Edge Detection of Motion-Blurred Images based on GANs

Feng Li
lifeng@dhu.edu.cn
Jiyu Li
lijiyu0219@126.com

School of Computer Science and Technology
Donghua University
Shanghai, China

Abstract

Motion blur is a challenging problem in many image processing tasks. It leads to the degradation of the image, especially the edge information within. This paper introduces an edge detection algorithm for motion-blurred images based on *Generative Adversarial Networks*, treating edge detection as an image translation problem. We determined the components and parameters of the networks through experimental comparisons and finally adopted *U-Net* and *PatchGAN* as the backbones. We also proposed a new loss calculation method called *Motion Loss*, which tolerates the random offset of edges due to motion blur. Finally, we performed several experiments on the GOPRO dataset. The results showed that applying *Motion Loss* could lead to better edge results and that our method worked well in the edge detection of motion-blurred images.

1 Introduction

It is unnecessary to reaffirm the importance of edge detection. The study of edge detection has lasted for decades, and the results are even better than human perception. However, sometimes edge information could be influenced (*e.g.*, blur, haze, *etc.*), and ordinary methods could not detect it accurately. This paper focuses on the edge detection of motion-blurred images, which often occur unexpectedly and paralyze most edge detection algorithms.

Motion blur is caused by the relative motion between camera and object, *e.g.*, high-speed object movement and camera shake, *etc*. It can be described by the following expression[7, 9]:

$$I_B = I_S * Kernel + Noise \tag{1}$$

Where $*$ represents the convolution operation, I_B represents a blurred image, and I_S represents a sharp image. *Kernel* represents the blur kernel, which is also called PSF (Point Spread Function). In other words, motion blur is caused by the convolution of the potential sharp image and blur kernel. The motion blur kernel is related to the trajectory of relative camera motion.

Edge information is represented by image gradients. Traditional edge detection algorithms such as Canny and Sobel calculate vertical and horizontal gradients to locate edges. However, the blurring procedure could destroy the gradient information, causing false positives produced by ordinary edge detection methods.

Our method is specifically proposed for edge detection in motion blur scenes. Instead of analyzing features of edges in motion-blurred images, we use GANs to generate edges using blurred images. We propose a novel conception of loss calculation during the training process, which we term *Motion Loss* (ML). It takes into account the movement of edges caused by motion blur. Several edge detection metrics such as *ODS* [1], *OIS* [1], and *Average Precision* (AP) were used for evaluation. The experiments on the GOPRO [18] dataset showed that our approach achieved better statistical results than ordinary edge detection methods like Canny and HED.

2 Related work

2.1 Motion blur

Current research on motion blur focuses on blur detection and deblurring. Motion blur can be caused by camera motion or object motion during exposure. Camera motion may lead to global blur in an image, while object motion causes blur locally. Although lots of information is missing due to motion blur, the blur itself represents some motion information such as direction and speed, and the blur kernel can be restored to some extent.

Motion blur detection aims to determine the blur region and extract the information within the blur. Chen *et al*. [3] use the high-frequency feature of motion blur to locate the region. They design a closed-form solution to estimate motion direction and calculate directional high-frequency energy to determine a blurred patch. Xing *et al*. [26] compute the first and second order directional derivatives and achieve the rudimentary segmentation by setting thresholds on derivatives. The refined segmentation can then be achieved by morphological operations. Caglioti and Giusti [24] use a variety of techniques to recover ball motion information, including speed, direction, and trajectory, from a single motion-blurred image with a known background.

Deblurring of motion-blurred images can be divided into non-blind and blind deblurring, depending on whether the blur kernel is known or not. Non-blind deblurring requires a known blur kernel to apply deconvolution on a blurred image. Common non-blind methods include the *Lucy-Richardson* algorithm and the *Wiener filter*. However, the blur kernel is difficult to be recorded, so many methods to estimate blur kernels are proposed. Fergus *et al*. [4] adopted a Bayesian approach to find the blur kernel implied by the distribution of blurred images. Then standard deconvolution algorithm can be used to reconstruct sharp images. Nimisha *et al*. [19] handled the motion blur with foreground and background respectively to obtain clear frames. Sun *et al*. [23] introduced *Convolutional Neural Networks* (CNNs) to estimate blur kernel and operate deblurring procedure on non-uniform motion blur. With the success of deep learning, there appeared many end-to-end deblurring methods using CNN. Nah *et al*. [18] proposed Deep Deblur, using multi-scale CNN at image deblurring; Yong *et al*. [27] proposed an attention mechanism for deblurring; Kupyn *et al*. [13] proposed *DeblurGAN* and *DeblurGANv2* [14], using several GAN structures and objective functions to achieve state-of-art performances.

There are some edge narrowing methods that can be applied to blurred images [10]. Schavemaker *et al*. [22] proposed a morphological method for image sharpening, which makes edges clearer. However, since edges in motion blur are pretty irregular, such methods could not precisely recover edges in their location. Ordinary edge detection methods such as Sobel and Canny rely on thresholds to determine an edge, but it is impossible to find a suitable threshold for motion-blurred images since there is too much noise brought by motion blur.

2.2 Generative Adversarial Networks

Goodfellow *et al.* [5] first proposed the concept of *Generative Adversarial Networks* (*GANs*), which was then widely used in the field of image processing. The GANs consist of a generator and a discriminator, where the purpose of the generator is to produce as realistic a data distribution as possible based on the input, and the discriminator is trying to distinguish between real data and generated data.

During the training process of the GANs, the generator and the discriminator are always in the process of a minimax game, as expressed in equation 2.

$$\min_G \max_D V(D,G) = \mathbb{E}_{x\sim p_{data}(x)}[\log D(x)] + \mathbb{E}_{z\sim p_z(z)}[\log(1-D(G(z)))]. \tag{2}$$

Where D represents the discriminator, G represents the generator, x is sampled from real data, and z is the random noise.

The vanilla GANs has many limits and problems. Luckily, the potential of GANs is foreseen by the public, and enormous improvements and applications based on GANs have been proposed. It is now widely applied to many areas, including image generation, semi-supervised learning, *etc*. [28]

3 Proposed method

In this paper, we use GANs to detect edges. The generator is used to generate edges using motion-blurred images, while the discriminator distinguishes the generated edges from real edges. The feedback of the discriminator is provided to the generator for further adjustment to generate more realistic edges. Thus, the training procedure runs iteratively. We consider edge detection as a process of image translation,*i.e.*, we use GANs to generate edge images from blurred images, and the edge results are stored in edge images.

Edges in blurred images do not coincide with edges in sharp images, since the blur procedure causes movement of edges. We propose a novel conception of loss calculation to deal with such a problem, which we term *Motion Loss* (ML). It combines edge offset and pix-level loss such as *L1* distance (MAE). Through theoretical analysis and experimental comparison, the architecture of the network and the ways of loss calculation were determined. We use *ODS* [1], *OIS* [1], and *Average Precision*(AP) to get statistical results. The experiments on the GOPRO dataset indicate that our proposed method achieves better performance in motion blur situations than ordinary edge detection methods such as Canny and *Holistically-Nested Edge Detection*(HED) [25], both in statistical results and visual appearance.

This paper uses the GOPRO dataset for training and testing. The GOPRO dataset was developed by Nah *et al.* [18] and was originally used for the task of deblurring. The dataset uses a high-speed camera to capture a series of sharp images and then averages each sharp image with several adjacent images to obtain their corresponding blurred images. Since our target is the edges, we then use the HED edge detection algorithm to perform edge detection on the sharp image to obtain the corresponding edge results(edge images), which forms the paired data of blurred images and the edge images as training and testing data. The overall process is shown in Figure 1.

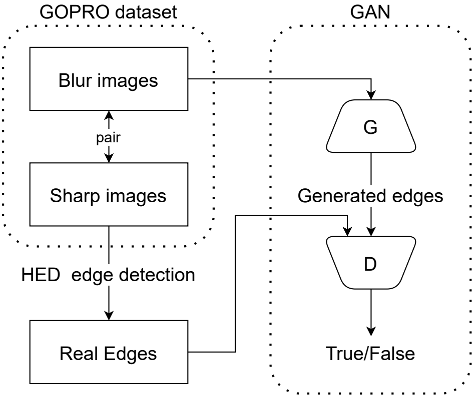

Figure 1: The overall process in this paper

3.1 Loss function

The loss function of GANs consists of two components: adversarial loss and content loss.

$$Loss = Loss_{adversarial} + \lambda Loss_{content} \tag{3}$$

Where $Loss_{adversarial}$ is adversarial loss and $Loss_{content}$ is the content loss.

3.2 Adverarial loss

According to equation 3, The objective function of GANs is a minimax expression. To simplify the training process, we convert it to a minimized form. The objective functions for minimizing the loss in the discriminator and generator are:

$$\min_D V(D) = \mathbb{E}_{x\sim p_{data}(x)}[\log(1-D(x))] + \mathbb{E}_{z\sim p_z(z)}[\log D(G(z))]. \tag{4}$$

$$\min_G V(G) = \mathbb{E}_{z\sim p_z(z)}[\log(1-D(G(z)))] \tag{5}$$

In Equations 4 and 5, the cross-entropy method is used to calculate the adversarial loss. Lease Square GAN(LSGAN) [16] and WGAN-GP [6] are also used to calculate the adversarial loss. LSGAN uses square operation instead of log function, and WGAN-GP uses Wasserstein distance and gradient loss. This paper compares the training results of the three different methods, as shown in Figure 2. It shows that using cross-entropy(b) would generated the clearest edge results, while the other two adversarial losses(c, d) tend to generate blurry edge results. The statistical result is shown in Table 1. ODS, OIS, and AP are used as evaluation indicators. The difference of OIS between them is smaller, however, according to the AP index, cross-entropy is the best. In addition, We noticed that curved edges occur more often when the LSGAN is used. Based on the above results, cross-entropy is adopted to calculate the adversarial loss in this paper.

3.3 Content loss

The combination of adversarial and content loss functions in the training process can get better results [11, 20]. Common content loss functions include *MAE*, *MSE*, *etc*. [29], but

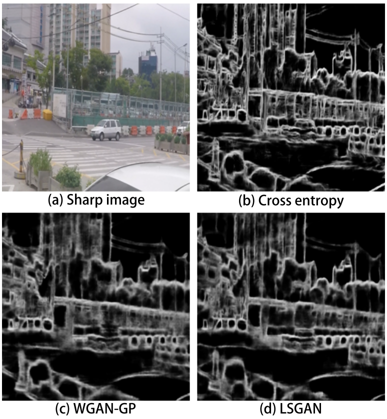

Figure 2: Results of the three adversarial losses

	ODS	OIS	AP
Cross-entropy	0.694	0.697	0.623
WGAN-GP	0.696	0.699	0.606
LSGAN	0.698	0.704	0.616

Table 1: Statistical results of the three adversarial losses

these methods for per-pixel are not appropriate when dealing with images of high-frequency features such as textures. Minimizing MSE loss is prone to averaging between pixels, producing overly smooth results [11, 13, 15] and making the edge blurred. Bruna *et al.* [2]and Johnson *et al.* [12] proposed the perceptual loss function using convolutional networks to calculate the loss. It has better results than MAE and MSE loss.

However, when dealing with motion-blurred images, none of the above approaches can handle the motion blur problem. Because the edges are randomly shifted during the motion blur process, the edges corresponding to the motion blur images do not match the real edges exactly in the same position, making it difficult to obtain a good result with traditional content loss calculation. Thus, this paper proposes the concept of *Motion Loss* for motion-blurred images. When comparing the generated edge with the real edge, the generated edge is shifted within a certain range. Traditional content loss is calculated for each shift. The smallest loss value among these shifts is selected as the final loss value, thus reducing the effect of edge offset caused by motion blur.

The calculation of *Motion Loss* is shown as follows:

For matrix M of size $m \times n$, We define a horizontal shift matrix H^p, which represents p positions shifted rightward in the horizontal direction, and a vertical shift matrix V^q represents q positions shifted upward in the vertical direction:

$$H^p_{i,j} = \delta_{i+p,j}, V^q_{i,j} = \delta_{i,j+q} \tag{6}$$

Where $H^p_{i,j}$,$V^q_{i,j}$ represents the elements in row i and column j of the matrix, δ(Kronecker delta) is calculated as follows:

$$\delta_{ij} = \begin{cases} 0 & \text{if } i \neq j, \\ 1 & \text{if } i = j. \end{cases} \tag{7}$$

The shift operation on the matrix M can be expressed as: $M_{shifted} = H^p M V^q$.

Assuming the horizontal shift range between $[a,b]$ and the vertical shift range between $[c,d]$, *Motion Loss* is calculated as follows:

$$Motion\ Loss = \min_{p\in[a,b],q\in[c,d]} Criterion(H^p M V^q, M_0) \tag{8}$$

Where M_0 is the ground truth of edge results. *Criterion* could be *MAE*, *MSE*, or *Perceptual Loss*. It is used to calculate the traditional content loss between the generated edges after shift process and the real edges. In this paper, we choose *MAE* as *Criterion*. The entire content loss calculation is defined as *Motion Loss with MAE*.

We compared the performance of different content loss functions: MAE, MSE, *Perceptual Loss*, and *Motion Loss with MAE*, as shown in Figure 3. It is shown that *Motion Loss with MAE* (c) could lead to best edge results; Using MAE loss lead to the detection of non-existent edges (d-1); MSE loss caused bending edges, which is quite obvious (e-2); The results of Perceptual loss is similar to that of the *Motion Loss with MAE*, but there are still some false negatives (f-1).

The statistical results is shown in Table 2. It can be seen in the table that the calculation method of *Motion Loss* can slightly improve the calculation effect of MAE, which is better than MSE and *Perceptual Loss*.

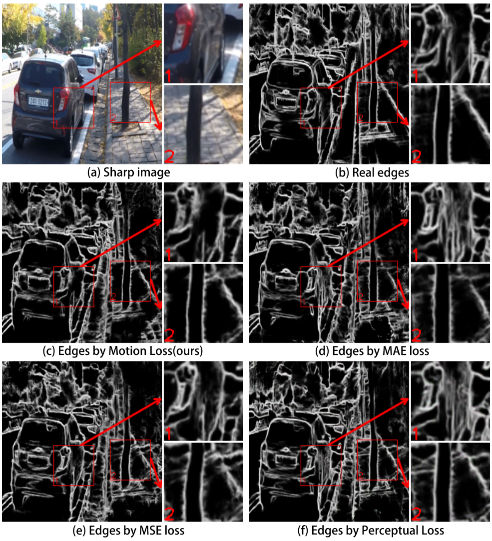

Figure 3: Comparison of *Motion Loss* with the other three content losses.

	ODS	OIS	AP
MAE loss	0.694	0.697	0.623
MSE loss	0.700	0.704	0.629
Perceptual loss	0.701	0.704	0.635
Motion Loss with MAE	0.705	0.708	0.634

Table 2: Statistical results of different content loss functions

3.4 Network structure

This paper selects *Conditional Generative Adversarial Network* (CGAN) [17] as the main framework. The generator part can use *U-Net* [21] and *ResNet* [8], and U-Net is found to be more effective through comparison experiments. The experiment result is shown in Figure 4,

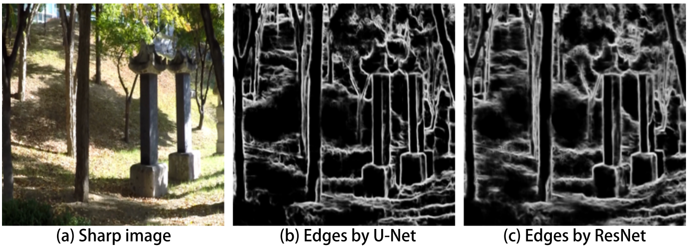

Figure 4: U-Net and ResNet

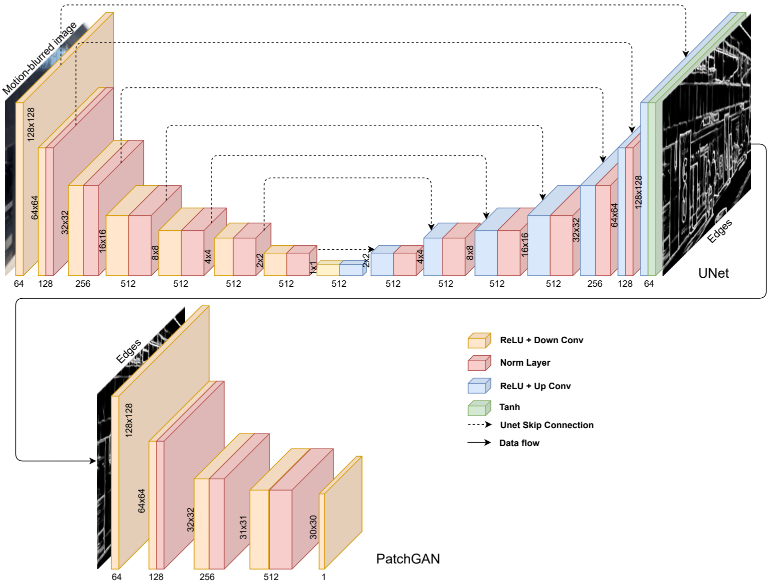

Figure 5: Overall network structure

where it can be seen that the edges generated using U-Net(b) are more visually satisfactory, while the result of ResNet(c) contains more blurry edges and false edges. Therefore, we use U-Net as the generator.

In the selection of discriminator, *PatchGAN* [ ] is selected in this paper. Unlike ordinary GAN, PatchGAN generates a matrix for each image to be judged. Each value of the matrix corresponds to a small area of the input image to have finer control over the image and pay more attention to the details of the image. The overall network structure of this paper is shown in Figure 5.

4 Experiments

4.1 Dataset

This paper applies the GOPRO dataset as the training and test data, but the data lacks edge information. Traditional edge detection, such as Sobel and Canny, requires a threshold to adjust detection sensitivity. However, it is difficult to decide a suitable threshold that can effectively detect all the edges for all images. Many recent papers have used *Holistically-Nested Edge Detection* (HED) as a tool to detect edges. HED is a hierarchical edge detection method using a *VGG* network, which detects the main contour edges of objects in the frame while being insensitive to texture features and noise within the object, resulting in relatively complete and uninterrupted edges. This paper uses HED to obtain edge results from sharp images in the dataset, which is assumed to be the ground truth of edge results. See Figure 1 for the whole process.

4.2 Results

In the pre-processing data stage, the main edges of the sharp images in the dataset were extracted by HED, to obtain paired dataset of “blurred image & edge image”, in which we selected 4400 pairs as training data and 1633 pairs as test data. To improve the training speed, the original image was resized to 256x256, the batch size was set to 32, the initial learning rate was 0.0002, and the number of epochs was 200, where the learning rate was kept constant for the first 100 epochs and decayed linearly for the next 100 epochs. The λ in Equation 3 was set to 100.0. We performed *Non-Maximum-Suppression*(NMS) on the test results and calculated its *OIS*, *ODS*, and *AP* indexes.

We compared our method with the HED and Canny algorithms for motion-blurred images, as shown in Figure 6. It shows that HED(c) produced quite blurry edge results. Although NMS can extract finer edges from them, the accuracy is still unsatisfying; Canny with an appropriate threshold(d) can extract the contour edges but also detected lots of false positives, and most detected edges were disconnected; Our algorithm achieved the best result among them. The statistical results are shown in Table 3, and the precision-recall curve (PR curve) is shown in Figure 7. The results showed that the algorithm in this paper worked well in the edge detection of motion-blurred images.

	ODS	OIS	AP
HED	0.585	0.596	0.437
Canny	0.629	0.630	0.412
Ours	0.705	0.708	0.634

Table 3: Evaluation metrics of this paper’s algorithm with HED and Canny on the test dataset

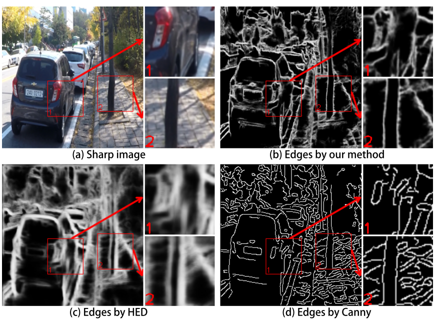

Figure 6: HED, Canny and our method

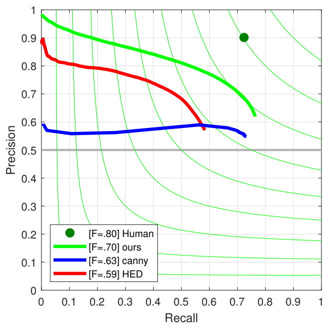

Figure 7: PR curves of the three methods

5 Conclusion

In this paper, we propose a method for edge detection of motion-blurred images along with a new approach for loss calculation called *Motion Loss*. We performed a series of experiments to compare different components of the network to determine the best solution for our application. Finally, we compare our method with other edge detection algorithms to evaluate the effectiveness of the method. The result shows that our approach could generate better edge results.

There are still some problems with edge detection for motion-blurred images, and the accuracy of detection is still somewhat different from that of edge detection for sharp images; the effect of edge detection also weakens as the degree of blurring increases. We will further optimize the network structure to improve the efficiency of the algorithm; design a loss function that is more suitable for motion-blurred edges, and conduct more tests on different types of blurred images.

References

[1] Pablo Arbelaez, Michael Maire, Charless Fowlkes, and Jitendra Malik. Contour detection and hierarchical image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell. , issue_date = May 2011*, 33(5):898–916 , numpages = 19, 2011. ISSN 0162-8828. doi: 10.1109/TPAMI.2010.161,acmid=1963088. URL http://dx.doi.org/10.1109/TPAMI.2010.161. IEEE Computer Society.

[2] Joan Bruna, Pablo Sprechmann, and Yann LeCun. Super-resolution with deep convolutional sufficient statistics. *arXiv preprint arXiv:1511.05666*, 2015.

[3] Xiaogang Chen, Jie Yang, Qiang Wu, and Jiajia Zhao. Motion blur detection based on lowest directional high-frequency energy, 2010.

[4] Rob Fergus, Barun Singh, Aaron Hertzmann, Sam T. Roweis, and William T. Freeman. Removing camera shake from a single photograph, 2006. URL https://doi.org/10.1145/1179352.1141956.

[5] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets, 2014. URL https://proceedings.neurips.cc/paper/2014/file/5ca3e9b122f61f8f06494c97b1afccf3-Paper.pdf. Curran Associates, Inc.

[6] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C. Courville. Improved training of wasserstein gans, 2017. URL https://proceedings.neurips.cc/paper/2017/file/892c3b1c6dccd52936e27cbd0ff683d6-Paper.pdf. Curran Associates, Inc.

[7] Ankit Gupta, Neel Joshi, C. Lawrence Zitnick, Michael Cohen, and Brian Curless. Single image deblurring using motion density functions. In *European Conference on Computer Vision*, Computer Vision – ECCV 2010, pages 171–184. Springer Berlin Heidelberg, 2010. ISBN 978-3-642-15549-9.

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[9] Michael Hirsch, Christian J. Schuler, Stefan Harmeling, and Bernhard Schölkopf. Fast removal of non-uniform camera shake, 2011.

[10] Dong Hongyan. *Research on Some Techniques in Edge Detection*. PhD thesis, National University of Defense Technology, 2008.

[11] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017.

[12] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European conference on computer vision*, pages 694–711. Springer, 2016.

[13] Orest Kupyn, Volodymyr Budzan, Mykola Mykhailych, Dmytro Mishkin, and Jiří Matas. Deblurgan: Blind motion deblurring using conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8183–8192, 2018.

[14] Orest Kupyn, Tetiana Martyniuk, Junru Wu, and Zhangyang Wang. Deblurgan-v2: Deblurring (orders-of-magnitude) faster and better. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8878–8887, 2019.

[15] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, and Zehan Wang. Photo-realistic single image super-resolution using a generative adversarial network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4681–4690, 2017.

[16] Xudong Mao, Qing Li, Haoran Xie, Raymond Y. K. Lau, Zhen Wang, and Stephen Paul Smolley. Least squares generative adversarial networks, Oct 2017.

[17] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets, 2014. URL https://arxiv.org/abs/1411.1784.

[18] Seungjun Nah, Tae Hyun Kim, and Kyoung Mu Lee. Deep multi-scale convolutional neural network for dynamic scene deblurring. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3883–3891, 2017.

[19] Nimisha, Rajagopalan, and R. Aravind. Generating high quality pan-shots from motion blurred videos. *Computer Vision and Image Understanding*, 171:20–33, 2018. ISSN 1077-3142. doi: https://doi.org/10.1016/j.cviu.2018.05.008. URL https://www.sciencedirect.com/science/article/pii/S1077314218300742.

[20] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A. Efros. Context encoders: Feature learning by inpainting, 2016.

[21] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[22] John GM Schavemaker, Marcel JT Reinders, Jan J Gerbrands, and Eric Backer. Image sharpening by morphological filtering. *Pattern Recognition*, 33(6):997–1012, 2000. ISSN 0031-3203.

[23] Jian Sun, Wenfei Cao, Zongben Xu, and Jean Ponce. Learning a convolutional neural network for non-uniform motion blur removal, June 2015.

[24] Caglioti Vincenzo and Giusti Alessandro. Recovering ball motion from a single motion-blurred image. *Computer Vision and Image Understanding*, 113(5): 590–597, 2009. ISSN 1077-3142. doi: https://doi.org/10.1016/j.cviu.2008.01.008. URL https://www.sciencedirect.com/science/article/pii/S1077314208000428. Computer Vision Based Analysis in Sport Environments.

[25] Saining Xie and Zhuowen Tu. Holistically-nested edge detection. In *Proceedings of the IEEE international conference on computer vision*, pages 1395–1403, 2015.

[26] Chao Xing, Yanjun Li, and Ke Zhang. Motion blur analysis based on image segmentation and blind deconvolution, 2010.

[27] Xu Yong, Zhu Ye, Quan Yuhui, and Ji Hui. Attentive deep network for blind motion deblurring on dynamic scenes. *Computer Vision and Image Understanding*, 205:103169, 2021. ISSN 1077-3142. doi: https://doi.org/10.1016/j.cviu.2021.103169. URL https://www.sciencedirect.com/science/article/pii/S1077314221000138.

[28] Maciej Zamorski, Adrian Zdobylak, Maciej Zięba, and Jerzy Świątek. Generative adversarial networks: Recent developments. In *International Conference on Artificial Intelligence and Soft Computing*, Artificial Intelligence and Soft Computing, pages 248–258. Springer International Publishing, 2019. ISBN 978-3-030-20912-4.

[29] Hang Zhao, Orazio Gallo, Iuri Frosio, and Jan Kautz. Loss functions for image restoration with neural networks. *IEEE Transactions on computational imaging*, 3(1):47–57, 2016. ISSN 2333-9403.