Image Denoising with Deep Convolutional Neural Networks

Aojia Zhao
Stanford University
aojia93@stanford.edu

Abstract

Image denoising is a well studied problem in computer vision, serving as test tasks for a variety of image modelling problems. In this project, an extension to traditional deep CNNs, symmetric gated connections, are added to aid faster convergence transfer of high level information normally lost during downsampling. Results show that in under 50,000 training images, gated connections begin to make noticeable improvements in feature learning and image denoising. An additional classification task shows marginal feature learning effects when denoising weights are used as pre-training.

1. Introduction

Image denoising has always been a central problem in computer vision. At its core, denoising is an inherently ill-posed problem due to the loss of information during noise addition.

$$I' = D(I) + h$$

Here, $D(I)$ is the degrading function with respect to original image I while h serves as additive noise. As degradation functions are not always guaranteed to be affine transformations, traditional techniques cannot fully recover noised out pixels of the clean image.

Recently, applications of CNNs in solving this problem has produced increasingly promising results. Intuitively, this comes from the change in mindset of recovering information from the remnants to learning key features describing the noisy image and predicting the original from those traits.

2. Background Literature

Prior to utilization of Deep Neural Nets, one of the prominent state-of-the-art metrics was the BM3D algorithm.[Dabov et al.] In it, the authors grouped similar 2D fragments and used inverse 3D transformations to achieve fine detail denoising. An alternative approach that also showed good performance was Iterative Regularization [Osher et al.], which attempted to reduce noise patterns through minimizing a standard metric like Bregman Distance.

With the rise of deep learning, one of the earlier works on applying DNN to an autoencoder for feature denoising, [Bengio et al.] showed that stacking multilayered neural networks can result in very robust feature extraction under heavy noise. A later paper on semantic segmentation, [Long et al.] shows the power of Fully Connected CNNs in parsing out feature descriptors for individual entities in images.

Recently, a proposed deep-CNN architecture by [Mao et al.] features a 30-layer convolutional-deconvolutional model designed for deep learning of image features. Their innovation is the inclusion of Symmetric Skip Connections (SSC) between alternating Conv-Deconv layers. The modification attempts to solve two problems with training deep CNNs. First, with increasing number of layers comes the vanishing gradient problem that prevents effective backpropagation towards front layers. This is due to the structure of gradient product at each layer, where error is sequentially diminished in magnitude. In theory, alternating connections allow gradients to backpropagate directly from an upsample, deconvolutional layer to the corresponding downsample, convolutional layer. Second, as details are inevitably lost during the downsampling layers, SSC can also serve as intermediate information flow gates akin to LSTM forget gates. To prevent massive information leak through these channels, gate coefficients can be modified during training to force learning at bottleneck layers.

3. Model Architecture

3.1. Conv-Deconv Stacked Structure

Drawing upon previously proven stacked autoencoder-decoder networks, this project implements a 10-layer CNN consisting of 5-Conv layers followed by 5-Deconv layers. As SSC resulted in faster convergence in the 30 and 20-layer structures presented in [Mao et al.], this project implements the inspired extension of Direct Symmetric Connection (DSC). DSC uses the same setup as SSC, connecting corresponding Conv to Deconv layers. However, in order

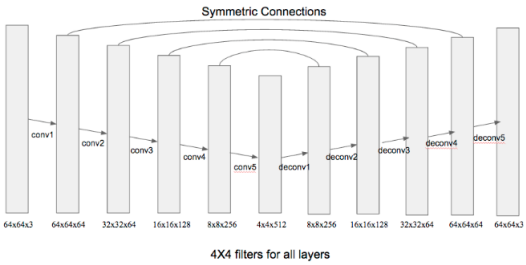

Figure 1. 10-layer model with DSC

Layer	Dimensions	Layer	Dimensions
Conv-1	64x64x64	Deconv-1	8x8x256
Conv-2	32x32x64	Deconv-2	16x16x128
Conv-3	16x16x128	Deconv-3	32x32x64
Conv-4	8x8x256	Deconv-4	64x64x64
Conv-5	4x4x518	Deconv-5	64x64x3

Table 1. Table of Layers

to further reduce number of weights required in a 10-layer model and speed up learning, every single Conv layer is connected to its corresponding Deconv layer, resulting in 4 direct connections.

The main idea during the downsampling layers is for the network to extract feature descriptors from training data. In the optimal setting, these weights should dictate a general representation that can group different types of image objects and rely those facts to the generative upsampling layers. Then, the deconv layers can build upon the cleaned, though bare-boned, feature-dense tensor from the bottleneck layer (Conv-5) and generative relevant details to complete a reproduction of the original.

To explain the DSC effects in detail, a typical downsampling layer will conduct the following

$$X_i = Conv(\delta * X_{i-1}, W_i) + b_i$$
$$X_i = Max(0, X_i)$$

Here, δ is the Gating Factor, controlling the amount of information flow to subsequent layers or to corresponding deconv layer. Similarly, the deconv layer will have

$$X'_i = Deconv(X'_{i-1} + (1-\delta) * X_i, W'_i) + b'_i$$
$$X'_i = Max(0, X'_i)$$

Where X' is the deconv layer inputs corresponding in order to conv layer information flow. In all layers, ReLU is applied to eliminate negative values. This is due to the RGB value properties of the input being ranging from 0 to 255, as well as reducing the effects of low gradients during backpropagation.

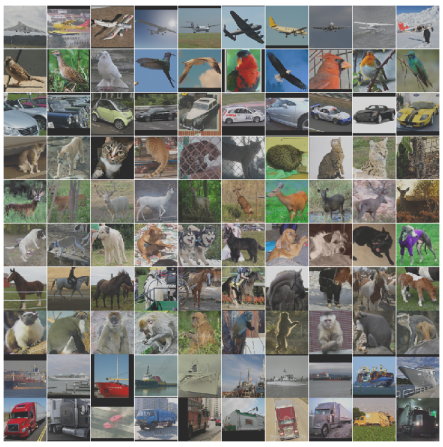

Figure 2. Some example STL-10 images [Coates et al.]

3.2. Loss Function

With the end goal of denoising an image and returning the same dimensional prediction, the most widely used loss minimizer is pixel-wise Mean Squared Error.

$$L = \frac{1}{N}\sum_{i=1}^{N} |X'_5 - Y|^2$$
$$X_0 = N(Y)$$

For each image in the training set, we apply a combination of Gaussian Noise and Salt & Pepper Noise. The resulting "noisy" image, X_0, is inputted to Conv-1 for training. The final denoised product, X'_5, is compared pixel-wise against the original ground truth, Y. With a proven track record for effective training, this was the classic loss function used to compute subsequent results.

An alternative approach of applying Perceptual Loss defined by [Li et al.] of using pre-trained weights to compare similarity of denoised images was applied with unsuccessful results. Methodology for adapting this approach is described below.

4. Data and Training

4.1. STL-10 Dataset

Though denoising training does not require specific labelled data due to its input to modified-input minimization structure, typical of an unsupervised learning problem, subsequent representation learning investigations required labelled data and thus restricted dataset choices. Additionally, due to the high quality considerations required of input images for fine grained detail learning, input sizes were restricted to be at least 64x64x3. Ultimately, the STL-10 dataset was chosen for its relatively large number of unlabelled, high quality images, 100k in total, as well as it's

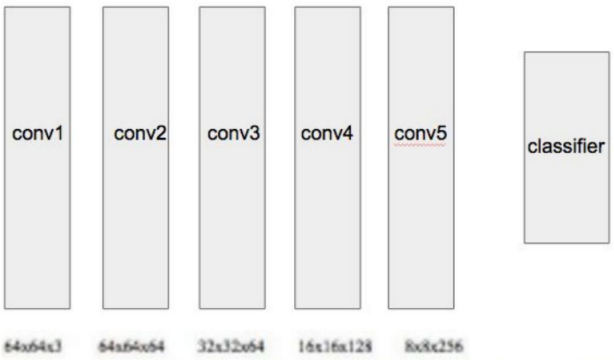

Figure 3. Perceptual Loss Classification Model

Layer	Dimensions
Conv-1	64x64x64
Conv-2	32x32x64
Conv-3	16x16x128
Conv-4	8x8x256
Conv-5	4x4x518
Fully-Connected	1024
Softmax Output	10

Table 2. Perceptual Loss Layers

labelled image section containing 13k images across 10 animal classes.

4.2. Training

Training was done over 3 separate series of related models: 10-layer with Perceptual Loss, 10-layer with MSE Pixel Loss, and 10-layer DSC with MSE Pixel Loss. In all three cases, Stochastic Gradient Descent with minibatching in Tensorflow was used as the minimizer.

In the first case, Perceptual Loss weights were learned using 13k labelled images through a 5-layer CNN followed by a fully-connected layer with drop-out, and then a Softmax readout layer over the 10 classes of animals. Minimization was through Cross Entropy with true labels as one-hot vector. Issues with over-fitting were not considered due to the non-convergence of the network even after going through all training examples. To calculate similarity between images, both denoised and ground truths are inputted and stopped after the fully-connected layer. The 1024-dimensional feature descriptors are then used to compute L2 distance as the minimization metric.

$$L = \sum_{i=1}^{1024} FC(X')_i - Y_i$$

For both the subsequent 10-layer training process, all 100k training data were used, passing in minibatches of 10 images per iteration, resulting in 10k iterations for both. In the DSC enabled model, Gate Factors for Conv-1 through Conv-4 are set to 0.1, 0.2, 0.3, 0.4 respectively. The idea is to allow minute amounts of information to travel between original noisy image and close to finished, denoised image, while at the same time allow larger influence to flow between center bottleneck layers so the middle layer doesn't have to necessarily learn all the distinct features for the network to converge.

4.3. Representation Learning

To judge whether the network has learned general representation from image denoising, one idea is to test denoising effects on images with only certain patches blurred out. The expected result from a convergent system is being able to distinguish segmentation of entities and generate denoised pixels fit for those boundaries.

Another investigation conducted after model training was applying learned weights of the Conv layers as pre-training initialization to the animal classification task. The idea is a learnt feature descriptor should contain distinguishing information that cluster different animals on some high dimensional level in the fully-connected layer. If such a representation exists, then training using these initialization on the 13k labelled data should converge much faster than truncated normal initialization used previously.

5. Results

In training the Perceptual Loss classification network, 13k labelled data images proved insufficient for classifier to converge under truncated normal initialization.

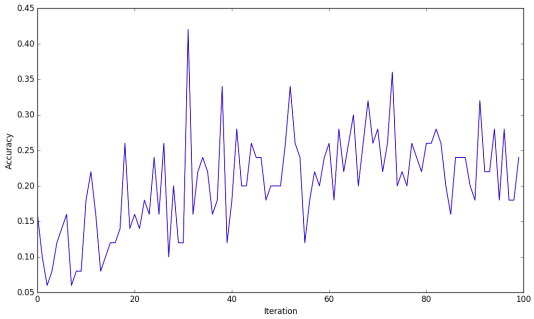

Figure 4. Accuracy Graph of Truncated Normal Initialization Classification

Accuracy at the end of 100 iterations averaged around 25 %, better than 10% expected of a random baseline but much worse than state-of-the-art algorithms. Though disappointing, weights were acquired and applied to measuring similarities between denoised and true images to train the main network.

Most likely due to the non-convergent structure of Perceptual Loss metric, main model weights do not appear to learn or converge. Behavior of the graph over 10000 iterations seems to fluctuate heavily, attempting to fit the

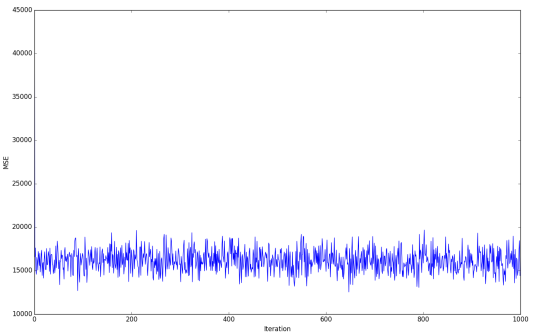

Figure 5. MSE for training with Perceptual Loss

L2 minimization. As weights on evaluation of testing images ultimately led to black squares on most denoised outputs, subsequent qualitative image results will be products of MSE pixel-wise training.

In terms of final convergence, both simple 10-layer and DSC 10-layer ended up not arriving at a stable loss plateau. In fact, in the case of non-DSC model, MSE loss did not noticeably drop at all, hinting at lack of training data for the massive dimensions of parameter weights, as well as depth of network, to properly propagate error to all layer elements. However, in the DSC enabled structure, some noticeable amounts of MSE minimization can be seen after roughly 5000 iterations, midway through training. Graph of MSE is presented below, with the first 1000 iterations omitted due to large range fluctuations several magnitudes outside pictured bound.

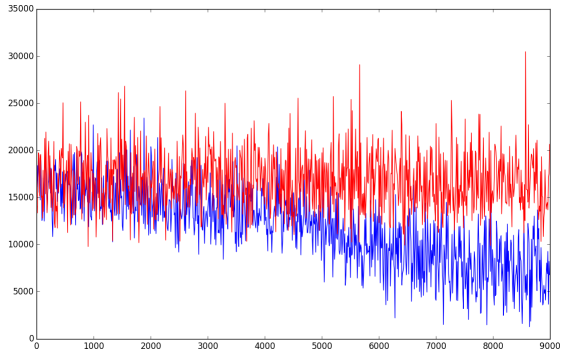

Figure 6. Graph of MSE vs Iteration. Blue is DSC enabled while red is simple 10-layer

This result does show the promise of gated connections between downsample and upsample layers, particularly in equal training data quantities vs traditional, one direction structures. At 10000 iterations, MSE of DSC structure averages around 7500 while the non DSC network still hovers around 15000. With these improvements said, features are

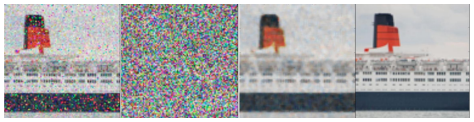

Figure 7. Foggy Ship at Sea. From left to right: Noisy, Conv-1 Visualization, Denoised, Clean

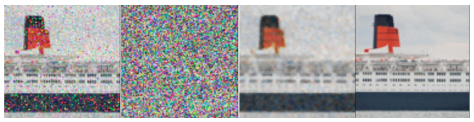

Figure 8. Foggy Ship at Sea. From left to right: Noisy, Conv-1 Visualization, Denoised, Clean

clearly still not fully learned by the layer weights as seen in the ship picture. Conv-1 output in particular seems to not hold much more than an outline of the red stem along with a light border for the edge at the bottom. It can be deduced that most likely the information passage here is still through the first DSC connecting to Deconv-5, while main information flow likely got zeroed out somewhere in the bottleneck layer.

Next, to look at how well representation has been learned to distinguish boxed blurs, the following image of bird is partially noised out. Surprisingly, the output of Conv-1 shows a very good outline of the edges of the bird, though the patch of noise is also included in the body. One explanation for this sharp contrast as opposed to the ship previously may be the blurry background in the bird picture along with the bright color contrast of foreground and background. In the ship image, both the sky and ship body is white, making segmentation difficult for the system. On the other hand, the bird has a bright yellow head with a brown body while the background is murky gray. These factors, combined with the fortunate fact the bound patch noise still seems to blend with the bird's main body, allows Conv-1 to extract key object features.

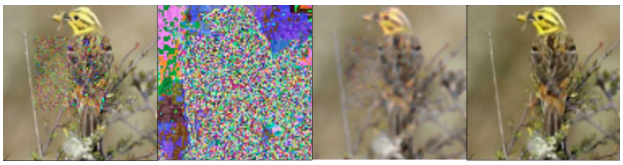

Figure 9. Partially Blurred-Out image of bird. From left to right: Noisy, Conv-1 Visualization, Denoised, Clean

Lastly, applying the learned weights of DSC model to the animal classification problem, we observe a fast convergence in terms of the Cross Entropy loss, unlike that of the truncated normal initialization previously. In terms of accuracy, it is observable that initial accuracy is much higher than normal initialization, with 15% correctly classified by iteration 20 compared to 5% before. Yet, even with the fast convergence, accuracy over test label images seems

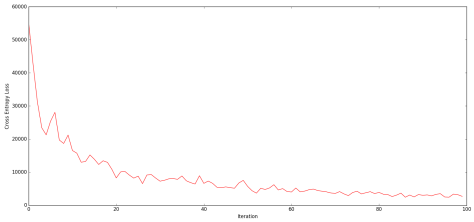

Figure 10. Cross Entropy Classification Loss with Pre-trained weights

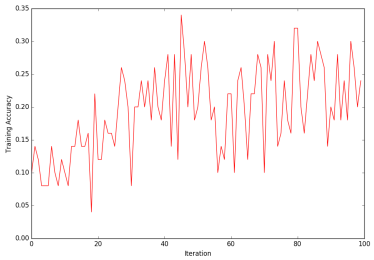

Figure 11. Accuracy Graph of DSC Weights Pre-trained Initialization Classification

to plateau around 35%, much like the non-pretrained classifier. This indicates most likely more data is needed to travel out of the local optimum, and does not invalidate the effectiveness of pre-trained weights. It is predictable that with more labelled data, the pre-trained classifier would reach optimal accuracy faster, due to learned representations of object segmentation in the weights.

6. Conclusion

In this project, a deep Convolutional-Deconvolutional model with Direct Symmetric Connections is applied to solve the classic task of image denoising. Training over 100k unlabelled images, as well as applying subsequent learned weights to training a classification task over 13k labelled images, the DSC included model performed noticeably better than traditional downsampling-upsample structures. Furthermore, representation is learned through unsupervised training indirectly, as weights when used for pre-training to a classification task converged significantly faster than truncated normal initialization. Future work include examining larger amounts of data for the denoiser to converge towards a better optimum, as well as finding better Perceptual Loss metrics for alternative Loss Function training.

References

K. Dabov, A. Foi, V. Katkovnik, and K. O. Egiazarian. Image denoising by sparse 3-d transform-domain collaborative filtering. IEEE Trans. Image Process., 16(8):20802095, 2007.

S. Osher, M. Burger, D. Goldfarb, J. Xu, and W. Yin, An iterative regularization method for total variation-based image restoration, Multiscale Model. Simul., vol. 4, no. 2, pp. 460489, 2005

P. Vincent, H. Larochelle, Y. Bengio, and P. Manzagol. Extracting and composing robust features with denoising autoencoders. In Proc. Int. Conf. Mach. Learn., pages 10961103, 2008.

Long, J., Shelhamer, E., and Darrell, T. Fully convolutional networks for semantic segmentation.CoRR,abs/1411.4038, 2014

Mao, Xiao-Jiao, Chunhua Shen, and Yu-Bin Yang. "Image Restoration Using Very Deep Convolutional Encoder-Decoder Networks with Symmetric Skip Connections." arXiv preprint arXiv:1603.09056 (2016).

Li, Fei Fei, and Justin Johnson. "ArXiv.org Cs ArXiv:1603.08155." [1603.08155] Perceptual Losses for Real-Time Style Transfer and Super-Resolution. N.p., n.d. Web. 17 Dec. 2016.

Adam Coates, Honglak Lee, Andrew Y. Ng An Analysis of Single Layer Networks in Unsupervised Feature Learning AISTATS, 2011