Deep Generative Filter for Motion Deblurring

Sainandan Ramakrishnan[1], Shubham Pachori*[2], Aalok Gangopadhyay*[2], Shanmuganathan Raman[2]
Veermata Jijabai Technological Institute, Mumbai - 400031[1]
Indian Institute of Technology, Gandhinagar - 382355[2]
saiyanlife415@gmail.com[1], {shubham_pachori, aalok, shanmuga}[2] @iitgn.ac.in

Abstract

Removing blur caused by camera shake in images has always been a challenging problem in computer vision literature due to its ill-posed nature. Motion blur caused due to the relative motion between the camera and the object in 3D space induces a spatially varying blurring effect over the entire image. In this paper, we propose a novel deep filter based on Generative Adversarial Network (GAN) architecture integrated with global skip connection and dense architecture in order to tackle this problem. Our model, while bypassing the process of blur kernel estimation, significantly reduces the test time which is necessary for practical applications. The experiments on the benchmark datasets prove the effectiveness of the proposed method which outperforms the state-of-the-art blind deblurring algorithms both quantitatively and qualitatively.

1. Introduction

Motion blur is a common problem which occurs predominantly when capturing an image using light weight devices like mobile phones. Due to the finite exposure interval and the relative motion between the capturing device and the captured object, the image obtained is often blurred. In [19], it was shown that standard network models, trained only on high-quality images, suffer a significant degradation in performance when applied to those degraded by blur due to defocus or subject/camera motion. Thus, there is a serious need to tackle the issue of blurring in images. Blur induced due to motion in images is spatially non-uniform and the blur kernel is unknown. Due to depth variation, the segmentation boundaries of the objects and the relative motion between the camera and scene objects, estimating spatially variant non-uniform kernel is quite difficult. In this paper, we introduce a generative adversarial network (GAN) based deep learning architecture to address this challenging problem. We obtain significantly better results than the state-of-the-art algorithms proposed to solve the problem of image deblurring.

2. Related Work

Most of the previous works in the literature tackle the problem of camera deshaking by modelling it as a blind deconvolution problem and using image statistics as priors or regularizers to obtain the blur kernels. While these methods have achieved great success in benchmark datasets, restrictive assumptions in their methods and algorithms limit their practical applicability. Also, most of these works in the literature have been dedicated to solve the problem of blind deconvolution assuming the blur kernel to be spatially uniform. Very few works have been proposed to solve this challenge by taking spatially varying blur kernel. To tackle the problem of non-uniform blind deblurring, previous works divide the image into smaller regions and estimate the blur kernels for each region separately [4]. Once the kernels are obtained for each of the local regions in the image, they are then deblurred and combined using OLA (Overlap Add) method to generate the final deconvolved image. Proposed works which exploit deep learning methods first try to predict the probabilistic distribution of motion blur information in a small region of the given image and then try to utilize this blurring observation to recover the sharp image [18]. Only one work to the very best of our knowledge has attempted to directly recover the sharp image from the given blurred image [16]. However, it is computationally expensive as authors exploit multi-scale framework to obtain the deblurred image. Therefore, we aim to recover the artifact-free image directly without using the multi-scale framework. An exhaustive survey of blind deblurring algorithms can be found in [11].

3. Proposed Method

In our model, we enable every convolutional unit in the deep network to make independent decisions based on the entire array of lower level activations. Unlike [12] and [16] which use residual blocks as primary workhorses through

* denotes the equal contribution.

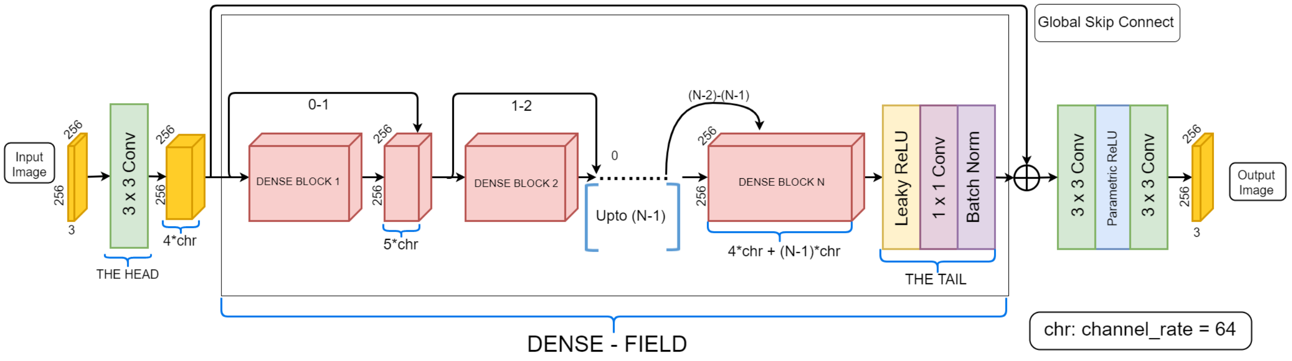


Figure 1. Our Convolutional Neural Network Architecture.

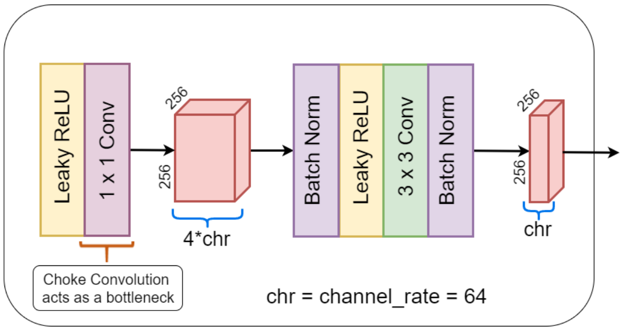


Figure 2. Structure inside our Dense Block.

element-wise summation of lower level activations with higher level outputs, we want information from different semantic levels to flow unaltered throughout the network. To achieve this, we propose a densely connected 'generative network'.

3.1. Model Architecture

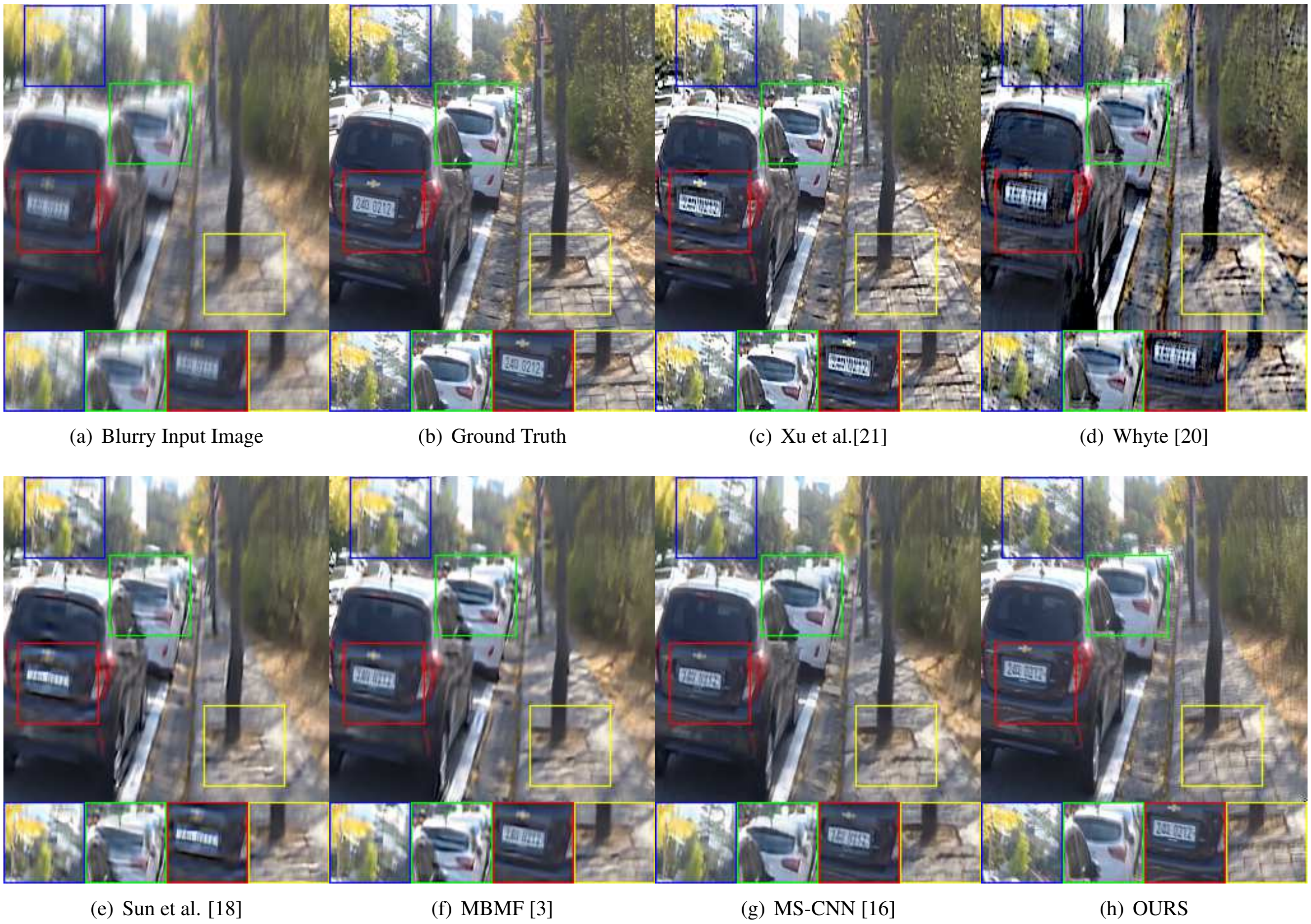
Our architecture consists of a densely connected generator and a discriminator. The task of the generator is to recycle features spanning across multiple receptive scales to generate an image that fools the discriminator into thinking that the generated image came from the target distribution. Thus, we can generate visually appealing and statistically consistent deblurred image given a blurred image. The task of the discriminator is to correctly identify from which distribution each of its input images came from by analysing different patches in each image to make a decision. We elaborate both our generator and discriminator models in detail.

3.1.1 The Generator

Unlike [6], we do not reduce the dimension of the information and keep it constant throughout the network. While this does give rise to memory constraints, it protects the network from generating checkerboard artifacts found commonly in networks relying on deconvolution to generate visually appealing images [7]. Instead, through feature reuse across all levels in the generator network, our model exhibits high generation performance with a much smaller network depth than the other CNN-based methods used for non-uniform motion deblurring [16],[3],[18]. This enables smoother training, faster test time and allows efficient memory usage. Our generator model as shown in Fig. 1 consists of 4 parts which are the head, the dense field, the tail, and the global skip connection. We describe each of them in detail below.

a) The Head: We define the hyper-parameter 'channel-rate' (chr) as the constant number of activation channels that are output by each convolutional layer. The value of channel-rate is 64. The head comprises of a simple 3×3 convolutional layer which convolves over the raw input image and outputs $4\times$channel-rate (256) feature activations. This provides sufficient first-level activation maps to trigger the densely connected stack of layers.

b) The Dense Field: This section consists of N number of convolutional 'blocks' placed sequentially one after the other, all having their outputs fully connected with the output of the layer ahead of them. The dense connection is efficiently achieved in practice by concatenating output activation maps of every ith layer in the dense field with the output maps of $(i+1)$th layer. Hence, the number of activation maps input to the mth dense block will be equal to '$4\times$chr $+ (m-1)\times$chr'. The structure of a dense block is shown in Fig. 2. The first operation is a Leaky ReLU [15] which not only adds non-linearity to the incoming activations but also avoids using sparse gradients which could compromize GAN training stability. The 1×1 convolution 'chokes' the number of activation maps being convolved later to a maximum equal to '$4\times$chr'. This conserves parameter and data memory in the deeper layers of the dense field when the number of raw activation channels entering will be $6\times$chr



(a) Blurry Input Image (b) Ground Truth (c) Xu et al.[21] (d) Whyte [20]

(e) Sun et al. [18] (f) MBMF [3] (g) MS-CNN [16] (h) OURS

Figure 3. Comparison of deblurred images by our model and other algorihtms on one of the images taken from GoPro dataset [16].

Method	PSNR (dB)	SSIM	MS-SSIM	F-SIM	UIQI	IFC	VIF
Ours(A)	28.0345	0.8895	0.9678	0.8943	0.9612	4.0904	0.8691
Ours(B)	28.5798	0.9090	0.9701	0.9132	0.9683	4.2458	0.8749
Ours(final)	**28.9423**	**0.9220**	**0.9720**	**0.9248**	**0.9741**	**4.9455**	**0.8853**

Table 1. GoPRO Test Dataset (Ablation study on generative dense-net architecture), Ours(A): Residuals at extremes, dense in the middle, Ours(B): Dense across extremes, successive residuals in the middle.

Method	MBMF [3]	MS-CNN [16]	OURS
Time	0.72 sec	2.2 sec	**0.3** sec

Table 2. Average time to deblur the input image of size $256 \times 256 \times 3$.

(384) or more. The convolution at the final layer of each dense block uses ‘chr’ number of $3 \times 3 \times (4{\times}\text{chr})$ filters, giving rise to ‘chr’ number of activation maps at the end of each dense block. The 3×3 convolutions along the dense field are alternated between ‘spatial’ convolution and ‘dilated’ convolution with linearly increasing dilation factor [22]. We use dilated convolution [22] at every even numbered layer within the dense field. We have the dilation factor increasing linearly to a maximum till the centre of the dense field and then symmetrically reducing till we arrive at the tail. This helps to increase the receptive field at an exponential rate with every layer while the parameter space increases linearly and hence introduces higher disparity between the multiple scales of activation maps that arrive at subsequent dense layers. We avoid pooling and strided convolution operations to keep the dimensions of the output maps to be constant and equal to the image size throughout the network. Adding dropout at the end of each block helps us effectively add Gaussian noise to the input of each layer in the generator (G) which prevents the GAN collapse problem by enabling G to blindly model shake distributions other than a pure delta distribution.

c) The Tail: The Tail adds the non-linearity and through

Method	PSNR (dB)	SSIM	MS-SSIM	F-SIM	UIQI	IFC	VIF
ResGAN [12]	24.3460	0.7678	0.8697	0.8352	0.9715	2.1568	0.7043
Pix2Pix [7]	24.5987	0.7692	0.8680	0.8379	0.9675	2.0354	0.6992
Ours(1)	24.5281	0.7625	0.8551	0.8113	0.9421	1.9805	0.6835
Ours(2)	24.5412	0.7656	0.8602	0.8310	0.9455	2.0051	0.6981
Ours(3)	24.6991	0.7677	0.8681	0.8354	0.9532	2.1143	0.7038
Ours(4)	25.4897	0.7718	0.8694	0.8417	0.9679	2.3875	0.7315
Ours(5)	26.8134	0.8081	0.8840	0.8733	0.9758	2.5892	0.7581
Ours(final)	**27.0812**	**0.8362**	**0.9112**	**0.8936**	**0.9778**	**2.9348**	**0.7740**

Table 3. Quantitative Comparison of Progressive Model with Benchmarks on Synthetically blurred Places Dataset. Ours(1): Without Perceptual Loss, Ours(2): Without GAN (with (1)), Ours(3): Without conditional GAN (with(1,2)), Ours(4): Without global skip connection (with(1,3)), Ours(5): Without dilated convolution (with(1,3,4)) and Ours(final): with(1,3,4,5).

Method	PSNR (dB)	SSIM	MS-SSIM	F-SIM	UIQI	IFC	VIF	Norm-NR
Xu et al.[21]	25.1858	0.8960	0.9614	0.9081	0.9527	4.1811	0.8644	0.9570
Sun et al. [18]	24.6890	0.8561	0.9308	0.8691	0.9427	4.1132	0.8430	0.9532
MBMF [3]	27.1989	0.9082	0.9617	0.9138	0.9450	4.2032	0.8699	0.9581
MS-CNN [16]	28.4496	0.9165	**0.9729**	0.9073	0.9693	4.1969	0.8752	**0.9657**
Ours (final)	**28.9423**	**0.9220**	0.9720	**0.9248**	**0.9741**	**4.9455**	**0.8853**	0.9642

Table 4. Quantitative Comparison of our method with other state-of-the-art blind deblurring algorithms on GoPro Dataset.

1×1 convolution increases the number of feature maps to $4\times$chr.

d) The Global Skip Connection: Deep generative CNNs usually face the problem of often inadvertently memorizing high level representations of edges as it is non-trivial to generalize over first-level features using several convolution operations. This would lead the network to not be able to retrieve sharp boundaries at correct locations from the shaken images. We concancate the output from the head of the network with the output of the tail. This gives rise to a good improvement in generation performance because the gradients can now flow from the tail straight to the first level convolutional layer and impact the update in the lower layers [5]. But more importantly, this single connection 'drives' the entire dense field in the centre to expend its 'full knowledge' of the image towards understanding the residual between the ground truth and the blurred images. Meanwhile, it also optimizes gabor-like features of our CNN directly from the ground truth fed into the generator-end [23]. However, different from the traditional residual networks used in image restoration models, we do not use cascaded skip summations. Instead, we pass lower level knowledge to the upper layers through dense connection and direct the entire dense field to solely calculate the global residual, which as experiments show, enable our network to learn faster, achieve better convergence and show significantly better deblurring performance.

3.1.2 The Discriminator

In our GAN framework, the discriminator is the primary agent which guides the statistics that the generator employs to create restored images. Moreover, we do not want the depth of the discriminator network depth so much that it memorizes the easier task of classification. We employ a Markovian patch discriminator [13] with 10 convolutional layers, which is similar to a non-overlapping sliding window that tends to look for well-defined, structural features at several local patches. This also enforces rich coloration in the generated natural images [7].

3.2. Loss Functions

a) ℓ_1 **and Adverserial Loss**: Traditionally, learning-based image restoration works have used ℓ_1 or ℓ_2 loss between the ground truth and the rectified image as the chief objective function [1]. In case of an adverserial framework used for such a purpose [16], this loss is pooled with the adverserial loss which measures how well the generator is performing with respect to fooling the discriminator. However, using $\ell_1$loss solely in deep CNN models leads to overly smooth images, as pixel-wise error functions tend to converge at the mean of all possible solutions in the image manifold, whenever they encounter uncertainty [1]. This creates dull images with not many sharp edges and most importantly, with the blur still largely intact at edges and corners. At the same time, solely using adverserial loss does retain edges and gives rise to a more realistic color distribution [1]. However, it compromises on two things: it still has no abstract idea of structure and it only has the discriminator judging generator performance based on the output image alone with no regard to the blurred input. We remove these limitations by leveraging perceptual loss and adding it to the net loss function given in Eqn. 4.

b) **Perceptual Loss:** We need to augment structural knowl-

Method	PSNR (dB)	SSIM	MS-SSIM	F-SIM	UIQI	IFC	VIF	Norm-NR
Xu et al.[21]	25.95	0.7474	**0.8358**	0.8309	**0.9563**	2.4140	0.7478	**0.9271**
Whyte [20]	24.41	0.7312	0.8033	0.8293	0.9524	2.3910	0.7298	0.9103
Sun et al.[18]	24.58	0.7379	0.8059	0.8255	0.9393	2.3897	0.7303	0.9087
MBMF [3]	25.87	0.7420	0.8157	0.8136	0.9418	2.4385	0.7398	0.9201
MS-CNN [16]	26.79	0.7572	0.8168	0.8311	0.9535	2.4211	0.7317	0.9128
Ours (final)	**27.23**	**0.7651**	0.8217	**0.8712**	0.9538	**2.6158**	**0.7597**	0.9214

Table 5. Quantitative Comparison of our method with other state-of-the-art blind deblurring algorithms on Lai Dataset.

Method	PSNR (dB)	SSIM	MS-SSIM	F-SIM	UIQI	IFC	VIF	Norm-NR
Xu et al.[21]	**27.47**	0.7506	0.8115	**0.8810**	0.9642	2.5025	0.7698	0.9309
Whyte [20]	27.03	0.7467	0.8091	0.8802	0.9589	2.4556	0.7632	0.9287
Sun et al. [18]	25.12	0.7281	0.7748	0.7990	0.9401	2.1963	0.7267	0.9108
MBMF [3]	26.59	0.7418	0.8079	0.8741	0.9576	2.2407	0.7421	0.9221
MS-CNN [16]	26.51	0.7432	0.8083	0.8481	0.9587	2.2235	0.7298	0.9224
Ours (final)	27.08	**0.7510**	**0.8120**	0.8743	**0.9651**	**2.5192**	**0.7718**	**0.9318**

Table 6. Quantitative Comparison of our method with other state of the art blind deblurring algorithms on Köhler Dataset.

edge into the generator to counter the patch-wise judgement of the Markovian discriminator. One such loss function, as introduced in [8] is the Euclidean difference between deep convolutional activations of the ground truth and generated latent image which is also known as 'perceptual loss'. This loss term is given in Eqn. 1,

$$\mathcal{L}_{percep(VGG/i,j)} = \frac{1}{W_{i,j}H_{i,j}} \sum_{x=1}^{W_{i,j}} \sum_{y=1}^{H_{i,j}} (\phi_{i,j}(I^{\text{Groundtruth}})_{x,y} - \phi_{i,j}(G_{\theta_G}(I^{\text{Blurred}}))_{x,y})^2 \tag{1}$$

Here, $W_{i,j}$, $H_{i,j}$ are the width and height of the $(i,j)^{th}$ ReLU layer of VGG-16 network [17] and $\phi_{i,j}$ is the forward pass through VGG-16 network upto ReLU 3_3 layer.

3.2.1 Conditional adversarial framework

We feed two image pairs into the discriminator in our GAN framework. One pair consists of the input blurred image and the corresponding output image generated by the generator, whereas the other pair consists of the input blurred image and the corresponding ground truth deblurred image. This converges with the generator modelling the conditional distribution of the latent image, given the input image, a result that will help the generated images maintain high statistical consistency between a given input and its output. This is essentially what we need, because we want 'G' to maintain the output's dependency on the blurred input to accomodate different kinds and amounts of shake blur and prevent it from swaying too far away in its effort to fool the discriminator. Hence, we can view a conditional GAN as a 'relevance regularizer' in an image to image network. Mathematically, this would change the original GAN optimization problem used in our task which would be given by:

$$\min_{\theta_G} \max_{\theta_D} \mathbb{E}_{I^{\text{Groundtruth}} \sim p_{\text{train}}(I^{\text{Groundtruth}})}[\log D_{\theta_D}(I^{\text{Groundtruth}})] + \mathbb{E}_{I^{\text{Groundtruth}} \sim p_{\text{train}}(I^{\text{Groundtruth}})}[\log(1 - D_{\theta_D}(G_{\theta_G}(I^{\text{Blurred}}))] \tag{2}$$

to a conditional loss function which needs to be minimized, given by

$$\mathcal{L}^{\text{Generator}}_{\text{ConditionalGAN}} = -\mathbb{E}_{I\epsilon(I^{\text{Blurred}})}\left[\log D_{\theta_D}(G_{\theta_G}(I^{\text{Blurred}})|I^{\text{Blurred}})\right] \tag{3}$$

Thus, the combined loss function for our network is,

$$\mathcal{L}_{net} = \mathcal{L}^{\text{Generator}}_{\text{ConditionalGAN}} + (K_1)\mathcal{L}_{percep} + (K_2)\mathcal{L}_{L_1} \tag{4}$$

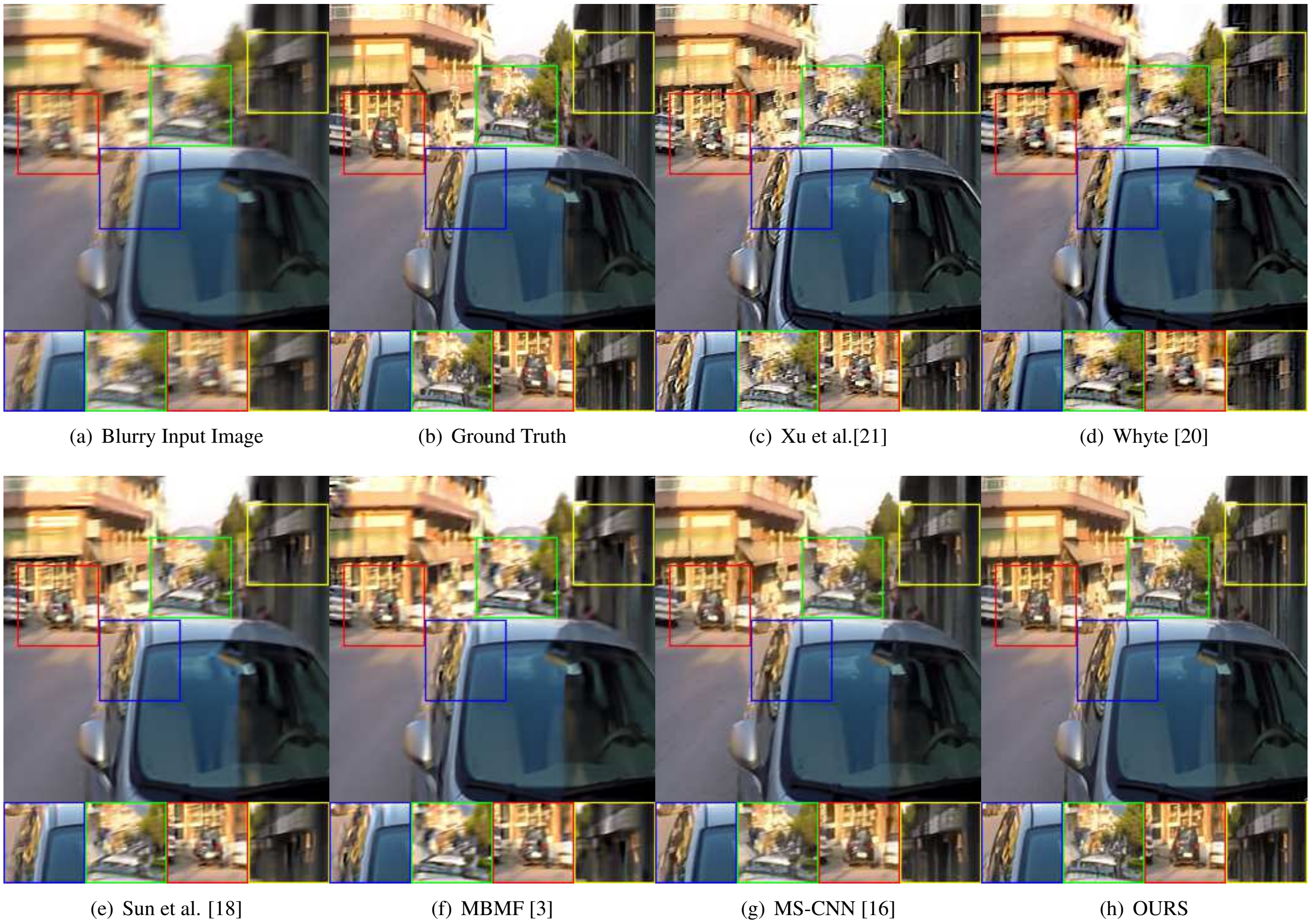
where, K_1 and K_2 are hypermeters which are set to 145 and 170 respectively in our experiments. From Table 3, we notice a significant boost in the performance across all metrics by introducing this technique. At this stage, our network has already outperformed the two baseline models modified and trained for our task: a very-deep, sequential ResNet model used by [12] and the hourglass, U-net model used by [7]. It is worth noting that our dense model with much fewer layers (10 dense blocks) not only outperformed, but also converged faster than the model in [12] with 15 residual blocks, showing that our model and the framework resonate much better.

4. Experiments

4.1. Experimental Settings

We implemented our model with torch7 library. All the experiments were performed on a workstation with i7 processor and NVIDIA GTX Titan X GPU.

Network Parameters: We optimize our loss function



(a) Blurry Input Image

(b) Ground Truth

(c) Xu et al.[21]

(d) Whyte [20]

(e) Sun et al. [18]

(f) MBMF [3]

(g) MS-CNN [16]

(h) OURS

Figure 4. Comparison of deblurred images by our model and other algorihtms on one of the images taken from GoPro dataset [16].

through the ADAM scheme [9] and converge it using stochastic gradient descent (SGD). Throughout the experiments, we kept the batch-size for training as 3 and fixed base learning rate and momentum to 0.0002 and 0.9 respectively. Similar to [7], we use instance normalization instead of training batch statistics during test-time.

Experiments for further architectural considerations : We also perform a simple ablation study over the architecture of our fully evolved model to isolate which connections in the dense net are more important towards image restoration to further explore our own model. We use two subdense architecures named 'A' and 'B' to do so. The results of the ablation studies are given in the Table 1.

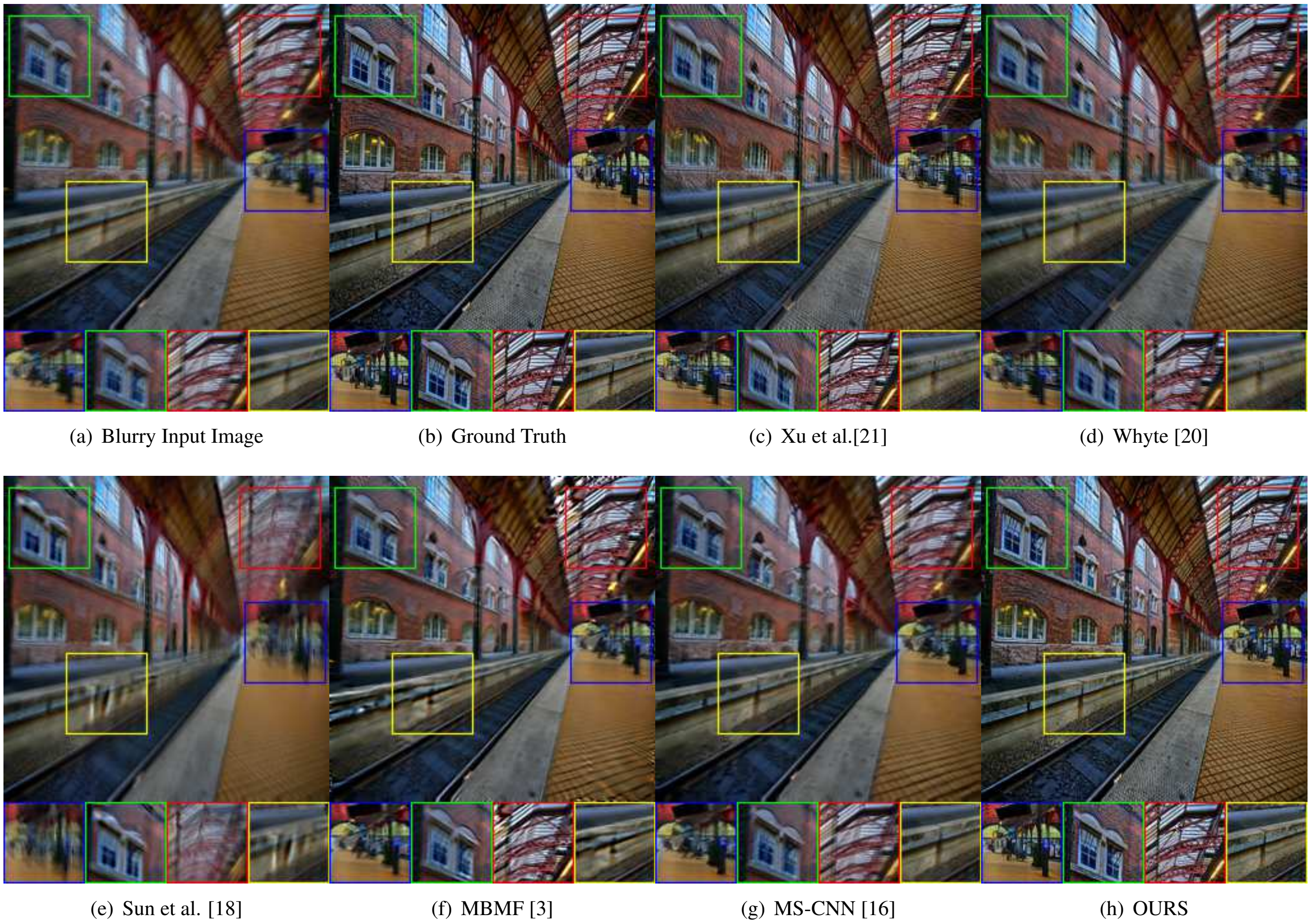
A) In this model, the three lower and higher extreme layers of our 'dense field' are replaced with successive residuals of [12] and [16] while the middle layers remain dense. We noticed a significant drop in performance compared to our final model by doing so. This is because the central part has 'forgotten' entry-level features which were crucial in calculating the global residual between the head and the tail.

B) Switching the locations of the residuals and the dense layers leads to better performance than having both a fully residual network [12] and a centrally dense network (A). Although it is slightly outperformed by our final model, it saves a dramatic amount of GPU memory by cutting down a lot of data concatenation. Hence, dense connections work best when connections between the farthest of layers is achieved. This helps the network to keep recycling lower features for globalizing the knowledge of the higher layers.

4.2. Datasets

To train our model, we extracted patches of size $256 \times 256 \times 3$ from GoPRO dataset and combined them with the images sampled randomly from MS-COCO [14] and Imagenet dataset [2] (which are resized to $256 \times 256 \times 3$) to generate our training dataset. We then apply non-uniform blurs similar to [11] on images sampled from MS-COCO and ImageNet datasets. We also perform data augmentation by using translational and rotational flipping, thus producing a final dataset consisting of 0.5 million training image pairs of blurred and deblurred images.

We perform comparison of progressive models on one


(a) Blurry Input Image (b) Ground Truth (c) Xu et al.[21] (d) Whyte [20]

(e) Sun et al. [18] (f) MBMF [3] (g) MS-CNN [16] (h) OURS

Figure 5. Comparison of deblurred images by our model and other algorihtms on one of the images taken from Lai et al.'s Dataset [11].

dataset generated synthetically by us and compare the performance of our method with the other state-of-the-art methods on three different benchmark datasets. Following Lai et al. [11], we used eight full reference metrics for quantitative analysis of our deshaking model. Detailed descriptions of these metrics can be found in [11]. For comparison we choose the state-of-the-art blind deblurring algorithms given by: Xu et al. [21], Whyte [20], Sun et al. [18], MBMF [3], and MS-CNN [16].

a) **Places Dataset [24]:** To perform the quantitative comparison of progressive models, we generate a synthetically blurred dataset in the same way as described earlier. We used the images from the Places dataset to generate pairs of deblurred and blurred images. The results are shown in Table 3. Note that Ours(1) in the Table 3 describes the dense generative net with only the ℓ_1 loss.

b) **GoPro Dataset** [16]: Images in this dataset were captured using GoPro and closely mimic the blur generated in real life. Out of total images, we used 438 images for our testing dataset and the rest of the images for creating the training dataset. We show the results of the quantitative comparison with the other state-of-the-art methods in Table 4. Our results show significant improvement in terms of image quality.

c) **Lai et al. Dataset** [11]: Lai et al. generated synthetic dataset by convolving nonuniform blur kernels and imposing several common degradations. To generate blur kernels they also recorded 6D camera trajectories. The comparative methods of our method with other algorithms are given in Table 5. The MS-CNN learning model [16], which also bypasses the kernel estimation step, was re-trained by us on the same dataset that we used for training our own model. On the other hand, we use the available testing codes of [18] and [3] for reporting the comparison.

d) **Köhler et al. Dataset** [10]: This benchmark dataset consists of four latent images. To construct a non-uniform blur dataset, twelve 6D camera trajectories were recorded over time assuming linear camera response function using which blurred images were captured. The captured scenes were planar and at a fixed distance from the camera. We report the quantitative results on this dataset in Table 6. From the table, we could infer that our model significantly outperforms the other methods. Qualitative comparisons of the different methods with ours could be seen in Fig. 3, Fig. 4 and Fig. 5. As evident from the figures, results produced by our method are visually superior compared to that of the-state-of-the-art.

5. Conclusion

We have designed a novel, end-to-end conditional GAN-based filter model which performs blind restoration of shaken images. Our results show that our model and framework outperforms the state-of-the-art for non-uniform deblurring. The fast execution time of our model makes it easily deployable in cameras and photo editing tools. We show that densely connected convolutional networks can be as effective for image generation as it is for classification.

Acknowledgement

Shubham Pachori and Shanmuganathan Raman were supported through an ISRO RESPOND grant.

References

[1] J. Bruna, P. Sprechmann, and Y. LeCun. Super-resolution with deep convolutional sufficient statistics. In *ICLR*, 2016.

[2] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR 2009. IEEE Conference on*, pages 248–255. IEEE, 2009.

[3] D. Gong, J. Yang, L. Liu, Y. Zhang, I. Reid, C. Shen, A. v. d. Hengel, and Q. Shi. From motion blur to motion flow: a deep learning solution for removing heterogeneous motion blur. In *IEEE CVPR*, 2017.

[4] S. Harmeling, H. Michael, and B. Schölkopf. Space-variant single-image blind deconvolution for removing camera shake. In *Advances in NIPS*, pages 829–837, 2010.

[5] K. He, X. Zhang, S. Ren, and J. Sun. Identity mappings in deep residual networks. In *ECCV*, pages 630–645. Springer, 2016.

[6] G. Huang, Z. Liu, K. Q. Weinberger, and L. van der Maaten. Densely connected convolutional networks. In *IEEE CVPR*, 2017.

[7] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. In *IEEE CVPR*, 2017.

[8] J. Johnson, A. Alahi, and L. Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *ECCV*, pages 694–711. Springer, 2016.

[9] D. Kingma and J. Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

[10] R. Köhler, M. Hirsch, B. Mohler, B. Schölkopf, and S. Harmeling. Recording and playback of camera shake: Benchmarking blind deconvolution with a real-world database. *Computer Vision–ECCV 2012*, pages 27–40, 2012.

[11] W.-S. Lai, J.-B. Huang, Z. Hu, N. Ahuja, and M.-H. Yang. A comparative study for single image blind deblurring. In *CVPR*, pages 1701–1709, 2016.

[12] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. In *IEEE CVPR*, 2017.

[13] C. Li and M. Wand. Precomputed real-time texture synthesis with markovian generative adversarial networks. In *ECCV*, pages 702–716. Springer, 2016.

[14] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *ECCV*, pages 740–755. Springer, 2014.

[15] A. L. Maas, A. Y. Hannun, and A. Y. Ng. Rectifier nonlinearities improve neural network acoustic models. In *ICML*, volume 30, 2013.

[16] S. Nah, T. H. Kim, and K. M. Lee. Deep multi-scale convolutional neural network for dynamic scene deblurring. In *IEEE CVPR*, 2017.

[17] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015.

[18] J. Sun, W. Cao, Z. Xu, and J. Ponce. Learning a convolutional neural network for non-uniform motion blur removal. In *IEEE CVPR*, pages 769–777, 2015.

[19] I. Vasiljevic, A. Chakrabarti, and G. Shakhnarovich. Examining the impact of blur on recognition by convolutional networks. *arXiv preprint arXiv:1611.05760*, 2016.

[20] O. Whyte, J. Sivic, and A. Zisserman. Deblurring shaken and partially saturated images. *IJCV*, 110(2):185–201, 2014.

[21] L. Xu, S. Zheng, and J. Jia. Unnatural l0 sparse representation for natural image deblurring. In *CVPR*, pages 1107–1114, 2013.

[22] F. Yu and V. Koltun. Multi-scale context aggregation by dilated convolutions. In *ICLR*, 2016.

[23] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *ECCV*, pages 818–833. Springer, 2014.

[24] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva. Learning deep features for scene recognition using places database. In *NIPS*, pages 487–495, 2014.