**Image Inpainter: Deblurring Using Conditional Generative Adversarial Networks**

Rina Damdoo, Sujal Agrawal, Ujjwal Prithyani, Vivek Meher, Vighnesh Gupta

Shri Ramdeobaba College of Engineering and Management,Nagpur, Maharashtra

{damdoor, agrawalsr\_14, prithyaniuv, mehervj, guptavs\_1}@rknec.edu

**Abstract**

*Image inpainting and image blur restoration are critical tasks in computer vision that involve repairing missing or damaged regions of an image and correcting blur effects, respectively. Both problems are inherently challenging due to the need for accurate content prediction while preserving the image's overall structure, texture, and context. Generative Adversarial Networks (GANs) have emerged as a powerful tool for addressing these issues, demonstrating significant efficacy in both inpainting and deblurring tasks. GANs utilize two neural networks—a generator and a discriminator—working in tandem to create realistic images: the generator produces synthetic images, while the discriminator distinguishes between real and synthetic images. This adversarial process enables the generator to progressively improve its output and the discriminator to become more adept at differentiating between real and generated content. In this study, we explore the application of GANs to the dual problems of image inpainting and blur restoration. Through extensive experimentation on diverse datasets of both synthetic and real-world blurred images, we demonstrate that proposed GAN-based approach effectively restores sharp details and textures, even under severe blur conditions. Proposed results underscore the potential of GANs as a versatile and robust solution for advanced image restoration tasks.*

1. **Introduction**

This work is on blind motion deblurring of a single photograph. GANs are known for the ability to preserve texture details in images, create solutions that are close to the real image manifold and look perceptually convincing. Inspired by recent work on image super-resolution and image-to-image translation by generative adversarial networks, we treat deblurring as a special case of such image-to-image translation. We present DeblurGAN– an approach based on conditional generative adversarial networks and a multi-component loss function. Unlike previous work we use Wasserstein GAN with the gradient penalty and perceptual loss . The unconditioned GANs, like original GAN introduced by Ian Goodfellow, have no control over the data generated. However, in conditional GANS, additional information is provided to manage the data generation process. This encourages solutions which are perceptually hard to distinguish from real sharp images and allows to restore finer texture details than if using traditional MSE or MAE as an optimization target.

We make three significant contributions to the field of image inpainting and deblurring. First, we used a novel GAN-based architecture and a custom-designed loss function that achieve state-of-the-art performance in motion deblurring while being faster than the most efficient existing methods. Second, we introduce an automated method for generating a diverse motion deblurring dataset using random trajectories from sharp images, which, when combined with existing datasets, leads to improved deblurring performance compared to training solely on real-world blurred images. Finally, we present a new dataset and evaluation framework for assessing deblurring algorithms based on their effectiveness in improving object detection results, offering a more practical and comprehensive measure of deblurring performance. These contributions advance both the efficiency and the evaluation standards of deblurring techniques, demonstrating significant improvements in both technical performance and practical applicability.

****

Figure 1: GoPro images processed by DeblurGAN. Blurred– left, DeblurGAN– center, ground truth sharp– right.

**2. Related work**

**2.1. Image Deblurring**

The common formulation of the non-uniform blur model can be expressed as:

IB = k(M) ∗ Is + N (1)

In this equation, IB represents the blurred image, k(M) denotes the unknown blur kernels determined by the motion field M, Is is the sharp latent image, ∗ signifies the convolution operation, and N stands for the additive noise.

Image deblurring problems are generally categorized into two types: blind and non-blind deblurring. Early research primarily focused on non-blind deblurring, which operates under the assumption that the blur kernels k(M) are known. Traditional techniques for non-blind deblurring often involve methods like the Lucy-Richardson algorithm, Wiener filter, or Tikhonov regularization to perform the deconvolution and estimate the sharp image Is.

In contrast, blind deblurring methods must estimate both the latent sharp image Is and the blur kernels k(M), as the blur function is unknown. This makes blind deblurring a more challenging, ill-posed problem, typically approached through heuristics, image statistics, and assumptions about the sources of the blur. These methods often assume that the blur caused by camera shake is uniform across the image. The process generally starts with estimating the camera motion to infer the blur kernel and then applies deconvolution to reverse the blur effects.

Over time, various methods have been developed to address these issues. Some methods use iterative approaches to refine estimates of the motion kernel and sharp image, leveraging parametric prior models to improve results. However, these iterative methods can suffer from long running times and difficulties in determining when to stop the iterations. Alternatively, some techniques make use of local linearity assumptions and simple heuristics for quick blur kernel estimation, though these are often limited to a narrow range of images.

Recently, advancements have been made in non-uniform blind deblurring through new algorithms that model the blurring process based on geometric principles, such as the rotational velocity of the camera during exposure or the assumption of 3D camera movement. With the rise of deep learning techniques, new approaches employing convolutional neural networks (CNNs) have been introduced. These methods range from using CNNs to estimate the blur kernel to predicting complex motion kernels in Fourier space or leveraging fully convolutional networks for motion flow estimation. More recent developments include kernel-free, end-to-end approaches that utilize multi-

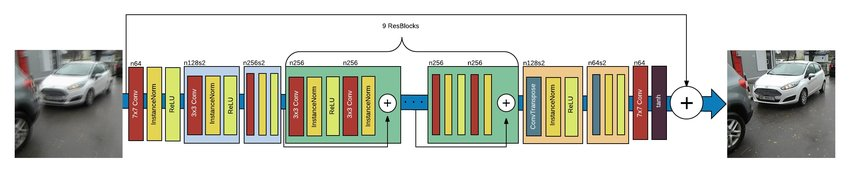


Figure 2: DeblurGAN generator architecture. DeblurGAN contains two strided convolution blocks with stride 1 2, nine residual blocks [[1]](#References_Page) and two transposed convolution blocks. Each ResBlock consists of a convolution layer, instance normalization layer, and ReLU activation.

scale CNNs for direct image deblurring, as well as methods that combine frameworks and dense networks for effective blind deblurring across various blur sources.

**2.2. Generative Adversarial Networks**

The idea of generative adversarial networks, introduced by Goodfellow et al. [[2]](#References_Page), is to define a game between two competing networks: the discriminator and the generator. The generator receives noise as an input and generates a sample. A discriminator receives a real and generated sample and is trying to distinguish between them. The goal of the generator is to fool the discriminator by generating perceptually convincing samples that cannot be distinguished from the real one. The game between the generator G and discriminator D is the minimax objective:



where Pr is the data distribution and Pg is the model distribution, defined by x̄ = G(z), z ~ P(z), the input z is a sample from a simple noise distribution. GANs are known for its ability to generate samples of good perceptual quality, however, training of vanilla version suffer from many problems such as mode collapse, vanishing gradients etc, as described in [[3]](#Referencws_page2). Minimizing the value function in GAN is equal to minimizing the Jensen-Shannon divergence between the data and model distributions on x. Arjovsky et al. [[4]](#References_Page) discuss the difficulties in GAN training caused by JS divergence approximation and propose to use the

Earth-Mover (also called Wasserstein-1) distance W(q p). The value function for WGAN is constructed using Kantorovich-Rubinstein duality [[5]](#Referencws_page2):



where D is the set of 1-Lipschitz functions and Pg is once again the model distribution The idea here is that critic value approximates K.W(Pr Pθ), where K is a Lipschitz constant and W(Pr Pθ) is a Wasserstein distance. In this setting, a discriminator network is called critic and it approximates the distance between the samples. To enforce Lipschitz constraint in WGAN Arjovsky et al. add weight clipping to [-c c]. Gulrajani et al. [[6]](#References_Page) propose to add a gradient penalty term instead:



to the value function as an alternative way to enforce the Lipschitz constraint. This approach is robust to the choice of generator architecture and requires almost no hyperparameter tuning. This is crucial for image deblurring as it allows to use novel lightweight neural network architectures in contrast to standard Deep ResNet architectures, previously used for image deblurring [[7]](#Referencws_page2).

**3. Proposed Methodology**

The goal is to recover sharp image Is given only a blurred image IB as an input, so no information about the blur kernel is provided. Debluring is done by the trained CNN Gθ , to which we refer as the Generator. For each IB it estimates corresponding IS image. In addition, during the training phase, we introduce critic the network Dθ and train both networks in an adversarial manner.

**3.1 Network Architecture**

Generator CNN architecture is shown in Figure 2. It is similar to one proposed by Johnson et al. [[8]](#References_Page) for style transfer task. It contains two strided convolution blocks with stride 1 2, nine residual blocks [[1]](#References_Page) (ResBlocks) and two transposed convolution blocks. Each ResBlock consists of a convolution layer, instance normalization layer [[9]](#Referencws_page2), and ReLU [[10]](#Referencws_page2) activation. Dropout [[11]](#Referencws_page2) regularization with a probability of 0.5 is added after the first convolution layer in each ResBlock. In addition, we introduce the global skip connection which we refer to as ResOut. CNN learns a residual correction IR to the blurred image IB, so IS = IB +IR. We find that such formulation makes training faster and resulting model generalizes better. During the training phase, we define a critic network Dθ , which is Wasserstein GAN [[2]](#References_Page) with gradient penalty [[6]](#References_Page), to which we refer as WGAN-GP. The architecture of critic network is identical to PatchGAN [[12]](#References_Page) [[13]](#References_Page). All the convolutional layers except the last are followed by InstanceNorm layer and LeakyReLU [[14]](#Referencws_page2) with = 02.

**3.2 Loss Function**

We formulate the loss function as a combination of con tent and adversarial loss:

****

where the equals to100 in all experiments. Unlike Isola et al. [[12]](#References_Page) we do not condition the discriminator as we do not need to penalize mismatch between the input and output. Adversarial loss Most of the papers related to conditional GANs, use vanilla GAN objective as the loss [[15]](#References_Page)[[7]](#Referencws_page2) function. Recently provides an alternative way of using least aquare GAN [[16]](#Referencws_page2) which is more stable and generates higher quality results. We use WGAN-GP [[6]](#References_Page) as the critic function, which is shown to be robust to the choice of generator architecture [[4]](#References_Page). Our premilinary experiments with different architectures confirmed that findings and we are able to use architecture much lighter than ResNet152 [[7]](#Referencws_page2), see next subsection. The loss is calculated as the following:



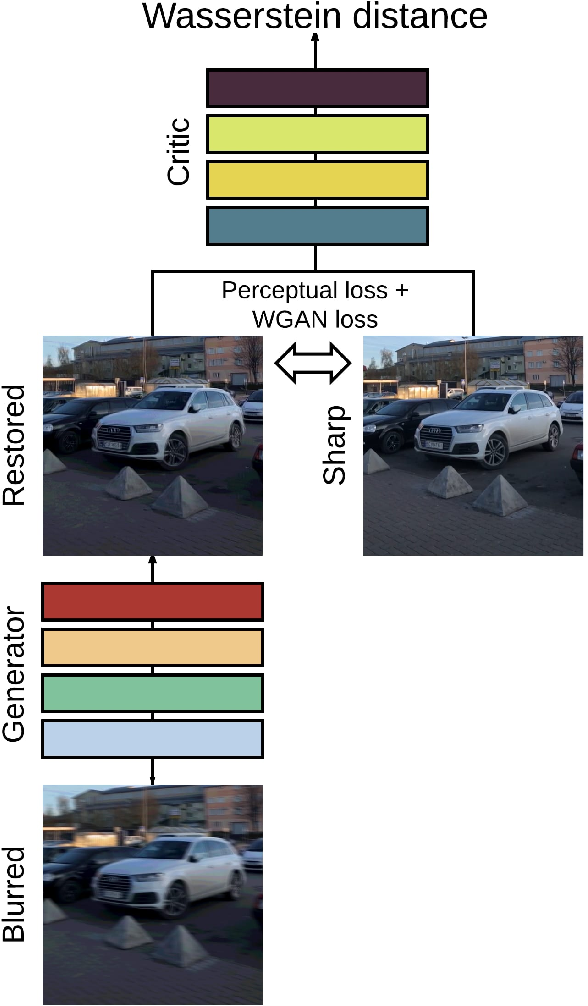


Figure 3: DeblurGAN training. The generator network takes the blurred image as input and produces the estimate of the sharp image. The critic network takes the restored and sharp images and outputs a distance between them. The total loss consists of the WGAN loss from critic and the perceptual loss [[8]](#References_Page). The perceptual loss is the difference between the VGG-19 [[17]](#Referencws_page2) conv3.3 feature maps of the sharp and restored images. At test time, only the generator is kept.

**4. Implementation Details**

In our study, we employed a pretrained version of the DeblurGAN model to address the image deblurring task. Our approach involved fine-tuning this pretrained model on a dataset of 1100 paired images consisting of blurred and deblurred car images. The following describes the details of our methodology, including the use of the pretrained model and the specifics of our fine-tuning process.

**4.1 Pretrained Model**

We utilized a pretrained DeblurGAN model that had been trained for 2000 epochs on the GoPro dataset [[7]](#Referencws_page2), which consists of high-resolution images captured with a high frame-rate camera. The model was initially trained using a single Maxwell GTX Titan-X GPU and employed the same network architecture and training procedures described in [[18]](#References_Page). The pretrained model had been trained with a batch size of 1, an initial learning rate of 10-4, and used the Adam optimizer with a learning rate schedule that linearly decayed from 10-4 to zero over the course of the training.

**4.2 HyperParameter Tuning**

To adapt the pretrained DeblurGAN model to proposed specific task, we performed additional fine-tuning for 10 epochs on the car image dataset. This dataset consisted of 1100 pairs of images, with each pair including a blurred image and its corresponding deblurred reference image. The images were preprocessed to a consistent resolution of 256x256 pixels to match the input requirements of the DeblurGAN model.

The initialization function sets up the ConditionalGAN model by initializing parameters such as the type of GAN (conditional), whether it's in training mode, and the device (CPU or GPU). It also initializes tensors for input and output images. The load function loads model checkpoints from saved files, restoring the model's state. It is useful for resuming training from a saved checkpoint or for deploying trained models for inference.

**set\_input** function prepares input data for the model by setting up input tensors for the forward pass. It takes input images (blurred and sharp) and converts them into appropriate tensor format for processing by the network. **get\_image\_path** function retrieves the file paths of input images used in the model. It can be useful for tracking the source of input data during training or testing.

Figure 3a Algorithm on which proposed model work.

**NormalizeImg** utility function normalizes images by scaling pixel values to a specified range. Normalization is essential for ensuring consistent input data for the model and stabilizing training.

**show\_MNIST** utility function, specific to MNIST dataset visualization, displays images in a grid format for visualization. It can be helpful for debugging and understanding the dataset used in the model. **encode\_labels** utility function encodes categorical labels into a one-hot encoding format. In CGANs, the generator takes both noise and class labels as input. Converting class labels into a one-hot encoding format ensures compatibility with the generator's input layer.

**4.3 Algorithm**

We also trained DeblurGAN on a combination of synthetically blurred images and images taken in the wild, where the ratio of synthetically generated images to the images taken by a high frame-rate camera is 2:1

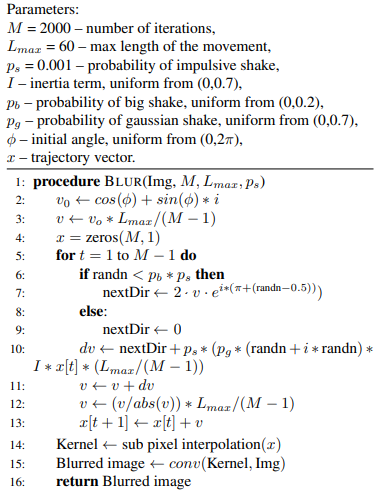




Figure 5: Blur image Dataset

 Figure 4: Deblur image Dataset

**5. Experimental Evaluation:**

**5.1 Dataset**

In proposed study, we utilized a dataset comprising a total of 1,050 image pairs, each pair consisting of a blurred and a corresponding deblurred image of various cars. Out of these, 1,000 image pairs were allocated for training purposes, while the remaining 50 pairs were designated for testing.

**5.2 Results**

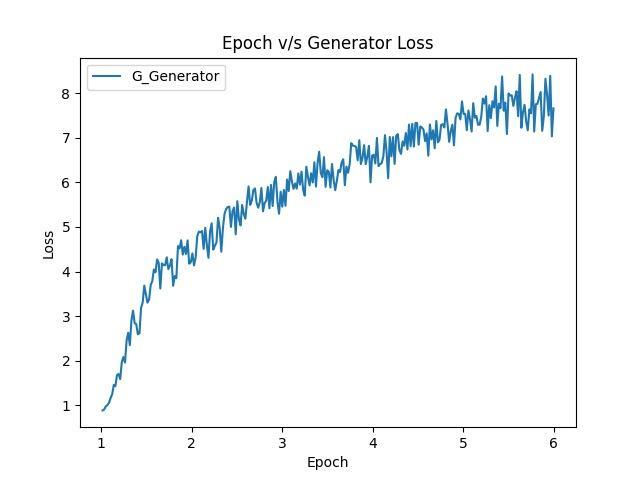
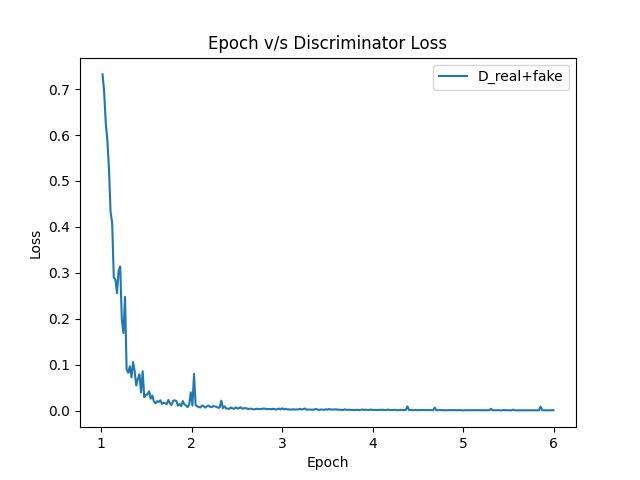
Figure 6: Input to our model

 Visual inspection of the deblurred images revealed notable improvements in clarity and detail recovery. The deblurred images exhibit enhanced edge sharpness and reduced blurring artifacts compared to the blurred inputs. Specific examples demonstrating the efficacy of the algorithm can be observed in Figures 6 and 7, which highlight significant enhancements in vehicle features and textures.

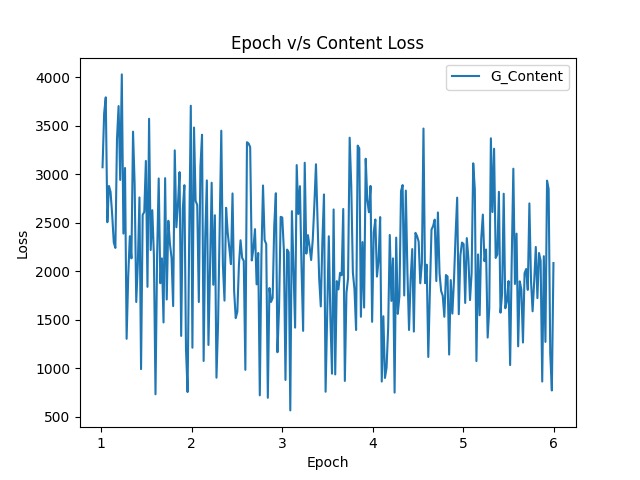
**5.2.1 Assessment of Performance Metrics:**

Figure 8 presents a graph comparing epoch v/s generator and discriminator loss values achieved by proposed deblurring algorithm against several methods. Proposed algorithm demonstrates a superior average value, indicating a higher level of image quality improvement.

Figure 7: Output of our model

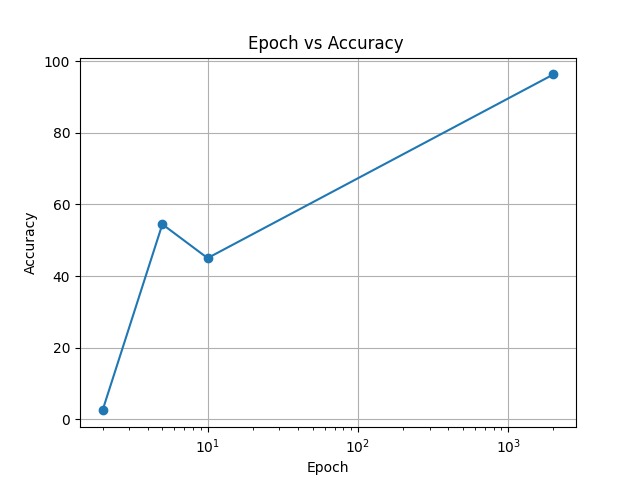
**Fig 8. Epoch v/s Generator Loss & Discriminator Loss**

**5.2.2 Content Loss Comparison:**

 Figure 9 displays a graph representing amount of content loss in deblurring images. Proposed method shows a lower average content loss, which signifies fewer discrepancies between the deblurred images and the original sharp images.

**Fig 9. Epoch v/s Content Loss**

**5.2.3 Computational Accuracy:**

****Figure 10 summarizes the average accuracy for proposed algorithm. Proposed approach achieves an average accuracy, which is higher & faster relative to other approaches.

**Fig 10. Epochs v/s Accuracy**

**6. Limitations:**

**Risk Of Overfitting To Training Distribution:**

When trained on a non-diverse dataset, blur deblur GANs are susceptible to overfitting, wherein the model learns to memorize training samples rather than generalize underlying blur deblur principles. Overfitting can manifest as poor performance on unseen data, particularly when the distribution of blur types or image characteristics deviates from those present in the training set. This limitation undermines the model's robustness and applicability to real-world scenarios.

**Bias In Deblurring Performance:**

Training on a non-diverse dataset can introduce bias into the blur deblur GAN model's performance, disproportionately favouring certain types of blur while neglecting others. This bias may manifest as varying levels of accuracy and quality in deblurring different types of images, depending on their similarity to the training data. As a consequence, the model's reliability and consistency in deblurring across diverse image datasets may be compromised.

**Training Time:**

The computational intensity of GAN training translates into prolonged training times, especially for complex models. This extended training duration can further exacerbate the resource requirements and costs associated with GAN experimentation and deployment.

**References:**

[1] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. arXiv preprint arXiv:1512.03385, 2015.

[2] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative Adversarial Networks. June 2014.

[3] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen. Improved Techniques for Training GANs. ArXiv e-prints, June 2016.

[4] M. Arjovsky, S. Chintala, and L. Bottou. Wasserstein GAN. ArXiv e-prints, Jan. 2017.

[5] C. Villani. Optimal Transport: Old and New. Grundlehren der mathematischen Wissenschaften. Springer Berlin Heidelberg, 2008.

[6] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. Courville. Improved Training of Wasserstein GANs. ArXiv e-prints, Mar. 2017.

[7] S. Nah, T. Hyun, K. Kyoung, and M. Lee. Deep Multi-scale Convolutional Neural Network for Dynamic Scene Deblurring. 2016.

[8] J. Johnson, A. Alahi, and L. Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In European Conference on Computer Vision, 2016.

[9] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. CoRR, abs/1412.6980, 2014.

[10] V. Nair and G. E. Hinton. Rectified linear units improve restricted boltzmann machines. In International Conference on Machine Learning (ICML), pages 807–814, 2010.

[11] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. J. Mach. Learn. Res., 15(1):1929– 1958, Jan. 2014.

[12] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. arxiv, 2016.

[13] C. Li and M. Wand. Precomputed Real-Time Texture Synthesis with Markovian Generative Adversarial Networks. ArXiv e-prints, Apr. 2016.

[14] B. Xu, N. Wang, T. Chen, and M. Li. Empirical evaluation of rectified activations in convolutional network. arXiv preprint arXiv:1505.00853, 2015.

[15] C. Ledig, L. Theis, F. Huszar, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi. Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network. ArXiv e-prints, Sept. 2016.

[16] X. Mao, Q. Li, H. Xie, R. Y. K. Lau, and Z. Wang. Least squares generative adversarial networks, 2016. cite arxiv:1611.04076.

[17] K. Simonyan and A. Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. ArXiv eprints, Sept. 2014.

[18] PyTorch. http://pytorch.org.