Spatially-Adaptive Pixelwise Networks for Fast Image Translation (Machine Learning 2022 Course)

Savinov Maxim [1] **Savinova Maria** [1] **Fominykh Anna** [1] **Shukhratov Islomjon** [1] **Grigory Babkin** [1]

Abstract

We investigate the performance of several models including ASAP-Net and pix2pixHD in segmentation task. Further, we estimate the performance of DeblurGAN. We propose its modification (DeblurGAN based on ASAP-Net) in debluring tasks in the motion blur conditions. The evaluation showed that the modified scheme presents the worst performance.

Github repo: github repository link
Video presentation: google drive video link

1. Introduction

This work is on blind motion deblurring of a single photograph. Significant progress has been recently achieved in related areas of image super-resolution [20] and inpainting [45] by applying generative adversarial networks (GANs) [10]. 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 [20] and image-to-image translation by generative adversarial networks [16], we treat deblurring as a special case of such image-to-image translation.

The task for the paper includes reproducing the key results of the papers for segmentation problem and to research the possibility of the usage such networks (ASAP-Net (Wang et al., 2018a), DeblurGAN (Kupyn et al., 2018), pix2pixHD (Wang et al., 2018b)) for other image-to-image translation task: debluring problem. The work plan was as follows:

- Download and prepare Cityscapes 512×1024 dataset. Prepare generator and discriminator networks for training with parameters that are described in the paper (Shaham et al., 2021).
- Train proposed model for segmentation task on prepared training dataset from scratch. Compute Frechet Inception Distance (FID). Compare results with values obtained by the authors in the paper. Report inference time and specification for the used GPUs.
- Train pix2pixHD model for the same data and report the same results as for the model of authors. Determine whether ASAP-Net is faster than pix2pixHD model.
- Investigate the model performance in deblurring tasks and research the possible application fo ASAP-Net scheme in deblurring task.

2. Related Work

Image-to-image translation using convolutional neural networks. Many challenges in image processing, computer graphics, and computer vision can be framed as the problem of "translating" an input mage representation into a corresponding output image, i.e. the problem of translating one possible representation of a scene into another, given enough training data. Convolutional neural nets (CNNs) have become the common engine driving a wide variety of image prediction tasks, and the community has already taken substantial steps in this regard. Although CNNs learn to minimize a loss function - an objective that assesses the quality of results — a lot of manual effort goes into constructing successful losses, even though the learning process is automated. If we use a basic technique and ask the CNN to minimize the Euclidean distance between predicted and ground truth pixels, the results will be skewed. This is due to the fact that Euclidean distance is reduced by averaging all feasible outputs, resulting in blurring.

Generative Adversial Networkds suggested a solution to this problem (GANs) (Goodfellow et al., 2014). GANs learn a loss that attempts to classify whether the output image is real or false while also training a generative model to reduce the loss. Images that are blurry or appear to be false will not

[1]Skolkovo Institute of Science and Technology, Moscow, Russia. Correspondence to: Savinov Maxim <Maxim.Savinov@skoltech.ru>, Savinova Maria <Maria.Savinova@skoltech.ru>, Fominykh Anna <Anna.Fominykh@skoltech.ru>, Shukhratov Islomjon <Islomjon.Shukhratov@skoltech.ru>, Grigory Babkin <Grigory.Babkin@skoltech.ru>.

Final Projects of the Machine Learning 2020 Course, Skoltech, Moscow, Russian Federation, 2020.

be accepted. GANs can be used for a variety of tasks that would normally require very diverse types of loss functions because they train a loss that adjusts to the data.

A conditional Generative Adversarial Network (cGAN) is commonly trained to learn a direct mapping from one domain to the other in current techniques (Goodfellow et al., 2014). Although these methods have achieved considerable progress in terms of visual quality, their model size and inference time have also increased dramatically. When operating on high resolution images, which is the most ideal setting in typical real-world applications, the computing expense of these techniques becomes even more evident.

Generative adversarial networks. A generator and a discriminator are the two pieces of a GAN. The generator attempts to map to the target image, while the discriminator attempts to distinguish between the generator and the real target images. The generator's purpose is to deceive the discriminator into thinking the generated and target images are the same. In the instance of deblurring, we condition the generator by feeding it the blurry picture instead of random noise, from which it attempts to construct a sharp image to trick the discriminator.

Debluring. Deblurring is the task of restoring a blurred image to a sharp one, retrieving the information lost due to the blur. Given a blurred image B, our goal is to predict a sharp image S, such that

$$B = S * K + N,$$

where K is the blur kernel, N is the noise, and $*$ denotes convolution.

Object motion, camera blur, and out-of-focus can all create blurring. Deblurring is a type of image translation in which the goal is to obtain a sharp image from a blurred one. GANs have demonstrated good performance in a variety of image-to-image translation tasks, including super resolution, image inpainting, and deblurring. Learning-based deblurring approaches may be divided into two types: one in which the blur kernel is estimated (Sun et al., 2015; Isola et al., 2017; Tao et al., 2018; Zhang et al., 2018), and the other in which the sharp picture is generated in an end-to-end fashion (Kupyn et al., 2018; Noroozi et al., 2017; Ramakrishnan et al., 2017). GANs have mostly been used to produce images in an end-to-end manner. Deblurring issues are separated into two categories: blind and non-blind deblurring. The premise behind non-blind debluring is that the blur kernel K is known. The technique in this scenario is to do the deconvolution procedure and produce an estimate of S. Blind deblurring algorithms estimate both latent sharp image S and blur kernels K when the blur function is unknown.

Table 1. Model performance comparison (I)

	ASAP-NET	PIX2PIXHD
FID	225.77	217.24
PSNR	15.46	16.42
SSIM	0.45	0.52

Table 2. Model performance comparison (II)

	DEBLURGAN	DEBLURGAN BASED ON ASAP-NET
FID	8.6	66.21
PSNR	34.5	21.65
SSIM	0.93	0.70

3. Dataset Description

The Cityscapes dataset (Cordts et al., 2016) is used, which is a big dataset for semantic interpretation of urban street scenes from over 50 cities over several months in good/moderate weather. There are 19 training classes. There are 5k finely annotated photos in this dataset (2975 train, 500 validation, and 1525 test images) and 20k coarsely annotated images in this dataset. To train our network for the segmentation challenge, we only employ fine labeled photos with ten item categories.

4. Machine Learning methods

We consider image-to-image translation task, in which a neural network is trained to implement a mapping between two image domains. That is, given a input image $x \in \mathbb{R}^{H \times W \times C}$ from the source domain, the output $y \in \mathbb{R}^{H \times W \times 3}$ should look like it belongs to the target domain. We now give a brief description of used models.

pix2pixHD (Wang et al., 2018b) is based on pix2pix architecture (Isola et al., 2017) learns a function to map from an input image to an output image using a conditional generative adversarial network (cGAN). The Conditional GAN, or cGAN, is a GAN architecture addition that allows for control over the image that is created, such as generating an image of a specific class. pix2pixHD GAN is a cGAN implementation in which the production of one picture is conditional on the presence of another image. The Generator and the Discriminator are the two fundamental components of the network. Convolution-BatchNormalization-ReLU blocks of layers are used in both the generator and discriminator models, as is customary for deep convolutional neural networks. To create the output image, the Generator changes the input image.

The fundamental component of the efficient runtime of ASAP-Net (Wang et al., 2018a) is a novel pixelwise genera-

Figure 1. The performance results for ASAP-Net and pix2pixHD networks

tor: Each pixel is treated separately from the rest, utilizing a pixel-specific, lightweight algorithm. Multi-Layer Perceptron is a type of perceptron that has multiple layers. First, unlike standard convolutional networks, which share network parameters across spatial positions, the MLPs' parameters vary geographically, essentially transforming each pixel with a new function. Second, a convolutional network that analyzes a substantially downsampled representation of the input predicts spatially changing parameters at low resolution. As a result, the MLPs are adaptable to the input image (i.e., the pixel-functions depend on the input image itself). Third, the local MLPs, in addition to the input pixel values. The model generates a tensor of weights and biases by first processing the input at a very low resolution. These are upsampled to full resolution, where they are used to parameterize pixelwise, spatially variable MLPs fp that compute the final output from the high-resolution input.

DeblurGAN is a Conditional Adversarial Network that is optimized utilizing a multi-component loss function (Kupyn et al., 2018). Both in terms of structural similarity and visual appearance, DeblurGAN provides best-in-class results. The deblurring model's quality is also assessed in an innovative approach on a real-world problem: object detection on (de-)blurred photos.

5. Experiments

We validate the performance of our method on segmentation and deblurring tasks and compare runtime and image quality (quantitatively and qualitatively) with previous work (Shaham et al., 2021).

We compare three competitive models and a single newly proposed model: (i) ASAP-Net, (ii) pix2pixHD, (iii) DeblurGAN, and (iv) DeblurGAN based on ASAP-Net architecture over 300 epochs. First and second models are evaluated in the segmentation task and the performance of last two models are checked in the conditions of motion blur. The last model has the structure of the DeblurGAN model, but the generator and discriminator are taken from the ASAP-Net network. DeblurGAN and DeblurGAN based on ASAP-Net were trained on set with random motion blur whose strength varied from low almost absent motion blur to highly blurred image. The models are compared in terms of Frechet Inception Distance (FID, which measures the similarity between the distributions of real and generated images) , peak signal-to-noise ratio (PSNR, represents the ratio between the maximum possible value (power) of a signal and the power of distorting noise that affects the quality of its representation), and structural similarity index measure (SSIM, measures distortions as a combination of three factors: loss of correlation, luminance distortion and contrast distortion). Also, the inference time and specifications for the used GPUs are presented.

Figure 2. DeblurGAN based on ASAP-Net loss

Speed. We benchmark the inference time of all models on an Nvidia GeForce 2080ti GPU. The speed comparison for ASAP-Net and pix2pixHD is presented in Fig. 1. It may be noticed that ASAP-Net works almost three times faster compared to pix2pixHD. The working time for DeblurGAN and DeblurGAN based on ASAP-Net are given in Fig. 2. The modified model is two times faster compared to DeblurGAN. This observation follows from the previous conclusion in Fig. 1, where we noted that ASAP-Net showed the best time.

Quantitative evaluation. The FID, PSNR and SSIM results for ASAP-Net and pix2pixHD are presented in Fig. 3 and Fig. 4. The summary is presented in Table 1. The FID values for ASAP-Net and pix2pixHD are nearly on the same level. PSNR and SSIM of the pix2pixHD model are slightly higher than of ASAP-Net. The FID, PSNR and SSIM results for DeblurGAN and DeblurGAN based on ASAP-Net are presented in Fig. 5 and Fig. 6. The summary is presented in Table 2. The FID values for DeblurGAN are significantly lower and PSNR is higher, which means that the quality should be better. The SSIM values for DeblurGAN tend to be one, which means that the performance of the model is high, whereas the SSIM value for the combined scheme gets lower over the epochs.

Qualitative evaluation. The qualitative evaluation may be done based on the resultant pictures presented in Fig. 1 and Fig. 2. The first two models show comparable results which may slightly vary from image to image. Both models show results close to ground truth and it is difficult to distinguish between the models. A different situation is observed for DeblurGAN and DeblurGAN based on ASAP-Net. DeblurGAN shows excellent results for every motion blur strength. However, the combined scheme leads to darkened blurred images, with a black spot in the same place.

6. Conclusion

Performance for four models (ASAP-Net, DeblurGAN, pix2pixHD, and DeblurGAN based on ASAP-Net) were evaluated for the deblurring task. ASAP-Net shows the fastest inference time, and its results are comparable to the pix2pixHD model. DeblurGAN and DeblurGAN based on ASAP-Net models were trained on images with motion blur. DeblurGAN showed the best performance in each evaluation. Although the working time of DeblurGAN based on ASAP-Net is faster comparing the original DeblurGAN, the quantitative and qualitative results are poor. The images are still blurred, brightness gets lower and an artifact appears on every image. It may be seen that is not reasonable to use the combined model, since it shows the worse results.

References

Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., and Schiele,

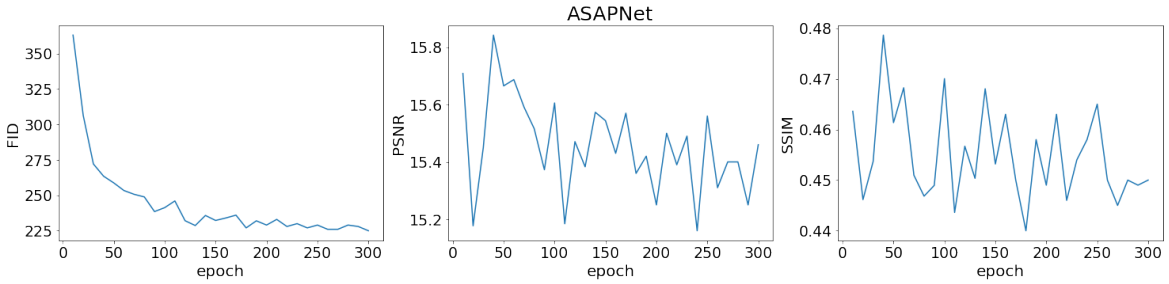

Figure 3. ASAP-Net

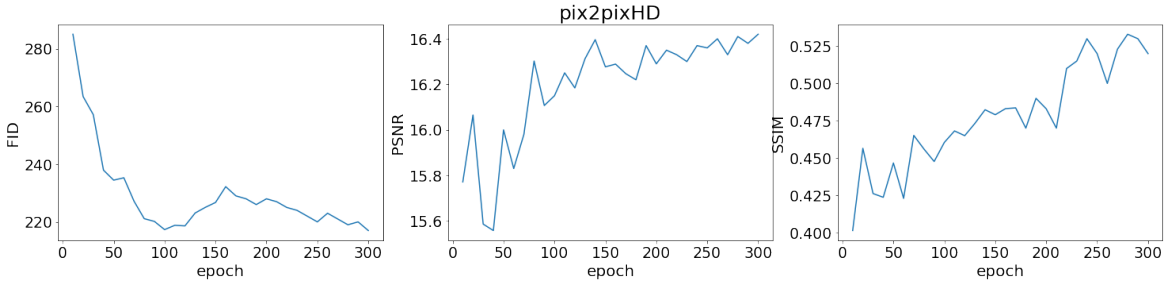

Figure 4. pix2pixHD

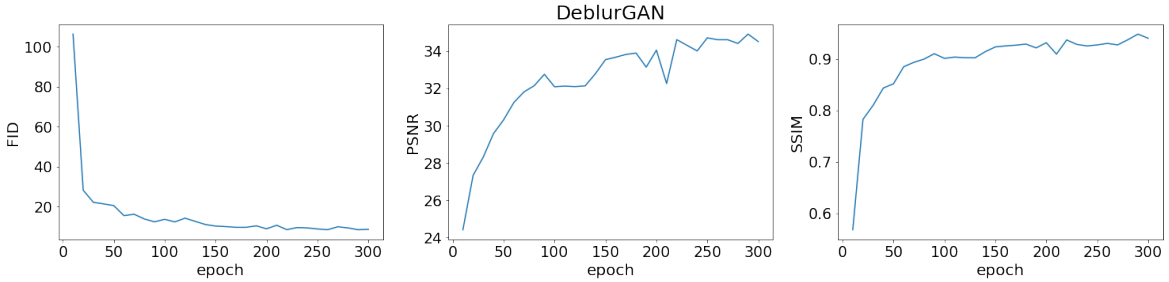

Figure 5. DebluGAN

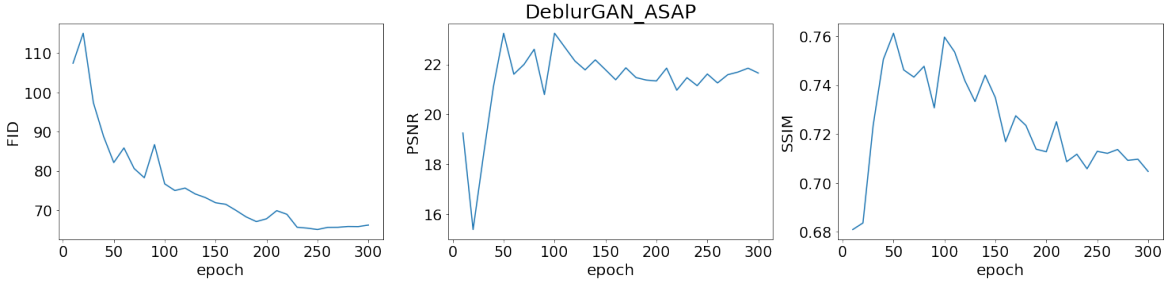

Figure 6. DeblurGAN based on ASAP-Net

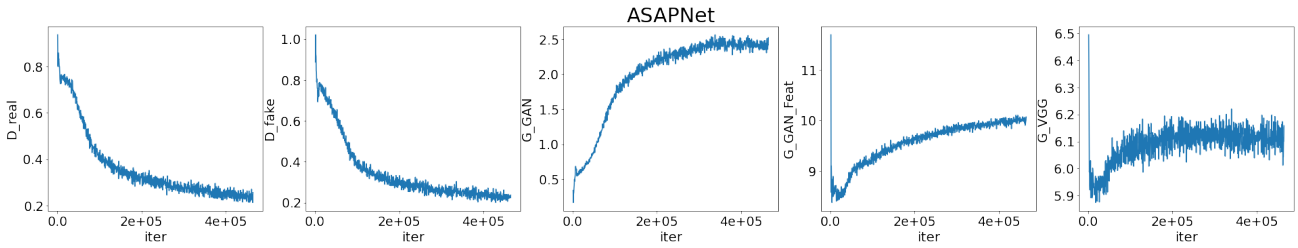

Figure 7. ASAP-Net loss

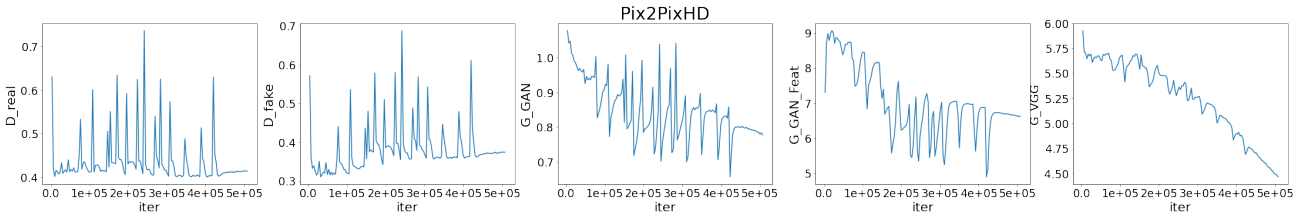

Figure 8. pix2pixHD loss

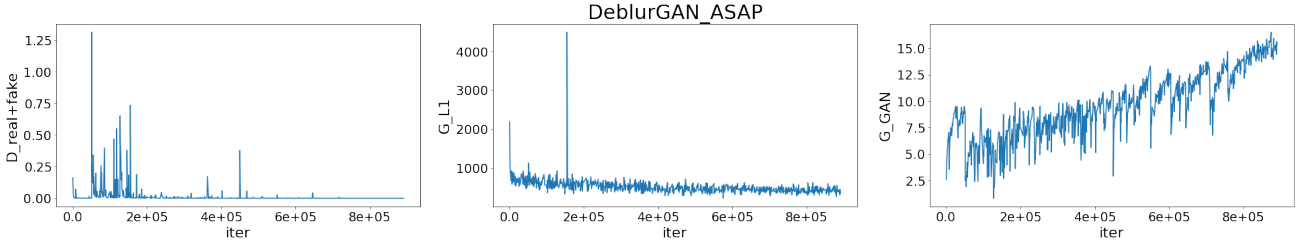

Figure 9. DeblurGAN loss

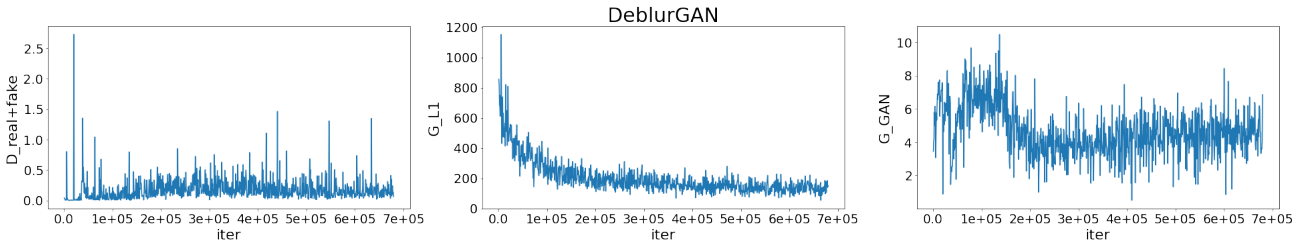

Figure 10. DeblurGAN based on ASAP-Net loss

B. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3213–3223, 2016.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

Isola, P., Zhu, J.-Y., Zhou, T., and Efros, A. A. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1125–1134, 2017.

Kupyn, O., Budzan, V., Mykhailych, M., Mishkin, D., and Matas, J. Deblurgan: Blind motion deblurring using conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 8183–8192, 2018.

Noroozi, M., Chandramouli, P., and Favaro, P. Motion deblurring in the wild. In *German conference on pattern recognition*, pp. 65–77. Springer, 2017.

Ramakrishnan, S., Pachori, S., Gangopadhyay, A., and Raman, S. Deep generative filter for motion deblurring. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pp. 2993–3000, 2017.

Shaham, T. R., Gharbi, M., Zhang, R., Shechtman, E., and Michaeli, T. Spatially-adaptive pixelwise networks for fast image translation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14882–14891, 2021.

Sun, J., Cao, W., Xu, Z., and Ponce, J. Learning a convolutional neural network for non-uniform motion blur removal. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 769–777, 2015.

Tao, X., Gao, H., Shen, X., Wang, J., and Jia, J. Scale-recurrent network for deep image deblurring. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 8174–8182, 2018.

Wang, T.-C., Liu, M.-Y., Zhu, J.-Y., Tao, A., Kautz, J., and Catanzaro, B. High-resolution image synthesis and semantic manipulation with conditional gans. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 8798–8807, 2018a.

Wang, T.-C., Liu, M.-Y., Zhu, J.-Y., Tao, A., Kautz, J., and Catanzaro, B. High-resolution image synthesis and semantic manipulation with conditional gans. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8798–8807, 2018b.

Zhang, J., Pan, J., Ren, J., Song, Y., Bao, L., Lau, R. W., and Yang, M.-H. Dynamic scene deblurring using spatially variant recurrent neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2521–2529, 2018.

A. Team member's contributions

Explicitly stated contributions of each team member to the final project.

Maxim Savinov (20% of work)

- Rewriting the ASAP-Net code
- Experimenting with model parameters on Cityscapes dataset
- Preparing the Section 5 of this report
- Setting up experiments on GPUs

Islomjon Shukhratov (20% of work)

- Preparing DebglurGan network for training
- Preparing the presentation
- Recording video presentation
- Preparing the GitHub Repo

Anna Fominykh (20% of work)

- Reviewing literate on the topic (2 papers)
- Working on pix2pixHD model and adjusting experimental parameters
- Preparing the Sections 3, 4 of this report

Grigory Babkin (20% of work)

- Preparing the dataset
- Preparing the Section 1 of the report
- Formatting images for article

Maria Savinova (20% of work)

- Reviewing literate on the topic (2 papers)
- Rewriting the DeblurGAN to embed the ASAP-Net modules
- Preparing Section 2 of the report

B. Reproducibility checklist

Answer the questions of following reproducibility checklist. If necessary, you may leave a comment.

1. A ready code was used in this project, e.g. for replication project the code from the corresponding paper was used.

   ☑ Yes.
   ☐ No.
   ☐ Not applicable.

   General comment: If the answer is **yes**, students must explicitly clarify to which extent (e.g. which percentage of your code did you write on your own?) and which code was used.

   Students' comment: The network code was taken from the github repository, however, the huge changes were required to make the code ease in running.

2. A clear description of the mathematical setting, algorithm, and/or model is included in the report.

   ☑ Yes.
   ☐ No.
   ☐ Not applicable.

   Students' comment: The article includes the description of observed models (namely, ASAP-Net, pix2pixHD, DeblurGAN).

3. A link to a downloadable source code, with specification of all dependencies, including external libraries is included in the report.

   ☑ Yes.
   ☐ No.
   ☐ Not applicable.

   Students' comment: The link to a github repository is attached.

4. A complete description of the data collection process, including sample size, is included in the report.

   ☑ Yes.
   ☐ No.
   ☐ Not applicable.

   Students' comment: The sample size and data collection process is described in the inproduction part and in the dataset section.

5. A link to a downloadable version of the dataset or simulation environment is included in the report.

   ☑ Yes.
   ☐ No.
   ☐ Not applicable.

   Students' comment: The link is in the description of the github repository.

6. An explanation of any data that were excluded, description of any pre-processing step are included in the report.

   ☑ Yes.
   ☐ No.
   ☐ Not applicable.

   Students' comment: We thoroughly describe the process of image bluring and algorithm combination.

7. An explanation of how samples were allocated for training, validation and testing is included in the report.

   ☐ Yes.
   ☐ No.
   ☑ Not applicable.

   Students' comment: We wasn't able to choose the sizes of sets by ourselves since the original paper states the exact fractions. These settings have been taken from the original paper.

8. The range of hyper-parameters considered, method to select the best hyper-parameter configuration, and specification of all hyper-parameters used to generate results are included in the report.

   ☐ Yes.
   ☐ No.
   ☑ Not applicable.

   Students' comment: The parameters have been taken from the original paper in order to be able to reproduce the results.

9. The exact number of evaluation runs is included.

   ☑ Yes.
   ☐ No.
   ☐ Not applicable.

   Students' comment: The number of epochs is mentioned in section 5.

10. A description of how experiments have been conducted is included.

    ☑ Yes.
    ☐ No.
    ☐ Not applicable.

    Students' comment: A description of how experiments is included in section 5.

11. A clear definition of the specific measure or statistics used to report results is included in the report.

    - [x] Yes.
    - [ ] No.
    - [ ] Not applicable.

    Students' comment: In the report FID, PSNR and MMIS metrics were used. They are described in the section 5.

12. Clearly defined error bars are included in the report.

    - [ ] Yes.
    - [ ] No.
    - [x] Not applicable.

    Students' comment:

13. A description of the computing infrastructure used is included in the report.

    - [x] Yes.
    - [ ] No.
    - [ ] Not applicable.

    Students' comment: The used GPU configuration is mentioned in section 5.