Edge-guided CNN for Denoising Images from Portable Ultrasound Devices

Yingnan Ma, Fei Yang, Anup Basu
Dept. of Computing Science, University of Alberta, Canada

INTRODUCTION

Ultrasound diagnostic technology is popular in the medical field because of its lower cost compared to MRI and CT devices. It is non-invasive and does not subject patients to harmful radiations, like X-rays and CT. Ultrasound generates images of structures within the human body by using low-power but high-frequency sound waves. However, the quality of ultrasound images captured from the portable scanners is relatively poor compared to standard ultrasound scanning systems in hospitals. To improve the quality of the ultrasound images obtained from portable ultrasound devices, we propose a new neural network architecture called Edge-guided Denoising Convolutional Neural Network (EDCNN), which can preserve significant edge information in ultrasound images when removing noise. We also study and compare the effectiveness of existing deep learning methods and classical filtering approaches in removing speckle noise in these images.

DATA ARRANGEMENT

We utilize our ultrasound dataset to train and test various learning approaches. Data was collected using our Clarius handheld ultrasound device focusing on two parts of the human body; namely, knuckles and heart. Our data includes 500 ultrasound images involving 400 ultrasound images of size 180 x 180 for training, 46 ultrasound images of size 321 x 481 for validation and 54 ultrasound images of size 440 x 380 for testing.

PROPOSED METHOD PART 1

Step 1: Edge Detection

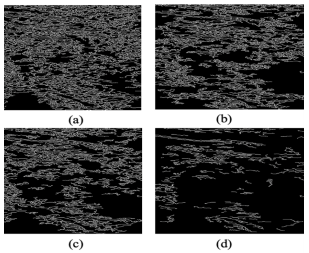
We use Canny edge detector to extract edges based on the difference in intensity. It uses an edge detection operator to obtain the intensity gradients and apply non-maximum suppression to roughly determine edges. After the gradient magnitude and gradient direction are obtained, a double threshold is applied to remove weak potential edges. In order to find an adaptive threshold to detect moderate edges, we experimented on four groups of double thresholds.

Step 2: Noise Addition

In the state of the art, Gaussian noise is uniformly added to the original images. However, the random noise added on the edges can mislead a neural network into mapping the edges as residuals. This situation is common when we train the network with high noise levels. We use edge information to guide the edge addition. we generate random Gaussian noise and re-position noise that appears on the edges to obtain the edge-guided noise. Noise addition step is only involved in the training and validation steps. Given the unknown noise level in ultrasound images, we generate noise within a moderate noise range instead of a fixed noise level.

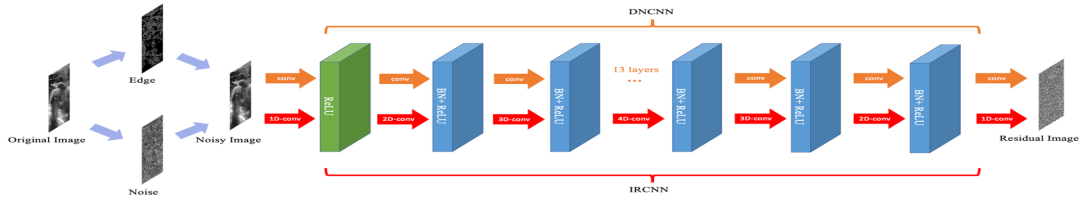
RESULTS OF THE PROPOSED ARCHITECTURE WITH DIFFERENT THRESHOLDS.

Methods	Thresholds	PSNR	SSIM
Edge-guided DnCNN	(50,150)	34.1772	0.9166
	(10,240)	**34.9980**	**0.9221**
	(50,220)	34.2571	0.9160
	(100,270)	34.4400	0.9181
Edge-guided IRCNN	(50,150)	36.2648	0.9369
	(10,240)	35.4938	0.9257
	(50,220)	35.7030	0.9305
	(100,270)	**37.0241**	**0.9433**



Edge detection using original Canny with threshold values: (a) T1 =50, T2 =150, (b) T1 = 10, T2 =240, (c) T1 = 50, T2 =220, and (d) T1 = 100,T2 = 270

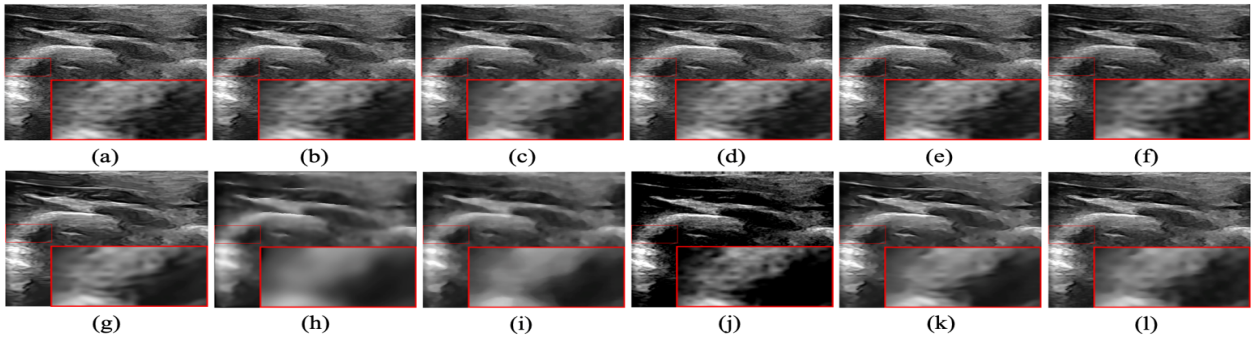
PROPOSED METHOD PART 2



Step 3: Combination with DnCNN & IRCNN

- We take the noise-added patches as the input and use the original patches as the ground truth to provide references for residual learning.
- DnCNN model is modified from the VGG architecture with pooling layers removed. Utilizing the convolutional filters, the size of the receptive field can be increased to (2d+1)×(2d+1), which can guarantee a larger image region for capturing context information. Utilizing batch normalization, the training speed can be boosted, and the final accuracy can be improved. To make the DnCNN model work for ultrasound images with a blind noise level, the depth of the neural network is set to 20 so that the size of the receptive field can be increased to 41 x 41. Initially, Conv + ReLU is taken as the first layer of the DnCNN model, where 64 feature maps are generated by 64 filters with size 3 x 3 x 1. Following Conv, rectified linear units (ReLU) are involved for non-linearity. From depth 2 to 19, DnCNN takes Conv + BN + ReLU as the layers, which involves 64 filters with size 3 x 3 x 64. Finally, the DnCNN model takes Conv as the last layer, where the output is reconstructed by a 3 x 3 x 64 filter.
- IRCNN contains seven dilated convolutional layers, which can enlarge the receptive field. The first layer is a Dilated Convolution (the dilated convolution is 3×3, and the dilation factor is 1) + ReLU. The second to sixth layers are Dilated Convolution (the dilated convolutions are 3×3, and the dilation factors are 2, 3, 4, 3, and 2) + BN + ReLU. The last layer is a Dilated Convolution with a dilation factor of 1. The number of feature maps for each middle layer is 64. Batch normalization and residual learning are used to accelerate training, and smaller-size training images are used to solve the problem caused by boundary artifacts.

EVALUATION & COMPARISON



Denoising results for different methods on sample image: (a) Original image (b) Edge-guided DnCNN (c) DNCNN (d) Edge-guided IRCNN (e) IRCNN (f) Noise2Self (g) FFDNet (h) Anisotropic Diffusion Filter (i) Median Filter (j) Wavelet Filter (k) Non-local Means Filter (l) Bilateral Filter.

PSNR AND SSIM RESULTS OF DIFFERENT METHODS ON TESTING SET.

Methods	E-DnCNN (Blind)	DnCNN (Blind)	E-IRCNN (Blind)	IRCNN (Blind)	Noise2Self (Blind)	FFDNet (Sig15)	Anisotropic Filter	Median Filter	Wavelet Filter	NLM Filter	Bilateral Filter
PSNR	**34.9980**	33.5115	**37.0241**	33.3894	29.4837	34.1458	31.3950	32.3649	28.2906	33.8018	34.6056
SSIM	**0.9221**	0.9093	**0.9433**	0.9067	0.9086	0.8434	0.6702	0.6955	0.1973	0.8183	0.8674

SUBJECTIVE USER EVALUATION RESULTS OF DIFFERENT METHODS ON TESTING SET.

Methods	E-DnCNN (Blind)	DnCNN (Blind)	E-IRCNN (Blind)	IRCNN (Blind)	Noise2Self (Blind)	FFDNet (Sig15)	Anisotropic Filter	Median Filter	Wavelet Filter	NLM Filter	Bilateral Filter
Overall	**253**	206	**250**	211	192	150	41	39	83	92	133
Average	**10.12**	8.24	**10.00**	8.44	7.68	6.00	1.64	1.56	3.32	3.68	5.32