





# Improving Adversarial Learning with Image Quality Measures for Image Deblurring

Jingwen Su<sup>(✉)</sup>  and Hujun Yin<sup>(✉)</sup> 

Department of Electrical and Electronic Engineering, The University of Manchester,  
Manchester M13 9PL, UK

{jingwen.su,hujun.yin}@manchester.ac.uk

**Abstract.** Generative adversarial networks (GANs) have become popular and powerful models for solving a wide range of image processing problems. We introduce a novel component based on image quality measures in the objective function of GANs for solving image deblurring problems. Such additional constraints can regularise the training and improve the performance. Experimental results demonstrate marked improvements on generated or restored image quality both quantitatively and visually. Boosted model performances are observed and testified on three test sets with four image quality measures. It shows that image quality measures are additional flexible, effective and efficient loss components to be adopted in the objective function of GANs.

**Keywords:** Generative adversarial networks · Image deblurring · Image quality measures

## 1 Introduction

Recently, deep neural networks with adversarial learning have become a prevalent technique in generative image modelling and have made remarkable advances. In topics such as image super-resolution, in-painting, synthesis, and image-to-image translation, there are already numerous adversarial learning based methods demonstrating the prominent effectiveness of GANs in generating realistic, plausible and conceptually convincing images [8, 14, 21, 25–27, 29].

In this paper we address image enhancement problems such as blind single image deblurring by casting them as a special case for image-to-image translation, under the adversarial learning framework. A straightforward way for realising quality improvement in image restoration is to involve image quality measure as constraints for training GANs. As it is known, the objective function of GANs defines gradient scale and direction for network optimization. Adversarial loss in GANs, an indispensable component, is the foundation that encourages the generation of images to be as realistic as possible. However, details and textures in the generated images are unable to be fully recovered and they are critical for the

human visual system to perceive image quality. Thus, image quality measures that compensate the overlooked perceptual features in images are necessary to take a part in guiding gradient optimization during the training.

An image quality based loss is proposed and added to the objective function of GANs. There are three common quality measures that can be adopted. We investigate their effects on generated/restored image quality, compared with the baseline model without any quality loss. The rest of this paper is structured as follows. Section 2 describes related work. Section 3 introduces the proposed method, followed by experimental settings, results and discussion in Sect. 4. Section 5 concludes the findings and suggests possible future work.

## 2 Related Work

### 2.1 Generative Adversarial Networks

GAN consists of a generative model and a discriminative model. These two models are trained simultaneously by the means of adversarial learning, a process that can significantly contribute to improving the generation performance. Adversarial learning encourages competition between the generator and the discriminator. The generator is trained to generate better fake samples to fool the discriminator until they are indistinguishable from real samples.

For a standard GAN (a.k.a the vanilla GAN) proposed by Goodfellow et al. [6], the generator  $G$  receives noise as the input and generates fake samples from model distribution  $p_g$ , the discriminator  $D$  classifies whether the input data is real. There are a great number of variants of GAN proposed afterwards, such as conditional GAN (cGAN) [17], least squares GAN (LSGAN) [16], Wasserstein GAN (WGAN) [1], and Wasserstein GAN with gradient penalty (WGAN-GP) [7].

### 2.2 Image Deblurring

Image deblurring has been a perennial and challenging problem in image processing and its aim is to recover clean and sharp images from degraded observations. Recovery process often utilises image statistics and prior knowledge of the imaging system and degradation process, and adopts a deconvolution algorithm to estimate latent images. However, prior knowledge of degradation models is generally unavailable in practical situations - the case is categorized as blind image deblurring (BID). Most conventional BID algorithms make estimations according to image statistics and heuristics. Fergus et al. [5] proposed a spatial domain prior of a uniform camera blur kernel and camera rotation. Li et al. [24] created a maximum-a-posterior (MAP) based framework and adopted iterative approach for motion deblurring. Recent approaches have turned to deep learning for improved performances. Xu et al. [23] adapted convolutional kernels in convolutional neural networks (CNNs) to blur kernels. Schuler et al. [20] built stacked CNNs that pack feature extraction, kernel estimation and image estimation modules. Chakrabarti [3] proposed to predict complex Fourier coefficients of motion kernels by using neural networks.

### 2.3 Image Quality Measures

Image quality assessment (IQA) is a critical and necessary step to provide quantitative objective measures of visual quality for image processing tasks. IQA methods have been an important and active research topic. Here we focus on four commonly used IQA methods: PSNR, SSIM, FSIM and GMSD.

Peak signal-to-noise ratio (PSNR) is a simple signal fidelity measure that calculates the ratio between the maximum possible pixel value in the image and the mean squared error (MSE) between distorted and reference images.

Structural similarity index measure (SSIM) considers image quality degradation as perceived change of structural information in image. Since structural information is independent of illumination and contrast [22], SSIM index is a linear combination of these three relatively independent terms, luminance  $l(x, y)$ , contrast  $c(x, y)$  and structure comparison function  $s(x, y)$ . Besides, the measure is based on local patches of two aligned images because luminance and contrast vary across the entire image. To avoid blocking effect in the resulting SSIM index map,  $11 \times 11$  circular-symmetric Gaussian weighing function is applied before computation. Patch based SSIM index is defined as in Eq. 1, while for the entire image, it is common to use mean SSIM (MSSIM) as the evaluation metric for the overall image quality (Eq. 2).

$$\begin{aligned} SSIM(x, y) &= l(x, y) \cdot c(x, y) \cdot s(x, y) \\ &= \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)} \end{aligned} \quad (1)$$

$$MSSIM(X, Y) = \frac{1}{M} \sum_{m=1}^M SSIM(x_m, y_m) \quad (2)$$

where  $x$  and  $y$  are two local windows from two aligned images  $X$  and  $Y$ .  $\mu_x$  is the mean of  $x$ ,  $\mu_y$  is the mean of  $y$ ,  $\sigma_x^2$  is the variance of  $x$ ,  $\sigma_y^2$  is the variance of  $y$ ,  $\sigma_{xy}$  is the covariance of  $x$  and  $y$ , constants  $c_1$  and  $c_2$  are conventionally set to 0.0001 and 0.0009 to stabilize the division.  $M$  is the total number of windows.

Feature similarity index measure (FSIM) is based on similarity of salient low-level visual features, i.e. the phase congruency (PC). High PC means the existence of highly informative features, where the Fourier waves at different frequencies have congruent phases [28]. To compensate the contrast information that the primary feature PC is invariant to, gradient magnitude is added as the secondary feature for computing FSIM index.

First, PC map computation of an image is conducted by generalizing the method proposed in [12] from 1-D signal to 2-D grayscale image, by the means of applying the spreading function of Gaussian. 2-D log-Gabor filter extracts a quadrature pair of even-symmetric filter response and odd-symmetric filter response  $[e_{n,\theta_j}(a), o_{n,\theta_j}(a)]$  at pixel  $a$  on scale  $n$  in the image. Transfer function is formulated as follows,

$$G(\omega, \theta_j) = \exp\left(-\frac{(\log(\frac{\omega}{\omega_0}))^2}{2\sigma_r^2}\right) \cdot \exp\left(-\frac{(\theta - \theta_j)^2}{2\sigma_\theta^2}\right) \quad (3)$$

where  $\omega$  represents the frequency,  $\theta_j = \frac{j\pi}{J}$  ( $j = \{0, 1, \dots, J-1\}$ ) represents the orientation angle of the filter,  $J$  is the number of orientations.  $\omega_0$  is the filter center frequency,  $\sigma_r$  is the filter bandwidth,  $\sigma_\theta$  is the filter angular bandwidth. And the PC at pixel  $a$  is defined as,

$$PC(a) = \frac{\sum_j E_{\theta_j}(a)}{\epsilon + \sum_n \sum_j A_{n,\theta_j}(a)} \quad (4)$$

$$E_{\theta_j}(a) = \sqrt{F_{\theta_j}(a)^2 + H_{\theta_j}(a)^2} \quad (5)$$

$$F_{\theta_j}(a) = \sum_n e_{n,\theta_j}(a), \quad H_{\theta_j}(a) = \sum_n o_{n,\theta_j}(a) \quad (6)$$

$$A_{n,\theta_j}(a) = \sqrt{e_{n,\theta_j}(a)^2 + o_{n,\theta_j}(a)^2} \quad (7)$$

where  $E_{\theta_j}(a)$  is the local energy function along orientation  $\theta$ .

Gradient magnitude (GM) computation follows the traditional definition that computes partial derivatives  $G_h(a)$  and  $G_v(a)$  along horizontal and vertical directions using gradient operators. GM is defined as  $G(a) = \sqrt{G_h(a)^2 + G_v(a)^2}$ .

For calculating FSIM index between  $X$  and  $Y$ , PC and GM similarity measure between these two images are computed as follows,

$$S_{PC}(a) = \frac{2PC_X(a) \cdot PC_Y(a) + T_1}{PC_X^2(a) + PC_Y^2(a) + T_1} \quad (8)$$

$$S_G(a) = \frac{2G_X(a) \cdot G_Y(a) + T_2}{G_X^2(a) + G_Y^2(a) + T_2} \quad (9)$$

$$S_L(a) = S_{PC}(a) \cdot S_G(a) \quad (10)$$

where  $T_1$  and  $T_2$  are positive constants depending on dynamic range of PC and GM values respectively. Based on similarity measure  $S_L(a)$ , the FSIM index is defined as,

$$FSIM(X, Y) = \frac{\sum_a^\Omega S_L(a) \cdot PC_m(a)}{\sum_a^\Omega PC_m(a)} \quad (11)$$

where  $PC_m(a) = \max(PC_X(a), PC_Y(a))$  is to balance the importance between similarity between  $X$  and  $Y$ ,  $\Omega$  is the entire spatial domain of image. Introduced in [28],  $FSIM_c$  is for colour images by incorporating chromatic information.

Gradient magnitude standard deviation (GMSD) mainly utilizes feature properties in image gradient domain to derive quality measure. GMSD metric calculates the standard deviation of gradient magnitude. Prewitt filter is commonly adopted as the gradient operator. Similar to FSIM index, GM similarity measure is firstly computed using Eq. 9. The difference is the Eq. 12. So the smaller GMSD the higher image perceptual quality.

$$GMSD(X, Y) = \sqrt{\frac{1}{N} \sum_{a \in \Omega} (S_G(a) - \text{mean}(G(a)))^2} \quad (12)$$

where  $N$  is the total number of pixels in image,  $\text{mean}(G(a)) = \frac{1}{N} \sum_{a \in \Omega} S_G(a)$ .

### 3 The Proposed Method

We propose modified GAN models that are able to blindly restore sharp latent images with better quality from single blurred images. Quality improvement of restored images is realized by adding a quality loss into the training objective function. We compare three image quality measure based losses, which are based on SSIM, FSIM and MSE. We apply these quality losses to two types of GAN models, LSGAN and WGAN-GP, respectively.

#### 3.1 Loss Function

For simplicity, we first define variables and terms as follows. Batch size is  $m$ , input blurred image samples  $\{I_B^{(i)}\}_{i=1}^m$ , restored image samples  $\{I_R^{(i)}\}_{i=1}^m$ , and original sharp image samples  $\{I_S^{(i)}\}_{i=1}^m$ . The adversarial loss  $\mathcal{L}_{\text{ad}}$ , content loss  $\mathcal{L}_X$ , quality loss  $\mathcal{L}_Q$  are as follows.

**Adversarial Loss.** For LSGAN,

$$G : \mathcal{L}_{\text{ad}} = \frac{1}{m} \sum_{i=1}^m \frac{1}{2} (D(G(I_B^{(i)})) - 1)^2 \quad (13)$$

$$D : \mathcal{L}_{\text{ad}} = \frac{1}{m} \sum_{i=1}^m \frac{1}{2} [(D(I_S^{(i)}) - 1)^2 + D(G(I_B^{(i)}))^2] \quad (14)$$

For WGAN-GP,

$$\mathcal{L}_{\text{ad}} = \frac{1}{m} \sum_{i=1}^m D(I_S^{(i)}) - D(G(I_B^{(i)})) + \lambda [(\|\nabla_{\tilde{x}} D(\tilde{x})\| - 1)^2] \quad (15)$$

**Content Loss.**  $\mathcal{L}_X$  is a  $L_2$  loss based on the difference between the VGG-19 feature maps of generated image and sharp image. As proposed in [9], the VGG19 network is pretrained on ImageNet [4].  $\mathcal{L}_X$  is formulated as,

$$\mathcal{L}_X = \frac{1}{W_{j,k} H_{j,k}} \sum_{x=1}^{W_{j,k}} \sum_{y=1}^{H_{j,k}} (\phi_{j,k}(I_S^{(i)})_{x,y} - \phi_{j,k}(G(I_B^{(i)}))_{x,y})^2 \quad (16)$$

where  $\phi_{j,k}$  is the feature map of the  $k$ -th convolution before  $j$ -th maxpooling layer in the VGG19 network.  $W_{j,k}$  and  $H_{j,k}$  are the dimensions of feature maps.

**Quality Loss.** Based on SSIM and FSIM, quality loss functions are defined as in Eqs. 17 and 18. In addition we experiment a MSE based quality loss (Eq. 19) that computes between  $I_R^{(i)}$  and  $I_S^{(i)}$  and name this quality loss as Pixel Loss.

$$SSIMLoss : \mathcal{L}_Q = 1 - SSIM(I_R^{(i)}, I_S^{(i)}) \quad (17)$$

$$FSIMLoss : \mathcal{L}_Q = 1 - FSIM(I_R^{(i)}, I_S^{(i)}) \quad (18)$$

$$PixelLoss : \mathcal{L}_Q = MSE(I_R^{(i)}, I_S^{(i)}) \quad (19)$$

Combining the adversarial loss  $\mathcal{L}_{\text{ad}}$ , content loss  $\mathcal{L}_X$  and image quality loss  $\mathcal{L}_Q$ , the overall loss function is formulated as,

$$\mathcal{L} = \mathcal{L}_{\text{ad}} + 100\mathcal{L}_X + \mathcal{L}_Q \quad (20)$$

### 3.2 Network Architecture

We adopted the network architecture proposed in [13]. The generator has two strided convolution blocks, nine residual blocks, two transposed convolution blocks. The residual block was formed by one convolution layer, an instance normalization layer and ReLU activation. Dropout regularization with rate of 50% was adopted. Besides, global skip connection learned a residual image, which was added with the output image to constitute the final restored image  $I_R$ . The discriminator was a  $70 \times 70$  PatchGAN [8], containing four convolutional layers, each followed by BatchNorm and LeakyReLU with  $\alpha = 0.2$  except for the first layer.

## 4 Experiments

### 4.1 Datasets

The training dataset was sampled from the train set of the Microsoft Common Object in COntext (MS COCO) dataset [15], which contains over 330,000 images covering 91 common object categories in natural context. We adopted the method in [2] to synthesize motion blur kernels. Kernel size was set as  $31 \times 31$ , motion parameters followed the default setting in the original paper. In total, we generated 250 kernels to randomly blur MS COCO dataset images. We randomly selected 6000 images from the MS COCO train set for training and 1000 from the test set for evaluation. Besides, trained models were tested on two other datasets, the GoPro dataset [18] and the Kohler dataset [11].

GoPro dataset has 3214 pairs of realistic blurry images and their sharp version at  $1280 \times 720$  resolution. Images are 240 fps video sequences captured by GoPro Hero 4 camera in various daily or natural scenes. Blurry images are averaged from a varying number of consecutive frames, in order to synthesize motion blur of varying degrees. This is a common benchmark for image motion deblurring. We randomly select 1000 pairs for evaluation.

Kohler dataset contains four original images, 48 blurred images that are generated by applying 12 approximations of human camera shakes on original images respectively. The dataset is also considered as a benchmark for evaluation of blind deblurring algorithms.

## 4.2 Implementation

We performed experiments using PyTorch [19] on a Nvidia Titan V GPU. All images were scaled to  $640 \times 360$  and randomly cropped to patches of size  $256 \times 256$ . Networks were optimized using the Adam solver [10]. Initial learning rate was  $10^{-4}$  for both generator and critic. For LSGAN models, learning rate remained unchanged for the first 150 epochs and linearly decayed to zero for the rest 150 epochs, and it took around 6 days to finish the training. For WGAN-GP models, learning rate was maintained for 50 epochs and then linearly decreased to zero for another 50 epochs. Training took around 3 days to converge.

## 4.3 Results and Analysis

We name the model without quality loss as the baseline model. Evaluation metrics include PSNR, SSIM, FSIM and GMSD. Quantitative performances are given in Tables 1, 2 and 3. Examples of resulting images are shown in Figs. 1, 2 and 3.

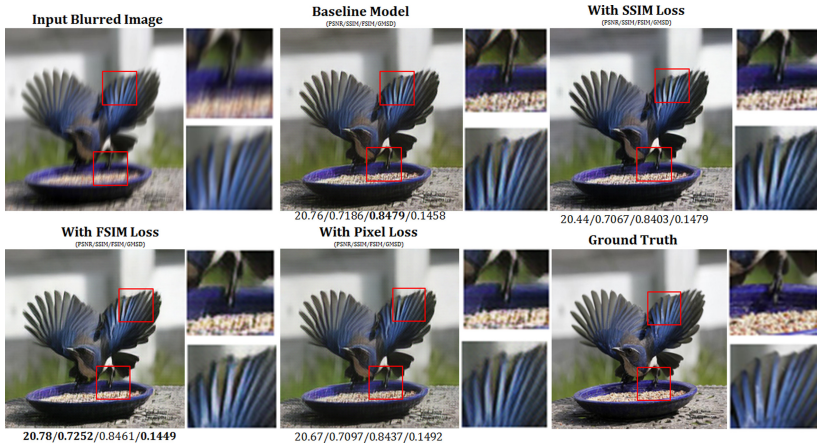
**MS COCO Dataset.** From Table 1, we can observe that WGAN-GP model with SSIM loss function has the best performance on all four measures. The WGAN-GP model with FSIM loss function has a comparable performance with subtle differences in values. But significant improvements from the baseline model that does not include quality losses demonstrate the usefulness of SSIM or FSIM loss. From the examples shown in Fig. 1, restored images by SSIM loss and FSIM loss contain more details visually and also have better quantitative evaluation results than their counterparts.

**Table 1.** Model performance evaluation measures averaged on 1000 images of MS COCO dataset.

| Measures | Baseline |         | With SSIM Loss |               | With FSIM Loss |         | With Pixel Loss |         |
|----------|----------|---------|----------------|---------------|----------------|---------|-----------------|---------|
|          | LSGAN    | WGAN-GP | LSGAN          | WGAN-GP       | LSGAN          | WGAN-GP | LSGAN           | WGAN-GP |
| PSNR(dB) | 18.14    | 19.57   | 17.50          | <b>20.16</b>  | 19.91          | 19.98   | 16.50           | 19.52   |
| SSIM     | 0.7447   | 0.8032  | 0.7513         | <b>0.8218</b> | 0.8175         | 0.8196  | 0.7412          | 0.8022  |
| FSIM     | 0.8178   | 0.8207  | 0.8170         | <b>0.8217</b> | 0.8208         | 0.8216  | 0.8172          | 0.8205  |
| GMSD     | 0.1460   | 0.1419  | 0.1438         | <b>0.1391</b> | 0.1404         | 0.1393  | 0.1439          | 0.1413  |

**GoPro Dataset.** We can find similar performances on MS COCO dataset, although the training was solely based on synthetic blurred images from MS COCO dataset. Still WGAN-GP is the model that gives better performance. In terms of PSNR and SSIM metrics, performance of WGAN-GP model with SSIM loss function is ranked the first. And FSIM loss function encourages the model to produce better results with regard to FSIM and GMSD metrics.





**Fig. 1.** Results generated by WGAN-GP model with various loss functions on MS COCO dataset.

**Table 2.** Model performance evaluation measure averaged on 1000 images of GoPro dataset.

| Measures | Baseline |         | With SSIM Loss |               | With FSIM Loss |               | With Pixel Loss |         |
|----------|----------|---------|----------------|---------------|----------------|---------------|-----------------|---------|
|          | LSGAN    | WGAN-GP | LSGAN          | WGAN-GP       | LSGAN          | WGAN-GP       | LSGAN           | WGAN-GP |
| PSNR(dB) | 18.58    | 20.01   | 19.06          | <b>20.99</b>  | 20.42          | 20.86         | 19.06           | 20.17   |
| SSIM     | 0.8017   | 0.8446  | 0.8047         | <b>0.8650</b> | 0.8557         | 0.8627        | 0.8160          | 0.8480  |
| FSIM     | 0.8266   | 0.8379  | 0.8188         | 0.8379        | 0.8337         | <b>0.8386</b> | 0.8263          | 0.8379  |
| GMSD     | 0.1181   | 0.1029  | 0.1195         | 0.0988        | <b>0.0962</b>  | 0.0990        | 0.1117          | 0.0991  |



**Fig. 2.** Results generated by WGAN-GP model with different loss functions on GoPro dataset.



**Table 3.** Model performance evaluation measure averaged on 48 images of Kohler dataset.

| Measures | Baseline |         | With SSIM Loss |         | With FSIM Loss |         | With Pixel Loss |         |
|----------|----------|---------|----------------|---------|----------------|---------|-----------------|---------|
|          | LSGAN    | WGAN-GP | LSGAN          | WGAN-GP | LSGAN          | WGAN-GP | LSGAN           | WGAN-GP |
| PSNR(dB) | 15.94    | 17.81   | 16.50          | 18.19   | <b>18.37</b>   | 17.79   | 13.18           | 17.73   |
| SSIM     | 0.6042   | 0.6775  | 0.5689         | 0.6818  | <b>0.6908</b>  | 0.6886  | 0.5444          | 0.6810  |
| FSIM     | 0.7562   | 0.7565  | <b>0.7607</b>  | 0.7559  | 0.7555         | 0.7539  | 0.7512          | 0.7564  |
| GMSD     | 0.1984   | 0.2040  | <b>0.1971</b>  | 0.2055  | 0.2048         | 0.2044  | 0.2002          | 0.2031  |

**Kohler Dataset.** Compared to results of above two datasets, results given in Table 3 are generally low. Considering images from Kohler dataset are approximations of human camera shake, models trained by synthetic blurred images have limited generalization on tackling such real blurry images. But SSIM and FSIM loss still demonstrate their effectiveness in improving image quality as shown in Table 3 and Fig. 3, although the example in Fig. 3 is a challenging one to restore.



**Fig. 3.** Results generated by WGAN-GP model with different loss functions on Kohler dataset.

As one can observe from Tables 1, 2 and 3, quantitative results show that image quality measure based loss functions are effective components for GANs to further improve generated image quality. Among the three loss functions, models trained with the SSIM loss and the FSIM loss have comparable performances and generate the best results, compared to the baseline model and model trained with the pixel loss. Experimentation on three different datasets in two different types of GAN model demonstrates effectiveness of inclusion of such quality loss functions.

For visual comparison, models with the SSIM or FSIM loss function restore images with better texture details and edges. However, if we carefully observe the details in generated image patches, we can find that with the SSIM loss function, the patches have window artifacts while models with FSIM loss produce smoother details when zoomed in. It is because the SSIM loss is computed by basing on local windows of images while the FSIM loss is computed pixel by pixel. And in the result images generated by models trained with pixel loss function, details are still blurred, illustrating that L2 loss in the spatial domain has little contribution to image quality improvement. In general, compared to FSIM loss, SSIM loss has the advantage of computation efficiency and performance stability in various quantitative evaluation metrics and visual quality.

It is also noted that WGAN-GP generates better results than LSGAN and converges faster. But training WGAN-GP model is more difficult; during the experimentation, model training diverges more often than LSGAN. Parameter tuning becomes a crucial step in experiment setting for training WGAN-GP, and it is very time-consuming to find a feasible model structure and network parameters.

## 5 Conclusion

In this paper, we tackled the problem of image deblurring with the framework of adversarial learning models. Losses based on image quality measures were proposed as additional components in the training objective function of the GAN models. Experimental results on various benchmark datasets have demonstrated the effectiveness of adding such image quality losses and their potential in improving quality of generated images.

For future work, training data could include more diverse datasets to improve generalization ability of the network. So far the weightings of these various losses in the overall objective function have not been fine-tuned; further experiments could be conducted on further improving the performance by fine-tuning these parameters. Besides, considering flexibility and adaptability of these image quality losses, applications for solving other image enhancement and restoration tasks would also be worth investigating in the future.

## References

1. Arjovsky, M., Chintala, S., Bottou, L.: Wasserstein GAN. arXiv preprint [arXiv:1701.07875](https://arxiv.org/abs/1701.07875) (2017)
2. Boracchi, G., Foi, A.: Modeling the performance of image restoration from motion blur. *IEEE Trans. Image Process.* **21**(8), 3502–3517 (2012)
3. Chakrabarti, A.: A neural approach to blind motion deblurring. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) *ECCV 2016*. LNCS, vol. 9907, pp. 221–235. Springer, Cham (2016). [https://doi.org/10.1007/978-3-319-46487-9\\_14](https://doi.org/10.1007/978-3-319-46487-9_14)
4. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: ImageNet: a large-scale hierarchical image database. In: *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, pp. 248–255. IEEE (2009)

5. Fergus, R., Singh, B., Hertzmann, A., Roweis, S.T., Freeman, W.T.: Removing camera shake from a single photograph. In: ACM SIGGRAPH 2006 Papers, pp. 787–794. Association for Computing Machinery (2006)
6. Goodfellow, I., et al.: Generative adversarial nets. In: Advances in Neural Information Processing Systems, pp. 2672–2680 (2014)
7. Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., Courville, A.C.: Improved training of Wasserstein GANs. In: Advances in Neural Information Processing Systems, pp. 5767–5777 (2017)
8. Isola, P., Zhu, J.Y., Zhou, T., Efros, A.A.: Image-to-image translation with conditional adversarial networks. In: Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition, pp. 1125–1134 (2017)
9. Johnson, J., Alahi, A., Fei-Fei, L.: Perceptual losses for real-time style transfer and super-resolution. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. Lecture Notes in Computer Science, vol. 9906, pp. 694–711. Springer, Cham (2016). [https://doi.org/10.1007/978-3-319-46475-6\\_43](https://doi.org/10.1007/978-3-319-46475-6_43)
10. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. arXiv preprint [arXiv:1412.6980](https://arxiv.org/abs/1412.6980) (2014)
11. Köhler, R., Hirsch, M., Mohler, B., Schölkopf, B., Harmeling, S.: Recording and playback of camera shake: benchmarking blind deconvolution with a real-world database. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012. LNCS, vol. 7578, pp. 27–40. Springer, Heidelberg (2012). [https://doi.org/10.1007/978-3-642-33786-4\\_3](https://doi.org/10.1007/978-3-642-33786-4_3)
12. Kovese, P., et al.: Image features from phase congruency. *Videre J. Comput. Vis. Res.* **1**(3), 1–26 (1999)
13. Kupyn, O., Budzan, V., Mykhailych, M., Mishkin, D., Matas, J.: DeblurGAN: blind motion deblurring using conditional adversarial networks. In: Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition, pp. 8183–8192 (2018)
14. Ledig, C., et al.: Photo-realistic single image super-resolution using a generative adversarial network. In: Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition, pp. 4681–4690 (2017)
15. Lin, T.-Y., et al.: Microsoft COCO: common objects in context. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8693, pp. 740–755. Springer, Cham (2014). [https://doi.org/10.1007/978-3-319-10602-1\\_48](https://doi.org/10.1007/978-3-319-10602-1_48)
16. Mao, X., Li, Q., Xie, H., Lau, R.Y., Wang, Z., Paul Smolley, S.: Least squares generative adversarial networks. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2794–2802 (2017)
17. Mirza, M., Osindero, S.: Conditional generative adversarial nets. arXiv preprint [arXiv:1411.1784](https://arxiv.org/abs/1411.1784) (2014)
18. Nah, S., Hyun Kim, T., Mu Lee, K.: Deep multi-scale convolutional neural network for dynamic scene deblurring. In: Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition, pp. 3883–3891 (2017)
19. Paszke, A., et al.: Automatic differentiation in PyTorch. In: NIPS-W (2017)
20. Schuler, C.J., Hirsch, M., Harmeling, S., Schölkopf, B.: Learning to deblur. *IEEE Trans. Pattern Anal. Mach. Intell.* **38**(7), 1439–1451 (2015)
21. Sønderby, C.K., Caballero, J., Theis, L., Shi, W., Huszár, F.: Amortised map inference for image super-resolution. arXiv preprint [arXiv:1610.04490](https://arxiv.org/abs/1610.04490) (2016)
22. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. *IEEE Trans. Image Process.* **13**(4), 600–612 (2004)

23. Xu, L., Ren, J.S., Liu, C., Jia, J.: Deep convolutional neural network for image deconvolution. In: *Advances in Neural Information Processing Systems*, pp. 1790–1798 (2014)
24. Xu, L., Zheng, S., Jia, J.: Unnatural l0 sparse representation for natural image deblurring. In: *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, pp. 1107–1114 (2013)
25. Yeh, R., Chen, C., Lim, T.Y., Hasegawa-Johnson, M., Do, M.N.: Semantic image inpainting with perceptual and contextual losses **2**(3) (2016). arXiv preprint [arXiv:1607.07539](https://arxiv.org/abs/1607.07539)
26. Zhang, H., et al.: Stackgan: text to photo-realistic image synthesis with stacked generative adversarial networks. In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 5907–5915 (2017)
27. Zhang, H., et al.: Stackgan++: realistic image synthesis with stacked generative adversarial networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **41**(8), 1947–1962 (2018)
28. Zhang, L., Zhang, L., Mou, X., Zhang, D.: FSIM: a feature similarity index for image quality assessment. *IEEE Trans. Image Process.* **20**(8), 2378–2386 (2011)
29. Zhu, J.Y., Park, T., Isola, P., Efros, A.A.: Unpaired image-to-image translation using cycle-consistent adversarial networks. In: *Proceedings of the IEEE International Conference on Computer Vision*, pp. 2223–2232 (2017)