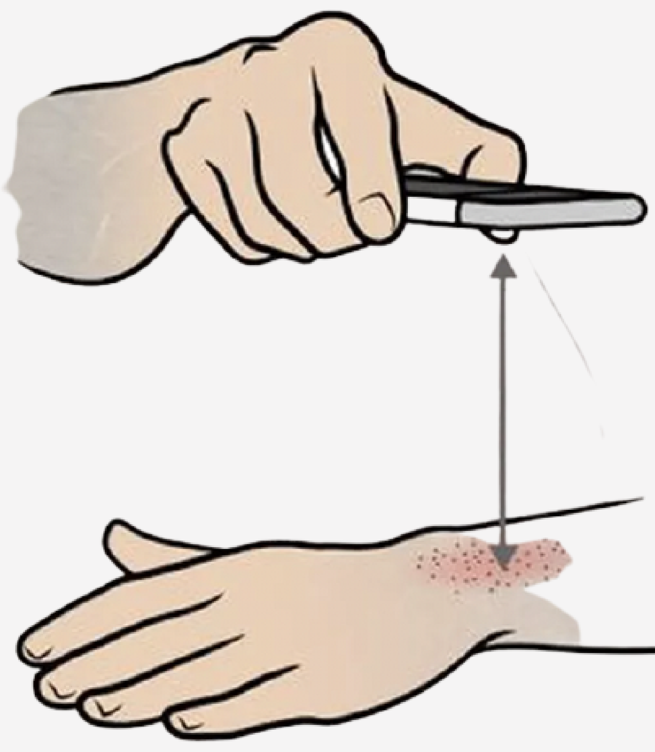
DTU Compute
Department of Applied Mathematics and Computer Science

Skin images quality improvement

A. Contini, A. Rossi, C. Papadopoulos & H. Y. Lam
Student IDs: s210197, s212153, s210603, s212087



Introduction

The necessity of dealing with this problem arises from the quality issues encountered in the image quality of photos sent in by patients. Improving the light and shade conditions would be of great help for the diagnosis of skin conditions, both for dermatologists and classification networks. Given a set of high-quality images, we applied transformations that mimic the common quality issues. Then, we trained an Autoencoder to reconstruct the original, high-quality, image.

ISIC Data

The International Skin Imaging Collaboration (ISIC)[1] is an organization aiming to facilitate the application of digital skin imaging to reduce melanoma mortality. Their archive contains over 70,000 public images of any kind of skin disease. We **selected 843 images** from 3 different years (2016, 2018, 2019) with heterogeneous skin tans, but mostly of a quality we deemed adequate as our target. We randomly sampled **30%** of them as **validation set**, while as a **test set** we found **50 pictures of arguable quality**.

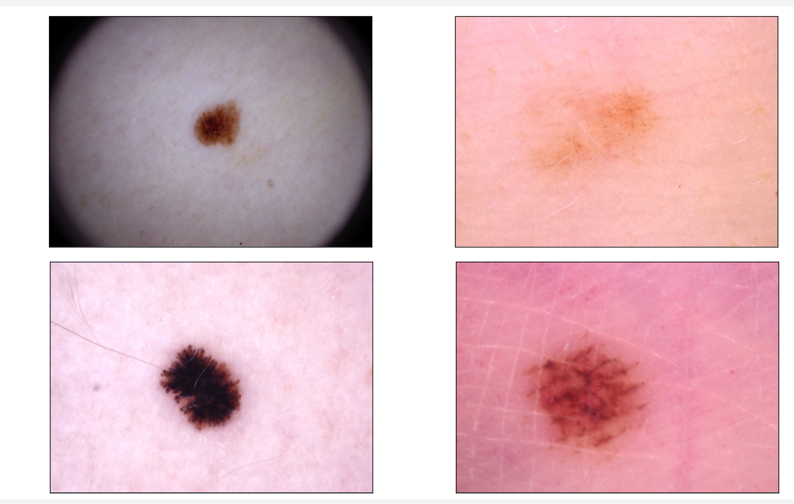


Figure 1: Pink and white skin images with different lightning.

Noises applied to images

The aim is to **improve the image quality, regardless of the starting brightness level**. To make this problem a supervised task, we took our training dataset, and applied an artificial noise to them. In this way, our network will learn to reconstruct the image without the noise, trying also to not hurt resolution. We took inspiration from the image themselves for the noise to apply: Color Jittering was our choice, which can turn an image from very light to very dark and vice versa. We applied this **random modification to 60% of the images**, to let the network also learn from already good images, and at each epoch the set of modified images is different. This will result in **regularization behavior**.

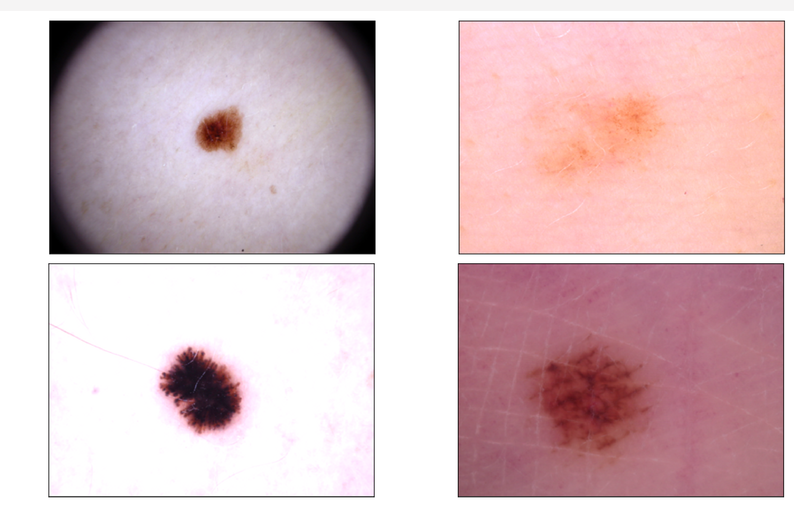


Figure 2: Images after applying the noises.

Network architecture

We have chosen (among many different architectures) a **denoising Autoencoder**[2] with **skip-connections**[3] (Fig. 3), using only convolutional layers and dropout after each layer, with Leaky ReLU activation function. This architecture has proven to be the best performing, both in terms of loss and visual results.

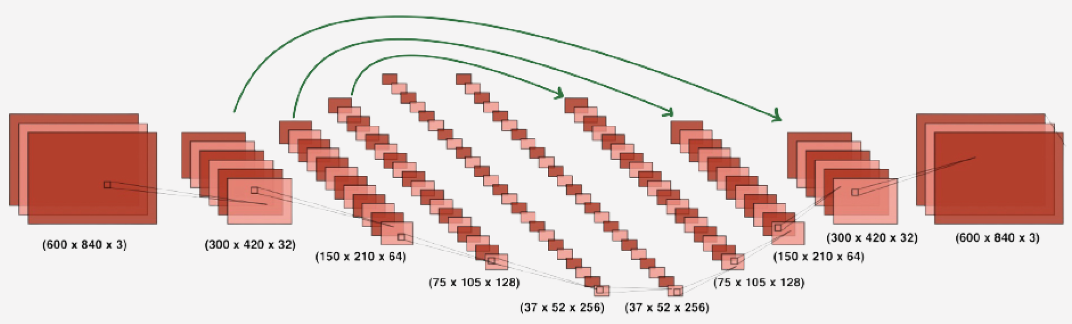


Figure 3: Architecture of the neural network.

No linear and max pooling layers: we noticed from the very beginning that dense layers were not beneficial for our task. This is consistent with the work in [4]. A decrease in performance was also observed adding Pooling layers.

Skip connections: To implement a deeper network we added skip connections which take the activation from one layer in the encoder and feed it to a deeper one in the decoder. This avoids vanishing/exploding gradients and doesn't hurt performance.

Up-sampling for increase resolution: To increase the resolution of the images, we artificially decreased the size of the images and then resized them back to normal resulting in a lower resolution as inputs. Then through the comparison between the reconstructed and the original image, the Autoencoder was able to output better resolution images.

Parameter tuning: We ran many trials and ended up with the current network parameters: 100 Epochs, learning rate of 0.002, batch size of 16 images, and 4 convolutional layers with dropout layers with 15% drop rate in between.

Loss functions

After running several trials we filtered out bad performing losses like MSE and PSNR. The remaining loss functions we ended up with were: MAE, SSIM and a mix of MAE and SSIM. Out of those **SSIM performed the best** both on the loss and the perceived image quality.

Results

In Fig.4. is the train/validation loss along with few images taken from the validation set. We can clearly see that we have significant improvements without any display of overfitting.

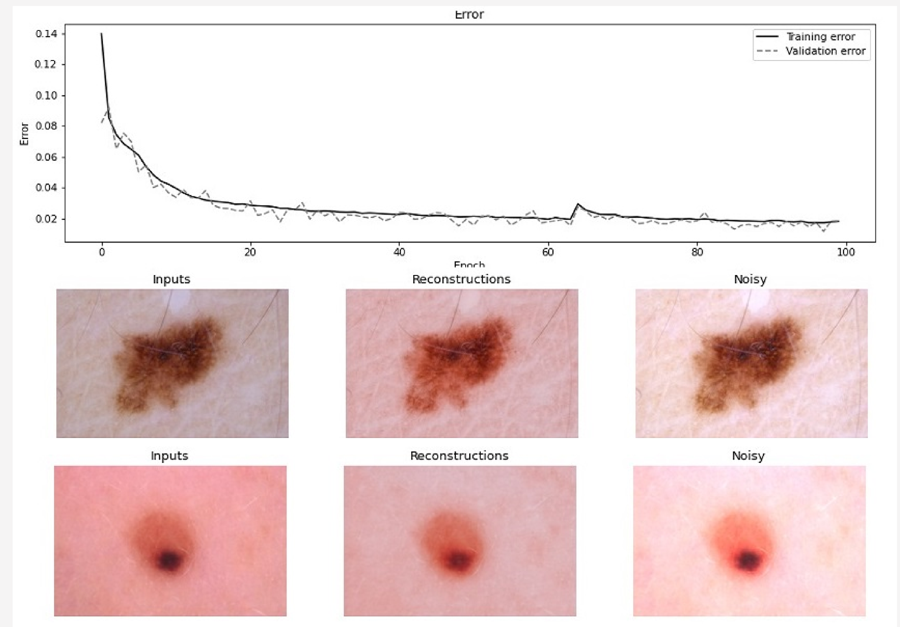


Figure 4: The training outcome.

Results on the independent test set: In Fig.5 we have the outcome of the test set images. On the left we can see the original image while on right we have their reconstruction done by our network.

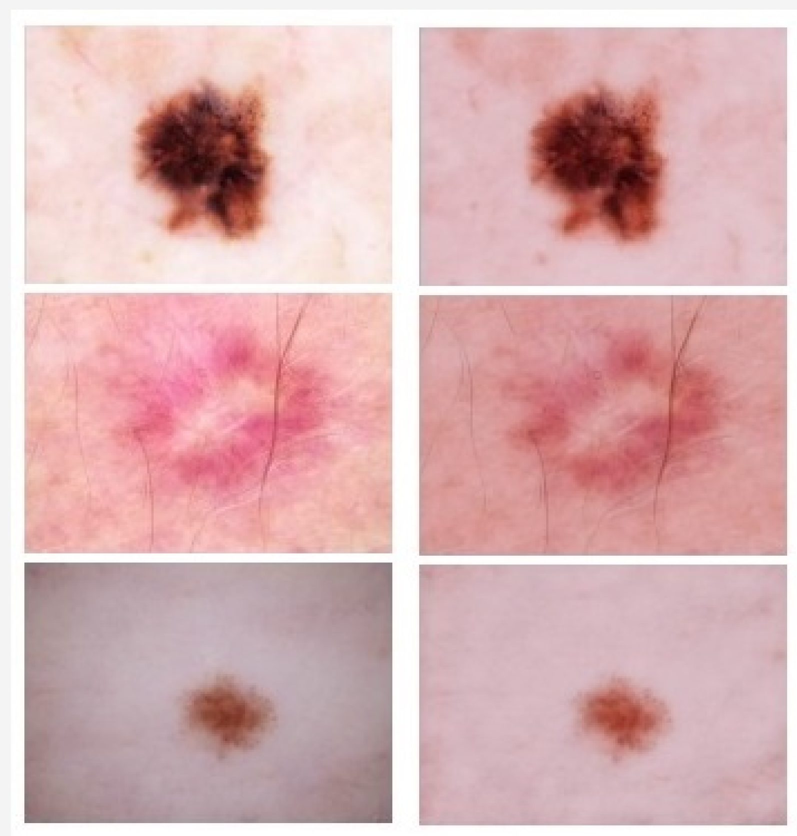


Figure 5: Some test images.

Further developments

1. Using a model-based hyperparameter search to identify the best network parameters.
2. Add classifier on top of the Autoencoder to improve the overall performance of the network to retain details while denoising the images.

References

[1] https://www.isic-archive.com/!/topWithHeader/tightContentTop/about/aboutIsicOverview
[2] Vincent, Pascal; Larochelle, Hugo (2010). "Stacked Denoising Autoencoders: Learning Useful Representations in a Deep Network with a Local Denoising Criterion". Journal of Machine Learning Research. 11: 3371–3408.
[3] Martinez,Bibiloni and Hidalgo, IEEE (2021) "Hair Segmentation and Removal in Dermoscopic Images Using Deep Learning"
[4] https://github.com/rtflynn/Cifar-Autoencoderperformant-autoencoders
[5] Juhwan Kim, Seokyong Song and Son-Cheol Yu, IEEE (2017) "Denoising auto-encoder based image enhancement for high resolution sonar image"