Image-to-image Translation Using Pix2pix GANs

Himanshu Sahni, Sheik Meeran Rasheed Abdul Rahuman,
Estéban Nocet-Binois, Vasiliki Kostara, Cyrena Howland
Chalmers University of Technology, Advanced Machine Learning with Neural Networks
(Group 8 - Team Octopi)

Abstract:
Image translations produced by generative AI are everywhere in our digital lives. From deepfakes, raising concerns about disinformation [1], to AI-assisted transformations for entertainment [2], these technologies are pervasive. We aim to move beyond these extremes, the motivating question behind our exploration being: *What kinds of AI technology produce these influential image translations, and what is their potential to be used constructively in society?* We find that Generative Adversarial Networks (GANs) are the driving force behind these technologies [1] [2] [3], and we explore two of the most well documented techniques of GANs and widely applicable to a variety of image-to-image translations: cGANs (pix2pix) and CycleGANs. We apply them to four translation tasks with potential constructive applications in society: satellite images to maps, semantic representations to building facades and their inverse translations. We then compare the relative performance of cGANs and CycleGANs in these tasks and consider whether its enough for them to be used in the real world applications we identified. We find that, for these particular applications, while pix2pix outperformed CycleGANs, it still didn't perform well enough to be directly applied. Despite this, with specific formulation for each application, we see potential for GANs to be a powerfully constructive tool in society alongside the more infamous and frivolous applications of disinformation and portrait generation.

I. INTRODUCTION

Generative Adversarial Networks (GANs) have emerged as a powerful framework for generating high resolution data, noted by their extraordinarily realistic output representations. GANs consist mainly of two neural networks, the generator and the discriminator [4]. The generator learns to produce samples, which the discriminator tries to distinguish from real input samples. Through this adversarial training method, GANs can successfully produce high quality realistic outputs and capture intricate details, characteristics that demonstrate their effectiveness against high complexity.

Since GANs were first introduced by Ian Goodfellow et al. [5], they have been vigorously studied and modified for different applications, with quite popular ones being image-to-image (I2I) translation tasks. In particular, conditional GANs (cGANs) and cycle-consistent GANs (CycleGANs) comprise popular variants of GANs, due to their revolutionary contributions to I2I translation [6] [7]. By adding conditional information on the input images [8], cGANs produce controlled outputs with specified details. On the other hand, CycleGANs can be effectively trained to map between different domains, even without aligned image pairs [9].

Inspired by this, we identified four image-to-image translation tasks to explore that had useful applications in society: satellite images to maps for applications like disaster response [10] [11] and maps to satellite images for advanced geographical information systems [12]; semantic representations to building facades and, inversely, building facade photos to semantic representations to help architects in historical preservation and renovation [13], the aim being to evaluate whether GANs have the potential to drive innovation and efficiency in various sectors, when used appropriately.

To determine which GANs method would work best for these tasks, we experimented with some of the most extensively documented and widely used architectures for image-to-image translation: cycle-consistent GANs (CycleGANs) and pix2pix GANs, a type of conditional GANs (cGANs). Pix2pix excels where paired data is available, making it ideal for tasks like converting maps to satellite images. CycleGANs, however, are suitable for a broader range of tasks where paired data is unavailable, like transforming building facade photos to semantic representations.

Our objective is to explore the effectiveness of these models in accomplishing our selected tasks, and to evaluate their viability in real world applications of these tasks. Are these models merely fascinating examples of what AI can do, or can they really be utilized to solve practical problems and deliver tangible value to society?

FIG. 1: AI generated image of Bollywood actress Deepika Padukone using Lensa AI application (powered by GANs) following social media trends.

II. BACKGROUND

A. Generative Adversarial Networks (GANs)

GANs are a class of deep learning models purposed to create new data instances resembling the training data. In order to achieve a satisfactory level of realism, GANs rely on pairing a generative with a discriminative multilayer perceptron, referred to as the generator and the discriminator respectively. A representation of a typical GANs architecture is presented in FIG. 2.

Firstly, the generator receives a noise vector input, $\mathbf{z}$, with prior distribution p_z, and maps its representation to data space, x, as an output through a function G [5]: $G : z \rightarrow x$. Afterwards, both the generated and the real samples become inputs to the discriminator. In turn, the discriminator outputs a single scalar, $D(\boldsymbol{x})$, representing the probability that $\mathbf{x}$ originates from real data instances, rather than generated ones.

Initially, the generator produces low quality samples, which improve along with training. However, in time, it's able to capture data correlations expressed by complex distributions [4]. This is achieved by backpropagation during training. Since, the discriminator aims to classify real and generated samples correctly, its parameters, θ_d, are updated through backpropagation by performing stochastic gradient ascent [5]:

$$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \left[logD\left(x^{(i)}\right) + log\left(1 - D\left(G\left(z^{(i)}\right)\right)\right)\right] \quad (1)$$

where m denotes the number of real and the number of fake samples provided to the discriminator. Afterwards, the generator parameters, θ_g, are updated to maximize $D(G(\mathbf{z}))$, in other words, the generated samples should be indistinguishable from real. For this purpose, the generator parameters are updated with stochastic gradient descent [5]:

$$-\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} log\left(1 - D\left(G\left(z^{(i)}\right)\right)\right) \quad (2)$$

Combining equations (1) and (2), the adversarial loss can be expressed as [5]:

$$\min_G \max_D V(D,G) = \mathbb{E}_{x\sim p_{data}(\mathbf{x})}\left[log\left(D(\mathbf{x})\right)\right] + \mathbb{E}_{z\sim p_z(\mathbf{z})}\left[log\left(1 - D\left(G(\mathbf{z})\right)\right)\right] \quad (3)$$

This process continues iteratively until global convergence. According to Goodfellow et al. [5], global GANs optimality can be theoretically achieved if and only if $p_g = p_{data}$. Thus, convergence can be practically achieved by optimizing θ_g. Eventually, convergence should yield generated samples that cannot be distinguished from real, when the classification process becomes completely random [4].

B. Conditional GANs (cGANs)

Conditional GANs constitute an extension of the general GAN framework. By adding conditional information to both the generator and the discriminator, cGANs achieve control over the generated data. The conditional information, denoted by $\mathbf{y}$, can take various forms depending on the desired output and the kind of application. For instance, in image generation $\mathbf{y}$ can take the form of class labels representing specific characteristics. Thus, with cGANs the outputs become directed, yet this contributes in wider applicability for many generation tasks.

GANs and cGANs are described by comparable architectures. Indeed, they both include a generating and a discriminating process. However, by introducing $\mathbf{y}$ the inputs and the outputs for each perceptron are modified along with their loss functions. In particular, as observed in FIG. 3, the input for the generator becomes the

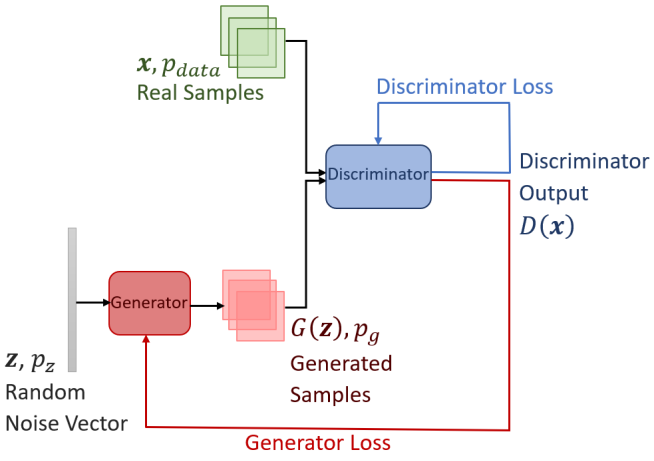


FIG. 2: Typical GAN architecture.

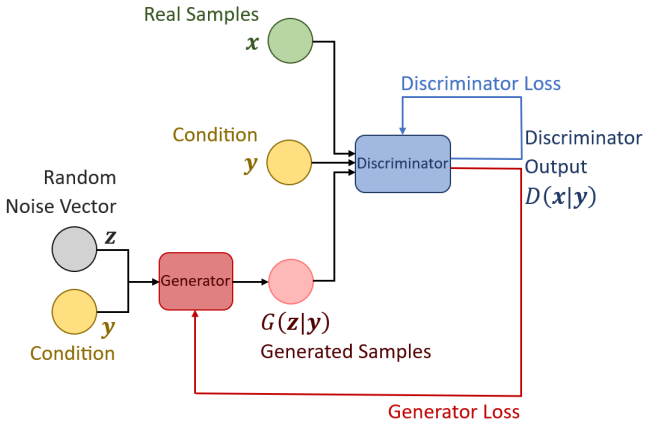


FIG. 3: Conditional GAN architecture.

prior p_z combined in joint hidden representation with $\mathbf{y}$ [14], while the output is generated by mapping the input to data space given the condition $\mathbf{y}$: $G(\mathbf{z}|\mathbf{y})$. Continuing, these generated conditioned samples are provided to the discriminator along with the real data samples and the conditional information, as an additional input layer. Still, the discriminator outputs a single scalar, yet in cGANs it represents the probability that $\mathbf{x}$ originates from real data instances given the condition $\mathbf{y}$: $D(\boldsymbol{x}|\mathbf{y})$.

Consequently, by introducing conditional stochasticity, the adversarial loss will be revised according to the output of the discriminator [14]:

$$\min_G \max_D V(D,G) = \mathbb{E}_{x\sim p_{data}(\mathbf{x})}\Big[log\big(D(\mathbf{x}|\mathbf{y})\big)\Big] + \mathbb{E}_{z\sim p_z(\mathbf{z})}\Big[log\Big(1 - D\Big(G\big(\mathbf{z}|\mathbf{y}\big)\Big)\Big)\Big] \quad (4)$$

What is more, adding a distance loss term between the real and the generated data can be beneficial to the generating process [8]. The significance of this lies in the fact that the generator is tasked to produce outputs converging to the real data. This loss term is expressed by using the Manhattan distance, L_1, between the two datasets:

$$\mathcal{L}_{L_1}(G) = \mathbb{E}_{\mathbf{x},\mathbf{y},\mathbf{z}}\Big[\|\mathbf{x} - G(\mathbf{z}|\mathbf{y})\|_1\Big] \quad (5)$$

Finally, the objective function of the network can be expressed as the sum of the adversarial loss and the distance loss, where λ is a scalar hyperparameter determining the importance of the distance loss compared to the adversarial loss.:

$$\mathcal{L}(D,G) = \min_G \max_D V(D,G) + \lambda\mathcal{L}_{L_1}(G) \quad (6)$$

C. Cycle-consistent GANs (CycleGANs) in Image-to-image Translation

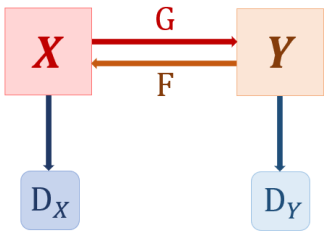


FIG. 4: Cycle-consistent GAN architecture.

CycleGANs are a variant of GANs that was introduced in 2017 [15]. They address a significant limitation found in many other GANs: the requirement for paired data that is not always feasible for various applications and real-life situations. CycleGANs overcome this limitation by allowing learning the mappings between two domains implicitly. This characteristic makes them a versatile and robust tool for image-to-image translation tasks.

CycleGANs consist of two generators and two discriminators (FIG. 4). The generators perform mapping between two domains, x and y in each direction: $G : x \to y$ and inversely $F : y \to x$, while the discriminators classify if an image is generated or it originates from a domain. Coherent mappings in both directions are achieved through the cycle consistency loss. This loss quantifies the discrepancy between the original image and the image obtained after the cycle translation process by calculating the mean absolute difference between the two images. According to cycle consistency, translating an image in one direction and then reversing the translation using the trained generators should result in an image that is close to the original input image. Therefore, cycle consistency loss is comprised of two components: forward and backward cycle consistency losses, as shown in Eq. 7:

$$\mathcal{L}_{\text{cyc}}(G,F) = \mathbb{E}_{x\sim p_{data}(x)}\Big[\|F\big(G(x)\big) - x\|_1\Big] + \mathbb{E}_{y\sim p_{data}(y)}\Big[\|G\big(F(y)\big) - y\|_1\Big] \quad (7)$$

As shown in Equation 8, one can include the hyperparameter λ in the objective function to control the model's emphasis on preserving cycle consistency:

$$\mathcal{L}(G,F,D_x,D_y) = \min_G \max_{D_y} V(D_y,G) + \min_F \max_{D_x} V(D_x,F) + \lambda\mathcal{L}_{\text{cyc}}(G,F) \quad (8)$$

To conclude, cGANs allow for more precise control over generated outputs, while CycleGANs offer more flexibility and is thus useful for more creative tasks. That is why cGANs are preferred in supervised tasks like semantic to real image, where we can get enough data mapping directly instances from the two domains. On the other hand, CycleGANs are generally used for tasks like style transfer and domain adaptation, where getting paired data can be a tedious process. The choice between them depends therefore on data availability, desired control, and specific application needs.

III. METHOD

A. pix2pix

1. *Architecture*

Primarily, we were inspired by the methods and network architectures described by Phillip Isola et al. in [8]. Our implementation aimed to generate realistic and visually coherent output images based on the given input images. Like [8], for both our discriminator and generator architectures we used networks of the form `convolution-BatchNorm-ReLu`. The key features are discussed below, while more detailed information can be found in the supplementary materials document.

(i) Generator

For the generator component, we designed an encoder-decoder U-Net architecture to facilitate the translation process. We incorporated skip connections between corresponding layers of the encoder and decoder to preserve fine details during the translation process. The complete architecture for the generator is provided in supplementary materials document.

(ii) Discriminator

Based on the work of [8] and [14], we implemented a discriminator architecture known as PatchGAN. This architecture focuses on assessing the authenticity of image patches at a specific scale. By applying the discriminator convolutionally across the entire image, we obtained the average responses of all patches, which ultimately determined the discriminator's output. The PatchGAN discriminator aims to classify whether each $N \times N$ patch in the image is real or fake, thereby enabling a more localized and detailed analysis of the image structure during the adversarial training process. The complete architecture for the discriminator is provided in the supplementary materials document.

2. *Loss Functions*

(i) Discriminator Loss

In our PatchGAN, which produces an $N \times N$ output grid to assess overlapping patches in the input image, the output size was $30 \times 30 \times 1$, corresponding to predictions for 70×70 patches. To evaluate the generated images, we computed the loss by applying sigmoid cross-entropy on the $N \times N$ output grid with a zero matrix of the same size, which is referred to as the generated loss. Additionally, we calculated the real loss by pairing the input images alongside their corresponding target images from our dataset. The real loss was determined by applying the sigmoid cross-entropy between the $N \times N$ output and a matrix of ones matching the $N \times N$ size. By summing the generated loss and the real loss, we obtained the total discriminator loss (FIG. 5).

(ii) Generator Loss

Similarly to the real loss described above, our generator loss was determined by applying the sigmoid cross-entropy between the $N \times N$ discriminator output and a matrix of ones. To enhance the visual quality of the generated images, we introduced an L_1 loss term (FIG. 6). This additional term was employed to calculate the L_1 distance between the target image and the generated image, as described in section II B. To control the impact of the L_1 loss, we multiplied it by the λ parameter before incorporating it into the overall generator loss. By including the L_1 loss term, our intention was to guide the generator to produce images that closely matched the target images in terms of visual details. This supplementary loss term proves its significance in training the generator, driving it towards a more visually appealing and realistic output, based on the provided input.

3. *Training and Optimization*

To optimize our networks, we practiced the standard approach from [5]: we alternated between one gradient

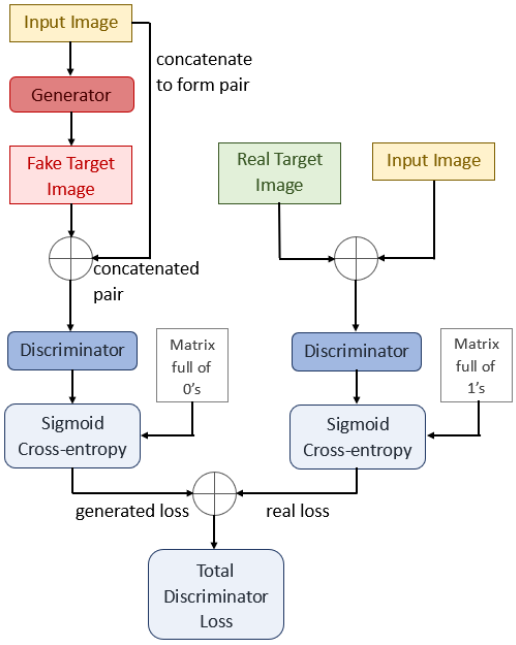


FIG. 5: Discriminator loss diagram.

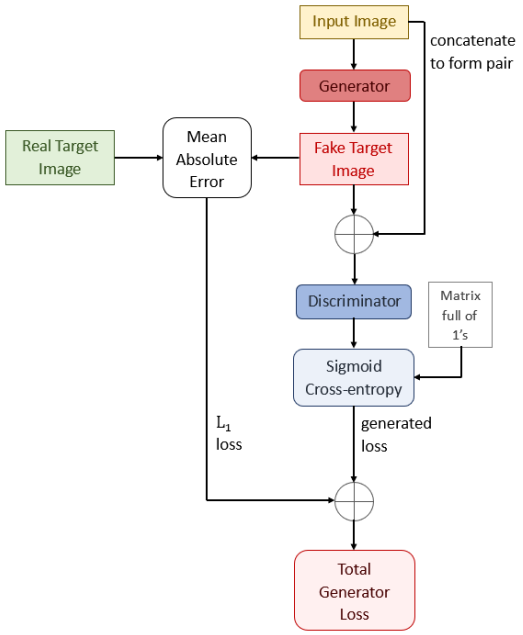


FIG. 6: Generator loss diagram.

descent step on Discriminator, then one step on Generator. We applied the Adam solver [16], with learning rate $\alpha = 0.0002$ and, initially, momentum parameters $\beta_1 = 0.5$ and $\beta_2 = 0.999$. Lastly, to mitigate the execution of computationally intensive code, we set batch size to 1.

B. CycleGAN

As an extracurricular part of this project, we steered our interest in CycleGANs [17]. In this implementation, the generators transform images between two domains using convolutional layers, ResNet blocks to capture and preserve image details and transpose convolutional layers, which produced the final output image. The discriminator architectures consisted of several convolutional layers with increasing numbers of filters. Each convolutional layer was followed by an InstanceNormalization layer and a LeakyReLU activation. The spatial dimensions of the image were gradually reduced, after using convolutional layers with strides. The discriminator's final layer was a convolutional layer with a single filter, which produced a patch-level output in the same way as Pix2Pix.

The main difference with the cGAN implementation was the lack of any conditional data as input, neither for the discriminator, nor for the generator. Instead, the overall generator model contained two generators and the overall discriminator model, two discriminators respectively, performing translation in both directions and propagating feedback using two adversarial loss functions for each translation, as described in section II C. However, additionally to the cycle-consistency loss, an identity loss is included. Although the cycle-consistency loss computes an absolute mean distance between the two domains, the identity loss computes an absolute mean distance ensuring that an image remains unchanged, if provided to its original domain and directly translated to it [15]: $\mathcal{L}_{\text{identity}}(G, F) = \mathbb{E}_{y \sim p_{data}(y)}\Big[\|G(y) - y\|_1\Big] + \mathbb{E}_{x \sim p_{data}(x)}\Big[\|F(x) - x\|_1\Big]$.

IV. EXPERIMENTS & EXECUTION

A. pix2pix

1. Comparison of Discriminator Momentum Parameter β_1

During training, we explored multiple β_1 momentum parameter values for the discriminator, including 0.5 and 0.99. We varied this discriminator momentum parameter to investigate its impact on training dynamics. From here on out, when we talk about changing a momentum parameter and setting it to a particular value, we are referring to the discriminator's momentum parameter, and the generator's momentum parameter is kept constant at 0.5. By varying this momentum parameter, we aimed to assess convergence speed, stability, and the model's ability to learn effectively. A lower momentum (0.5) facilitates faster adaptation to new gradients, potentially leading to quicker convergence in simpler landscapes, while a higher momentum (0.99) promotes stability and prevents convergence in sub-optimal solutions. By comparing the models trained with different discriminator momentum parameters, we sought to identify the optimal setting for improved performance, faster convergence, and enhanced generalization capabilities.

2. ImageGAN vs 70×70 PatchGAN

We investigated the effect of changing the patch size as well. We compared the 70×70 PatchGAN to an ImageGAN. The PatchGAN architecture, with a smaller receptive field (70×70), focuses on analyzing local image patches, allowing for more fine-grained discrimination. On the other hand, the ImageGAN architecture operates at the full resolution of the image, capturing global information and considering the image as a whole. We aimed to determine which architecture produces better results in terms of image quality and realism. By evaluating the outputs, we aimed to gain insights into the effectiveness of these architectures.

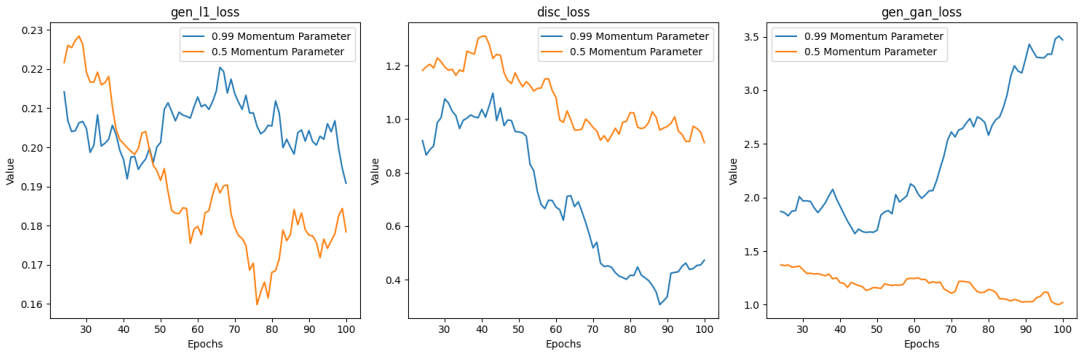
3. Generality of the pix2pix GAN

After exploring different architectures and hyperparameters to determine the optimal parameters we explore the generality of pix2pix GANs, by testing it on a variety of tasks and datasets such as:

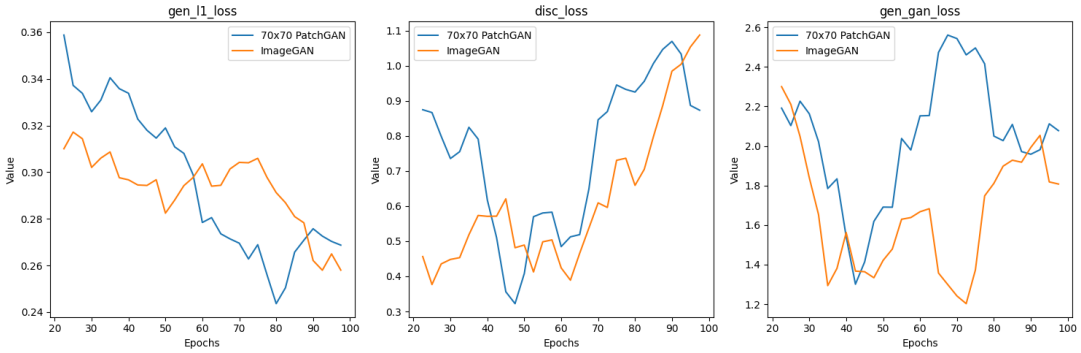
- Architectural labels $\longleftrightarrow$ photo, trained on CMP Facades [18].
- Map $\longleftrightarrow$ aerial photo, trained on data scraped from Google Maps [19].

B. CycleGAN

We utilized CycleGAN for the translation of real photos to semantic architectural labels. Despite having access to a paired dataset for this particular task, we still opted to employ CycleGAN due to its ability to learn mappings between domains without relying on explicitly paired training examples. By utilizing the paired dataset, we aimed to assess the performance and effectiveness of CycleGAN in capturing the desired structural and stylistic characteristics of the semantic facades from real photos. This allowed us to compare the outcomes of CycleGAN with our previous experiments using pix2pix GAN.



(a) Comparison between 0.5 and 0.99 momentum parameter models using PatchGAN and window size 25.



(b) Comparison between PatchGAN and ImageGAN models, with momentum parameter 0.5 and window size 10.

FIG. 7: Comparison of the moving average of loss graphs for the translation of maps to aerial photos over 100 epochs. From left to right: L_1 loss (`gen_l1_loss`), total discriminator loss (`disc_loss`) and generator adversarial loss (`gen_gan_loss`).

V. RESULTS

In this section we present indicative results of our methods and experiments. More outputs of our implementations of pix2pix and CycleGANs can be found in the supplementary materials document.

A. pix2pix

1. *Comparison of* 0.5 *versus* 0.99 *momentum parameter*

In FIG. 7 we present a comparison of the loss graphs for the translation of maps to aerial photos. According to the depiction of the L_1 loss in Fig. 7a, with a momentum parameter of 0.5 the model performs slightly better than 0.99, showing that the generator is learning to produce images comparable to the target image slightly better with 0.5. On the right, when comparing the discriminator and the generator loss graphs, one can notice that with a 0.99 momentum parameter, the discriminator is dominating over the generator, indicating that the training is not converging. Therefore, a momentum parameter of 0.5 yields preferable results.

2. *Comparison of PatchGAN vs ImageGAN*

Based on the loss graphs presented in Fig. 7b, both models perform adequately. After 100 epochs, the PatchGAN model demonstrates marginally superior results regarding the L_1 loss. Moreover, concerning the generator and discriminator losses, both models perform similarly without either component overpowering the other. After examining the image outputs in the supplementary materials document, one can verify that both models accurately capture the structural characteristics of the buildings, despite the limitation in capturing color information. Notably, the 70×70 PatchGAN model utilizes fewer parameters and can be applied to higher resolution images. Hence, employing PatchGAN for such applications yields results equivalent to those of ImageGAN, while requiring less computational power and offering broader applicability.

3. *Image outputs on both datasets*

We present on Fig. 8 the image outputs for the two datasets: facades and aerial images. For the discriminator we utilize PatchGAN with a momentum parameter of

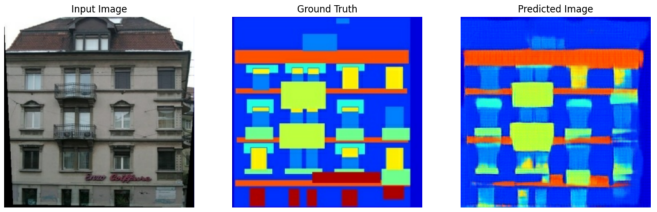
0.5. After training for 100 epochs, we can visually assess the quality of the generated images.

Regarding the facade dataset, the generated images exhibit significant results in terms of structural coherence and realistic texture. For the real to semantic images (Fig. 8a), the model effectively captures the structural details of the facades, such as windows, doors, balconies and decorative features, resulting in visually coherent schematic outputs that closely resemble the ground truth facades. For the semantic to real images (Fig. 8b), the generated images showcase clear boundaries and coherent architectural elements, even though they may not perfectly match the ground truth. Specifically, the lack of detailed elements consists of the clear delimitation of the building, as well as information on the real color and textural elements.

Similarly, the Pix2Pix model exhibits excellent performance for the aerial images dataset, achieving visual quality and fidelity. For the semantic to real images (Fig. 8c), the model successfully learns and reproduces the underlying patterns and characteristics of the aerial maps and converts it to realistic satellite images. It reproduces the buildings and roads, adding vegetation appropriately, as well as reproducing bodies of water or forests. For the real to semantic aerial images (Fig. 8d), it competently extracts structural information out of the satellite images, such as buildings, roads and green areas.

In order to assess the ability of our model to continuously improve, we additionally ran the translation from maps to satellite for 200 epochs. In particular, in terms of improving the sharpness of various elements, one can realise in Fig. 9 that the overall quality exceeds that of Fig. 8d, producing more defined buildings and green areas with sharper edges.

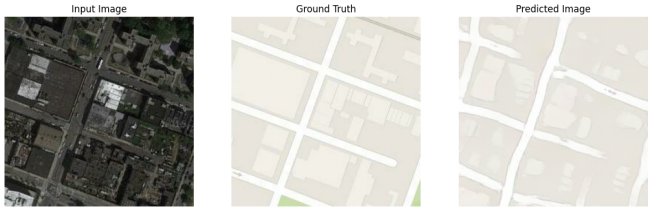
Overall, the best results with pix2pix on our translation tasks were when we used the bilinear resizing technique during prepossessing of images when relevant (Aerial dataset), applied a 70×70 PatchGAN or ImageGAN and set the discriminator momentum parameter to 0.5, instead of 0.99.



(a) Photo to Facade

(b) Facade to Photo

(c) Aerial map to Satellite image



(d) Satellite image to Aerial map

FIG. 8: Image output for different datasets, using PatchGAN (with momentum parameter 0.5) for Pix2Pix after 100 epochs.

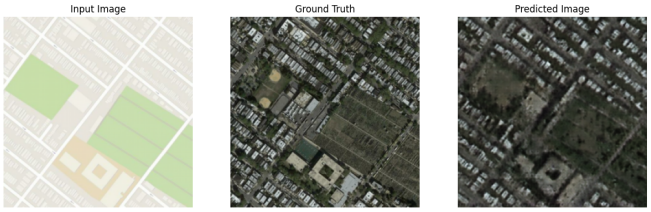
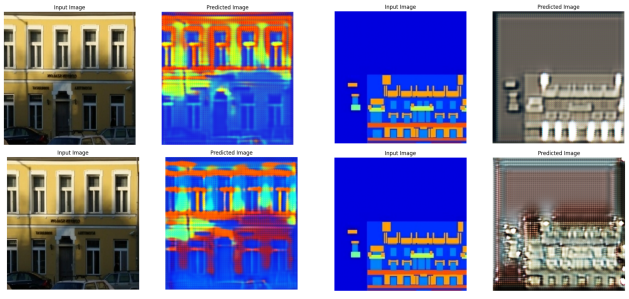


FIG. 9: Image outputs for Satellite to Maps after 200 epochs, using PatchGAN

B. CycleGAN

After several time-consuming trials, we observed that the results obtained by CycleGAN were noticeably worse than those achieved with pix2pix GANs. We did not gain any significant improvement over the different periods in which the models were trained, as visible in Fig. 10.



(a) Photo to Semantic (b) Semantic to Photo

FIG. 10: CycleGAN outputs for the Facade dataset over different epochs (10 and 30 from top to bottom).

Several factors hindered the CycleGAN's progress. Primarily, the complexity of this model, due to its purpose of mapping in two domains, and the constraints imposed by limited time and GPU resources. In further analysis, CycleGAN is a more intricate model, comprising of two generator and two discriminator networks, while pix2pix GANs includes only one of each. The increased complexity of CycleGAN involves a higher number of parameters and layers, rendering it overly challenging to train effectively, especially when paired data is available. Consequently, each epoch of training required a substantial amount of time to complete, compared to pix2pix GANs, or any cGANs, which was approximately ten times faster.The GPU resource constraints further compounded the issue. The extended running times of CycleGAN significantly limited the number of epochs that could be executed. As a result, the model underwent insufficient training, producing undesirable outputs.

VI. DISCUSSION

If given more time, there are a few things we would have liked to explore further with the pix2pixGANs.

Our reference pix2pix study [8] used a convolutional translation, where their PatchGAN could be applied to arbitrarily large images. They did this by applying the generator convolutionally on larger images than those it was trained on. Since their technique didn't require image downsizing, it didn't result in the loss of clarity and resolution. Yet, our resizing did, especially with the Aerial dataset.

Since our ultimate intention was to evaluate the performance of these techniques and determine if it was enough to be directly applied to real world applications, a more empirical and objective way to measure this would have been ideal. Candidates include perceptual studies using Amazon Mechanical Turk (AMT), to collect feedback and ratings from a diverse group of human evaluators. Additionally, we would have also liked to utilize FCN scores, which measure the performance of the generated images using pre-trained classifiers. By combining perceptual studies with FCN scores, we can obtain comprehensive insights into the perceptual quality, visual fidelity, and semantic accuracy of our cGAN models.

In summary, the findings presented in this review indicate that cGANs show great potential for image-to-image translation tasks, particularly when dealing with complex and structured graphical outputs. These networks have the ability to learn task-specific loss functions tailored to the given data, making them suitable for a wide range of applications and scenarios.

On the other hand, CycleGANs proved challenging, despite their promising applicability. Even with the simplest implementations, they can burden with computational complexity and resource demands. Therefore, to overcome these disadvantages, CycleGANs should be employed with more specialized hardware that includes ample memory capacity, graphics and processor power, and be utilized where its strengths are required, such as with unpaired datasets. Our application of CycleGANs was not the best use of its strengths, since we had paired datasets. However, due to CycleGANs potential, we would like to apply it to tasks its more suited for, as it could undoubtedly have many useful applications.

VII. CONCLUSION

We began this exploration hoping to gain insight into GANs potential usefulness to society. This is a broad and general question, but we chose to explore it by choosing four particular image-to-image translation tasks that had potentially useful applications. These were satellite images to maps which has applications like disaster response [10] [11] and maps to satellite photos for advanced geographical information systems [12]; semantic representations to building facades and, inversely, building facades photos to semantic representations to help architects in historical preservation and renovation [13]. The question then is, at the end of our exploration: *Did pix2pix or CycleGANs perform enough to be able to be used and make a meaningful impact to these fields?*

Although remarkably good at a wide range of tasks, the particular techniques didn't perform well enough to be directly applied to real world applications, like the ones we mentioned. For example, pix2pix for translating satellite images to maps for disaster response, although good at capturing underlying structure, sometimes produced results that could be mistaken as destroyed buildings, and deformed roads and earth that could be mistaken for evidence of natural disaster. (see Fig. 2a in Supplementary Materials). Having thoroughly explored some of the most widely applicable and well documented GANs for image-to-image translation, the next step would be to explore other GANs architectures that may be better for the specific applications we think GANs might have a useful impact to. GANs that have been applied with some success, for disaster response for example include:

- Physics-informed GANs for coastal flood visualization which uses a modified pix2pix called pix2pixHD which produces imagery that is physically-consistent. [11]

- SAM-GAN, which introduces image style loss and topological consistency loss to improve the model's pixel-level accuracy and topological performance for map to satellite translations. [20]

All in all, GANs formulated for specific image-to-image translation tasks have the potential to be incredibly constructive to society but alone, pix2pix or CycleGANs alone would not be enough for the applications we considered.

VIII. ACKNOWLEDGEMENTS

We would like to express our deepest gratitude to our project supervisor and teaching assistant at Chalmers University of Technology, Harshith Bachimanchi, for his invaluable guidance and the teacher at Chalmers University of Technology, Giovanni Volpe, for his helpful feedback. We would also like to thank our family and friends for providing us with moral support and constructive criticism throughout the entire study period. Lastly, we would like to note our own excellent cooperation and state that each one of us contributed equally to every part of this project.

[1] Thumos, Generative ai and deepfakes (2022).
[2] M. M. Kircher and C. Holtermann, How is everyone making those a.i. selfies?, The New York Times (2022).
[3] T. Shen, R. Liu, J. Bai, and Z. Li, "Deep Fakes" using generative adversarial networks (gan), ECE228 Class Reports (2018).
[4] Google Developers, Gan (generative adversarial networks) overview, `https://developers.google.com/machine-learning/gan/gan_structure` (Accessed 2023).
[5] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, Generative adversarial nets, arXiv preprint arXiv:1406.2661 (2014), arXiv:1406.2661 [stat.ML].
[6] Y. Pang, J. Lin, T. Qin, and Z. Chen, Image-to-image translation: Methods and applications, IEEE Transactions on Multimedia **24**, 3859 (2022).
[7] A. Shokraei Fard, D. C. Reutens, and V. Vegh, From cnns to gans for cross-modality medical image estimation, Computers in Biology and Medicine **146**, 105556 (2022).
[8] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, Image-to-image translation with conditional adversarial networks, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, (2017).
[9] J. Hoffman, E. Tzeng, T. Park, J.-Y. Zhu, P. Isola, K. Saenko, A. A. Efros, and T. Darrell, Cycada: Cycle-consistent adversarial domain adaptation, arXiv preprint arXiv:1711.03213 (2017), submitted on 8 Nov 2017 (v1), last revised 29 Dec 2017 (this version, v3), arXiv:1711.03213 [cs.CV].
[10] B. Lütjens, B. Leshchinskiy, C. Requena-Mesa, F. Chishtie, N. Díaz-Rodríguez, O. Boulais, A. Piña, D. Newman, A. Lavin, Y. Gal, and C. Raïssi, Physics-informed gans for coastal flood visualization, (2021).
[11] X. Rui, Y. Cao, X. Yuan, Yu Kang, and W. Song, Disastergan: Generative adversarial networks for remote sensing disaster image generation, (2021).
[12] A. Timsina, P. Chhetri, S. Kamat, S. Thapa, D. R. Shah, R. Shah, and R. Sharma, Mapgan: Generating maps from aerial images, International Journal of Advanced Engineering (2022).
[13] C. Sun, Y. Zhou, and Y. Han, Automatic generation of architecture facade for historical urban renovation using generative adversarial network, Building and Environment **212**, 108781 (2022).
[14] M. Mirza and S. Osindero, Conditional generative adversarial nets, arXiv preprint arXiv:1411.1784 (2014).
[15] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, Unpaired image-to-image translation using cycle-consistent adversarial networks, arXiv preprint arXiv:1703.10593 (2017).
[16] D. P. Kingma and J. Ba, Adam: A method for stochastic optimization, arXiv preprint arXiv:1412.6980v9 (2015).
[17] Paperspace, Unpaired image-to-image translations with cyclegans (2021).
[18] TensorFlow CycleGAN, Cyclegan facades dataset, `https://www.tensorflow.org/datasets/catalog/cycle_gan#cycle_ganfacades` (Year).
[19] TensorFlow CycleGAN, Cyclegan maps dataset, `https://www.tensorflow.org/datasets/catalog/cycle_gan#cycle_ganmaps` (Year).
[20] C. H. B. D. Jian XuORCID, Xiaowen Zhou and H. Li, Sam-gan: Supervised learning-based aerial image-to-map translation via generative adversarial networks, (2023).
[21] R. Tyleček and R. Šára, Spatial pattern templates for recognition of objects with regular structure, *Pattern Recognition*, , 364 (2013).
[22] DrivenData, Building segmentation for disaster resilience (2022).
[23] Paperspace, A complete guide to gans (2021).