COLUMBIA UNIVERSITY
IN THE CITY OF NEW YORK

Robust and Compact Deep Learning Features for Image Matching and Retrieval

Shih-Fu Chang

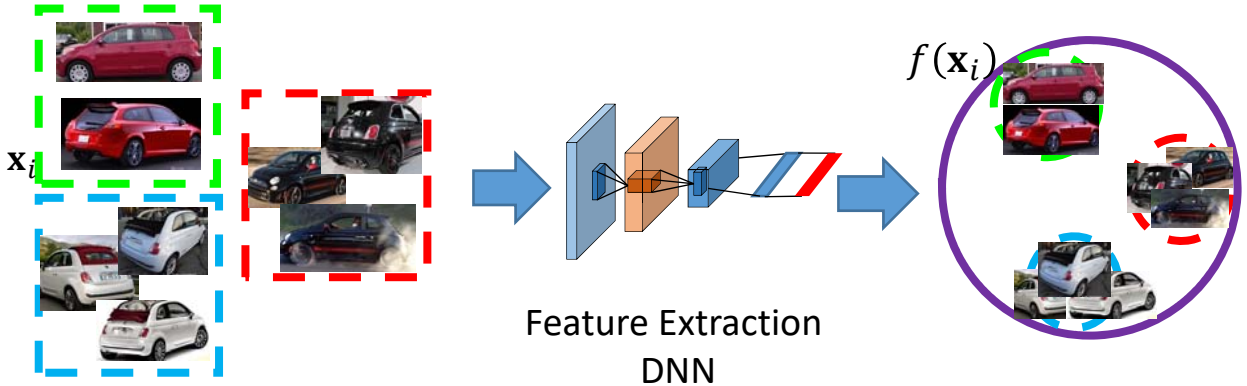
With Xu Zhang and Svebor Karaman

ECCV 2018 Invited Talk

COLUMBIA | ENGINEERING
The Fu Foundation School of Engineering and Applied Science

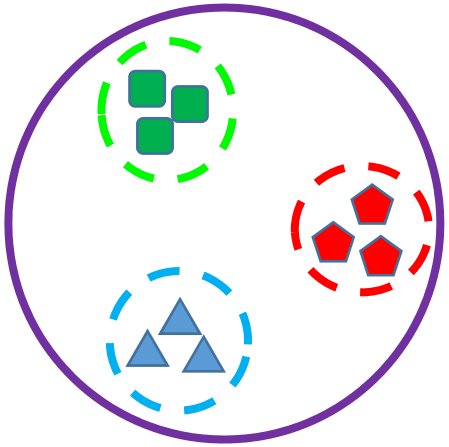
Feature Extraction

- A function (DNN) mapping an image (patch) to a feature vector
- Desirable properties
  - Concentrated: Samples from the same class to be close
  - Spread-out: Samples from different classes spread out in space
  - Compact: Easy to store and process



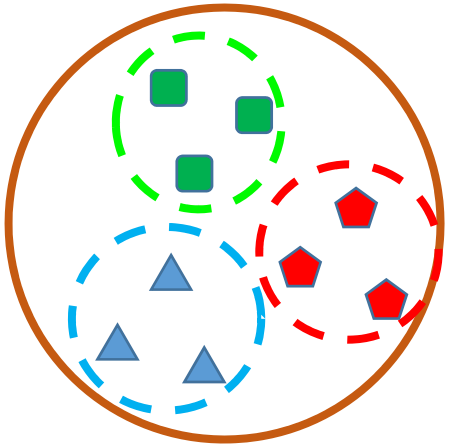
Good Feature and Bad Feature

"Good"
Feature



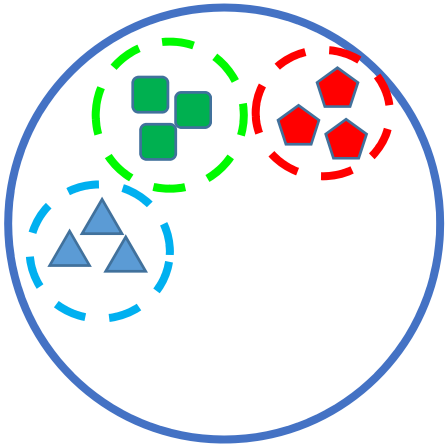
Concentrated
and
Spread-out

Non-Concentrated
Feature



Spread-out
not
Concentrated

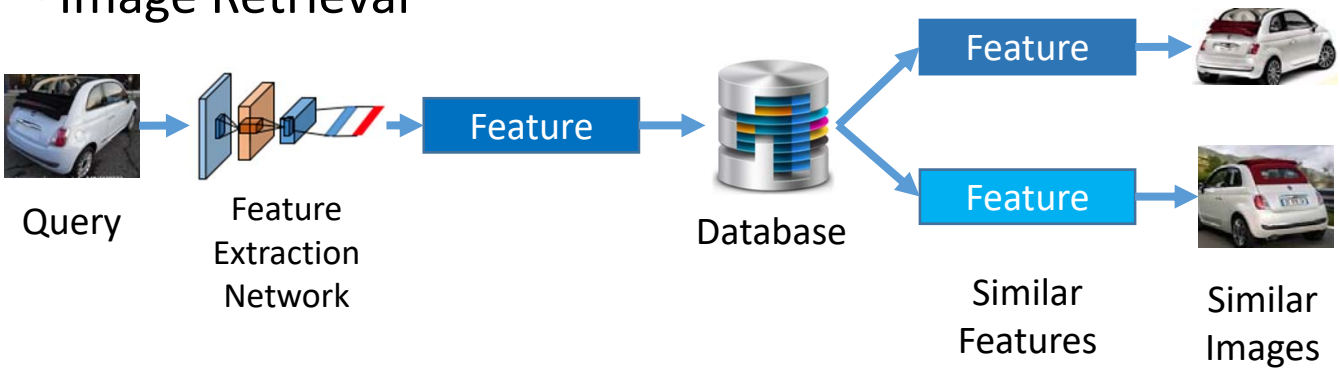
Non-Spread-out
Feature



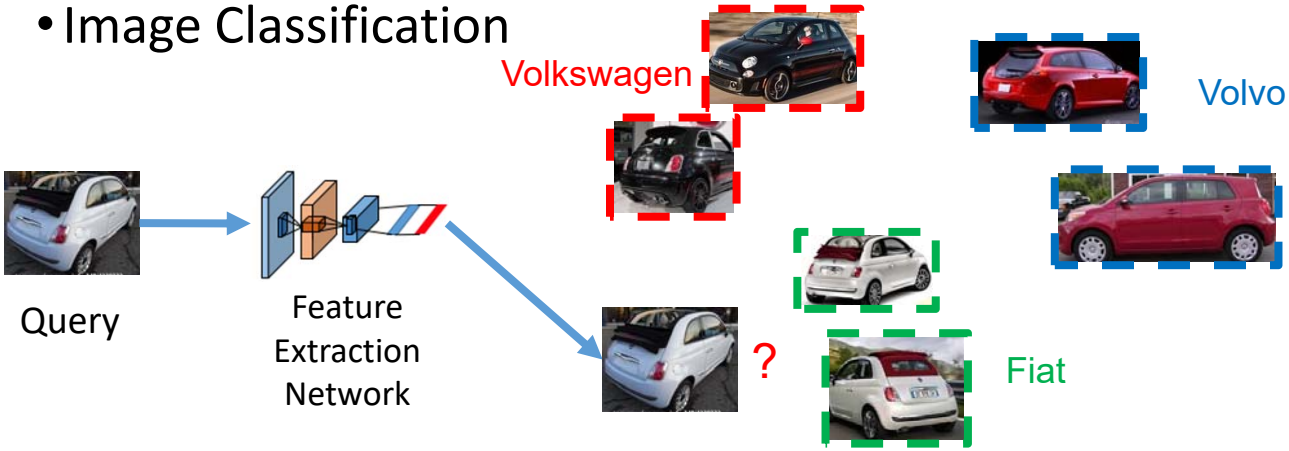
Concentrated
not
Spread-out

Applications

- Image Retrieval



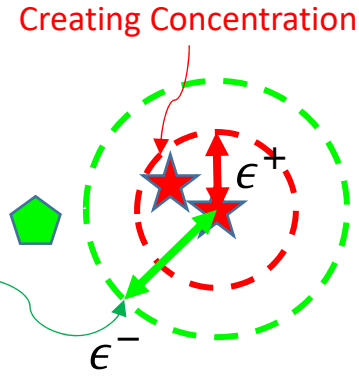
- Image Classification



Familiar Approaches

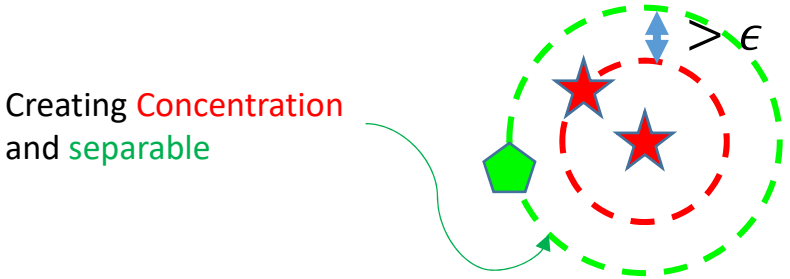
- Contrastive loss (pairwise) [Chopra CVPR15]

$$1)\ell\left(\mathbf{x}_i, \mathbf{x}_i^+\right) = \max\left(0, \| f(\mathbf{x}_i) - f(\mathbf{x}_i^+) \|_2 - \epsilon^+\right)$$
$$2)\ell\left(\mathbf{x}_i, \mathbf{x}_i^-\right) = \max(0, \epsilon^- - \| f(\mathbf{x}_i) - f(\mathbf{x}_i^-) \|_2)$$



- Triplet loss [Hoffer IWSBPR15]
  - $\ell_{tri}\left(\mathbf{x}_i, \mathbf{x}_i^+, \mathbf{x}_i^-\right) =$

$$\max\left(0, \epsilon - (\| f(\mathbf{x}_i) - f\left(\mathbf{x}_i^-\right) \|_2 - \| f(\mathbf{x}_i) - f(\mathbf{x}_i^+) \|_2)\right)$$



Limitation

- Consider local relations only

- Complexity
  - Large number of pairs or triplets: O(n^2)
  - Sensitive to sampling methods

- Idea: consider global properties in feature space

COLUMBIA UNIVERSITY
IN THE CITY OF NEW YORK

Learning Spread-Out Local Feature Descriptor

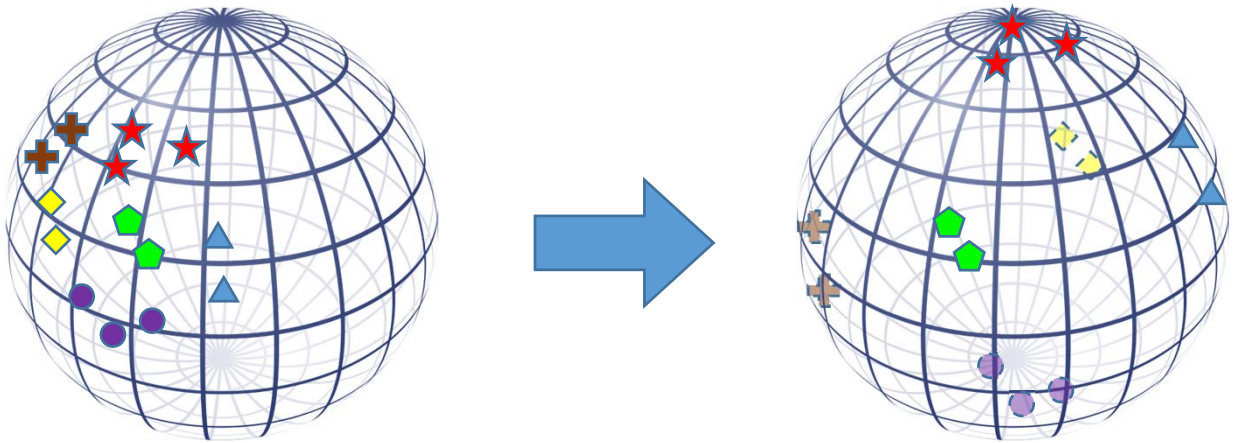
Xu Zhang*, Felix Xinnan Yu†, Sanjiv Kumar†, Shih-Fu Chang*

ICCV 2017

COLUMBIA | ENGINEERING
The Fu Foundation School of Engineering and Applied Science

Motivation

- to spread out non-matching points on unit sphere S^d (Considering the whole space)
- Spread-out helps utilize the full space
- Learn distribution of the non-matching descriptor to be close to uniform distribution on unit sphere



Not Spread-Out (Bad)

Spread-Out (Good)

Properties of Uniform Distribution on Sphere

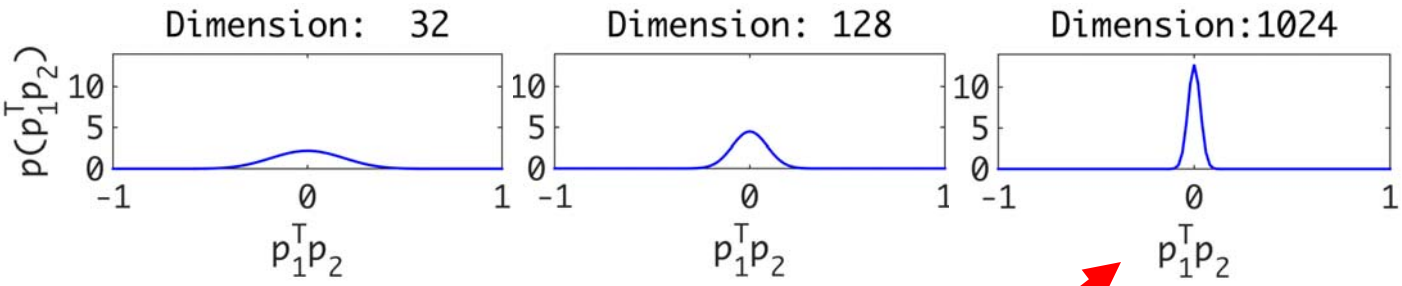
$$\mathrm{E}(\mathbf{p}_1^{\mathrm{T}}\mathbf{p}_2) = \mathrm{E}(\mathbf{p}_1^{\mathrm{T}})\,\mathrm{E}(\mathbf{p}_2) = 0$$

Mean

$$\mathrm{E}((\mathbf{p}_1^{\mathrm{T}}\mathbf{p}_2)^2) = \frac{1}{d}$$

2nd Order Moment

Probability density of $\mathbf{p}_1^T\mathbf{p}_2$, with different dimensionalities



With high dimensionality,
$\mathbf{p}_1, \mathbf{p}_2$ has high probability to be close to orthogonal!!

Global Orthogonal Regularization

$$\ell_{GOR} = (\frac{1}{N}\sum_{i=1}^{N} f(\mathbf{x}_i)^{\mathrm{T}} f(\mathbf{x}_i^-))^2 + \max(\frac{1}{N}\sum_{i=1}^{N}(f(\mathbf{x}_i)^{\mathrm{T}} f(\mathbf{x}_i^-))^2 - \frac{1}{d}, 0))$$

Mean of Non-Matching Pairs

$\mathrm{E}(\mathbf{p}_1^{\mathrm{T}}\mathbf{p}_2) = 0$

2nd Order Moment of Non-Matching Pairs

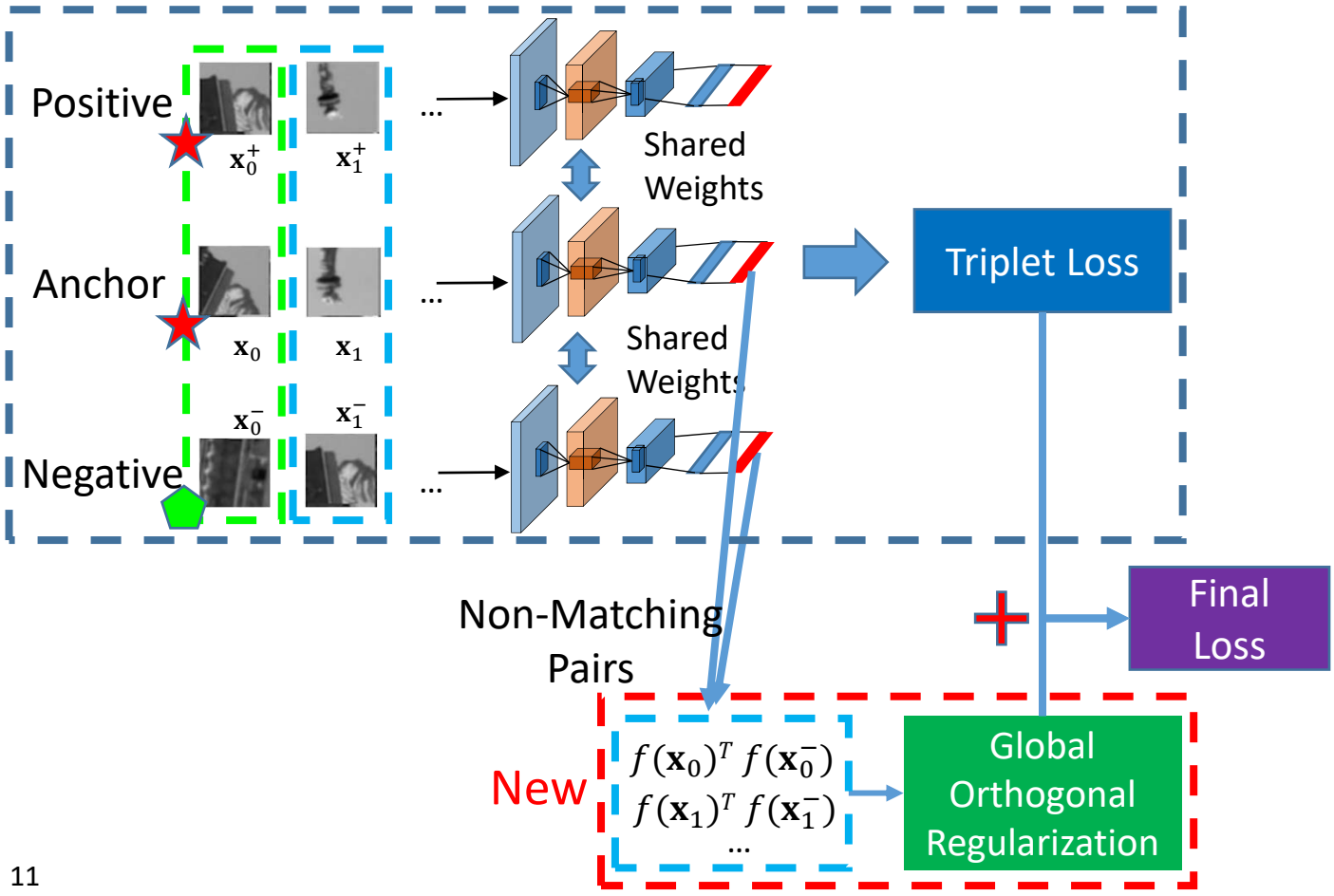
$\mathrm{E}((\mathbf{p}_1^{\mathrm{T}}\mathbf{p}_2)^2) = \frac{1}{d}$

Final Loss

$$\ell_{*\_GOR} = \ell_* + \alpha \ell_{GOR}$$

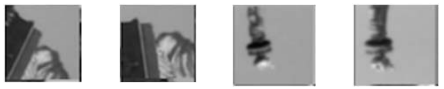
Trade-off parameter

Training: add spread-out constraint to existing networks



Experiment

Patch samples in UBC patch dataset



- Training
  - Local descriptor benchmark, UBC patch dataset [Brown PAMI11]
    - three sets: Yosemite, Notre Dame, and Liberty.
    - Each set has more than 450k local image patches
    - Sampled the output of Difference of Gaussians (DoG) detector
    - Each patch has a size of 64*64.
  - Randomly sample 1M triplets from training set for training. Run 10 epochs.
- Testing
  - 100k testing pairs
- Metric
  - FPR95: False Positive Rate at 95% True Positive Rate.

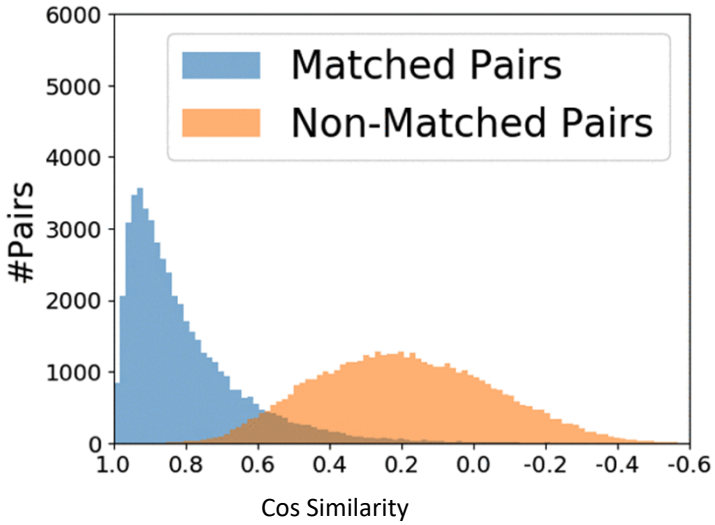
UBC patch dataset

FPR95(%)

	Training		Not	Lib	Not	Yos	Yos	Lib	Mean
	Testing		Yos		Lib		Not		
	Descriptor	#dim							
	SIFT[Lowe IJCV04]	128	27.29		29.84		22.53		26.55
Pairwise	MatchNet [Han CVPR15]	512	11	13.58	8.84	13.02	7.7	4.75	9.82
	DeepDesc [Simo-Serra ICCV15]	128	16.19		8.82		4.54		9.85
	Pairwise + GOR	128	**6.88**	**6.99**	**6.46**	**8.33**	**3.73**	**3.40**	**5.97**
Triplet	Triplet Loss [Balntas, BMVC16]	256	7.95	8.10	7.64	9.88	3.83	3.39	6.79
	Triplet Loss + AS + GOR	128	**5.15**	**5.40**	**4.80**	**6.45**	**2.38**	**1.95**	**4.35**
Hard Mining	HardNet [Anastasiya, NIPS 2017]	128	2.82	3.00	2.17	3.86	1.28	0.78	2.32
	HardNet + GOR	128	**2.53**	**2.85**	**2.01**	**3.52**	**1.21**	**0.76**	**2.15**

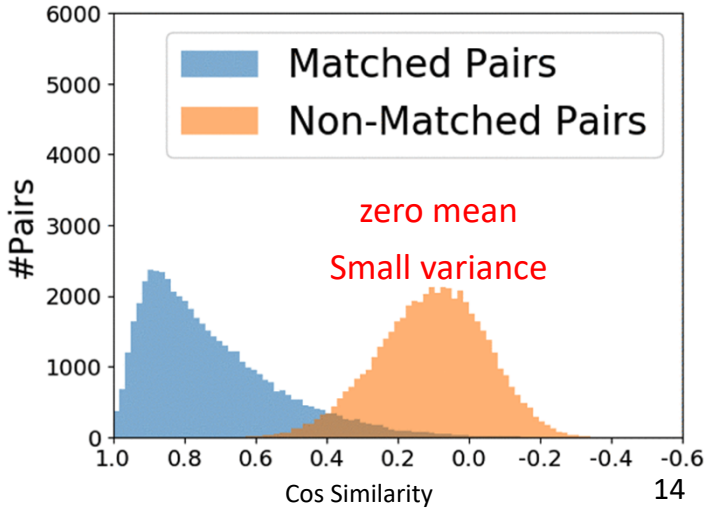
Are features more spread-out?

Baseline



- Train: Yosemite
  Test: Liberty

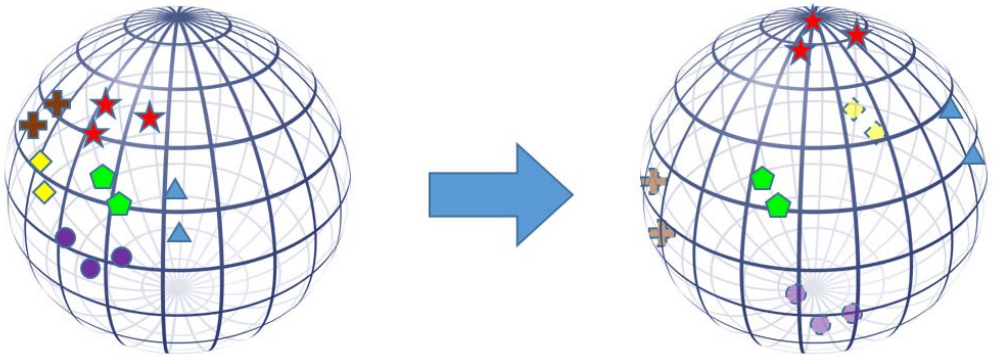
Baseline + GOR



Ideally $\mathrm{E}\left(\mathbf{p}_1^{\mathrm{T}}\mathbf{p}_2\right) = 0$

Summary #1 – spread-out constraint

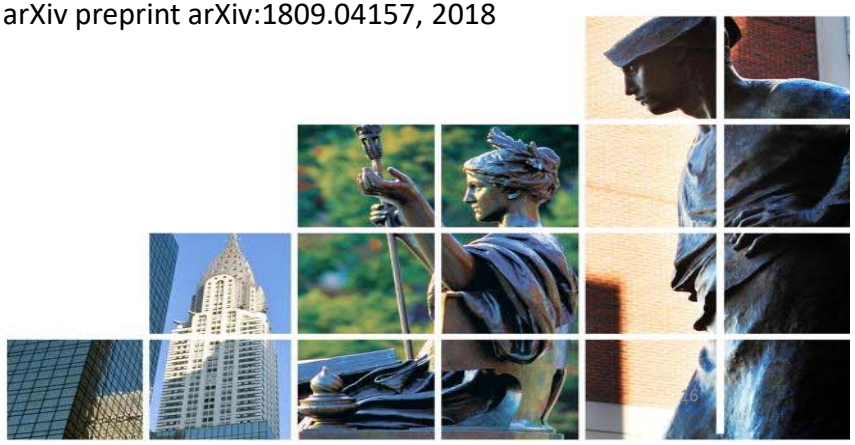
- A simple Global Orthogonal Regularization constraint makes features more spread out over the entire space
- Can be easily added to other metrics like pairwise, triplet and hard mining



COLUMBIA UNIVERSITY
IN THE CITY OF NEW YORK

Making Classifier Features Spread-out and Concentrated

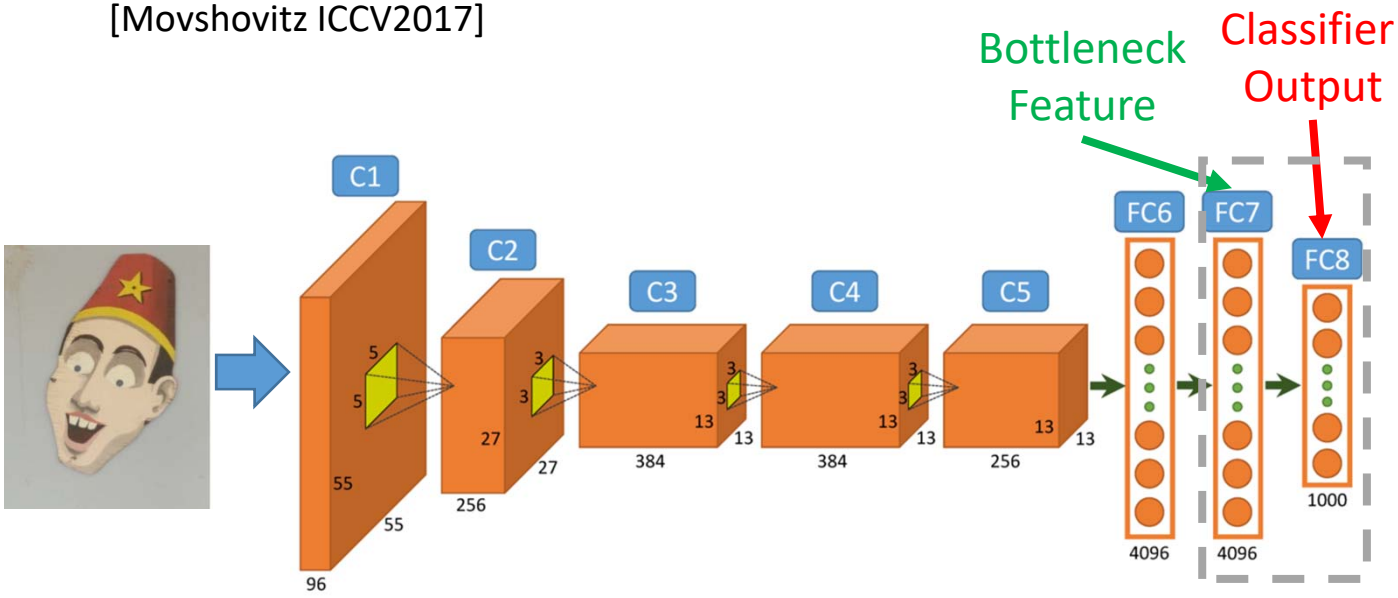
"Heated-Up Softmax Embedding"
X. Zhang, F. Yu, S. Karaman, W. Zhang, SF Chang - arXiv preprint arXiv:1809.04157, 2018



COLUMBIA | ENGINEERING
The Fu Foundation School of Engineering and Applied Science

Classifier Based Descriptor

- The second to the last layer (bottleneck) of the DNN classifier has been widely used.
- Showed strong performance in recent literatures [Movshovitz ICCV2017]



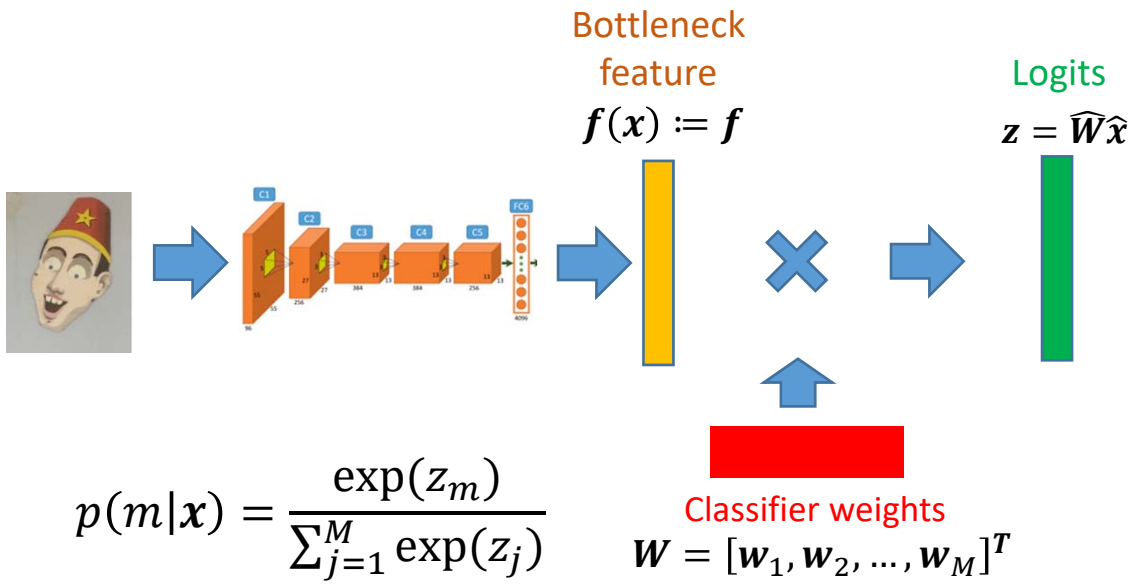
Structure of AlexNet

Classifier

Image Credit Hu et.al, 2015

Classifier Feature

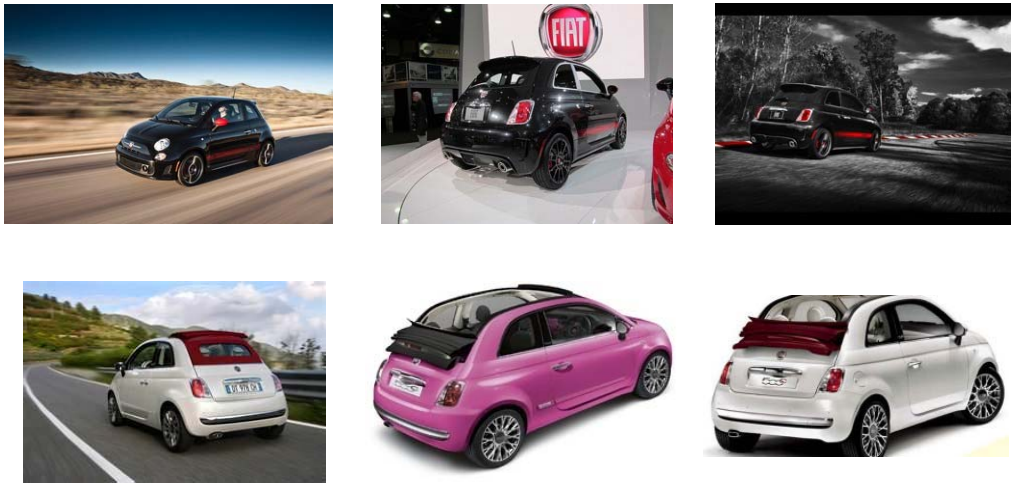
- A popular structure of DNN classifier trained with softmax function and cross entropy loss



$$p(m|x) = \frac{\exp(z_m)}{\sum_{j=1}^{M} \exp(z_j)}$$

Benchmark

- Dataset
  - **Stanford Car Dataset (Car196):** fine-grained car category dataset, which contains 16,185 images of 196 car models. First 98 categories of 8,054 images for training, while the other 98 categories of 8,131 images are used for test.



Classifier Based Descriptor

- Showed strong performance in recent literature
  [Razavian CVPRW2014, Movshovitz ICCV2017]

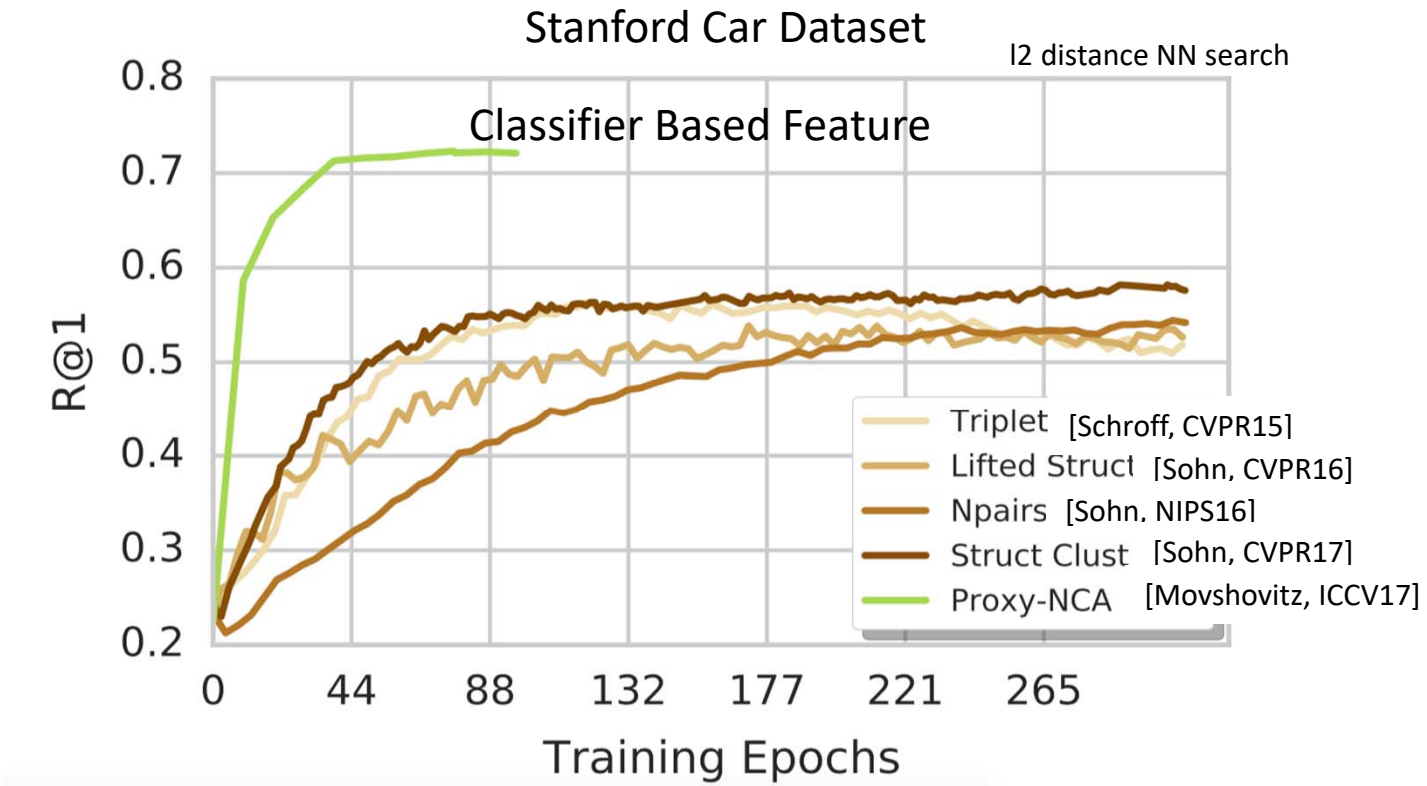
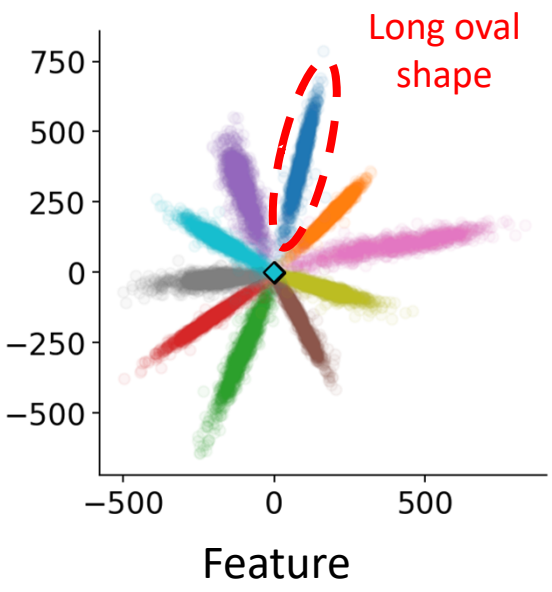
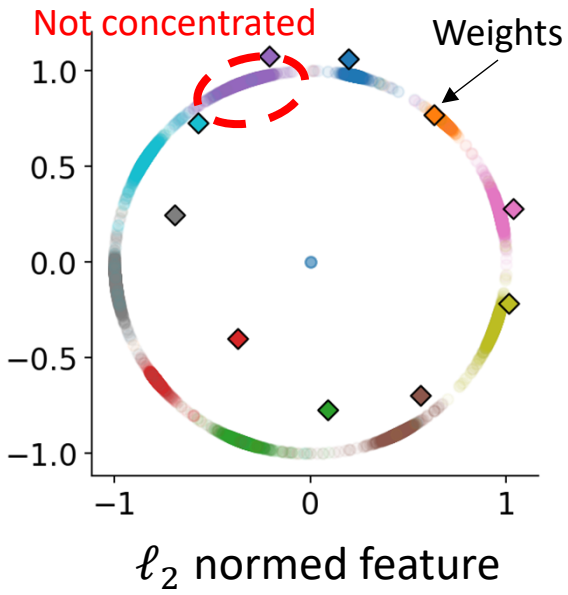


Image Credit Movshovitz, 2017

Distribution of Existing Classifier Features Still not Desirable

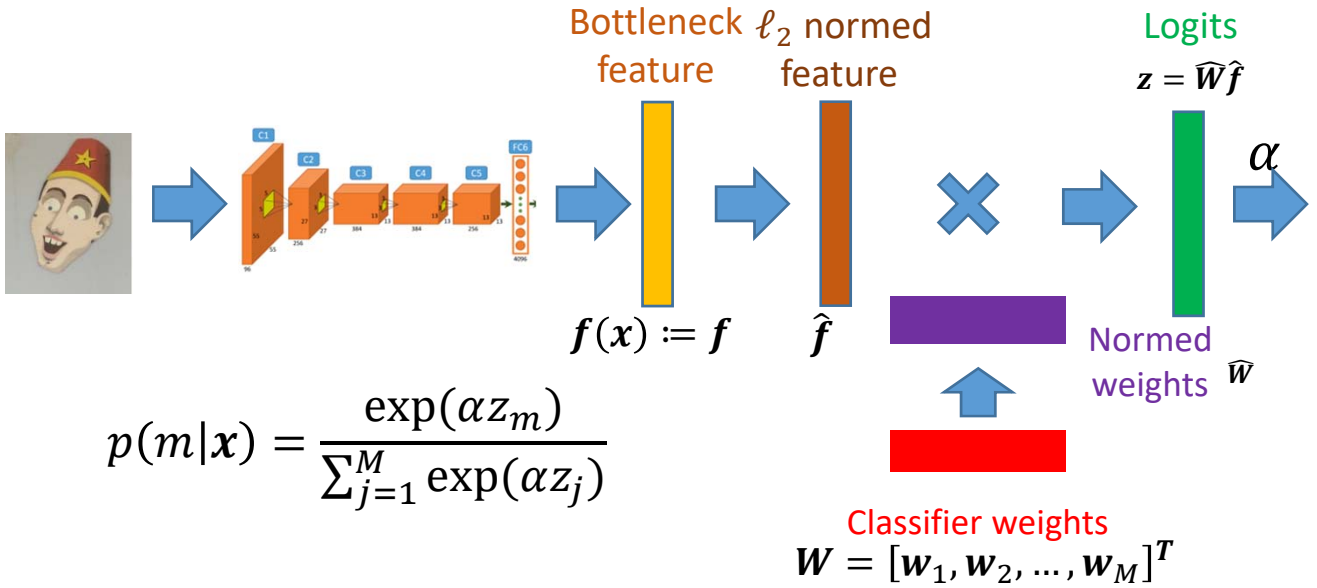
- Feature trained on MNIST, each color for one digit
- Observation: current classifier features are still not concentrated, even after l2 normalization.





Deeper Analysis of Training Pipeline

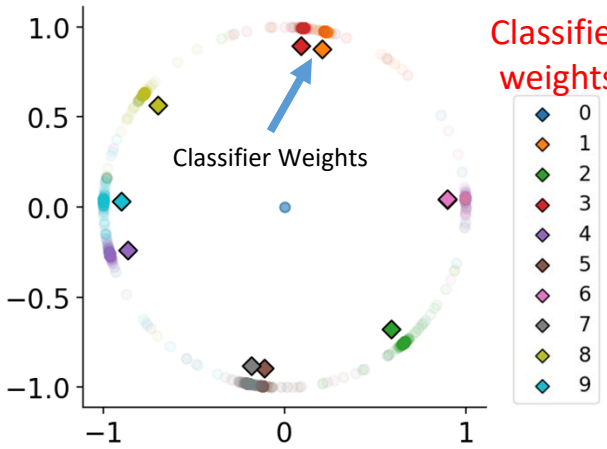
- Applying ℓ_2 normalization to both feature and weight shows strong performance in face verification and image retrieval [Wang MM 2017, Liu CVPR 2017]
- Use temperature parameter α for gradient tuning



$$p(m|x) = \frac{\exp(\alpha z_m)}{\sum_{j=1}^{M} \exp(\alpha z_j)}$$

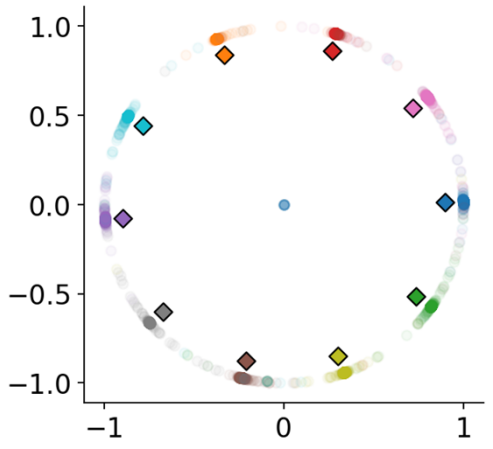
Temperature parameter α in Softmax Function

- α (1/T) affects the distribution of the final features
- Idea of adjusting temperature during training
  (Hinton et al, 2015 NIPS-W)



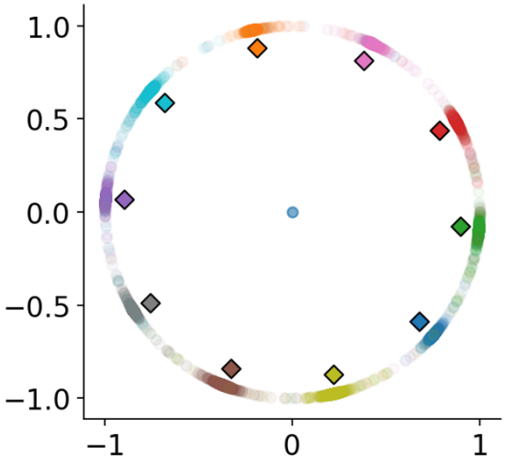
$\alpha = 0.25$

Classifier weights

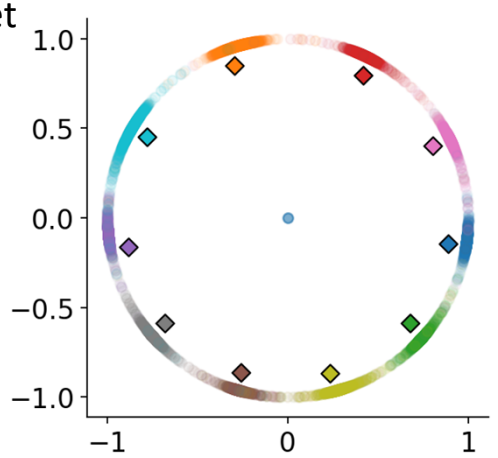


$\alpha = 4$

Feature trained with normalization in MNIST dataset



$\alpha = 16$



$\alpha = 64$

Understand Effect of Temperature parameter $\boldsymbol{\alpha}$

$$p(m|\boldsymbol{x}, \alpha) = \frac{\exp(\alpha z_m)}{\sum_{j=1}^{M} \exp(\alpha z_j)}$$

- For all samples

$$\lim_{\alpha \to +\infty} p(m|\mathbf{x}, \alpha) = \begin{cases} 1/K & z_m = \max(z_1, ..., z_M) \\ 0 & \text{otherwise,} \end{cases}$$

K is the number of the maximum logits

- For Correct samples (samples correctly classified by the classifier)

$$\lim_{\alpha \to +\infty} p(m|\mathbf{x}, \alpha) = \begin{cases} 1 & m = y \\ 0 & \text{otherwise,} \end{cases}$$

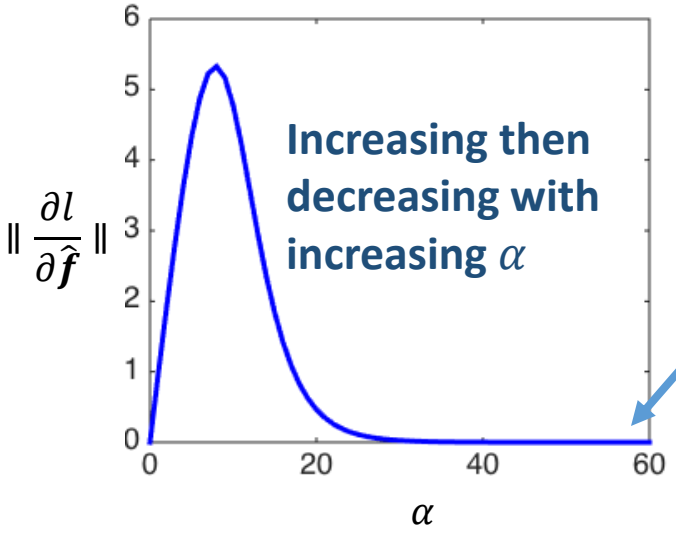
Gradient in Classifier Training (Correct Samples)

- gradient with respect to descriptor

$$\| \frac{\partial l}{\partial \hat{\boldsymbol{f}}} \| = \| \sum_{m=1}^{M} \alpha\big(p(m|\boldsymbol{x},\alpha) - q(m|\boldsymbol{x})\big)\hat{\boldsymbol{w}}_{\boldsymbol{m}} \|$$

Ground-Truth distribution
one-hot

For Correct Sample



$$\lim_{\alpha \to +\infty} \| \frac{\partial l}{\partial \hat{\boldsymbol{f}}} \| = 0$$

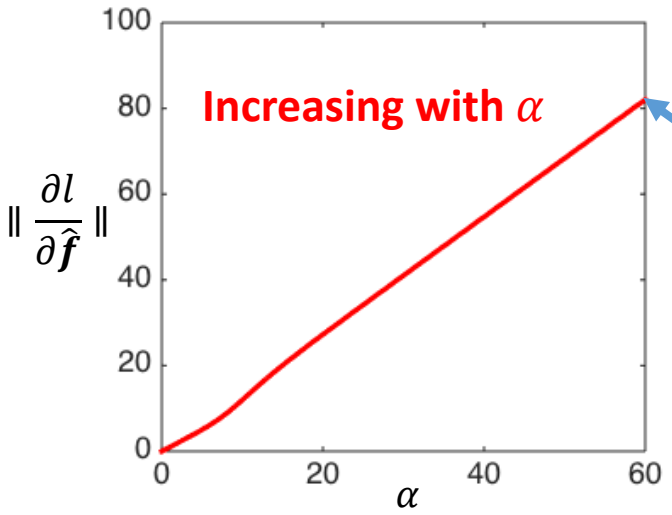
Gradient in Classifier Training (Incorrect Samples)

- gradient with respect to descriptor

$$\| \frac{\partial l}{\partial \hat{\boldsymbol{f}}} \| = \| \sum_{m=1}^{M} \alpha \big(p(m|\boldsymbol{x}, \alpha) - q(m|\boldsymbol{x}) \big) \hat{\boldsymbol{w}}_{\boldsymbol{m}} \|$$

Ground-Truth distribution
one-hot

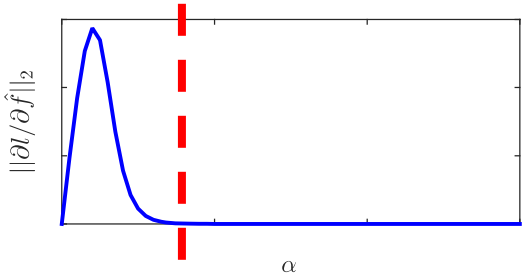
For Incorrect Sample



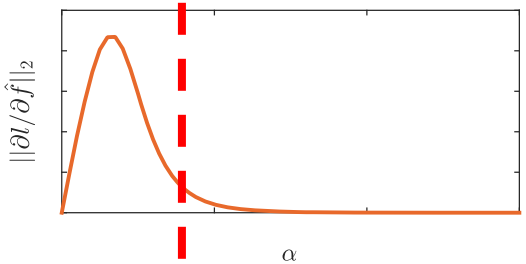
$$\lim_{\alpha \to +\infty} \| \frac{\partial l}{\partial \hat{\boldsymbol{f}}} \| = +\infty$$

Initially train with Intermediate α

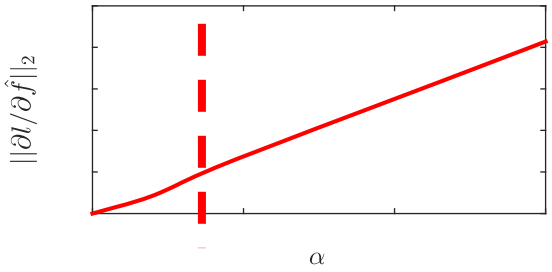
- For Correct Centroid Sample



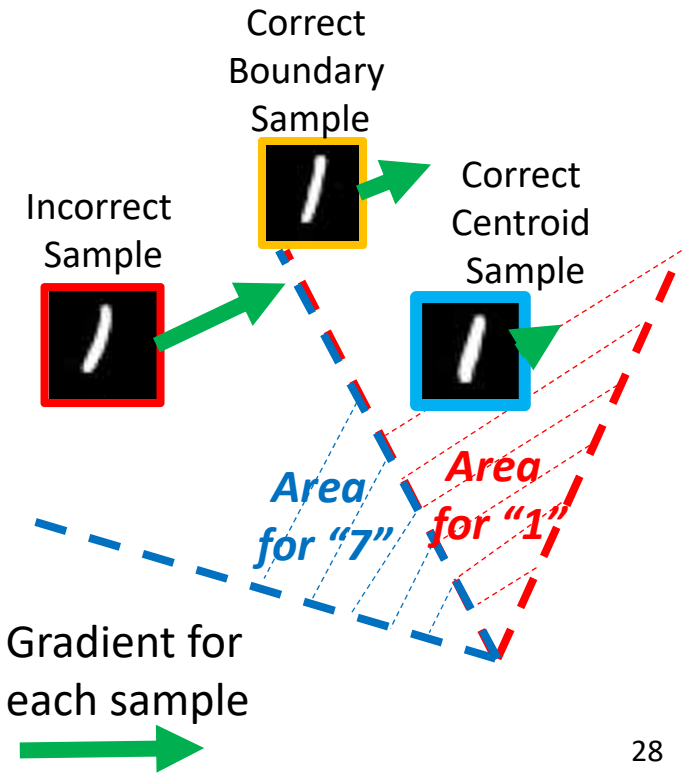
- For Correct Boundary Sample



- For Incorrect Sample

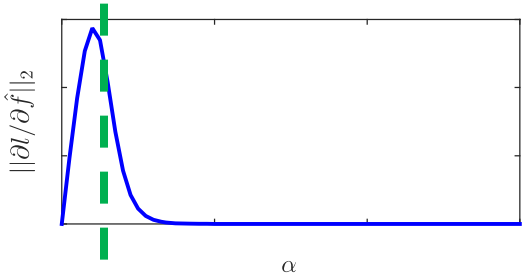


- Start with intermediate α to assign large gradient to incorrect sample and medium gradient to boundary sample.

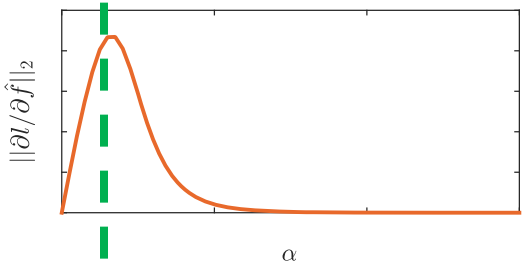


Then use heated-up α

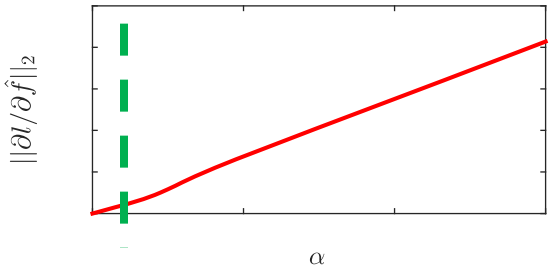
- For Correct Centroid Sample



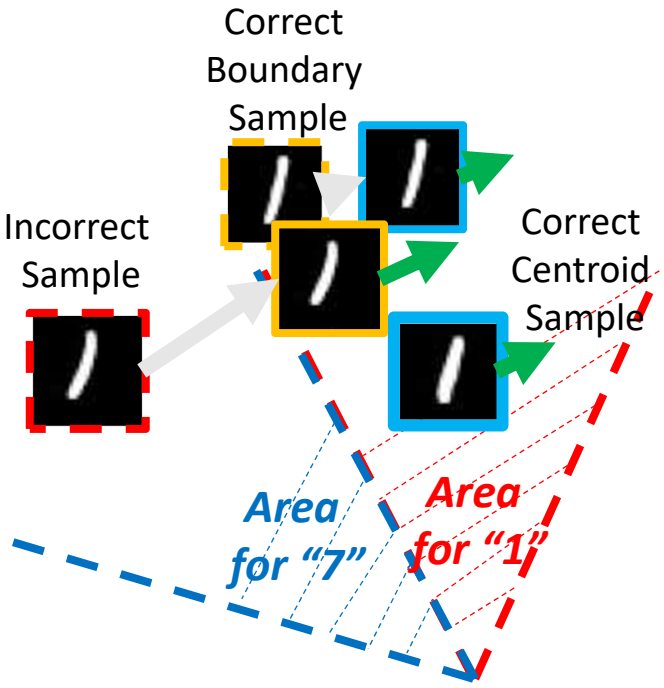
- For Correct Centroid Sample



- For Incorrect Sample



- Then use small α to assign large gradient to all the samples to further compress distribution.



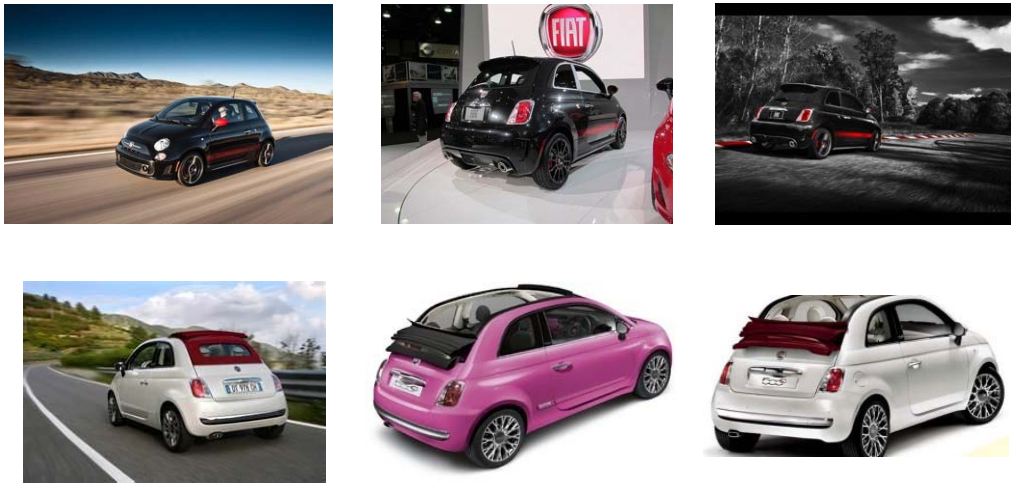
Heating up the Feature Training Process

- Training Strategy
  - use intermediate α value to start the training process
- Heated-up Feature
  - When most of the training samples are correctly classified, using small α value (higher temperature) to get a more concentrated descriptor.
- Batch Normalization instead of l2 normalization
  - Batch normalization adds more variance and help spread out features

"Heated-Up Softmax Embedding"
X. Zhang, F. Yu, S. Karaman, W. Zhang, SF Chang - arXiv preprint arXiv:1809.04157, 2018

Experiment

- Dataset
  - **Stanford Car Dataset (Car196):** fine-grained car category dataset, which contains 16,185 images of 196 car models. First 98 categories of 8,054 images for training, while the other 98 categories of 8,131 images are used for test.



Car196

	METHOD	DIM	NMI	R@1	R@2	R@4
[Schroff, CVPR15]	TRIPLET S.H.	128	53.35	51.54	63.78	73.52
[Sohn, CVPR16]	LIFTED	128	56.88	52.98	66.70	76.01
[Sohn, NIPS16]	NPAIRS	64	57.79	53.90	66.76	77.75
[Sohn, CVPR17]	STRUCT C.	64	54.44	58.11	70.64	80.27
[Law, ICML17]	D.C.	98	61.12	67.54	77.77	85.74
[Harwood, ICCV17]	FANNG	64	59.50	64.65	76.20	84.23
[Movshovitz, ICCV17]	PROXY NCA	64	64.90	73.22	82.42	86.36
	SOFTMAX	64	59.52	60.76	73.58	82.50
l2 normalization	S.M.+LN	64	62.40	68.59	78.55	86.18
Batch norm	S.M.+BN	64	65.81	71.12	80.62	87.82
l2 normalization Heated-up	S.M.+HLN	64	66.87	71.93	81.68	88.34
Batch norm Heated-up	S.M.+HBN	64	**68.10**	**74.70**	**83.90**	**89.77**

Qualitative Result

Query

Top 3 returned images



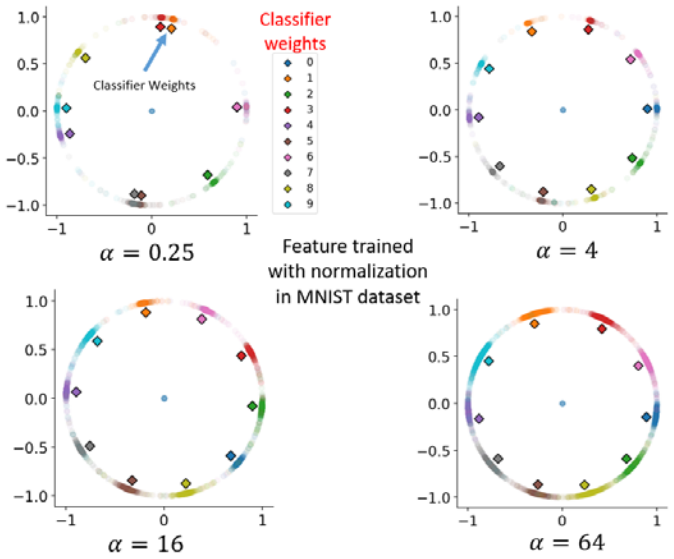







Summary #2 – temperature control

- Proper heat-up procedure (intermediate alpha → small alpha) makes features more spread out and concentrated



COLUMBIA UNIVERSITY
IN THE CITY OF NEW YORK

Compact features for large-scale indexing

Svebor Karaman, Xudong Lin, and Shih-Fu Chang

COLUMBIA | ENGINEERING
The Fu Foundation School of Engineering and Applied Science

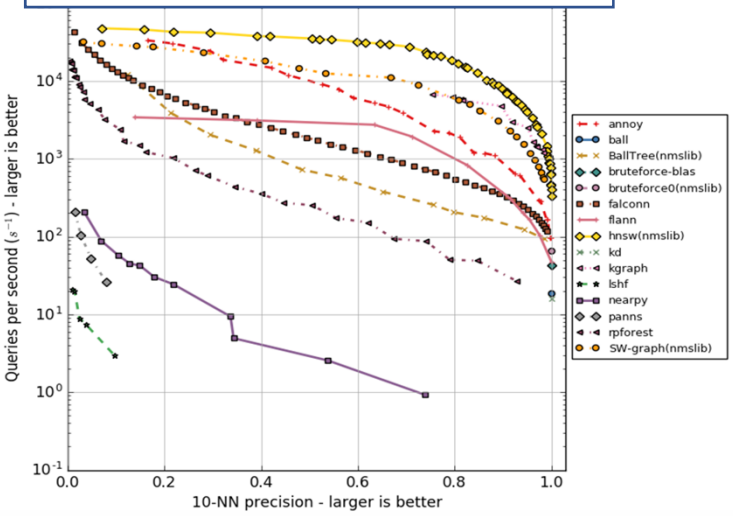
COLUMBIA UNIVERSITY
IN THE CITY OF NEW YORK

DVMM lab
digital video | multimedia laboratory

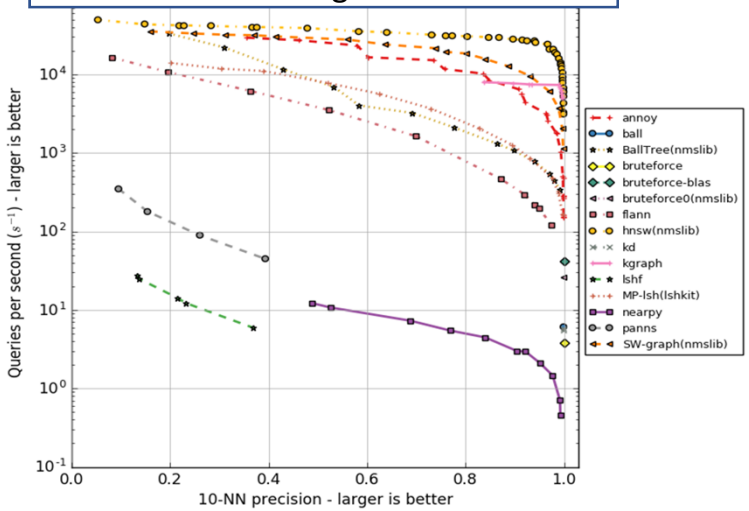
Need of Logarithmic Search

- Billion scale datasets are now common, exhaustive search unacceptable
- Standard ANN method (e.g. tree based) achieve limited speed/accuracy
- New graph based indexing dominates recent benchmarks https://github.com/erikbern/ann-benchmarks

1.19M Glove wordvectors



1M SIFT image features



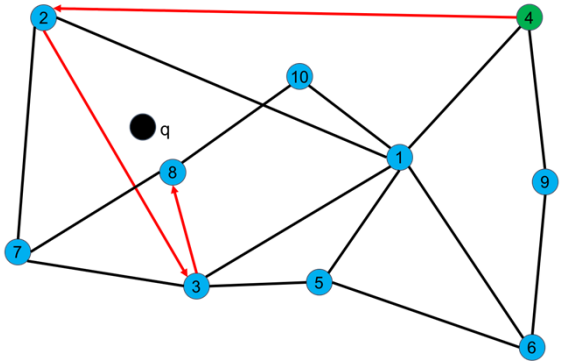
Navigable Small World: Overview

Navigable Small World Graph [Malkov14]

- Any vertex reachable from any other vertex in a few hops
- Typical distance between two random nodes: $L \propto \log(N)$

NN "greedy search"

- Given a query, select a random entry point
- Search over neighbors of current point
- Go to closest 'neighbor' and iterate
- Multiple searches or backtracking for K-NN



[Malkov14] Y. Malkov, A. Ponomarenko, A. Logvinov, V. Krylov. "Approximate nearest neighbor algorithm based on navigable small world graphs", Information Systems, 2014.

Navigable Small World: Construction Process

Sequential Insertion (illustration for 3-NN graph)

Navigable Small World: Construction Process

Edges in the graph serve two purposes:

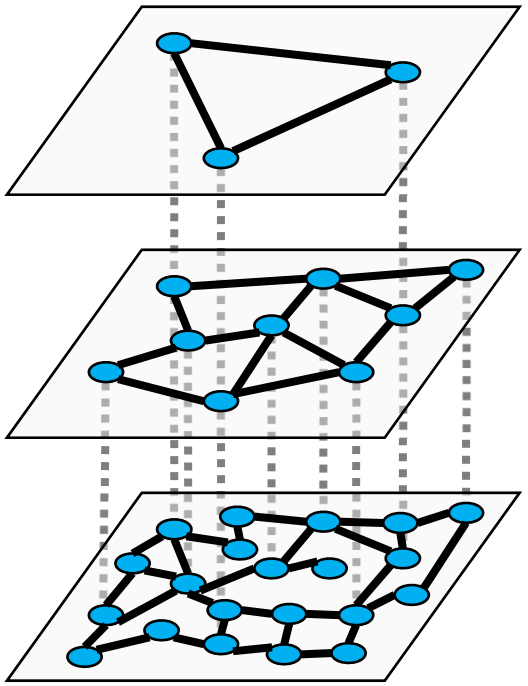
- **short-range links**: created in later stages, connections to a sample's nearest neighbors
- **long-range links**: created in earlier stages, responsible for the small world navigation property of the graph

- Empirically, hop distance between two random nodes: $L \propto \log(N)$
- Any vertex reachable from any other vertex in a few hops

Hierarchical Navigable Small World

Hierarchical NSW (HNSW) [Malkov16]

- Exponentially decaying probability of sample insertion in higher layers
- Separate long-range and short-range links in different layers
- Node degree bounded in each layer
- Edges diversification with occlusion rule
- hop distance between two random nodes: $L \propto \log(N)$

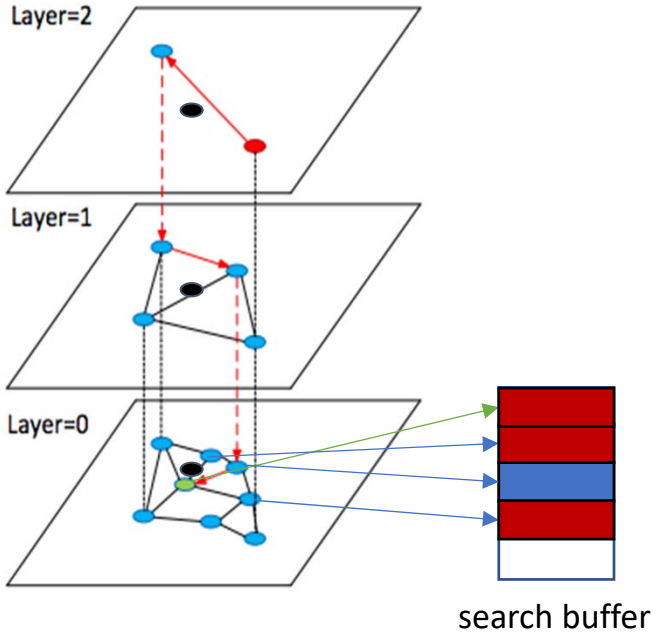


[Malkov16] Y. Malkov, D. A. Yashunin. "Efficient and robust approximate nearest neighbor search using Hierarchical Navigable Small World graphs." *arXiv preprint arXiv:1603.09320*(2016).

Hierarchical Navigable Small World: Search

Top-down process:

- From top layer, in each layer, greedy search to find best NN
- Next (lower) layer, search starts from previous layer's best NN
- Bottom layer, push neighborhood of current best **unvisited** sample to a search buffer kept sorted, until all buffer samples are **visited**
- Node degree bounded in each layer → log(N) search complexity for 1-NN



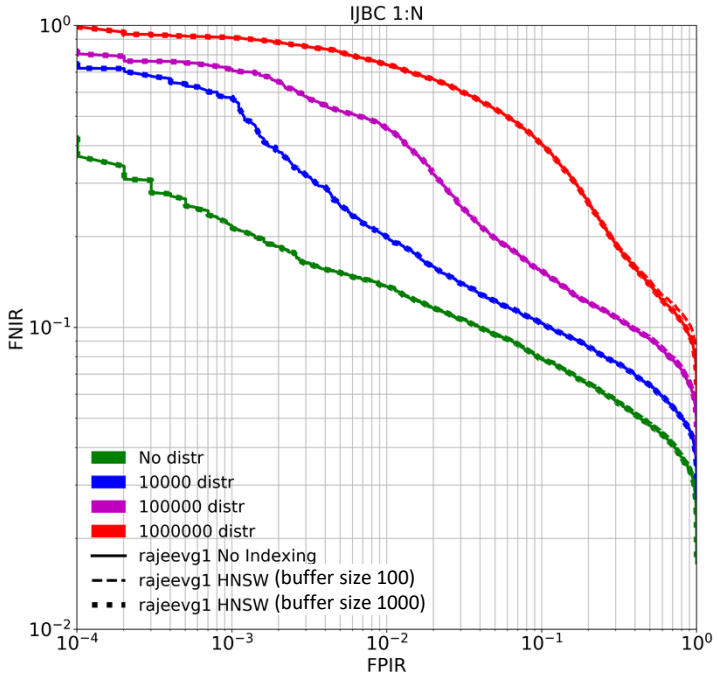
[Malkov16] Y. Malkov, D. A. Yashunin. "Efficient and robust approximate nearest neighbor search using Hierarchical Navigable Small World graphs." *arXiv preprint arXiv:1603.09320*(2016).

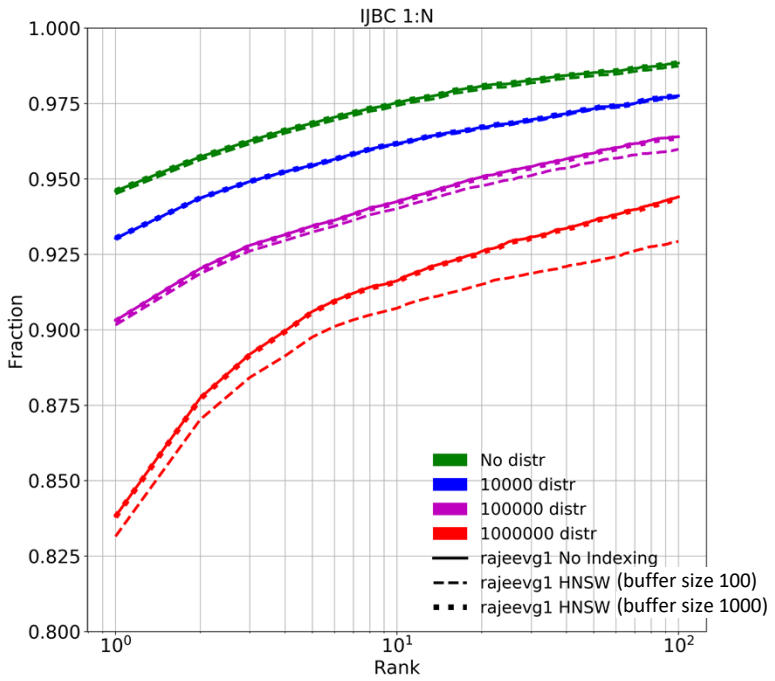
COLUMBIA UNIVERSITY
IN THE CITY OF NEW YORK

DVMM lab
digital video | multimedia laboratory

IJBC Face Search Performance Evaluation (up to 1M samples)

- IJBC: challenging unconstrained face recognition dataset
- **HNSW fully preserves accuracy of exhaustive search**



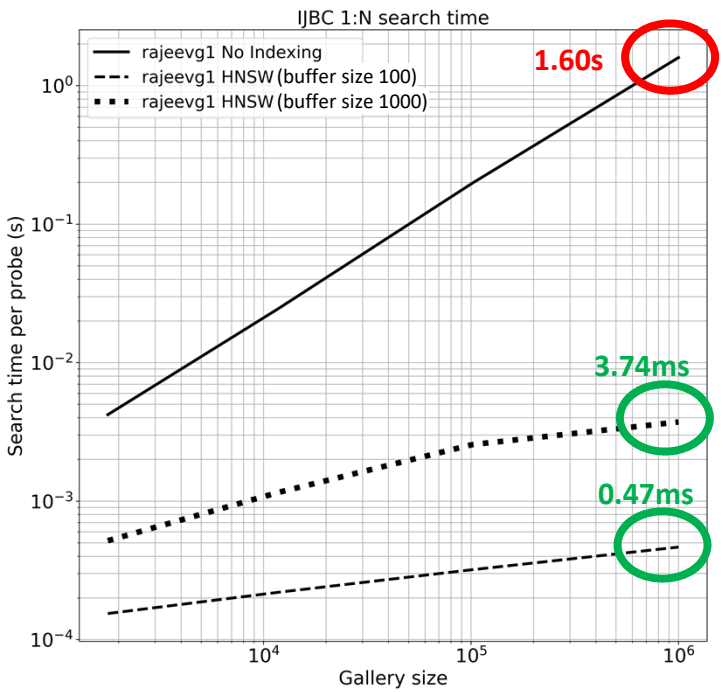


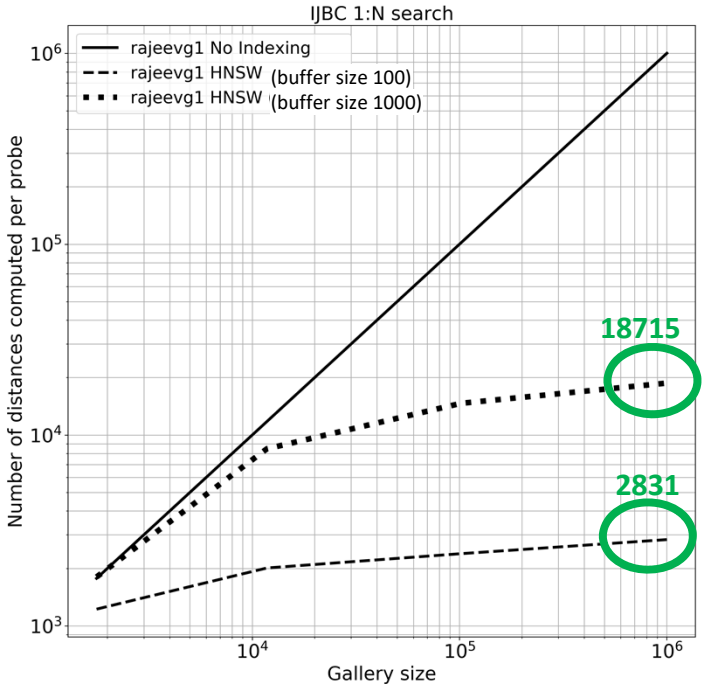
COLUMBIA UNIVERSITY
IN THE CITY OF NEW YORK

DVMMlab
digital video | multimedia laboratory

IJBC Face Search Speed Evaluation

- **Logarithmic search time and number of distances computed**





Graph based methods storage requirement problem

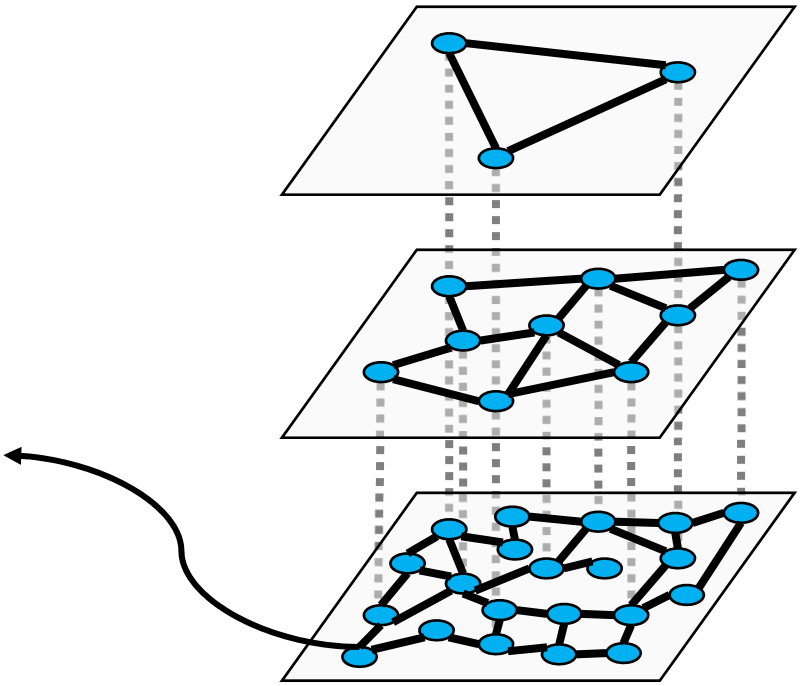
- Graph index: store **links** information and all **features** (no compression)
- Dataset of 1B samples:
  - Graph with 32 neighbors per samples: **119 GB** to store the links
  - 128 dimensional real-valued features (SIFT): 477 GB
  - 960 dimensional real-valued features (GIST): **3.5 TB**
  - 256 bits hash codes: **30 GB**
- Idea: compress feature vector by hashing
- Hashing useful for search from mobile

Compressing HNSW

- Can we reduce the required storage ?
  (original 128 or 960-dim floating point)

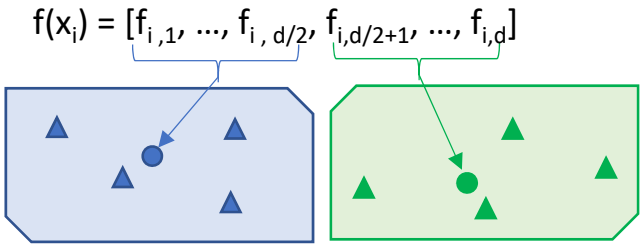
- Compress the samples' representation
  - Product quantization
  - Hashing

Compressed features, $c(f(x_i))$, **~100 bits**



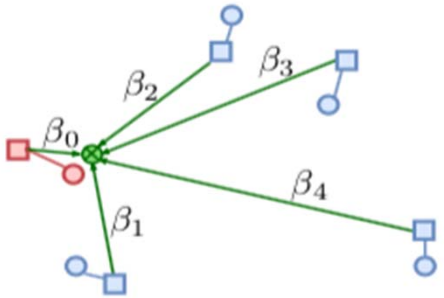
Compressing HNSW (1): Product Quantization

- Split space of d dimensions into 2 subspaces
- Train K-means in each subspace
- PQ code is the combination of the closest cluster id c_i in each subspace: $PQ(x_i) = \{c_k^{(1)}, c_j^{(2)}\}$



Link & Code [L&C]: two-stage strategy

1. Product Quantized codes for coarse search
2. Re-ranking with refined representation, considering sample's neighbors



[PQ] H. Jegou, M. Douze & C. Schmid, "Product quantization for nearest neighbor search" TPAMI, 2011.
[L&C] M. Douze, A. Sablayrolles & H. Jégou, "Link and code: Fast indexing with graphs and compact regression codes". CVPR'2018

Link & Code Experimental Results

- Memory budget 64 bytes (24 bytes for links, 40 bytes for features)
- Refined representation helps in some cases (M=8)
- Achieve 62.5% accuracy with 9.6X compression

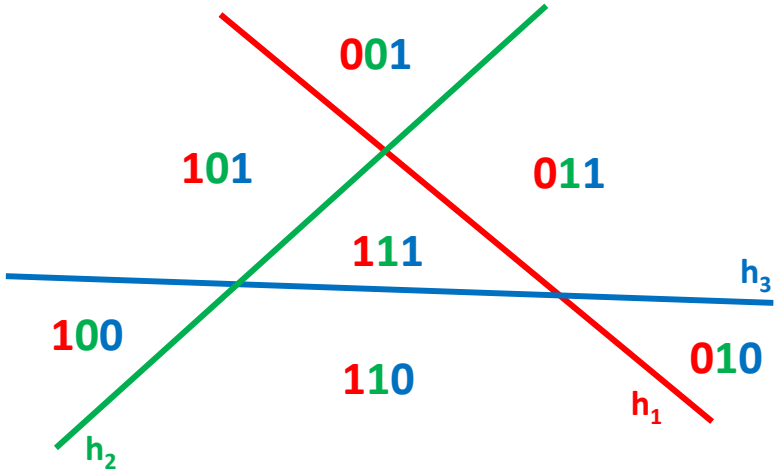
codec	vector quantization error ($\times 10^3$)		R@1 exhaustive		R@1 T=1024	R@1 T=16384
Deep:	100M	1B	100M	1B	100M	1B
L6&OPQ40	24.3	24.3	0.608	0.601	0.427	0.434
L6&OPQ40 M=0	22.7	22.5	0.611	0.600	0.429	0.435
L6&OPQ36 M=4	21.9	21.5	0.608	0.607	0.428	0.434
L6&OPQ32 M=8	20.0	19.8	0.625	0.612	0.438	0.438

Deep 1B (or subsampled to 100M)

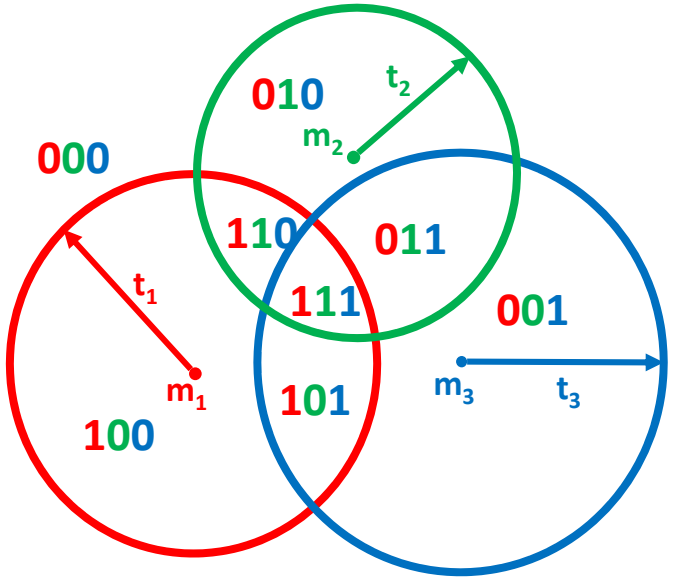
[OPQ] T. Ge, K. He, Q. Ke & J. Sun, "Optimized product quantization for approximate nearest neighbor search" CVPR'2013.
[L&C] M. Douze, A. Sablayrolles & H. Jégou, "Link and code: Fast indexing with graphs and compact regression codes". CVPR'2018

Compressing HNSW (2): Hashing

Hashing: binary representation $H(x_i)$ = [010…110]



Linear hashing, e.g., LSH



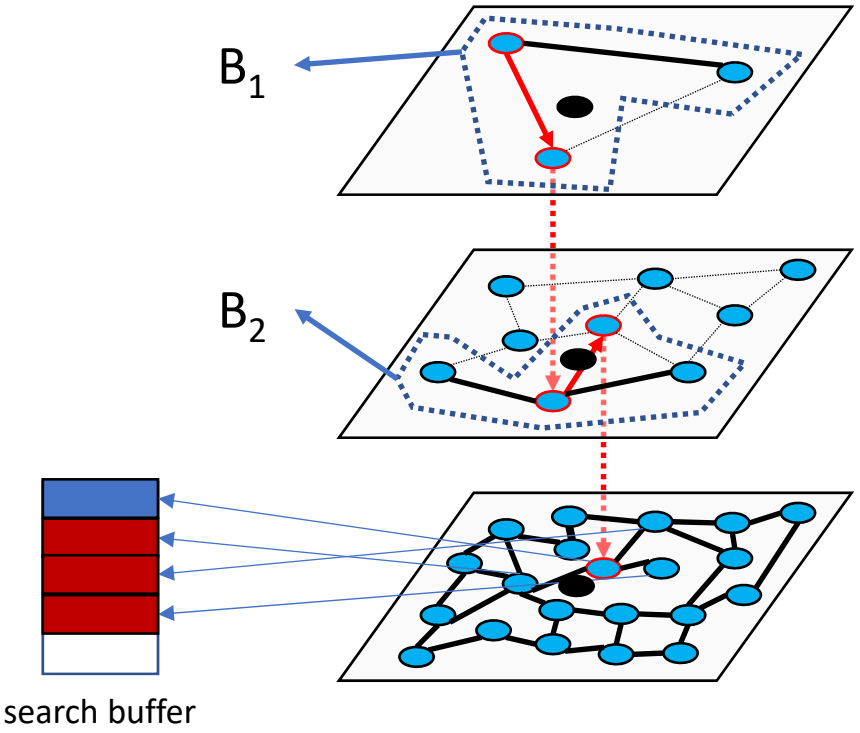
Hypersphere based hashing [SpH]

[SpH] Heo, Lee, He, Chang and Yoon, "Spherical Hashing: Binary Code Embedding with Hyperspheres", TPAMI, 2015.

Compressing HNSW (2): Hashing Objective

Preserve HNSW performance with features:

- Top layers (e.g. B_1 and B_2): hashing shall preserve rank 1 result in batch in order to preserve search path
- Bottom layer: hashing shall preserve ranks in search buffer to preserve search results



Compressing HNSW (2): Hashing Training

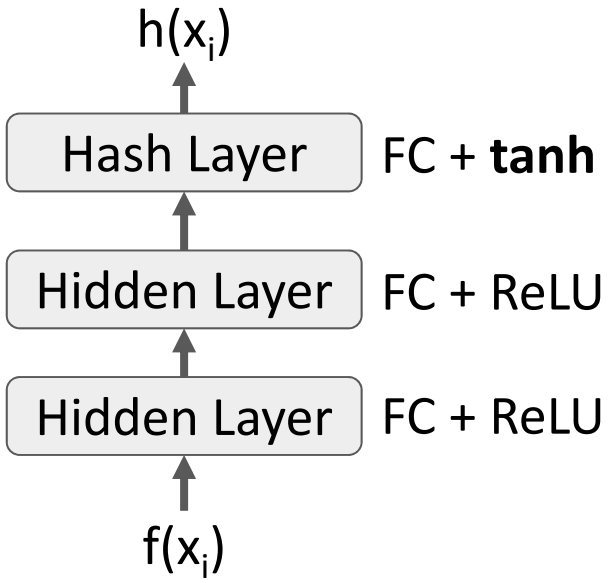
- Top batch: first sample **q**, current sample + neighbors sorted $\{x_1, \ldots, x_{maxM}\}$
- **Rank 1 loss**:

$$\sum_{i=2}^{\text{maxM}+1} \mathbb{1}\left[d(h(x_1), h(q)) > d(h(x_i), h(q))\right]$$

- Bottom batch: first sample **q**, and sorted samples $\{x_1, \ldots, x_{N-1}\}$ in search buffer
- **Full ranking loss**:

$$\sum_{i=1}^{N-1} \sum_{j=i+1}^{N-1} \mathbb{1}\left[d(h(x_i), h(q)) \geq d(h(x_j), h(q))\right]$$

- Minimize rank loss + regularization with SGD



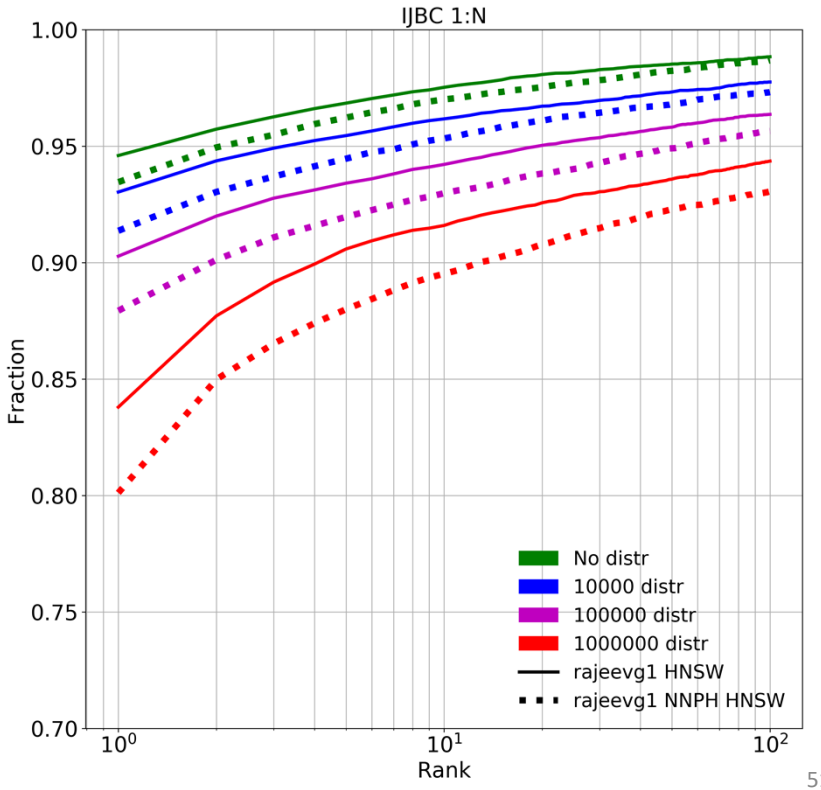
COLUMBIA UNIVERSITY
IN THE CITY OF NEW YORK

DVMMlab
digital video | multimedia laboratory

Hashing-compressed HNSW search performance on IJBC

Hashing for HNSW graph
(work in progress)

- discriminative face features from UMD
- Compress features with 512 hash bits
- Allows 8X compression with small performance loss at 1M samples



Summary

- Objective: spread-out, concentrated, compact features
- A simple Global Orthogonal Regularization constraint makes features more spread out over the entire space
- Proper heat-up procedure makes features more spread out and concentrated
- Compact hashing for graph-based logarithmic search