Received Date:21st November, 2025
Revision Date: 23rd December, 2025
Accepted Date: 28th January, 2026

Building Facade Design through Paired Image-to-Image Translation using Pix2Pix

Riden Shankhadev[1*], Sejal Panta[2], Prerana Shrestha[3], Sudeep Shakya[4]
[1] Dept. of Computer Engineering, Kathmandu Engineering College, Nepal, Email: ridensdev1@gmail.com
[2] Dept. of Computer Engineering, Kathmandu Engineering College, Nepal, Email: sejalpanta@gmail.com
[3] Dept. of Computer Engineering, Kathmandu Engineering College, Nepal, Email: preranastha9999@gmail.com
[4] Head of Department, Dept. of Computer Engineering, Kathmandu Engineering College, Nepal, Email: sudeep.shakya@kecktm.edu.np

Abstract— This project leverages the pix2pix image-to-image translation framework to generate realistic building facades from semantic label maps using the CMP Facade dataset. The primary objective was to reproduce the original conditional GAN architecture and validate its performance on the facade-generation task. Beyond model reproduction, we developed a lightweight web-based interface using a standard JavaScript canvas that allows users to draw block-level layout sketches. These sketches are then processed by the trained model to generate corresponding facade images, providing a simpler and faster method for prototyping architectural facades compared to traditional manual sketching techniques. The interactive workflow preserves the structural and textural fidelity of the generated buildings, while making facade visualization more accessible and user-friendly. Experimental results demonstrate that the reproduced model effectively captures building features, and the interface offers a practical tool for rapid architectural concept development and visualization. Overall, this work presents an application-focused replication of pix2pix that bridges the gap between theoretical model reproduction and practical, interactive design workflows, highlighting its potential for supporting early-stage architectural prototyping and creative experimentation.

Keywords — pix2pix, image-to-image translation, building facade generation, conditional GAN, CMP Facade dataset, interactive prototyping

I. Introduction

Building visualization is one of the crucial design processes which allows architects and clients to see a realistic representation of proposed buildings before the construction begins. The traditional methods of creating detailed 2D visualizations from rough sketches can be time-consuming and heavily depend on manual labor rather than automated technologies. Architects usually depend on software such as AutoCAD, SketchUp and Revit for creating basic 2D and 3D views of their designs. Mastering these tools is not easy as it demands a good amount of time and effort on the part of the learner. Even after achieving mastery, producing high quality visualizations can be laborious work with slow progress. However, with the progress and advancements made in deep learning, particularly in image-to-image translation, it has become possible to automate these processes. Through the use of the Pix2pix framework that leverages Conditional Generative Adversarial Networks (cGANs), it is possible to translate from label images to building images, thereby saving time and reducing the manual effort required.

In a standard GAN, there are two competing networks: the generator and the discriminator. The generator is a network that creates new data records from random noise which aims to produce outputs that are similar to real data. The discriminator on the other hand evaluates these outputs along with real data from the training set, learning to distinguish between genuine and synthetic data. Both the generator and discriminator utilize CNNs (Convolutional Neural Networks), with the generator typically using a U-Net architecture and the discriminator using Patch GAN. Conditional Generative Adversarial Network (cGAN) is a sophisticated version of Generative Adversarial Networks (GANs) that is designed to enable more controlled and precise data generation. It was introduced by Mehdi Mirza and Simon Osindero in 2014 with the aim of enhancing the basic GAN architecture by incorporating additional information into the generation and evaluation processes. This additional information is known as the conditioning variable which can be anything relevant to the task at hand, such as class labels, images, or other structured data.

* *Corresponding Author*

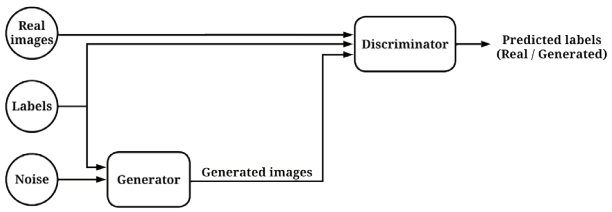

Fig. 1 Conditional GAN

By leveraging the conditional GAN–based architecture introduced in the pix2pix framework by Isola et al. [1], the model can perform precise and contextually accurate image-to-image translations, making it a powerful tool for applications where the relationship between input and output data is critical. This extension of GANs into the image-to-image translation showcases the versatility and potential of adversarial learning in generating high-quality, contextually relevant images. The applications can be found in areas such as image-to-image translation, text-to-image synthesis, super-resolution, etc.

This project aims to apply Pix2Pix using cGAN to produce effective and efficient building visualizations. To achieve this goal, simple color-coded sketches will be generated by the user using an interface facilitating the drag-and-drop method for placing necessary blocks like doors, windows, etc., thereby outlining a preliminary structure of the building. In this interface, architectural elements such as beams, doors, and windows will be represented by specific colors, aiding in the creation of comprehensive color-coded diagrams for input into the model. This approach is demonstrated in the provided figure 2, where simple color blocks of building facades, depicted with distinct color codes, are transformed into 2D visualizations. Hence, by simply drawing rectangles with the colors representing windows, doors, walls, and other parts of a building in the canvas, people can quickly see what a building might look like. This makes the design process faster, more accurate, and easier to visualize for both

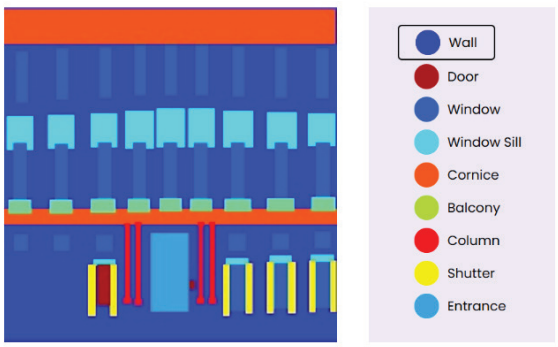

Fig. 2 Input Labeled Image

II. Literature review

Numerous studies have explored the application of conditional GANs (cGANs), particularly Pix2Pix, for image-to-image translation tasks. The foundational work by Isola et al. [1] established cGANs as a versatile solution capable of learning not only the mapping from input to output images but also the loss function to train the mapping. Their method proved effective in tasks such as turning edge maps into realistic objects, semantic labels into street scenes, and grayscale images into colored ones. Building on this, several frameworks have focused on enhancing translation quality and diversity. For instance, the domain-adversarial method introduced in [2] incorporated supervised and unsupervised learning with a feature comparator and used metrics like SSIM and PSNR for evaluation. Khati et al. [3] developed a Pix2Pix-based facade generation system where users could sketch or upload label maps, and the model produced detailed photorealistic building facades using adversarial and L1 losses.

More advanced variations have also emerged. Vit-GAN [4] introduced a hybrid model combining cGANs with vision transformers and a Markovian discriminator, outperforming traditional architectures like U-Net. Similarly, Luo et al. [5] used Pix2Pix with DCNNs to generate building designs based on local Hong Kong styles, applying extensive preprocessing and semantic segmentation for architectural detail. Another study [6] employed a pix2pixHD-based model for high-resolution facade synthesis, integrating user-provided style references using the DPST framework.

Pix2Pix's flexibility has also been shown in a variety of paired image translation tasks, such as maps to satellite images [7], architecture labels to facades [8], and even low-rise residential facade generation where AttU-Net outperformed the original U-Net [8]. Comparative studies [9] also highlighted Pix2Pix's strength in paired datasets, whereas CycleGAN is better suited for unpaired scenarios. Several other works [10][11] have reiterated the usefulness of Pix2Pix in architecture-related applications, emphasizing its potential for improving both design automation and education through realistic image synthesis. Overall, these studies collectively support the strong applicability and adaptability of cGAN-based models, especially Pix2Pix, in the image translation domain.

III. Methodology

A. Dataset

The dataset used in this project is the CMP Facades collection created by Tyleček and Šára at the Center for Machine Perception (CMP), Czech Technical University in Prague [12]. The dataset consists of two parts: the base dataset containing 378 images, and the extended dataset containing 228 images. Combined, these form a total of 606 paired images of building facades, where each pair includes a labeled input image and its corresponding facade output image. This dataset is also used in the pix2pix 'facades' example for image-to-image translation.

1) Input Labeled Image:

The labeled image of resolution 256*256 consists of a segmentation map that represents various architectural components, such as walls, windows, doors, balconies, and other structural elements. These components are color-coded to help the model understand the spatial arrangements and relationships between the features. The color coding used in the segmentation map, as shown in figure 2, is as follows:
Dark Blue: Background/Wall
Dark Red: Door
Bright Royal Blue: Window
Cyan: Window Sill
Orange: Cornice
Light Green: Balcony
Red: Column
Yellow: Shutter
Sky blue: Entrance

2) Input Facade Image:

The input facade image of resolution 256*256 corresponds to a real-world facade of a building that the model aims to generate from the input labeled segmentation map. Figure 3 is a sample of input facade images.

Fig. 3 Input Facade Image

3) Paired Image:

The combination of labeled image and corresponding output is the paired image which is of 256*512 resolution and is used to train the Pix2Pix model, enabling it to learn the mapping between abstract architectural layouts and realistic building facades.

Fig. 4 Paired Image

B. System Block Diagram

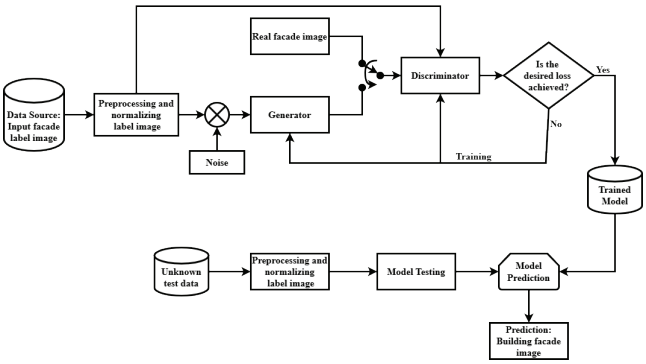

Fig. 5 System Block Diagram

This system block diagram represents the use of Conditional Generative Adversarial Network (cGAN) in the Pix2Pix framework for generating building facade designs. During the training phase, labeled facade images are preprocessed and normalized before being input into the generator, which produces facade images. Noise is added to improve robustness, and the discriminator evaluates these generated images alongside real facade images to distinguish between real and fake. The generator and discriminator are trained adversarially until the desired loss (0.69) is achieved, producing a trained model. Then, in the testing phase, unseen labeled facade images are preprocessed and passed through the trained model, which generates realistic building facade designs as predictions. The system ensures the output is conditionally aligned with the input labels, making it suitable for architectural visualization and design automation.

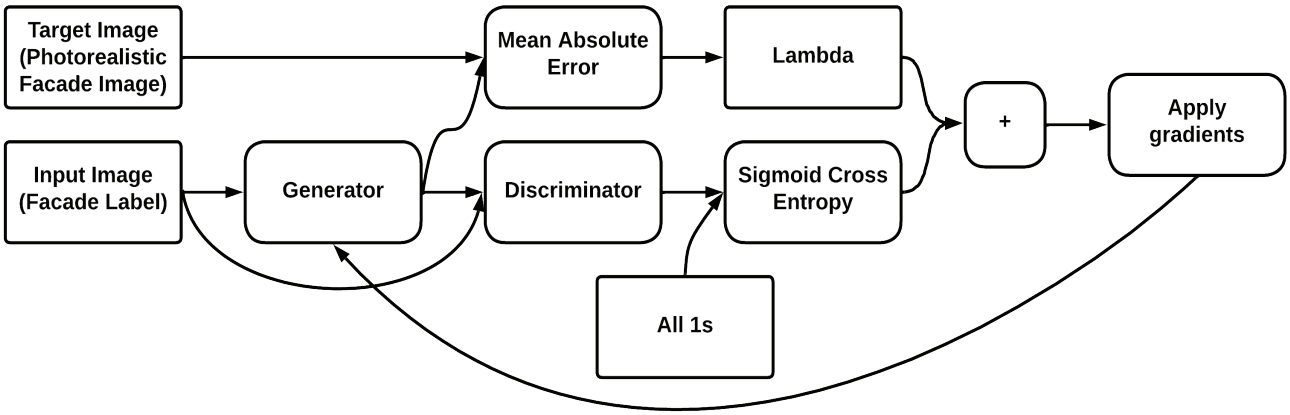

Fig. 6 Training Process of Generator

During training, the generator takes a labeled facade image as input and generates a realistic design. The output is compared with the target image using L1 loss (Mean Absolute Error) to measure pixel-level differences. Additionally, binary cross-entropy loss is calculated as the generator tries to fool the discriminator into classifying fake images as real (label "1"). The total generator loss combines both losses, weighted by a factor (λ=100), and gradients update the generator to improve its outputs.

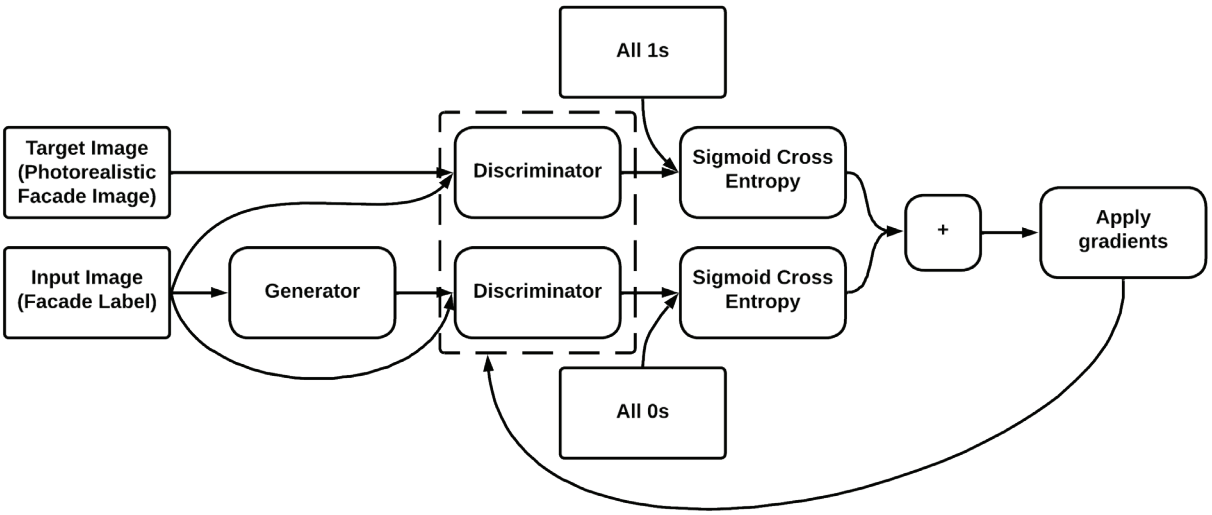

Fig. 7 Training Process of Discriminator

For the discriminator, both real and generated images are evaluated. It is trained to output "1" for real and "0" for fake images. Binary cross-entropy losses for both are calculated and combined to update the discriminator's parameters. As the discriminator improves at spotting fake images, the generator is pushed to create more realistic outputs.

C. Dataset Preparation

Step 1: Collect paired training data consisting of input images and their corresponding target images.

Step 2: Resize each 256 x 256 image to a larger height and width of 286 x 286.

Step 3: Randomly crop back to 256 x 256.

Step 4: Randomly flip the image horizontally (random mirroring).

Step 5: Normalize the images to the [-1, 1] range.

D. U-net Generator Network

U-Net is a convolutional neural network architecture originally introduced for biomedical image segmentation, but it has proven effective for various semantic segmentation tasks. The architecture gets its name from its symmetric, U-shaped structure, consisting of two main parts: a contracting path (encoder) and an expanding path (decoder), connected via skip connections that help preserve spatial and contextual information throughout the network.

1) Input Layer:

The starting point of the network, where an image of size (256, 256, 3) is received as input.

2) Sequential Blocks (Downsampling - Encoder):

A series of convolutional layers reduce the spatial dimensions while increasing the feature depth. Batch Normalization stabilizes training and helps prevent model collapse. LeakyReLU Activation is used to maintain gradient flow. The transformations include:

(256, 256, 3) → (128, 128, 64) → (64, 64, 128) → (32, 32, 256) → (16, 16, 512) → (8, 8, 512) → (4, 4, 512) → (2, 2, 512) → (1, 1, 512)

3) Sequential Blocks (Upsampling - Decoder):

Transposed Convolutional layers progressively restore spatial resolution. Skip Connections merge the corresponding downsampled feature maps with unsampled ones. Batch Normalization ensures stable gradient propagation. Dropout is applied in deeper layers to prevent overfitting.

(1, 1, 512) → (2, 2, 1024) → (4, 4, 1024) → (8, 8, 1024) → (16, 16, 1024) → (32, 32, 512) → (64, 64, 256) → (128, 128, 128)

4) Concatenate:

Feature maps from the down sampling blocks are concatenated with the corresponding upsampling blocks to retain spatial details and enhance feature propagation.

5) Final Conv2D Transpose Layer:

This layer converts maps into an output image of shape (256, 256, 3). It uses Tanh Activation to normalize pixel values between [-1,1], ensuring compatibility with image data processing.

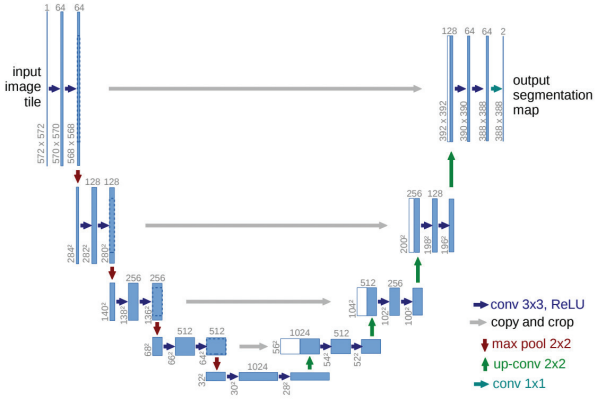

Fig. 8 U-net Architecture (adapted from [13])

E. Discriminator Network

The architecture of the discriminator is as follows:

1) Input Layer:

Two input layers receive images of shape (256, 256, 3) each, representing real and generated images for comparison.

2) Concatenate:

Merges the two input tensors, resulting in a combined shape of (256, 256, 6), facilitating feature comparison.

3) Sequential Blocks:

Three sequential layers progressively reduce spatial dimensions and increase feature channels as follows:

(256, 256, 6) → (128, 128, 64)

(128, 128, 64) → (64, 64, 128)

(64, 64, 128) → (32, 32, 256)

4) ZeroPadding2D:

Adds padding to expand the spatial dimensions from (32, 32, 256) to (34, 34, 256) and later from (31, 31, 512) to (33, 33, 512) for maintaining feature alignment before convolution operations.

5) Conv2D Layers:

Extracts deep features through convolution, progressively reducing spatial size. The transformations include:

(34, 34, 256) → (31, 31, 512)

(33, 33, 512) → (30, 30, 1) as the final output layer for discrimination.

6) Batch Normalization:

Normalizes feature maps to stabilize training and accelerate convergence.

7) LeakyReLU Activation:

Introduces non-linearity and prevents dead neurons by allowing small negative values, aiding in better gradient flow.

F. Generator Loss

1) Adversarial Loss:

It encourages the generator to produce realistic images. It is calculated using binary cross-entropy between the discriminator's output for generated images and the target label.

$L_{Adverserial} = - E_x [\log D(x,G(x))]$

where,

D(x,G(x)): Discriminator's output, which is the probability that the generated image G(x) is real.

E_x: Expected value over all input images x

2) L1 Loss:

It encourages the generator to produce images that are close to the ground truth. It is calculated as the L1 distance between the generated image and the target image.

$L_{L1}(G) = E_{x,y}[\|y - G(x)\|1] = 1/n \sum n |y_i - \hat{y}_i|$

where,

x: The input image

y: The real/target image

$E_{x,y}$: The expected value over the dataset of input-output image pairs (x,y)

G(x): The image generated by the generator when given input x

$\|y - G(x)\|1$: The L1 norm, or sum of absolute pixel-wise differences between the real image and the generated image

n: The total number of pixels in the image

y_i: The true pixel value at position i in the target image

$\hat{y}_i = G(x)_i$: The predicted/generated pixel value at position i

3) Total generator loss:

It is the combination of the L1 loss and the adversarial loss, controlled by the hyperparameter λ.

L_G = Adversarial Loss + λ × L1 Loss

where,

λ is the weighting factor for the L1 loss.

G. Discriminator Loss

It is calculated as the binary cross-entropy loss for classifying real images as real and generated images as fake. It indicates how well the discriminator is performing in distinguishing real from fake images.

$L_D = - E_{x,y} [\log D(x,y)] - E_x [\log(1 - D(x,G(x)))]$

where,

x: Input image

y: Ground truth target image (e.g., real building image)

G(x): Generated (fake) image from the generator

D(x,y): Discriminator output (probability that (x, y) is real)

D(x,G(x)): Discriminator output for a fake pair.

IV. Verification and Validation

In the pix2pix GAN, traditional evaluation metrics like accuracy or F1-score, commonly used in CNNs and other single-network architectures, are not directly applicable. This is because pix2pix consists of two adversarially trained networks (generator and discriminator), which require loss-based evaluation. The performance of the system is validated by monitoring multiple loss components during training.

For this, it needs to be ensured that neither the generator nor the discriminator dominates the training process hence, a reference point of log(2)≈ (0.69) is used as an indicator of equilibrium. This value originates from the binary cross-entropy (BCE) loss function:

$L = - [y \log (p) + (1 - y) \log (1 - p)]$

where,

y = True label, 1 for real data and 0 for fake data

p = Predicted probability which is the output of the discriminator (after sigmoid)

When the discriminator is perfectly uncertain (i.e., p=0.5), the expected loss becomes:

$L = - \log (0.5) = \log (2) \approx 0.693$

At this point, the discriminator cannot distinguish between real and generated images, which indicates that the generator has reached a level of realism where adversarial training is balanced. Hence, if the losses fluctuate around this value during training, the model is considered to have achieved stable convergence.

The losses trends for Pix2Pix CGAN during training at λ=100 are illustrated below where the y-axis represents the discriminator loss and the x-axis shows the training duration spanning up to 3.8 hours indirectly reflecting the 50,000 training steps. The graph displays up to step 49, which when scaled by a factor of 1000 (49 × 1,000), corresponds to the full training run.

A. Discriminator loss:

The discriminator loss exhibits fluctuations but shows an overall downward trend, indicating that the discriminator effectively learns to distinguish between real and generated

images. The balance between the two networks is maintained, suggesting that the discriminator does not overfit to real data and that healthy adversarial training is achieved.

Fig. 9 Discriminator Loss Graph

B. Generator L1 Loss:

The L1 loss remains relatively stable with slight fluctuations indicating that the generated images are progressively becoming more similar to the ground truth in terms of pixel-wise accuracy. The stable trend implies that the generator is not overly focused on fooling the discriminator at the cost of losing structural similarity.

Fig. 10 Generator L1 Loss Graph

C. Generator Adversarial Loss:

The generator's adversarial loss remains relatively high, indicating that the generator is continuously being challenged by the discriminator. There is a gradual decline with fluctuations, showing that the generator is improving in producing realistic images but still faces difficulty in perfectly fooling the discriminator.This pattern suggests a healthy competition between the two networks.

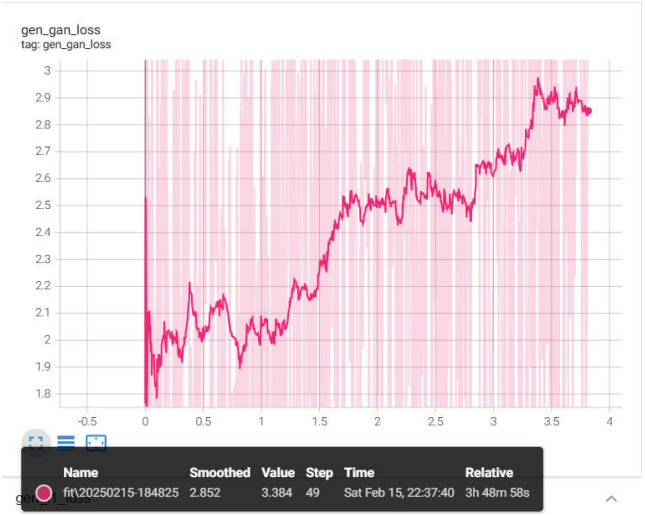

Fig. 11 Generator Adversarial Loss Graph

D. Total generator loss:

The total loss combines adversarial and L1 loss, maintaining a steady trend with a slight downward movement. This indicates that the model is effectively balancing between generating realistic images and ensuring pixel-wise accuracy. A steady decline implies successful convergence and that the hyperparameter λ=100 is appropriately balancing the loss components.

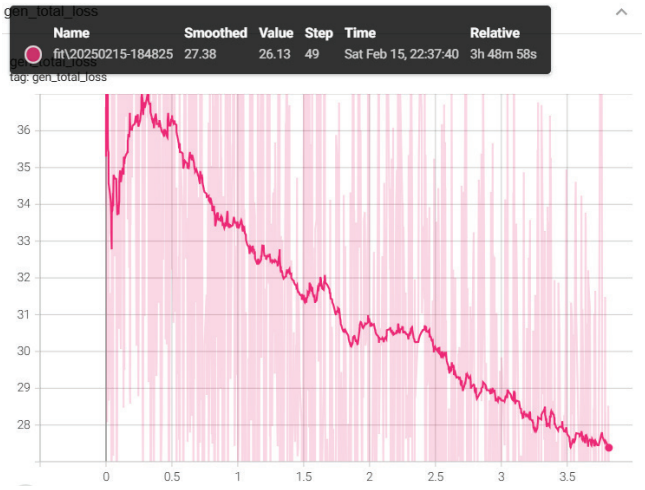

Fig. 12 Generator Total Loss Graph

E. Conclusion for Model Verification and Validation

The observed balance between discriminator and generator losses indicates that the model is performing well. The discriminator is not overpowering the generator, and the generator is steadily improving. The consistent trends across the different loss components show that the model is converging without major instability, suggesting good generalization. This balance confirms that the model is well-tuned and ready for further evaluation or deployment.

V. Results

The trained Pix2Pix model was evaluated on its ability to generate realistic building facades from label images. After training for 50,000 steps, the discriminator loss stabilized at 0.22, ensuring it correctly distinguishes real and generated images without overfitting, while the generator's adversarial loss remained high but showed improvement, reflecting ongoing challenges in fooling the discriminator. The L1 loss stayed stable around 0.23, confirming structural similarity between generated and real images, and the total generator loss settled at 26.13, demonstrating a well-balanced trade-off between realism and pixel accuracy. The model achieved an average PSNR of 12.50 and an average SSIM of 0.23 on the test set, indicating that the generated facades capture general structural patterns but have lower pixel-wise similarity compared to the ground truth. Figure 13 shows example results, where input semantic maps are translated into realistic facades. While the pixel-level metrics are relatively low, the visual fidelity of the generated images is consistent with the original facades, and PSNR/SSIM may not fully reflect the perceptual quality of GAN-generated images, as these metrics emphasize pixel-wise similarity over visual realism, yet the generated facades remain visually plausible.

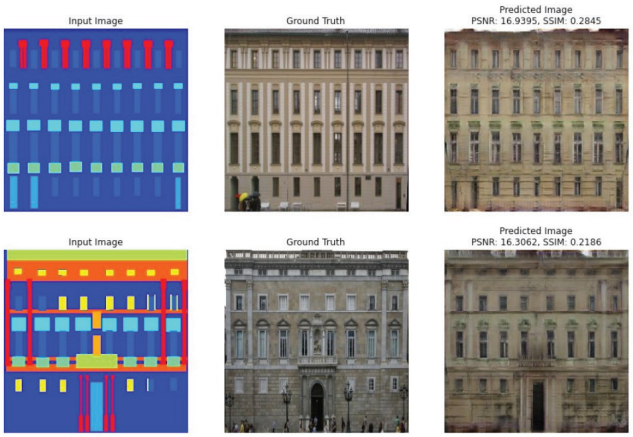

Fig. 13 Output of the model on test dataset along with PSNR and SSIM values

VI. Conclusion

In this project, we demonstrated that a Pix2Pix GAN can effectively generate realistic building facades from semantic labels. The model trained stably, maintaining a balance between generator and discriminator, and produced facades that preserve structural patterns and visual fidelity, even though pixel-level metrics such as PSNR and SSIM are limited in capturing perceptual quality. These findings highlight the potential of GANs for architectural image generation, opening opportunities for AI-assisted design refinement and automated facade reconstruction.

VII. Future Enhancements

The developed system can be improved in the future by implementing the following:

- Combination with depth estimation for 3D facade synthesis.
- Experimentation can be done with higher resolution images and Pix2PixHD for higher-resolution outputs.

Acknowledgement

We would like to express our sincere gratitude to Assoc. Professor Er. Sudeep Shakya, Head of the Department of Computer Engineering, Kathmandu Engineering College, for his invaluable guidance, encouragement, and support throughout this research. We are also deeply thankful to our project and year coordinator, Er. Sharad Chandra Joshi, whose expert advice, constructive feedback, and continuous guidance significantly contributed to the successful completion of this work. We further extend our appreciation to the Department of Computer Engineering for providing the necessary resources, facilities, and academic support that made this research possible.

References

[1] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-Image Translation with Conditional Adversarial Networks," Berkeley AI Research (BAIR) Laboratory, 2018.

[2] Z.Li, W.Wang, and Y.Zhao, "Image Translation by Domain-Adversarial Training," Computational Intelligence and Neuroscience, no.1, p. 8974638, 2018.

[3] A. Khati, A. Misri, M. Kotak, and D. H. Chavan, "Generating Building Infrastructure Modeling Using cGAN," in Proceedings of the 4th International Conference on Advances in Science & Technology (ICAST2021), May 2021.

[4] Y. Gündüç, "Vit-GAN: Image-to-image Translation with Vision Transformers and Conditional GANs," arXiv preprint, Oct. 2021. [Online]. Available: https://doi.org/10.48550/arXiv.2110.09305.

[5] J. Luo, B. Yu, H. Peng, Y. Shi, Y. Li, and A. Fingrut, "Deep Generative Modeling Tasks: Automatic Generation of Building Facades with Pix2Pix GAN for Hong Kong City Expansion and Renovation," in Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics (SMC), 2023.

[6] J. Zhang, T. Fukuda, N. Yabuki, and Y. Li, "Synthesizing Style-Similar Residential Facade from Semantic Labeling According to the User-Provided Example," in Proceedings of the 28th International Conference of the Association for Computer-Aided Architectural Design Research in Asia (CAADRIA), 2023.

[7] J. Henry, T. Natalie, and D. Madsen, "Pix2Pix GAN for Image-to-Image Translation," ResearchGate Publication, 2021, pp. 1-5.

[8] D. Wan, R. Zhao, S. Zhang, H. Liu, L. Guo, P. Li, and L. Ding, "A Deep Learning-Based Approach to Generating Comprehensive Building Facades for Low-Rise Housing,", vol. 15, no. 3, p. 1816, 2023. [Online]. Available: https://doi.org/10.3390/su15031816.

[9] R. K. Senapati, R. Satvika, A. Anmandla, G. A. Reddy, and C. A. Kumar, "Image-to-image translation using Pix2Pix GAN and cycle GAN," in International Conference on Data Intelligence and Cognitive Informatics, Singapore, 2023, pp. 573-586.

[10] Z. Li, B. Guan, Y. Wei, Y. Zhou, J. Zhang, and J. Xu, "Mapping New Realities: Ground Truth Image Creation with Pix2Pix Image-to-Image Translation," arXiv preprint, Apr. 2024. [Online]. Available: https://arxiv.org/abs/2404.19265.

[11] Q. Yu, J. Malaeb, and W. Ma, "Architectural Facade Recognition and Generation Through Generative Adversarial Networks," in 2020 International Conference on Big Data & Artificial Intelligence & Software Engineering (ICBASE), IEEE, 2020, pp. 310-316.

[12] R. Tyleček and R. Šára, "CMP Facade Database," Center for Machine Perception (CMP), Czech Technical University in Prague, Prague, Czech Republic, 2015. [Online]. Available: https://cmp.felk.cvut.cz/~tylecr1/facade/. [Accessed: Nov. 21, 2025].

[13] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," in Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015, vol. 9351, O. Navab, J. Hornegger, W. Wells, and A. Frangi, Eds. Cham, Switzerland: Springer, 2015, pp. 234–241. [Online]. Available: https://doi.org/10.1007/978-3-319-24574-4_28.