GANs
EE608: Final Report

Aryan Garg
Computer Science & Engineering, B.Tech.
Indian Institute of Technology, Mandi
b19153@students.iitmandi.ac.in

I. Introduction (Problem Statement)

With the introduction of Generative Adversarial Networks (GANs) [5] in 2014 by Ian Goodfellow *et al*., unsupervised and semi-supervised learning tasks have seen tremendous results in terms of fidelity, diversity, and coherence.

GANs are deep learning architectures with theoretical foundations rooted in game theory. The ultimate goal of two competing networks (generator & discriminator) is to reach the Nash equilibrium [18]. However, achieving this equilibrium in practice is not feasible [4, 7, 17]. The generative research community has thus seen improvements through an ensemble of architecture and loss variants [10, 12]

An emphasis on the deep learning aspect is prevalent in the generative literature [8]. Here, we present a comprehensive study of GANs including small experiments where we combine different low-level image processing algorithms in the pre-processing pipeline, for some well-known GAN architectures.

We replicate nine architectures from scratch and assess their results using metrics [7, 20] that correlate well with the human visual system as well. We also perform latent space exploration and other experiments for some of the GANs.

II. Proposed Method

Since GANs proved their utility and constant improvements have been proposed since 2014 till date, the careful choice of paper replication is an explicit reflection of the same progression of improvements.

Starting from the original idea of **GAN** [5] in 2014, we implement loss based improvements that surfaced shortly after: *Wasserstein*-GAN or **WGAN** [1] and WGAN-Gradient-Penalty or **WGAN-GP** [6]. Then follows the historic Deep-Convolutional GAN or **DCGAN** [19] that extracts the power of convolutional layers for GANs. Following that we implement Super-Resolution GAN or **SRGAN** [16] as well. Then we take a step back and look at some non-standard yet revolutionary ideas and implement **infoGAN** [2] for an information-theoretic perspective that also leads to a satisfactory level of disentanglement of features. Then comes **realness-GAN** [28] with their famous anchors and Kullback-Leibler Divergence contributions to the vanilla GAN model. We also implemented these conditional-GANs: **pix2pix** [13] that takes you from a semantic map to a photo-realistic output, **cycleGAN** [33] that uses two generators and two discriminators to enforce a cyclic consistency between two domains. We use the Monet-Art Dataset for Style-Transfer (Kaggle). Lastly, we implement some revolutionary ideas that deviates far from the original GANs within the DCGAN architecture (fused) but also produce some of the finest quality results: **styleGAN** [14].

All our methods are trained on a wide variety of datasets including the likes of MNIST [15], FashionMNIST [29], Animal Faces-HQ or AFHQ [3], Facades [24], a cherry-picked dataset for Super-resolution from Kaggle (hosted at the repository as well), Monet-Art, etc.

The models are evaluated on one or more of the following evaluation metrics:

- **Peak-Signal-to-Noise Ratio** (PSNR) and implicitly **Mean Squared Error** (MSE). However, MSE is never a good choice for evaluating perceptual results as it does not correlate well with the human-visual system [25].
- **Structural Similarity Index** (SSIM) [32] and **Multi-Scale-SSIM** [26]
- **Inception Score** from the InceptionV3 [23, 20] network. [20]
- **Fréchet Inception Distance** (FID) [7]. Note that mainly FID was used during most experiments.
- **Memorization-informed Fréchet Inception Distance** or Mi-FID
- **Learned Perceptual Image Patch Similarity** (LPIPS) or Perceptual Loss or the VGG-Loss [21, 31]

After 9 successful paper replications, we go a step ahead and use simple image processing techniques to train these models for experimental evidence that these augmentation pipelines improve metrics and generalizing power of the model. We also hypothesize that augmentations help in stabilizing training to an extent.

Denoising Diffusion Probabilistic Model [9] was chosen due to the growing popularity of diffusion models and their strengths over traditional GANs. However, it was not implemented due to time constraints.

Image Processing Techniques:

The pre-processing pipeline for all GANs will be an ensemble of image distortions. Image distortions, used in conventional image processing include mean contrast stretching, luminance shifting, gaussian noise contamination, impulsive noise contamination, JPEG compression, blurring, spatial scaling (zooming), spatial shifting and rotation, or simply affine

transformations. However, for our comparisons sake, we use the following three channel-dimension agnostic transformations:

1) **Random-Affine Transform**: The shift limit, scale limit, and rotate limits are model specific to preserve the dataset features yet transform each sample sufficiently well producing numerous permutations.
2) **Random Horizontal Flip** with a probability of 0.5
3) **Gaussian Noise** with an always odd and positive kernel size varying from 3 to 7 based on the dataset. The standard deviation (σ) is the default parameter between 0.1 and 2.0.

III. Results and Comparisons

All code is available on our GitHub[1].

A. GANs

*1) **The Original GAN**:* Instead of the traditional MNIST-vanilla GAN toy set up, we take things a bit further by replicating vanilla GAN from scratch and training it on the 10-class Fashion-MNIST [29] for 5 epochs for all experiments.

We perform 4 image-preprocessing experiments in total after replication and quantitatively compare the results using Frechet Inception Distance [7]. See Table I.

Augmentation(s)	Val G-Loss	Val D-Loss	FID
None	0.799	**0.661**	**287.37**
Blur	**0.679**	0.680	306.86
Blur+Flip	0.695	0.669	318.66
Blur+Flip+Affine	0.738	0.695	460.43

TABLE I: Quantitative Comparison: Validation Losses & FID Scores of Augmentation Experiments. Here, Blur $\longrightarrow$ adding Gaussian Noise, Flip $\longrightarrow$ Random Horizontal Flip with 50% probability and Affine $\longrightarrow$ Random Affine Transformation. See M1 GAN.ipynb on GitHub[1] for explicit hyper-parameter settings of these augmentations.

The perceptual generation output 1 suggests that the **blur+flip experiment** gives most coherent or semantic detail.

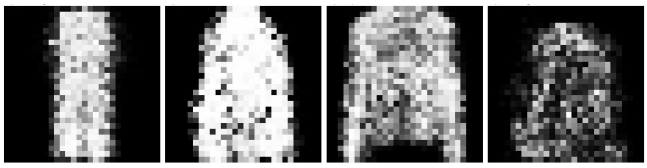


Fig. 1: Qualitative Comparison: Randomly Sampled Synthesized Outputs from different experiments. From Left to Right: No Augmentations experiment (pants), Blur experiment (Jacket), Blur+Flip (Jacket), Blur+Flip+Affine (Bag). (See naming convention of transforms from I)

[01] EE-608 Project: Github

*2) **Wasserstein-GAN or WGAN** & WGAN-Gradient Penalty:* For WGAN [1], we set weight-clipping at 0.01 for all image augmentation experiments. And for WGAN-GP [6], lambda or gradient penalty multiplier is set at 10. Obviously, weight-clipping is turned off for WGAN-GP. We train both models for 5 epochs each on the fashion-MNIST dataset to compare not only between the two WGAN variants but also with the original GAN. *Note that WGAN uses RMSProp instead of the traditional Adam Optimizer.*

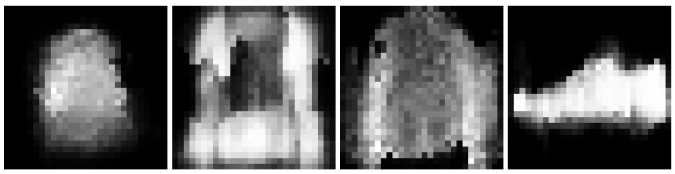
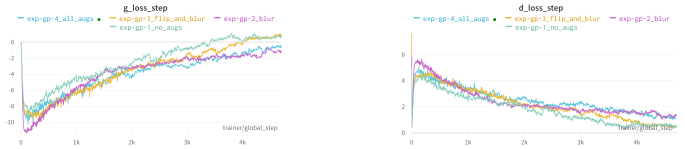


Fig. 2: WGAN Synthesized Ouput. Left to Right: Flip & Blur (Bag), only blur (Jacket/Shirt), no augmentations(Jacket) and all augmentations(Shoe). See table II for a quantitative analysis.

Augmentation(s)	G-Loss	D-Loss
None	0.057	**-0.34**
Blur	0.027	-0.31
Blur+Flip	**0.021**	-0.31
Blur+Flip+Affine	0.028	-0.29

TABLE II: Training Progression of Augmentation Experiments for WGAN (epoch wise). Note that these metrics are running-average smoothed. See Fig **??** Here, Blur $\longrightarrow$ adding Gaussian Noise, Flip $\longrightarrow$ Random Horizontal Flip with 50% probability and Affine $\longrightarrow$ Random Affine Transformation. See M2 and M3 WGAN+GP.ipynb on GitHub[1]. See Fig 2 for qualitative comparison.

After adding gradient penalty, the following loss curves were observed (Fig 3). Perceptual outputs were incoherent for 5 epochs hence not shown as they can not be compared to epoch 5 results of other models.



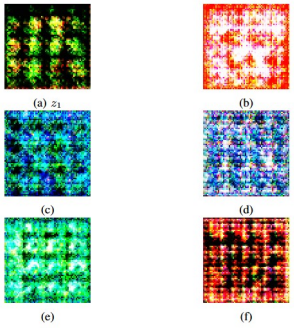
(a) WGAN step G-Loss (b) WGAN step D-Loss

Fig. 3: Training Progression of WGAN-GP (step-wise progression). See Table **??**

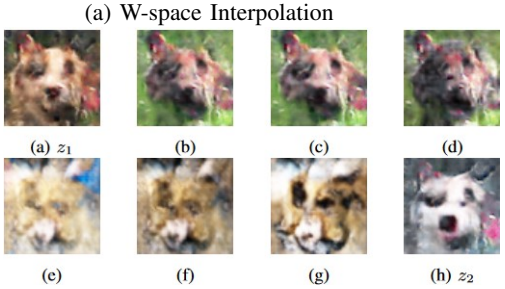
*3) **DCGAN + Style-GAN**:* This fusion model was a part of an Assignment-based selection at the VAL Lab, IISc, Bangalore. The tasks involved are:

- Implementing DCGAN and recording compute statistics over the Dog Subset of AFHQ dataset [3]. See Fig 5

- Hyper-parameter tuning where three hyper-params were experimented/searched for:
  1) Learning Rate
  2) Schedulers (Step, Exponential and Cosine-Annealing)
  3) Optimizers (RMSProp, SGD, Nesterov-SGD)
- Fusing StyleGAN improvements to DCGAN: Adding the Adaptive-Instance Normalization [11] layers, adding a mapping network that takes $Z \longrightarrow W$, and finally, starting from a fixed latent vector. See Fig 6.
- **Disentanglement:** Latent Space Interpolation. Performed for both: Z & W spaces. See 4.
- Bonus experiment: Spectral Normalized Discriminator. See Fig 7.



(a) W-space Interpolation



(b) Z-space Interpolation

Fig. 4: Interpolating the latent spaces of the fused network (Mapping + AdaIN-DCGAN). We see that W-space is sparse which means disentanglement was successful.

See GitHub for code & an 8+ page comprehensive report[1].

4) ***RealnessGAN****:* We replicate RealnessGAN and train it on MNIST for numerous epochs only in vain, to get noise. Nonetheless, we do report the training losses(Fig 8) and present the incoherent generation(Fig 9).

We chalk up the instability during training to the overly convoluted distribution distance measuring techniques on simple datasets like MNIST. We hypothesize that realnessGAN, afterall, is **not** a very general framework.

5) ***infoGAN****:* We train infoGAN [2] on the MNIST dataset for 100 epochs similar to how the authors have done it. For training losses see Fig 10 and generated output, see Fig 11

6) ***SRGAN****:* We perform the Super-Resolution task by replicating SRGAN [16] on a hand-crafted dataset on Kaggle. Here are the results (Fig 12, 13)

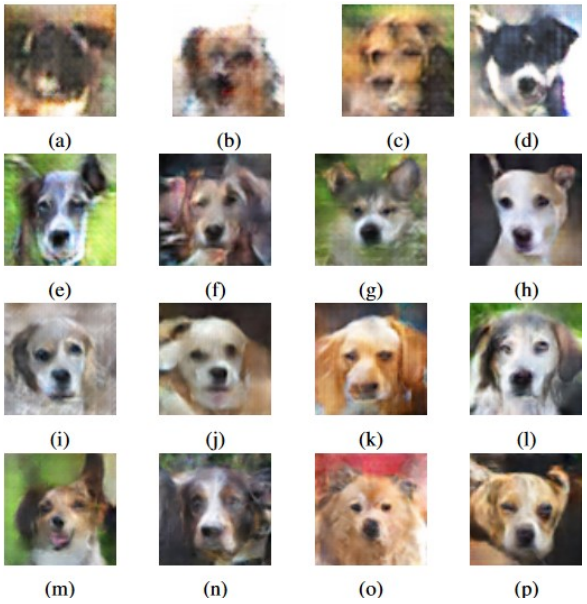


Fig. 5: Validation Pipeline Generated Dogs from the vanilla DCGAN Generator. (extracted from other report hosted on GitHub)

However, it was extremely hard to train SRGAN and it would more often than not collapse into a noisy incoherent output.

7) ***cycleGAN****:* This model was implemented as part of a competition on Kaggle: I'm Something of A Painter Myself, where monet styled art and realistic photos were the two domains. Overall, our notebook submission won the **bronze medal** and we stood 80th overall in the competition.

Here are some generated images in both domains. See Monet-Art Domain: Fig 15 and Realistic Image Domain: Fig 14:

8) ***pix2pix****:* We only replicate a basic pix2pix model on the Facades dataset [24] and present the training progression/generation results using the segmentation map as the conditional input (no experimentations due to time constraints). See Fig 16 and Fig 17.

IV. Conclusion

A. Image Augmentation Experiments

We see that Flip+Blur resulted in the highest fidelity outputs (qualitatively).

B. Learnings from Training

GANs are very sensitive and training them requires a careful searching of hyper-parameters first. Otherwise, one of the two dominates the other and results in incoherent noisy outputs from the generator.

C. Future Work

Delving into diffusion models [22, 30, 27] should be the logical next step as compute power is increasing according to Moore's law and they overcome GAN's shortcomings significantly.

Fig. 6: Validation Pipeline Generated Dogs from the fused DCGAN+StyleGAN Generator

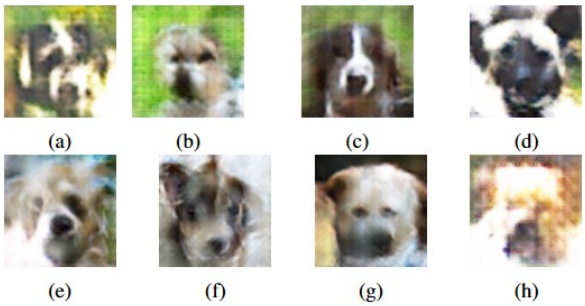


Fig. 7: Spectral Normalized Discriminator with Vanilla DCGAN Generator's validation pipeline synthesis.

(a) Generator Loss (b) Discriminator Loss

Fig. 8: All loss curves of realnessGAN training on MNIST.

(a) Noise (1/2) (b) Noise (2/2)

Fig. 9: Noise Generated Output after 40 epochs

References

[1] Martin Arjovsky, Soumith Chintala, and Léon Bottou. *Wasserstein GAN*. 2017. arXiv: 1701.07875 [stat.ML].

[2] Xi Chen et al. *InfoGAN: Interpretable Representation Learning by Information Maximizing Generative Adversarial Nets*. 2016. arXiv: 1606.03657 [cs.LG].

[3] Yunjey Choi et al. "StarGAN v2: Diverse Image Synthesis for Multiple Domains". In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2020.

[4] Farzan Farnia and Asuman Ozdaglar. *GANs May Have No Nash Equilibria*. 2020. arXiv: 2002.09124 [cs.LG].

[5] Ian Goodfellow et al. "Generative Adversarial Nets". In: *Advances in Neural Information Processing Systems*. Ed. by Z. Ghahramani et al. Vol. 27. Curran Associates, Inc., 2014. URL: https://proceedings.neurips.cc/paper_files/paper/2014/file/5ca3e9b122f61f8f06494c97b1afccf3-Paper.pdf.

[6] Ishaan Gulrajani et al. *Improved Training of Wasserstein GANs*. 2017. arXiv: 1704.00028 [cs.LG].

[7] Martin Heusel et al. *GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium*. 2018. arXiv: 1706.08500 [cs.LG].

[8] Saifuddin Hitawala. *Comparative Study on Generative Adversarial Networks*. 2018. arXiv: 1801.04271 [cs.LG].

[9] Jonathan Ho, Ajay Jain, and Pieter Abbeel. *Denoising Diffusion Probabilistic Models*. 2020. arXiv: 2006.11239 [cs.LG].

[10] Yongjun Hong et al. "How Generative Adversarial Networks and Their Variants Work: An Overview". In: *ACM Comput. Surv.* 52.1 (Feb. 2019). ISSN: 0360-0300. DOI: 10.1145/3301282. URL: https://doi.org/10.1145/3301282.

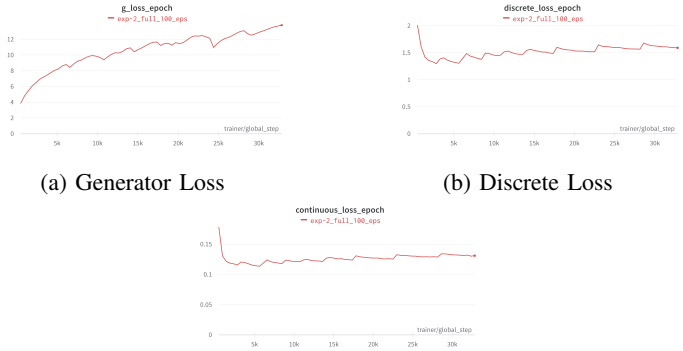
[11] Xun Huang and Serge Belongie. *Arbitrary Style Transfer in Real-time with Adaptive Instance Normalization*. 2017. arXiv: 1703.06868 [cs.CV].

[12] Guillermo Iglesias, Edgar Talavera, and Alberto Díaz-Álvarez. *A survey on GANs for computer vision: Recent research, analysis and taxonomy*. 2022. arXiv: 2203.11242 [cs.LG].

[13] Phillip Isola et al. *Image-to-Image Translation with Conditional Adversarial Networks*. 2018. arXiv: 1611.07004 [cs.CV].

[14] Tero Karras, Samuli Laine, and Timo Aila. *A Style-Based Generator Architecture for Generative Adversarial Networks*. 2019. arXiv: 1812.04948 [cs.NE].

[15] Y. Lecun et al. "Gradient-based learning applied to document recognition". In: *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2324. DOI: 10.1109/5.726791.



(a) Generator Loss (b) Discrete Loss

(c) Continuous Loss

Fig. 10: All loss curves of infoGAN training.

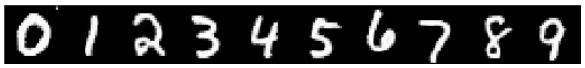
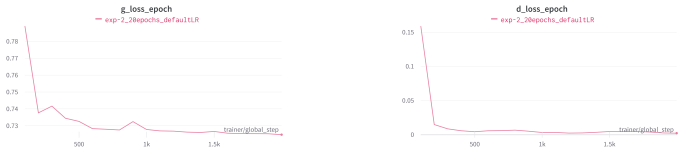


Fig. 11: InfoGAN Grid of Generated Digits

Fig. 12: Pyramid Resolution Comparison (similar to NB along with the dataset). Left to Right: $96x96$, 2x resolution and 4x resolution



(a) SRGAN G-Loss (b) SRGAN D-Loss

Fig. 13: Loss Curves for Successful SRGAN training.

[16] Christian Ledig et al. *Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network*. 2017. arXiv: 1609.04802 [cs.CV].

[17] Shuang Liu, Olivier Bousquet, and Kamalika Chaudhuri. *Approximation and Convergence Properties of Generative Adversarial Learning*. 2017. arXiv: 1705.08991 [cs.LG].

[18] John Nash. "Non-Cooperative Games". In: *Annals of Mathematics* 54.2 (1951), pp. 286–295. ISSN: 0003486X. URL: http://www.jstor.org/stable/1969529 (visited on 03/26/2023).

[19] Alec Radford, Luke Metz, and Soumith Chintala. *Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks*. 2016. arXiv: 1511.06434 [cs.LG].

[20] Tim Salimans et al. *Improved Techniques for Training GANs*. 2016. arXiv: 1606.03498 [cs.LG].

[21] Karen Simonyan and Andrew Zisserman. *Very Deep Convolutional Networks for Large-Scale Image Recognition*. 2015. arXiv: 1409.1556 [cs.CV].

[22] Jascha Sohl-Dickstein et al. *Deep Unsupervised Learning using Nonequilibrium Thermodynamics*. 2015. arXiv: 1503.03585 [cs.LG].

[23] Christian Szegedy et al. *Rethinking the Inception Architecture for Computer Vision*. 2015. arXiv: 1512.00567 [cs.CV].

[24] Radim Tylecek and Radim Sára. "Spatial Pattern Templates for Recognition of Objects with Regular Structure". In: *German Conference on Pattern Recognition*. 2013.

[25] Zhou Wang and Alan C. Bovik. "Mean squared error: Love it or leave it? A new look at Signal Fidelity Measures". In: *IEEE Signal Processing Magazine* 26.1 (2009), pp. 98–117. DOI: 10.1109/MSP.2008.930649.

[26] Zhou Wang, Eero P. Simoncelli, and Alan Conrad Bovik. "Multiscale structural similarity for image quality assessment". In: *The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003* 2 (2003), 1398–1402 Vol.2.

[27] Lilian Weng. "What are diffusion models?" In: *lilianweng.github.io* (July 2021). URL: https://lilianweng.github.io/posts/2021-07-11-diffusion-models/.

[28] Yuanbo Xiangli et al. *Real or Not Real, that is the Question*. 2020. arXiv: 2002.05512 [cs.LG].

[29] Han Xiao, Kashif Rasul, and Roland Vollgraf. *Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms*. 2017. arXiv: 1708.07747 [cs.LG].

[30] Ling Yang et al. "Diffusion models: A comprehensive survey of methods and applications". In: *arXiv preprint arXiv:2209.00796* (2022).

[31] Richard Zhang et al. "The Unreasonable Effectiveness of Deep Features as a Perceptual Metric". In: *CVPR*. 2018.

[32] Hamid R. Sheikh Zhou Wang Alan C. Bovik and Eero P. Simoncelli. *Image Quality Assessment: From Error Visibility to Structural Similarity*. 2004.

[33] Jun-Yan Zhu et al. *Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks*. 2020. arXiv: 1703.10593 [cs.CV].

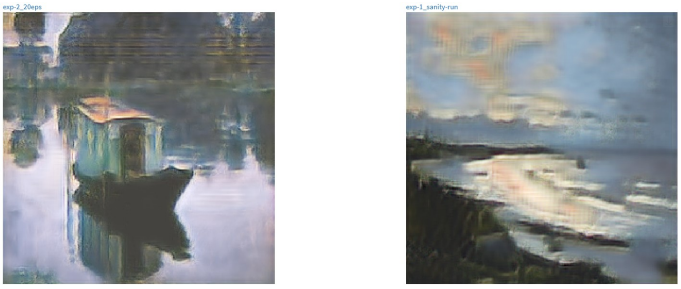


Fig. 14: Generated Photo-realistic Images. Left: 20 epoch run & Right: 5 epoch run results

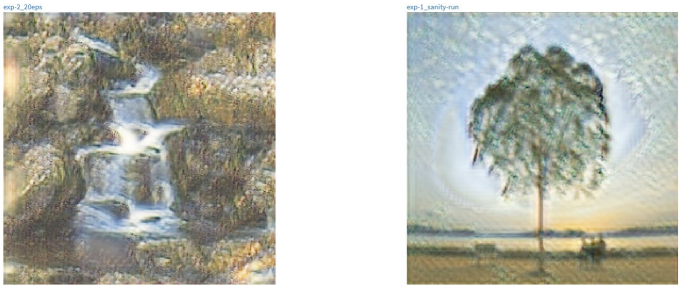


Fig. 15: Generated Monet Art Styled Images. Left: 20 epoch run & Right: 5 epoch run results

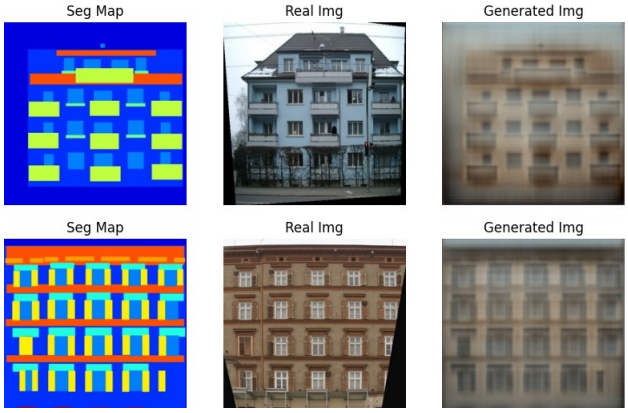


Fig. 16: Pix2Pix: Training Progression (1/2)

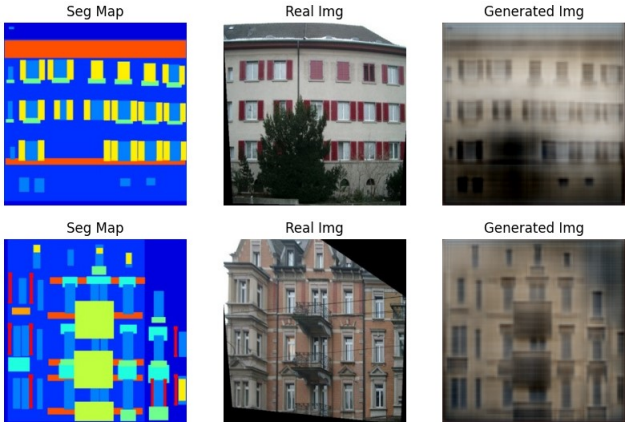


Fig. 17: Pix2Pix: Training Progression (2/2)