Motion Deblur Using GAN

Madhav Kanda Dhruv Patel Ksheer Agrawal Prajwal Singh Shanmuganathan Raman

Indian Institute Of Technology, Gandhinagar

Introduction

- Motion blur is a common problem in images captured using lightweight devices like mobile phones.
- It occurs due to the finite exposure interval and the relative motion between the capturing device and the captured object
- Blur induced in images is often non-uniform. Since the blur-kernel is unknown, it is difficult, estimating the spatially non uniform kernel.
- We have implemented the existing Generative Adversarial Network (GAN) based architecture for motion deblurring [4] and proposed changes to improve the metric score.

Generative Adversarial Network

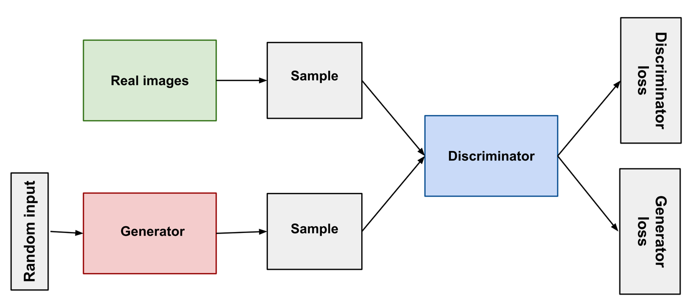


Figure 1. GAN model for understanding[1]

- GAN model consists of two neural networks: a generator and a discriminator.
- The generator network generates synthetic data samples that resemble the training data, while the discriminator network tries to differentiate between the synthetic data and the real data.
- The generator is trained to produce realistic data while the discriminator is trained to distinguish between real and synthetic data. The networks work together in an adversarial manner to improve the generator's ability to generate realistic data.

GAN based debluring

- The dimension of the information is kept constant throughout the network.
- Generator has three main components, as shown below. It also uses Global Skip connection for better image generation performance.
- The discriminator used is a Markovian Patch Discriminator. This enforces rich coloration in natural images.

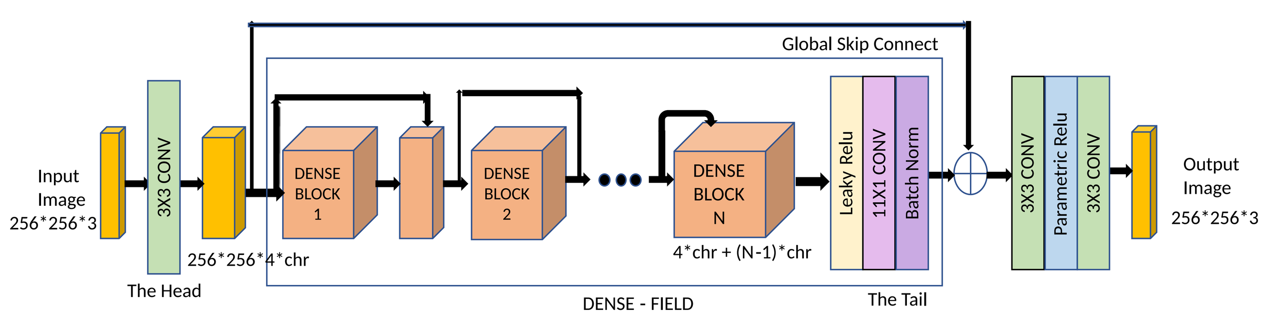


Figure 2. Generator model architecture

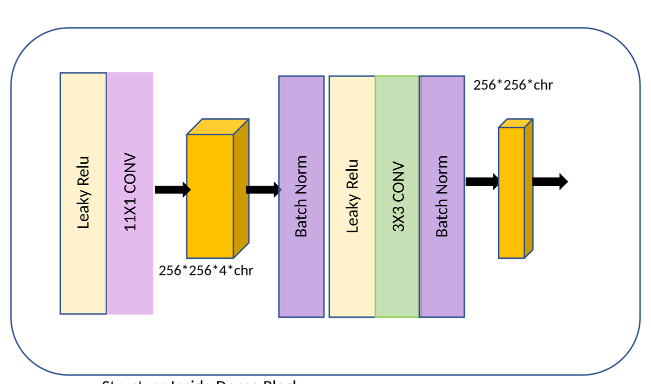


Figure 3. Dense Block architecture

Loss

For the loss, we use the summation of three losses generative adversial loss, L1 loss and LPIPS loss.

$$\mathcal{L}\mathrm{GAN}(G, D) = \mathbb{E}x \sim p_{\text{data}}(x)[\log D(x)] + \mathbb{E}z \sim pz(z)[\log(1 - D(G(z)))]$$

$$\mathrm{LPIPS}(I_1, I_2) = \sum_{i=1}^{N} w_i \cdot d_i(f_i(I_1), f_i(I_2)),$$

$$d_i(a, b) = \frac{1}{2}(1 - \mathrm{SSIM}(a, b))^2,$$

$$\mathrm{SSIM}(a, b) = \frac{(2\mu_a\mu_b + c_1)(2\sigma_{ab} + c_2)}{(\mu_a^2 + \mu_b^2 + c_1)(\sigma_a^2 + \sigma_b^2 + c_2)}.$$

Dataset

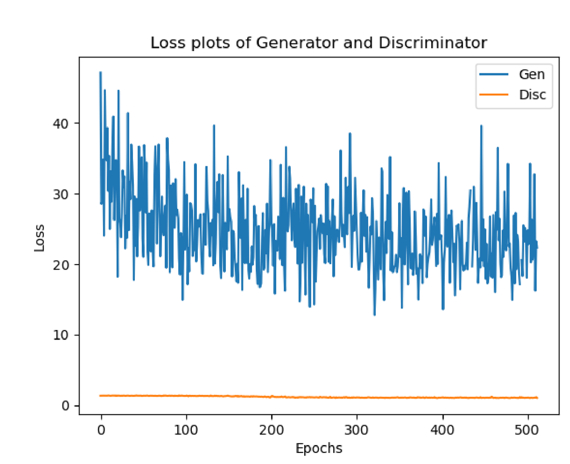
- Used the GoPro dataset which has corresponding separate blur and sharp images of objects.
- Concatenated the corresponding blur and sharp images of the object to obtain a single image of 512 X 256px. This image is fed into the model for training and testing purposes.

Data type	Statistics
Train Images	2304
Test Images	534
Input size	512 X 256

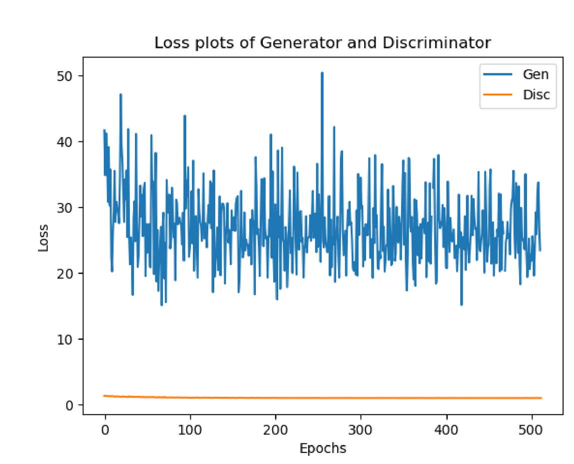
Table 1. GoPro Dataset

Experiments

- Implemented the GAN model for Image deblurring based on the architecture [4] in Pytorch for experimentation purposes.
- In contrast to the VGG loss commonly used in generator networks, we employ the Learned Perceptual Image Patch Similarity **(LPIPS)** loss function in our approach, as it provides more accurate coverage of color contrast
- Implemented **differential augmentation** technique to ensure a consistent flow of gradients from the discriminator during training, preventing them from becoming zero and promoting effective learning and generalization of our model.
- Incorporated **WGAN** loss as it introduces a gradient penalty to enforce the Lipschitz constraint on the discriminator. This helps to improve stability and convergence of training process.



((a)) Using Diffaug loss.



((b)) Using VGG loss.

Results

((a)) Using Diffaug loss.

((b)) Using VGG loss.

((c)) Using LPIPS loss.

((d)) Using WGAN loss.

Figure 5. Comparison of deblurred images using different losses on one of the GoPro images

((a)) Using Diffaug loss.

((b)) Using VGG loss.

((c)) Using LPIPS loss.

((d)) Using WGAN loss.

Figure 6. Comparison of deblurred images using different losses on one of the GoPro image

Conclusions

Methods	PSNR	SSIM	MS-SSIM	F-SIM	VIF
Original	21.01	0.789	0.904	0.891	0.412
WGAN	22.022	0.819	0.931	0.892	0.536
LPIPS	23.47	0.8438	0.948	0.903	0.506
Diffaug	23.51	0.8396	0.9470	0.900	0.503

Table 2. Performance comparison of different deblurring methods

Future Work

- Compare the work with other similar works done using diffusion models [2]
- Our model achieves an overall good metric score in a low data regime. Thus, we will explore more techniques to enhance the results in low data regime.

References

[1] Google Developer. Gan, 2013. [Online; accessed April 23, 2022].

[2] Orest Kupyn, Tetiana Martyniuk, Junru Wu, and Zhangyang Wang. Deblurgan-v2: Deblurring (orders-of-magnitude) faster and better, 2019.

[3] Chuan Li and Michael Wand. Precomputed real-time texture synthesis with markovian generative adversarial networks, 2016.

[4] Sainandan Ramakrishnan, Shubham Pachori, Aalok Gangopadhyay, and Shanmuganathan Raman. Deep generative filter for motion deblurring. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV) Workshops*, Oct 2017.