Learning Discriminative Shrinkage Deep Networks for Image Deconvolution

Pin-Hung Kuo[1] Jinshan Pan[2] Shao-Yi Chien[1] Ming-Hsuan Yang[3,4,5]

[1]National Taiwan University [2]Nanjing University of Science and Technology

[3]Google Research [4]University of California, Merced [5]Yonsei University

ECCV TEL AVIV 2022

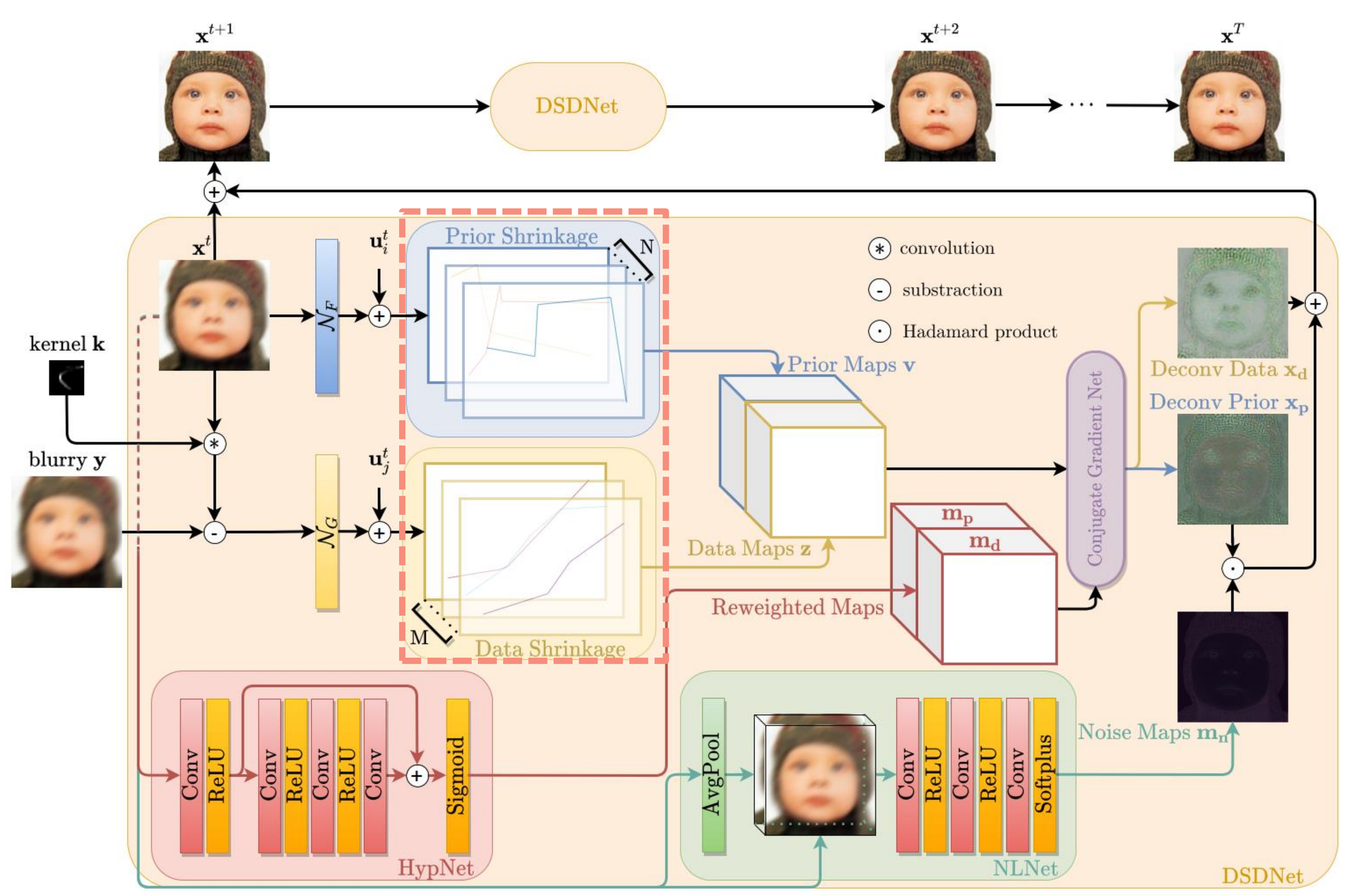
Introduction

Goals: Design a deblurring deep learning model based on Maximum-a-Posteriori estimation.

Challenges:

- How to learn the **shrinkage functions** corresponding to various learned filters?
- With the regularization / data terms, how to solve the **deconvolution problem**?

Architecture



ADMM

We model the problem as:

$$\min_{\mathbf{u},\mathbf{v},\mathbf{x},\mathbf{z}} \sum_{i=1}^{N} R_i(\mathbf{v}_i) + \sum_{j=1+N}^{M+N} R_j(\mathbf{z}_j) \quad s.t. \quad \mathbf{F}_i\mathbf{x} = \mathbf{v}_i, \quad \mathbf{G}_j(\mathbf{y} - \mathbf{H}\mathbf{x}) = \mathbf{z}_j$$

and solve it by ADMM:

$$\mathbf{v}_i^{t+1} = \mathbf{prox}_{\lambda_i R_i}(\mathbf{F}_i\mathbf{x}^t + \mathbf{u}_i^t),$$

$$\mathbf{z}_j^{t+1} = \mathbf{prox}_{\lambda_j R_j}(\mathbf{G}_j(\mathbf{y} - \mathbf{H}\mathbf{x}^t) + \mathbf{u}_j^t),$$

$$\left(\sum_{i=1}^{N} \rho_i \mathbf{F}_i^\top \mathbf{F}_i + \sum_{j=N+1}^{N+M} \rho_j \mathbf{H}^\top \mathbf{G}_j^\top \mathbf{G}_j \mathbf{H}\right) \mathbf{x}^{t+1}$$

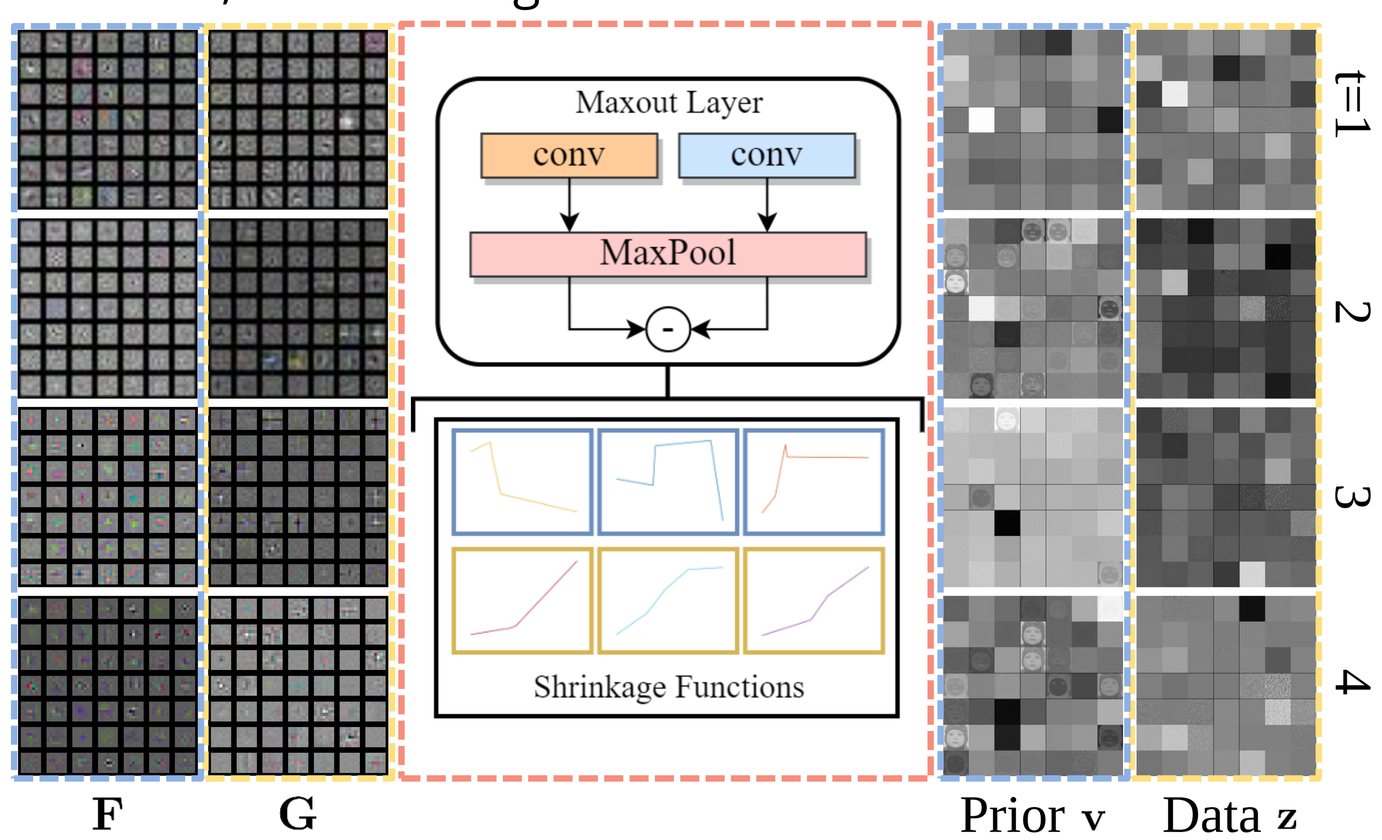
$$= \left(\sum_{i=1}^{N} \rho_i \mathbf{F}_i^\top (\mathbf{v}_i^{t+1} - \mathbf{u}_i^t) + \sum_{j=N+1}^{N+M} \rho_j \mathbf{H}^\top \mathbf{G}_j^\top (\mathbf{G}_j\mathbf{y} - \mathbf{z}_j^{t+1} + \mathbf{u}_j^t)\right)$$

$$\mathbf{u}_i^{t+1} = \mathbf{u}_i^t + \mathbf{F}_i\mathbf{x}^{t+1} - \mathbf{v}_i^{t+1},$$

$$\mathbf{u}_j^{t+1} = \mathbf{u}_j^t + \mathbf{G}_j(\mathbf{y} - \mathbf{H}\mathbf{x}^{t+1}) - \mathbf{z}_j^{t+1}.$$

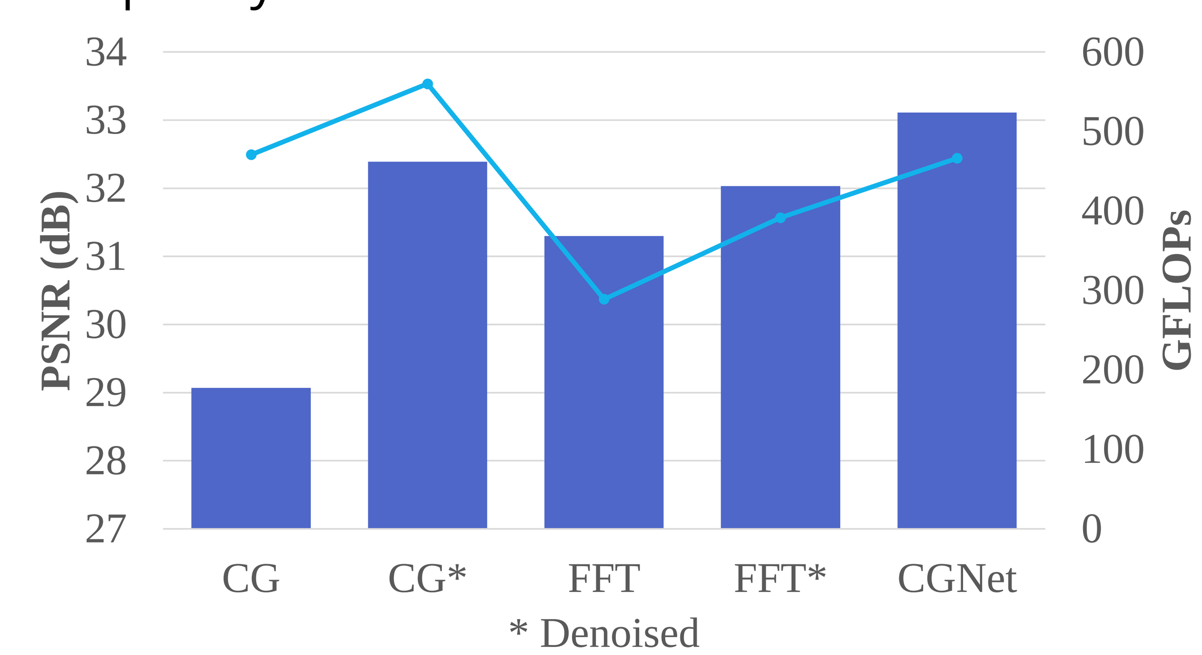
Discriminative Shrinkage Function

As Maxout Layers can linearly approximate any function, the shrinkage functions are learned.



CG Net

To solve the deconvolution problem, we design the CGNet to replace conventional CG or FFT. For FFT and CG, we test them w/o and w/ denoising. CGNet achieves the highest performance with reasonable complexity.

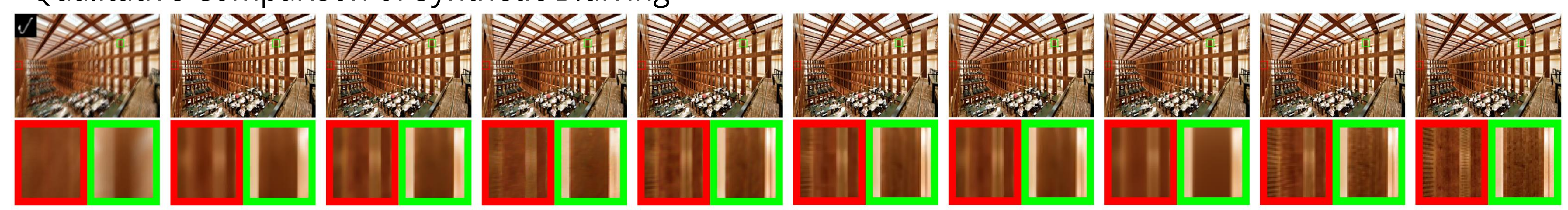


Results

Quantitative Results on Benchmark Datasets

Dataset	noise	IRCNN [73] PSNR / SSIM	SFARL[48] PSNR / SSIM	ADM_UDM [21] PSNR / SSIM	CPCR [12] PSNR / SSIM	KerUNC [41] PSNR / SSIM	VEM [42] PSNR / SSIM	DWDN [10] PSNR / SSIM	SVMAP [11] PSNR / SSIM	DRUNet [72] PSNR / SSIM	DSDNet(Light) PSNR / SSIM	DSDNet(Full) PSNR / SSIM
LEVIN [30]	1%	30.61 / 0.883	25.41 / 0.600	31.48 / 0.922	28.43 / 0.858	32.02 / 0.928	32.05 / 0.927	34.89 / 0.957	35.24 / 0.962	31.94 / 0.922	35.48 / 0.960	**36.62** / **0.965**
	3%	29.70 / 0.864	16.82 / 0.255	28.61 / 0.812	25.61 / 0.765	21.72 / 0.416	29.47 / 0.867	31.94 / 0.916	31.20 / 0.893	30.86 / 0.905	32.13 / 0.918	**32.89** / **0.925**
	5%	28.98 / 0.854	13.07 / 0.157	27.83 / 0.827	23.68 / 0.703	18.25 / 0.272	27.79 / 0.819	30.21 / 0.883	30.12 / 0.876	29.79 / 0.880	30.24 / 0.883	**30.94** / **0.893**
BSD100 [38]	1%	29.20 / 0.817	24.21 / 0.568	29.39 / 0.836	28.77 / 0.829	29.23 / 0.829	29.54 / 0.848	31.10 / 0.881	31.52 / 0.888	30.36 / 0.872	31.50 / 0.892	**32.01** / **0.898**
	3%	27.54 / 0.762	15.80 / 0.245	26.92 / 0.722	25.96 / 0.712	22.10 / 0.430	27.09 / 0.746	28.47 / 0.797	27.94 / 0.762	28.10 / 0.798	28.73 / 0.812	**29.08** / **0.820**
	5%	27.04 / 0.756	12.56 / 0.146	26.04 / 0.697	25.75 / 0.688	18.99 / 0.297	26.11 / 0.698	27.50 / 0.762	27.59 / 0.763	27.19 / 0.767	27.64 / 0.774	**27.96** / **0.782**
SET5 [3]	1%	30.15 / 0.853	26.21 / 0.632	30.52 / 0.868	30.59 / 0.875	30.45 / 0.864	31.00 / 0.875	32.18 / 0.893	32.31 / 0.892	30.84 / 0.881	32.78 / 0.899	**33.43** / **0.905**
	3%	28.66 / 0.813	15.50 / 0.211	27.64 / 0.709	27.94 / 0.799	21.39 / 0.376	28.40 / 0.804	29.54 / 0.838	28.78 / 0.812	29.21 / 0.841	29.94 / 0.843	**30.40** / **0.851**
	5%	27.55 / 0.789	11.91 / 0.122	26.75 / 0.756	26.64 / 0.754	17.74 / 0.241	26.46 / 0.732	28.13 / 0.806	28.02 / 0.793	27.85 / 0.805	28.46 / 0.804	**28.89** / **0.814**

Qualitative Comparison of Synthetic Blurring

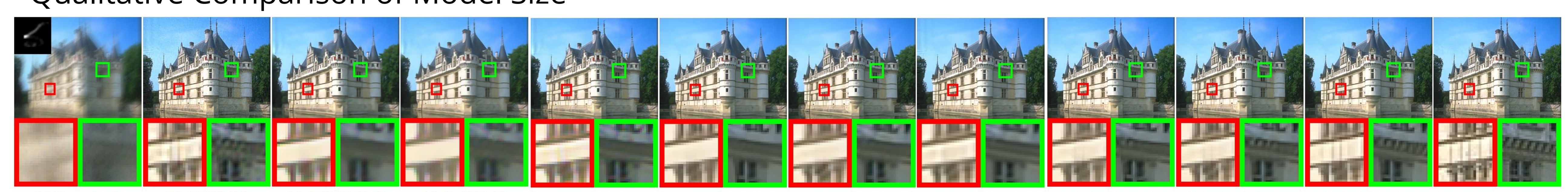


(a) Blurry input (b) IRCNN [73] (c) ADM_UDM [21] (d) KerUNC [41] (e) VEM [42] (f) DWDN [10] (g) SVMAP [11] (h) DRUNet [72] (i) DSDNet (ours) (j) Ground truth

Qualitative Comparison of Real Blurring

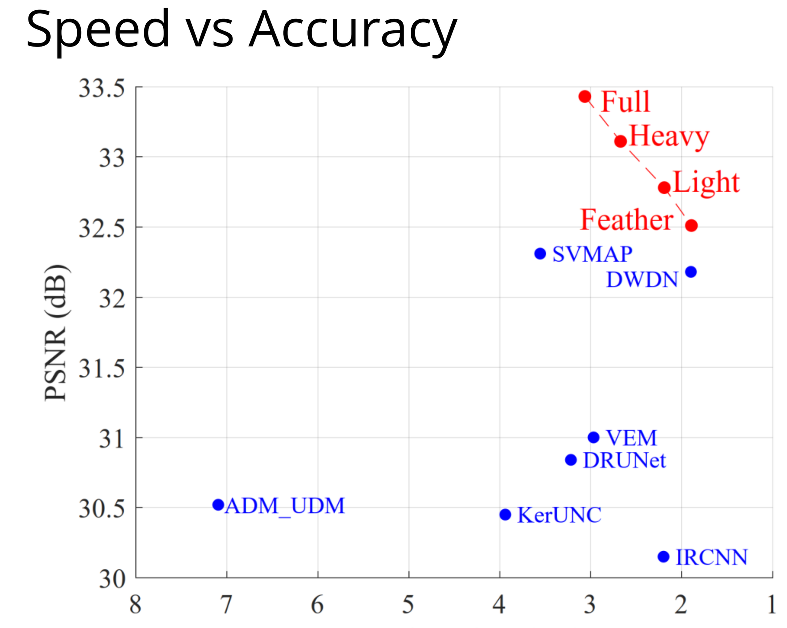
(a) Blurry input (b) IRCNN [73] (c) ADM_UDM [21] (d) KerUNC [41] (e) DWDN [10] (f) SVMAP [11] (g) DRUNet [72] (h) DSDNet (ours)

Qualitative Comparison of Model Size



(a) Blurry input (b) IRCNN [14] (c) ADM_UDM [4] (d) KerUNC [8] (e) VEM [9] (f) DWDN [2] (g) SVMAP [3] (h) DRUNet [13] (i) DSDNet (Light) (j) DSDNet (Heavy) (k) DSDNet (Full) (l) Ground truth

Speed vs Accuracy



Quantitative Results on Real Blurring

	IRCNN [73]	ADM_UDM [21]	KerUNC [41]	VEM [42]	DWDN [10]	SVMAP [11]	DRUNet [72]	DSDNet
BRISQUE	43.484	36.598	37.816	33.663	34.027	35.508	46.774	**33.129**
PIQE	78.700	67.605	65.674	**44.942**	51.348	56.032	81.074	49.788

Specification of 4 Sizes

	Feather	Light	Heavy	Full
T	2	3	3	4
M, N	24	24	49	49

Code and Models

https://github.com/setsunil/DSDNet