*** Design and Implementation of Deep Learning Architecture for De-Noise of RGB Images ***

*

NATIONAL INSTITUTE OF TECHNOLOGY KARNATAKA,SURATHKAL
MANGLORE, KARNATAKA

Submitted By
(171EC213) Deepak Jyani
(171EC233) Parmeshwar Barupal

E- mail Address : -
Deepak - deepkjyani@gmail.com
Parmeshwar - rjparam7@gmail.com

Submitted To
Dr.Shyam Lal

I.

***Abstract*—This document is a model and instructions for LaTeX. This and the IEEEtran.cls file define the components of your paper [title, text, heads, etc.]. *CRITICAL: Do Not Use Symbols, Special Characters, Footnotes, or Math in Paper Title or Abstract.**

***Index Terms*—component, formatting, style, styling, insert**

II. Introduction

1. An algorithm is any in image processing, applications and analysis, denoising is one of the most significant techniques currently used. Removing random noise and reserving the details of an image is fundamental goal of image denoising approaches.

Identify applicable funding agency here. If none, delete this.

2. This approach reduces the visibility of lower contrast objects in addition the noisy image construct unwanted visual quality.

3. Image denoising autoencoder is classic method used for deep learning for RGB color images .

4. Autoencoder implemented with a neural network in almost all contexts is used for compression and decompression.

5. Image denoising autoencoder is classical issue in the field of digital image processing where compression and decompression function are lossy and data specific.

6.Consecutively to improve and recuperate superior facts that are concealed in the data. Noise removal is essential during digital imaging applications.

7. As compared to the original image inputs, the decompressed outputs will be degraded because of autoencoders are lossy similar to JPEG and MP3 compression. This differs from lossless arithmetic compression.

8. Noise removal is essential during digital imaging applications. In Gaussian probability distribution noise in digital metaphors is initiate to be preservative in nature with consistent supremacy in the entire bandwidth.

III. Motivation

An algorithm is any in image processing, applications and analysis, denoising is one of the most significant techniques currently used. Removing random noise and reserving the details of an image is fundamental goal of image denoising approaches. This approach reduces the visibility of lower contrast objects in addition the noisy image construct unwanted visual quality. Consecutively to improve and recuperate superior facts that are concealed in the data. Noise removal is essential during digital imaging applications. In Gaussian probability distribution noise in digital metaphors is initiate to be preservative in nature with consistent supremacy in the entire bandwidth.

1. It Low-light photography is always challenging because of noise. .

2. Cameras will take photos with high ISO sensitivity to compensate low brightness, but at the same time amplify noise signal. .

3. Digital images often contains "NOISE" which takes away their clarity and sharpness.

IV. Problem statement

******* Design and Implementation of Deep Learning Architectures For De-Noise Of RGB Images. ******* .

V. Benchmark Methods

Benchmarking is a way of discovering what is the best performance being achieved whether in a particular company, by a competitor or by an entirely different industry. This information can then be used to identify gaps in an organization's processes in order to achieve a competitive advantage.

The proposed DND benchmark for denoising algorithms consists of 50 scenes selected from our captured images. We chose images that look like typical photographs,but also included images with interesting structures that we believe to be challenging for the algorithms tested. A subset of the test images is lists the number of scenes per camera included in the benchmark dataset.

Benchmarking is a way of discovering what is the best performance being achieved – whether in a particular company, by a competitor or by an entirely different industry. This information can then be used to identify gaps in an organization's processes in order to achieve a competitive advantage.

Since many of the benchmarked algorithms are too slow to be applied to mega pixel-sized images,we crop 20 bounding boxes of 512 x 512 pixels from each image in the dataset, yielding 1000 test crops in total. They overlap at most 10 percentage and do not contain pixels that were annotated as changing between the two exposures. We provide the algorithms with an estimate of the global noise standard deviation ¯ by computing the standard deviation of the residual noise image R(yp) on each crop.As the different color channels usually look quite distinct, we denoise each channel separately.

A. Measurements for denoising algorithms

The target of denoising algorithm is to recover the latent clean image from its noisy observation. To evaluate denoising algorithms, different measurements have been adopted to compare the denoised estimation and ground truth high quality images. The most commonly used measurement is the peak signal to noise ratio (PSNR) index. PSNR is most easily defined via the mean squared error (MSE). Given the ground truth image G and denoised estimation E, the MSE is defined as :

$$MSE = Sum of M, N[E(m,n)G(m,n)]^2/(M*N) \quad (1)$$

where E(m, n) and G(m, n) are the pixel values at position (m, n) of image E and G, respectively, and M and N are the image size. Based on MSE, the definition of PSNR is:

$$PSNR = 10log10(R^2/MSE) \quad (2)$$

where R is the maximum fluctuation in the image data type. Although plenty of works have pointed out that PSNR is not a good fit to measure the perceptual similarity between two images, it is still the most commonly used index to compare two images.

Besides the MSR and PSNR, perceptual quality measurements have also been proposed to evaluate denoising algorithms. One of the representative measurement is the structural similarity (SSIM) index [83]. The SSIM index is calculated on various windows of an image. The measure between two windows x and y is:

$$SSIM(x,y) = (2x*y+c1)(2xy+c2)/(x^2+y^2+c1) \quad (3)$$

µx and µy are the average value of window x and y, respectively. x and y are the variance of x and y, and xy is the covariance of x and y*c1 = $(k1 * R)^2$ $and\ c2 = (k2 * R)^2$

SSIM and its extensions have been widely applied in different tasks to compare the estimated and ground truth images. For the image denoising task, also the feature structural similarity (FSIM) has been adopted in some works. Very recently, Zhang et al. proposed the Learned Perceptual Image Patch Similarity metric based on deep features. However, all the above mentioned perceptual quality measures are only proxies to the mean opinion score (MOS) as obtained based on the ratings from human subjects. Recently challenges on perceptual image superresolution , image enhancement and learned image compression resort to MOS for rankings.

VI. Proposed Architecture / Methods

The proposed framework mainly contains a chain of convolutional layers and symmetric deconvolutional layers. The framework is fully convolutional (and deconvolutional. Deconvolution is essentially unsampling convolution). Rectification layers are added after each convolution and deconvolution. For low-level image restoration problems, we useneitherpoolingnorunpoolinginthenetworkasusually pooling discards useful image details that are essential for these tasks. It is worth mentioning that since the convolutional and deconvolutional layers are symmetric, the network is essentially pixel-wise prediction, thus the size of input image can be arbitrary. The input and output of the network are images of the same size w*h*c, where w, h and c are width, height and number of channels.

A. Architure

Our main idea is that the convolutional layers act as a featureextractor,which preserve the primary components of objects in the image and meanwhile eliminating the corruptions. After forwarding through the convolutional layers, the corrupted input image is converted into a “clean” one. The subtle details of the image contents may be lost during this process. The deconvolutional layers are then combined to recover the details of image contents. The output of the deconvolutional layers is the recovered clean version of the input image. Moreover, we add skip connections from a convolutional layer to its corresponding mirrored deconvolutional layer. The passed convolutional feature maps are summed to the deconvolutional feature maps element-wise, and passed to the next layer after rectification. Deriving from the above architecture, we have used two networksvin our experiments, which are of 20 layers and 30 layers respectively, for image denoising, image super-resolution, JPEG deblocking and image inpainting.

B. Convolution Neural Network

We use Residual Block which was presented in the course as a different approach in our architecture compared to DnCNN and other methods available online as far as we know. Fig 7 is the residual block that we use. It is commonly believed that residual learning helps avoid vanishing gradient and thus deeper neural network can converge faster. Another important technique which also aids training process is ReLU activation function and Batch Normalization. We utilize both of these trendy techniques in our proposed network.

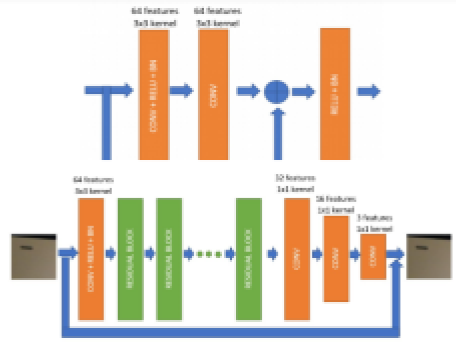


Figure 7: Our proposed network architecture

With five residual blocks, our network consists of 14 convolution layers. Our network take in RGB image as input, and produce RGB image as output. We also use skipping connection from

C. Deconvolution Decoder

Architectures combining layers of convolution and deconvolution have been proposed for semantic segmentation recently. In contrast to convolutional layers, in which multiple input activations within a filter window are fused to output a single activation, deconvolutional layers associate a single input activation with multiple outputs. Deconvolution is usually used as learnable up-sampling layers.

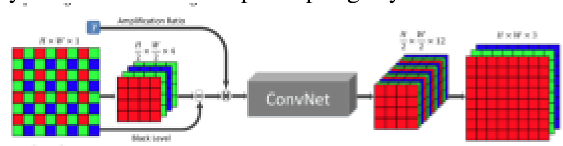


Figure 2: The pipeline proposed in the paper 'Learning to see in the dark'

In our network, the convolutional layers successively down-sample the input image content into a small size abstraction. Deconvolutional layers then up-sample the abstraction back into its original resolution.

fully convolutional and deconvolutional, i.e., without pooling and un-pooling. The reason is that for low-level image restoration, the aim is to eliminate low level corruption while preserving image details instead of learning image abstractions. Different from high-level applications such as segmentation or recognition, pooling typically eliminates the abundant image details and can deteriorate restoration performance.

VII. Training And Implementation Details

Number of technique proposes for compression and decompression in which autoencoding is popular scheme that uses three functions.

- Data specific
- Lossy Function
- Training Setup.

A. Data Specific

In this section, we describe in detail the cohort selection, data extraction, data cleaning and feature extraction methods we employed to preprocess our dataset.

The data extracted from database has lots of erroneous entries due to noise, missing values, outliers, duplicate or incorrect records, clerical mistakes, etc. We identified and handled the following three issues with the extracted data.

In data-specific function, it only capable to constrict data alike to what they have been educated on. Like if we use MPEG-2 Audio Layer III (MP3) compression algorithm, which merely be compressed sound in broad-spectrum, although not concerning the exact type of sounds. Similarly, if we trained our model on images of faces autoencoder generate poor squeezing images of plants, for the reason that the facial appearance it would learn about the face dataset.

B. Lossy Function

Loss function which compares with reconstructed (decompressed) and compressed images. Stochastic Gradient Descent function can optimize in order to minimize reconstructed loss. Parametric functions of neural network will be chosen for encoder and decoder and to be distinguishable with regard to distance function. We use more filters per layer as compared to the previous convolutional autoencoder which is slightly different model for reconstructed images to improve quality.

Gradient descent is a simple optimization procedure that you can use with many machine learning algorithms. ... Stochastic gradient descent refers to calculating the derivative from each training data instance and calculating the update immediately.

However, the gradient descent algorithm may be infeasible when the training data size is huge. Thus, a stochastic version of the algorithm is often used instead.

To motivate the use of stochastic optimization algorithms, note that when training deep learning models, we often consider the objective function as a sum of a finite number of functions:

$$F(x) = 1/n * sumi = 1ton(fi(x)) \quad (4)$$

C. Training Setup

In general, there are three types of layers in our network: convolution, deconvolution and element-wise sum. Each layer is followed by a Rectified Linear Unit (ReLU) . Let X be the input, the convolutional and deconvolutional layers are expressed as:

$$F(x) = Max(0, Wk * X + Bk) \quad (5)$$

where Wk and Bk represent the filters and biases, and denotes either convolution or deconvolution operation for theconvenienceofformulation.Forelement-wisesumlayer, the output is the element-wise sum of two inputs of the same size, followed by the ReLU activation:

$$F(x) = Max(0, X1 + X2) \quad (6)$$

Learning the end-to-end mapping from corrupted images to clean images needs to estimate the weights represented by the convolutional and deconvolutional kernels. Specifically, given a collection of N training sample pairs Xi,Y i, where Xi is a noisy image and Y i is the clean version as the groundtruth. We minimize the following Mean Squared Error (MSE):

L(a) = 1/N * Sum of i= 1 to N (f(Xi; o)-Yi)^2

The aim of restoration is to eliminate corruption while preserving the image details as mush as possible. Previous works typically use shallow networks for low-level image restoration tasks. The reason may be that deeper networks can destroy the image details, which is undesired for pixel-wise dense regression. Even worse, using very deep networks may easily suffer from training issues such as gradient vanishing.Using skip connections in a very deep network can address both of the above two problems. Firstly, we design experiments to show that using skip connections is beneficial for image detail presering. Specifically, two networks are trained for image denoising with a noise level of = 70.

VIII. Simulation Result And Discussion

After applying Gaussian noise with noise factor value 0.2 on dataset our results show images are blur as in fig. Images in both training as well as test datasets have been clipped between 0 and 1.

Simulation Model Summary

Layer (type)	Output Shape	Param
Encoder		
conv2d (8) (Conv2D)	(None, 26, 26, 32)	320
Max Pooling2d (4) (Maxpooling2)	(None, 13, 13, 32)	0
conv2d (9) (Conv2D)	(None, 13, 13, 8)	2312
Max Pooling2d (5) (Maxpooling2)	(None, 7, 7, 8)	0
Decoder		
conv2d (10) (Conv2D)	(None, 7, 7, 8),	584
Up Sampling2d (3) (Upsampling2)	(None, 14. 14, 8)	0
conv2d (11) (Conv2D)	(None, 14, 14, 8)	584
Up Sampling2d (4) (Upsampling2)	(None, 28, 28, 8)	0
conv2d (12) (Conv2D)	(None, 28, 28, 32)	2336
Up Sampling2d (5) (Upsampling2)	(None, 56, 56, 32)	0
conv2d (13) (Conv2D)	(None, 56, 56, 1)	289
Total Params = 6,425		
Trainable Params = 6,425		
Non-trainable Params = 0		

Below Table shows the PSNR values, averaged over all crops and color channels (SSIM values are available in the supplemental material). We make several interesting observations.

The general tendency also holds across noise levels. This is quite surprising as the by now classic BM3D approach was previously considered to have been outperformed by the other approaches; our realistic noise dataset shows that this is not the case. The discriminative methods fall short, which suggests that they generalize poorly to noise distributions that were not used during training. The generative FoE model performs surprisingly competitive in linear raw space, but is the only baseline that performsworseafterVST.ThissuggeststhatFoEbenefitsfrom the more realistic likelihood in linear raw space .

Compare Between PSNR And SSIM in Noisy and De-noisy images

Image No.	Noisy PSNR	Clean PSNR	Noisy SSIM	Denoisy SSIM
I1	18.65db	25.4db	0.308	0.499
I2	26.85db	31.60db	0.75	0.86
I3	28.48db	30.12db	0.801	0.91
I4	31.46db	35.69db	0.92	0.96
I5	23.15db	30.95db	0.51	0.85
I6	23.97db	37.23db	0.79	0.93

A. Compare Between Noisy And De-noisy Images

Evaluation of image restoration tasks including image denoising, image super-resolution, JPEG image deblocking, non-blind image debluring and image inpainting are conducted and compared against a few existing stateof-the-art methods in each topic. Peak Signal-to-Noise Ratio (PSNR) and Structural SIMilarity (SSIM) index are calculated for evaluation.

Noisy Image

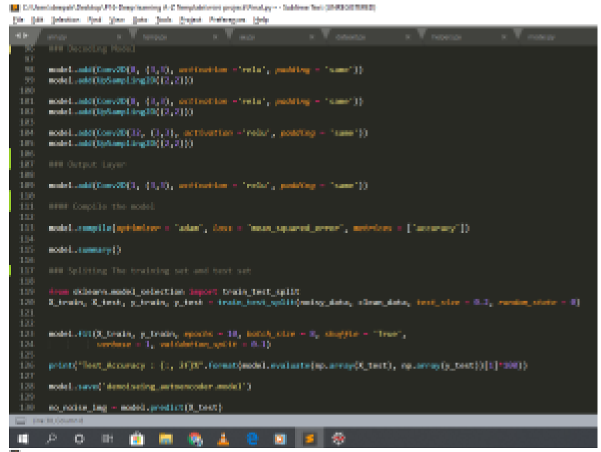
PSNR = 23.54db SSIM = 0.56

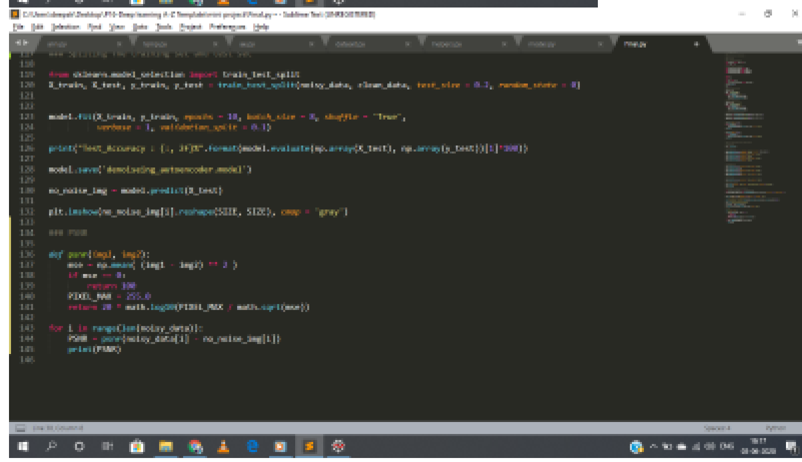
De-Noisy Image

De-Noisy Image

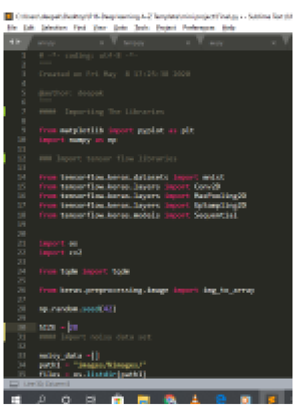
PSNR = 35.85db   SSIM = 0.78

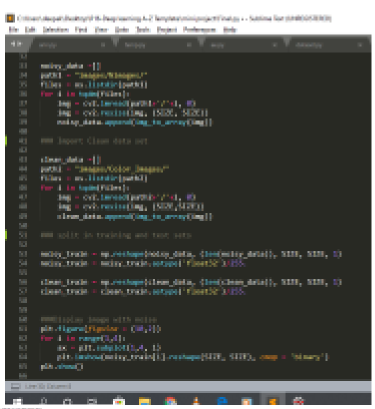
we wanted to understand qualitatively the effect of the corruption+denoising training. To this end we display the filters obtained after initial training of the first denoising autoencoder on traing set images. we can see that PSNR and SSIM values are increasing in De-noisy image in compare to Noisy image. i.e. that We get better images then previous images.

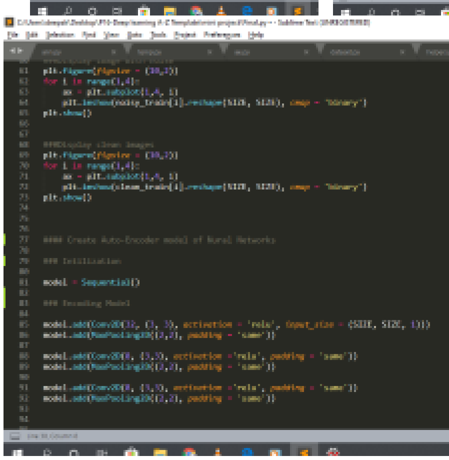




IX. Code







X. Abstract

Usually, cameras will take photos with high ISO sensitivity to compensate low brightness, but at the same time amplify noise signal. Photographers can also keep ISO low by increasing exposure time or aperture, however, result photos can be blur due to motion or depth of field. Therefore, many post-processing methods have been developed to restore original quality of noised imaging, and this has been an active area of research for years. In recent years, researchers in convolution neural network and deep learning have proposed many successful deep learning methods to denoise photos. Our project aims to explore denoising problem and propose a pipeline based on materials presented in Image Processing (EC386) project course at Denoise of RGB Images. Using presented dataset, our network learns and processes directly on RGB images with real life noise and produce restored images. We evaluate and compare our method to BM3D[4] method.

Image denoising autoencoder is classical issue in the field of digital image processing where compression and decompression function are lossy and data specific. In this paper, we use autoencoder technique on RGB (Red, Green and Blue) color scheme dataset, we added Gaussian noise on dataset then encode by using 2D convolutional neutral network. Similarly, we decode noisy dataset to train our model. After training, the proposed method can learn denoising data and returns effective results.

Image De-Noise techniques are traditionally evaluated on Image corrupted by synthesized i.i.d. Gaussian Noise. We aim to obviate this unrealistic setting by developing a methodology for Benchmarking De-Noising Techniques on real photograps. We capture pairs of images with different ISO values and appropriately adjusted exposure times, where the nearly noise - free derive the ground truth, careful post-processing is needed. We correct spatial misaligment, cope with inaccuracies in the exposure parameters throughs a linear intensity transform based on novel heteroscedastic tobit regression model, and remove residual low - frequency bias that stems, e.g., from minor illumination changes. We then capture a novel benchmark dataset, the Darmstadt Noise Dataset (DND), with consumer camera as of differing sensor sizes. One intersting finding is that various recent techniques that perform well on synthetic noise are clearly outperform by BM3D on photograps with real noise.

XI. Conclusion

To benchmark denoising algorithms on real photographs, we introduced an acquisition procedure based on pairs of images of the same scene, captured with different analog gains and exposure time. While in theory the per-pixel mean intensity should stay constant,in practice ween countered residual errors. To derive ground-truth data, we proposed and evaluated a procedure for handling residual errors stemming from inaccurate gain and exposure time changes, relying on a novel heteroscedastic Tobit regression model. Weal so correct for lighting changes in a transformed space, as well as spatial mis-alignments. Our experiments showed the efficacy of this post-processing on simulated data, as well as its necessity on real photographs. We will make our novel ground-truth dataset of real photographs publicly available as a benchmark. We used it for evaluating various denoising algorithm sand observed that BM3D continues to outperform recent denoising methods on real photographs, which is in contrast to findings on previously considered synthetic settings. More generally, our analysis revealed that the common scientific practice for evaluating denoising techniques has rather limited relevance for realistic settings.

we have proposed a deep encoding and decoding framework for image restoration. Convolution and deconvolution are combined,modeling the restoration problem by extracting primary image content and recovering details.
More importantly,we propose to use skip connections, which helps on recovering clean images and tackles the optimization difficulty caused by gradient vanishing, and thus obtains performance gains when the network goes deeper.Experimental results and our analysis show that our network achieves better performance.

XII. Referenece

[1.] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in Proc. IEEE Conf. Comp. Vis. Patt. Recogn., 2016.

[2.] J. Mairal, F. R. Bach, J. Ponce, G. Sapiro, and A. Zisserman, "Nonlocal sparse models for image restoration," in Proc. IEEE Int. Conf. Comp. Vis., 2009, pp. 2272–2279.

[3.] W. Dong, L. Zhang, G. Shi, and X. Li, "Nonlocally centralized sparse representation for image restoration," IEEE Trans. Image Process., vol. 22, no. 4, pp. 1620–1630, 2013.

[4.] U. Schmidt, J. Jancsary, S. Nowozin, S. Roth, and C. Rother, "Cascades of regression tree fields for image restoration," IEEE Trans. Pattern Anal. Mach. Intell., vol. 38, no. 4, pp. 677–689, 2016.

[5.] U. Schmidt and S. Roth, "Shrinkage fields for effective image restoration," in Proc. IEEE Conf. Comp. Vis. Patt. Recogn., 2014, pp. 2774–2781.

[6.] D. Zoran and Y. Weiss, "From learning models of natural image patches to whole image restoration," in Proc. IEEE Int. Conf. Comp. Vis., 2011, pp. 479–486.

[7.] H. Liu, R. Xiong, J. Zhang, and W. Gao, "Image denoising via adaptive soft-thresholding based on non-local samples," in Proc. IEEE Conf. Comp. Vis. Patt. Recogn., 2015, pp. 484–492.

[8.]Bengio, Y., Le Cun, Y. (2007). Scaling learning algorithms towards AI. In L. Bottou, O. Chapelle, D. DeCoste and J. Weston (Eds.), Large scale kernel machines. MIT Press.

[9.] Hinton, G. E., Osindero, S., Teh, Y. (2006). A fast learning algorithm for deep belief nets. Neural Computation, 18, 1527–1554.

[10.] J. Zhang, K. Hirakawa, and X. Jin. Quantile analysis of image sensor noise distribution. In ICASSP, pages 1598–1602, 2015.

[11.] F. Zhu, G. Chen, and P.-A. Heng. From noise modeling to blind image denoising. In CVPR, pages 420–429, 2016.

[12.] X. Jiang, Y. Zhang, W. Zhang, and X. Xiao. "A novel sparse auto-encoder for deep unsupervised learning." In Advanced Computational Intelligence (ICACI), 2013 Sixth International Conference on, pp. 256-261. IEEE, 2013.

[13.] X. Mao, C. Shen, and Y.-B. Yang. Image restoration using very deep convolutional encoderdecoder networks with symmetric skip connections. In Advances in neural information processing systems, pages 2016.

[14.] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics.In Proc. 8th Int'l Conf. Computer Vision, volume 2, pages 416–423, July 2001.

[15.] P. Milanfar. A tour of modern image filtering: New insights and methods, both practical and theoretical. IEEE Signal Processing Magazine, 30(1):106–128, 2013.