Image to Image transformation using GANs (9)

Shree Harsha K N, *AM23S008,* and Vamsi Krishna Badri, *CS23Z078,*

Abstract

This project addresses the challenge of domain adaptation in computer vision, focusing on the disparity between high-quality DSLR images and low-quality USB-cam images. The objective is to employ an image-to-image translation model, specifically the pix2pix algorithm, for domain adaptation. By transforming low-quality images to match the characteristics of high-quality images, this model aims to bridge the performance gap observed when testing on USB-cam images. The dataset utilized for training is derived from the DPED dataset, and despite the constraint of training on 100 x 100 image patches, the model demonstrates promising results with minimal added blur.

I. Introduction

High-quality DSLR images often exhibit superior performance when utilized to train models, leading to a domain gap when testing on lower-quality USB-cam images. To address this issue, the project leverages image-to-image translation, a technique proven effective in adapting models across domains. The chosen algorithm, pix2pix, is employed to transform mobile-cam images into high-quality DSLR-like counterparts, with a focus on maintaining consistent dimensions.

A. DSLR Photo Enhancement Dataset

To address the challenge of image translation from lower-quality smartphone images to superior-quality DSLR images, we introduce the DSLR Photo Enhancement Dataset (DPED)[1], a large-scale real-world dataset designed for general photo quality enhancement tasks. DPED comprises photos captured simultaneously by three smartphones (Sony, iPhone, BlackBerry) and a DSLR camera (Canon), mounted on a tripod and activated remotely. Over 22,000 photos were collected in diverse settings and lighting conditions during a three-week period. To overcome the computational challenges of training convolutional neural networks (CNNs) on aligned high-resolution images, we adopted a patch-based approach, extracting 100×100 pixel patches using a non-overlapping sliding window technique. Experimentation showed that larger patch sizes did not significantly improve performance and required more computational resources. Patches with a cross-correlation above 0.9 were included in the dataset, ensuring minimal displacements. A subset of approximately 100 original images was reserved for testing, while the rest were used for training and validation. This meticulous approach, with precisely matched training and test patches, facilitated feasible training while preserving accuracy in image matching.

II. ALGORITHMIC DESCRIPTION

Pix2pix GAN[2], or Generative Adversarial Networks, is a deep learning algorithm that uses a pair of neural networks to generate realistic images from input images. The algorithm consists of two parts: a generator and a discriminator. The generator takes an input image and generates a corresponding output image. The discriminator then takes both the input image and the generated output image and determines whether the generated image is realistic or not. The generator is trained to produce images that are realistic enough to fool the discriminator, while the discriminator is trained to correctly distinguish between real and generated images. During training, the generator and discriminator are trained in alternating fashion. The generator tries to produce more realistic images with each iteration, while the discriminator tries to become better at distinguishing between real and generated images. The training process continues until the generator produces images that are indistinguishable from real images. Pix2pix GAN is particularly effective for image-to-image translation tasks, such as converting a black and white image into a color image or converting a low-resolution image into a high-resolution image. The algorithm has been used in various applications such as image restoration, style transfer, and semantic segmentation.

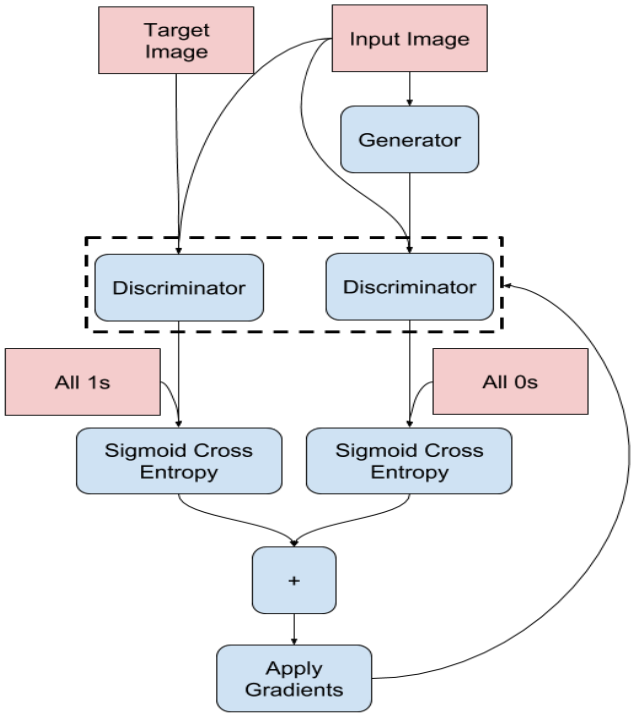
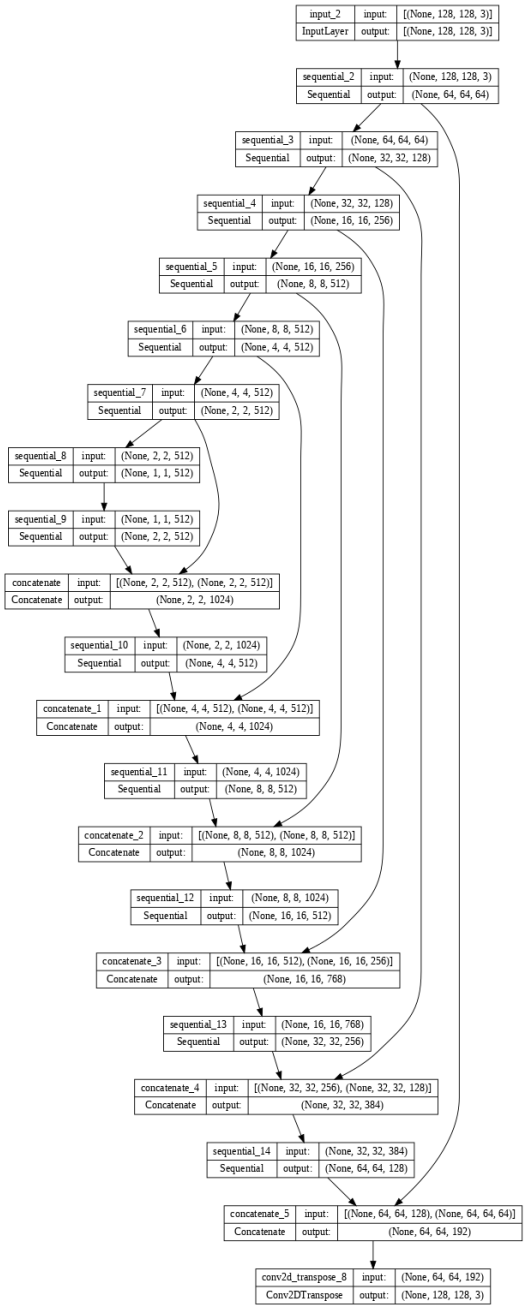
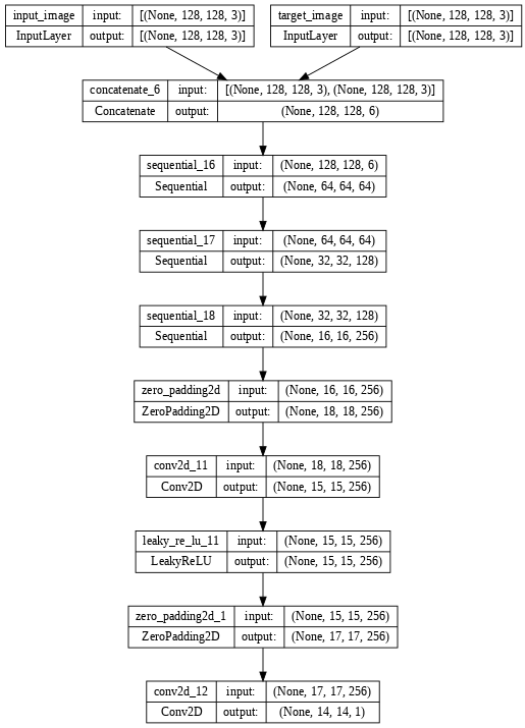


Fig. 1: Flowchart of Pix2Pix



(a) Generator architecture[3]



(b) Discriminator architecture[3]

III. OUTPUT

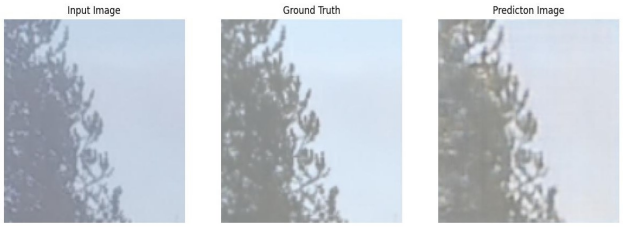
We trained the Model with 1000 image sample pairs from DPED[1] dataset with a batch size of 10 upto 100 epochs with a NVIDIA Tesla T4 provided by google colab as free to use resource. Training took approximately 43 mins. Once trained, the model can be used for inference on any image, which will go through a processing stage, which resizes images to 128 * 128 and then generates the image, it takes, 5 sec to generate the output.

A. Blackberry Images

(a) epoch =14

(b) epoch=45

(c) epoch=90

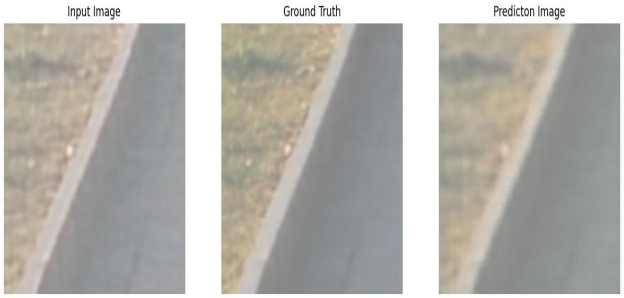


(d) epoch =30

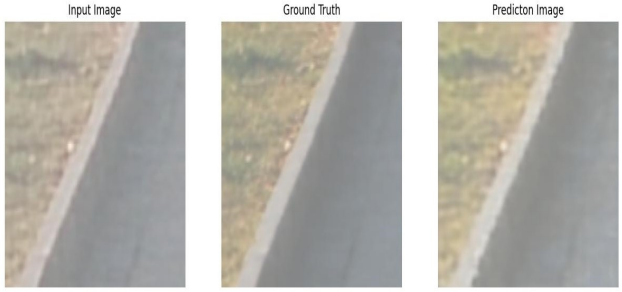
(e) epoch=79

(f) epoch=86

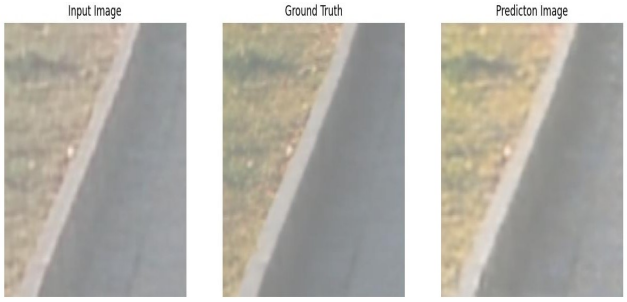
B. Sony Images



(a) epoch =3

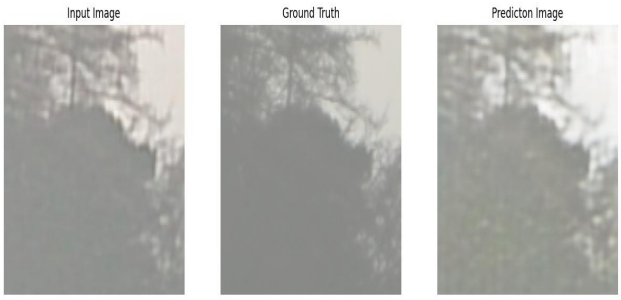


(b) epoch=64



(c) epoch=99

(d) epoch =12

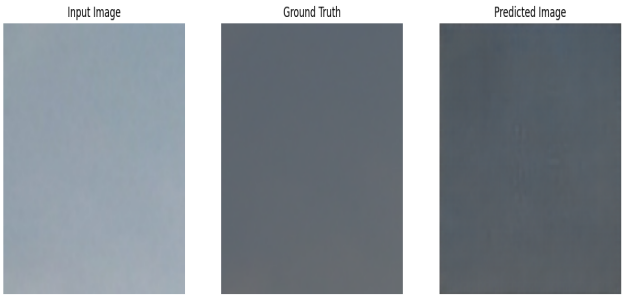


(e) epoch=63

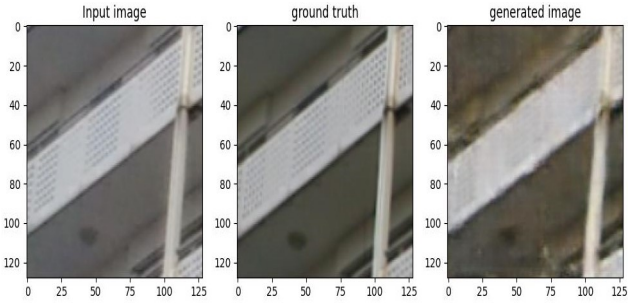
(f) epoch=92

C. Quantitative evaluation

We use classical distance metrics, namely PSNR and SSIM scores: the former measures signal distortion w.r.t. the reference, the latter measures structural similarity which is known to be a strong cue for perceived quality.

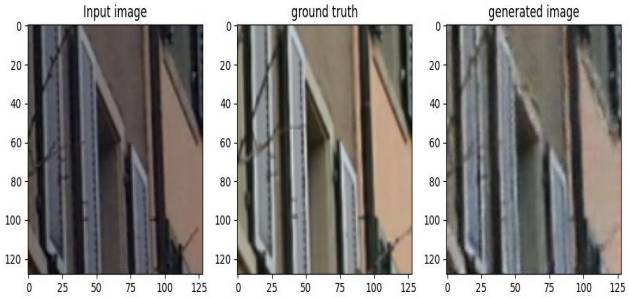


(a) SSIM:0.85411215, PSNR:19.880325

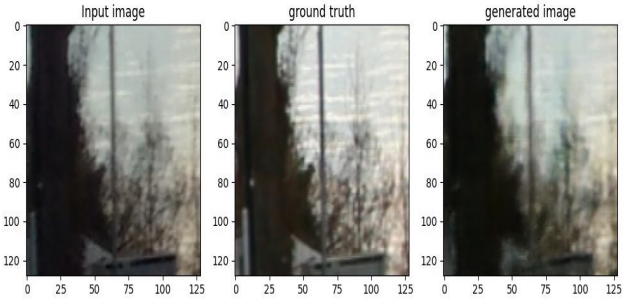


(b) SSIM:0.9487559, PSNR:17.235355

(c) SSIM:0.9395153, PSNR:17.168617

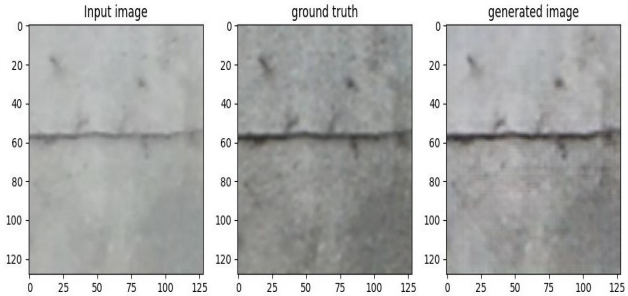


(d) SSIM:0.85199744, PSNR:19.780933

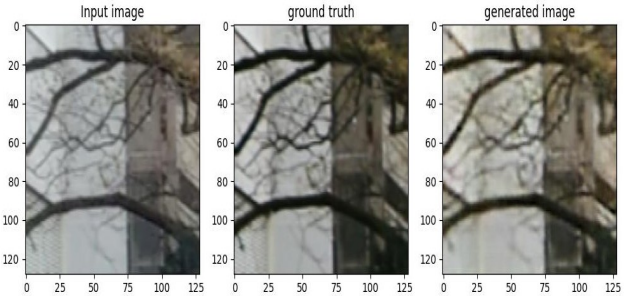


(e) SSIM:0.8059537, PSNR:17.709656

(f) SSIM: 0.90895295, PSNR: 24.369741



(g) SSIM:0.84850055, PSNR:23.684502



(h) SSIM:0.8330156, PSNR:25.415565

D. Observation

Training Pix2Pix on smaller image sizes may result in limited generalization to higher dimension images during inference. It's generally recommended to train a model on the target resolution to capture the details and patterns specific to that scale. Using a model trained on smaller images for higher dimensions may lead to loss of information and lower quality results due to the limited receptive field and feature representation learned during training. We tried to create patches of size 100 * 100 from the higher resolution image and run inference on each of the patch and stitch the image back again, due to the pix2pix architecture we need image size's which are powers of 2, easier to downsample and upsample to the same resolutions. Hence we resized the training images from 100 * 100 to 128 * 128, and the generated images borders were also padded by zero's same as training set. In the entire stitched image, we can observe the difference between the patches.

(a) Inference output(Blackberry)

(b) Inference output(Sony)

(c) Original Image[1]

IV. Conclusion

In conclusion, the implemented pix2pix algorithm proves effective in addressing the domain gap between DSLR and USB-cam images. Despite the constraint of training on 100 x 100 image patches, the model successfully translates low-quality images to high-quality counterparts, with minimal added blur. Future work could focus on refining the model to reduce the introduced blur by incorporating a composite perceptual error function that combines content, color and texture losses and exploring strategies to accommodate larger image sizes. Additionally, evaluating the model on diverse datasets and real-world scenarios would provide further insights into its generalization capabilities. The success of this approach opens doors for broader applications, such as enhancing image quality in various domains and scenarios.

References

[1] Ignatov, Andrey, Nikolay Kobyshev, Radu Timofte, Kenneth Vanhoey, and Luc Van Gool,*"Dslr-quality photos on mobile devices with deep convolutional networks."*, Proceedings of the IEEE International Conference on Computer Vision, pp. 3277-3285. 2017"

[2] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, Alexei A. Efros, *"Image-to-Image Translation with Conditional Adversarial Networks"*, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 2017, pp. 5967-5976, doi: 10.1109/CVPR.2017.632.

[3] Plots are generated using *tf.keras.utils.plot_model* Link to colab notebook