Detection and Blur-Removal of Single Motion Blurred Image using Deep Convolutional Neural Network

Diksha Adke
Dept. Of Electronics and Telecommunication, SIES Graduate School of Technology
Navi Mumbai, India
dikshaa@ieee.org

Atharva Karnik
Dept. Of Electronics and Telecommunication, SIES Graduate School of Technology
Navi Mumbai, India
atharvak@ieee.org

Honey Berman
Dept. of Electronics and Telecommunication, SIES Graduate school of Technology
Navi, Mumbai, India
berman.honey17@siesgst.ac.in

Shyamala Mathi
Dept. of Electronics and Telecommunication, SIES Graduate school of Technology
Navi Mumbai, India
shyamala.mathi@siesgst.ac.in

***Abstract*—This paper proposes a simple and efficient motion blur detection and removal method based on Deep CNN. The domain of computer vision has gained significant importance in recent years due to insurgence in the fields of self-driving cars, UAVs, medical image processing, etc. Due to low light conditions and the camera's fast motion, a large portion of image data generated is wasted. Such motion-blurred images impose a great challenge to the algorithms used for decision-making in machine vision. Although there have been significant improvements in denoising such image data, these methods are challenged by time constraints, insufficient data to train, reconstructed image quality, etc. The proposed paper employs a learning method to detect and deblur the single input image even in the absence of a ground-truth sharp image. We have used a synthetic dataset for experimental evaluation. This synthetic dataset that we have created and used for training the DCNN model has been made available for open source on Kaggle at the following link: https://www.kaggle.com/dikshaadke/motionblurdataset**

***Keywords*—Deep Convolutional Neural Network, Deep Learning, Supervised Learning, Computer Vision in Robotics, Motion Blur, Automation**

I. Introduction

Last few years have seen a surge in the technologies which rely on visual image inputs such as self-driving cars, UAVs/Drones, rescue-bots, etc. The major source of image degradation is image blur, and hence, deblurring has always been a crucial research area in the 'Computer Vision' and 'Robotics' domain. Motion-blur is the cause of tremendous losses in applications of algorithms like depth prediction, feature detection, motion estimation, object segmentation, etc. Hence, it has attracted significant attention in recent years. With a rapid progress in automation in every industry where many applications rely on imagery data, it is crucial that this sensory data is free of noise. Image deblurring is a process of obtaining latent sharp image by extracting and cancelling the noisy data elements from a blurry image. Due to randomness in the intensity and phase for noise (see Fig. 1), motion-blur is hard to remove. It occurs due to two main reasons, such as camera shake, poor lighting conditions, etc. Current deblurring methods can be categorized as follows:

A. Knowledge of Blur Kernel

Deblurring approaches can be categorized into 2 classes viz. 'Blind' and 'Non-blind' deblurring [1]-[4]. This classification is done based on the knowledge of blur kernel, i.e., whether the blur kernel is known or not. Preliminary knowledge about blur kernel and its parameter is essential for non-blind deblurring approaches. However, in the blind deblurring approach, it is assumed that the blurring operator is undetermined. It is not feasible to acquire the Point Spread Function (PSF) of the noisy blurred image in most cases. Hence, blind deblurring methods have a wider application range than non-blind deblurring methods. In the case of blind deblurring, current methods usually define the blur kernel of the entire image as a uniform function. As per the standard procedure of blind deblurring, the blur kernel is estimated prior to the non-blind deconvolution operation.

B. Spatial Analysis of Noise

A blurred image is either globally (uniform) blurred [9], [10], or locally (non-uniform) blurred. These methods consider that the blur kernels for all the pixels vary according to their spatial location, i.e., the camera is steady, and subjects are in random motion. Generally, due to multiple moving objects locally blurred images are commonly occurring. Globally (uniform) [1], [5]-[8] blurred deblurring methods assume an identical noisy Point-Spread-Function throughout the image, i.e., the camera is in motion, while subjects are steady. Inspired by motion deblurring using learning approaches [11], [12], we propose an architecture based on two models for classifying motion-blur images and deblurring them. We have considered a spatially uniform deblurring method for our paper. Our other contribution is a novel synthetic dataset that contains 33,316 labelled images, that we have used to train and evaluate our algorithm.

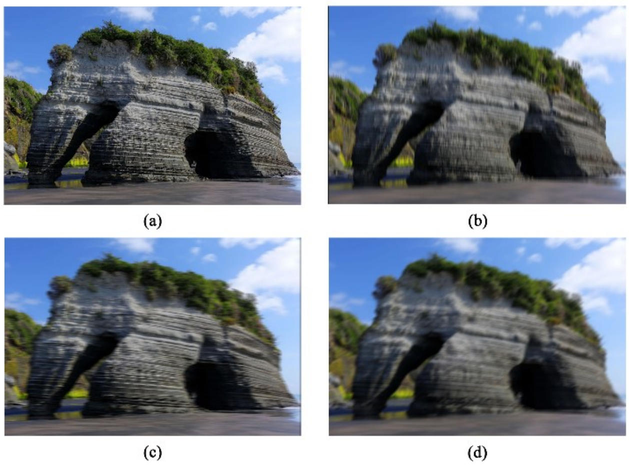

Fig. 1. (a) Sharp image (b) Vertically motion-blurred image (c) Horizontally motion-blurred image (d) Random motion-blurred image

Fig. 2. **Left**: Ground truth image. **Middle**: Blurry input image. **Right**: Deblurred result of the proposed method

II. Related Work

The recent years have observed tremendous work in blind image blur detection and deblurring. In this section, we focus on the two most prominent methods: an optimization-based method for deblurring the images and learning-based methods for blur detection and deblurring.

Optimization problem: In this approach, the gradient descent method is used for the optimization noisy Point-Spread-Function i.e., the blur kernel [1], [2], [3], [13], [14], [15]. The single most important advantage of these methods is the absence of ground truth image data. However, a large number of complex mathematical computations are required to obtained the latent denoised images. Thus, it curbs their application range in real-time robotic visual perception.

The second deblurring strategy considers the job as a **learning problem**. It uses convolutional neural networks (CNNs) to retrive the denoised sharp image. Integrated features for the neural network have been used for blur detection by [18]. The basic idea behind this approach is that, mostly, compared to sharp regions, the blur region is less affected due to low pass filtering. The edge ratio, the filter evolution-based descriptors, and the point spread function (PSF) serve as region definers for the neural networks. Such type of method which combines various characteristics leading to a discriminative feature for blur detection has been proposed by Liu *et al.* [19]. Another blur detection method executes multiple weak attributes to boost the performance of the final feature descriptor [20]. The target here is to calculate how blurry an image is by categorizing it as excellent, good, fair, poor, or bad. They have used eight features to feed the neural network.

Frequency domain metric, coherence, HVS based metric, local phase, mean brightness level, perceptual blur metric, the variance of the HVS frequency response, spatial domain metric, and contrast are these eight features. For this technique, results described that under most circumstances, the integrated features work superior to individual features.

Xu et al. [24] have constructed a convolutional network having two sub-nets, comprising of a deconvolution network and an outlier-rejection network. This network has been trained on already known sharp ground truth images. Further, [25] has proposed a deeper network consisting 15 layers for text image deblurring. For the single image deblurring task Cycle- GAN [26] which was proposed by Zhu et al. has been adapted by Madam et al. [27]. Zhu et al. have acquired satisfactory results specifically on text, face images by training the network on unpaired sharp images. however, this approach is unsuccessful on blurry images.

To deblur the image, this learning base approach is based on the recent advancements in DCNN. Some state-of-the-art methods have also been devised for single image deblurring [16], [17] which outperform optimization-based approaches when compared upon quality and efficiency. However, these methods need complete supervision in the form of corresponding pairs of blurred and sharp images. But, acquiring such pairs is tough because of few reasons, one of which arises due to limitation of the camera's frame rate. Not every camera is fast enough to capture video data at speeds like 900-1000 frames per second. Hence, it is very difficult to use such cameras which will generate the training data from the recorded frames. Secondly, it is difficult to get high-quality images in practical scenarios where a significant motion blur occurs (e.g., nighttime). A high frame-rate (HFR) is a restricting parameter in the shutter speed, and thus it may render the image invisible or highly underexposed.

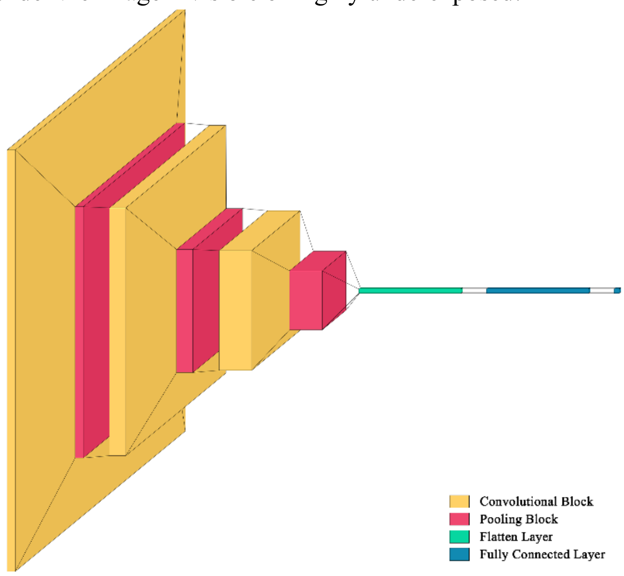

Fig. 3. Discriminator Model Visualization

III. METHODOLOGY

Overview: Motion blur input can degrade the precision of computer vision algorithms. Hence the main aim of our project is to detect the noisy images affected by the motion of the object or the camera and yield a deblurred image output. To achieve this purpose, we created an architecture consisting of a discriminator model based on a convolutional neural network and joined it with the second deconvolutional SRN network using the transfer learning approach. This discriminator would detect only motion-blurry images from input data stream and feed to deblurring network, thereby eliminating the need to process entire stream of data. We also created a big hybrid dataset to train our 1st model by processing and merging 2 datasets. The training of the network has been done on Nvidia GT710 GPU.

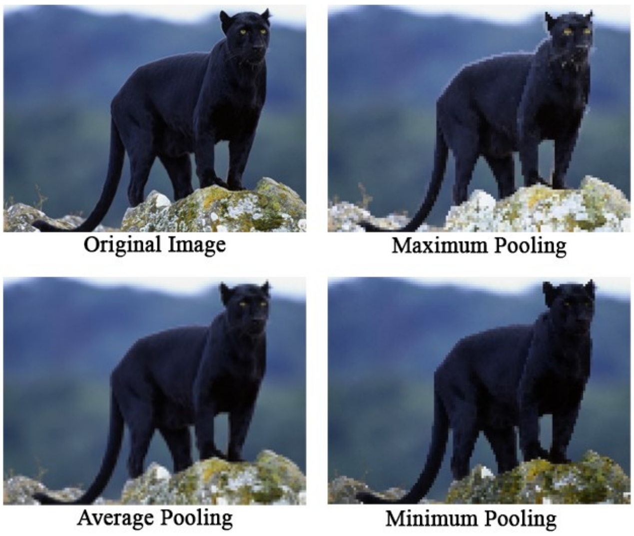

Fig. 4. Effect of Pooling

The 9-layered discriminator embodies Convolutional Neural Network of varying filters. A detailed illustration of the Discriminator model can be seen in Fig. 3. For the output layer of the discriminator, sigmoid activation function is employed, which yields better results in binary return values. For model optimization, Adam optimizer is used with binary cross entropy loss function. This sub-model is further discussed in 'Implementation Details'. The use of max pooling allows the model to trace the low-level features in the image like points, edges, etc. Note that initial experimentation was done using 'Average pooling' layers to conserve the overall illumination of image, however it reduced the frequency of image pixels as the layers progressed, resulting in longer training duration and reduced accuracy of model. A comparison of pooling examples can bee seen in Fig. 4.

$$H_p(q) = -\frac{1}{N}\sum_{i=1}^{N} y_i \cdot log\big(p(y_i)\big) + (1 - y_i) \cdot log\big(1 - p(y_i)\big)$$

… *(Binary Cross-entropy Loss Function)*

The second sub-model is a deconvolutional Scale-Recurrent Network [11][16] where transfer learning is used from a novel model developed by Peidong et al. The model comprises a ResBlock having LSTM architecture, followed by an Encoder-Decoder network. This type of architecture effectively transforms input image sequences into feature maps with image information, later decoded as an output file of size as that of the input image, but higher S/N (Signal-to-Noise) ratio. There are many existing neural network architectures [16],[17],[21] that have performed well in supervised learning tasks. Here, the recurrent neural nets are used to carry the information from adjacent input images as sequences, resulting in better training efficiency and improved deblur performance on unseen data. It also eliminates the need for ground-truth images while training the network. Hence, proposed method can be used to deblur individual frames. This sub-model is further discussed in 'Implementation Details'.

Fig. 2 shows the results of our network on 2 different images. Images on left, and middle are sharp ground truth, and motion blurry images respectively from GoPro dataset. On the right side, the denoised output of our network is shown. As the noise level varies spatially in blurry images, the denoised output inevitably retains few blurry regions. However from PSNR evaluation, which is discussed in Table. 1, and few random noisy images from internet (Fig. 6), it can be seen that the output images will perform much better on any segmentation or pattern recognition, etc. tasks.

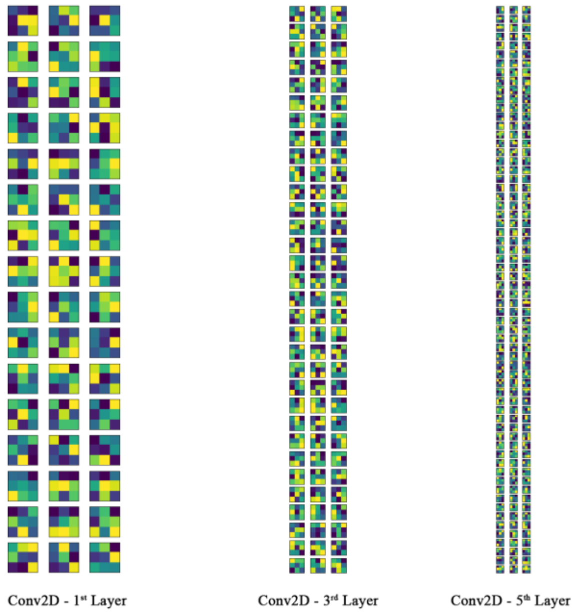

Fig. 5. Trained Filters Visualization

IV. EXPERIMENTATION

A. Datasets

Data preparation: As the most common confluence of cameras and motion happens on the roads, we decided to include cars images in the bigger synthetic dataset. The 'Cars-196' dataset provided by Stanford University AI Lab [23] contains 16,185 images of 196 classes of cars. This open-source dataset is also available on TensorFlow. This data is split into 8,144 training images and 8,041 testing images, where each class has been split roughly in a 50-50 split. Classes are typically at the level of 'Make, Model, Year', e.g., '2008 BMW M3 Coupe' or '2010 Tesla Model 3', denoted by the name of the folder.

We have modified the aforementioned original dataset by image convolution with the order of blur kernels ranging uniformly from 6 to 20. The motion blur applied is of uniform

Fig. 6. Noisy input image and reconstructed deblurred image.

distribution for all directions. This Dataset contains 32,370 images, with around 50-50 split for 16,290 images paired as ‘sharp’ & ‘motion-blurred’ for training and 16,080 images paired in the same manner for testing.

As the above-mentioned datasets contains only cars data, for better accuracy on universal images, we mixed this dataset with a smaller, high quality dataset by Aleksey Alekseev. This dataset comprises a total of 1050 images. All of these images are in a form of a triplet, as follows: a sharp image, its corresponding motion-blurred, and a defocused image. We have preprocessed this dataset before blending it with the previously mentioned dataset. The Discriminator model has been trained on this dataset.

Therefore, we created a new large dataset with the parameters of images to be of either motion-blur or non-motion-blur class. Images created using blur kernel from Cars Dataset and the motion-blurred images from the smaller dataset are separated into ‘motion-blur’ class while the rest of the sharp and defocused images are part of the ‘non-motion-blur’ class.

B. *Implementation Details*

We implemented our network in TensorFlow 2.4.1 and Pytorch 9.0. We created a sequential CNN model with a total of 40,165,409 parameters. The objective of this model is to decide whether an input image is motion-blurred or not. As the task of this model is to classify the given input image into 2 classes. the model uses Binary Cross-entropy loss function. Adam optimizer has been used for fitting the data to model. In particular, we trained it for 30 epochs, having a 10^{-4} learning rate. The model achieved 98% classification accuracy in training with 87% validation accuracy. A visual representation of the trained filter weights can be seen in Fig. 5. The filters are designed for 3-channel RGB images with number of filters increasing as the network progresses. The kernel size of filters throughout the network is kept as 3*3. For the 1st layer, 16 filters are applied, which become 32 and 64 for the 3rd and 5th layer respectively. Max pooling layers of size 2*2 are used between the convolutional filters. The 78,400 flattened neurons of the last pooling layer is connected to a fully connected dense neural network of size 512, followed by the output neuron. Sigmoid activation function is used for the output neuron, while remaining all layers employ a faster function ‘Rectified Linear Unit’ or ‘ReLU’.

The results of the first sub-model are then fed to the second sub-model from [11], which is pretrained on the Fastec Dataset to complete the network implementation. It includes an SRN-based DeblurNet and a PWC-net which enables the model to train on data without ground-truth images. It is developed from ground work by Tao *et al.* [16] and is much more time-efficient than the former model. However, it should be noted that deblurring performance is slightly reduced in the later one.

C. *Evaluation Metrics*

Peak-Signal-to-Noise Ratio (PSNR) is a qualitative measure of the maximum strength of the useful signal in a noisy stream of data. As the name suggests, it is the ratio of peak signal strength to the strength of noise in the received or reference data. It is widely used as a metric in image processing applications as it states the quality of image pair based on the differences in pixel values between those images. Thus, it can estimate the quality of reconstructed image in with respect to the original or ground truth image. Secondly, it is very easy to calculate the PSNR due to its low time complexity, which makes it an easier and efficient method in quality assessment for various computer vision applications. The PSNR value is calculated as follows:

$$PSNR = 10\, log_{10}\left[\frac{(L-1)^2}{MSE}\right]$$

Here, L denotes the maximum intensity levels of image i.e., number of bits used to encode 1 pixel value. For most computational applications, $(L\text{-}1)$ comes out to be 255, considering 8-bit image engines. MSE is the ‘Mean Squared Error’ of the image pair, which is further calculated as,

$$MSE = \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} [O(i,j) - R(i,j)]^2$$

Where, O denotes the original image and R denotes the reconstructed image having m columns & n rows.

Table 1. Comparative analysis of results on GoPro Dataset

Sr. No.	Method	PSNR (dB)
1.	DeblurGAN [21]	31.10
2.	SRN-Deblur [16]	30.26
3.	RNNDeblur [22]	29.18
4.	*Proposed method*	30.41

A higher PSNR value signifies a higher deblurred image quality. We have used this parameter to examine the quality of the result. Table 1 shows the comparison of PSNR values by various methods, tested on the GoPro dataset. It is clear from the table that the DeblurGAN method by Kupyn *et al.* employing a probabilistic ‘Wasserstein metric’, is highly effective for image deblurring tasks. The RNNDeblur method by Zhang *et al.* based on IIR (infinite impulse response) approximation, is slightly inferior to all the other methods.

The proposed model has given the PSNR value of 30.41 dB which is comparatively better than SRN-Deblur method.

V. Conclusion

In this paper, we have discussed an algorithm based on a learning method that detects and deblur the motion blur image. The model successfully works on a single input image without requiring the ground truth sharp image. Experimental results (Fig. 2, Fig. 6) have shown the successful working of the model, which can be fed to various computer vision algorithms. It can be seen from Table 1 that, our proposed model has performed better in terms of PSNR value. Thus, denoised images from this model would be more useful to various image processing algorithms. The presence of a recurrent architecture in deblurring part signifies that this model can work on real-time, unseen motion blurry data input.

References

[1] D. Krishnan and R. Fergus, “Fast image deconvolution using hyper-Laplacian priors,” in Proc. Conf. Adv. Neural Inf. Process. Syst., Vancouver, BC, Canada, 2009, pp. 1–9.

[2] R. Fergus, B. Singh, A. Hertzmann, S. T. Roweis, and W. T. Freeman, “Removing camera shake from a single photograph,” ACM Trans. Graph., vol. 25, no. 3, pp. 787–794, Jul. 2006.

[3] S. Cho and S. Lee, “Fast motion deblurring,” in Proc. ACM SIGGRAPH Asia, Yokohama, Japan, 2009, Art. no. 145.

[4] D. Zoran and Y. Weiss, “From learning models of natural image patches to whole image restoration,” in Proc. IEEE Int. Conf. Comput. Vis., Barcelona, Spain, Nov. 2011, pp. 479–486.

[5] S. Cho and S. Lee, “Fast motion deblurring,” ACM Trans. Graph. (SIG-GRAPH ASIA), vol. 28, no. 5, 2009, Art. no. 145.

[6] R. Fergus, B. Singh, A. Hertzmann, S. T. Roweis, and W. T. Freeman, “Removing camera shake from a single photograph,” in Proc. ACM Trans. Graph., 2006, pp. 787–794.

[7] W. Richardson, “Bayesian-based iterative method of image restoration,” J. Opt. Soc. Amer., vol. 62, no. 1, pp. 55–59, 1972.

[8] Q. Shan, J. Jia, and A. A, “High-quality motion deblurring from a single image,” ACM Trans. Graph., vol. 27, no. 3, pp. 73:1–73:10, 2008.

[9] A. Gupta, N. Joshi, C. L. Zitnick, M. Cohen, and B. Curless, “Single image deblurring using motion density functions,” in Proc. Eur. Conf. Comput. Vision, 2010, pp. 171–184.

[10] O. Whyte, J. Sivic, A. Zisserman, and J. Ponce, “Non-uniform deblurring for shaken images,” in Proc. Conf. Comput. Vision Pattern Recognit., 2010, pp. 491 498.

[11] P. Liu, J. Janai, M. Pollefeys, T. Sattler and A. Geiger, "Self-Supervised Linear Motion Deblurring," in IEEE Robotics and Automation Letters, vol. 5, no. 2, pp. 2475-2482, April 2020, doi: 10.1109/LRA.2020.2972873.

[12] R. Yan and L. Shao, "Blind Image Blur Estimation via Deep Learning," in IEEE Transactions on Image Processing, vol. 25, no. 4, pp. 1910-1921, April 2016, doi: 10.1109/TIP.2016.2535273.

[13] L. Xu and J. Jia, “Two-phase kernel estimation for robust motion deblur- ring,” in Proc. Eur. Conf. Comput. Vision, 2010, pp. 157–170.

[14] S. Zheng, L. Xu, and J. Jia, “Forward motion deblurring,” in Proc. IEEE Int. Conf. Comput. Vision, 2013, pp. 1465–1472.

[15] H. Zhao, H. Yang, H. Su and S. Zheng, "Natural Image Deblurring Based on Ringing Artifacts Removal via Knowledge-Driven Gradient Distribution Priors," in IEEE Access, vol. 8, pp. 129975-129991, 2020, doi: 10.1109/ACCESS.2020.3007972.

[16] X. Tao, H. Gao, Y. Wang, X. Shen, J. Wang, and J. Jia, “Scale-recurrent network for deep image deblurring,” in Proc. IEEE Conf. Comput. Vision Pattern Recognit., Jul. 2018, pp. 8174–8182.

[17] S. Nah, T. H. Kim and K. M. Lee, "Deep Multi-scale Convolutional Neural Network for Dynamic Scene Deblurring," 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 257-265, doi: 10.1109/CVPR.2017.35.

[18] J. Rugna and H. Konik, “Automatic blur detection for metadata extraction in content-based retrieval context,” in Proc. SPIE Internet Imag. V, vol. 5304. San Diego, CA, USA, 2003.

[19] R. Liu, Z. Li, and J. Jia, “Image partial blur detection and classification,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Anchorage, AK, USA, Jun. 2008, pp. 1–8.

[20] A. Ciancio, A. L. N. T. da Costa, E. A. B. da Silva, A. Said, R. Samadani, and P. Obrador, “No-reference blur assessment of digital pictures based on multi-feature classifiers,” IEEE Trans. Image Process., vol. 20, no. 1, pp. 64–75, Jan. 2011.

[21] O. Kupyn, V. Budzan, M. Mykhailych, D. Mishkin, and J. Matas, “Deblurgan: Blind motion deblurring using conditional adversarial networks,” in Proc. IEEE Conf. Comput. Vision Pattern Recognit., Jul. 2018, pp. 8183–8192.

[22] J. Zhang et al., "Dynamic Scene Deblurring Using Spatially Variant Recurrent Neural Networks," 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018, pp. 2521-2529, doi: 10.1109/CVPR.2018.00267.

[23] Jonathan Krause, Michael Stark, Jia Deng, Li Fei-Fei, "3D Object Representations for Fine-Grained Categorization," 4th IEEE Workshop on 3D Representation and Recognition, at ICCV 2013 (3dRR-13). Sydney, Australia. Dec. 8, 2013.

[24] L. Xu, J.-S. Ren, C. Liu, and J. Jia, “Deep convolutional neural network for image deconvolution,” in Proc. Adv. Neural Inf. Process. Syst., 2014, pp. 1790–1798.

[25] M. Hradis, J. Kotera, P. Zemcik, and F. Sroubek, “Convolutional neural networks for direct text deblurring,” in Proc. Brit. Mach. Vision Conf., 2015, pp. 6.1–6.13.

[26] J. Y. Zhu, T. Park, P. Isola, and A. A. Efros, “Unpaired image-to-image translation using cycle-consistent adversarial networks,” in Proc. IEEE Int. Conf. Comput. Vision, 2017, pp. 2242–2251.

[27] N. T. Madam, S. Kumar, and A. N. Rajagopalan, “Unsupervised classspecific deblurring,” in Proc. Eur. Conf. Comput. Vision, 2018, pp. 358–374.