# DeblurGAN_PRO

Nazarii Kuspys, Dmytro Lutchyn

# Briefly about starting point

Our team had some knowledge in the image inpainting area.

We were inspired by DeblurGAN-v2 ICCV 2019 presentation, as well as, tried these architectures for generating new data.

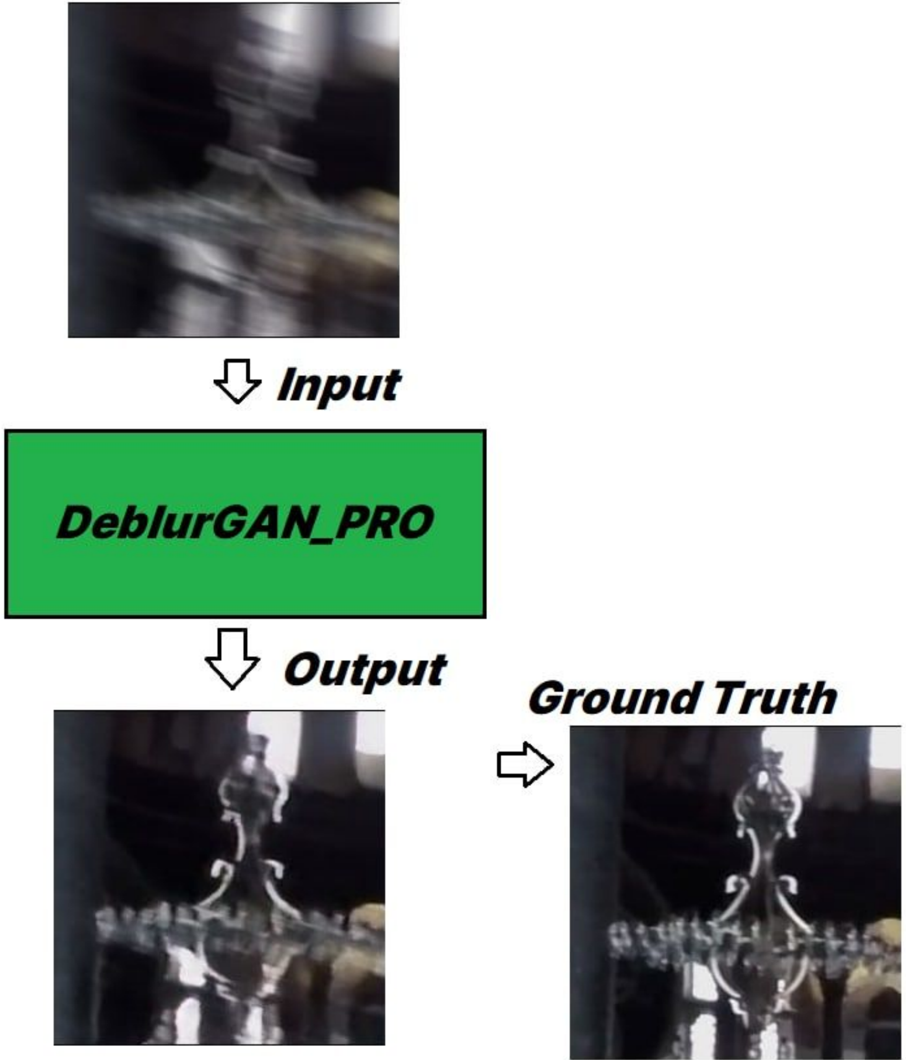
After such a hype around GAN architectures in 2019-2020, we tried to investigate why nobody used these architectures in the latest papers.

| Rank | Model | PSNR | SSIM | Extra Training Data | Paper | Code | Result | Year | Tags |
|---|---|---|---|---|---|---|---|---|---|
| 1 | **MPRNet-local** | 33.31 | 0.964 | × | Revisiting Global Statistics Aggregation for Improving Image Restoration | | | 2021 | |
| 2 | **DeepRFT+** | 33.23 | 0.963 | × | Deep Residual Fourier Transformation for Single Image Deblurring | | | 2021 | |
| 3 | **HINet-local** | 33.08 | 0.962 | × | Revisiting Global Statistics Aggregation for Improving Image Restoration | | | 2021 | |
| 4 | **Uformer-B** | 32.97 | 0.967 | × | Uformer: A General U-Shaped Transformer for Image Restoration | | | 2021 | Transformer |
| 5 | **Restormer** | 32.92 | 0.961 | × | Restormer: Efficient Transformer for High-Resolution Image Restoration | | | 2021 | Transformer |
| 6 | **MAXIM** | 32.86 | 0.961 | × | MAXIM: Multi-Axis MLP for Image Processing | | | 2022 | MLP |
| 7 | **HINet** | 32.71 | | × | HINet: Half Instance Normalization Network for Image Restoration | | | 2021 | |
| 8 | **MIMO-UNet++** | 32.68 | 0.959 | × | Rethinking Coarse-to-Fine Approach in Single Image Deblurring | | | 2021 | |
| 9 | **MPRNet** | 32.66 | 0.959 | × | Multi-Stage Progressive Image Restoration | | | 2021 | |
| 10 | **BANet** | 32.44 | 0.957 | × | BANet: Blur-aware Attention Networks for Dynamic Scene Deblurring | | | 2021 | |
| 11 | **MB2D** | 32.16 | 0.953 | ✓ | Blur More To Deblur Better: Multi-Blur2Deblur For Efficient Video Deblurring | | | 2020 | |
| 12 | **RADNet** | 32.15 | 0.9560 | ✓ | Region-Adaptive Dense Network for Efficient Motion Deblurring | | | 2020 | |
| 13 | **SAPHNet** | 32.02 | 0.953 | ✓ | Spatially-Attentive Patch-Hierarchical Network for Adaptive Motion Deblurring | | | 2020 | |

**From paperswithcode**

# What is deblurring, using neural net?

How neural network gets rid of motion blur in the images (example)



# GoPro dataset

GoPro -- (synthetic) dataset for deblurring consists of 3,214 blurred images with the size of 1,280×720 that are divided into 2,103 training images and 1,111 test images. The dataset consists of pairs of a realistic blurry image and the corresponding ground truth sharp image that are obtained by a high-speed camera. Blurred image is generated using 240 FPS videos via GOPRO4 Hero Black camera by averaging different number (7-13) of the successive frames, so the strength of blur does not follow the same pattern.

Seungjun Nah

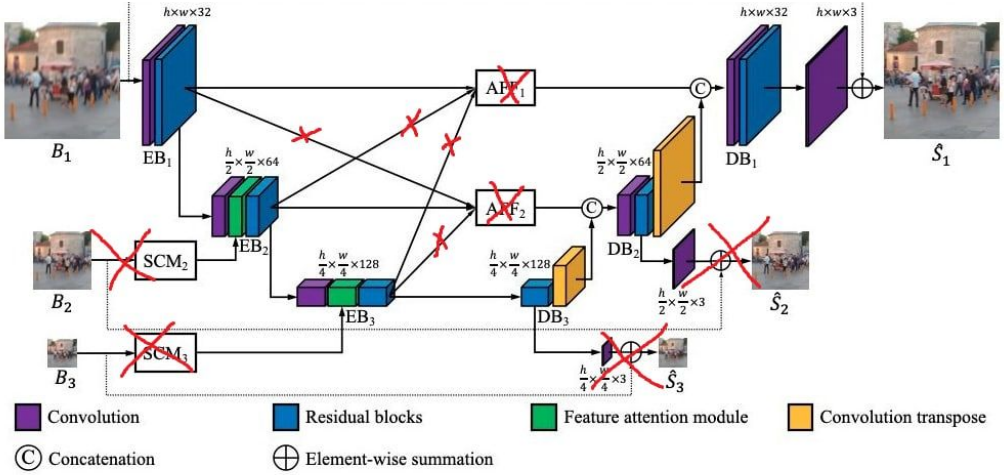
# What have we already done?

- Implemented training pipeline: based on pytorch lightning
- Added hydra configuration support to achieve scalability of project
- Added support for 3 different GAN architectures: Basic GAN (from SRGAN paper), Wasserstein GAN with GP, Relativistic GAN (from RAGAN_ls paper)
- Added two generator architectures: feed forward generator from SRGAN and MIMO-Unet_Pure

# Briefly about Generator - MIMO-Unet Pure

As our main idea is to compare GAN architectures, as well as, find out different pros and cons of using them, we wanted to make the architecture inference fast as possible, because we were time limited and did not have lots of GPUs for training.

Also, there are some comparison of usefulness of components in the paper, so that is why we deleted FAM, SCM, and nested skip connections. It helped a lot with the inference time.

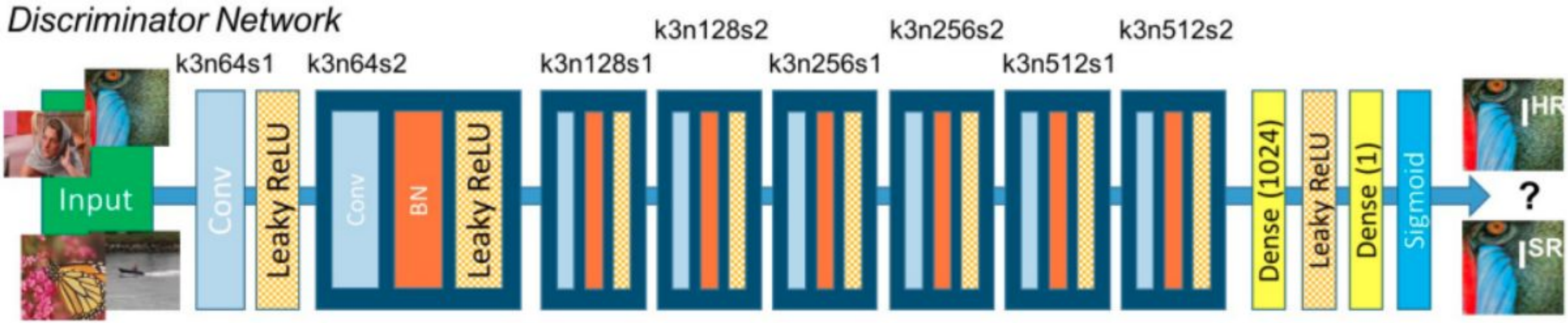



| MISE | MOSD | AFF | MSFR | PSNR | Params. |
|---|---|---|---|---|---|
| | | | | 31.16 | 6.46 |
| ✓ | | | | 31.17 | 6.72 |
| | ✓ | | | 31.33 | 6.47 |
| | | ✓ | | 31.33 | 6.54 |
| ✓ | ✓ | | | 31.38 | 6.73 |
| | ✓ | ✓ | | 31.38 | 6.54 |
| ✓ | | ✓ | | 31.39 | 6.80 |
| ✓ | ✓ | ✓ | | 31.46 | 6.81 |
| ✓ | ✓ | ✓ | ✓ | **31.73** | 6.81 |

Table 4. Effectiveness of different components of MIMO-UNet on the GoPro test dataset.

# Discriminator - N layers

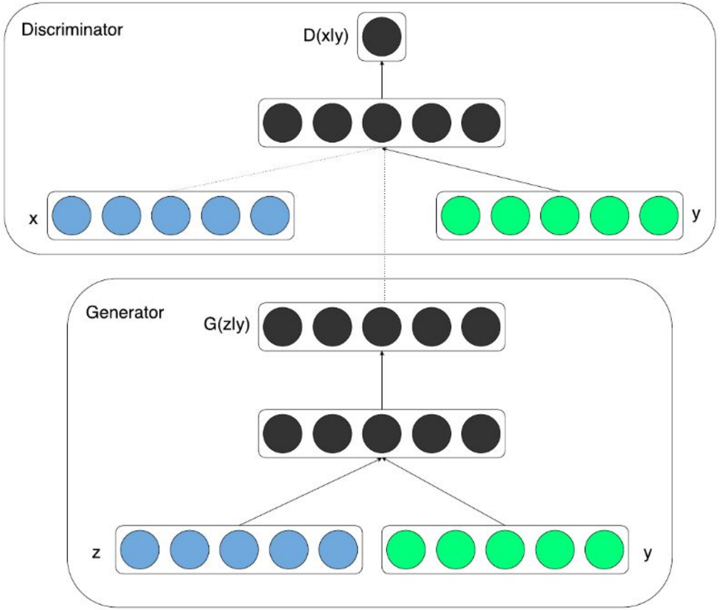
It is a basic feed forward net that consists of N blocks that are similar to residual blocks. Neural net makes from the image feature map. Such architectures are widely used for discriminators



# Briefly about Adversarial training (conditional GANs)

In the **generator** the prior input noise p(z) and y are combined in joint hidden representation, and the adversarial training framework allows for considerable flexibility in how this hidden representation is composed.

In the **discriminator** x and y are presented as inputs to a discriminative function (the condition has to be presented in some hidden representation).




The objective function becomes:

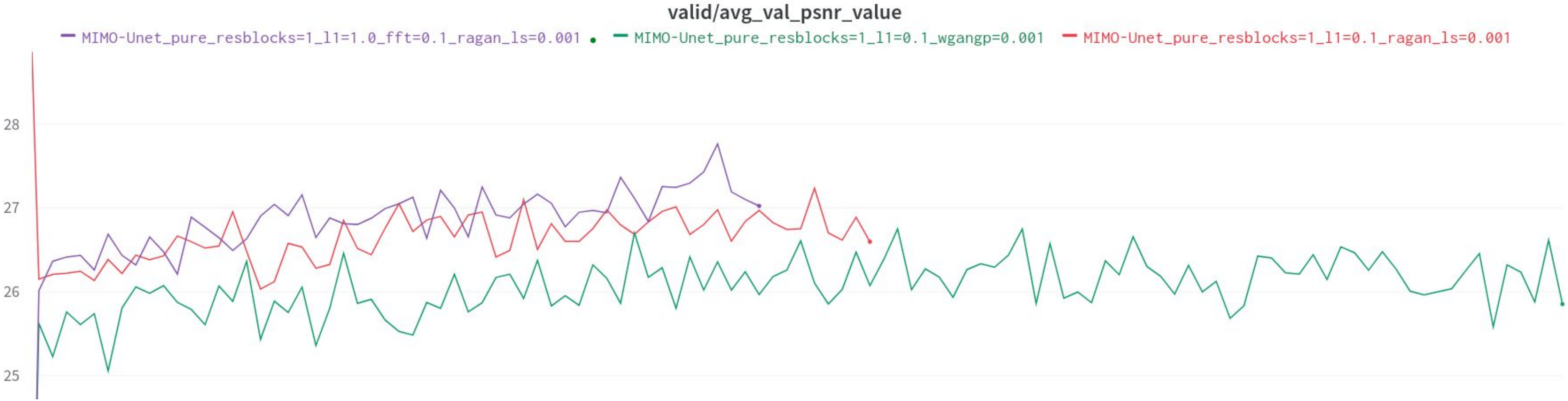
$$\min_G \max_D V_{GAN}(D,G) = \mathbb{E}_{x\sim p_{data}(x|y)}[\log D(x,y)] + \mathbb{E}_{z\sim p_z(z)}[\log(1 - D(G(z|y),y))]$$

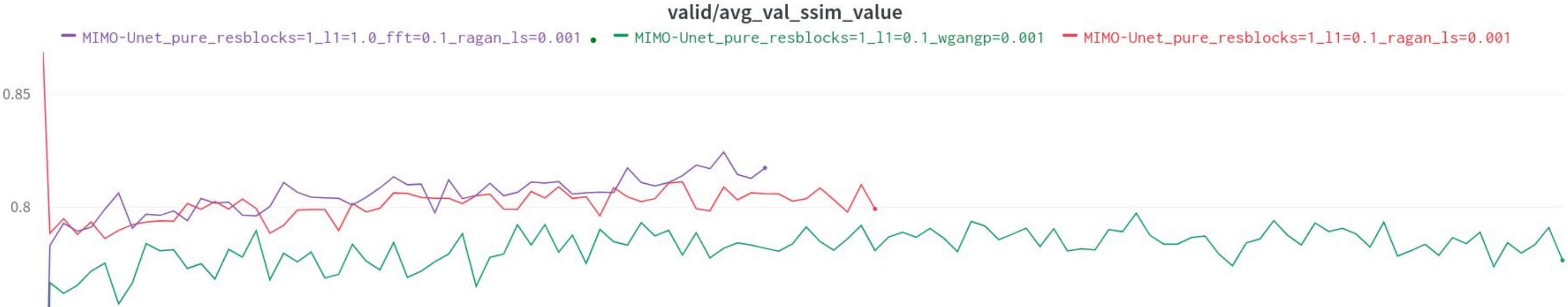
# Basically about WGAN-GP and Ragan_ls

Both are losses that are used for training GANs.

- Main idea of WGAN-GP: we can call the discriminator - a critic, and make him give higher scores for real images than for generated images. (GP - term that helps for KL-divergence)
- Main idea of Ragan_ls: to make a feedback on the current training step, discriminator finds the average of previous scores, and computes the relativistic loss function - estimate the probability that the given real data is more realistic than a randomly sampled fake data by using a discriminator

# Achieved results



# Achieved results

# Conclusions

1. Training GANs is not an easy task :) (it slows training x3 and it is hard to control overperforming one of the players by other)
2. For precise deblur, it is better to use fft loss (that was showed by the Google Research Group in 2021 for image inpainting tasks), as it gives better feedback for generator.
3. There were a theoretical hypothesis that GANs can reconstruct areas, that have patterns which did not appear before in the training, but, unfortunately, we could not prove that fact in practice.