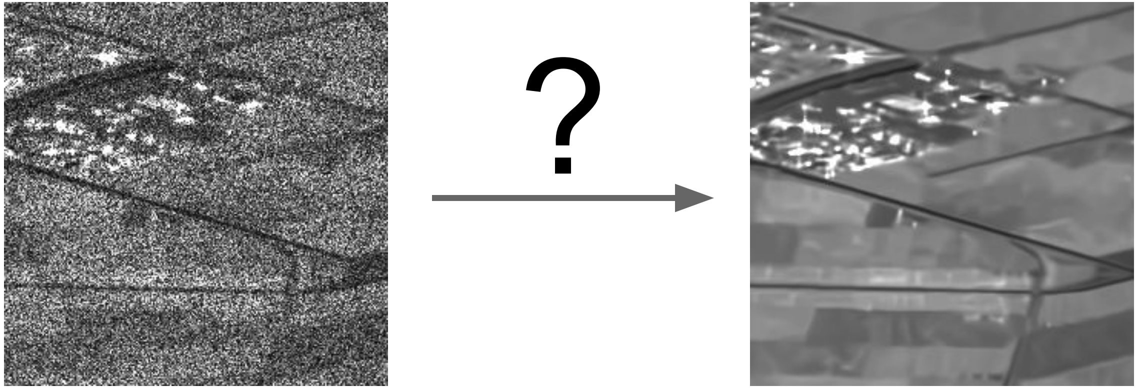
FFDNet for SAR despeckling



Björn Michele, Lucas Elbert

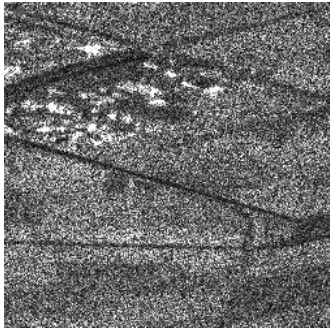
Supervisors: Florence Tupin, Emanuele Dalsasso

Roadmap

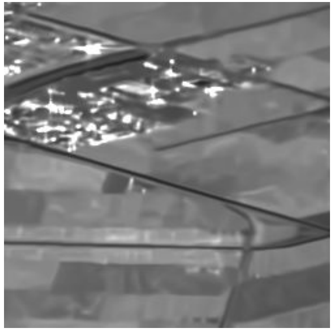
- SAR images & Speckle noise
- FFDNet
- Training approaches
- Noise2Noise
- Comparison with other methods
- Conclusion & Further Work

Sar images & Speckle noise

- SAR measures:
  - Intensity $I = |z|^2$, Amplitude $A = |z|$
- Pixel = Cell with many scatterers
- Backscattered waves interfere
  - deterministic but no knowledge
  - Noise → random variable
- Goodman model:
  - Reflectivity $R = E[I]$
  - $p(I|R)$ exponential distribution
- I, A multiplicative noise
  - $I = R*S$
- log(I), log(A) additive noise
  - $\log(I) = \log(R)+\log(S)$



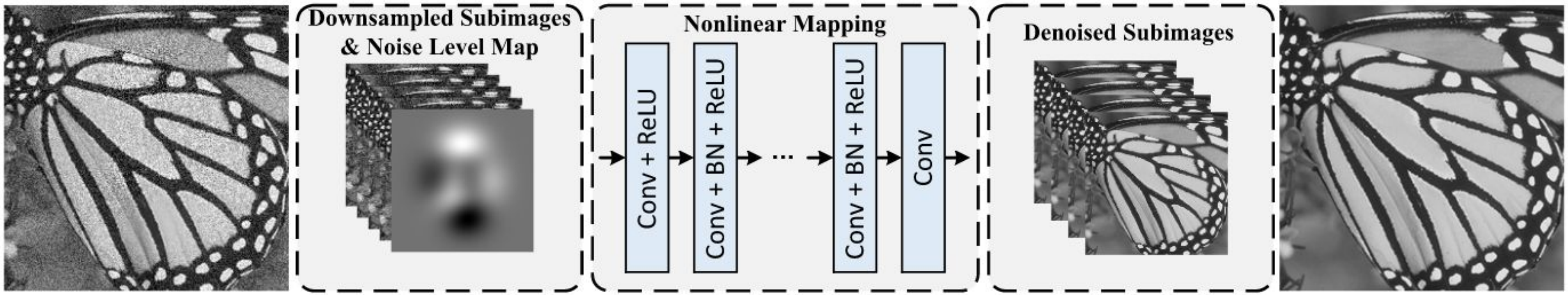
“Noisy” version



Ground truth $R = E[I]$

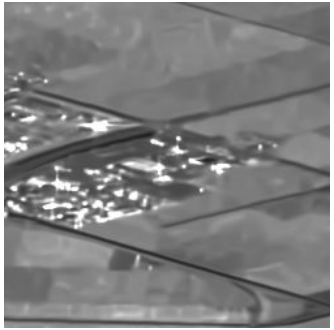
FFDNet

- Neural Network approach to denoise the images
- Designed for the removal of spatial variant additive white Gaussian noise in optical images
- Two main ideas:
  - Usage of several downsampled subimages to reduce learnable parameter
  - Per pixel noise level map (Std. map))as additional input → allows to denoise images with different noise level
- Begin: Lossless downsampling, End: Lossless upsampling

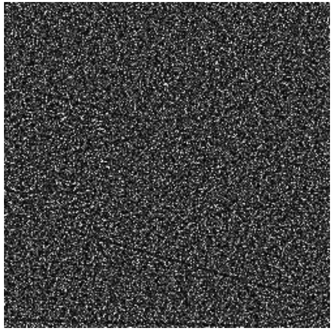


Data generation

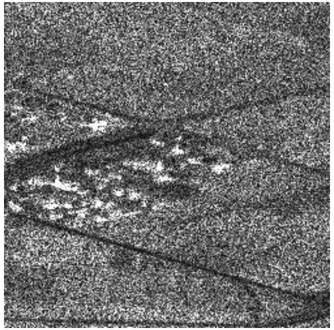
- We train on real SAR images with synthetically added speckle noise
- Basis are dennoised one-look amplitude images
- Use the Goodman model to generate artificial speckle noise



x



=



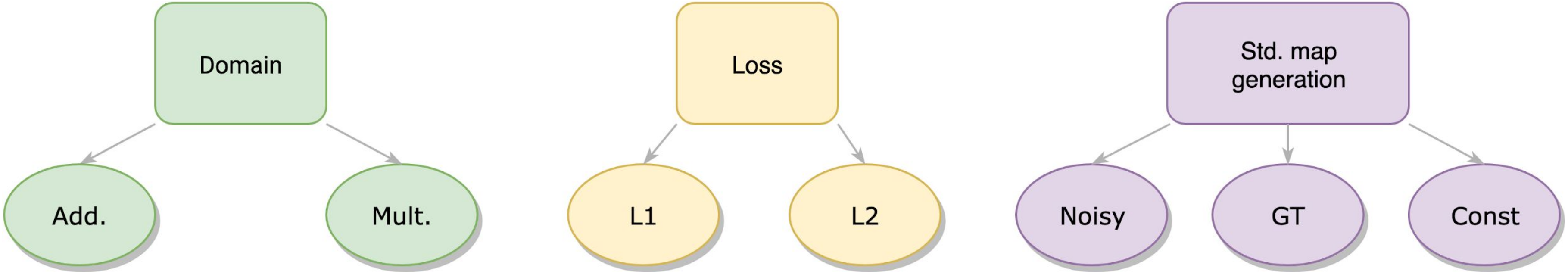
Ground Truth

Artificial speckle noise

Noisy image

Training approaches

- We tested several approaches:



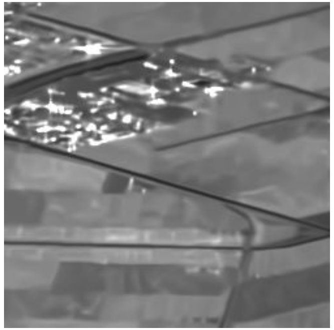
- FFDNet is originally trained with the L2-loss, we also tried out the L1 loss as we expected sharper edges with that loss
- Loss function: $L = || X - FFNet(Y) ||_{L1/L2}$, X is the denoised GT image, Y is the noisy image

Evaluation metrics

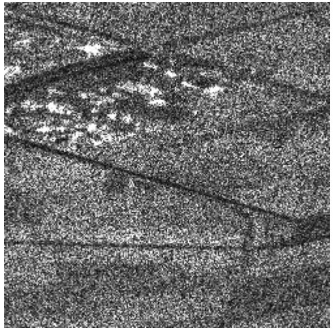
- Peak-signal-to-noise-ratio (Quality)

$$\mathrm{PSNR} = 10\log_{10}\left(\frac{MAX_R^2}{MSE}\right)$$

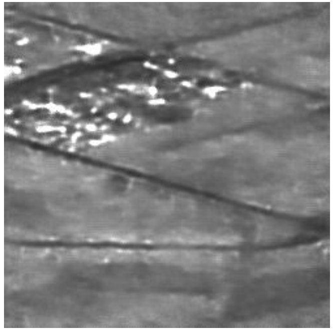
  - The higher, the better
  - Not always conform with human perception
- Ratio (Model consistency)
  - Multiplicative noise model
  - S=I/R, noise = input / predicted reflectivity
  - E[S] = 1, S exponentially distributed
- Manually inspect images (qualitative evaluation)



Ground truth



PSNR 11.9



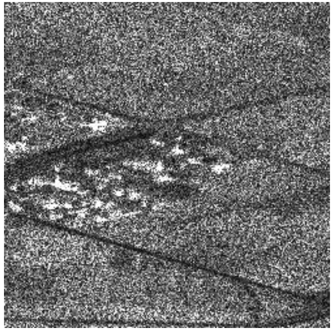
PSNR 22.3

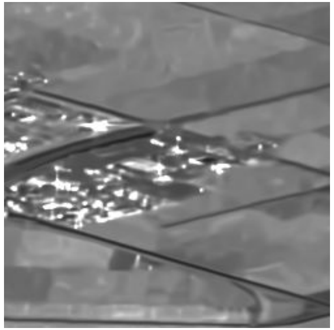
Quantitative results

Method	Std. Map	Loss	PSNR (denoised)	PSNR (noisy)	Ratio (mean)
Multip.	Std. (Train: Noisy, Test: Noisy), Ws: 7	L1	**24.26**	12	0.99
	Std. (Train: Noisy, Test: Noisy), Ws: 7	L2	23.44	12	0.92
Add.	Std: const.	L1	21.85	12	1.03
	Std: const.	L2	22.97	12	0.98

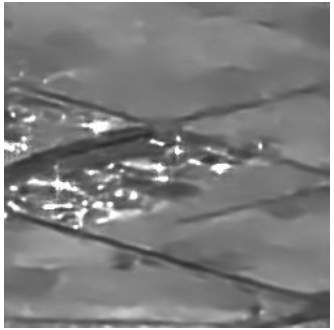
- All approaches are able to improve the noisy input
- **Best results** are obtained surprisingly with the **multiplicative** noise
- Mean and Std. of the ratio are close to theoretical values

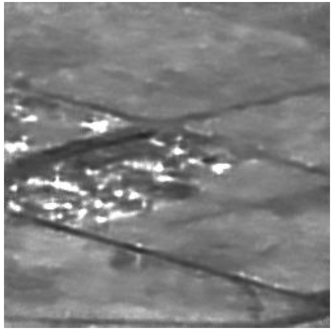
Qualitative results - Denoised image

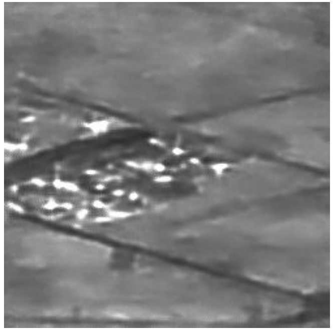

Input (PSNR: 12.57)


Target

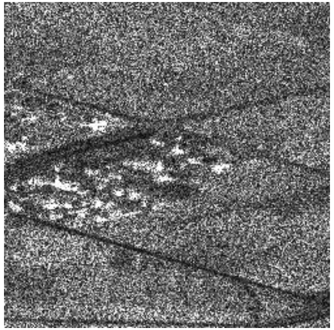

Mult. L2 (PSNR: 21.68)


Mult. L1 (PSNR: 22.17)

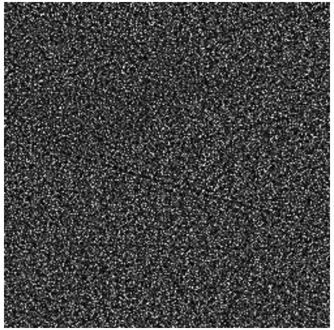

Add. L2 (PSNR: 19.47)


Add. L1 (PSNR: 18.25)

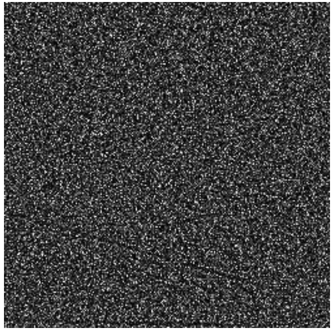
Qualitative results - Ratio



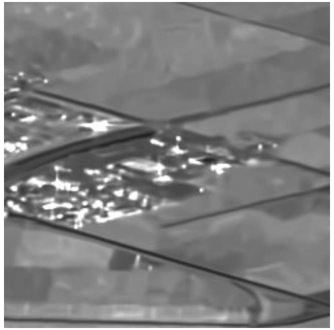
Input



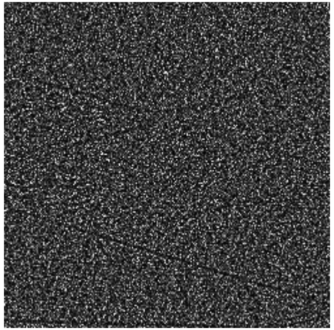
Mult. L2 (μ: 0.927, σ: 0.83)



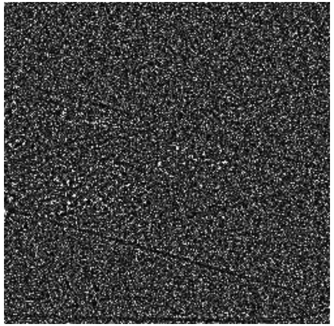
Mult. L1 (μ: 0.995 , σ: 0.95)



Target

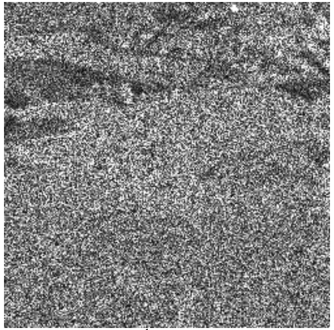


Add. L2 (μ: 0.97, σ: 1.00)

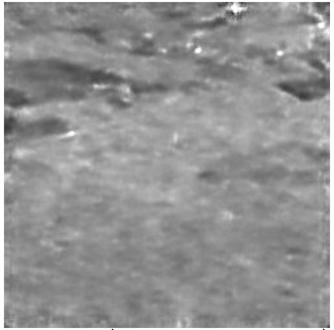


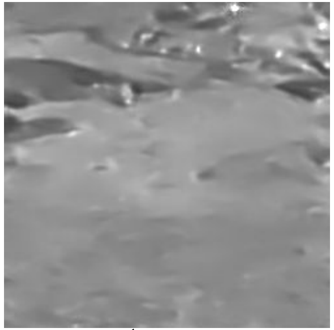
Add. L1 (μ: 1.04, σ: 1.176)

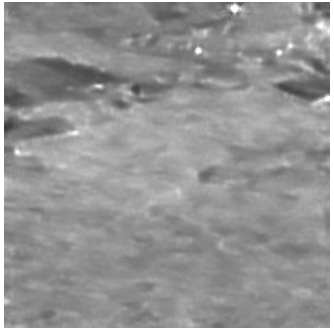
Qualitative results - Denoised image

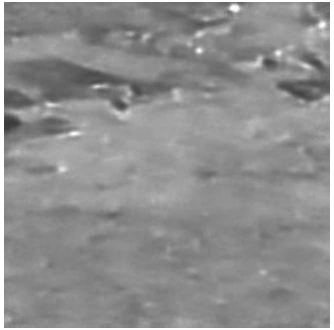

Input (PSNR: 8.98)


Target


Mult. L2 (PSNR: 22.78)

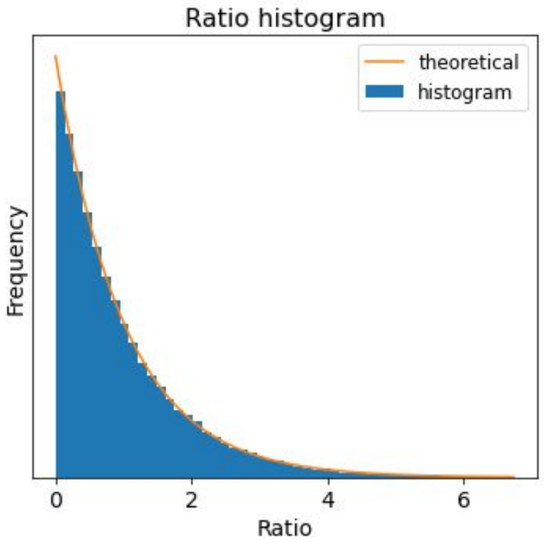

Mult. L1 (PSNR: 24.36)


Add. L2 (PSNR: 22.39)


Add. L1 (PSNR: 22.21)

Qualitative results

- L1 loss seems to generate sharper edges and more homogenous parts
- Removal of the (artificial) speckle, but looks too blurred
- The ratio distribution confirm with theory
- PSNR values are highly dependant on GT images



Noise2Noise

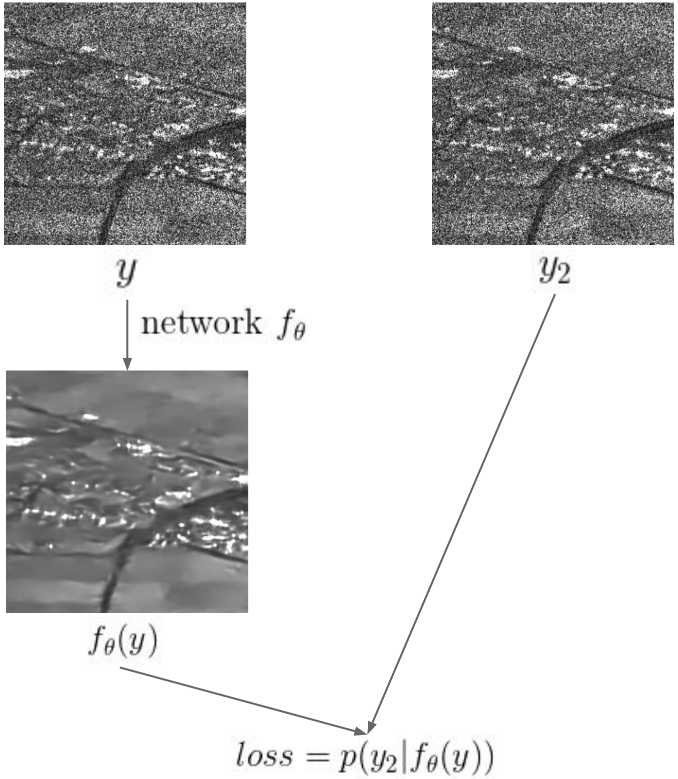
- How to learn without ground truth?
- Replace ground truth by another noisy image
- x: ground truth, y, y2: noisy versions
- “standard” network:

$$\underset{\theta}{argmin} E_y[E_{x|y}[L(f_\theta(y), x)]]$$

- Noise2Noise network:

$$\underset{\theta}{argmin} E_y[E_{y_2|y}[\tilde{L}(f_\theta(y), y_2)]]$$

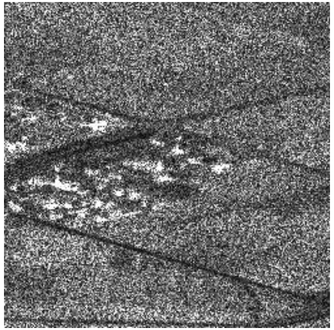
- Does the optimal network change?
- Need to ensure $f_{\theta^*}(y) = x$
- $L_2 - loss$, Fisher-Tippet-loss



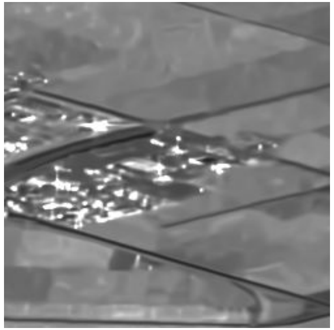
Noise2Noise – Results

	Method	Std-map	Loss	PSNR (denoised)	PSNR (noisy)	Ratio (mean)
	Mult.	Std. (Train: Noisy, Test: Noisy), Ws: 7	L1	**24.26**	12	0.99
Noise2Noise	Mult.	Std. (Train: Noisy, Test: Noisy), Ws: 7	L2	22.76	12	1.04
	Add.	Std: const.	L2	22.77	12	0.98
		Std: const.	Fisher-Tippet	**24.61**	12	0.96

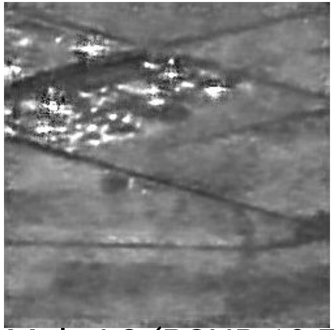
Noise2Noise - Qualitative



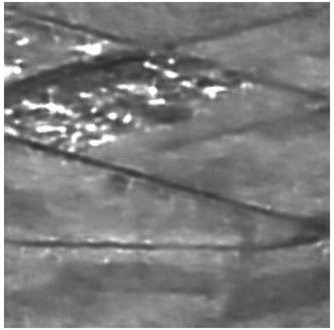
Input (PSNR: 12.57)



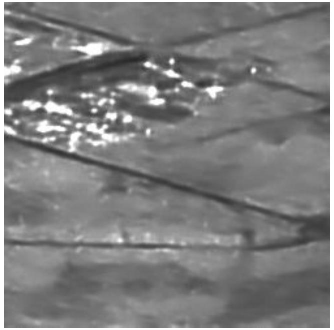
Target



Mult. L2 (PSNR 19.7)



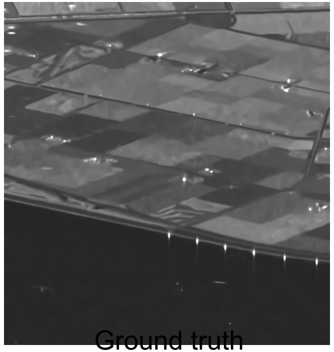
Add. L2 (PSNR 21.85)



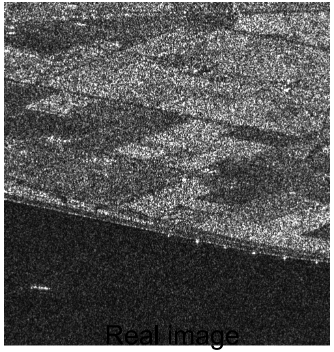
Add. Fisher-Tippet (PSNR 24.57)

Real images

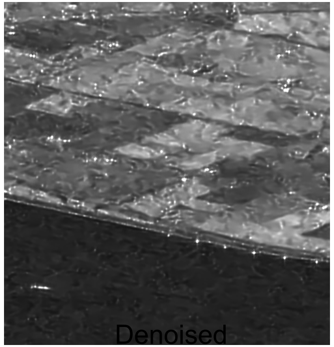
- Goodman model assumes pixel independence
- In real images pixels are correlated
- Denoising produces shimmer
- Fix: Downsampling
  - Better but not perfect
- → Train on real images?



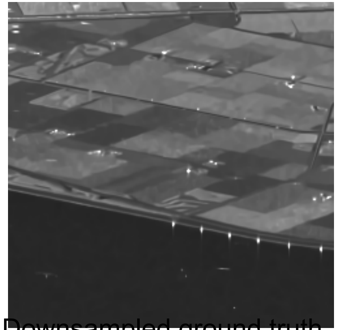
Ground truth



Real image
(PSNR 9.6)

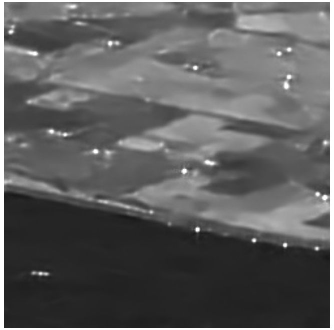


Denoised
(PSNR 16.9)



Downsampled ground truth

Downsampled real image
(PSNR 9.6)



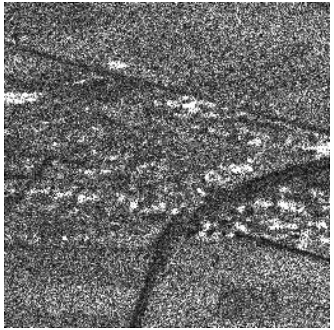
Denoised downsampled
(PSNR 17.7)

Comparison with other methods

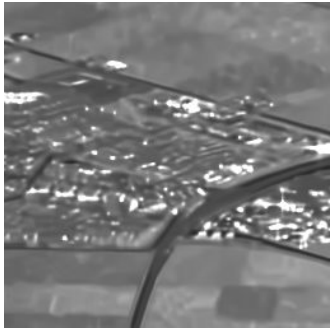
- Other deep-learning methods: SAR-CNN, U-Net (Noise2Noise)
- Evaluation on 140 images (7 different image crops with each 20 noise realizations), artificial speckle

Method	PSNR (denoised)	PSNR(noisy)	Ratio (mean)
SAR-CNN	**25.59**	12.19	0.988
U-Net (Noise2Noise)	25.13	12.19	0.998
Mul. L1 (FFDNet)	24.94	12.19	0.99
Noise2Noise Add. Fisher-Tippet (FFDNet)	22.25	12.19	0.956

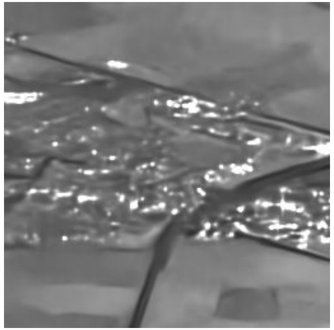
Comparison with other methods - Qualitatively



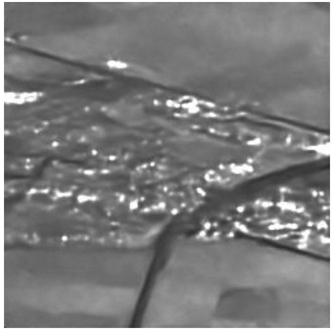
Input



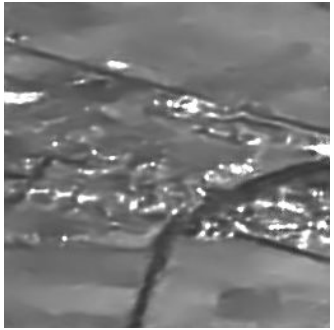
Ground Truth



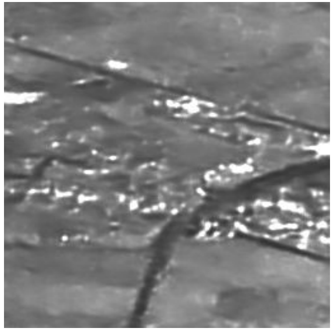
SAR-CNN



U-Net (Noise2Noise)



FFDNet Mul. L1



FFDNet Add. (Noise2Noise) Fisher-Tippet

Take home messages

- FFDNet can be adapted to denoise SAR images
- Supervised training : best results with multiplicative approach
- L1-loss produces sharper results
- Unsupervised training : Noise2Noise
- Artificial noise can be used for developing the method, large losses in our case for the transfer to real data → but methods also denoises real speckle noise

Further work

- Training of the denoise network with real noise data → especially the Noise2Noise approach could be interesting, as no denoised GT is necessary
- Iterative computation: Calculation of the noise maps on already denoised images
- How are the improvements on multi-look images ? Cross-training between single look and multi-look images helpful ?

[1] Zhang, K., Zuo, W., & Zhang, L. (2018). FFDNet: Toward a fast and flexible solution for CNN-based image denoising. *IEEE Transactions on Image Processing*, *27*(9), 4608-4622.
[2] Lehtinen, J., Munkberg, J., Hasselgren, J., Laine, S., Karras, T., Aittala, M., & Aila, T. (2018). Noise2noise: Learning image restoration without clean data. *arXiv preprint arXiv:1803.04189*
[3] Deledalle, C. A., Denis, L., Tabti, S., & Tupin, F. (2017). MuLoG, or how to apply Gaussian denoisers to multi-channel SAR speckle reduction?. *IEEE Transactions on Image Processing*, *26*(9), 4389-4403.

Thanks for your attention