Neural Nearest Neighbors Networks

Code available!
github.com/visinf/n3net

Tobias Plötz Stefan Roth

visual inference

TECHNISCHE UNIVERSITÄT DARMSTADT

Motivation

- K-nearest neighbors (KNN) selection is building block for many methods, e.g. non-local image restoration algorithms [1, 3]
- However: KNN selection is not differentiable
- This prevents learning the feature space for matching

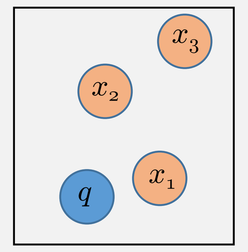
We propose a **deterministic** and **differentiable relaxation to KNN** and leverage it in a **novel non-local processing layer (N³ block)**.

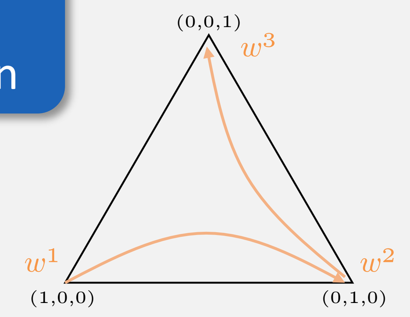
1 Continuous nearest neighbors

Differentiable, deterministic relaxation of KNN selection

Step 1 – K-nearest neighbors selection

q : query item
x_i : database items
X^j : j-th nearest neighbor





w^j define deterministic path on corners of simplex

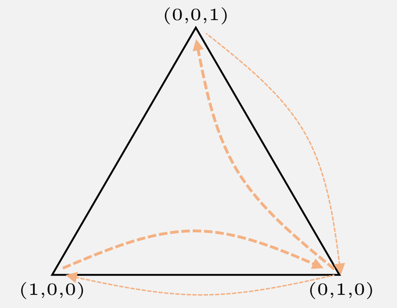
For KNN, X^j given by weighted sum: $X^j \equiv \sum_{i \in I} w_i^j x_i$
w^j deterministic and 1-hot coded

✓ Deterministic
✗ Differentiable

Step 2 – Stochastic k-nearest neighbor selection

Sample w^j from sequence of discrete distributions:

$$\mathbb{P}[w^{j+1} = i \mid \alpha^{j+1}, t] \equiv \mathrm{Cat}(\alpha^{j+1}, t) = \frac{\exp(\alpha_i^{j+1}/t)}{\sum_{i' \in I} \exp(\alpha_{i'}^{j+1}/t)}$$



w^j define stochastic path on corners of simplex

First logits correspond to distances to query item:

$$\alpha_i^1 \equiv -d(q, x_i)$$

Update logits such that same weight cannot be sampled again:

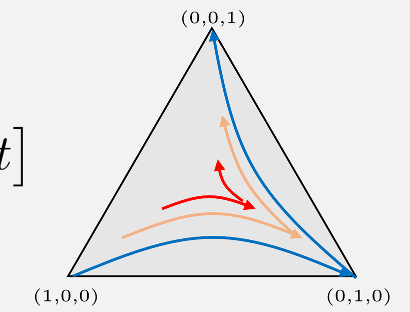
$$\alpha_i^{j+1} \equiv \alpha_i^j + \log(1 - w_i^j) = \begin{cases} \alpha_i^j, & \text{if } w^j \neq i \\ -\infty, & \text{if } w^j = i. \end{cases}$$

✗ Deterministic
😐 Differentiable

Step 3 – Continuous relaxation

Relax discrete w^j to continuous expectation $\bar{w}^j$:

$$\bar{w}_i^{j+1} \equiv \mathbb{E}[w_i^{j+1} \mid \bar{\alpha}^{j+1}, t] = \mathbb{P}[w^{j+1} = i \mid \bar{\alpha}^{j+1}, t]$$



w^j define deterministic path within the simplex

Relax discrete logit update:

$$\bar{\alpha}_i^{j+1} \equiv \bar{\alpha}_i^j + \log(1 - \bar{w}_i^j) \quad \text{with} \quad \bar{\alpha}_i^1 \equiv \alpha_i^1$$

Use $\bar{w}^j$ to define continuous NN: $\bar{X}^j \equiv \sum_{i \in I} \bar{w}_i^j x_i$

Recovers hard KNN as $t \to 0$.

✓ Deterministic
✓ Differentiable

Contributions

- Unlike soft attention mechanisms, e.g. [6], retrieve **multiple weighted averages**, yielding a set of non-local neighbors.
- Go beyond differentiable k-nearest neighbors classifiers [2, 5], retrieving a **set of non-local neighbors** instead of non-local averaging of labels.
- **Do not assume fixed set of target neighbors** unlike [7].
- **Back-propagating** through the non-local layer **enables feature learning** in the context of non-local processing [1, 3].

2 N³ block

Non-local processing layer based on *self-similarity*

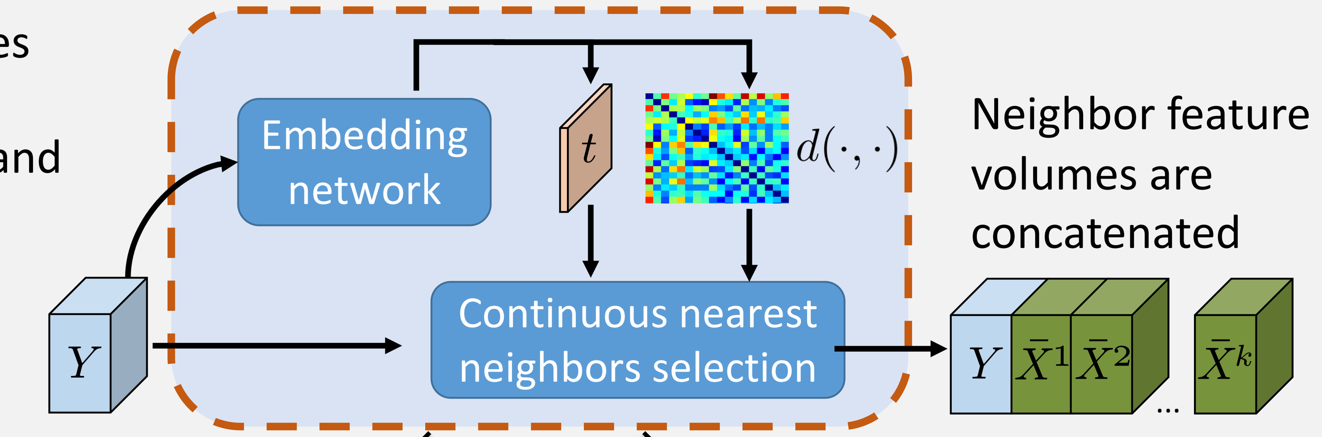
Goal: Calculate nearest neighbor feature volumes

Self-similarity: Queries and database derived from same set of features

Predict teperature t

Full differentiability

Neural nearest neighbor block



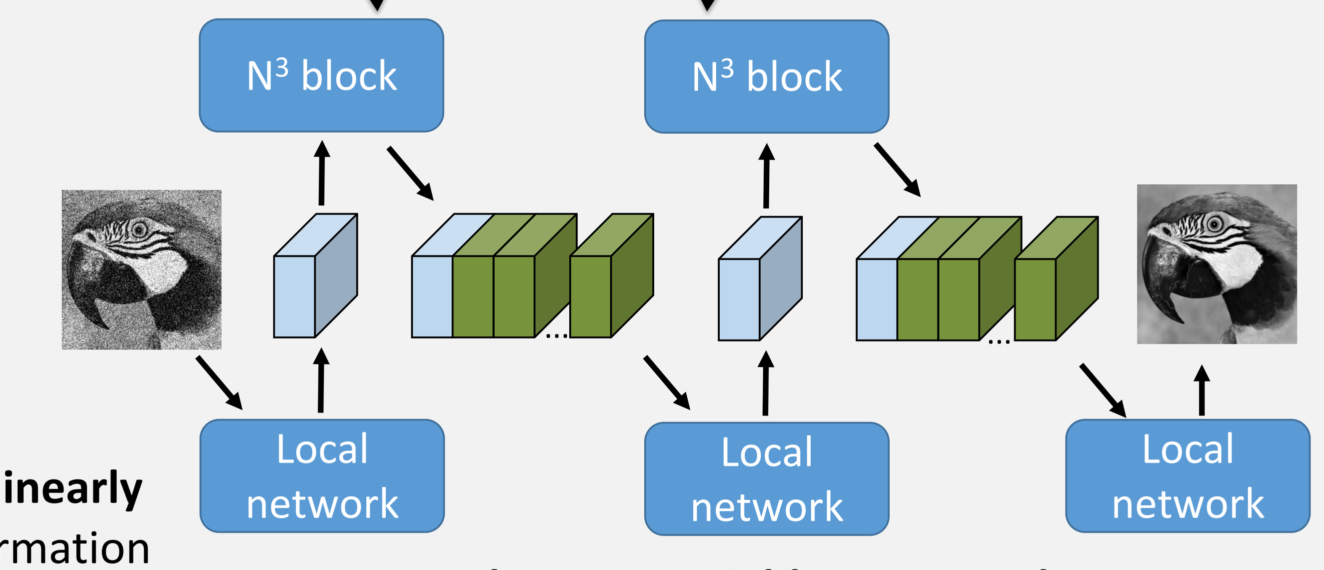
Neighbor feature volumes are concatenated

N³Net

Interleave local network blocks with N³ blocks

Local blocks derived from base architectures, e.g. DnCNN [9]

N³Net can learn to **non-linearly** aggregate neighbor information



Neural nearest neighbors network

3 N³Net: Neural Nearest Neighbors Network

Improved accuracy from augmenting local networks with N³ block

References

[1] K. Dabov, A. Foi, V. Katkovnik, K. Egiazarian. Image denoising with block-matching and 3D filtering. SPIE, 2006.
[2] J. Goldberger, G. E. Hinton, S. T. Roweis, R. R. Salakhutdinov. Neighbourhood components analysis. NIPS*2005.
[3] S. Lefkimmiatis. Universal denoising networks: A novel CNN-based network architecture for image denoising. CVPR, 2018.
[4] T. Plötz, S. Roth. Benchmarking denoising algorithms with real photographs. CVPR, 2017.
[5] O. Vinyals, C. Blundell, T. Lillicrap, K. Kavukcuoglu, D. Wierstra. Matching networks for one shot learning. In NIPS*2016.
[6] X. Wang, R. Girshick, A. Gupta, K. He. Non-local neural networks. CVPR, 2018.
[7] K. Q. Weinberger, L. K. Saul. Distance metric learning for large margin nearest neighbor classification. JMLR, 10:207–244, Feb. 2009.
[8] K. Yi, E. Trulls, Y. Ono, V. Lepetit, M. Salzmann, P. Fua. Learning to find good correspondences. CVPR, 2018.
[9] K. Zhang, W. Zuo, Y. Chen, D. Meng, L. Zhang. Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising. Trans. on Image Proc. , 2017.

Acknowledgements. The research leading to these results has received funding from the European Research Council under the European Union's Seventh Framework Programme (FP/2007-2013)/ERC Grant agreement No. 307942.

Experiments: Gaussian Denoising

- Baseline architecture: DnCNN [9]
- 3x 6-layer DnCNN interleaved with 2x N³ Block (k=7)
- Training on 400 images of BSDS dataset
- Individual model per noise level

Results:

- Continuous nearest neighbors with learned embedding outperforms KNN selection on static embedding
- Significant and consistent gain over DnCNN baseline
- Outperforms other strong local denoisers
- Outperforms competing non-local methods with KNN selection on fixed feature space

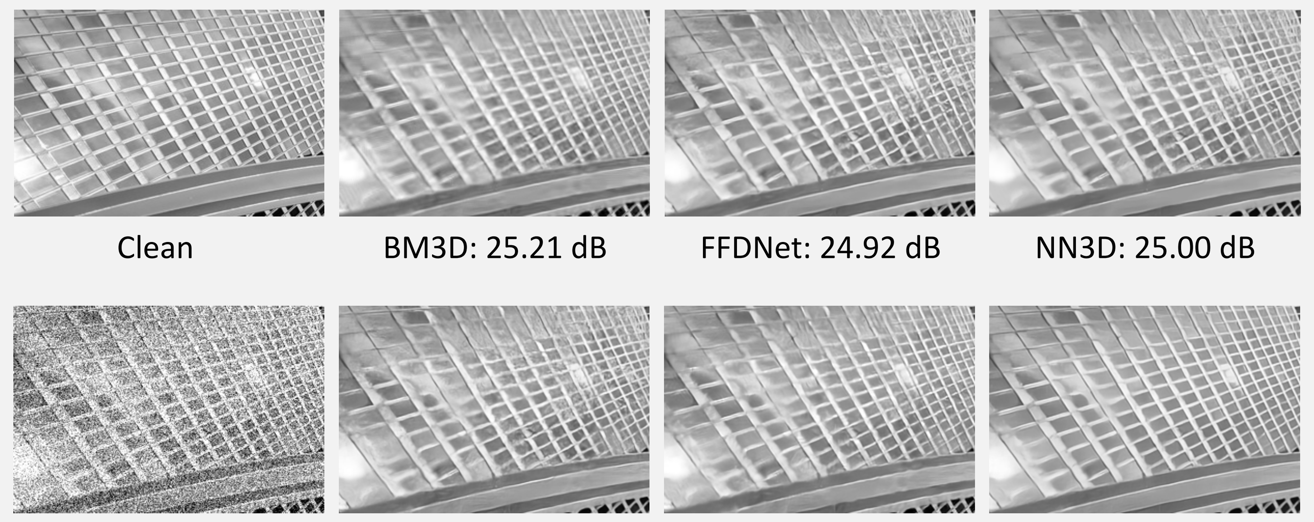
PSNR values [dB] for Gaussian denoising (higher is better).

Dataset	σ	BM3D	NLNet	UNLNet	NN3D	DnCNN	RED30	FFDNet	N³Net
Set12	25	29.96	30.31	30.27	30.45	30.44	–	30.43	**30.55**
	50	26.70	27.04	27.07	27.24	27.19	27.24	27.31	**27.43**
	70	25.21	–	–	25.61	25.56	25.71	25.81	**25.90**
BSD68	25	28.56	29.03	28.99	29.19	29.23	–	29.19	**29.30**
	50	25.63	26.07	26.07	26.19	26.23	–	26.29	**26.39**
	70	24.46	–	–	24.89	24.85	–	25.04	**25.14**
Urban100	25	29.71	29.92	29.80	30.09	29.97	–	29.92	**30.19**
	50	25.95	26.15	26.14	26.47	26.28	26.32	26.52	**26.82**
	70	24.27	–	–	24.53	24.36	24.63	24.87	**25.15**

Non-local baselines (BM3D, NLNet, UNLNet, NN3D) | Local CNN baselines (DnCNN, RED30, FFDNet) | Ours (N³Net)

Model	Matching on	PSNR [dB]	SSIM
1 × DnCNN (d=17)	–	29.97	0.879
1 × DnCNN (d=18)	–	29.92	0.885
3 × DnCNN (d=6), KNN block (k=7)	noisy input	30.07	0.891
3 × DnCNN (d=6), KNN block (k=7)	DnCNN output (d=17)	30.08	0.890
3 × DnCNN (d=6), Concrete block (k=7)	learned embedding	29.97	0.889
2 × DnCNN (d=6), N³ block (k=7)	learned embedding	29.99	0.888
3 × DnCNN (d=6), N³ block (k=7)	learned embedding	**30.19**	**0.892**

Ablation study on Urban100:
Learning the embedding space with our N3 block works best.



Clean | BM3D: 25.21 dB | FFDNet: 24.92 dB | NN3D: 25.00 dB
Noisy: 14.16 dB | DnCNN: 24.76 dB | UNLNet: 25.47 dB | N³Net: 25.57 dB

Experiments: RAW Image Denoising

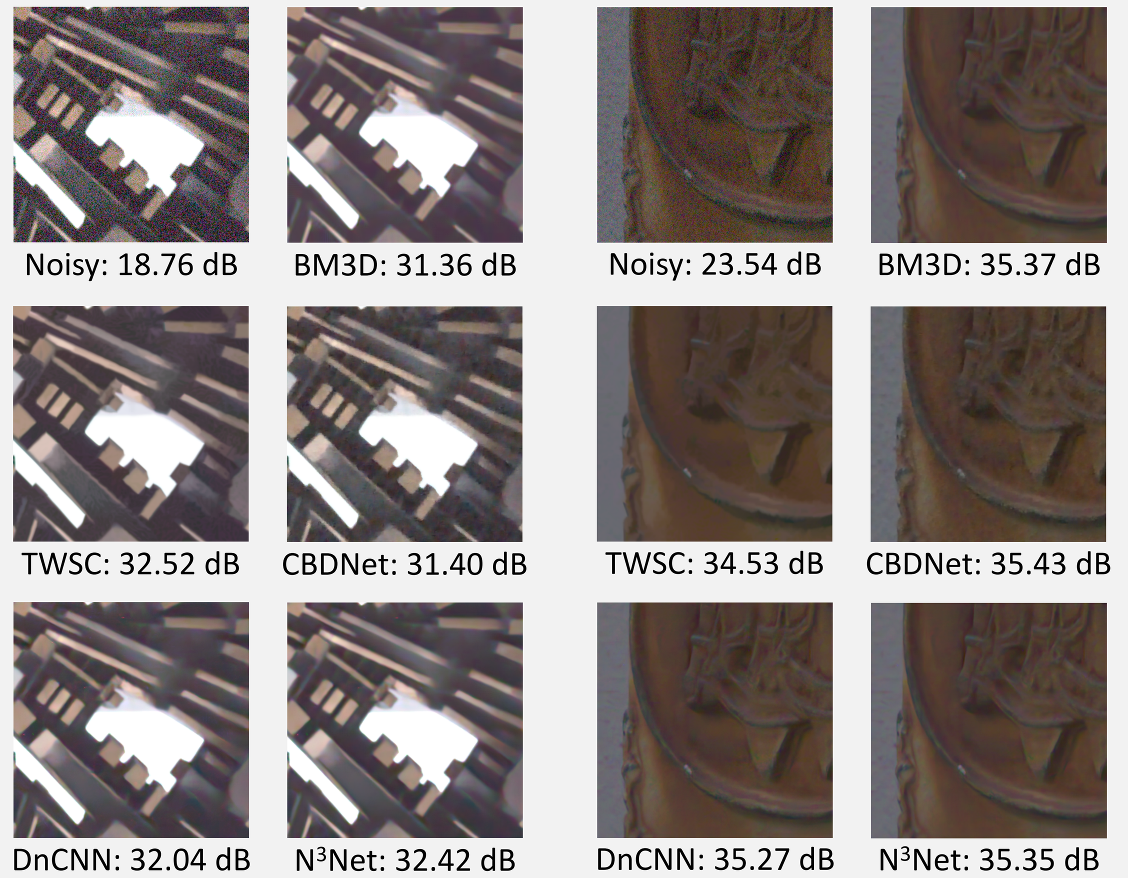
- Baseline architecture: DnCNN [6]
- 3x 6-layer DnCNN interleaved with 2x N³ Block (k=7)
- Trained on 400 images of BSDS, 800 images of DIV2K, and 3793 images of Waterloo dataset
- Single model for wide range of noise level functions
- Data augmentations to approximately invert camera processing

Results:

- New state of the art on raw image denoising benchmark DND [4]
- Outperforms DnCNN baseline
- Outperforms dedicated sRGB denoising methods

	Raw		sRGB	
	PSNR	SSIM	PSNR	SSIM
BM3D	46.64	0.9724	37.78	0.9308
DnCNN	47.37	0.9760	38.08	0.9357
N³Net	**47.56**	**0.9767**	**38.32**	0.9384
TWSC	–	–	37.94	0.9403
CBDNet	–	–	38.06	**0.9421**

Results on Darmstadt Noise Dataset [4]



Noisy: 18.76 dB | BM3D: 31.36 dB | Noisy: 23.54 dB | BM3D: 35.37 dB
TWSC: 32.52 dB | CBDNet: 31.40 dB | TWSC: 34.53 dB | CBDNet: 35.43 dB
DnCNN: 32.04 dB | N³Net: 32.42 dB | DnCNN: 35.27 dB | N³Net: 35.35 dB

Experiments: Correspondence Classification

- Baseline architecture: Context Normalization Net [8] (CNNet)
- Added one N³ block (k=3) at middle layer of CNNet
- Input: Set of pairs of image coordinates
- Output: Classification score for each putative correspondence

Results: N³ block significantly improves accuracy across all settings

	St. Peter / St. Peter			St. Peter / Reichstag			Brown / Brown		
Thr.	No Net	CNNet	N³Net	No Net	CNNet	N³Net	No Net	CNNet	N³Net
5°	0.014	0.271	**0.316**	0.0	0.173	**0.231**	0.054	0.236	**0.293**
10°	0.030	0.379	**0.431**	0.038	0.337	**0.442**	0.110	0.333	**0.391**
20°	0.071	0.522	**0.574**	0.111	0.500	**0.601**	0.232	0.463	**0.510**

Mean average precission values (higher is better) for different error thresholds.
N³Net achieves highest accuracy for indoor and outdoor scenes.