Super Resolution with Adaptive Data Augmentation using Generative Adversarial Networks

Visual and Multimedia Recognition Course Project

Alberto Baldrati, Giovanni Berti

Supervisors: Prof. Alberto Del Bimbo, Ing. Leonardo Galteri

Dipartimento di Ingegneria dell'Informazione
Università degli Studi di Firenze

Index

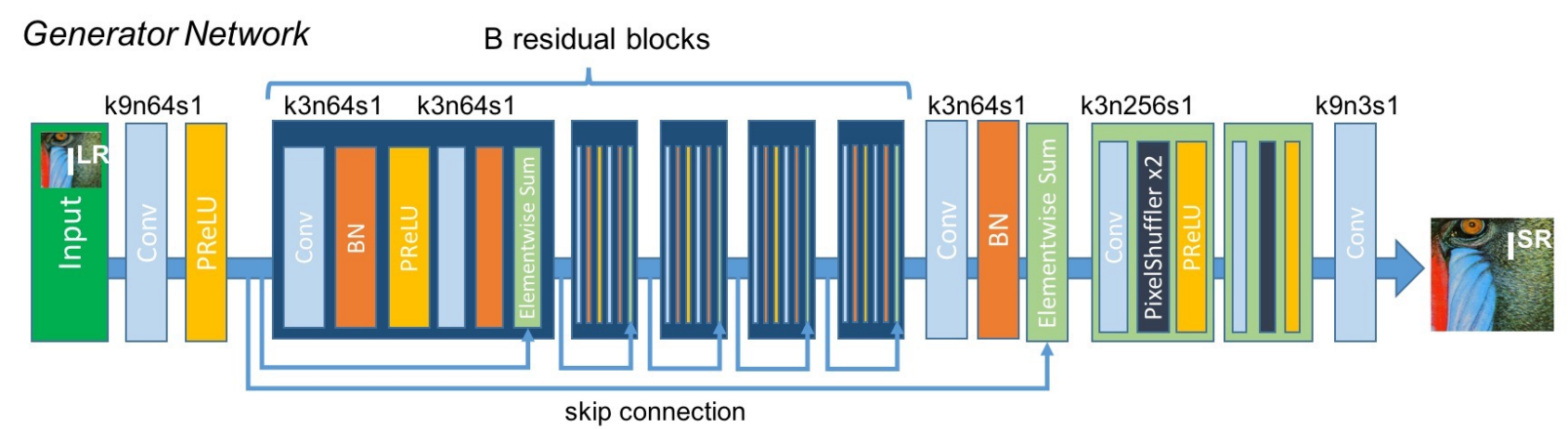
Introduction

Introduction

- In Karras et al., 2020 is presented a novel data augmentation technique that seeks to stabilize training and improve GANs performance in limited data regimes
- Its mathematical formulation ensures that reasonably no artifacts are introduced due to augmentation
- This data augmentation technique dynamically adapts to discriminator overfitting
- The original work applied this approach to image generation from latent space
- We implement and test this method in a super resolution setting

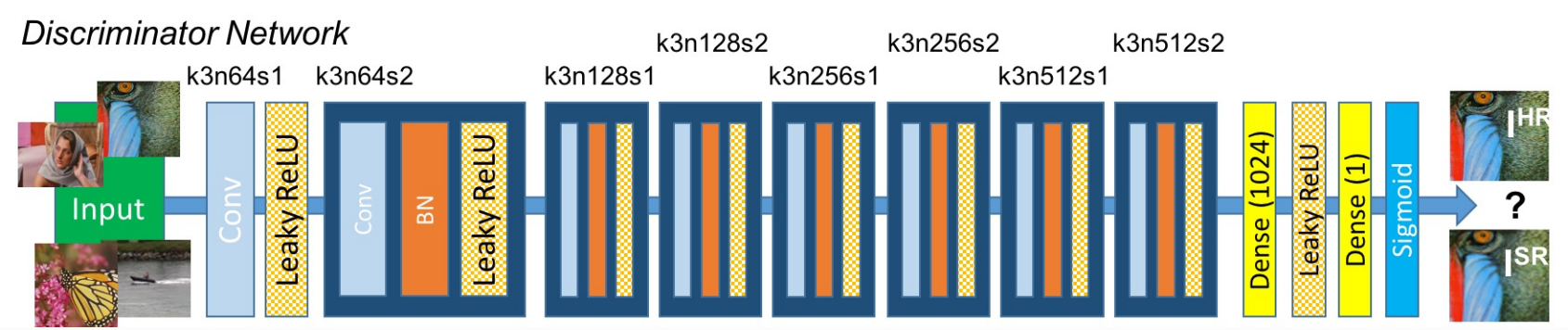
Network Architecture

Original SRGAN
Generator



- The generator is designed to give a $4\times$ upsampling
- Because of the purely convolutional structure of the generator it is independent of input size
- Employs pixel shuffling Shi et al., 2016 to perform sub-pixel convolutions

Original SRGAN
Discriminator



- Discriminator structure inspired by VGG network (filter bank size doubles every other block)
- Trailing feedforward network constraints input size
- Because of this we train discriminator and generator with fixed-size patches instead of full images

Architecture modifications

- Input-Output residual connection with bicubical upsampling added, which implies:
  - Faster convergence and local minima avoidance(He et al., 2015)
  - Initial weights correspond to approximately an identity function
  - Generator only has to learn the residual from a bicubic upsampling to a fully super-resolved image instead of the whole image from scratch
- Removed Batch Normalization layers in all *B* residual blocks
- Other architectural choices were tested (e.g. activation function, LR scheduling...), and this was the most stable and performant

Training procedure

Losses

- In super resolution tasks, the training loss is usually made up of two components: a perceptual loss and an adversarial loss
- The perceptual loss signals the generator the overall structure of the output images, while the adversarial loss is computed using the discriminator, and is the primary force that pushes the generator into outputting finer details.
- We chose a balancing factor such that the adversarial loss is about $5\% - 10\%$ of the perceptual loss
- The most natural choice for a perceptual loss is the pixel-wise MSE loss

$$l^{SR}_{MSE} = \frac{1}{\dim(\hat{y})} \|\hat{y} - y\|_2^2$$

Losses
Perceptual loss

- MSE loss (mainly in non adversarial training) tends to achieve high PSNR lacking high frequency content resulting in overly smooth textures

- In Ledig et al., 2017 the **VGG loss** is proposed:

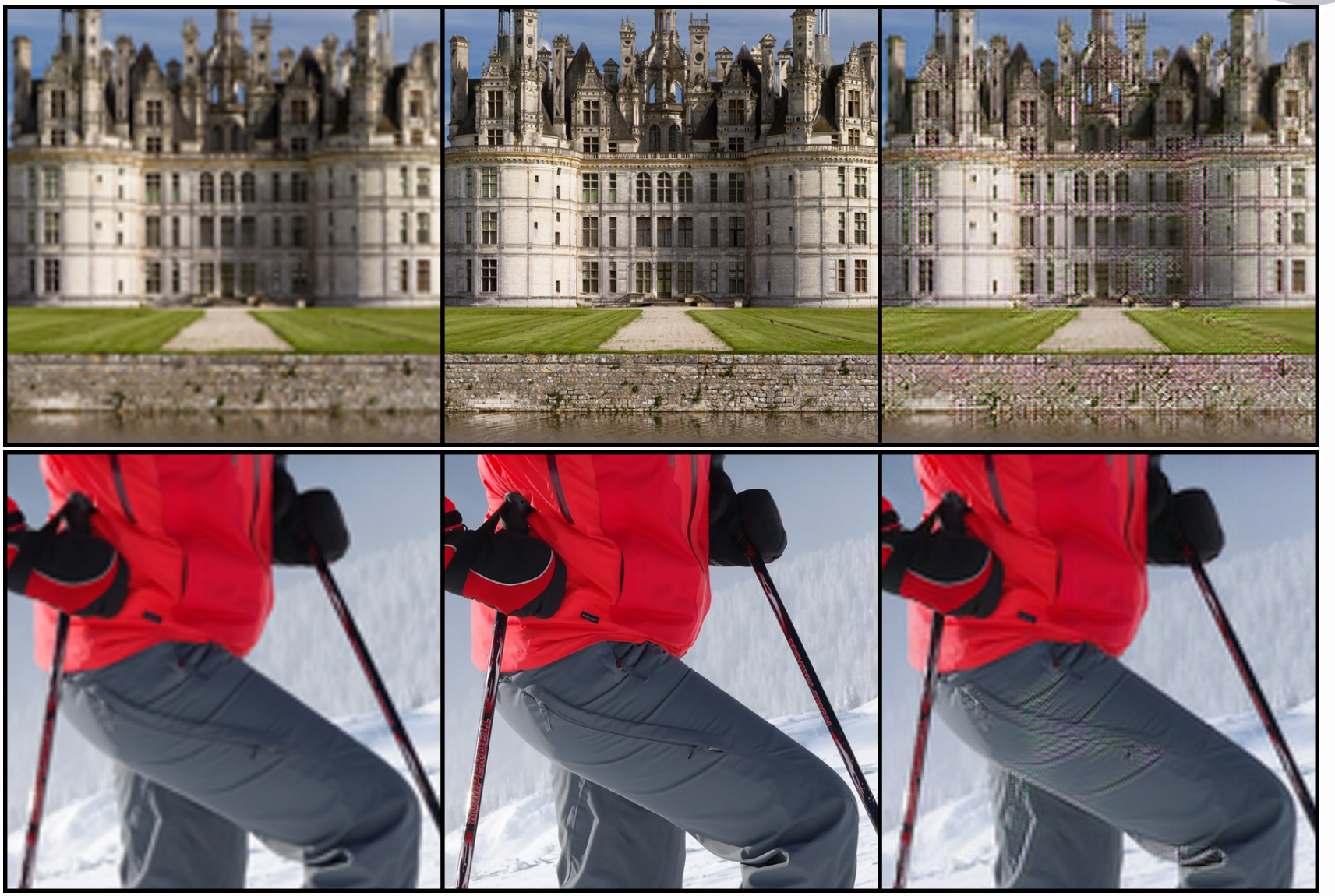
$$l^{SR}_{VGG} = \frac{1}{\dim(\phi(\hat{y}))} \left\| \phi(\hat{y}) - \phi(y) \right\|_2^2$$

  where $\phi(\cdot)$ denotes a fixed feature map produced by an intermediate layer of the VGG network

- In our experiments VGG loss led to greater instability and image artifacts, because of that we use the standard **MSE loss** as perceptual loss

Losses

VGG Loss induced artifacts



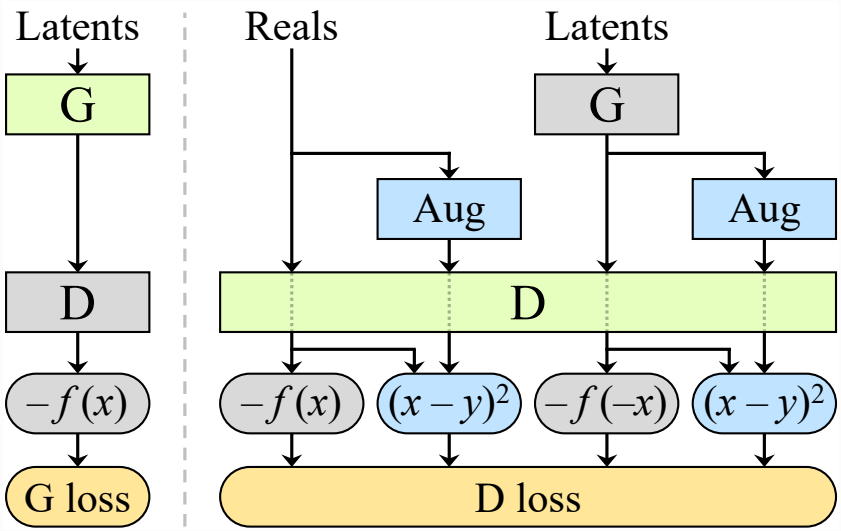
Training process

- Patch based training due to discriminator architecture, patch size: 128×128
- Each low resolution patch is constructed by taking a random crop from a high resolution image and downsampling
- The generator is pre-trained with the perceptual loss only for a number of epochs before training in an adversarial setting
- Generator and discriminator steps are taken in turns in order to reach a Nash equilibrium

Data Augmentation

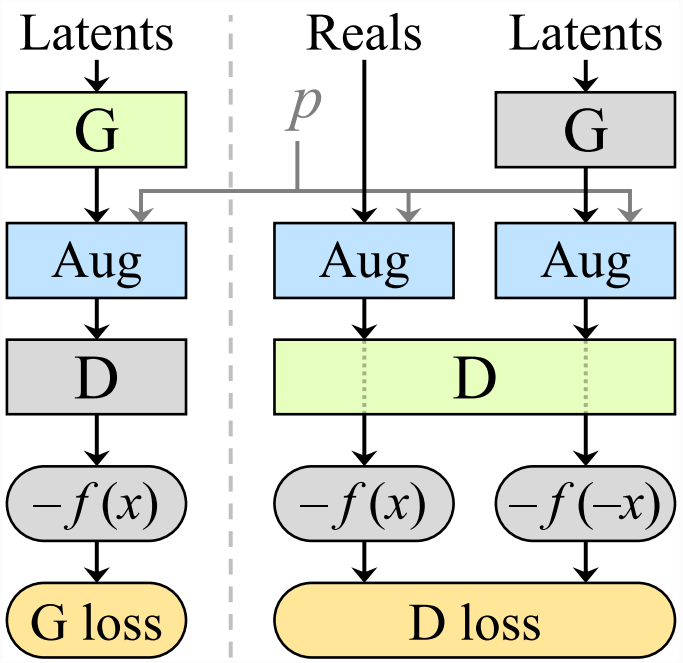
Previous Work
Balanced Consistency Regularization

- Proposed by Zhao et al., 2020
- Discriminator has consistency terms in loss: augmentation shouldn't change discriminator output
- Generator trained as usual
- This formulation *leaks* augmentation (lack of generator consistency term makes it that generator can include effects of augmentation into generated images)



Stochastic discriminator augmentation

- Proposed by Karras et al., 2020
- The discriminator works only with augmented distributions
- Image transformations are applied stochastically with augmentation probability *p*
- As long as transformations are invertible in the distribution sense, the training process should manage to invert them



Adaptive discriminator augmentation

- With Stochastic Discriminator Augmentation p becomes an additional hyperparameter
- Training dynamics is quite sensitive to p
- The authors designed an overfitting heuristic to dynamically change p depending on discriminator performance

$$r_t = \mathbb{E}\left[\text{sign}\left(D_{\text{train}}\right)\right]$$

where D_{train} is the discriminator output on the ground truth dataset before the activation function

- When r_t is above a chosen target value p is incremented, when below p is decremented

Image transformations

Transformation pipeline

- The original authors proposed a pipeline consisting in a composition of many different image transformations
- Pixel blitting, geometric transformations, color transformations, image-space filtering and image-space corruptions
- From a mathematical point of view this pipeline is guaranteed to be non-leaking
- Finite precision and finite sampling make it leaking for p values near 1
- The authors show empirically that as long as p is lower than 0.85 the pipeline doesn't introduce leaks

Image transformations

Examples

Image transformations

Pipeline modifications

- We introduced some modifications to the original pipeline to adapt it to our setting
- Because of our patch-based training we removed translation and cutout
- Our experiments showed that image filtering and arbitrary angle image rotation introduced artifacts in generated images, thus we removed both from the augmentation pipeline

Adaptive discriminator augmentation

Transformations parameters

Transformation	Parameters	Type
x flip	N/A	$\mathcal{U}\{0,1\}$
90 degree rotation	N/A	$\mathcal{U}\{0,1,2,3\}$
Isotropic scaling	$\sigma = 0.2 \log 2$	Lognormal$(1, \sigma^2)$
Anisotropic scaling	$\sigma = 0.2 \log 2$	Lognormal$(1, \sigma^2)$
Brightness	$\sigma = 0.2$	$\mathcal{N}(0, \sigma^2)$
Contrast	$\sigma = 0.5 \log 2$	Lognormal$(0, \sigma^2)$
Saturation	$\sigma = 0.2 \log 2$	Lognormal$(1, \sigma^2)$
Hue rotation	N/A	$\mathcal{U}(-\pi, +\pi)$
Luma flip	N/A	$\mathcal{U}\{0,1\}$
Noise	$\sigma = 0.1$	Halfnormal$(0, \sigma^2)$

Experiments

Experiments

- We conducted experiments on two dataset: FFHQ (Karras et al., 2019) and DIV2K (Ignatov et al., 2019)

- We considered several random subsamplings of the original datasets in order to test the effects of data augmentation when varying the number of data points available for training

- To avoid local minima and head start the generator, in all experiments we employ a pre-training phase where only the perceptual loss is active

FFHQ EXPERIMENTAL SETUP

- The dataset considered is FFHQ, downsampled to a 512×512 size
- FFHQ dataset contains 70k high resolution images of human faces, such regularities in the dataset make the super resolution task easier than in the general case
- The dataset has been split into a 50k portion for training and two 10k portions for validation and testing.
- All networks were pre-trained for **250k** iterations and trained in adversarial setting for **1.5M** iterations

FFHQ experimental setup

Training Hyperparameters

- **Optimizer**: Adam (both for the generator and the discriminator), with a learning rate $\eta = 1 \times 10^{-4}$
- **Patch size**: 128×128
- **Number of residual blocks B**: 16
- **Batch size**: 32
- ***k* minibatch update frequency**: 8 minibatches
- ***p* update amount**: 1×10^{-3}
- **Adversarial loss balancing factor**: 4×10^{-5}
- **r_t target value**: 0.6
- **One-sided label smoothing factor**: 0.9 , applied to ground truth images Salimans et al., 2016

FFHQ EXPERIMENTAL RESULTS

Dataset Percentage	Data Aug	Metric			
		FID	LPIPS	PSNR	SSIM
100%	No	2.62	0.0962	82.94	0.826
50%	No	2.91	0.0981	82.85	0.825
25%	No	2.81	0.1008	82.80	0.826
10%	No	2.99	0.1002	82.72	0.825
5%	No	2.71	0.0984	82.99	0.827
1%	No	2.69	0.1002	83.08	0.828
100%	Yes	2.87	0.0981	83.51	0.834
50%	Yes	2.93	0.1010	83.38	0.837
25%	Yes	2.81	0.1007	83.41	0.831
10%	Yes	3.10	0.1024	83.21	0.834
5%	Yes	2.81	0.0994	82.98	0.831
1%	Yes	2.70	0.1001	83.02	0.830

FFHQ Qualitative Results

Full dataset without data augmentation

FFHQ Qualitative Results

Full dataset with data augmentation

FFHQ Qualitative Results

1% subsampling without data augmentation

FFHQ Qualitative Results

1% subsampling without data augmentation

FFHQ TRAINING BEHAVIOUR

FULL DATASET

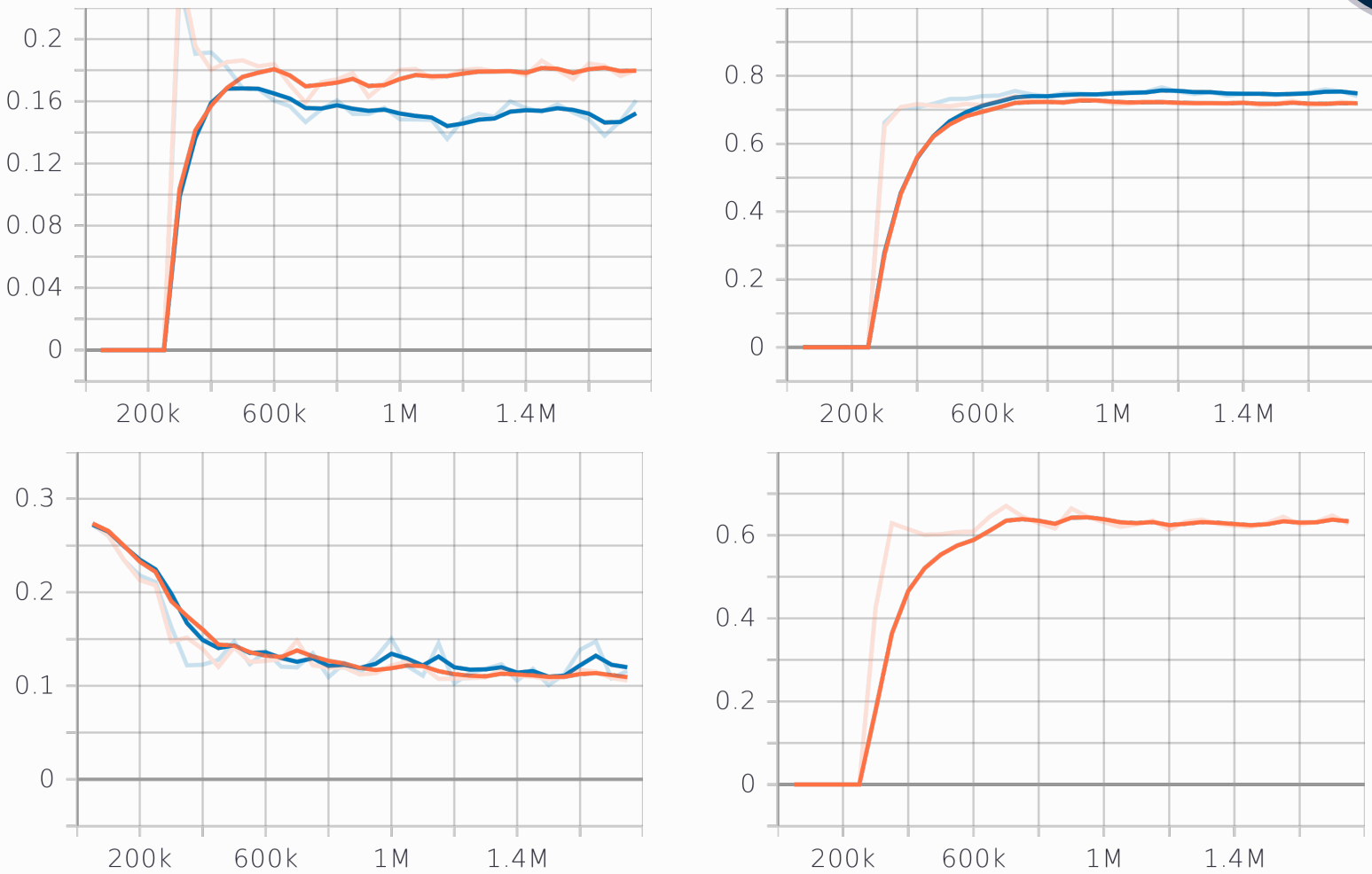


Figure: In order: average of fake images discriminator output, average of real images discriminator output, LPIPS, r_t on the full FFHQ dataset. In orange●: with augmentation. In blue●: without augmentation.

DIV2K EXPERIMENTAL SETUP

- Much more variegate dataset, thus a much harder task
- **3.2M** pre-training iterations and **2.4M** adversarial iterations
- Decreased p update amount to 5×10^{-4}
- Increased adversarial loss balancing factor to 6×10^{-5}
- Other hyperparameters have the same setup as FFHQ

DIV2K EXPERIMENTAL RESULTS

Dataset Percentage	Data Aug	Metric			
		FID	LPIPS	PSNR	SSIM
100%	**No**	66.25	0.1923	52.70	0.7503
50%	**No**	66.92	0.1893	52.61	0.7455
25%	**No**	67.08	0.2013	52.51	0.7428
10%	**No**	68.44	0.2024	52.40	0.7409
5%	**No**	68.35	0.2051	52.17	0.7296
1%	**No**	68.78	0.2265	50.36	0.6928
100%	**Yes**	66.44	0.1883	52.65	0.7434
50%	**Yes**	67.62	0.1926	52.67	0.7475
25%	**Yes**	67.17	0.1919	52.42	0.7423
10%	**Yes**	66.93	0.2011	52.30	0.7428
5%	**Yes**	67.14	0.2032	52.32	0.7432
1%	**Yes**	67.88	0.2101	51.03	0.7125

DIV2K Qualitative Results

Full dataset without data augmentation

DIV2K Qualitative Results

Full dataset with data augmentation

DIV2K Qualitative Results

1% subsampling without data augmentation

DIV2K Qualitative Results

1% subsampling with data augmentation

DIV2K Qualitative Results

Comparison

Figure: In order: full dataset without and with augmentation respectively, 1% dataset subsampling without and with data augmentation respectively

DIV2K TRAINING BEHAVIOUR

FULL DATASET

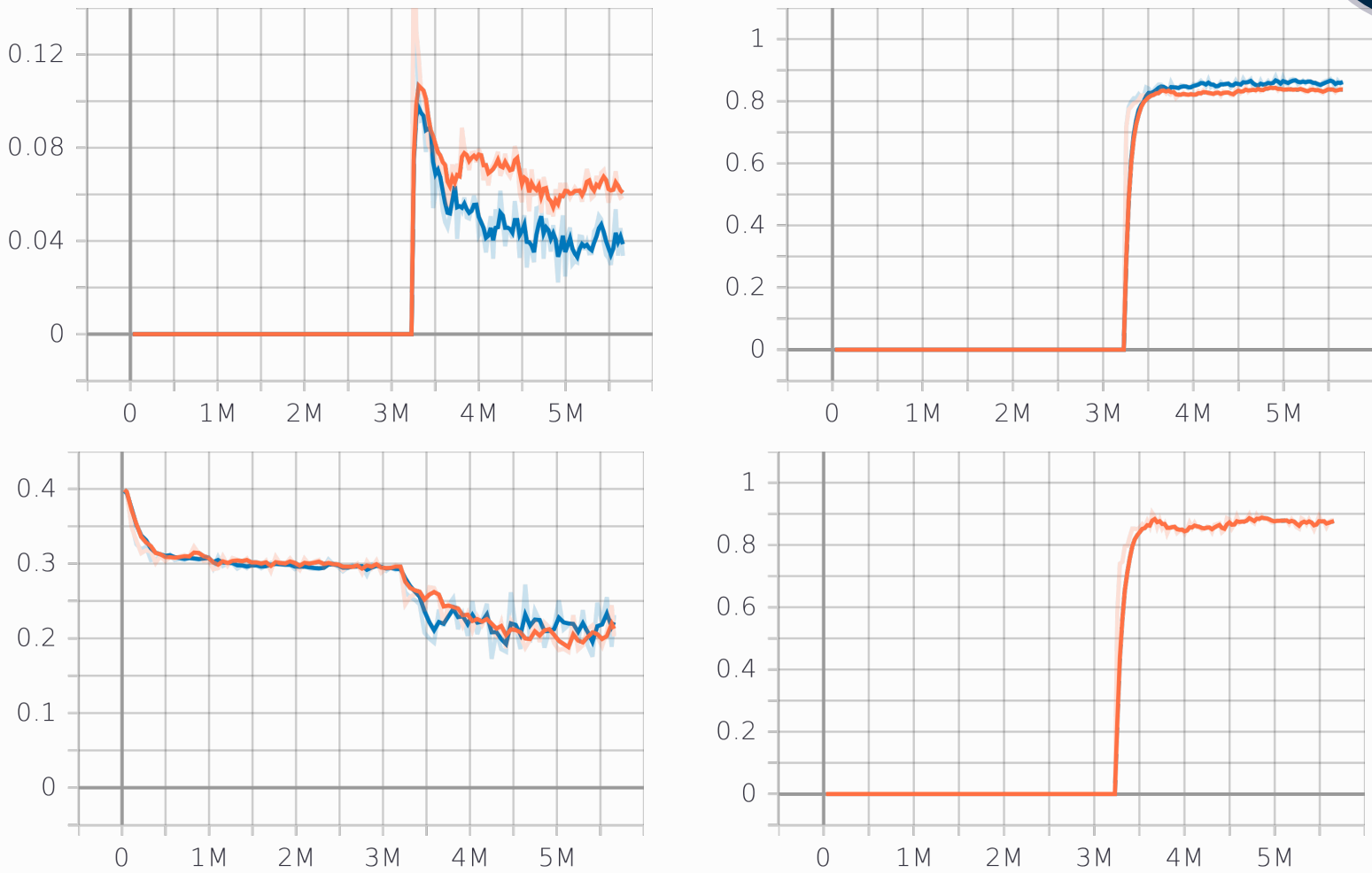


Figure: In order: average of fake images discriminator output, average of real images discriminator output, LPIPS, r_t on the full DIV2K dataset. In orange●: with augmentation. In blue●: without augmentation.

DIV2K TRAINING BEHAVIOUR

1% SUBSAMPLING

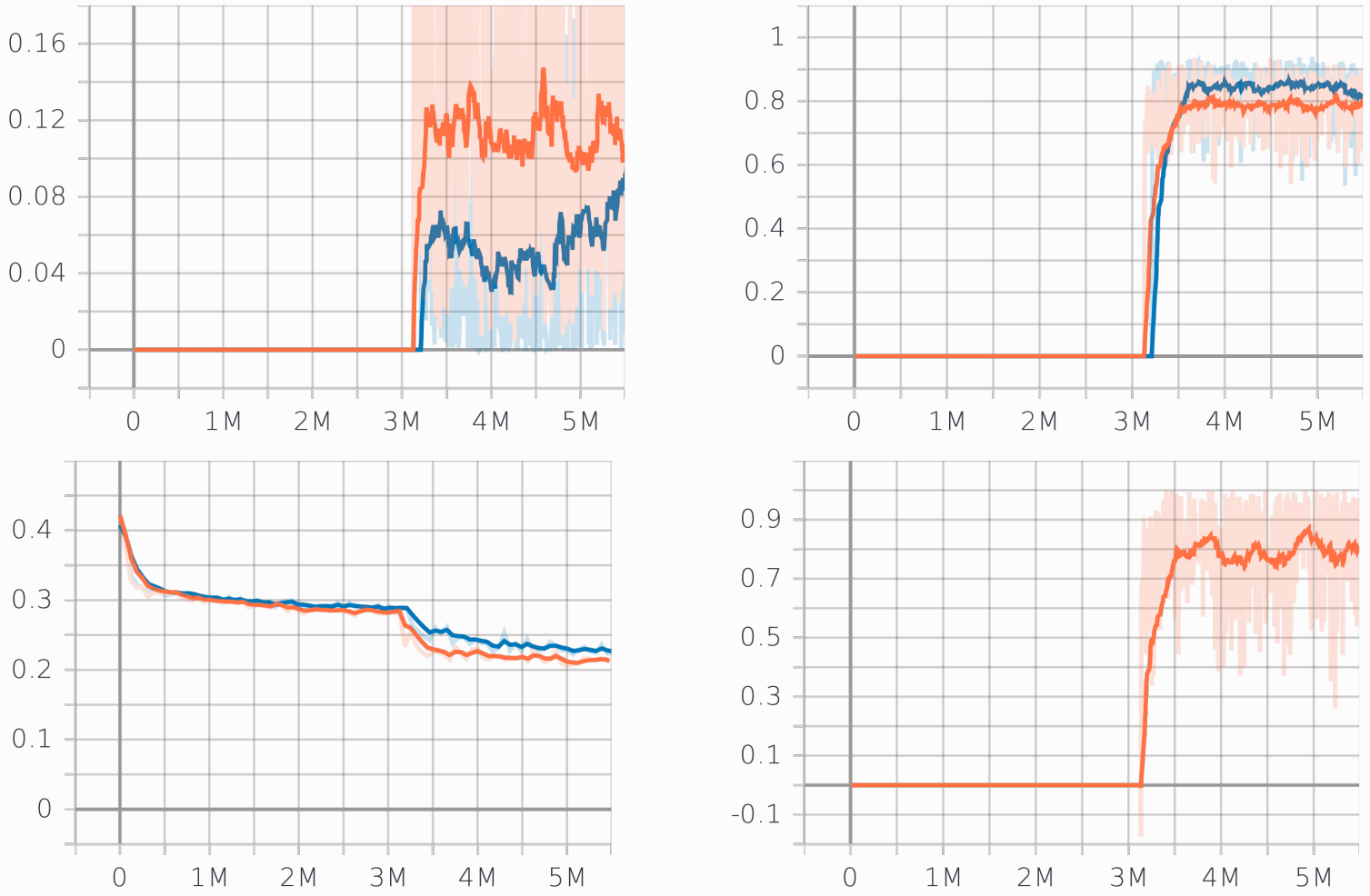


Figure: In order: average of fake images discriminator output, average of real images discriminator output, LPIPS, r_t on 1% of DIV2K dataset. In orange●: with augmentation. In blue●: without augmentation.

Conclusions

Conclusions

- We have shown that even in a super resolution setting Adaptive Discriminator Augmentation manages to stabilize training
- We have shown that in the context of super resolution modifications to the original augmentation pipeline are necessary in order not to introduce artifacts in generated images
- The effect of augmentation on model performance is marginal
- On very small datasets the performance boost given by data augmentation becomes more prominent


Code available at
https://gitlab.com/reddeadrecovery/superaugan

References I

- He, K., Zhang, X., Ren, S., and Sun, J. (2015).
  Deep residual learning for image recognition.
  CoRR, abs/1512.03385.

- Ignatov, A., Timofte, R., et al. (2019).
  Pirm challenge on perceptual image enhancement on smartphones: report.
  In *European Conference on Computer Vision (ECCV) Workshops*.

- Karras, T., Aittala, M., Hellsten, J., Laine, S., Lehtinen, J., and Aila, T. (2020).
  Training generative adversarial networks with limited data.

- Karras, T., Laine, S., and Aila, T. (2019).
  A style-based generator architecture for generative adversarial networks.

- Ledig, C., Theis, L., Huszar, F., Caballero, J., Cunningham, A., Acosta, A., Aitken, A., Tejani, A., Totz, J., Wang, Z., and Shi, W. (2017).
  Photo-realistic single image super-resolution using a generative adversarial network.

References II

- Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., and Chen, X. (2016).
  Improved techniques for training gans.

- Shi, W., Caballero, J., Huszár, F., Totz, J., Aitken, A. P., Bishop, R., Rueckert, D., and Wang, Z. (2016).
  Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network.

- Zhao, Z., Singh, S., Lee, H., Zhang, Z., Odena, A., and Zhang, H. (2020).
  Improved consistency regularization for gans.