An End-to-End Pipeline for Medical Image Enhancement using GANs Architecture

Apoorv Dankar, Tirth Patel, Jaskaran Singh

Abstract—Medical Imaging is used by radiologists for diagnostic purposes and to check for abnormalities, and these imaging techniques involve radiation. Overexposure to radiation can have an adverse impact on the human body, and using less radiation gives us a noisy output. Hence, radiologists find it difficult as there is a trade-off between the amount of radiation that can be used and the quality of the image. Moreover, noise in medical images can occur due to fluctuation of photons, a reflection of radiations from the subject, or due to instrumental vibration or faults. The proposed approach is a pipeline which starts with denoising using GANs architecture, in which two models have been trained, one for handling all kinds of noise and the second one specifically for Poisson noise. Further, post-processing methods like single-shot HDR using Retinex Filtering and Edge Enhancement using unsharp masking have been done to get a structurally more similar and enhanced denoised image.

Index Terms—Computational Imaging, Medical Imaging, Denoising, Generative Adversarial Networks, HDR, Edge enhancement

1 Introduction

With the advancement in deep neural networks, the use and applications of computer vision have risen quite rapidly in multiple domains ranging from robotics, military, and autonomous vehicles to medical image processing. The medical computer vision field is developing quickly as this domain uses the theoretical aspects of computational imaging methods and applies them to real-world problems that directly help the populace. This motivated us to use the techniques to solve such problems.

Most medical imaging techniques, such as X-rays, CT scans, etc., all rely on radiation. We need better-quality images so doctors and physicians can accurately and efficiently extract useful information about the underlying concern from these images. To get a better image with less noise, we need to expose the patients to high doses of radiation that can have adverse effects on the human body [1]. It is a constant struggle for radiographers to find a balanced exposure so that the image produced is not noisy. X-rays are also very susceptible to noise because of uneven scattering of photons which causes the receptors to receive different amounts of photons which causes Poisson noise in the images [2]. Apart from this, how the film is processed and handled is another way of introducing noise into the final image. On the other hand, speckle noise manifests as a granular look in an image [2]. It is caused by random variations in the return signal from an object that isn't discovered to be larger than a single image processing component [3]. We saw this as an opportunity to develop a pipeline which, given a noisy x-ray image, constructs a denoised and visually similar image. The proposed pipeline consists of multi-tier GANs-based [4] denoising architecture, which focuses on denoising different types of noises with different importance. We then post-process the images with multiple computational imaging techniques to make them visually similar to the ground truth image, thus retaining the maximum possible information even after denoising. The main reason for this is that after denoising, the images tend to blur out, which can lead to the loss of essential features of the image, like edges and contrast among different parts under observation.

This will help the patients being x-rayed to face reduced amounts of radiation which will be better for their bodies without compromising the quality of the X-rays. This makes it easier for doctors to deduce information from these images.

- *Apoorv Dankar is with the Department of Computer Science, University of Toronto. E-mail: dankar@cs.toronto.edu*
- *Tirth Patel is with the Department of Computer Science, University of Toronto. E-mail: tirth@cs.toronto.edu*
- *Jaskaran Singh is with the Department of Computer Science, University of Toronto. E-mail: jaskaransb@cs.toronto.edu*

2 Related Work

Many conventional and new methods have been used to denoise and enhance medical images to produce images with less noise and more details. The conventional methods [5] used for image denoising include linear and non-linear filters, which come under spatial domain filtering. These filters have Gaussian Filters, Wiener Filtering, Median Filters, Bilateral Filters and a few others. These filters can eliminate a decent amount of noise, but that comes at the trade-off of image blurring and losing edge details. Bilateral and Non-local means have provided decent results for denoising medical images [6] when preprocessing is done before applying denoising techniques. Thresholding techniques had to be used before applying Bilateral and NLM filters to preserve edges and details. Comparing all the conventional methods for medical images [7], it is evident that a particular way performs better on a specific type of noise only.

Nowadays, deep-learning based denoising methods are prevalent, primarily based on CNNs. DnCNN is a modern method which introduced batch standardization and

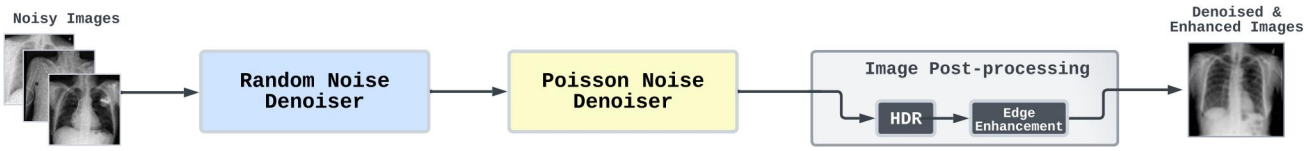

Fig. 1. Proposed pipeline for Medical Image Enhancement

residual learning for denoising images [7] [8]. The main demerit of the neural network methods for denoising is that the learning is done on particular noise levels only, and they don't perform well on different noise levels [8]. CNN architecture may move away from traditional methodologies and toward deep learning methods, yet, the significant difficulty remains computational time and space. TV (Total Variation) based regularization methods [7] have also been proposed for denoising, and these are useful in solving the issue of smoothness, but they have drawbacks. Flat areas are approximated by a fixed constant resulting in a staircase effect and sometimes loss of contrast [9].

The latest methods to denoise include denoising using generative adversarial networks. Denoising using GANs has been found to denoise real-world images, and the architecture of the model consists of the use of residual blocks, skip connections, and batch normalization. The images look real while preserving the edge details and avoiding blurriness [10]. The learning-based methods for denoising the images that use neural networks are trained on noisy images that are artificially created from clean images with a known type of noise. Real-world noisy images are very different from synthetic ones and conventional methods outperform learning-based methods when we denoise real-world images [11]. Denoising using GANs follows a better approach in the model building; in this approach, the model is trained on noise instead of noisy images and denoises real images as well [11].

As mentioned before, denoising images generate blurriness and can remove essential details from the images, which is very harmful in the case of medical imaging. Hence, it's necessary to perform some preprocessing or post-processing on images before or after performing the denoising [7] [12]. HDR (High Dynamic Range) is a technique that can be used on X-ray medical images to enhance and improve edge details [12]. The quality of the image formed depends upon the amount of radiation (mAs) and peak photon energy (kVp). Hence, images generated using different peak photon energy exhibit variable visibility of bones and tissues [12]. Therefore, combining x-ray images with varying photons of peak energies and getting a single image can provide a more enhanced and detailed result. HDR images are generated using a set of low dynamic range images of different exposures and then combining them, but due to the unavailability of x-ray images with different peak photon energy, [13] presents us with an algorithm to obtain HDR image from a single image by generating virtual images of different exposures and then fusing these multi-level illuminations. In this algorithm, the image is broken down into its reflectance and illumination components. The brighter areas are improved by scaling the reflectance component, and the illumination component can be turned up or down to generate images with different illuminations [13].

Another post-processing technique is to increase image sharpness by using the method of unsharp masking [14]. A blurred image is generated, which is then subtracted from the original image to obtain an edge-enhanced image. It detects the borders of the various tones and boosts contrast to make the image look sharper [14].

3 Proposed Method

The main contribution of this paper is a novel end-to-end pipeline for denoising and enhancement of medical images, x-rays in particular. The first and second parts of the pipeline are dedicated to removing the various types of noises present in radiographed images.

3.1 The Pipeline

As mentioned earlier, we devise a multi-tier denoising architecture that follows a "T"-shaped denoising technique to provide breadth and depth of denoising across different types of noises prevalent in these images. The first denoiser offers the range of denoising by denoising the image for different types of noises, viz Poisson noise, Gaussian noise and speckle noise, which are frequently observed in x-rays [2]. Since this component has tried to learn all kinds of noise, it gives average performance across all noise inputs (equal importance to each noise type). The second level of denoising provides depth by focusing on removing the Poisson noise, which is the primary source of noise in radiographed images [2]. The denoiser used in this component specializes in removing Poisson noise, thus giving higher importance to it in the overall structure.

One of the main issues with all the conventional denoising techniques was that they blur out the images quite a lot [5]. The reason behind this is that they try to average out the neighboring pixels to get rid of the randomness of the noise. But in this process, they lose high-frequency details like edges in the images, thus reducing the visual similarity with the ground truth images. In natural images, this might work to some extent. However, for medical images, this is highly problematic because edges contain many essential details that are useful in disease detection and general inference [7]. In our denoising case, the blurring out of the edge, although much less than traditional methods, still exists. To enhance our images to contain such important information to

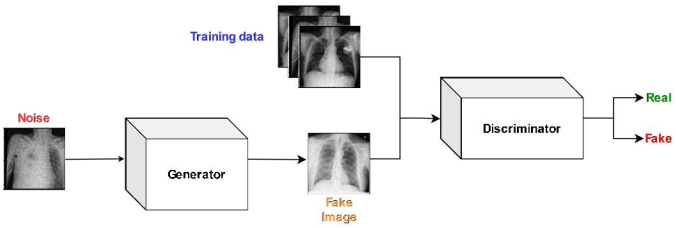

Fig. 2. GAN Architecture for Denoising Image [11]

the maximum possible extent, we perform some post-processing methods to increase the visual similarity with the ground truth images. The methods we have considered for image enhancement are high dynamic range and edge preservation techniques. The output image produced by the denoiser is fed to a post-processor component, where the image is converted to an HDR image using a single-shot HDR conversion mechanism [13]. Then this image is passed through an edge enhancer which reverses the effect of the blur, which crept in the denoising process to make the edge more prominent.

GANs for Denoising

In a generative adversarial network (GAN) [4], two neural networks compete against one another in the form of a zero-sum game, where one agent's gain is another agent's loss (Fig. 2).

Let x be the data representing an image. The discriminator network, or $D(x)$, generates a scalar probability that x came from training data instead of fake image generated by the generator. Reasonably, $D(x)$ ought to be high when x came from training data and low when x came from the generator. Since we are working with X-ray images, in this case, $D(x)$ requires an image with a resolution of 128×128. With z acting as a latent space vector, the generator is now defined by $G(z)$. The generator's objective is to determine the distribution $p_{train}$from which training data came to produce fake samples from it.

As a result, $D(G(z))$ represents the likelihood (scalar) that the generator $G's$ output is the actual image. The fundamental concept behind a GAN model is to have a Generator G that is trained to generate the desired image from noisy or down-sampled input, and a Discriminator D that is trained to distinguish between the original image and the generated image. The adversarial generator and discriminator model are trained simultaneously so that after training, the generator would be proficient at producing realistic-looking images. As a result, D seeks to increase the likelihood that it correctly distinguishes between reals and fakes $(logD(x))$, and G seeks to reduce the likelihood that D will anticipate that its outputs are fake $(log(1-D(G(z)))$. Thus, the loss function is defined as [4]

$$\min_G \max_D V(D,G) = \mathbb{E}_{x\sim p_{\text{data}}(x)}[\log D(x)] + \mathbb{E}_{z\sim p_z(z)}[\log(1-D(G(z)))] \quad (1)$$

3.1.1 Random Noise Denoiser

Fig. 3 depicts the Generator model's architecture. We employed deep residual and convolutional neural networks so that the generator could learn to reduce noise and produce the fake image with little loss. Both the parameter selection and the layering order are empirical. For this model, we try to denoise the noisy images, which are a combination of Poisson, Gaussian, and Speckle noise.

In the generator network, we created a deep residual convolutional neural network with eleven convolutional layers, three residual blocks, and two skip connections. Overall, we perform Convolution (decrease channels), apply a ResNet/Residual block, and finally achieve Deconvolution (increase channels). For deconvolution, we resize the image and perform convolution to decrease the channel length. Our model has an architecture of Conv-Residual-Deconv. For the initial task of Convolution, we created a Convolutional block that consists of three different layers. At first, the input is passed through Convolutional Layer. In the next step, Batch Normalization (BN) is applied. Batch Normalization is a technique for training deep neural networks that normalizes the contributions to a layer for every mini-batch. This layer's job is to take the output from Convolutional Layer and normalize them before passing them on as the input of the next layer. BN helps improve the model performance, mitigate the internal covariate shift, and apply a small regularization effect.

Finally, an activation function of the Parametric Rectified Linear Unit (PReLU) is used. The non-saturating activation functions, such as PReLU, enable us to train a deep neural network by solving the vanishing gradient problem [15]. The slope of ReLU is zero in the negative range, so once the neuron gets negative, it is unlikely to recover. It is known as the dying ReLU problem, which can be resolved using ReLU with a non-zero gradient for negative inputs [15]. Hence, the coefficient is introduced as the learnable parameter in the PReLU and is represented as

$$\text{PReLU}(x) = \max(0,x) + a \cdot \min(0,x) \quad (2)$$

As shown in the Fig. 3, there are total 4 Convolutional blocks.

The following layers are the residual layers, where a skip connection between the residual network and the Convolutional block is added. This residual network consists of a Convolutional block which is then added up with the previous output, and finally, a PReLU activation is applied. This combination is stacked up to 3 layers. In this network, the output channels remain changed, i.e., 128 with a filter size of (3×3) with stride=1 and padding=" same."

Finally, we need to upscale or increase the channels of the image. Hence, we resize the image and perform the deconvolution with the help of the convolution block itself. The deconvolution block consists of a resize function and all the layers of the Convolution block. The skip connection passes the input images to the end of the network so the model in between learns only the noise. There are four deconvolution blocks; further, a tanh activation is applied. The tanh activation function is described as below:

$$tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (3)$$

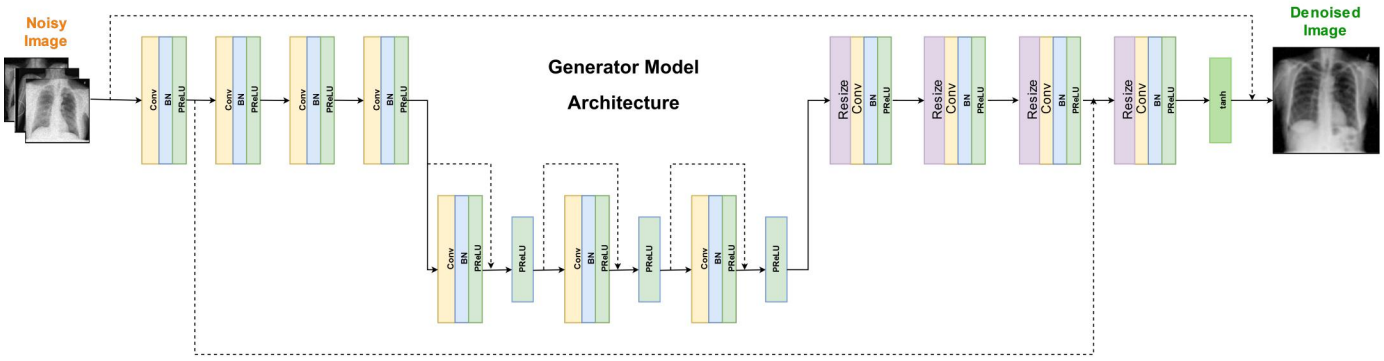

Fig. 3. Generator Architecture

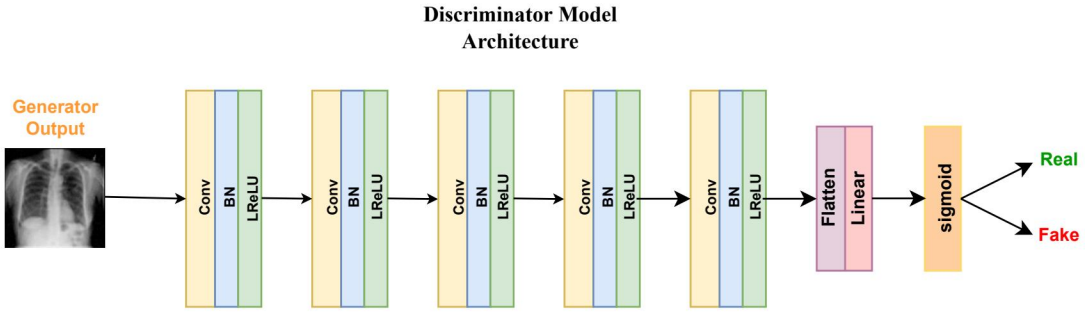

Fig. 4. Discriminator Architecture

This function has a -1 to +1 interval as its range. The benefit of this function is that the zero inputs will be mapped close to zero in the tanh graph and the negative inputs will be highly negative. In order to match the output image range, we employ this activation at the end of the network. Tanh activation makes it possible to attain the noise range [-1,1]. In this manner, the noise pattern can be learned by the generator model, which can then attempt to reduce the noise and produce denoised images that are close to the actual data.

The discriminator network, which outputs the likelihood that the given input (the generator's output) is real or fraudulent, is a straightforward convolutional network. This network is made to produce a score for both the fake and actual images. The network has a great ability to discriminate between genuine and false images, as seen by the fact that it will provide a score very close to 1 for the real label image and 0 for the created image. In our case, the images labels are clear images, and the fake images are obtained by the noised images after passing through the generator.

The network structure is quite simple, which consist of series of Convolution layers, then connects with a full link layer and finally was sent to the sigmoid function. The structure for Discriminator network is shown in the Fig. 4. There are total 5 convolutional blocks, which consists of the convolution layer, batch normalization and a LeakyReLU activation function.

The "drying ReLU" issue is resolved by the LeakyReLU activation function because it does not contain zero-slope components. LeakyReLU is advantageous because we are not dealing with negative values in this situation, and it also expedites the training process because the mean activation is close to zero. It is defined as,

$$\text{LeakyReLU}\ (x) = \max(0, x) + \text{(-ve) slope} \cdot \min(0, x) \quad (4)$$

Then, a fully connected Linear layer with sigmoid activation function at last is added in order to normalize the confidence score to a probability between 0 and 1.

3.1.2 *Poisson Noise Denoiser*

In this section, we try to eliminate the Poisson noise in the medical images and try to enhance the quality of image to achieve greater PSNR/SSIM value. For this model, we created a similar dataset mentioned in the above section, but only adding the Poisson Noise. This component provides the depth part of the "structured denoising.

Further, the similar structure GAN model was trained and the PSNR/SSIM and the Generator/Discriminator losses were noted. The graph for the same is attached below in the Experimental Results section.

3.1.3 *Single shot HDR*

The algorithm implementation from [13] provides us with an approach to obtain HDR images from a single image of a single exposure by generating and combining multi-level illuminations of a single image. The algorithm is based on Retinex theory, in which we estimate the illumination component of the given image, and then the component other than the illuminance component, which is left, acts as the reflectance component. In this approach, the image is divided into its reflectance and illumination channel, and this decomposition is done using the weighted least square filter.

We first obtain the luminance component from the image, and to obtain the reflectance component, we take the difference of luminance and the estimated illumination. For the illumination estimation, we could have gone with the gaussian kernel, but it has a drawback: it can lead to halo artifacts [13] near the borders with bright background. Hence, to solve this drawback, the algorithm suggests implementing WLSF (Weighted Least Square Filter), which solves the halo artifacts problem and is an edge-preserving algorithm, helping us maintain even more details.

Once the decomposition is done, the reflectance channel is scaled up, which helps enhance bright areas of the image. For x-ray images, HDR is performed using images obtained from x-ray beams with different peak photon energy. But, as we have only a single image, we can use the illumination channel to generate different illuminations for the image by scaling the channel up and down. These illuminations will correspond to different peak photon energies. Now, as illumination is adjusted, increasing it in bright areas can lead to saturation of pixel values and in these pixels, sharpness cannot be enhanced easily. To prevent distortion, instead of working on the illumination channel for these pixels, we work on the reflectance channel. Selective reflectance scaling has been used to adjust these kinds of pixels. We fix a certain threshold for illumination, and if we find it higher than the threshold, we tune the reflectance.

For the generation of images with different illumination, a scale factor which is sigmoid function is multiplied by the luminance to darken or brighten the illuminance channel. We find ratios between the standard exposure and under-exposure areas and between over-exposure and under-exposure. Based on these ratios, illumination is increased more in darker regions than in brighter regions. After generating the virtual images with different peak photon energy, we use the tone mapping technique. Tone Mapping has been done using a unique technique, where appropriate weight maps are used to combine all the virtual illumination images. At last, we combine the new illumination and scaled-up reflectance to obtain the final image. The output is an enhanced image with more details.

3.1.4 Edge Enhancement

We have implemented the unsharp masking method to enhance the edges of the image after applying the GAN-based denoising. In this method, we obtain the sharpened image by subtracting the blurred image from the original image. Unsharp masking is characterized in frequency domain language as receiving a highpass filtered image by removing a lowpass filtered version of itself from the provided image. This method has the capability of taking the human visual system response and can sharpen images even in the presence of noise as well.

The high pass filter is used to improve the noisy image utilizing the linear unsharp filtering strategy. Unsharp masks help sharpen images. However, excessive sharpening might cause the image to lose its natural appearance. This approach has two significant limitations: the contrast in the darker section is considerably more profound than in, the lighter part. The approach also increases the noise, which is an issue. In our example, Edge Enhancement was necessary to improve the details in the medical images so that essential components are noticed.

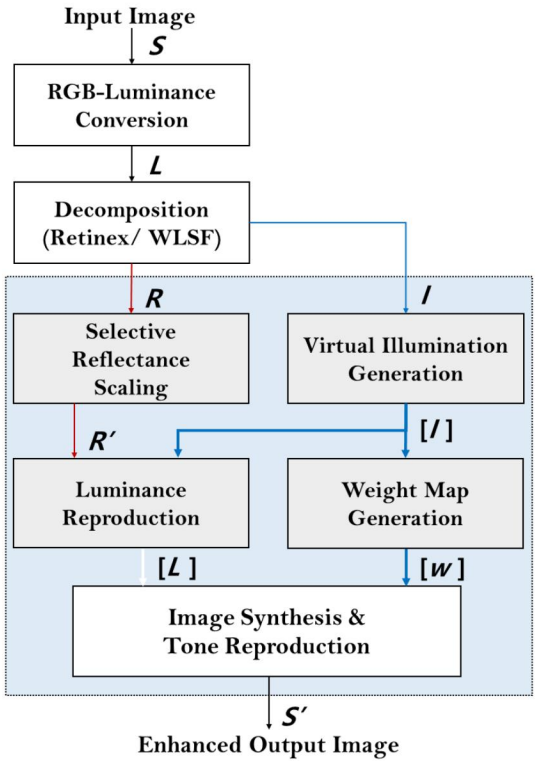

Fig. 5. Single shot HDR (Photo Credits: Park et al. [13])

3.2 Evaluation Metrics

3.2.1 Peak Signal to Noise Ratio

Any image processing could result in a significant loss of quality or information. Objective and subjective methods should be used to evaluate image quality [16], [17]. Subjective approaches don't take into account certain criteria and are based on human judgment [18]. Comparing things using explicit numerical criteria is the foundation of objective methods [19], [20], and there are many possible references, including the ground truth or prior knowledge expressed in terms of statistical parameters and tests [21], [22], [23]. In this project, we use PSNR and SSIM to compare and analyze the quality of the denoised images. Given a denoised image, f and the original image g, both of size M×N, the PSNR between f and g is defined as

$$\mathrm{PSNR}(f,g) = 10\log_{10}\left(\frac{255^2}{MSE(f,g)}\right) \tag{5}$$

$$\text{where, } \mathrm{MSE}(f,g) = \frac{1}{MN}\sum_{i=1}^{N}\sum_{j=1}^{M}(f_{ij}-g_{ij})^2 \tag{6}$$

When MSE approaches to zero, the PSNR tends to Infinity. This shows that higher the PSNR, higher is the quality of the image. On the other hand, lower PSNR means that there is a significant numerical difference between the two images compared.

3.2.2 Structural Similarity Index Measure

The other well-known metric used for measuring similarity between two images is SSIM developed by Wang et al. [24] SSIM is considered to be correlated with the quality perception of HVS(Human Visual System). SSIM is designed by modelling an image distortion as a combination of three factors that are loss of correlation, luminance, and contrast distortion. The SSIM is defined as

$$\mathrm{SSIM}(f,g) = l(f,g) \cdot c(f,g) \cdot s(f,g) \tag{7}$$

where,
$l(f,g) = \frac{2\mu_f\mu_g+C_1}{\mu_f^2+\mu_g^2+c_1}$ This is luminance comparison, measures the closeness of two images.
$c(f,g) = \frac{2\sigma_f\sigma_g+c_2}{\sigma_f^2+\sigma_g^2+c_2}$ This is contrast comparison, measures the closeness of contrast of two images.
$s(f,g) = \frac{\sigma_{fg}+c_s}{\sigma_f\sigma_g+c_3}$ This is structure comparison function, which measures the correlation coefficient between two images.

3.3 Implementation Details

Our implementation of each component was carried out individually and then compiled together and tested. For the implementation of GANs, we coded the structure from scratch and used the layers of convolution, batch normalization, and parametric ReLU provided in the PyTorch package. The preprocessing of the image included them being resized down to 128x128 from 1024x1024. The reason for this was the limited computing resources for the project.

The convolutional layers in the generator model have a varying filter sizes as we down-sample and up-sample the image in the network. During the entire training process, the shapes of the images remained unchanged. In our case, we have a filter size of (9×9) at the first and last forward block. The remaining intermediate layers have a fix (3×3) filter size. The padding is "same," and the stride is set to 1. The convolution layers in the discriminator model have varying output dimensions with different kernel sizes of (3×3) and (5×5) with a stride of 2.

For the random noise denoiser, we added Gaussian, Poisson and Speckle noise to the X-ray images. The Gaussian noise was added with the zero mean gaussian distribution with a standard deviation of 0.03. The Poisson noise was similarly added with a factor of log of unique values of image. Further, the speckle noise was created with the help of Gaussian distribution of standard deviation of 0.01. All these noises were injected into the original image randomly with the help of numpy package in python.

As Poisson noise is prevalent in medical images, we create a similar noisy dataset as above but with only Poisson noise for training Poisson Noise Denoiser.

With the structure of the model ready and the preprocessing of the images done, we started the training steps of the pipeline. We trained the GANs model individually. We trained each model with 200 unique chest x-ray images with different types of noise added randomly to each image. These 200 images were trained for 50 epochs resulting in a total iteration of 10000. The batch size hyperparameter was chosen to be 10. The learning rate used was 0.0002 and was fixed throughout the training process. The optimizer we used was Adam with binary cross entropy loss in the Discriminator model training.

Similarly, we used the Adam optimizer with binary cross-entropy loss for Generator. But in this case, we also added another loss function: the pixel loss, which quantifies the pixel-wise similarity between the images generated by Generator and ground truth. This was implemented using mean squared error loss over all the pixels.

We trained both the models and saved the weights to a ckpt file that can be loaded (we have provided this in our code on https://github.com/apoorvdankar/MedGANs. For the usability of the pipeline, we just load the content of the ckpt file and call the Generator part of the GANs. The test image passes through the Generators of the two denoisers and then through the single shot HDR converter and Edge enhancer to give the final output image.

4 Experimental Results

The model was trained for 50 epochs with the given learning rate and batch size. Fig. 7 represents the graph for Generator and Discriminator Loss for both models. The evaluation metrics for both models is reported in Fig. 8. The detailed comparison of this method with conventional approaches is given in Table 1. Further, in this section, we shall analyze the denoised image qualitatively and quantitatively.

4.1 Qualitative Results

In the leftmost panel of Fig 6, we have four noisy images for qualitative analysis and reporting. We randomly selected these images using a random data loader from the test images dataset directory. We then added above explained noises at random to each of these images. The next image is the output of the first component of the pipeline: Random Noise Denoiser. We observe that the noise level in the image has decreased on average but this has also blurred the images a little bit. The third image is the output from the Poisson Noise denoiser. Here also we see the noise level has decreased but at the cost of blurring out the edges even more. The penultimate image batch is the output of the post-processing component. The results here include the output only of the edge-enhancer. We noticed that adding the HDR component decreases the PSNR quite a lot. We suspect we might have to modify and hyper-tune the parameters of single-shot HDR conversion to get the best results. We need to try out the combination of HDR and EdgeEnhancer to get optimal results consistent with our hypothesis. Here, we have only included the results after applying Edge Enhancer. We notice that this step increases the clarity of edges a bit as compared to the previous step. Finally, the last block is the ground truth image to which the noise was added and all the processing steps were done.

4.2 Quantitative Results

This section will discuss the quantitative aspect of analyzing the denoised Images. All metrics reported in this section are averaged over 15 test images. In the experiment, we had noisy images of PSNR of 25.88 dB and SSIM of 68.52.

The traditional methods give lower PSNR values compared to neural networks. The Gaussian filter denoising

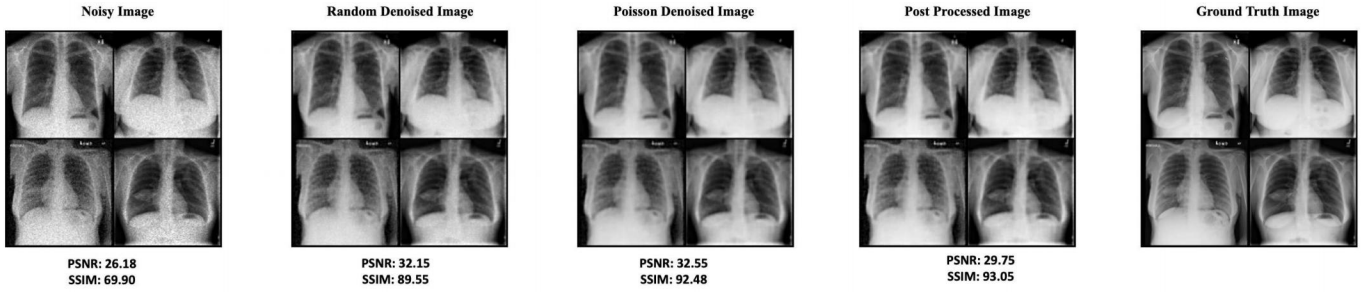

Fig. 6. Component-wise output of the pipeline. **Left to Right:** Noisy Image, Output of Random Noise Denoiser, Output of Poisson Noise Denoiser, Output of Post-processing component, Ground-Truth image. (Metric values averaged over this batch)

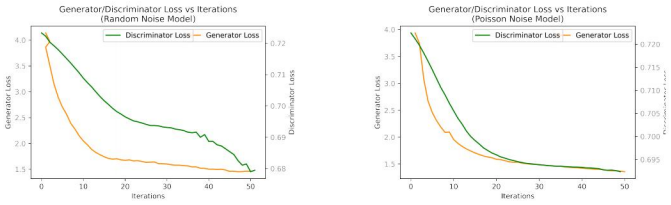

Fig. 7. Generator and Discriminator loss over training epochs for **Left:** Random Noise Denoiser model **Right:** Poisson Noise Denoiser model

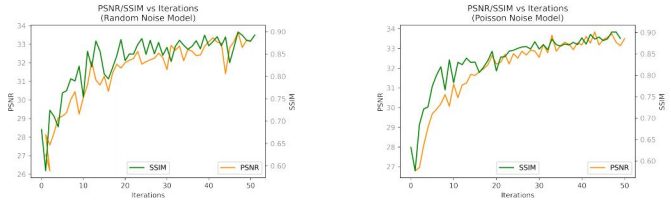

Fig. 8. **Metrics:** PSNR and SSIM value over training epochs for **Left:** Random Noise Denoiser model **Right:** Poisson Noise Denoiser model

with sigma=0.5 gave a denoised image of PSNR 29.59 dB/SSIM 0.86. The Median Filter with a kernel size of 7 gave us an image with PSNR 26.79/SSIM 0.81. Other conventional methods PSNR and SSIM are mentioned in Table 1.

We experimented with modern methods such as ADMM+DnCNN and ADMM+TV solver. ADMM with DnCNN as solver gave good results with the PSNR 30.14 dB and SSIM 0.91.

We also experimented with parameter tuning for each conventional method (e.g., changing the sigma value). The detailed results with PSNR and SSIM for each of the parameters for a particular method are mentioned in table 2.

Our methodology of using a GAN-based Random Noise Denoiser and Poisson Noise Denoiser gave the best metric values. The Random Denoiser gave a denoised image with PSNR 31.92 and SSIM 0.893. The combination of Random Noise and Poisson Noise denoiser gave the PSNR 32.27 and SSIM 0.926. The Edge Enhancement improved the image visual similarity, and hence we achieved the highest SSIM with 0.933. The evaluation metrics for each method are mentioned in Table 1

Method	PSNR	SSIM
Noisy Image	25.88	0.68
Gaussian ($\sigma = 0.5$)	29.59	0.82
Median Filter (size=7)	26.79	0.81
Bilateral ($\sigma = 1, \sigma_$intensity=0.5)	30.79	0.89
NLM ($\sigma = 3$, window=2	28.15	0.84
ADMM + DnCNN solver	30.14	0.91
ADMM + TV	28.73	0.89
GANs (Random Noise)	31.92	0.893
GANs (Random + Poisson)	**32.27**	0.926
GANs + Edge Enhancer	29.36	0.933

TABLE 1
Metric comparison of different methods for denoising

Method	Parameters		PSNR	SSIM
Gaussian Filter	σ (sigma)	0.5	29.59	0.82
		1	27.15	0.90
		2	23.63	0.82
Median Filter	Filter size	6	26.42	0.82
		7	26.79	0.81
		8	25.20	0.78
Bilateral Filter	σ (sigma)	1	30.79	0.89
		2	27.89	0.88
		3	25.91	0.82
NLM	σ (sigma)	1	27.85	0.86
		2	28.06	0.84
		3	28.15	0.84

TABLE 2
Metric comparison of conventional methods with different hyper parameters

5 Conclusion

In this paper, we propose an end-to-end novel pipeline to perform medical image denoising, especially for x-rays, with multi-tier GANs-based denoisers and post-processing steps using single shot HDR and edge enhancement using unsharp masking. We show that our GANs-based denoisers, with their 'T' shaped denoising technique perform better than conventional denoising methods. The post-processing steps enhance the images and make them more visually similar to the ground truth image by amplifying the edges which are quite important in medical images. We note that using such post-processing methods decrease the PSNR but increase the SSIM metric under specific conditions and hyper-parameter values as mentioned earlier in the experimental results section. An important future step will be to

come up with settings such that the PSNR is also increased with these methods. Also it will be very useful to get some domain experts' comments on the outputs of the pipeline which will help us modify the pipeline to be more suited to the demands of the real world.

ACKNOWLEDGMENTS

We would like to thank Prof. David Lindell and the course CSC2529 Computational Imaging (University of Toronto) for providing us with such a wonderful opportunity to work on this topic. We are also very thankful to the course staff and teaching assistants for their guidance and feedback on the project and project proposal. The compute required for the training of the Generative Adversarial Networks was provided by Google Colab notebook GPUs.

REFERENCES

[1] W. Huda and R. B. Abrahams, "Radiographic techniques, contrast, and noise in x-ray imaging," *American Journal of Roentgenology*, vol. 204, no. 2, pp. W126–W131, 2015, pMID: 25615772. [Online]. Available: https://doi.org/10.2214/AJR.14.13116

[2] E. Manson, V. A. Ampoh, E. Fiagbedzi, J. Amuasi, J. J. Flether, and C. Schandorf, "Image noise in radiography and tomography: Causes, effects and reduction techniques," 2019.

[3] P. Hammersberg, M. Stenström, H. Hedtjärn, and M. Mångård, "Image noise in x-ray imaging caused by radiation scattering and source leakage, a qualitative and quantitative analysis," *J Xray Sci Technol*, vol. 8, no. 1, pp. 19–29, Jan. 1998.

[4] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," 2014. [Online]. Available: https://arxiv.org/abs/1406.2661

[5] L. Fan, F. Zhang, H. Fan, and C. Zhang, *Brief review of image denoising techniques*, Jul 2019, vol. 2, no. 1.

[6] M. M. Mohan and V. Sheeba, *A Novel Method of Medical Image Denoising Using Bilateral and NLm Filtering*, 2013.

[7] R. Patil and S. Bhosale, *Medical Image Denoising Techniques: A Review Rajesh Patil*, 01 2022, vol. 4.

[8] V. Murali and P. V. Sudeep, *Image Denoising Using DnCNN: An Exploration Study*, J. Jayakumari, G. K. Karagiannidis, M. Ma, and S. A. Hossain, Eds. Singapore: Springer Singapore, 2020.

[9] L. Rudin and S. Osher, *Total variation based image restoration with free local constraints*, 12 1994, vol. 1.

[10] A. Alsaiari, R. Rustagi, A. Alhakamy, M. M. Thomas, and A. G. Forbes, *Image Denoising Using A Generative Adversarial Network*, 2019.

[11] T. D. Linh, S. M. Nguyen, and M. Arai, *GAN-Based Noise Model for Denoising Real Images*, 2020.

[12] P. Skurowski and K. Wicher, *High Dynamic Range in X-ray Imaging*, 01 2019.

[13] J. S. Park, J. W. Soh, and N. I. Cho, *Generation of High Dynamic Range Illumination from a Single Image for the Enhancement of Undesirably Illuminated Images*. USA: Kluwer Academic Publishers, jul 2019, vol. 78, no. 14. [Online]. Available: https://doi.org/10.1007/s11042-019-7384-z

[14] A. Polesel, G. G. Ramponi, and V. J. Mathews, *Image Enhancement via Adaptive Unsharp Masking*, 02 2000, vol. 9.

[15] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," *CoRR*, vol. abs/1502.01852, 2015. [Online]. Available: http://arxiv.org/abs/1502.01852

[16] R. Kreis, *Issues of spectral quality in clinical 1H-magnetic resonance spectroscopy and a gallery of artifacts*, 10 2004, vol. 17.

[17] I. Avcibas, B. Sankur, and K. Sayood, "Statistical evaluation of quality measures. j electron imaging," *J. Electronic Imaging*, vol. 11, pp. 206–223, 04 2002.

[18] L. Macdonald and M. Luo, "Colour imaging, vision and technology," *Color Research and Application - COLOR RES APPL*, vol. 27, pp. 455–455, 12 2002.

[19] M. Cadik, "Evaluation of two principal approaches to objective image quality assessment," *Proceedings of the International Conference on Information Visualization*, vol. 8, pp. 513 – 518, 08 2004.

[20] T. Nguyen and D. Ziou, "Contextual and non-contextual performance evaluation of edge detectors," *Pattern Recognition Letters*, vol. 21, pp. 805–816, 08 2000.

[21] O. Elbadawy, M. El-Sakka, and M. Kamel, "An information theoretic image-quality measure," vol. 1, pp. 169–172 vol.1, 1998.

[22] A. Medda and V. E. DeBrunner, "Color image quality index based on the uiqi," *2006 IEEE Southwest Symposium on Image Analysis and Interpretation*, pp. 213–217, 2006.

[23] R. Dosselmann and X. Yang, "Existing and emerging image quality metrics," *Canadian Conference on Electrical and Computer Engineering*, vol. 2005, pp. 1906 – 1913, 06 2005.

[24] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.