Super-resolution of diabetic retinopathy images using an RBF network

ECE 477/595

Adam Saunders

Introduction

Diabetic retinopathy (DR) is a disease that affects the nerves in the eyes of people with diabetes, and it can lead to blindness. In fact, it is the leading cause of new cases of blindness in the developed world [1]. Diagnosis of DR can be tedious, so we can use computer-aided diagnosis systems to help detect and grade the severity of DR. Techniques like super-resolution (SR) can help increase the fine details on an image of the eye, helping make diagnosis easier. In this project, we create an RBF network to achieve SR on DR images.

For this project, we use the APTOS 2019 dataset, a publicly available dataset of DR images of varying size [2]. The dataset provides grades for the images from 0 (no DR) to 4 (proliferative DR).

Methods

This project implements a previous RBF network designed for SR [3]. However, here we apply the network to DR images. First, we must preprocess the images. The original images are in color and vary in size. So, we convert the image to grayscale and resize to 128 by 128 pixels. This image is our high-resolution (HR) image.

Next, we must create a low-resolution (LR) image to model an imperfect, noisy sampling of the HR image. We apply a Gaussian blur with a filter size of 5 and a standard deviation of 0.5. Next, we sample every other pixel to obtain a 64 by 64 pixel image. Then, we add Gaussian noise with a mean of 0 and a standard deviation of 1×10^{-4}. Finally, since the network requires the input image to be the same size as the output image, we use bicubic interpolation to resize the image to 128 by 128 pixels. Figure 1 shows the data preprocessing method.

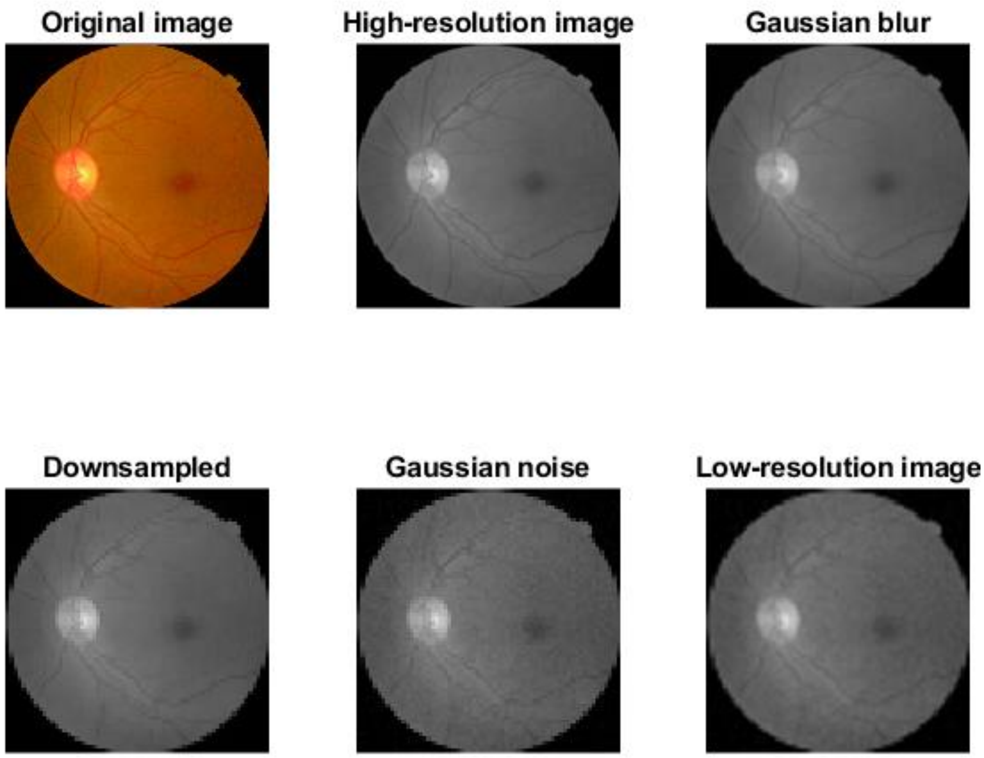

Figure 1: Image preprocessing

The network will train on the LR and HR image pairs. For each LR image, we zero-pad the images by 2. Then, we sample a 5 by 5 pixel window from the LR image. We flatten this window into a 25 by 1 vector and input this vector into the RBF network. The RBF network has a single output node that creates a value in between 0 and 1. This output represents the SR pixel at the center of the 5 by 5 window. So, we slide the window around the image, creating a SR pixel for each 5 by 5 window sampled from the LR image. Figure 2 shows how we generate the SR image.

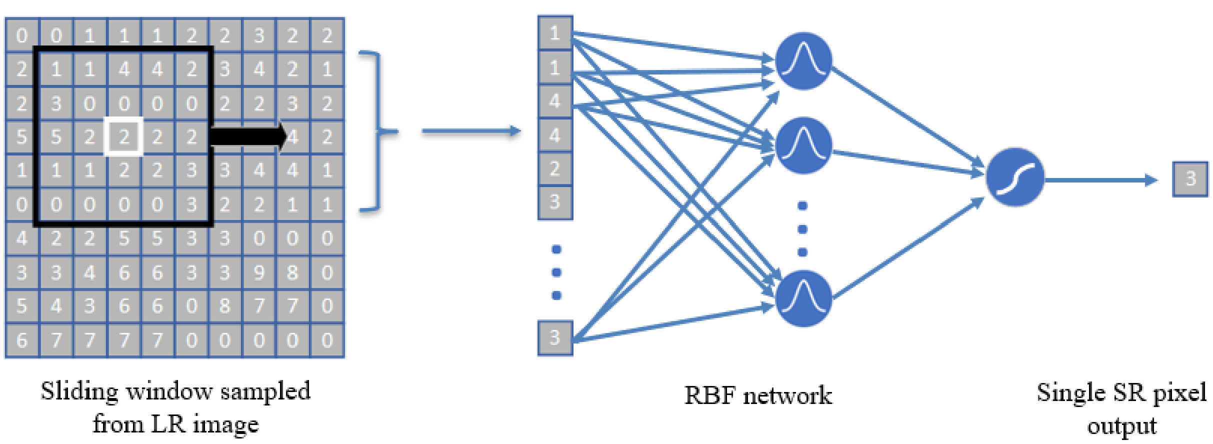

Figure 2: SR image generation

The RBF network architecture is similar to what we used for Assignment 3. We use k-means clustering to cluster the training data into 100 clusters. Then, we calculate the value of the Gaussian function at the squared distance between the input vector and the cluster centers. We input these values into a single-layer perceptron layer with only one output node. We use a sigmoid activation function on the output node to scale the output between 0 and 1. Finally, we use the delta rule to update the weights. For testing, we do not update the weights and we use the cluster centers found for training. Figure 3 shows a flowchart summarizing the training and testing process.

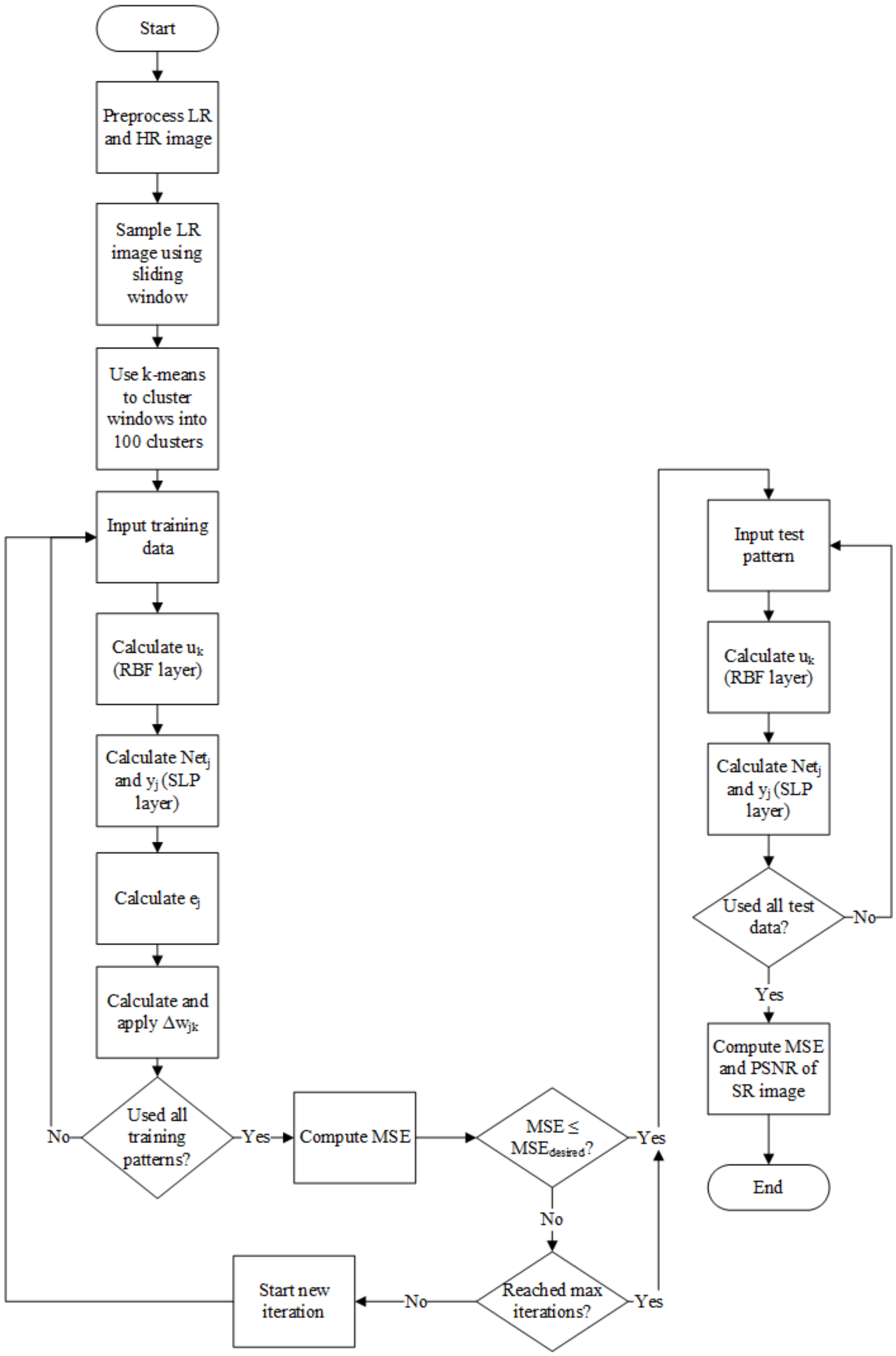

Figure 3: SR RBF training and testing process

We train a model for each grade of DR severity using two training images and one testing image. We select similar-looking images by hand since the APTOS 2019 dataset has a large variety of images. We train the network for 10,000 iterations using a learning rate of 0.01. For the testing images, we calculate the peak signal-to-noise ratio (PSNR) and mean squared error (MSE) between the LR and HR image and between the SR and HR image.

Results

Figure 4 shows a comparison of the network outputs for grade 0 (no DR) and grade 4 (proliferative DR). Qualitatively, the network appears to have reduced the aliasing along the borders of the LR image. Also, the network removed some of the noise from the LR image.

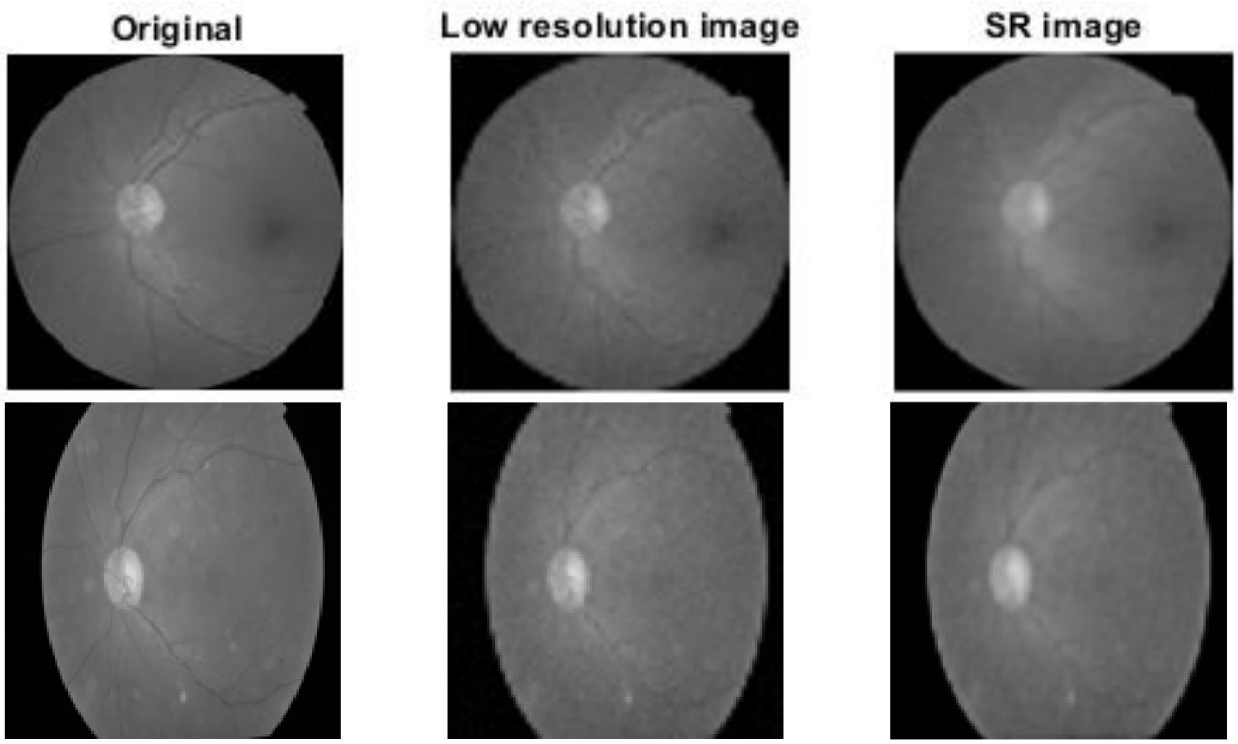

Figure 4: Example output from SR RBF network (top: no DR, bottom: proliferative DR)

Next, Figure 5 shows a comparison of the PSNR and MSE of the images. Notice that the SR RBF network achieved a higher PSNR and a lower MSE for each grade of DR. The average increase in PSNR was 2.46 dB, and the average decrease in MSE was 3×10^{-4}.

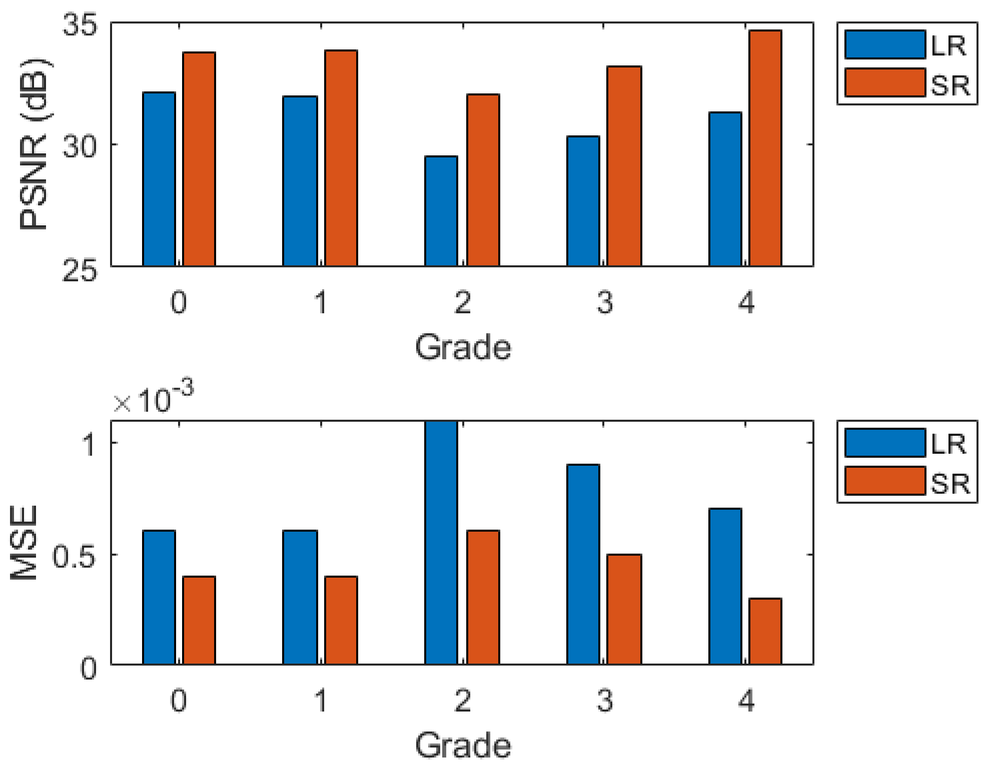

Figure 5: Comparison of PSNR and MSE for LR and SR image

Discussion

The network was successful at increasing PSNR and decreasing in MSE for each grade tested. However, we do not know if this increase in PSNR is due to the network removing noise or actually increasing the high-frequency content of the image. This SR RBF network is useful as a preprocessing technique, especially since it appears to remove noise and increase the quality of the image. Additionally, it needs very few images to train.

The network would be more useful if it could take advantage of the original resolution of the images. Also, the network could take advantage of the color provided by the original images. Future improvements should also focus on creating one model for all of the grades of DR. If we wish to use this network as a preprocessing technique for a computer-aided diagnosis system, then we would not know the grade of the image ahead of time. So, we need one model for all of the grades of DR. However, the model proposed here provides a starting point for future designs, as it was successful in achieving SR.

References

[1] "Diabetic Retinopathy," *National Center for Chronic Disease Prevention and Health Promotion.* [Online]. Available: https://www.cdc.gov/visionhealth/pdf/factsheet.pdf.

[2] "APTOS 2019 Blindness Detection," *Kaggle*, 2019. [Online]. Available: https://www.kaggle.com/competitions/aptos2019-blindness-detection/.

[3] S. Bing-hua, J. Wei-qi, and N. Li-hong, "Radial Basis Function Neural Network Based Super-Resolution Restoration for an Undersampled Image," *Journal of Beijing Institute of Technology*, vol. 13, no. 2, 2004.