Blind Image Deconvolution using Pretrained Generative Priors

Muhammad Asim*
Fahad Shamshad*
Ali Ahmed
Department of Electrical Engineering,
Information Technology University,
Lahore, Pakistan
{msee16001,fahad.shamshad,ali.ahmed}@itu.edu.pk

1 Supplementary Material

1.1 Algorithm Parameters

The choice of free parameters for SVHN, Shoes and CelebA of *Deep Deblur* and *Deep Deblur with Slack* are given in Table 1 and 2.

Dataset	λ	γ	Steps(t)	Step Size	Random Restarts
SVHN	0.01	0.01	6,000	$0.01\exp^{-\frac{t}{1000}}$	10
CelebA	0.01	0.01	10,000	$0.01\exp^{-\frac{t}{1000}}$	10
Shoes	0.01	0.01	10,000	$0.01\exp^{-\frac{t}{1000}}$	10

Table 1: *Deep Deblur* Parameters.

Dataset	τ	ζ	ρ	Steps(t)	Step Size	Random Restarts
CelebA	100	0.5	10^{-3}	10,000	0.005 (adam)	10
Shoes	100	0.5	10^{-3}	10,000	0.005 (adam)	10

Table 2: *Deep Deblur with Slack* Parameters.

Both algorithms were implemented in Tensorflow and the code will be made publicly available.

2 Generative Models

The generative model of SVHN images is a trained VAE with the network architecture described in Table 3. The dimension of the latent space of VAE is fixed at 100, and training is carried out on SVHN with a batch size of 1500, and a learning rate of 10^{-5} using the Adam optimizer. After training, the decoder part is extracted as the desired generative model $G_{\mathcal{I}}$.

* The authors contributed equally.

Model Architectures		
Model	**Encoder**	**Decoder**
Blur VAE	conv(20, 2×2, 1) $\rightarrow$ relu $\rightarrow$ maxpool(2×2, 2) $\rightarrow$ conv(20, 2×2, 1) $\rightarrow$ relu $\rightarrow$ maxpool(2×2, 2) $\rightarrow$ fc(50), fc(50) $\rightarrow$ z_k	z_k $\rightarrow$ fc(720) $\rightarrow$ relu $\rightarrow$ reshape $\rightarrow$ upsample(2×2) $\rightarrow$ convT(20, 2×2, 1) $\rightarrow$ relu $\rightarrow$ upsample(2×2) $\rightarrow$ convT(20, 2×2, 1) $\rightarrow$ relu $\rightarrow$ convT(1, 2×2, 1) $\rightarrow$ relu
SVHN VAE	conv(128, 2×2, 2) $\rightarrow$ batch-norm $\rightarrow$ relu $\rightarrow$ conv(256, 2×2, 2) $\rightarrow$ batch-norm $\rightarrow$ relu $\rightarrow$ conv(512, 2×2, 2) $\rightarrow$ batch-norm $\rightarrow$ relu $\rightarrow$ fc(100), fc(100) $\rightarrow$ z_i	z_i $\rightarrow$ fc(8192) $\rightarrow$ reshape $\rightarrow$ convT(512, 2×2, 2) $\rightarrow$ batch-norm $\rightarrow$ relu $\rightarrow$ convT(256, 2×2, 2) $\rightarrow$ batch-norm $\rightarrow$ relu $\rightarrow$ convT(128, 2×2, 2) $\rightarrow$ batch-norm $\rightarrow$ relu $\rightarrow$ conv(3, 1×1, 1) $\rightarrow$ sigmoid

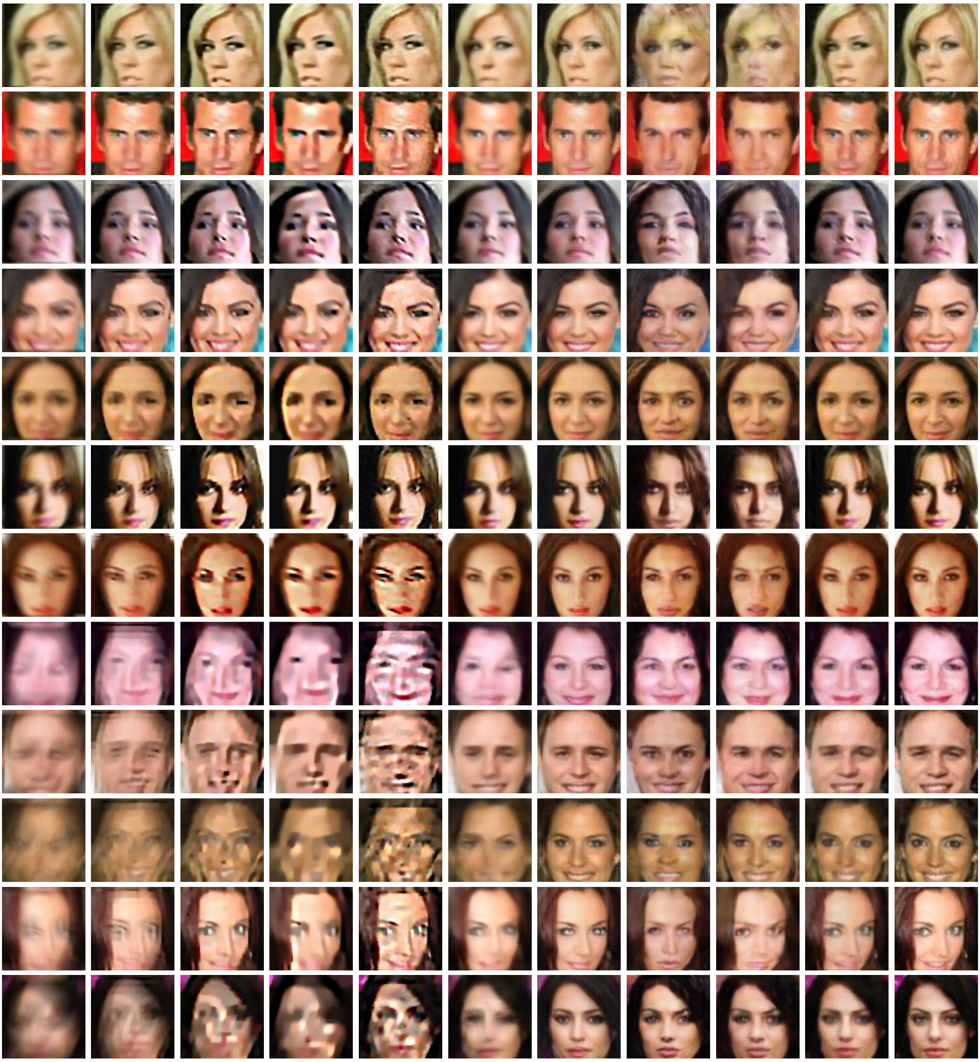
Table 3: Architectures for VAEs used for Blur and SVHN. Here, conv(m,n,s) represents convolutional layer with m filters of size n and stride s. Similarly, convT represents transposed convolution layer. Maxpool(n,m) represents a max pooling layer with stride m and pool size of n. Finally, fc(m) represents a fully connected layer of size m. The decoder is designed to be a mirror reflection of the encoder in each case.

For CelebA and Shoes dataset, the generative model $G_{\mathcal{I}}$ is the default deep convolutional generative adversarial network (DCGAN)[1].

The generative model of motion blur dataset is a trained VAE with the network architecture given in Table 3. This VAE is trained using Adam optimizer with latent dimension 50, batch size 5, and learning rate 10^{-5}. After training, the decoder part is extracted as the desired generative model $G_{\mathcal{K}}$.

References

[1] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *Advances in Neural Information Processing Systems*, pages 2234–2242, 2016.



(a) y (b) i_{DP} (c) i_{OH} (d) i_{DF} (e) i_{EP} (f) i_{CNN} (g) i_{DeGAN} (h) $\hat{i}_{\text{DD}}$ (i) i_{range} (j) $\hat{i}_{\text{DDS}}$ (k) i

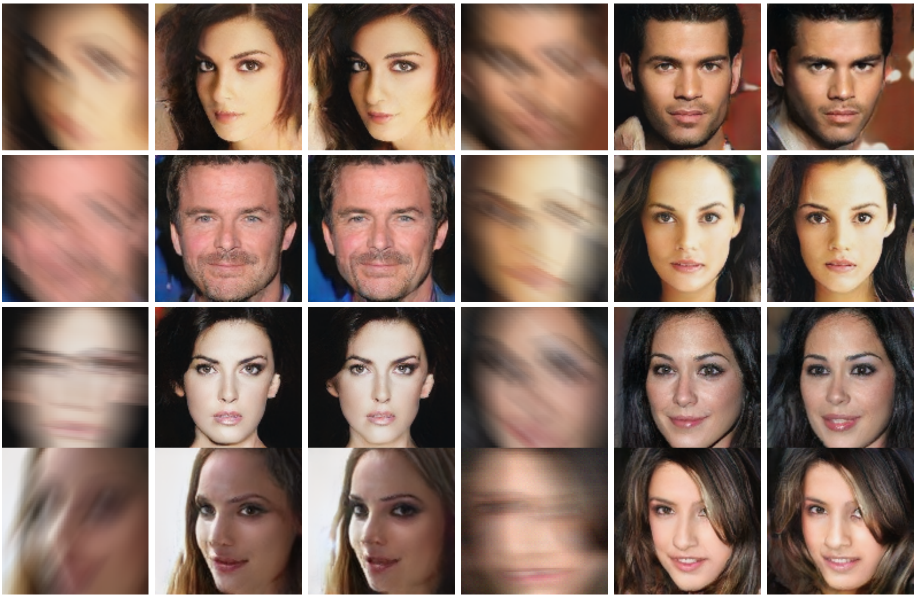
Figure 1: Comparison of image deblurring on CelebA for *Deep Deblur* and *Deep Deblur with Slack* with baseline methods. Deblurring results of *Deep Deblur with Slack*, $\hat{i}_{\text{DDS}}$, are superior than all other baseline methods, especially under large blurs. Deblurred images of DeblurGAN, i_{DeGAN}, although sharp, deviate from the original images, i, whereas *Deep Deblur with Slack* tends to agree better with the groundtruth.

(a) y (b) i_{DP} (c) i_{OH} (d) i_{DF} (e) i_{EP} (f) i_{CNN} (g) i_{DGAN} (h) $\hat{i}_1$ (i) i_{range} (j) i

Figure 2: Comparison of image deblurring on SVHN for *Deep Deblur* with baseline methods. Deblurring results of *Deep Deblur*, $\hat{i}_{\text{DD}}$, are superior than all other baseline methods, especially under large blurs. Better results of *Deep Deblur* on SVHN are explained by the close proximity between range images i_{range} and original groundtruth images i.

(a) y (b) i_{DP} (c) i_{OH} (d) i_{DF} (e) i_{EP} (f) i_{CNN} (g) i_{DGAN} (h) $\hat{i}_1$ (i) i_{range} (j) $\hat{i}_2$ (k) i

Figure 3: Comparison of image deblurring on Shoes for *Deep Deblur* and *Deep Deblur with Slack* with baseline methods.



(a) Blurry (b) Deblurred (c) New Sample (d) Blurry (e) Deblurred (f) New Sample

Figure 4: Image deblurring results using PGGAN as generator $G_{\mathcal{I}}$. New samples, generated by $G_{\mathcal{I}}$, were blurred, and *Deep Deblur* was used to deblur these blurry images.