Team 46: Generation of Real World Images from Simulation Images

Nischal Maharjan
7058343

Jaykumar Bhagiya
7055903

Hevra Petekkaya
7055462

Abstract

In machine learning, data abundance is crucial, especially with the rise of deep learning, which requires large datasets to learn meaningful patterns due to its complexity. In the context of self-driving cars, collecting real-world data is challenging, so researchers often rely on simulated data, which can lead to distribution shift issues. Our approach aims to address this by generating realistic images from simulated data using a generator-discriminator model trained with adversarial loss for image translation.

1. Introduction

The success of deep learning since 2012, especially in image recognition, highlighted the need for large labeled datasets like ImageNet. The emergence of transformer-based architectures in NLP has amplified this demand, as both deep learning and transformer models perform better with large datasets.

In autonomous driving, ensuring reliability in diverse environments is crucial due to varying weather conditions and dynamic elements like pedestrians and vehicles. However, collecting diverse real-world data is challenging due to the high cost of precise equipment, accurate labeling needs, and unpredictable weather. To address these issues, researchers have increasingly turned to simulated data using tools like **CARLA, LGSVL, AirSim, and Apollo Simulation,** which simplify dataset generation for self-driving cars. These tools provide the easy method for generation of datasets for self-driving cars.

Despite this, models trained solely on simulated data often lack robustness in real-world scenarios, as there is minimal transfer of robustness from synthetic to natural distribution shifts, as noted in "Measuring Robustness to Natural Distribution Shifts in Image Classification" [8]. This project aims to balance the ease of dataset generation with model robustness. We evaluated a YOLO model trained on both real and simulated images for car detection and found that while it performed well on its own dataset, its performance dropped when tested on real images, as shown in Table 1. The goal is to generate more realistic images from simulated ones to address data scarcity.

Trained on	Evaluated on	mAP50
Real images	Real images	0.976
Simulated images	Simulated images	0.99
Simulated images	Real images	**0.752**

Table 1. Effect of Distribution shift. Yolo model was trained and evaluated on both real image and simulated images

2. Related Works

The Pix2Pix model [4] leverages conditional adversarial networks for image translation, featuring a UNet-based generator and a PatchGAN discriminator trained with adversarial and L1 losses. This approach maps between domains using paired images, where adversarial loss is essential to avoid blurry outputs typical of autoencoders relying solely on reconstruction loss.

For unpaired image translation, CycleGAN [9] employs two networks with separate discriminators: one generates predictions and the other reconstructs the input image, ensuring cycle consistency. Additionally, StyleGAN [5] offers style transfer by preserving high-level attributes while applying a given style. Although this project focused on paired images, these advanced techniques offer promising scopes for future enhancement.

3. Methodology

3.1. Dataset

For this project, we have leveraged the KITTI Vision Benchmark Suite dataset introduced in [3] and the Virtual KITTI dataset introduced in [2]. Virtual KITTI is a photo-realistic synthetic video dataset designed for learning and evaluating computer vision models for various video understanding tasks. The dataset contains a total of 2,126 image pairs, which we split into 1,488 pairs for the training set and 638 pairs for the validation set.

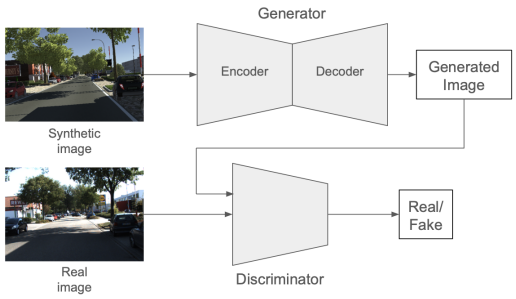

Figure 1. Image Translation Pipeline

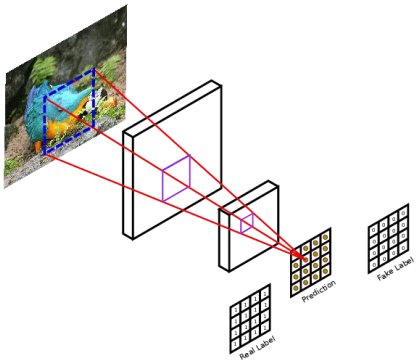

Figure 2. Patch Discriminator
Source

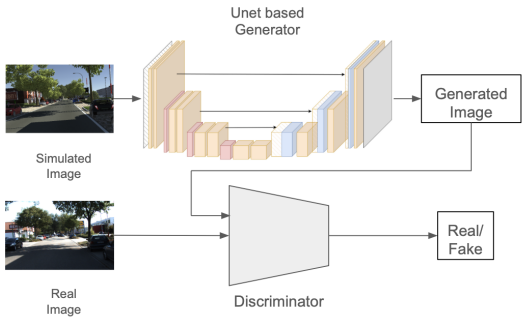

Figure 3. Unet Generator

3.2. General Image Translation Pipeline

Inspired by [4], the high-level architecture of our image translation pipeline is shown in Figure 1. The overall architecture consists of a Discriminator and a Generator, where the Generator is comprised of an Encoder and a Decoder. The Generator's task is to produce a realistic image from a simulated input image, while the Discriminator's role is to classify whether an image is real or generated. We use adversarial loss, as described in Equation 1

$$L_{cGAN} = \mathbb{E}_{x,y}[log(D(x,y)] + \mathbb{E}_{x,z}[log(1 - D(x, G(x,z)] \quad (1)$$

where D represents the Discriminator and G represents the Generator. The Generator aims to produce images that closely match the ground truth rather than merely fooling the Discriminator. To achieve this, we incorporate an L1 loss, as suggested by the authors of [4]. The L1 loss is described in Equation 2, and our final loss function is given in Equation 3.

$$L_{L1} = \mathbb{E}_{x,y,z}[||y - G(x,z)||_1] \quad (2)$$

$$L_{final} = L_{cGAN} + L_{L1} \quad (3)$$

3.2.1 Patch Discriminator

In traditional GANs, the Discriminator typically classifies the entire image as either real or fake. However, PatchGAN [4] employs a different approach, where the Discriminator evaluates and classifies each patch of the image independently, as illustrated in Figure 2. This patch-wise classification strategy offers several advantages, particularly in enhancing the quality of generated images. By focusing on smaller patches, the Discriminator becomes more sensitive to local features such as texture and fine details. This increased attention to detail encourages the Generator to produce images with more realistic and higher-quality textures. Additionally, PatchGAN reduces computational complexity, as the Discriminator only needs to assess individual patches rather than the entire image, optimizing the process without sacrificing accuracy.

3.3. U-Net based Architecture

In tasks requiring pixel-level precision, UNet architectures [7] are a preferred choice due to their ability to effectively capture both spatial and contextual information. Since our objective is to transform a synthetic image into a realistic one, the UNet-based generator is well-suited for this task. The input to the UNet is the synthetic image, and the output is a high-fidelity, realistic image.

The UNet architecture, as shown in Figure 3, consists of two main components: the encoder and the decoder. The encoder extracts essential features from the input image through a series of convolutional layers, each followed by down-sampling. This process compresses the spatial information into a dense, context-rich representation, capturing the underlying structure necessary for generating realistic details. The decoder then reconstructs the image from this dense representation by performing up-convolutions to gradually restore the spatial dimensions. This process is facilitated by skip connections, which link corresponding layers in the encoder and decoder. These skip connections ensure that fine details from the original image are preserved, leading to a more accurate and realistic output.

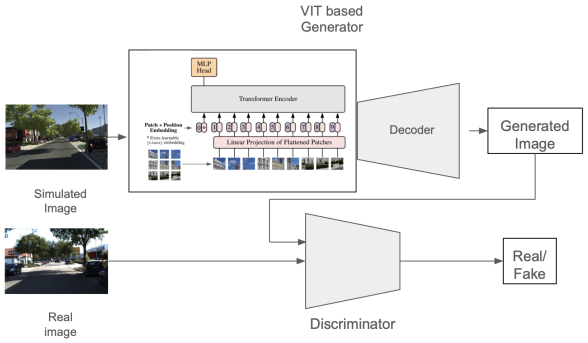

Figure 4. ViT based Generator

3.4. ViT based Architecture

As we previously employed the U-Net architecture for both the encoder and decoder in our generator, it initially delivered promising results. However, over time, the model began to overfit, particularly when processing complex synthetic images, which led us to explore alternative architectures. This prompted the integration of a Visual Transformer (ViT) [1] as the encoder within the generator, while retaining the conventional decoder, and maintaining the overall GAN architecture as seen in the figure 4.

The ViT is employed as the encoder by breaking down the input image into smaller non-overlapping patches, treating each patch as a token similar to words in NLP tasks. The ViT then applies a self-attention mechanism across these tokens, capturing long-range dependencies and global context within the image. These enriched feature representations from encoder are passed to a traditional decoder, which reconstructs the image in the generator. The overall architecture remains consistent with standard GANs: the generator, now equipped with the ViT encoder, produces an image from the synthetic input, while the discriminator evaluates its realism.

3.5. Swin-Transformer based Architecture

In Section 3.3, we used traditional CNN layers, while Section 3.4 introduced the attention mechanism through ViT. Although ViT outperforms the traditional UNet model by leveraging attention, it operates on fixed-size patches with uniform resolution and channels throughout. UNet, on the other hand, captures multi-scale information through its hierarchical structure, which adjusts resolution and increases channels as the network deepens—an advantage for detecting objects at various scales. The Swin Transformer [6] merges UNet's hierarchical, local-context processing with ViT's global context understanding. This hybrid architecture is well-suited for diverse vision tasks, including dense predictions like image generation. We replaced the entire generator model with the Swin Transformer-based

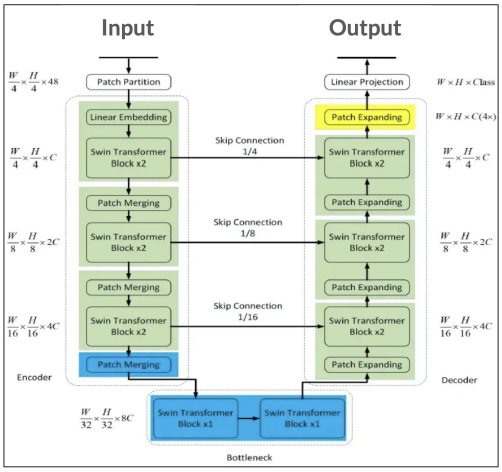

Figure 5. Swin Generator
Source

architecture, as illustrated in Figure 5, which combines UNet's hierarchical design with ViT's transformer blocks.

The Swin Transformer achieves its hierarchical structure through patch merging and expansion. On the encoder side, the patch size for each token is progressively increased from 4×4 to 32×32 via patch merging, and then reduced on the decoder side using patch expansion. We also incorporate skip connections between the Swin Transformer blocks in the encoder and decoder, as illustrated in Figure 5, similar to the UNet architecture. Swin Transformer reduces computational complexity by employing a shifted window approach. Instead of allowing each token to attend to every other token, attention is restricted to tokens within a window of size M. In subsequent transformer blocks, the window is shifted to enable communication across different regions.

4. Experiments

In our series of experiments, we evaluated different architectures for image-to-image translation, focusing on their performance across several metrics. The evaluation metrics include RMSE, which measures pixel-level errors. Perceptual loss gauges the visual similarity to real images. The Inception score assesses image quality and diversity, while FID quantifies the distance between the distributions of real and generated images.

Data Augmentation : For the augmentation, horizontal flipping and Gaussian noise was applied to the synthetic images, but only if the flip was not performed. Here, U-Net model outperformed its augmented counterpart. See table 2.

ViT : The ViT-Complex model, with 6 attention heads and 6 encoder blocks, delivered the somewhat improvements. The ViT-Color model, utilizing color-specific augmentations, only improved perceptual quality. Among dif-

Model	RMSE ↓	Perceptual ↓	Inception ↑	FID ↓
Unet	**0.180**	**0.048**	**3.5**	**259.86**
Unet Aug	0.196	0.050	2.73	307.30

Table 2. Performance metrics for Unet

ferent patch sizes, ViT-8 achieved the highest quality. For this see table 3 and 4.

Model	RMSE	Perceptual	Inception	FID
ViT-Complex	**0.151**	0.0388	3.01	**210.33**
ViT-Color	0.169	**0.0320**	2.66	267.12
ViT-Aug	0.165	0.0365	**3.10**	280.90

Table 3. Performance metrics for ViT-based GAN models with a patch size of 16

Model	RMSE	Perceptual	Inception	FID
ViT-8	**0.144**	**0.0316**	**3.19**	**191.85**
ViT-16	0.154	0.0515	2.80	252.30
ViT-32	0.162	0.0412	2.99	283.78

Table 4. Performance metrics for ViT-based GAN models with different patch sizes.

Swin Transformer : With table 5, Swin-(12,6) model excelled with the best RMSE, Inception score, and competitive FID. The Swin-(6,6) model showed slightly higher RMSE and FID but had strong perceptual quality.

Model	RMSE	Perceptual	Inception	FID
Swin-(6,6)	0.224	0.0600	1.79	435.18
Swin-(12,6)	**0.204**	**0.0419**	**3.24**	429.64
Color-(12, 6)	0.236	0.0484	2.10	**416.57**

Table 5. Performance metrics for Swin Transformer-based models with different window sizes.

5. Results and Analysis

Following figure 6 compares image generation results across different models. It shows input synthetic images, ground truth, and generated outputs from U-Net, ViT-8, and Swin. ViT-8, the top-performing model, produces the most accurate and realistic images, highlighting its effectiveness over the U-Net and Swin models.

The augmentation did not significantly improve performance, possibly due to the limited effectiveness of these transformations in enhancing model generalization. The Swin Transformer models, under performed compared to ViT-based models, likely because the Swin architecture struggled with capturing complex image details. This under-performance might be further exacerbated by the limited dataset, which may have hindered the Swin models' ability to learn robust features.

Figure 6. Input synthetic images and their corresponding Ground Truth and Generated images. **Row 1:** Input Image, **Row 2:** Ground Truth **Row 3:** UNet, **Row 4:** ViT and **Row 5:** Swin

6. Future Enhancement

Due to time constraints, the experiments were limited, but there is significant potential for future work. Hyperparameter tuning of Swin Transformers could yield improved results. Additionally, incorporating the Virtual Kitti dataset [2], which includes simulated images under various weather conditions, and exploring techniques like CycleGAN [9] or StyleGAN [5] to generate realistic weather-specific images could further enhance the project.

7. Conclusion

In conclusion using adversarial loss and generator discriminator architecture we are able to formulate image translation pipeline to get realistic image from simulated images. In presence of more amount of data the above mentioned methods are expected to perform better. Comparing we see ViT based generator has comparatively high performance.

References

[1] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *CoRR*, abs/2010.11929, 2020. 3

[2] Adrien Gaidon, Qiao Wang, Yohann Cabon, and Eleonora Vig. Virtualworlds as proxy for multi-object tracking analysis. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4340–4349, 2016. 1, 4

[3] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3354–3361, 2012. 1

[4] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei 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, 2017. 1, 2

[5] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks, 2019. 1, 4

[6] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows, 2021. 3

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

[8] Rohan Taori, Achal Dave, Vaishaal Shankar, Nicholas Carlini, Benjamin Recht, and Ludwig Schmidt. Measuring robustness to natural distribution shifts in image classification. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020. 1

[9] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks, 2020. 1, 4