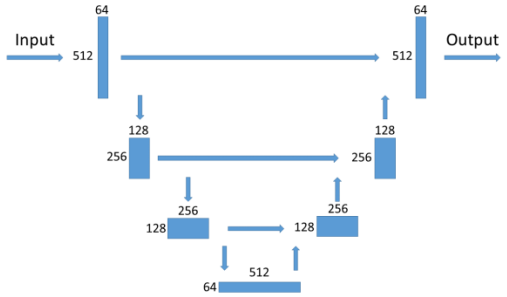
Multi-scale Processing of Noisy Images using Edge Preservation Losses

Nati Ofir and Yosi Keller, BIU

U-Net Architecture



Faint Edge Detection

Classic vs. Deep:

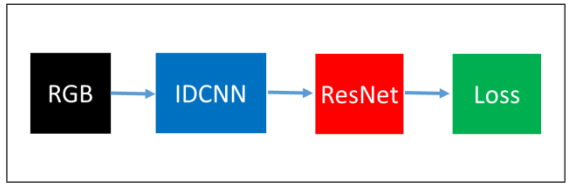
(a) (b) (c)

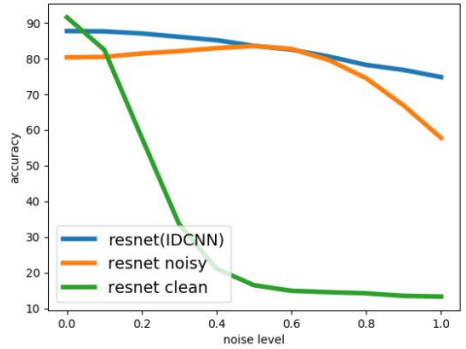
Fig. 1. Example of a medical image with many curved edges. (a) The original image. (b) The proposed FED-CNN approach results. (c) FastEdges [22] results. Both methods achieve high quality of detection while ours run in milliseconds and FastEdges runtime is more than seconds.

Dice Coefficients Loss:

$$Di(y, y') = -\frac{\sum_p y'(p) \cdot y(p)}{\sum_p y'(p) + \sum_p y(p)}$$

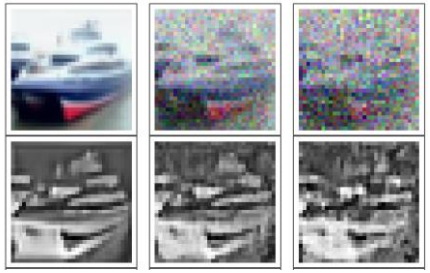
Noisy Image Classification





Algorithm	CIFAR10	CIFAR100
$resnet(IDCNN)$	**82.7**	**53.3**
$resnet_{noisy}$	77.5	46.0
$resnet_c$	34.1	16.9

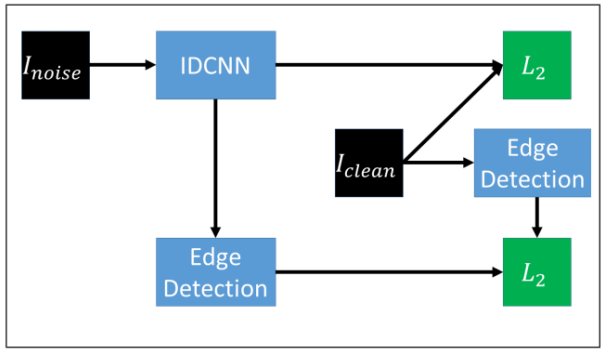
IDCNN Heat-Maps:



Natural Image Denoising

(a) (b) (c)

Fig. 2. Denoising result at additive noise of 50 standard deviation, of the proposed multi-scale network trained by our edge preservation loss. (a) The noisy input image. (b) The results of the proposed scheme. (c) Denoising results of the state-of-the-art DnCNN [34] approach. Our method achieves the highest SSIM [31] scores in our experiments at all the noise levels.

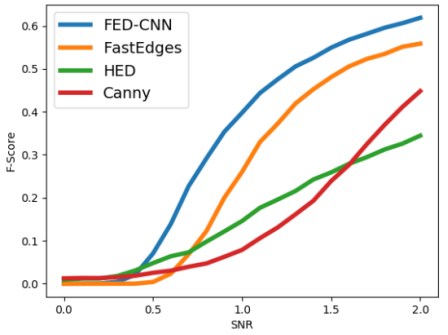


Edge Preservation Loss:

$$L_E = ||\frac{\partial}{\partial x} I_c - \frac{\partial}{\partial x} IDCNN(I_n)||_2^2$$

Algorithm	$\sigma = 15$	$\sigma = 25$	$\sigma = 50$
IDCNN-E	31.00/**0.9**	28.86/**0.85**	25.95/**0.75**
IDCNN	30.80/0.89	28.73/0.84	25.93/**0.75**
DnCNN	31.74/**0.9**	29.89/**0.85**	25.69/0.71
BM3D	31.07/0.88	28.26/0.81	24.57/0.67

FED-CNN Results



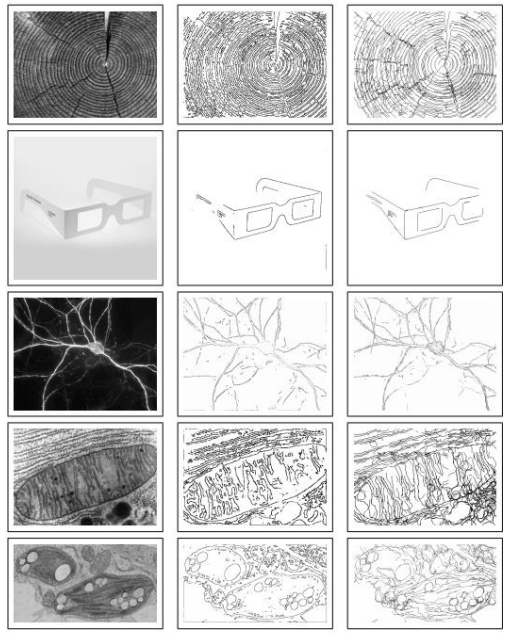


Fig. 9. Examples of real images. Left: the original gray scale images. Middle: our results. Right: FastEdges [22] results. Both methods achieve high quality of detections.