Denoise Microscopic Data With Deep Learning

Qurat Ul Ain Naqvi Sonali Godavarthy Zakriya Shakeel
Universität Siegen

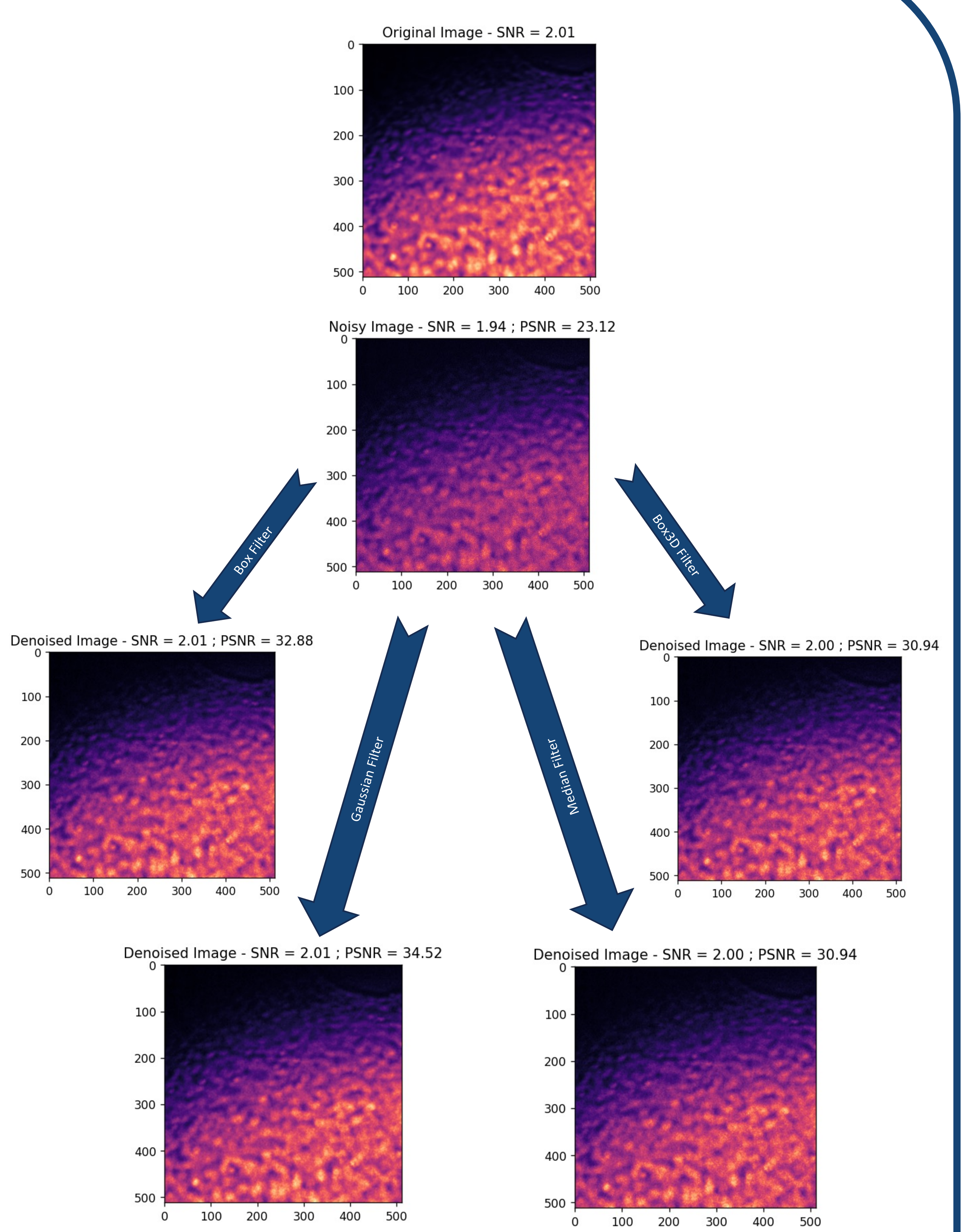
Introduction and Motivation

- **Denoising** is the process of removing or reducing noise from a image.
- **Sources of Noise**: Poor lightening conditions, Vibration, Motion, etc.
- **Types of noise**: Gaussian, Poisson, Salt-and-Pepper Noise
- **AIM:** To find best approach for denoising Microscopic images
- We applied 4 Non-DL Filters namely Box Filter, Gaussian Filter, Median Filter and Box3D Filter.
- We also applied 'Improving Blind Spot Denoising for Microscopy' DL research paper method.
- To compare the feasibility of denoising with various DL, Non-DL approaches

Non-DL Approach

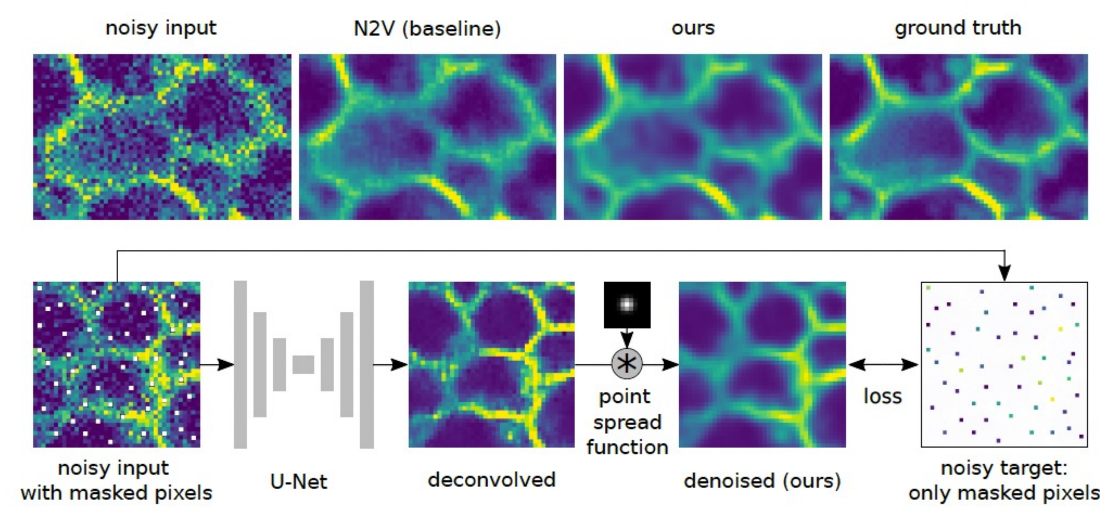
- Added Gaussian noise to image and applied various denoising filters like:
  - Box Filter
  - Gaussian Filter
  - Median Filter
  - Box3DFilter
- And compared them with SNR, PSNR values

Non-DL Results



DL Approach

- **Research Paper**: Improving Blind Spot Denoising for Microscopy
- Self Supervised denoising Approach
- Unpaired noisy image dataset
- Maximize the Signal to Noise Ratio and minimize the undesired effects
- Outputs a deconvolved result
- Uses baseline as NOISE2VOID



- **Image Formation:**

$$\chi = noisy\ image$$
$$z = Phantom\ Image\ (Deconvolved\ Output)$$
$$s = distorted\ image\ on\ the\ detector\ (signal)$$
$$h = Point\ Spread\ Function\ (PSF)$$
$$x \sim P_{NM}(x|s) = noise\ model$$
$$P_{NM}(x_i|s_i) = probability\ distribuion$$

$$result\ of\ convolution, s = z * h$$
$$P_{NM}(x|s) = \prod_i^N P_{NM}(x_i|s_i)$$

- **Denoising Task**:

$$estimate\ \tilde{s}\ \approx s$$
$$estimate\ \tilde{z} \approx z$$

- **Blind Spot Training**: Masking of random pixels in the input image, network tries to. Predict their value from surrounding patch

$$x_i^{RF} = blind\ spot\ receptive\ field$$
$$\theta = network\ parameters$$
$$\hat{s}_i = f(x_i^{RF}; \Theta)$$
$$training\ loss = \sum_i (\hat{s}_i - x_i)^2$$

- **Positivity Constraint:** excited fluorophores can not take negative values, so we add an additional component to loss for not getting any negative values.

$$N = number\ of\ pixels$$
$$\lambda = positivity\ constraint\ hyperparamter$$
$$\frac{1}{|M|}\sum_{i\epsilon M}(\hat{s}_i - x_i)^2 + \lambda\frac{1}{N}\sum_{i=1}^{N}\max(0, -\hat{z}_i)$$

Dataset

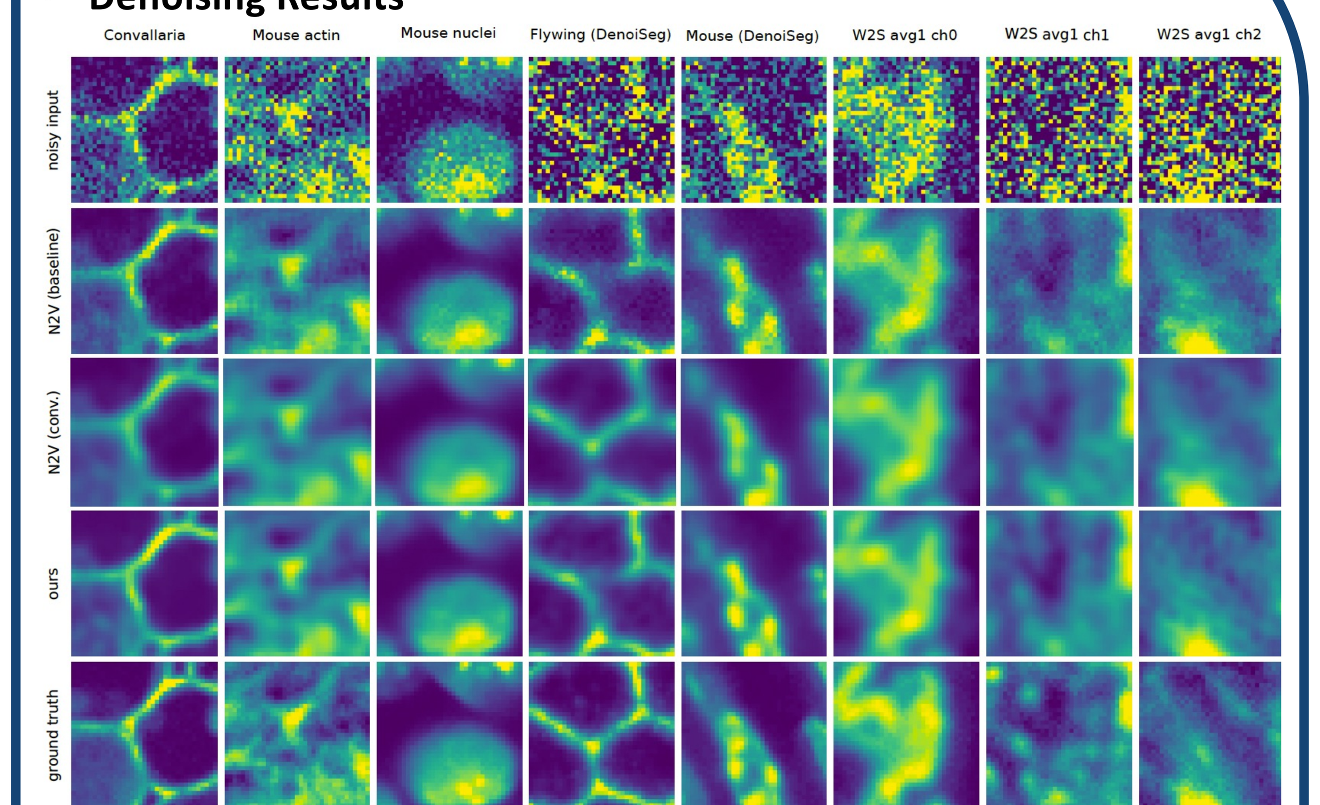
- **LIVECell:** A large-Scale Live Cell dataset
- **Fluorescence Microscopy Data:** *Convallaria, Mouse actin, etc.*
- Text data from the book 'The beetle' was. Used to find standard deviation of Gaussian PSF (σ)

Training

- 3 depth U-Net, 1 input channel
- Epochs = 200
- Initial learning rate = 0.001
- Adam Optimizer
- Batch size = 1
- Positivity constraint, $\lambda = 1$
- Trained for 3 different standard deviation of Gaussian PSF
  - σ = 0.5
  - σ = 1.0
  - σ = 1.5

DL Results and Performance

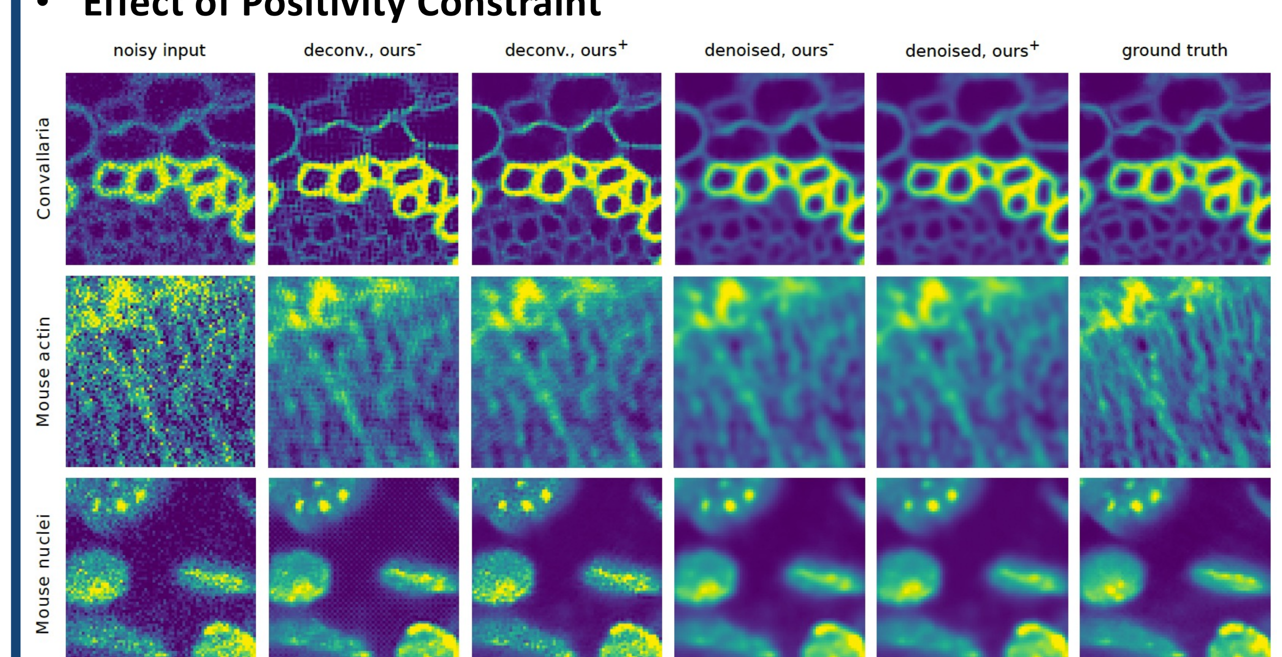
- **Denoising Results**



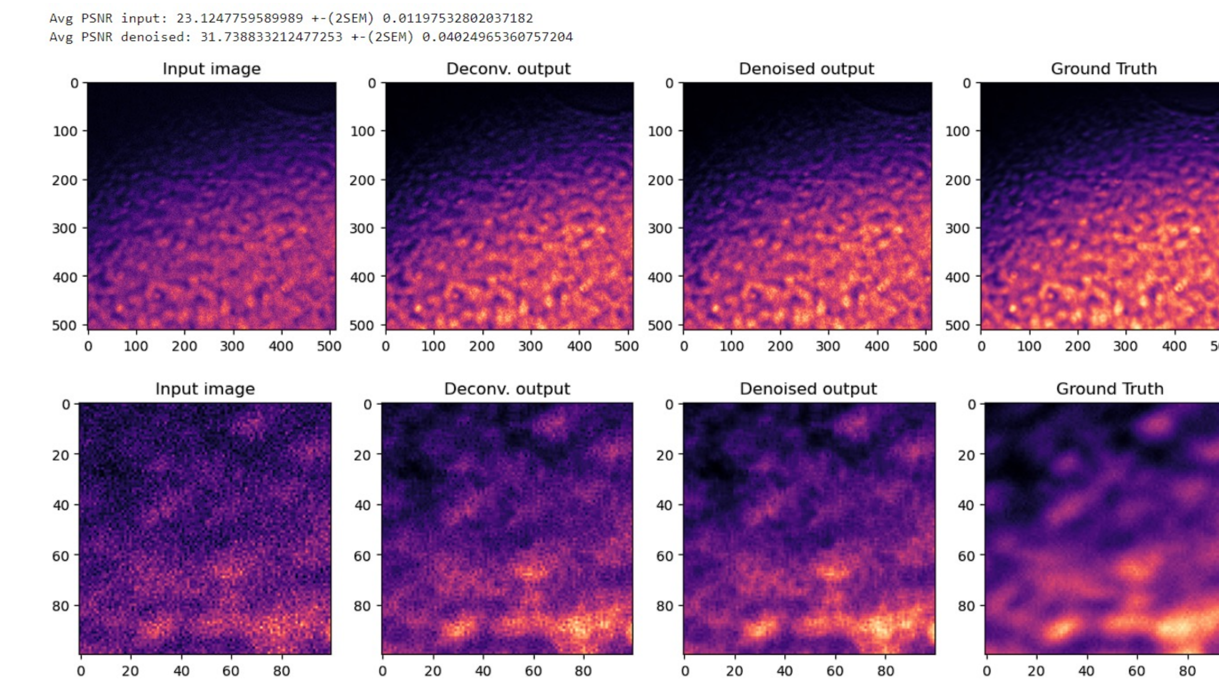
- **Quantitative Results**

dataset/ network	raw data	self-supervised						superv.
		no noise model				noise model		
		N2V	N2V conv.	ours⁻	ours⁺	PN2V	DivN.	CARE
Convallaria	28.98	35.85	32.86	**36.39**	36.26	36.47	36.94	36.71
Mouse actin	23.71	33.35	33.48	33.94	**34.04**	33.86	33.98	34.20
Mouse nuclei	28.10	35.86	34.59	**36.34**	36.27	36.35	36.31	36.58
Flywing (DenoiSeg)	11.15	23.62	23.51	24.10	**24.30**	24.85	25.10	25.60
Mouse (DenoiSeg)	20.84	33.61	32.27	**33.91**	33.83	34.19	34.03	34.63
W2S avg1 ch0	21.86	34.30	34.38	**34.90**	34.24	-	34.13	34.30
W2S avg1 ch1	19.35	31.80	32.23	**32.31**	32.24	-	32.28	32.11
W2S avg1 ch2	20.43	34.65	**35.19**	35.03	35.09	32.48	35.18	34.73
W2S avg16 ch0	33.20	38.80	38.73	**39.17**	37.84	39.19	39.62	41.94
W2S avg16 ch1	31.24	37.81	37.49	**38.33**	38.19	38.24	38.37	39.09
W2S avg16 ch2	32.35	40.19	40.32	40.60	**40.74**	40.49	40.52	40.88

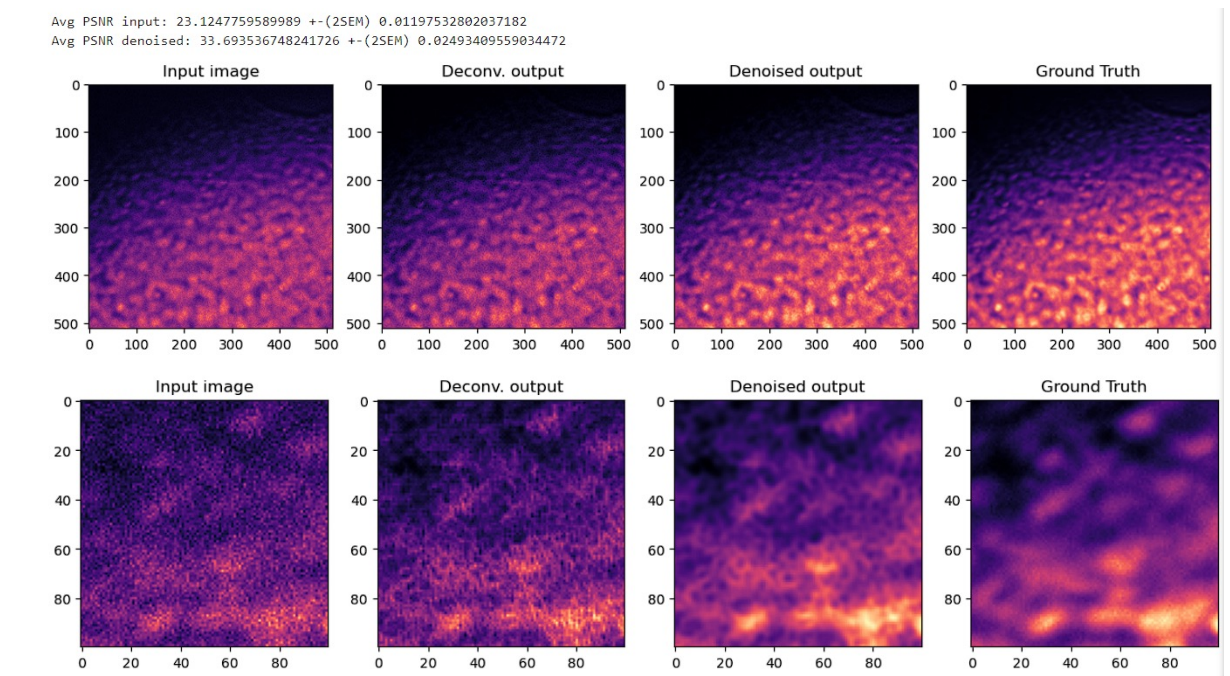
- **Effect of Positivity Constraint**



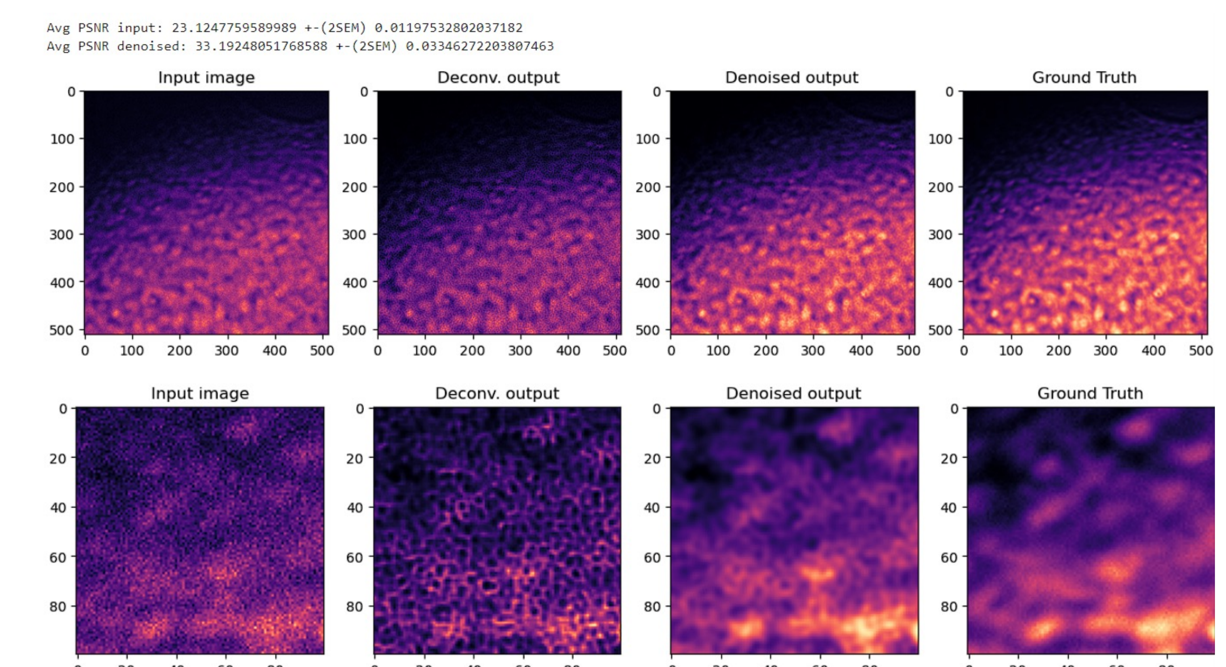
- **With PSF, $\sigma = 0.5$**



- **With PSF, $\sigma = 1.0$**



- **With PSF, $\sigma = 1.5$**



Conclusion

- We used the Non-DL and DL approaches to denoise our dataset.
- We trained the DL model with PSF =0.5, 1.0, 1.5
- DL model used Tiff file with each file containing around 100 images
- We found out the following results:
  - The best Non-DL denoising approach is Gaussian filter (PSNR=34.52)
  - The best DL denoising is with PSF=1.0 (PSNR=33.69)
  - We conclude that Gaussian filter gives the best denoised image for our dataset