Neural Nearest Neighbors Networks

Project No.: 1 | Team Chaos | Team No.: 28 |

Team Members

S. No.	Name	Roll. No.	E-Mail
1	Aman Kumar Singh	2018102025	aman.s@students.iiit.ac.in
2	Avneesh Mishra	2021701032	avneesh.mishra@research.iiit.ac.in
3	Jhansi Mallela	2021802001	jhansi.mallela@research.iiit.ac.in
4	Nayan Anand	2021701014	nayan.anand@research.iiit.ac.in

Course: **Statistical Methods in AI** (Monsoon 2021, IIIT Hyderabad)

GitHub Link: TheProjectsGuy/SMAI21-CS7.403-Project

Problem Statement

Existing non-local methods that aim to exploit the self-similarity of natural signals mostly rely on k-Nearest Neighbors (kNN) matching in a fixed feature space. The non-differentiable selection rule of kNN (which is primarily flat with sudden changes for any parameterization of the representation) makes it very challenging to find an optimal feature space representation where the nearest neighbor search would yield the best results. Solutions to tasks such as image denoising, single image super-resolution, and correspondence classification can benefit significantly from a dynamic feature space representation depending on comparison candidates.

Work such as Neighborhood Component Analysis delivers a selection rule as k categorical distributions. The primary aim of this work is to provide a continuous deterministic relaxation to replace the one-hot coded weight vectors employed in this field. The described procedure yields a novel fundamental block that can be used in building neural networks that work well in exploiting non-local similarities in data. This block, defined as the N^3 block, outperforms benchmark algorithms in the tasks mentioned above.

Goals and Approach

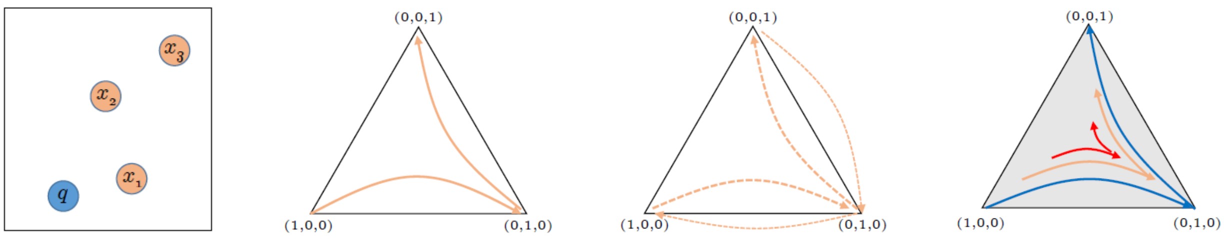
The principal goal of this project is to implement a differentiable relaxation of the kNN selection rule. Thereby facilitating the aggregation of information from many neighbors at once into abstract feature spaces, over which the nearest neighbor search best exploits the structure of the data (local and non-local spaces). We strongly follow the work of *T Plotz et al.* [1] for our implementation and compare the effectiveness of our proposed method with the benchmarks in the tasks of

1. **Image Denoising**: An image (with added Gaussian noise) is retrieved through the computation of a residual. Comparisons with images from Darmstadt Noise Dataset (with real image denoising) are also performed.
2. **Single Image Super-Resolution**: An image is upsampled from a lower resolution. Comparisons are made with state-of-the-art algorithms for different resolution scaling for the Set5 dataset.
3. **Correspondence Classification**: The likelihood of a putative correspondence being correct is computed.

This is done by first creating the novel N[3] block proposed by Plotz in [1]. This is in the form of a relaxation in nearest neighbor selection criteria, listed as follows (least to most relaxed)

1. Simple kNN Rule
2. KNN rule as a limit distribution (from Neighborhood Component Analysis)
3. Continuous deterministic relaxation of the KNN rule

A brief description of the above three are given in the following figure, followed by a more formulated description.



(a) (b) (c) (d)

On comparing the query with candidates, as shown in (a), the kNN rule selects corners (selection of absolute points based on distances) as seen in (b). The Stochastic Neighbors selection (from Neighborhood Component Analysis) performs a walk on the corners as seen in (c). The proposed continuous relaxation enables exploration of the items interior to the simplex, as seen in (d), thereby allowing for the best weightage.

Simple kNN Rule

The kNN rule first creates a permutation that sorts the database of candidate items according to the distance to the query. Let q be a query item for a database of candidate items x_i where $i \in I = \{1, \dots, M\}$. Let $d(\cdot,\cdot)$ be the distance metric between two items. The permutation $\pi_q: I \to I$ is given by

$$\pi_q(i) < \pi_q(i') \Rightarrow d(q, x_i) \leq d(q, x_{i'}), \qquad \forall\, i, i' \in I$$

This means to sort the list of x_i candidates by their distance from q in ascending order, and then picking the first k items (closest to q). The described method has two problems: non-differentiability and not choosing the best feature representation of candidates (d could be the Euclidean distance).

KNN Rule as Limit

Here, the selection rule is visualized as a distribution of k categorical distributions over the indices I (over candidate items). Here, the negative of distances is used as non-normalized predictions α_i (also known as logits). This is scaled by a temperature parameter T to obtain a normalized probability distribution over I, which is encapsulated in ω.

$$\mathbb{P}[\omega^1 = i \mid \alpha^1, T] \;\equiv\; \mathrm{Cat}(\alpha^1, t) = \frac{\exp(\alpha_i^1/T)}{\sum_{j\in I} \exp(\alpha_j^1/T)}$$

Where $\alpha_i^1 = -d(q, x_i)$. The above is like the softmax (with temperature). The higher the temperature, the more equally normalized the result; the lower the temperature, the more peaked the result. Note that ω is treated as a one-hot vector. The above is generalized to an arbitrary k through sampling $\mathbb{P}$ and applying updates to α (logits).

$$\alpha_i^{t+1} = \alpha_i^t + \log\left(1 - \omega_i^t\right) = \begin{cases} \alpha_i^t & \text{if } \omega^t \neq i \\ -\infty & \text{if } \omega^t = i \end{cases}$$

The updated distribution parameters are given by

$$\mathbb{P}[\omega^{t+1} = i \mid \alpha^{t+1}, T] \equiv \mathrm{Cat}(\alpha^{t+1}, T) = \frac{\exp(\alpha_i^{t+1}/T)}{\sum_{j\in I}\exp(\alpha_j^{t+1}/T)}$$

The stochastic nearest neighbors $\{X^1, X^2, \ldots, X^k\}$ of q are derived using the distribution parameter ω^t (at each time, pick the best distribution over candidates and proceed to next), using

$$X^n = \sum_{i\in I} \omega_i^n x_i$$

The above form makes it possible to compute derivatives over pairwise distances. However, gradients over discrete variables can still be treacherous (over a non-uniform terrain). Continuous expectations of ω are used to rectify this problem.

Continuous Deterministic Relaxation

The one-hot coded weight vectors are replaced with their expectations as follows (for the first time step)

$$\bar{\omega}_i^1 \equiv \mathbb{E}\left[\omega_i^1 \mid \alpha^1, T\right] = \mathbb{P}[\omega^1 = i \mid \alpha^1, T] = \mathrm{Cat}(\alpha^1, t) = \frac{\exp(\alpha_i^1/T)}{\sum_{j\in I}\exp(\alpha_j^1/T)}$$

The update to the logits is applied using

$$\bar{\alpha}_i^{t+1} = \bar{\alpha}_i^t + \log\left(1 - \bar{\omega}_i^t\right) \text{ with } \bar{\alpha}_i^1 = \alpha_i^1 = -d(q, x_i)$$

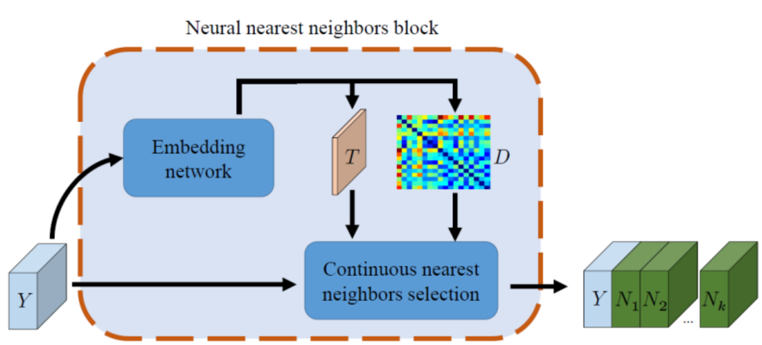
Using this, we can define the k continuous nearest neighbors $\{\bar{X}^1, \bar{X}^2, \ldots, \bar{X}^k\}$ of q to be

$$\bar{X}^n = \sum_{i\in I} \bar{\omega}_i^n\ x_i$$

The above equations will result in a much smoother terrain for finding derivatives. This concept of being able to find the gradients for the distance metric d and temperature T is exploited in the implementation of the novel N^3 block

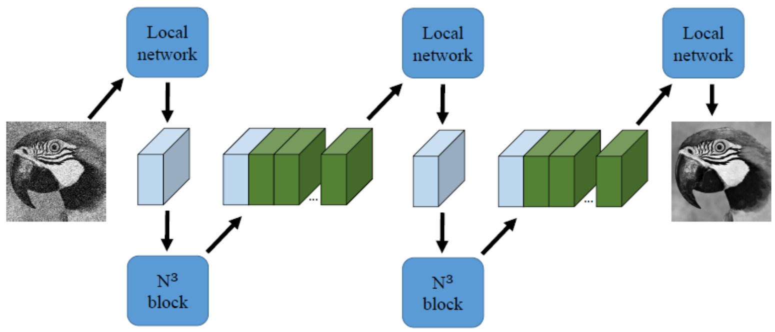
N^3 Block Implementation

The N3 block is supposed to incorporate the continuous deterministic relaxation to estimate the optimal embedding of candidates (for calculating the distance) and the temperature parameter.



The nearest neighbor block incorporates two networks, one to estimate the feature embedding f_E and another to estimate the temperature parameter f_T (which directly yields T). The output D is constructed by finding the pairwise distance among the candidates, that is $D_{ij} = d\left(f_E(x_i), f_E(x_j)\right)$. These are passed to the nearest neighbor selector that generates the continuous nearest neighbors $\{N_1, N_2, \ldots, N_k\}$ of Y (these are $\{\bar{X}^1, \bar{X}^2, \ldots, \bar{X}^k\}$ and q, respectively, in the equations above).

The N^3 block can be embedded into various networks to induce inference based on properties found from multiple non-local candidates. This can be done in the fashion shown below (particular to image denoising)



We call the resulting augmented network N^3 Net (for any particular task). **DnCNN** proposed by *Zhang et al.* [2] is modified for image denoising. **VDSR** presented by *Kim et al.* [3] is augmented for single image super-resolution.

In an end-to-end trainable pipeline, using the N^3 blocks can integrate non-local processing, which, through implementation, can be shown to outperform benchmark models that do not.

Datasets

There are various datasets used in this project, depending on the example task to demonstrate the effectiveness of the N^3 Net.

Image Denoising (Gaussian Noise)

S. No.	Dataset	Description	Purpose
DS1.1	BSDS500	200 training, 200 testing, and 100 validation images primarily for image segmentation and boundary detection	Training N^3 Net created by augmenting DnCNN, proposed by *Zhang et al.* [2]. Sixty-eight images from the validation are used for evaluation, whereas 400 from training and test are used for training
DS1.2	Set12	Set of 12 commonly used images	Model evaluation and comparison
DS1.3	Urban100	100 images of urban scenes	Model evaluation and comparison

Real Image Denoising

S. No.	Dataset	Description	Purpose

DS2.1 (or DS1.1)	BSDS500	Contains 200 training, 200 testing, and 100 validation images	Training (training + test split, 400 photos)
DS2.2	DIV2K	High quality (2K) images	Training (800 images)
DS2.3	Waterloo database	Generalized Image Quality Assessment dataset	Training (3793 images)
DS2.4	Darmstadt Noise Dataset	Contains noisy images shot with varying ISO and cameras	Model evaluation (50 images)

Single Image Super-Resolution

S. No.	Dataset	Description	Purpose
DS3.1	BSDS500	Contains 200 training, 200 testing, and 100 validation images	Training (200 images from the training set)
DS3.2	Set5	Contains five commonly used images for validating and comparing super-sampling	Model evaluation and comparison

Correspondence Classification

Outdoor Setting

S. No.	Dataset	Description	Purpose
DS41.1	St. Peter	No information found	Training and testing (splits available)
DS41.2	Reichstag	No information found	Testing (regularization test)

Indoor Setting

S. No.	Dataset	Description	Purpose
DS42.1	Brown	No information found	Training and testing

Expected Deliverables

The following can be expected out of this project

ED No.	Description
ED1	Creating data loaders and visualizing all data used in this project (from datasets listed above)
ED2	Verification of benchmarks mentioned in [1] for local and non-local processing networks - **Image denoising**: Local (DnCNN, FFDNet, etc.) and non-local (BM3D, NLNet, etc.) - **Single Image Super-Resolution**: Bicubic interpolation, local (VDSR, MemNet, etc.), non-local (SelfEx, etc.) - **Correspondence classification**: Context Normalization Network
ED3	Implementing the N^3 block using TensorFlow
ED4	Integrating the N3 block with DnCNN for image denoising and VDSR for single image super-resolution.
ED5	Verification of the results (comparisons with benchmarks): for the image denoising operation (*Gaussian noise estimation* and *real image denoising*).
ED6	Verification of results for Single Image Super-Resolution experiment

ED7	Verification of results for Correspondence Classification experiment
ED8	Incorporation of more datasets for verification and training, like SIDD, etc.
ED9	Incorporation of more variation in models for training and validation across all tasks.

Milestones & Timeline

Timeline	ED	Milestones
12th Oct 2021	-	Project Announcement (link)
26th Oct 2021	-	Project Allocation (link)
11th Nov 2021	-	**Project Proposal Submission**
	ED1 & 2	Implementation and verification of benchmarks with data
17th to 20th Nov 2021	-	Mid Evaluation
N/A	-	Implementation of N3 Block
N/A	-	Implementation of N3 Nets and verification of benchmark score
N/A	-	Implementation of Real Image Denoising on the Darmstadt Noise Dataset
N/A	-	Implementation of Single Image Super Resolution experiment
N/A	-	Implementation of Correspondence classification experiment
N/A	-	Verification of results for the above experiments
N/A	-	Verifying the conclusion of the paper and presentation preparation
N/A	-	Finishing steps
1st to 4th Dec 2021	-	Final Presentation
4th Dec 2021 at 23:59	-	Submission

Work Distribution

Entire work is distributed in the following manner over the members of the team (tentative)

Aman Kumar Singh – 2018102025

1. Implementation of N3 Block
2. Implementation of Single Image Super-Resolution benchmarks
3. Adapting the N3 Net for Single Image Super-Resolution
4. Finishing steps

Avneesh Mishra – 2021701032

1. Implementation of N3 Block
2. Repo creation and common folder creation
3. Data download and visualization for Image denoising
4. Benchmark verification for Image denoising (DnCNN)
5. Comparison with benchmark for Image denoising (for gaussian noise)

Jhansi Mallela – 2021802001

1. Implementation of N3 Block
2. Understanding the problem of correspondence classification
3. Implementation of the baseline CNNet for correspondence classification
4. Augmentation of CNNet with N3 block

Nayan Anand – 2021701014

1. Implementation of N3 Block
2. Implementation of Real Image Denoising on the Darmstadt Noise Dataset
3. Implementation of Single Image Super Resolution experiment

References

Main references

[1] Plotz, Tobias and Roth, Stefan, "Neural Nearest Neighbors Networks," vol. 31, NeurIPS 2018. arXiv: 1810.12575

[2] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a Gaussian Denoiser: Residual Learning of Deep CNN for Image Denoising," IEEE Transactions on Image Processing, vol. 26, no. 7. Institute of Electrical and Electronics Engineers (IEEE), pp. 3142–3155, Jul. 2017. doi: 10.1109/tip.2017.2662206.

[3] J. Kim, J. K. Lee, and K. M. Lee, "Accurate Image Super-Resolution Using Very Deep Convolutional Networks," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, Jun. 2016. doi: 10.1109/cvpr.2016.182.

Other references

1. Residual learning of deep CNN for image denoising. IEEE T. Image Process., 26(7):3142–3155, 2017.
2. Viren Jain and H. Sebastian Seung. Natural image denoising with convolutional networks. In NIPS*2008, pages 769–776.
3. Stamatios Lefkimmiatis. Universal denoising networks: A novel CNN-based network architecture for image denoising. In CVPR, pages 3204–3213, 2018.
4. Weiqiang Ren, Yinan Yu, Junge Zhang, and Kaiqi Huang. Learning convolutional nonlinear features for k nearest neighbor image classification. In ICPR, pages 4358–4363, 2014.
5. Kai Zhang, Wangmeng Zuo, Shuhang Gu, and Lei Zhang. Learning deep CNN denoiser prior for image restoration. In CVPR, pages 2808–2817, 2017.
6. Jianchao Yang, John Wright, Thomas S. Huang, and Yi Ma. Image super-resolution via sparse representation. IEEE T. Image Process., 19(11):2861–2873, 2010