CS230

Image Restoration of Low-Quality Medical-Diagnostic Images

Fariah Hayee
Department of Electrical Engineering
Stanford University
fariah@stanford.edu

Katherine Sytwu
Department of Applied Physics
Stanford University
ksytwu@stanford.edu

Abstract

Medical diagnostics with retinal images is an active area of research in the deep-learning community. Building on recent progress in image denoising and super-resolution with deep convolutional networks, we explore the possibility of denoising low-resolution retinal images while retaining very small features. While these state-of-the-art techniques can remove noise, they also blur out small features like thin blood vessels, which are necessary for further medical classification. In this work, we compare two convolutional neural network architectures, an autoencoder and a deep convolutional network and find that the deep-netwrok performs better in denoising retinal images with added Gaussian noise, however smaller blood vessels are lost. We then explore the possibility of using a super-resolution GAN to regenerate the small features lost in denoising. We improve network performance by weighing the green channel losses higher as we find that green channel contains most small vessel information.

1 Introduction

Medical images contain subtle features that are crucial for reliable medical diagnosis. Recent developments in deep-learning have enabled detection of diabetic retinopathy and even prediction of cardiovascular disease risks from retinal fundus images [1]. For these classification networks, their attention maps, which highlight regions that contribute the most to the diagnosis, are highest around thin blood vessels. However, these thin blood vessels are only detectable with high resolution images, and thus need a high resolution camera.

Here, we propose to use deep-learning to improve low-quality retinal images such that they can subsequently be used in medical diagnostics. This could enable point-of-care diagnosis (i.e. image classification) with low-quality images (i.e. cheaper instrumental costs). While low-quality encompasses a number of different image problems (i.e. artifacts, down-sampling, blurring, etc.), here we focus on denoising images with Gaussian noise. Importantly, our final output image must retain all the features, including thin and thick blood vessels.

We explore two different architectures to denoise our images: a 9-layer autoencoder network and a 17-layer pre-trained deep convolutional neural network (DCNN) [2,3]. These two networks differ in their approach; the autoencoder attempts to learn the important features of each image, and then reconstruct the image without noise, while the DCNN uses its deep structure to learn the noise of the image, and which is then subtracted from the input. Both architectures have proven successful on general color images, but have yet to be tested on retinal images. Finally, we take our denoiser output and use a generative adversarial network (GAN) to re-generate blood vessels that the denoiser blurred out.

CS230: Deep Learning, Spring 2018, Stanford University, CA. (LateX template borrowed from NIPS 2017.)

2 Related work

Standard denoising techniques like autoencoding and deep-convolutional networks [2,3] have successfully worked on standard image sets. On the single image reconstruction front, filtering (linear, bicubic etc) techniques can be very fast, however they usually blur the features. More complex method require training image pairs and algorithms like edge-direction based on gradient profiles [4] and learning patch-specific regression [5]. Perpetual similarity based generative algorithms [6], which we base our network on, has been actively pursued to generate more real looking images in recent studies.

These denoising and reconstruction techniques have been applied on general images successfully; only recently have gained attraction from medical medical images community. Recent papers have denoised low dosage CT scans [7,8], mammograms, and dental x-ray images [9] using both autoencoder and convolutional neural network architectures.

3 Dataset and Features

Our dataset consists of 462 high resolution images of retinal images used for machine learning classification of exudates and microaneurysms [10]; center cropped to 1696×1696 pixels to remove the black background. 26 images are set aside for the test set and the rest for training. We augment the dataset by cropping the images into 80 pixel patches, creating a training set of 501,888 images for the DCNN. These patches were randomly shuffled and redistributed into 128 image mini-batches. The images were normalized to a value between 0 and 1.

To replicate a noisy, or low-quality image, we applied Gaussian noise to all color channels with a standard deviation of 25/255. This noisy image would then be the input to the denoiser networks. By applying our own distortion, we could quickly create a large labeled dataset for our supervised learning problem.

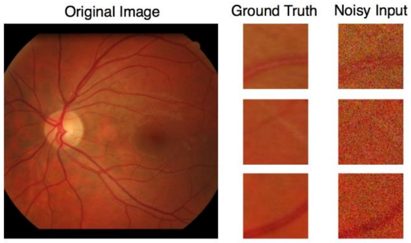

Figure 1: One example image from our dataset. The original image is 1696x1696 pixels in size and the image patches are 80x80 pixels.

4 Methods

4.1 Autoencoder Denoiser

Our baseline network is a convolutional autoencoder network with skip connections, an architecture that has successfully denoised regular color images [3]. There are 4 convolution layers with ReLU activation with increasing number of filters (16, 32, 64, 128) with 3x3 kernel to act as the encoder, and a symmetric set of 4 convolution layers to act as the decoder (the final layer is a convolution layer with 3 filters). While the traditional architecture also has max pool layers in between convolutional layers, both we and previous work [3] found that it blurs the output image and the initial details are lost. Finally, skip connections add every other encoder layer to its symmetric decoder counterpart, which allow for greater detail to be passed through.

4.2 Deep Convolutional Neural Denoiser

To improve on our initial results, we also tested a pre-trained deep convolutional neural network architecture (DnCNN) with a residual learning approach from Zhang, et al. [2]. It consists of 17 identical convolutional layers (64 filters, 3x3 kernel) with ReLU activation, and the interior 15 layers having batch normalization as well. The network learns the noise of the system, or the residual, and so the output is then subtracted from the input noisy image to get the cleaned image.

To improve on this network (see discussion below), we modified the loss function to penalize errors in the green channel more than those in the red or blue channels, giving the following modified loss function:

$$\ell(y, \hat{y}) = ||y_{\text{red}} - \hat{y}_{\text{red}}||_2^2 + 3||y_{\text{green}} - \hat{y}_{\text{green}}||_2^2 + 2||y_{\text{blue}} - \hat{y}_{\text{blue}}||_2^2 \tag{1}$$

The weights for each term in the loss function was chosen arbitrarily; if given more time, we would perform a hyperparameter tuning.

4.3 Super-resolution GAN

To generate the missing thin veins of the retinal images produced by the previous DCNN, we use a convolutional GAN that is trained to estimate a high-resolution (HR) image from the given low-resolution (LR) image. For a training set of N high-resolution images (I_n^{HR}) and corresponding N low-resolution images (I_n^{HR} , the generator (G) will try to minimize:$\hat{\theta}_G = \frac{1}{N} \sum_{n=1}^{N} l^{SR}(G(I_n^{LR}), I_n^{HR})$, where l^{SR} is the loss functions. The discriminatory network, D_{θ_D} is optimized in an alternating manner with G_{θ_G} to solve the adversarial min-max problem:

$$\min_{\theta_G} \max_{\theta_D} \mathbb{E}_{I^{HR}\ p_train(I^{HR})}[\log D_{\theta_D}(I^{HR})] + \mathbb{E}_{I^{LR}\ p_G(I^{LR})}[\log(1 - D_{\theta_D}(G_{\theta_G(I^{LR})}))]. \tag{2}$$

Solving this equation means that the generator G is trained to fool the differentiator D, which is trying to distinguish high-res images from the low-res ones.

The generator network is deep, consisting of 16 residual blocks. Each residual block contains two convolutional layers with 64 filters with 3×3 kernel size, followed by batch-normalization and ReLU activation. The resolution of the input image is increased by two sub-pixel convolution layers. The discriminator network consists of eight convolutional layers with increasing number of filters (64,128,256,512) and a constant 3x3 kernel. A stride of 2 is used to reduce the image dimensions when the number of features is doubled. The last two layers are dense layers, followed by a sigmoid activation layer to get the probability of a sample classification.

The loss function is defined as a weighted sum of the content loss and an adversarial component.

$$l = l_{\text{content} \quad \text{loss}} + 10^{-3} \times \sum_{n-=1}^{N} - \log D_{\theta_D}(G_{\theta_G}(I^{LR})) \tag{3}$$

Previous research has shown that, only MSE-based pixel-wise content loss can often miss high-frequency components. Thus, following Ref.[11], we have added a VGG-loss based on a pre-trained 19-layer VGG net calculating the euclidean distance between the feature representation of the reconstructed image $G_{\theta_G}(I^{LR})$ and the reference image I^{HR}. The second loss term (adversarial component) encourages G to find solutions that are near to the natural solutions.

5 Experiments

The autoencoder was trained on 148,224 patches of our retinal data for 3 epochs from scratch. We used Adam optimization, a learning rate of 0.001, learning rate decay of 0.01, and a loss function of mean-squared pixel-wise loss. The training error saturated at ~ 0.002.

The DCNN was pre-trained on the Berkeley Segmentation Dataset, which consists of color images of everyday objects. We continued training using our training set with a small learning rate so that we did not erase the weights of the original network. We used a learning rate of 3×10^{-6} which was gradually decreased to 2×10^{-7}. The training error saturated at ~ 0.3 (sum over entire mini-batch) which could be improved by even more learning rate decay.

In Figure 2, we show the results for the autoencoder network, pre-trained DCNN, and modified DCNN. We see that all three networks do well in recovering the major features of the retinal image. However, the autoencoder, which does the worst, has not fully removed most of the noise in the background. This could be due to the network architecture, since the skip connection directly adds one of the early encoded layers directly to one of the decoder layers; this could be improved by adding trainable weights to the skip connection.

In Figure 3, we plot out the error in the outputs of the pre-trained and modified DCNN for the three color channels (different test image than that in Figure 2). While the red and blue channels show a

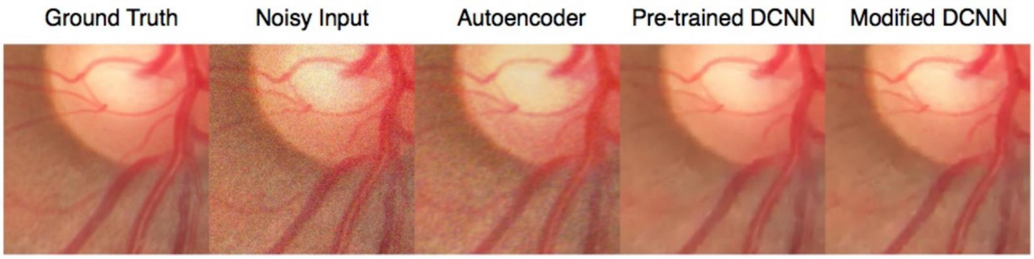

Figure 2: Comparison of the three denoiser architectures against the ground truth and noisy input. All three architectures recover the major features of the image, though the DCNN and modified DCNN have difficulty recovering thin veins (bottom left corner).

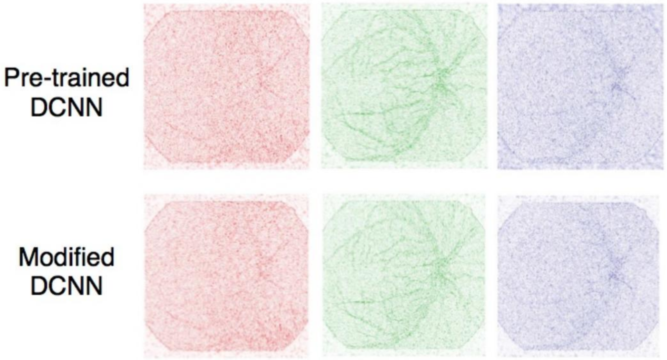

Figure 3: Difference in input vs output for the two DCNN networks (pre-trained on top, modified on bottom) for the three different color channels (red, green, and blue from left to right). Brighter areas represent more error; whiter areas represent less error. As seen from the images, most of the blood vessel structure is encoded in the green channel (and less so in the blue channel).

mostly random distribution of errors, we see a vein-like structure in the green channel. When we modify our loss function to penalize those errors more (modified DCNN), we see that the errors decrease, but are still there. With further training, we believe those errors would decrease further.

While the modified DCNN quantitatively does better than the pre-trained DCNN, we see that qualitatively, there is no noticeable difference. The modified DCNN images have slightly better contrast, but not enough of a visual difference.

SrGAN: We start with a pre-trained network provided by Ref. [11], which has been trained on 350 thousand images from ImageNet. For this training, the LR images are obtained by a bicubic downsampling with a factor of 4. We further train thenetwork on cropped patches of 384 pixels from 128 high-resolution images from our training set. The ground truth images are the patches from our original images whereas the LR image set is generated from the output of the DCNN network. We take mini-batches of 16 images during training, mostly because we are training the network in a CPU (256 GB RAM, 28 cores).

Adam optimizer is used for both G and D. Learning rate is 1^{-4} for both; however, G has a pre-training period where only G is optimized for 30 epochs. We see a loss saturation to 0.01 after 10 epochs at this stage. After this, G_{θ_G} and D_{θ_D} is optimized together.

After a primary training run, we found out that the network is not very successful in reconstructing the smaller blood vessels as can be seen in Fig. 4(c) We find that, like the autoencoder and DnCNN network, most of our loss is in the green channel, so we modified the MSE loss as mentioned in Eq. 1. We then again take the original pre-trained network and train further on our training dataset. The D_{θ_D} loss fluctuates between 0.2 to 1 and the G_{θ_G} loss fluctuates between $0.02 - 0.07$ after epoch 300. We

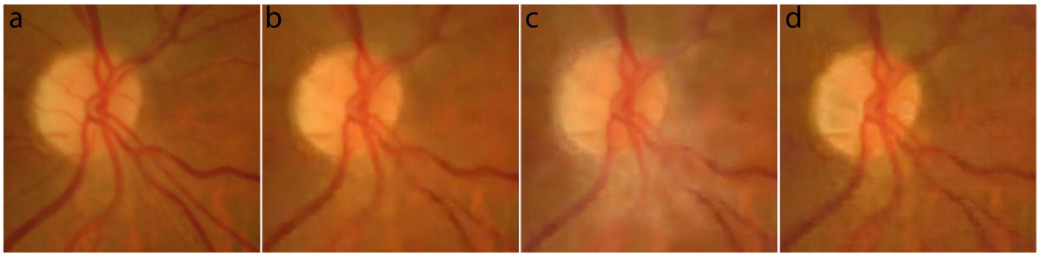

Figure 4: **Sr-GAN performance on a test data** (a) High resolution image (Ground truth) (b) Low-res input to the SrGAN network, which is the output of the DnCNN network. (c) Generated output by original pre-trained network without loss function modification (d) Generated high-res image by $G_{\theta_G}(I^{LR})$ when the loss function is weighted highly for the green channel. The small blood vessels are in the lower part of the image is recovered better in the modified network.

Network Architecture	PSNR	SSIM
Autoencoder	33.00	0.78
Pre-trained CNN	39.96	0.93
Modified CNN	40.43	0.95
Modified SrGAN	28.84	0.938

Table 1: Evaluation metrics of the 4 network architectures used. For both PSNR and SSIM, the higher values are better quality images. As reference, the noisy input has a PSNR of around 22 and a SSIM of around 0.1

use the same test image as before and find out this network performs better sometimes to retain the smaller blood vessels. However, we also notice, it generates 'unreal' features in some cases.

Evaluation: The usual evaluation metric for denoising applications is the peak signal to noise ratio (PSNR), defined as

$$\text{PSNR} = 10 \log_{10}(255^2/\text{MSE}) \tag{4}$$

where 255 is the maximum (peak) value of the image and MSE is the mean-squared error. A higher PSNR value indicates a better image. Another common evaluation metric is the structural similarity index (SSIM) which measures the similarity between images due to contrast, luminescence, and structure. This value is between 0 and 1, with 1 being a perfect reconstruction.

While both PSNR and SSIM are good metrics for whether noise was removed, we found that neither of them fully captured whether small vein structure were blurred out or restructured. As expected, the networks generally become better at denoising as the structure (or loss function) becomes more complex. While the GAN has a high SSIM value due to some successful reconstructions, it also have very low PSNR indicating there are 'unreal' features in the images, so we think this network should not be used for further medical diagnosis.

6 Conclusion/Future Work

In summary, we trained and/or modified three different network architectures (autoencoder, deep convolutional network, GAN) to denoise retinal images. By modifying the loss function to penalize errors in the green-channel more than errors The modified DCNN network performed the best out of all three, but still blurred out some of the small vein structures.

In future, we would further train the modified DCNN with a smaller learning rate and the srGAN with a larger training set. We would also modify the loss function to include a term that promotes vein structure, and/or a term that promotes contrast. We would also perform a hyperparameter search for the weighted loss function. Finally, we would also use a new evaluation metric that would capture how many vein structures were lost in the reconstruction in addition to PSNR and SSIM.

7 Contributions

F.H. and K.S. both came up with the project idea. Both F.H. and K.S. worked on the DCNN. K.S. worked on the autoencoder while F.H. worked on the srGAN.

References

[1] Poplin, R., et al. (2018) “Prediction of cardiovascular risk factors from retinal fundus photographs via deep learning.” *Nature Biomedical Engineering*

[2] Zhang, K., Zuo, W., Chen, Y., Meng, D., & Zhang, L. (2017) “Beyond a Gaussian Denoiser: Residual Learning of Deep CNN for Image Denoising." *IEEE Transactions on Image Processing.* **26**(7):3142-3155

[3] Mao, X.J., et al. (2016) “Image Restoration Using Convolutional Auto-encoders with Symmetric Skip Connections.” *NIPS*

[4] Y.-W. Tai, S. Liu, M. S. Brown, and S. Lin (2010) “Super Resolution usingEdge Prior and Single Image Detail Synthesis.” *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* 2400–2407

[5] D. Dai, R. Timofte, and L. Van Gool (2015) “Jointly optimized regressors for image super-resolution”. *Computer Graphics Forum* (34):95–104

[6] Bruna, J. et al. (2016) “Super-Resolution with Deep Convolutional Sufficient Statistics”, arXiv preprint arXiv:1511.05666.

[7] Chen, H., et al. (2017) “Low-dose CT via convolutional neural network.” *Biomedical optics express*, 8(2), 679-694.

[8] Nishio, M., et al. (2017) “Convolutional auto-encoder for image denoising of ultra-low-dose CT.” *Heliyon*, 3(8), e00393.

[9] Gondara, L. (2016). “Medical image denoising using convolutional denoising autoencoders.” *Data Mining Workshops (ICDMW), 2016 IEEE 16th International Conference* (pp. 241-246). IEEE.

[10] Decencière E, et al. (2013) “TeleOphta: Machine learning and image processing methods for teleophthalmology.” *IRBM* , http://dx.doi.org/10.1016/j.irbm.2013.01.010

[11] Ledig, C., et al.(2016) “Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network.” *IEEE Conference on Computer Vision and Pattern Recognition*

[12] Abadi, M., et al. (2015) "TensorFlow: Large-scale machine learning on heterogeneous systems." Software available from tensorflow.org.

[13] Vedaldi, A., et al. (2015) "MatConvNet – Convolutional Neural Networks for MATLAB" *Proceeding of the ACM Int. Conf. on Multimedia*