Scalability of Similarity Search to couple with Deep Learning

Gihan Jayatilaka (CMSC858N, 2023 Spring)

Contents

Before we get started

- It is assumed that you are familiar with basics of
  - image classification.
    - We use image classification as the use case because it is the basic tasks in the computer vision community.
- Abbreviations (might be confusing)
  - kNN : k - Nearest Neighbour
  - NN : Neural Network

Similarity search.

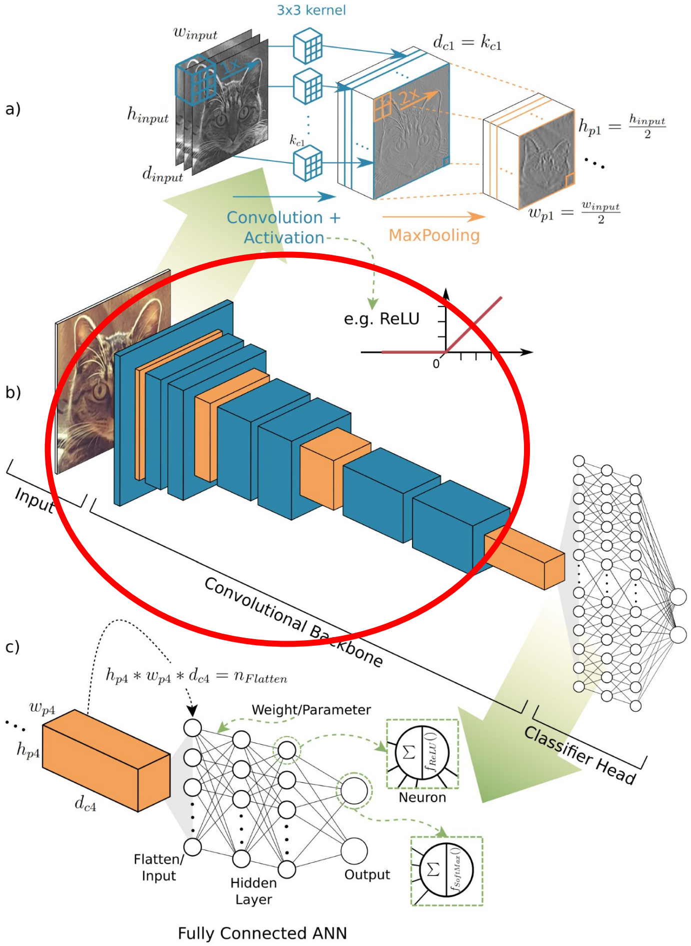
Objective: Given a set of elements and a comparator function to calculate the "similarity" of two elements, find the elements that are "most similar" to a given query.

Examples:

- Euclidean Nearest neighbour search tries to find the closest point when the similarity metric is the Euclidean distance.
- Another version of this is **kNN,** which tries to find the top k closest neighbors.

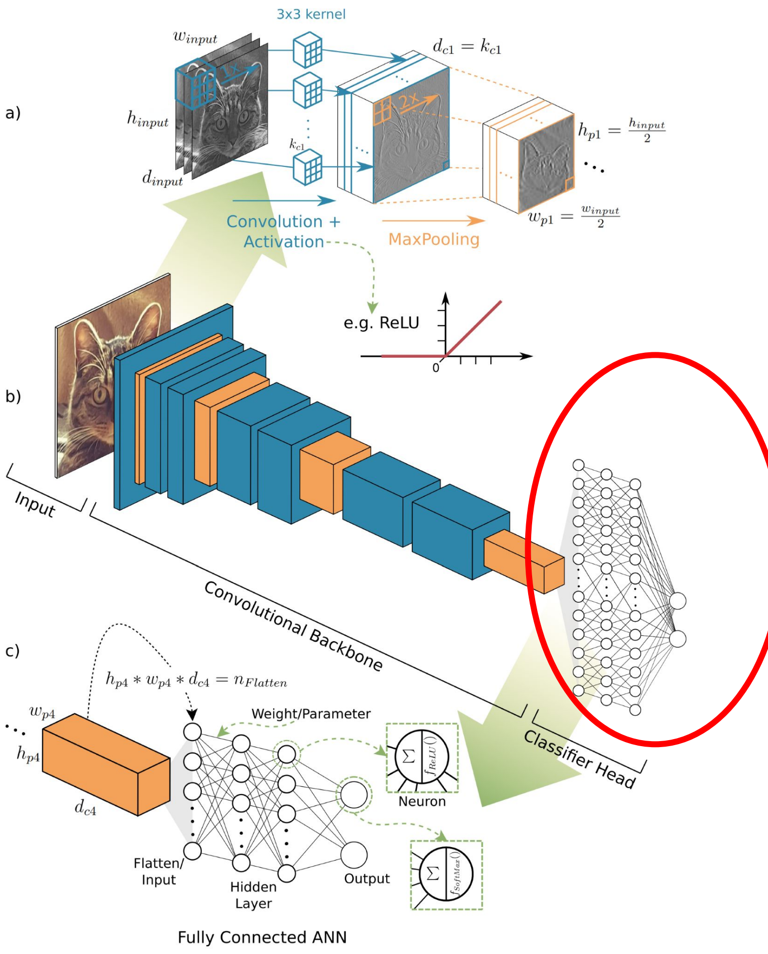
Deep Neural Networks as an Embedding Function

- Most of the Neural Networks for image classification tasks have could be understood as a combination of
    - **Backbone** (which embeds the images) mostly made up of Convolutional layers.
    -

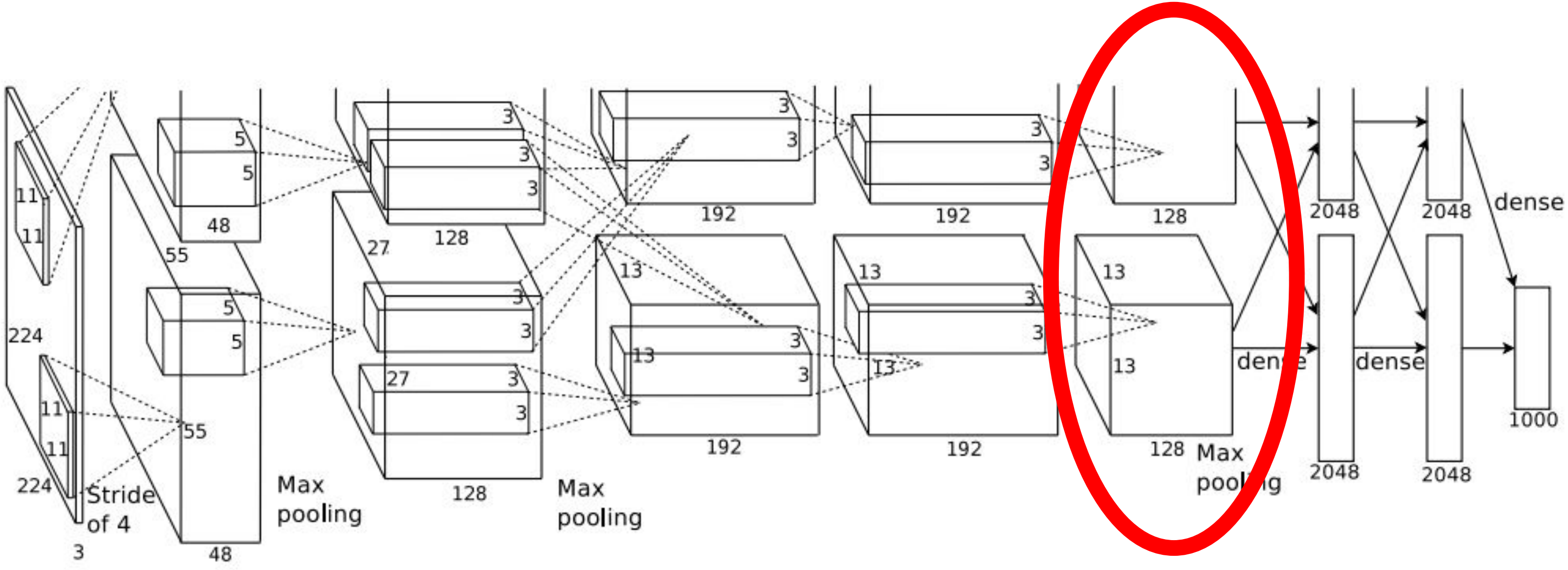


Deep Neural Networks as an Embedding Function

- Most of the Neural Networks for image classification tasks have could be understood as a combination of
  - Backbone (which embeds the images) mostly made up of Convolutional layers.
  - **Head** (which makes a decision on the embeddings) mostly made of fully connected layers.

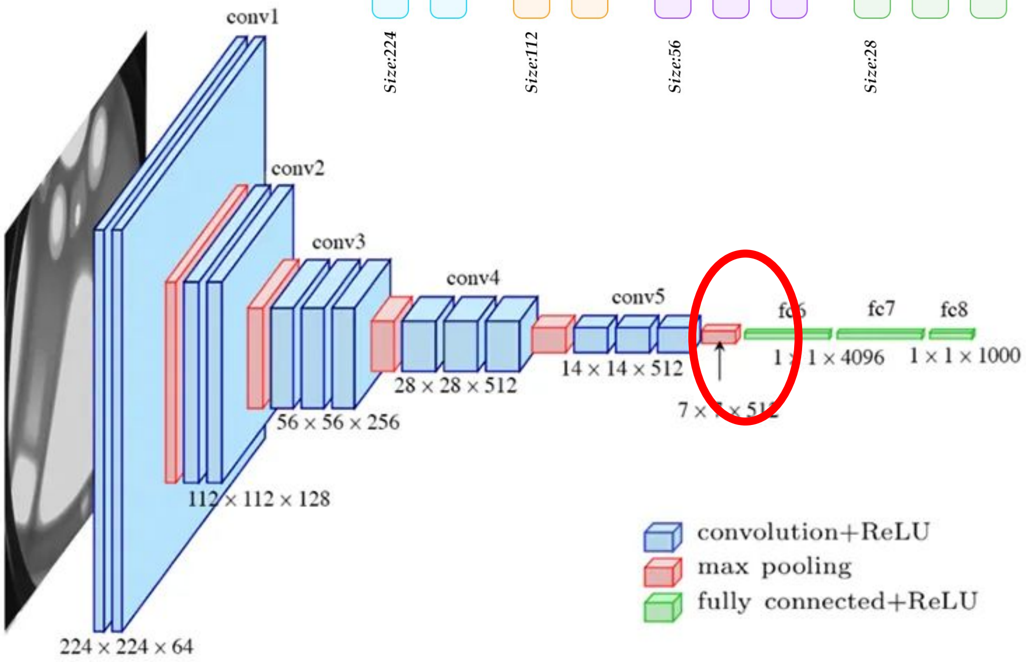


Alexnet [2012]

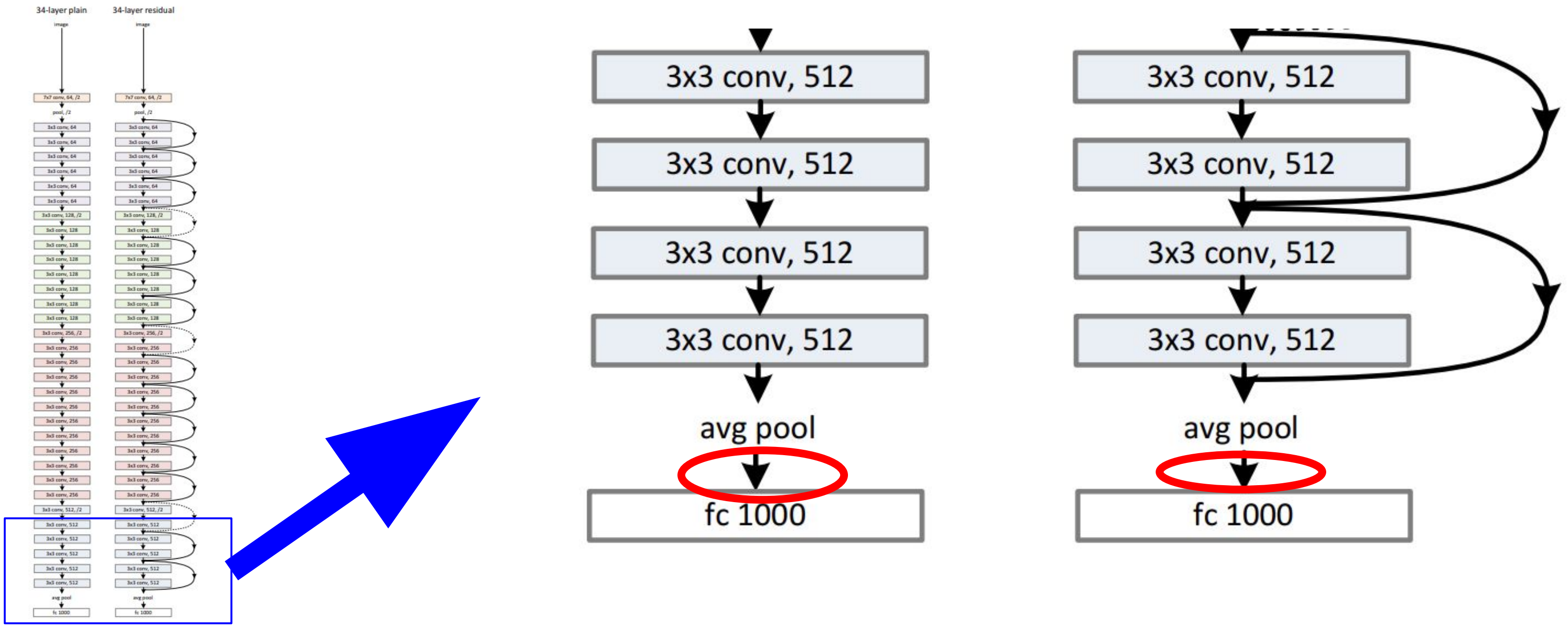


VGGnet [2015]


3x3 conv, 64
3x3 conv, 64
pool/2
3x3 conv, 128
3x3 conv, 128
pool/2
3x3 conv, 256
3x3 conv, 256
3x3 conv, 256
pool/2
3x3 conv, 512
3x3 conv, 512
3x3 conv, 512
pool/2
3x3 conv, 512
3x3 conv, 512
pool/2
fc 4096
fc 4096
fc 4096
Size:224
Size:112
Size:56
Size:28
Size:14
Size:7

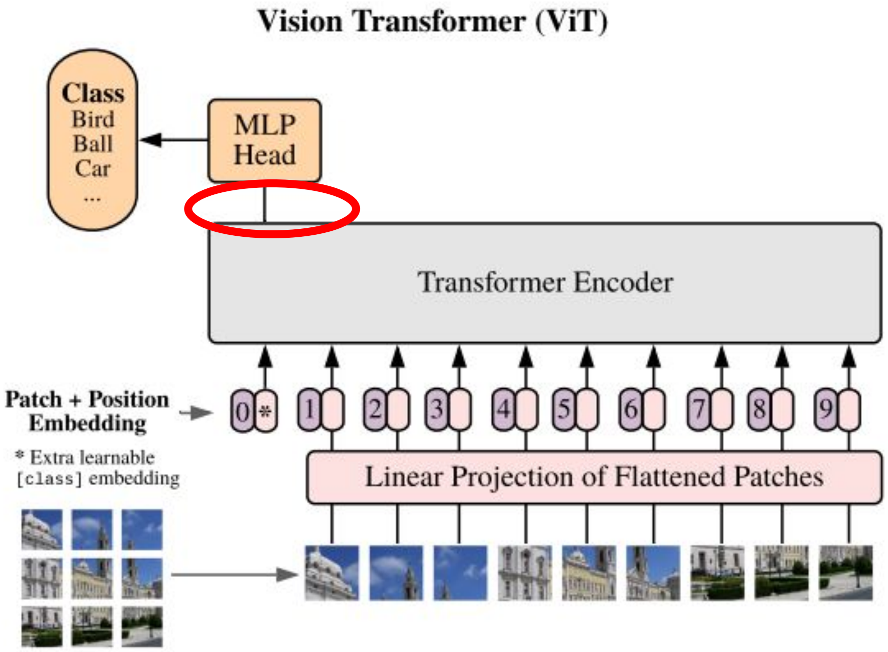

conv1
conv2
conv3
conv4
conv5
fc7
fc8
224 × 224 × 64
112 × 112 × 128
56 × 56 × 256
28 × 28 × 512
14 × 14 × 512
1 × 1 × 1000
convolution+ReLU
max pooling
fully connected+ReLU

Resnet [2015]



34-layer plain
34-layer residual
3x3 conv, 512
3x3 conv, 512
3x3 conv, 512
3x3 conv, 512
avg pool
fc 1000
3x3 conv, 512
3x3 conv, 512
3x3 conv, 512
3x3 conv, 512
avg pool
fc 1000

Vision Transformers [2021]



Similarity search on DNN embeddings.

- As of today, all the SOTA heads are trainable NNs.
- Intuitively, decision head should make classifications based on the structure of the entire dataset.
- Arguably, mini-batch-SGD trains the head to do this.

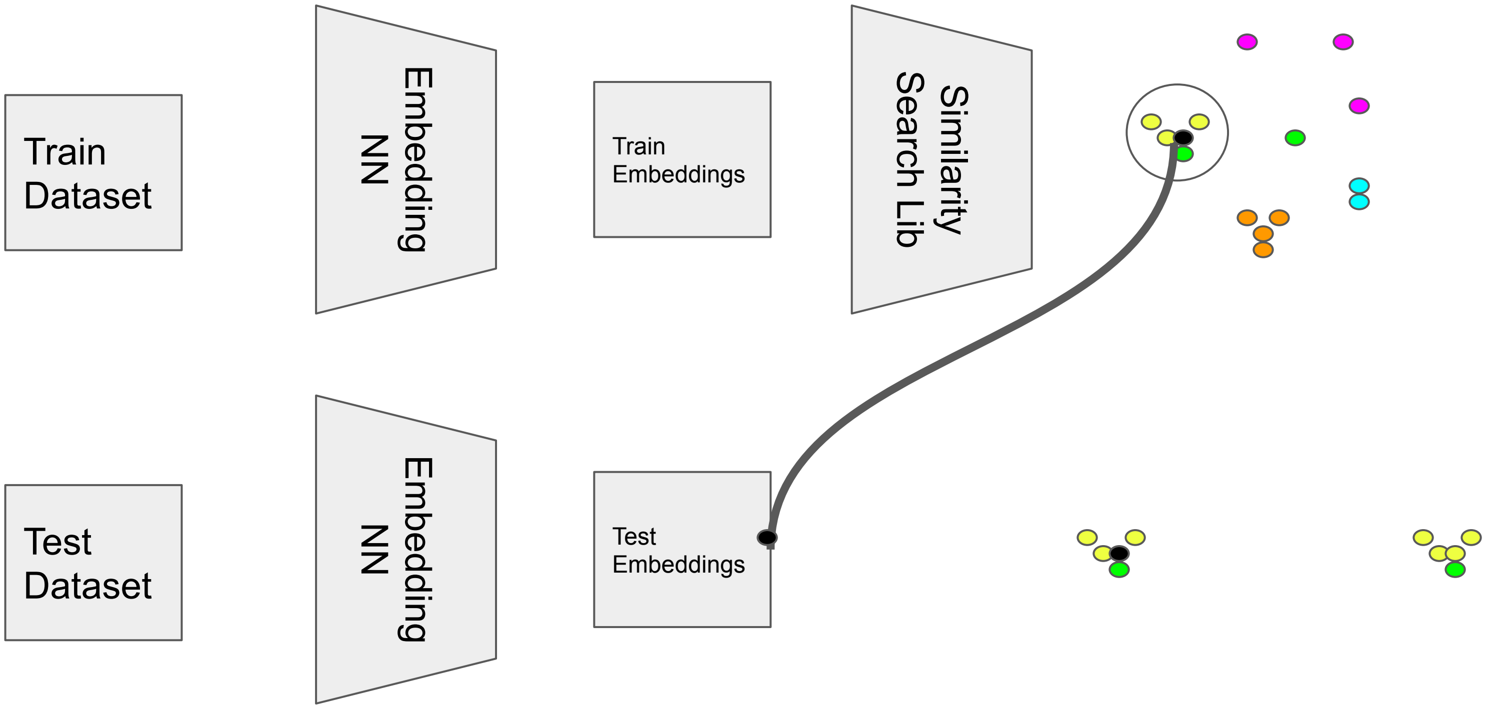
- What if we do similarity search on the entire dataset on the inference step?

Research questions

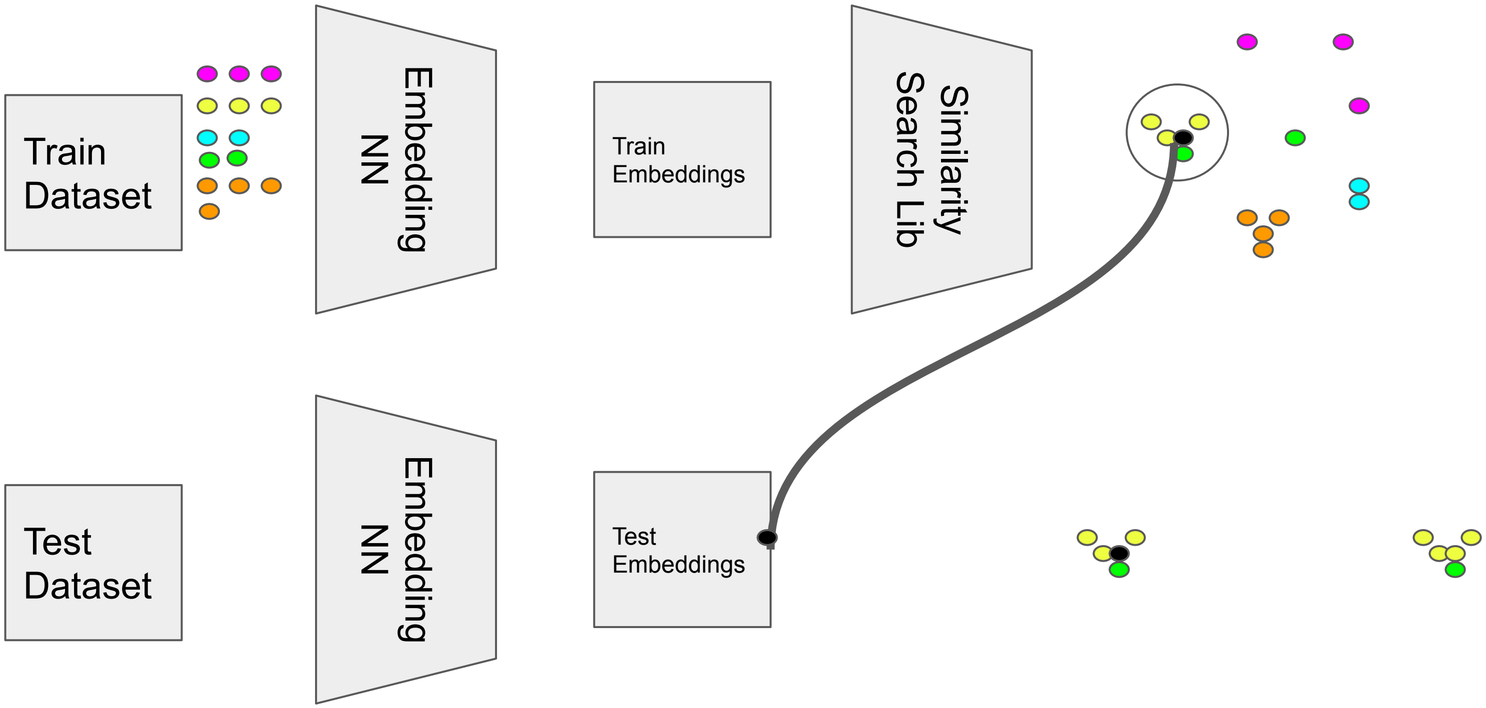
1. Can we use a similarity search to replace the trainable classification heads in modern NNs?

2. How efficient (space/time) are the existing parallel similarity search techniques to deal with high dimensional NN embeddings and huge datasets?

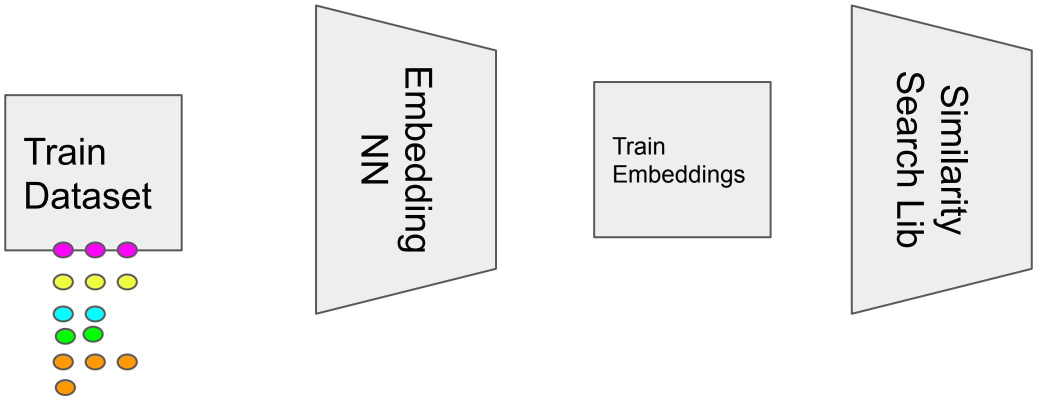
Experiments (Overview)



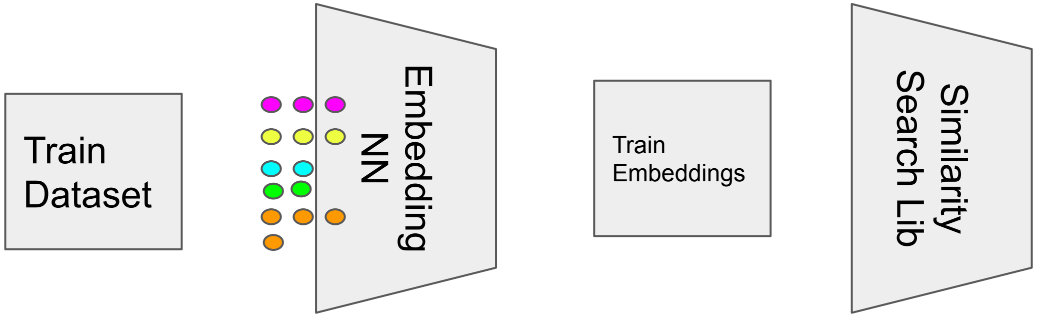
Experiments (Overview)



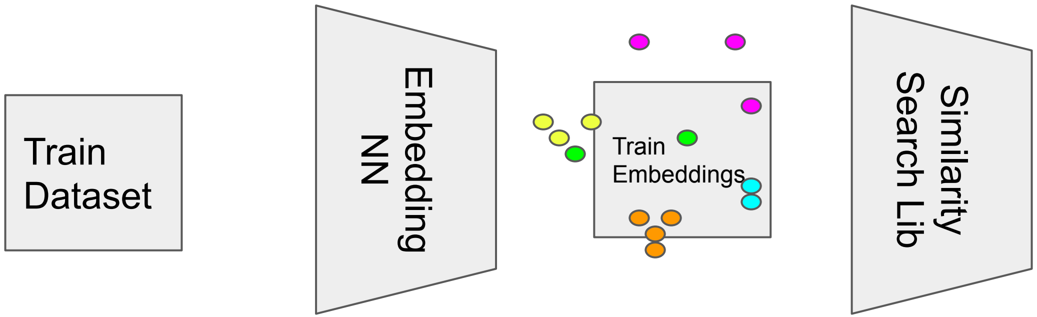
Experiments (Overview)



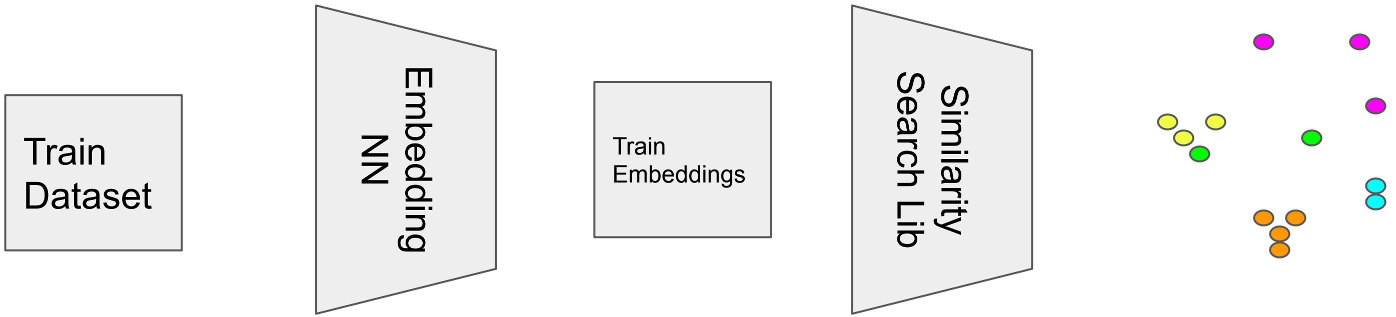
Experiments (Overview)



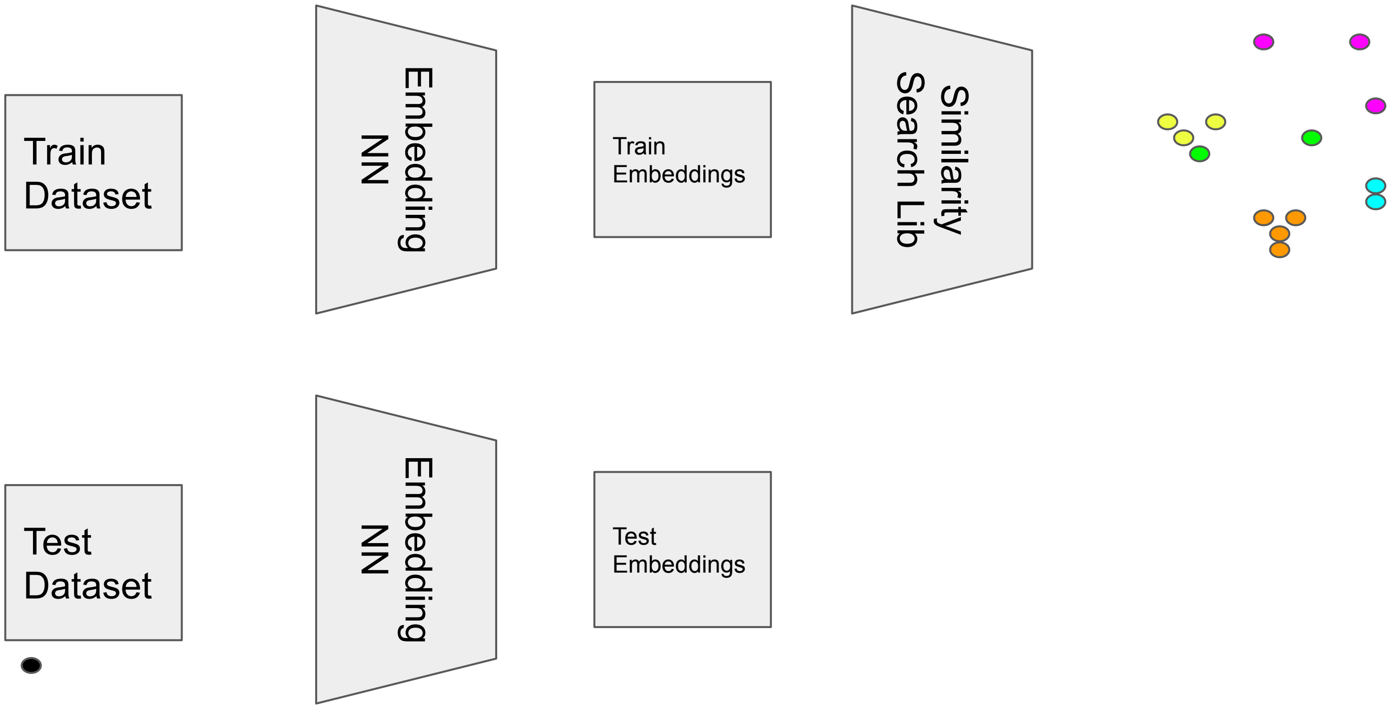
Experiments (Overview)



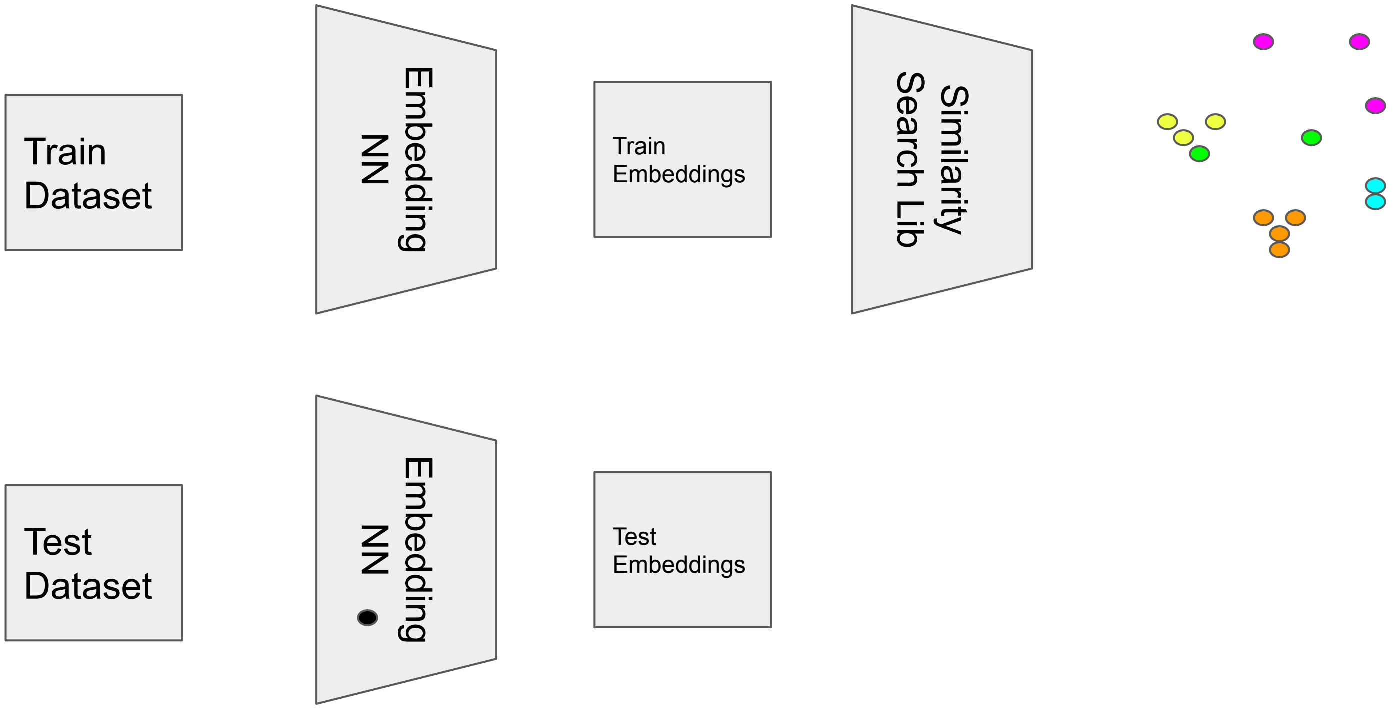
Experiments (Overview)



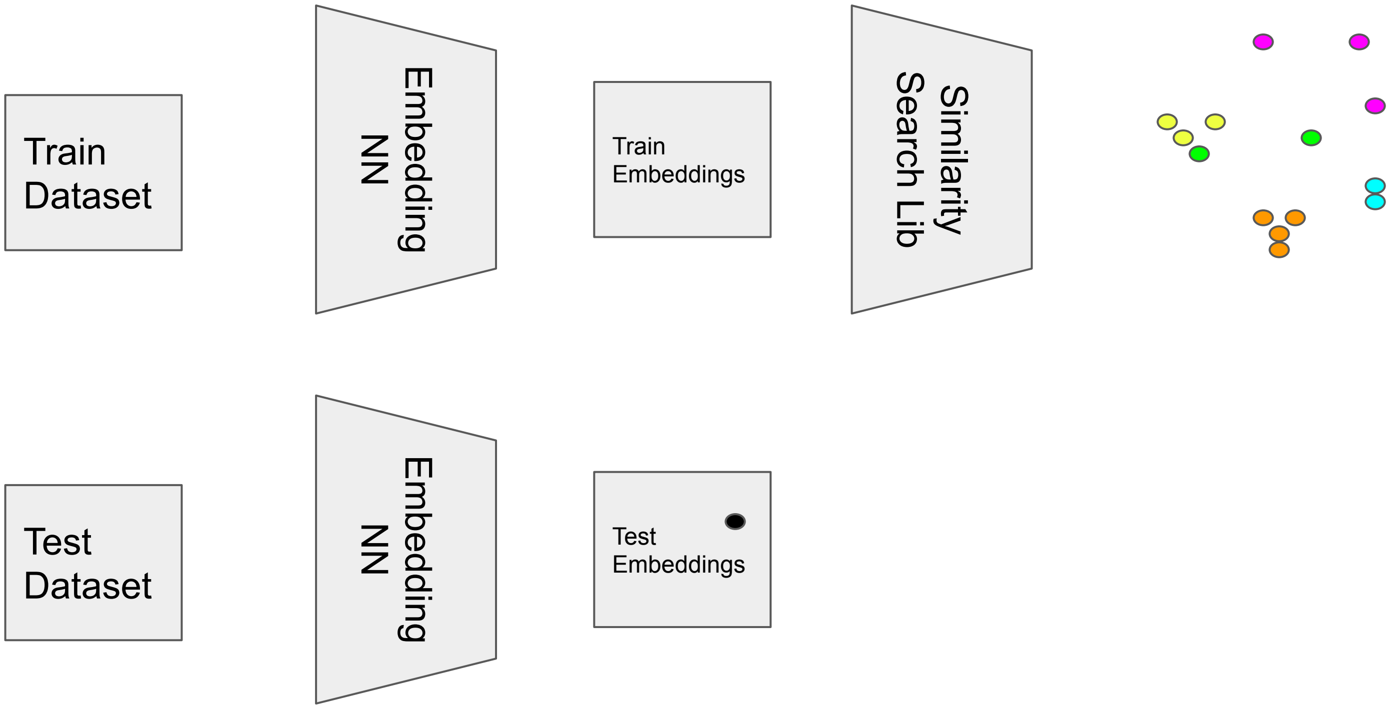
Experiments (Overview)



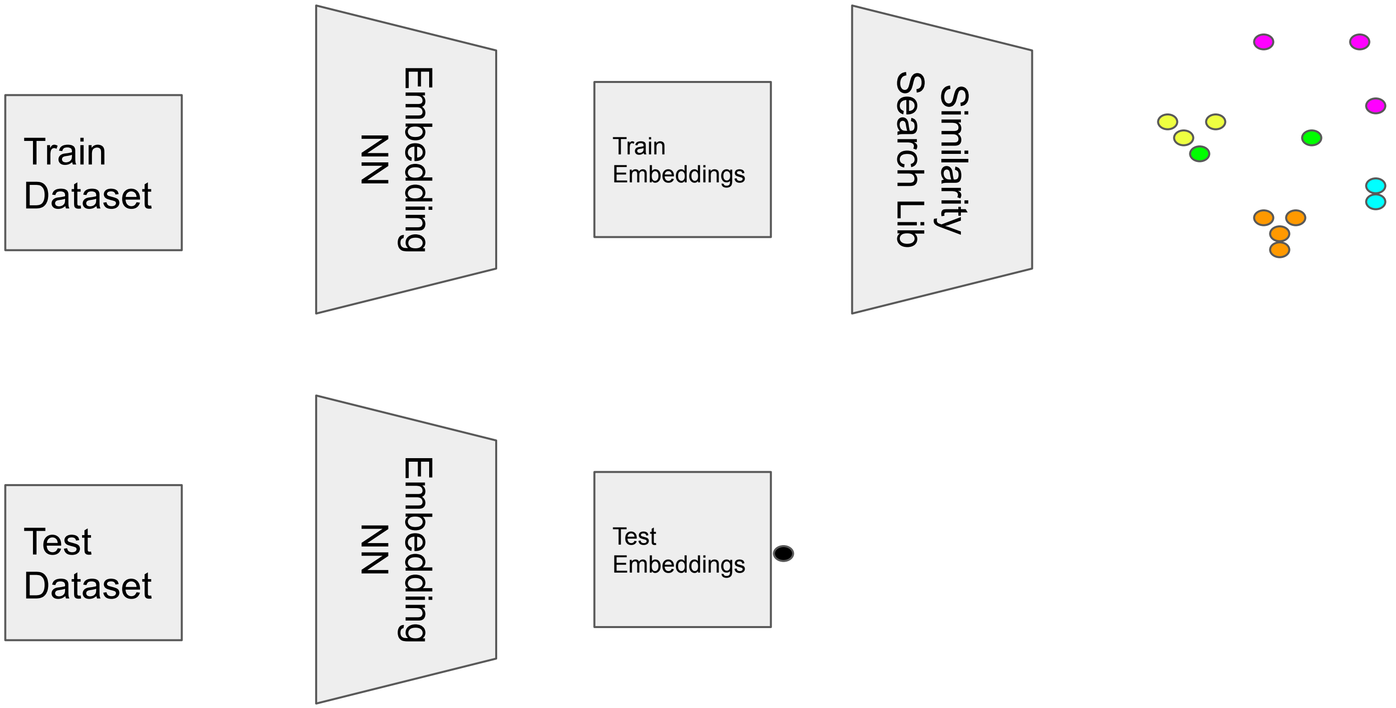
Experiments (Overview)



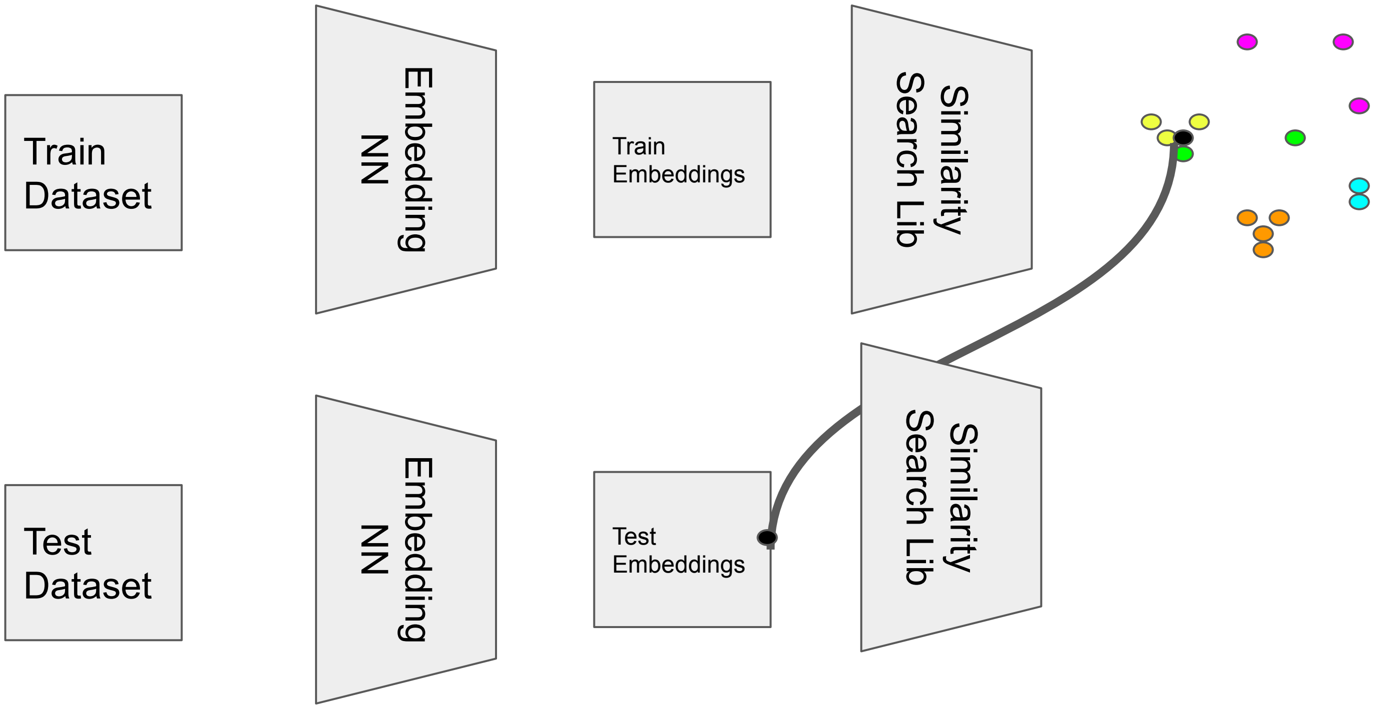
Experiments (Overview)



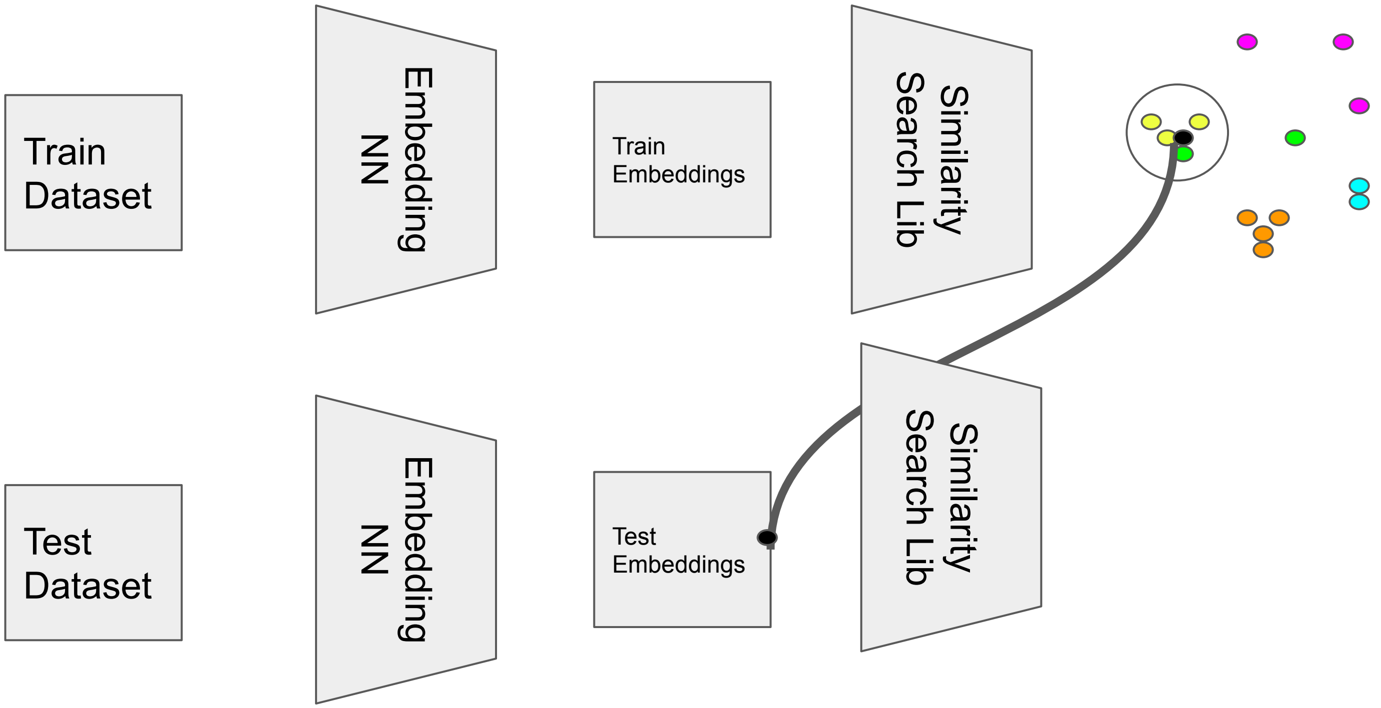
Experiments (Overview)



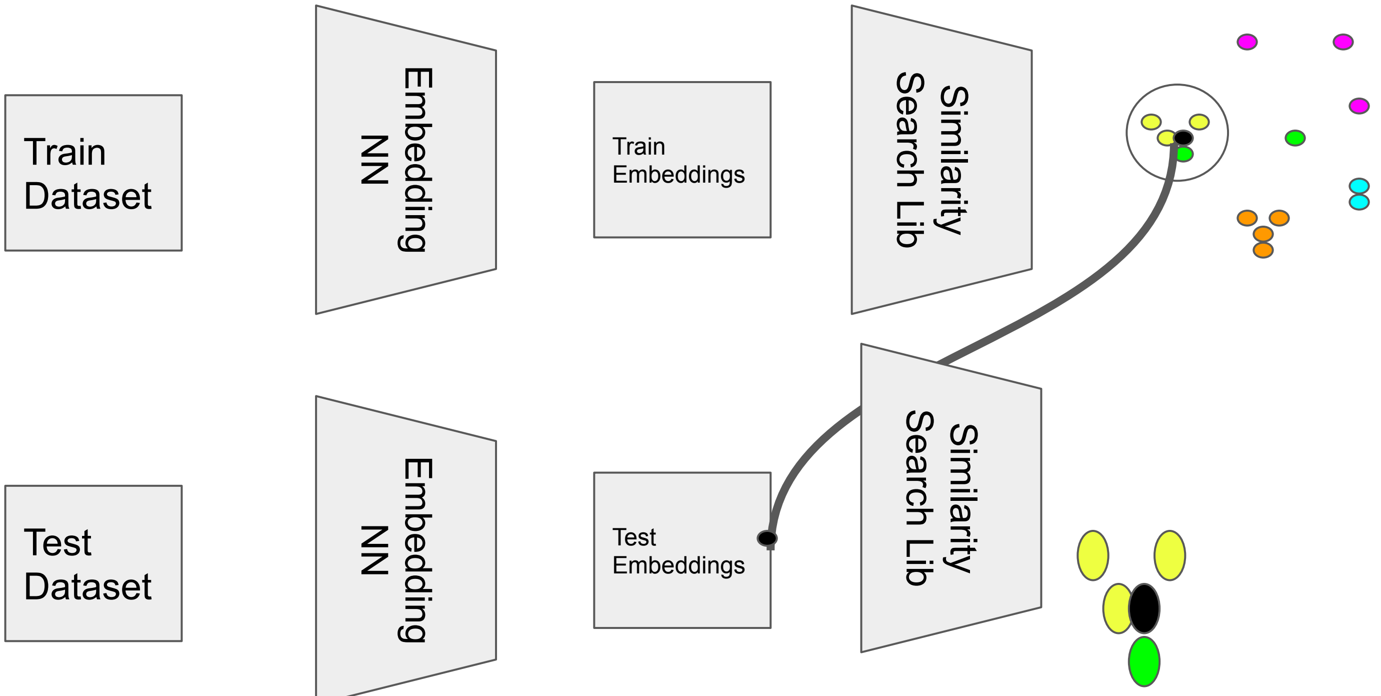
Experiments (Overview)



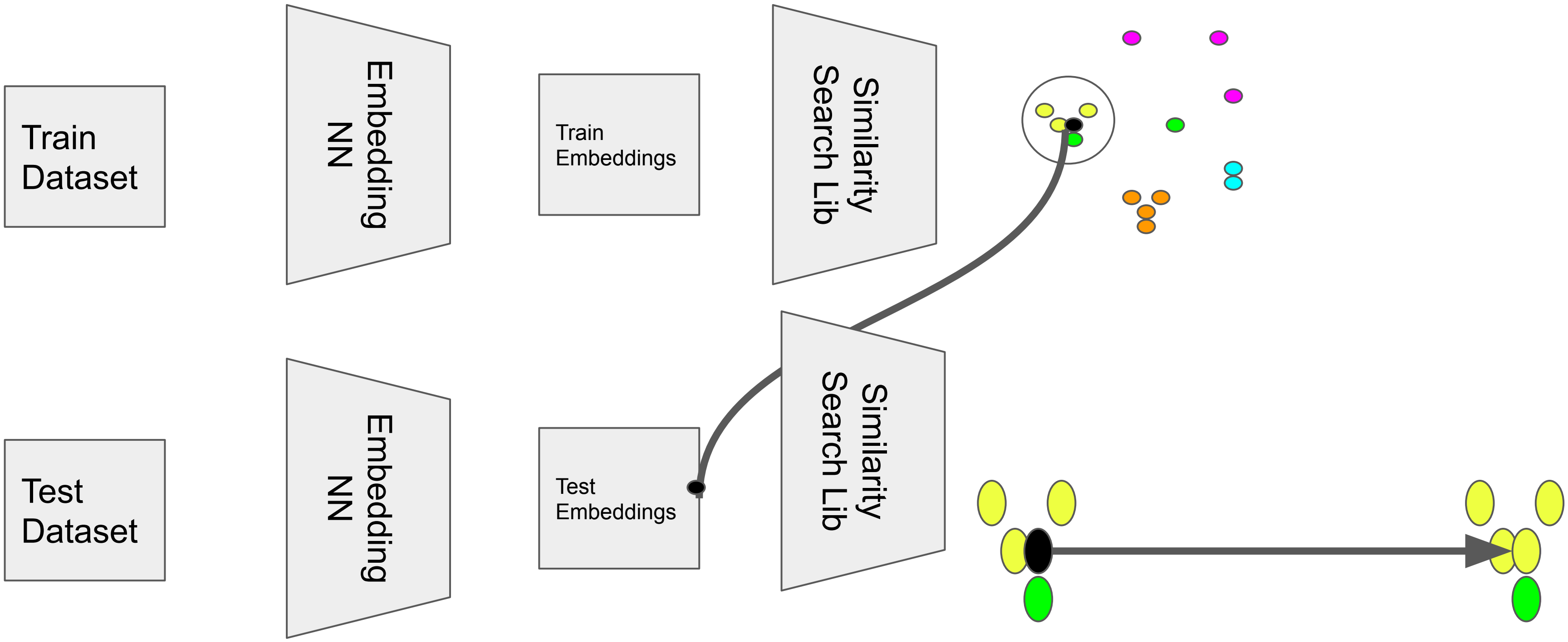
Experiments (Overview)



Experiments (Overview)



Experiments (Overview)



Experiments (Specifics)

Datasets	Tiny Imagenet 200 classes, 100,000 train and 10,000 validation images of 64x64. Imagenet 1000 classes, 1.2M train and 50,000 validation images of different sizes.
Embedding NN	ViT trained as DINO 21M parameters, 384 embedding dim
Similarity search engine	FAISS running on 32 processors and 64GB RAM A5000 GPU of 24GB RAM
Performance measures	Top 1 accuracy Time Memory usage

Results

	Embedding time / (mm:ss) Using 2x A5000 GPUs	Index building time / (s)	Train set search time/ (s)	Val set search time / (s)
Tiny Imagenet, CPU	04:58	0.03	11.40	0.75
Tiny Imagenet, GPU	04:58	0.02	7.94	0.76
Imagenet (10%), CPU	08:56	0.03	15.95	6.01
Imagenet (10%), GPU	08:56	0.12	1.32	0.46
Imagenet, GPU	64:22	0.29	108.72	4.24
Imagenet, CPU	64:22	0.30	4689.15 =78:09	364.18 =6:04

Results

	Top 1 accuracy Train / Val (ours)	**Top 1 accuracy (DINO + Linear)**	**Top 1 accuracy (DINO + Naive kNN)**	**Top 1 accuracy (SOTA)**
Tiny Imagenet, CPU	81.66 / 77.73			-- / 92.98
Tiny Imagenet, GPU	81.66 / 77.73			-- / 92.98
Imagenet (10%), CPU	79.70 / 73.07			-
Imagenet (10%), GPU	79.70 / 73.07			-
Imagenet, GPU	83.47 / 77.77	– / 80.01	— / 78.30	-- / 91.10
Imagenet, CPU	83.47 / 77.77	– / 80.01	— / 78.30	-- / 91.10

Results

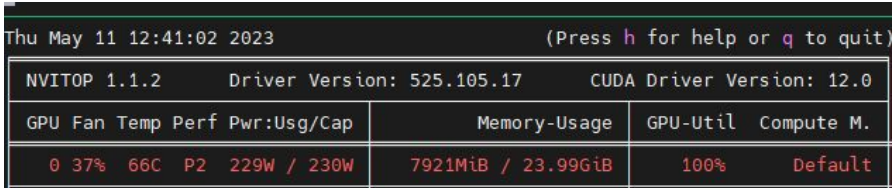
	Top 1 accuracy Train / Val (ours)	**Top 1 accuracy (DINO + Linear)**	**Top 1 accuracy (DINO + Naive kNN)**	**Top 1 accuracy (SOTA)**
Tiny Imagenet, CPU	81.66 / 77.73			-- / 92.98
Tiny Imagenet, GPU	81.66 / 77.73			-- / 92.98
Imagenet (10%), CPU	79.70 / 73.07			-
Imagenet (10%), GPU	79.70 / 73.07			-
Imagenet, GPU	83.47 / 77.77	– / 80.01	— / 78.30	-- / 91.10
Imagenet, CPU	83.47 / 77.77	– / 80.01	— / 78.30	-- / 91.10

Results

	Top 1 accuracy Train / Val (ours)	Top 1 accuracy (DINO + Linear)	Top 1 accuracy (DINO + Naive kNN)	Top 1 accuracy (SOTA)
Tiny Imagenet, CPU	81.66 / 77.73			-- / 92.98
Tiny Imagenet, GPU	81.66 / 77.73			-- / 92.98
Imagenet (10%), CPU	79.70 / 73.07			-
Imagenet (10%), GPU	79.70 / 73.07			-
Imagenet, GPU	83.47 / 77.77	– / 80.01	— / 78.30	-- / 91.10
Imagenet, CPU	83.47 / 77.77	– / 80.01	— / 78.30	-- / 91.10

Some interesting facts

FAISS is almost perfectly optimized. There is 100% GPU utilization through the process (CPU Memory reads are nicely scheduled).



References

- Jégou, H., Douze, M., Johnson, J., Hosseini, L. and Deng, C., 2022. **Faiss: Similarity search and clustering of dense vectors library**. Astrophysics Source Code Library, pp.ascl-2210.
- Le, Y. and Yang, X., 2015. **Tiny imagenet visual recognition challenge**. 0CS 231N, 7(7), p.3.
- Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M. and Berg, A.C., 2015. **Imagenet large scale visual recognition challenge**. International journal of computer vision, 115, pp.211-252.
- Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P. and Joulin, A., 2021. **Emerging properties in self-supervised vision transformers**. In Proceedings of the IEEE/CVF international conference on computer vision (pp. 9650-9660).

Remaining work / Future directions

- Benchmark other models (with higher embedding dimensions).
- Benchmark for different k in kNN search.
- Benchmark for multi GPU similarity search.
- Do memory benchmark – Scalene [Issue with python interfaces]

- Issue: Clusters are busy because of NeurIPS deadline.

Summary , Q+A

- Similarity search
- NN Encodings
- Similarity search as a classification tool.
  - Accuracy
  - Performance.
- Observations:
  - NN encoding + similarity search is “promising” (still not SOTA).
  - The similarity search time is not a big issue for small datasets.
  - Parallelization and GPU acceleration is crucial for larger datasets.
- Future directions