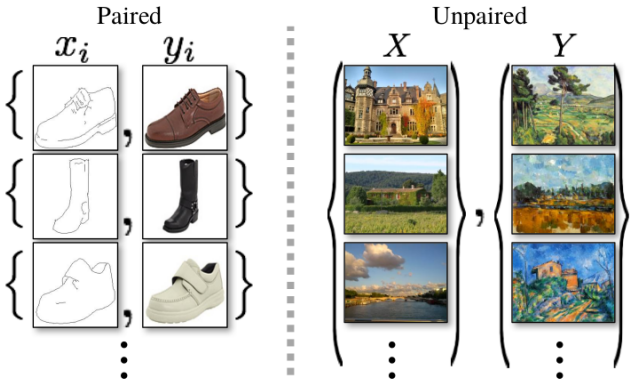

Figure 1. The left hand side depcits the paired images which pix2pix requires and the right hand side depicts the unpaired images CycleGAN can work on

1.2. Importance

Image-Image translation is increasingly becoming important due to an increased interest in applications like style transfer in images, domain adaptation use-cases such as understanding how large datasets can be leveraged to create and augment smaller datasets for better training, improving photo quality and so on. All such applications in the domain of Computer Vision and Graphics involve predicting pixels from pixels and general approaches to solve this problem would serve as baselines to extend upon for most of these tasks. In our project, we explore how changing different aspects of the model (such as, generator architectures and loss

objectives) can affect the result in the image-to-image translation task, in both paired and unpaired settings. Apart from studying the settings that help us generate good results, a general understanding of both the supervised and unsupervised setting and how the results differ between them can be important in making decisions in downstream applications of such models (for eg., to understand if there is a need to gather paired image data).

1.3. Background

Previous work in Image to image translation Most previous work in image to image translation has involved building models for specific tasks. For eg., Zhang et al [11] utilized a CNN based model to *hallucinate* a plausible color version of a gray scale image and successfully managed to fool 30% of human participants in a colorization Turing test. The proposed algorithm was specifically designed for gray-scale-to-colour image to image translation task. Similarly [16] proposed a very task-specific CNN based HED algorithm for producing a edge mapping for a given input image.

GANs and VAEs Generative models were proposed by Goodfellow et al [2] wherein a generator function produces new content based on an input and a second discriminator function learns to discriminate real and fake images, thereby allowing generation of robust images. Variational Autoencoder [6] introduced the deep learning technique of learning distributions over latent space by adding a regularization term in the loss function of neural networks.

1.4. Pix2Pix

The Pix2Pix GAN is a general approach for image-to-image translation done in a supervised setting where paired images are available for training. The framework expects an image from a source distribution A and it's paired expected output image from a target distribution B as training examples. Pix2Pix relies on a conditional generative adversarial network (cGAN) for the image to image translation. This points to an important feature which of the network which is different from traditional GANs: the loss function used is one which minimizes both the traditional GAN objective as well as an L1 loss between the generated output and the expected output in the target distribution. So we have:

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

$$L_{L1}(G) = E_{x,y,z}[|y - G(x, z)|_1]$$

The final objective is:

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

A salient feature of the pix2pix model is that it uses a UNet architecture [12] for the generator and a patch GAN archtitecture [8] for the discriminator. Further details on the same are described in relevant sections that follow. Another important modification in the model is the use of dropout in several layers to simulate stochasticity that is needed in the model.

1.5. Reason behind choosing Pix2Pix

Previous work in image to image translation such as [11] and [16] focused on designing a model very tightly bound to a specific task, even though the setting was always the same - predict pixels from pixels. Isola et al. [5] was the first work in the direction to develop a common model for a variety of image to image translation tasks (as far as we have researched). It made use of Generative Adversarial Networks (GANs) to learn a loss function which would adapt to the specific task as well as to the data. More specifically, it uses Conditional GANs (cGANs) to generate an output image conditioned on an input image. Experimenting and evaluating Pix2Pix [5] gave us opportunity to explore a variety to img2img tasks and also develop deeper understanding of GANs.

1.6. CycleGAN

CycleGAN [17] addresses the problem of image to image translation in an unsupervised setting such that it doesn't require specifically paired images capturing single attribute variation while keeping the rest same. It rather works on two collections of images capturing the variation in the features (eg: style). The model comprises of two GANs where one GAN inputs image from one of the collections and produces output which are likely to belong to the second collection. To achieve cyclical consistency and achieve translation based on the input, the output of this first GAN is passed to the second GAN to generate output which would likely resemble the original input. CycleGAN works to minimize this objective function which is captured by the following equation:

$L(G, F, D_X, D_Y)$
$= L_{GAN}(G, D_Y, X, Y) + L_{GAN}(F, D_X, Y, X)$
$+\lambda L_{Cyclical}(G, F)$

Here, X and Y are the two domains of input data, G and F capture the mappings such that $G : X- > Y$ and $F : Y- > X$. The first two terms in the objective function captures the adversial loss of the two GAN models and the third term represents cyclical consistency loss. λ controls the relative importance of the two types of losses.

1.7. Why we chose CycleGAN

Training deep learning models for image to image translation task requires large amount of paired images. This might be difficult or even impossible in many scenarios.

Motivated by this challenge, we also conduct experiments on CycleGAN [17] which allows training for image to image translation system by using two unrelated collections of images.

1.8. Description of Dataset

We used CMP Facade Database [15] for all our experiments. It contains 606 facade images which are manually annotated to explain the objects of classes of interest (not the entire image). There are 12 classes of objects which are highlighted - facade, molding, cornice, pillar, window, door, sill, blind, balcony, shop, deco and background. Train, val and test set comprised of 400, 100 and 106 images respectively. We picked this data set as it provided us with annotated images while also suiting our computational constraints in running multiple experiments. Figure 2 shows a paired example of the facades data set that we are using in our project.

2. Approach

We explore two state of the art techniques that serve as general methods to approach image to image translation - the pix2pix model [5] which operates in a paired, supervised setting and the cycleGAN model [17] which operates in an unpaired, unsupervised setting. We run 3 different experiments each on both the pix2pix and cycleGAN models to understand the following: (1) the impact of different GAN objectives in training the model (2) performance of 4 different generator architectures which use skip connections and (3) how varying the size of a buffer which is used to store previous 'n' generated images to update the discriminator impacts GAN performance. The rationale behind these specific experiments is to understand how these factors which are specific to training GANs impact the training, given that training needs to be done with a limited computational budget. While our resources limit our ability to make generalized claims or even verify or disprove any such claims made by large scale studies such as [9], we are still able to see how these factors do make a noticeable difference with respect to our training setting and dataset.

In addition to this, we experiment with an idea to replace the generator of the pix2pix model with a Variational Autoencoder (VAE) and use this to build a VAE-cGAN model. To our best knowledge, such an architecture has not been proposed before in the context of image to image translation problems. Previous work has introduced the following: (1) VAEs that are conditioned on labels (also known as CVAEs) [14] but not one that is conditioned on input images, and (2) VAE-GANs were introduced by Larsen *et al*. [7] wherein the key idea is to use a VAE as a generator. Our proposed approach is new and different from this in that our discriminator is still conditioned on the input image (as done in the original pix2pix model) and

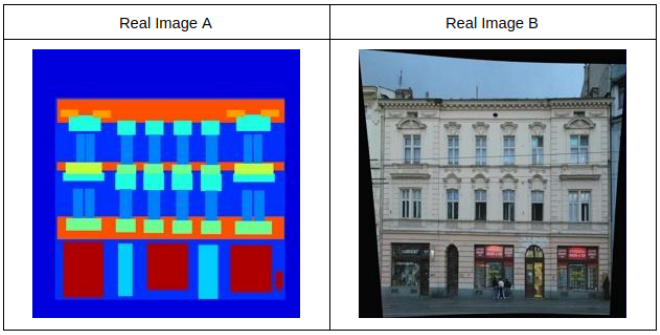

Figure 2. The translation task is to convert images of type A to images of type B

therefore is a function $D(x, y)$ where x is the input image from a source distribution and y is either the corresponding generated fake image or the actual expected image in the target distribution.

Rationale behind implementing a VAE-cGAN for image to image translation

The images produced by a VAE are usually seen to be more blurred than those produced by the generator of a GAN. However, where the VAE has an advantage over GANs is the fact that they explicitly learn to model the features of the images in a latent space. In the context of image-to-image translation this could be helpful as a VAE could potentially be more adept at generalizing to unseen pictures as they focus on the feature representation. Additionally, this may possibly make it easier to map the correspondence between features between the two classes of images (input images and output images). It may be useful to note that the UNet architecture used in the pix2pix model attempted to do this exact same thing by adding skip connections between layers i and $n - i$ to ensure that there's a way for necessary features to get transferred over. Thus, we believe that a VAE-cGAN which serves to achieve a similar mapping over is worthy of investigation.

Given our computing resource constraints, we expected to have issues in being able to run multiple runs for each experiment to test the performance of each loss objective function and generator net architecture. To overcome this, we decided to use the CMP Facades dataset, which has significantly fewer images and run our experiments for only upto 30 epochs unless it was necessary to continue the training process. We monitor the training progress by checking the intermediate generated samples which are made available via a visdom dashboard.

For the VAE-cGAN, given that it has significantly more parameters to train, we did not expect it to train as well as the GANs in the same number of epochs. Also, training any new GAN architecture is known to be an art with many practitioners proposing extensive tips and tricks to get the

training to succeed and avoid collapsing. While we made a bold move to try and implement the VAE-cGAN in the conditional GAN setting by following some of these best practices, we were unable to train the model well enough to generate reasonable results. Yet, we believe there were many important takeaways from our attempt at this and we have explained these in detail in the following section.

For our work, we build off of the pix2pix pytorch code repository. The repository is structured to allow for adding in new model architectures and we use this to plug our VAE based generator model. We use the base training code provided along with the option to train the model using various different options and architectures to run our various experiments, including the VAE training. For testing and visualization, we use the provided test and visualization features.

3. Experiments and Results

In image-to-image translation tasks, it is very difficult to get quantitative metric to define performance. Metrics such as per-pixel mean-squared error does not allow for the structure of the image to be analysed and therefore could be misleading (for example a fake image composed of similar colors to the real image would be seen as superior to one that is of a different color but is structurally sound). One way the models are evaluated is to choose a task and dataset where quantitative analysis is possible. For example we could try running the model on the cityscapes dataset, and checking the performance of previously trained segmentation networks on the generated images in the target domain.

In our case we could not use the cityscape dataset due to computational limitations, we instead chose the facades dataset. This is because, [5] show that even though the dataset is small (size 400), good results can be obtained at low computational cost.

In translation tasks where quantitative analysis is not possible (as it is in our case), qualitative evaluation techniques are instead employed to look at performance. One way of doing this is to see if human beings (usually through amazon turk) can identify if the images produced are fake or for them to rate how realistic they are on a prescribed scale. In this project, since we did not have access to Amazon Turk, we went through the images produced from the testing data set and qualitatively analysed them to decide which models or architectures performed well. We have added some of the images produced from the testing data set to buttress our arguments.

We have also shown images produced from different epochs during the training process so as to give the reader a sense of how the models improve as the number of epochs in the training increases.

3.1. Experiment using different GAN objective functions

We experiment with using 3 different GAN objective functions in training the pix2pix model to understand how different objective functions affect the training. We use 3 GAN modes in our experiments: (1) **LSGAN**: Least Squares GAN introduced by Mao *et al*. [10] which adopts a Least Squares loss function for the discriminator, (2) **WGAN-GP**, i.e., Wasserstein GAN with Gradient Penalty to stablize GAN's training introduced by Guljrani *et al*. [3] which is an improved version of the Wasserstein GAN [1] and (3) a **vanilla GAN**, which uses the GAN loss formulation from the original GAN paper [2].

We note that there have been recent studies [9] that have shown that with enough computational budget and hyperparameter tuning, GANs with different objective functions can still end up with similar performance.

Here, we investigate the qualitative performance of GANs (3) in intermediate epochs of training, by checking the intermediate outputs during training using each of the modes.

Pix2Pix We see that with all other training hyperparameters held constant, the LSGAN gives the best results on the facades dataset, as can be seen in 3. We note that the WGAN-GP objective ensures a significant improvement in the quality of fake image generated with increased training, but it does not quite match the quality of the corresponding LSGAN results.

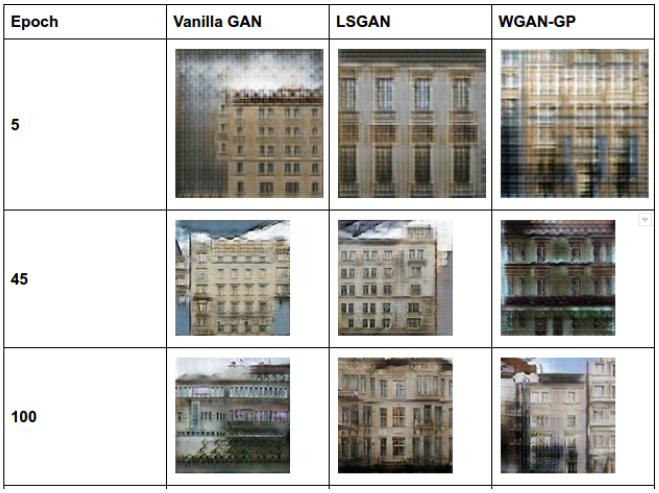

Figure 3. Pix2Pix: Intermediate results at end of 'n' epochs during training using different GAN modes.

CycleGAN As can be seen in Figure 4, both LSGAN and Vanilla GAN perform well with CycleGAN. But this is restricted to rectangular structures, as can be seen in Fig 5. However, we observe that the quality of images generated by WGAN-GP is much lower, similar to what we observed

with Pix2Pix. This is most likely due the gradient penalization done during training in WGAN-GP.

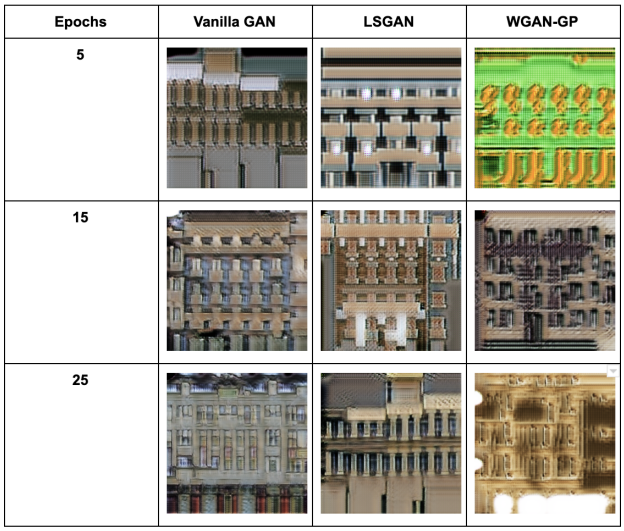

Figure 4. CycleGAN: Intermediate results at end of 'n' epochs during training using different GAN modes

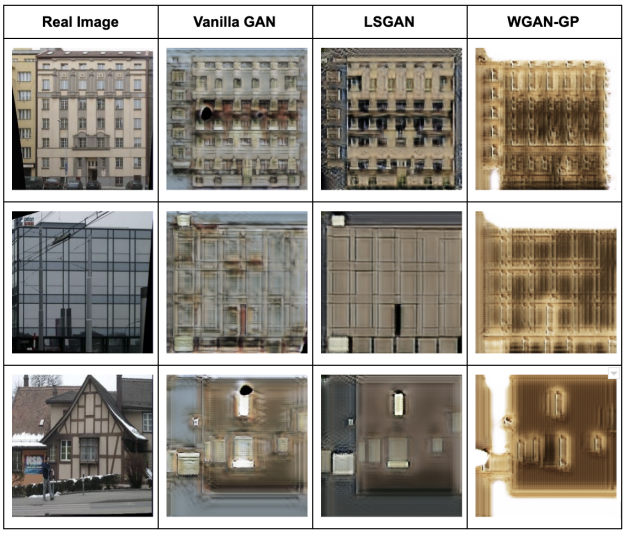

Figure 5. CycleGAN: Sample test outputs using different objective functions

3.2. Experiment using different Generator networks

We experiment with using four different generator networks - (1) Resnet 6 blocks architecture (2) Resnet 9 blocks architecture (3) UNet 128 and (4) Unet 256.

The Resnet architectures are designed as follows: We have 2 downsampling layers which include a Conv2d layer followed by a normalization and ReLU. This is followed by 6 (or 9) Resnet blocks where each Resnet block is a Conv2d with dropout and includes a residual connection as described by He *et al*. [4]. This is then followed by 2 upsampling layers which includes a ConvTranspose2d followed by norm and ReLU. As described in the pix2pix paper, we include a tanh at the end to complete the resnet generator.

The UNet generator architectures are designed as follows: The architecture comprises UNet skip connection blocks which allows for some feature information to bypass the bottleneck layer. Each UNet skip connection block (except the innermost block) has a downsampling (Conv2d, norm, ReLU) layer, an inner UNet block and an upsampling layer (ConvTranspose2d, norm, ReLU). The innermost block is the same without an inner UNet block. The outermost block has an additional tanh following the upsampling operation. Dropout is also added to the intermediate layers. The construction is done in a recursive fashion to allow the above architecture to be built.

The architectures experimented with here all include skip connections. The intuition behind this decision as explained in the original pix2pix paper is as follows: the task of conditional image to image translation involves generating an image which has roughly the same high level underlying structure, with the major difference in the outputs' surface appearance. Thus, having the skip connections in the architecture allows for this information that is present in both the input and output images to be shuttled across easily.

Pix2Pix We find that, on the facades dataset, the trained UNet architecture performs better than the Resnet architecture. The results from using the UNet architecture were found to be more consistent in terms of their agreement with the expected output and with lesser artifacts being generated in comparison to the Resnet based generator models. Sample outputs are shown in Figure 6.

CycleGAN When working with CycleGAN model, we observe that the images produced by the UNet are more realistic than those produced by the ResNet architecture. In particular the UNet architecture seem better at producing the right colors for the different aspects of the image. For example, in the second image of Figure 7 we observe that the UNet is able to pick the right color for the output whereas the ResNets choose brown (likely as result of the fact that most buildings in the training dataset are brown). In addition we see that increasing the size of the UNet (from 128 to 256) improve image translation. An example would be image 1, where UNet 256 is able to identify the windows of the biulding and color them blue while the UNet 128 is not able to do so.

An intuitive reason behind why UNet architectures perform better than ResNet in both Pix2Pix and CycleGAN could be that the UNet skip connections are made between $(i)th$ and the corresponding $(n-i)th$ layer and this sym-

metric nature of the skip connections contributes effectively to the transfer of necessary specific features to the generated image. It is important to also note the UNET-256 has 54.41 mn parameters while the ResNet 9K has only 11.38 mn parameters. Therefore, the difference in performance can also e attributed to the difference in number of parameters available to the two architectures.

Figure 6. Pix2Pix: Sample test outputs using generators trained using the different architectures. We see that the UNet 256 generator is overall more consistent in terms of the quality of images generated and has the least artifacts with sharper features

Figure 7. CycleGAN: Sample test outputs using generators trained using the different architectures for the CycleGAN model

3.3. Experiment using different pool size in updating discriminator gradients

Salismans *et al*. [13] explains a potential reason for GAN training mode collapse as being caused due to the fact that the discriminator processes examples independently leading to an inability to inform the generator to generate outputs that are dissimilar to one another. So, outputs tend to race towards a point that the discriminator believes is highly realistic. Now once this has occurred, it is found that a recovery to achieve a distribution with optimum amount of entropy is difficult. To avoid this, the discriminator is allowed to learn from a minibatch of previously generated examples which are saved in a buffer. Here, we experiment with different pool sizes to see how these impact the trained model and consequently our test results.

Figure 8. CycleGAN: Sample test outputs using generators trained using the different architectures for the CycleGAN model

Pix2Pix In our experiment we use a pool sizes of 0, 25, 50 and 100. We see that both pool sizes - 25 and 50 generate considerably better results in comparison to using pool sizes of 0 and 100. While the differences are not easy to see at first glance, a careful observation of the generated outputs will show that the features are consistently sharper and better for pool size of 50 in comparison to the others. Sample outputs are shown in Figure 9.

CycleGAN For CycleGAN, we tried pool size of 0, 25 and 50 pool sizes. From figure 10, we notice that pool size of 50 generates images with more depth and less blur as compared to those generated with pool size of 0 and 25.

3.4. Comparision between Pix2Pix and CycleGAN

For all our experiments, we found CycleGAN to produce more blurred images as compared to Pix2Pix even when comparing its 25th epoch (Figure 4) to the 5th epoch of Pix2Pix (Figure 3). This could be explained by the rationale that Pix2Pix is a supervised model as opposed to CycleGAN, which is an unsupervised approach.

3.5. VAE-cGAN

The VAE-cGAN is similar to the cGAN employed in the pix2pix model, the only difference being that the generator of the cGAN is replaced with a VAE (the discriminator is still conditioned on the real image B).

The structure of the Variational Autoencoder followed the standard architectural layout (of a VAE) where the en-

Figure 9. Pix2Pix: Sample test outputs on training using various pool sizes to store previously generated images. We see that training using pool sizes of 25 and 50 generates better images than that done using 0 and 100

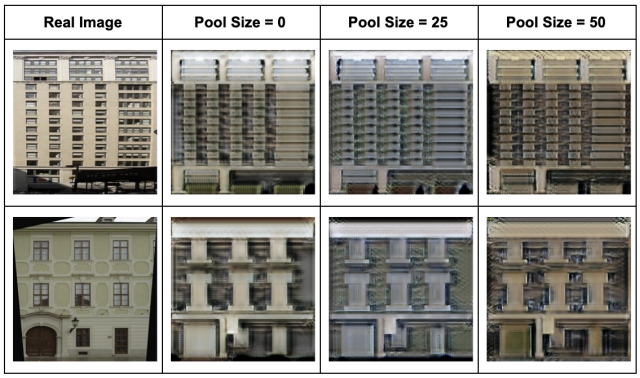

Figure 10. CycleGAN: Sample test outputs using various pool sizes to store previously generated images. We see that training using pool sizes of 50 generates better images than that done using 0 and 25

coder was composed of convolutional layers alternating with batch normalization while the decoder had transposed convolutional layers which also alternated with batch normalization layers. (It should be noted that between the encoder and decoder, we employed the re-parametrization trick, which is what differentiates a Variational Autoencoder from an Autoencoder).

When we ran the VAE-cGAN model, we observed that the VAE was not able to produce realistic images, even after several epochs of training. This can be observed from Figure 11 where the losses of the discriminator is consistently low while the generator is consistently high across epochs. This implies that the discriminator was easily able to predict that the images produced by the VAE were fake. When we analysed the images the VAE produced, it became clear why. From figure 12 it can be seen that the VAE was only producing images that were uniformly brownish-grey (the likely color of building facades) in color. Therefore, it became clear the VAE generator was not working.

Usually, while training VAEs, in addition to the reconstruction loss, the KL divergence loss (based on the μ and σ that is derived from the latent space of the images) is also added. We removed the KL divergence loss from the loss function and observed that the images being produced were no longer uniformly grey. From figure 13 we can see that the images produced continue to be without coherent structure. However, we can observe some hopeful signs of improved performance over the case where KL Divergence loss is included. For example in image two, the generated image has been colored significantly in red, like the corresponding real image B. We should note, that we cannot read too much into such findings as the images produced were clearly lacking in their ability to reconstruct the facades.

There could be multiple reasons why our VAE-GAN did not perform well. The most important reason is that GANs (and VAEs) in general are very difficult to train and very sensitive to hyper-parameter tuning. They routinely go into mode collapse (producing only a small set of image) or produce images of the most likely color so as to maximise performance. Though we did try hyperparameter tuning in an effort to improve performance, we could not observe much change in performance. From our knowledge a VAE-cGAN has never been employed before in image-to-image translation task, therefore, we were not able to find any heuristics to help improve the training of the VAE-cGAN.

In addition, to this, we should note that the VAE-cGAN we are using has one significant difference from the usual VAE-GANs, in that the discriminator in ours is conditioned on the real image B (just like in the pix2pix model). Therefore, it could be that because the discriminator is conditioned on the original image, the discriminator becomes very good at differentiating real and fake images, thereby not allowing the VAE to learn, as both a bad image and a slightly better but still bad image will result in equally bad losses. While this can be a problems in GANs in usual (where the discriminator overpowers the generator), in our case the difference between the generator and discriminator, could have been accentuated by the fact that discriminator is also conditioned on the original image.

One possible way to improve the performance of the VAE-cGAN is to instead use a cVAE-cGAN where we replace the VAE with a conditional VAE. In cVAEs, the encoder and the decoder, receive the class of the images as input as well (are conditioned on the class label). In the case of image-to-image translation there are no such classes, therefore instead we would have to condition the decoder on

the real image B. The decoder usually takes as input a vector z, which is sampled from the latent feature space. We would now have to concatenate this vector z with a vector representing the real image B. If we simply flatten the image (256x256x3) we would get an extremely large vector of the order 196K, which would not be prudent. We could instead perform max/average pooling to significantly reduce the size (to about 20x20x3 or a vector of size 1200) of the image before flattening. The downfall however is that it will lose significant amount of information in the process. Future work in this area of study could look at using this approach to see if performance can improve, as conditioning the decoder on the real image B could improve the ability of the VAE to produce realistic images.

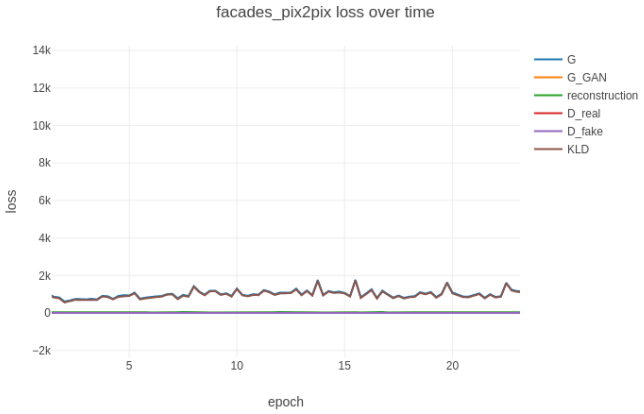

Figure 11. VAE training loss curve

4. Miscellaneous

4.1. Contributions to the Project

Table 1 (next page) contains the contributions of all the team members.

5. Conclusion

In our experiments, we obtain better results for pix2pix model as compared to CycleGAN. This was expected as pix2pix is a supervised approach for image to image translation task. For both pix2pix and CycleGAN, LSGAN converges faster than Vannila GAN and WGAN-GP. We also found UNet architecture for generator module to perform better than Resnet based architectures. Further, our experiments also conclude that adding a pool-size to buffer generated images while updating the discriminator gradients helps generate better results. We also proposed and experimented with a VAE-cGAN for image to image translation task. Though our model didn't produce realistic images, there is scope for exploration and would require future work to obtain better results.

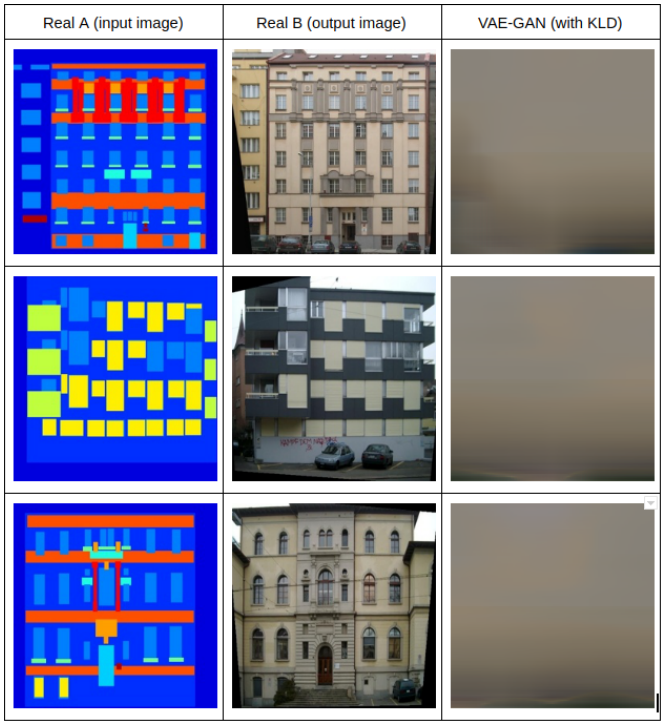

Figure 12. Image produced by VAE (with KL Divergence)

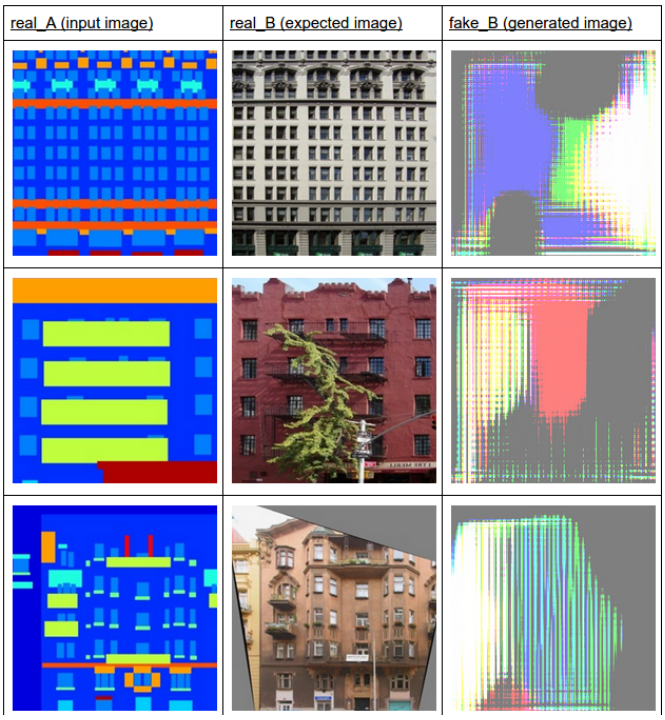

Figure 13. Image produced by VAE (without KL Divergence)

References

[1] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein gan, 2017. 4

Name	Contributed Aspects	Details
Arvind	CycleGAN experiments and VAE-cGAN implementation	Trained and analyzed the CycleGAN for different Generator architectures; Researched on the applicability of VAE-cGAN for image-to-image translation; Implemented the initial framework for the VAE; Analysed results from VAE-cGAN model.
Aastha	CycleGAN and VAE-cGAN	Trained and analyzed the CycleGAN for different objective functions and pool size for discriminator; Researched on VAEs models for image-to-image translation; Implementation of cVAE-cGAN; Training and analysis of VAE models
Sreehari	Pix2Pix experiments and VAE-cGAN implementation	Trained and analyzed the pix2pix model to perform all 3 experiments; Implemented the changes to integrate the VAE with the pix2pix discriminator alongwith changes to plug the code into the pix2pix framework; Trained the VAE-cGAN and experimented with different formulations for loss, sigmoid vs tanh, weight initializations, etc

Table 1. Contributions of team members.

[2] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks, 2014. 2, 4

[3] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron Courville. Improved training of wasserstein gans, 2017. 4

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015. 5

[5] P. Isola, J. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5967–5976, 2017. 1, 2, 3, 4

[6] Diederik P Kingma and Max Welling. Auto-encoding variational bayes, 2013. 2

[7] Anders Boesen Lindbo Larsen, Søren Kaae Sønderby, Hugo Larochelle, and Ole Winther. Autoencoding beyond pixels using a learned similarity metric, 2015. 3

[8] Chuan Li and Michael Wand. Precomputed real-time texture synthesis with markovian generative adversarial networks, 2016. 2

[9] Mario Lucic, Karol Kurach, Marcin Michalski, Sylvain Gelly, and Olivier Bousquet. Are gans created equal? a large-scale study, 2017. 3, 4

[10] Xudong Mao, Qing Li, Haoran Xie, Raymond Y. K. Lau, Zhen Wang, and Stephen Paul Smolley. Least squares generative adversarial networks, 2016. 4

[11] P. Isola R. Zhang and A. A. Efros. Colorful image colorization, 2016. ECCV. 2

[12] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation, 2015. 2

[13] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans, 2016. 6

[14] Kihyuk Sohn, Xinchen Yan, and Honglak Lee. Learning structured output representation using deep conditional generative models. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'15, page 3483–3491, Cambridge, MA, USA, 2015. MIT Press. 3

[15] Radim Tyleček and Radim Šára. Spatial pattern templates for recognition of objects with regular structure. In *Proc. GCPR*, Saarbrucken, Germany, 2013. 1, 3

[16] Saining "Xie and Zhuowen" Tu. Holistically-nested edge detection. In *Proceedings of IEEE International Conference on Computer Vision*, 2015. 2

[17] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks, 2017. 1, 2, 3