Лаборатория компьютерной
графики и мультимедиа
ВМК МГУ имени М.В. Ломоносова

Курс «Компьютерное зрение»

«Обучение метрик и поиск похожих изображений»

Антон Конушин и Тимур Мамедов

2025 год

План лекции

1. Постановка задачи, датасеты и метрики
2. Metric Learning
3. Эффективный поиск

Распознавание по лицу - верификация

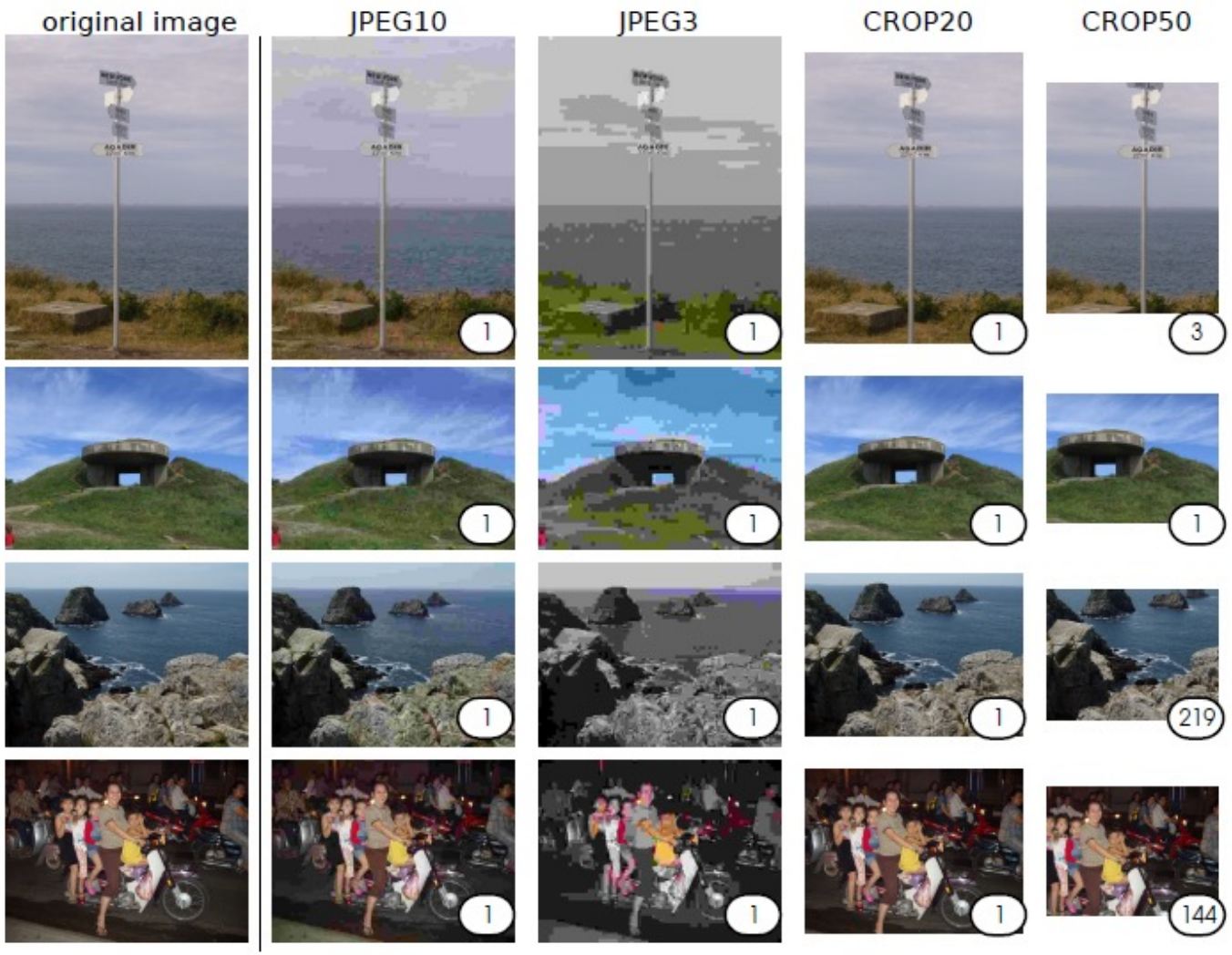
На обоих изображениях один и тот же человек, или нет?

Базовая задача распознавания лиц, которую и человеку проще всего решать

Распознавание человека

- «Watch list» - «белый» или «черный список»
  - Есть список людей с фотографиями
  - Необходимо определить, входит ли человек в этот список по его фотографии
- Сводим к попарному сравнению «запросов» с «изображениями из базы»
- Изображение может содержать лицо, портрет, ладонь, отпечаток пальца, всю фигуру

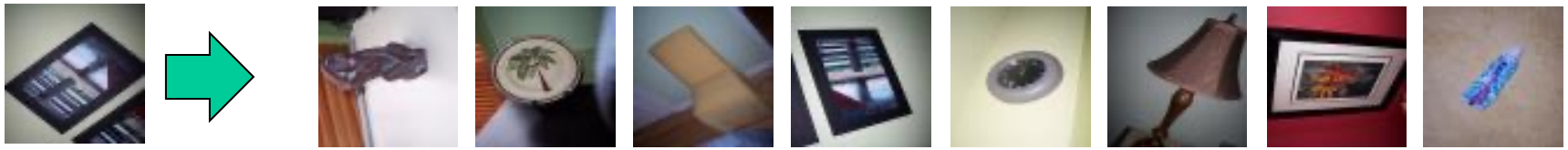
Поиск полудубликатов



Полудубликаты (Near-duplicates) – слегка измененная версия изображения (ракурс, цвета)

Поиск похожих изображений

- Или «поиск изображений по содержанию» (Content-based image retrieval)
- Есть коллекция изображений. Мы формулируем запрос в виде изображения-примера («найди то же самое»)



- Поиск по «визуальному сходству» изображения в целом или объектов в изображениях
- Чаще всего – нужно найти тот же самый объект, и другие похожие на него
- Классификация с открытым и заранее неизвестным набором классов

Общая схема поиска похожих изображений

Коллекция изображений

Построение индекса коллекции

Вектор-признак изображения (дескриптор) → Индекс коллекции (список вектор-признаков)

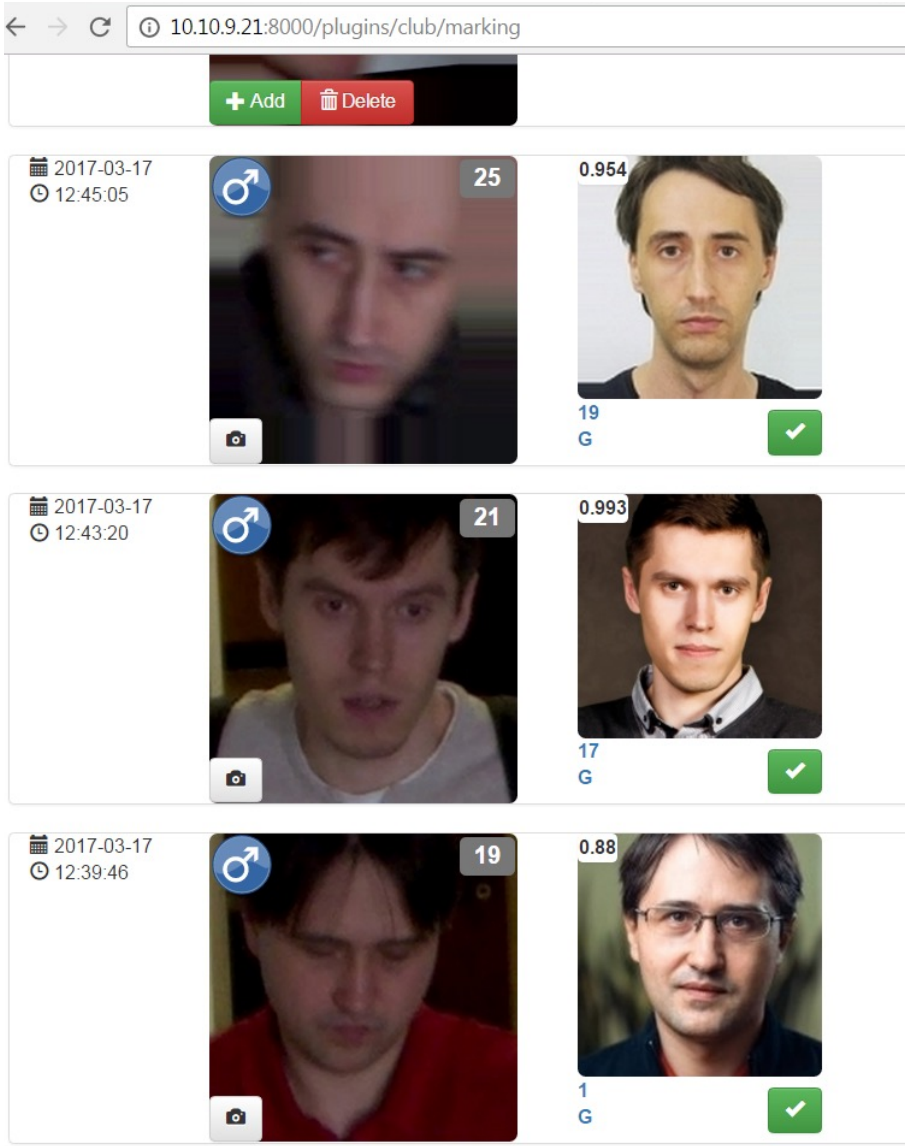
Изображение - запрос

Поиск изображения в коллекции

Вектор-признак запроса → Поиск по индексу → Ранжируем список похожих

Ищем ближайших соседей по выбранной метрике

Коллекции для распознавания человека



- Много вариантов:
  - Лицо
  - Радужка
  - Отпечаток пальцев
  - Фигура (и т.д.
- Тоже многомилионных размеров и автоматическое построение
- У Гугла – 250 млн. изображений лиц

Megaface challenge (2016)

MegaFace and MF2: Million-Scale Face Recognition

The MegaFace challenge has concluded, reaching a benchmark performance of over 99%. Because its goals have been met, and ongoing maintenance of this platform would require considerable administrative effort, MegaFace is being decommissioned and MegaFace data are no longer being distributed. ››

Distractors

1 Million Photos

690,572 Unique Users

Training Set

4.7 Million Photos

672,057 Unique Identities

7 Mean photos / person (3 min, 2469 max)

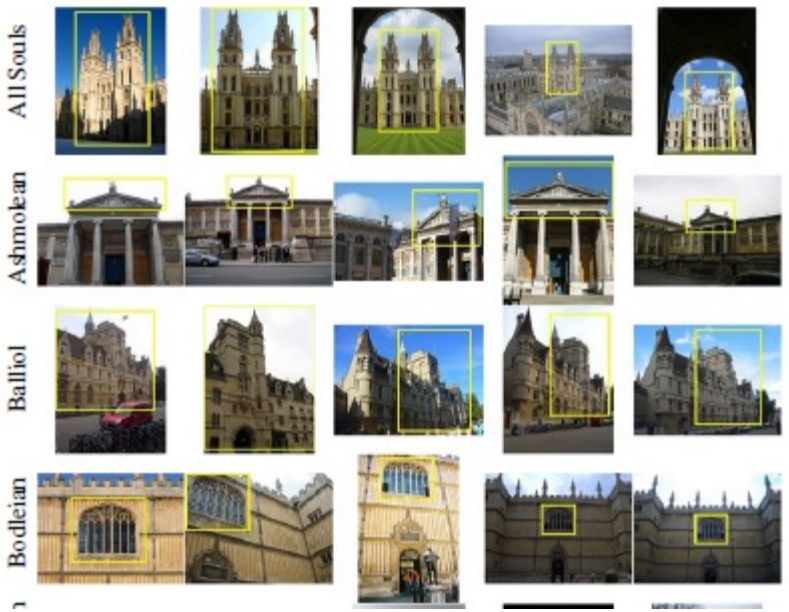
Test Sets

FaceScrub Celebrities

FGNet Age-invariant non-celebrities

http://megaface.cs.washington.edu/

Датасет Oxford Buildings dataset



- Oxford5K
  - 5062 изображения 1024x768 достопримечательностей (landmark) Окфорда
- 100K и 1M
  - коллекции изображений-дистракторов по самым популярным во flickr запросам

- 55 изображений запросов – по 5 изображений на 11 достопримечательностей
- По каждому запросу база размечена (Good – хорошо видно, OK - > 25%, Junk <25%, Absent – не видно)

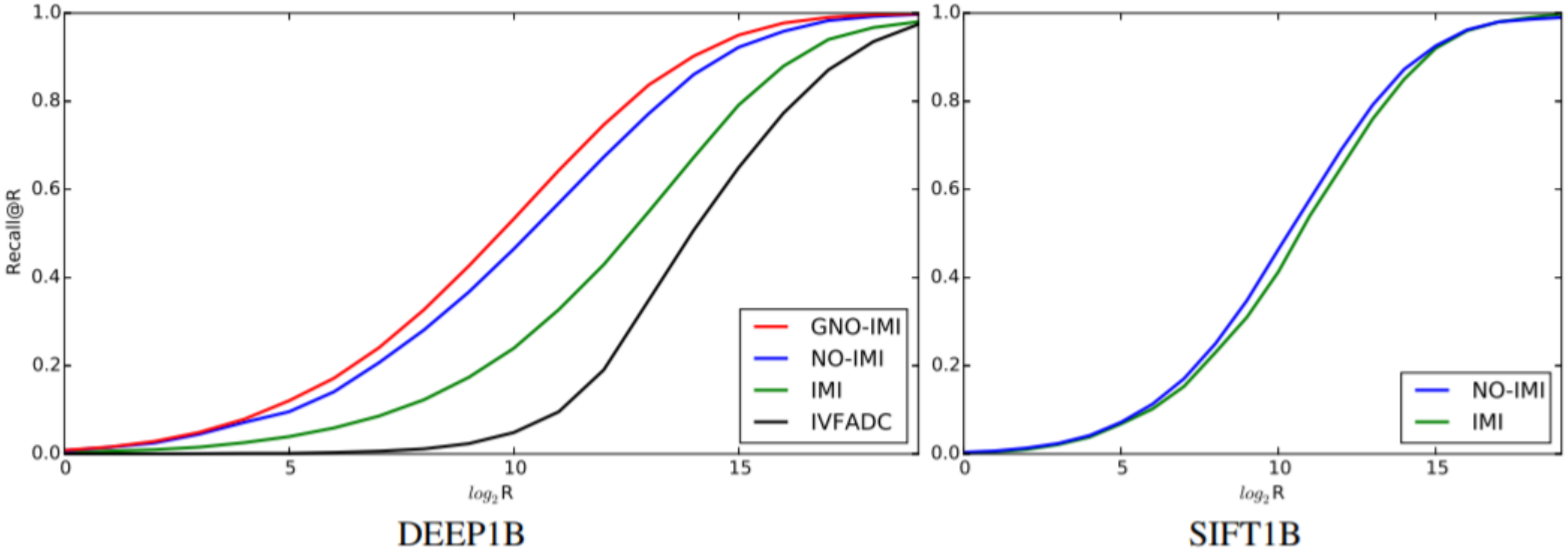
J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, “Object retrieval with large vocabularies and fast spatial matching,” CVPR, 2007.

Google Landmarks

- Сбор коллекций в полуавтоматическом режиме путем запросов к поисковой системе
- 762k изображений в индексе, 4.1M изображений для обучения, 200к достопримечательностей
- Получены из Wikimedia, полуавтоматическая разметка, 800 человеко-часов
- Всего 118к тестовых запросов

Weyand et al. Google Landmarks Dataset v2. CVPR 2020

Метрики качества – Recall@R



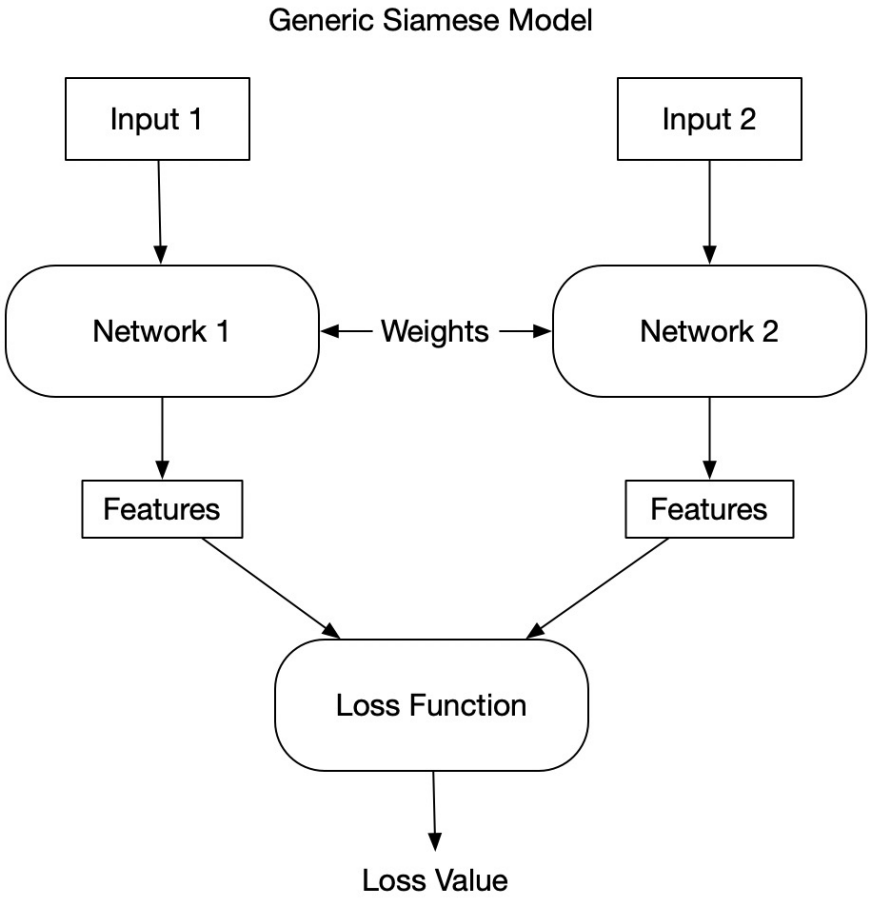
- R – длина выдаваемого списка
- Recall@R – вероятность наличия истинного соседа в списке длины R
- Функцию можно сэмплировать (Recall@1, Recall@10) или усреднять mAP

https://openaccess.thecvf.com/content_cvpr_2016/papers/Babenko_Efficient_Indexing_of_CVPR_2016_paper.pdf

План лекции

1. Постановка задачи, датасеты и метрики
2. Metric Learning
3. Эффективный поиск

Contrastive loss



$$\mathscr{L}_{\text{cont}}(x_1, x_2, \theta) = \begin{cases} \mathscr{D}^2_{f_\theta}(x_1, x_2), & y_1 = y_2 \\ \max(0, \alpha - \mathscr{D}^2_{f_\theta}(x_1, x_2)), & y_1 \neq y_2 \end{cases}$$

- Пришли из «сиамских» сетей
- Минимизируем расстояние между примерами одного класса
- Требуем чтобы расстояние между примерами разных классов было больше параметра a

Hadsell et al. Dimensionality Reduction by Learning an Invariant Mapping. CVPR 2005

Triplet loss

Loss function: $\mathscr{L}_{\text{triplet}} = \max\left(0, \mathscr{D}^2_{f_\theta}(x_a, x_p) - \mathscr{D}^2_{f_\theta}(x_a, x_n) + \alpha\right)$

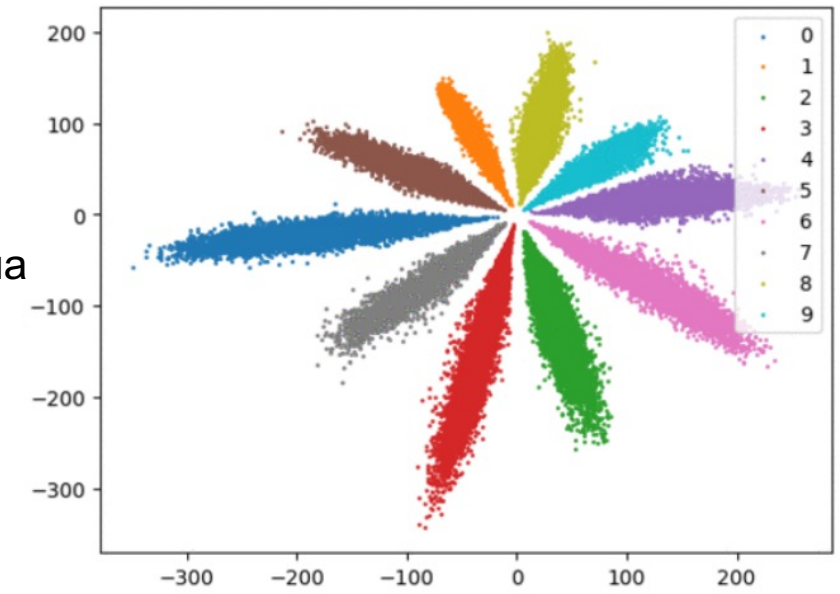
Training: for x_a choose $\arg\max \mathscr{D}_{f_\theta}(x_a, x_p)$ and $\arg\min \mathscr{D}_{f_\theta}(x_a, x_n)$ online from a large batch s.t.

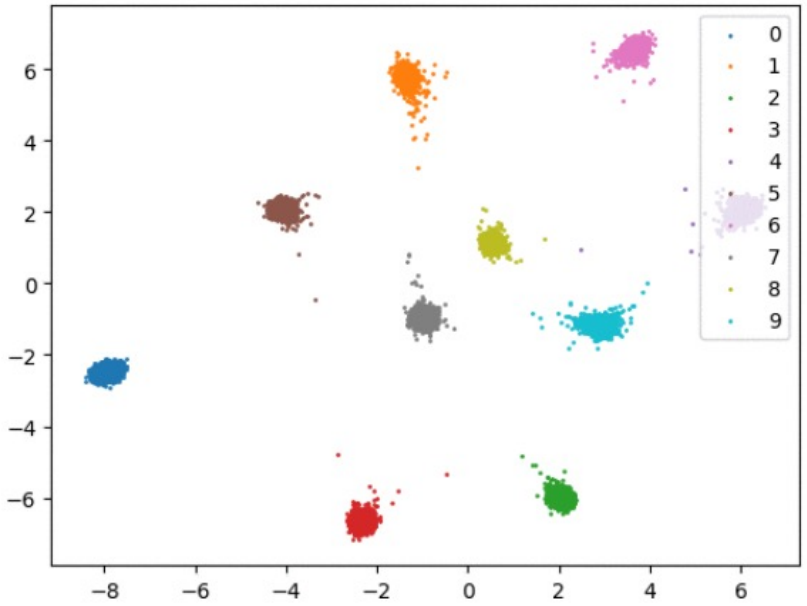
$$\mathscr{D}_{f_\theta}(x_a, x_n) < \mathscr{D}_{f_\theta}(x_a, x_p) + \alpha$$

Schroff et al. FaceNet: A Unified Embedding for Face Recognition and Clustering. CVPR 2015

Center loss

Пример на MNIST





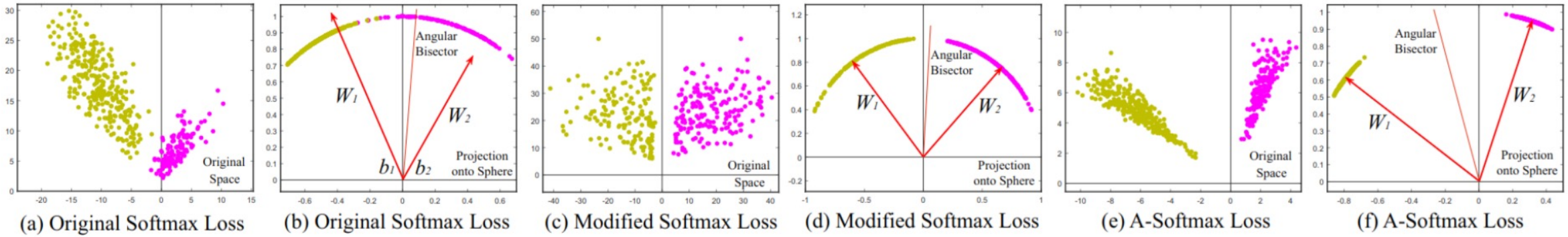
$$\mathcal{L} = \mathcal{L}_S + \lambda \mathcal{L}_C$$

$$\mathcal{L}_C = \sum_{i=1}^{m} \left\| x_i - c_{y_i} \right\|_2^2$$

Обучение: обновляем центры C_{y_i} после шага градиентного спуска медленно с параметром $\mathrm{a} = 0.99$

Wen et al. A Discriminative Feature Learning Approach for Deep Face Recognition. ECCV 2016

SphereFace



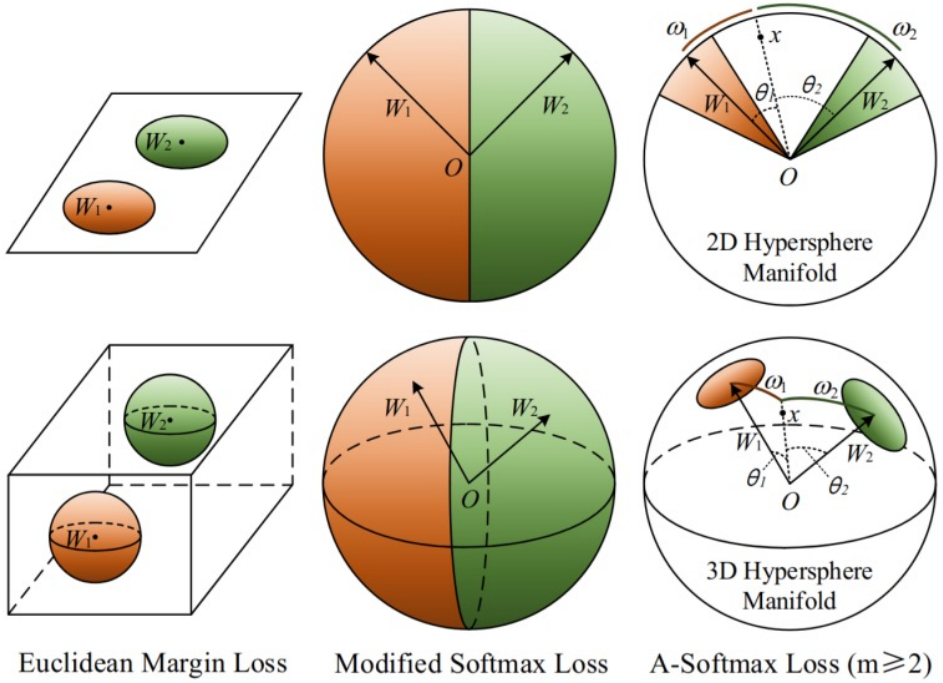
(a) Original Softmax Loss (b) Original Softmax Loss (c) Modified Softmax Loss (d) Modified Softmax Loss (e) A-Softmax Loss (f) A-Softmax Loss

Loss Function	Decision Boundary
Softmax Loss	$(\boldsymbol{W}_1-\boldsymbol{W}_2)\boldsymbol{x}+b_1-b_2=0$
Modified Softmax Loss	$\|\boldsymbol{x}\|(\cos\theta_1-\cos\theta_2)=0$
A-Softmax Loss	$\|\boldsymbol{x}\|(\cos m\theta_1-\cos\theta_2)=0$ for class 1 $\|\boldsymbol{x}\|(\cos\theta_1-\cos m\theta_2)=0$ for class 2

Table 1: Comparison of decision boundaries in binary case. Note that, θ_i is the angle between $\boldsymbol{W}_i$ and $\boldsymbol{x}$.

$$L_{\text{modified}}=\frac{1}{N}\sum_i-\log\Big(\frac{e^{\|\boldsymbol{x}_i\|\cos(\theta_{y_i,i})}}{\sum_j e^{\|\boldsymbol{x}_i\|\cos(\theta_{j,i})}}\Big)$$

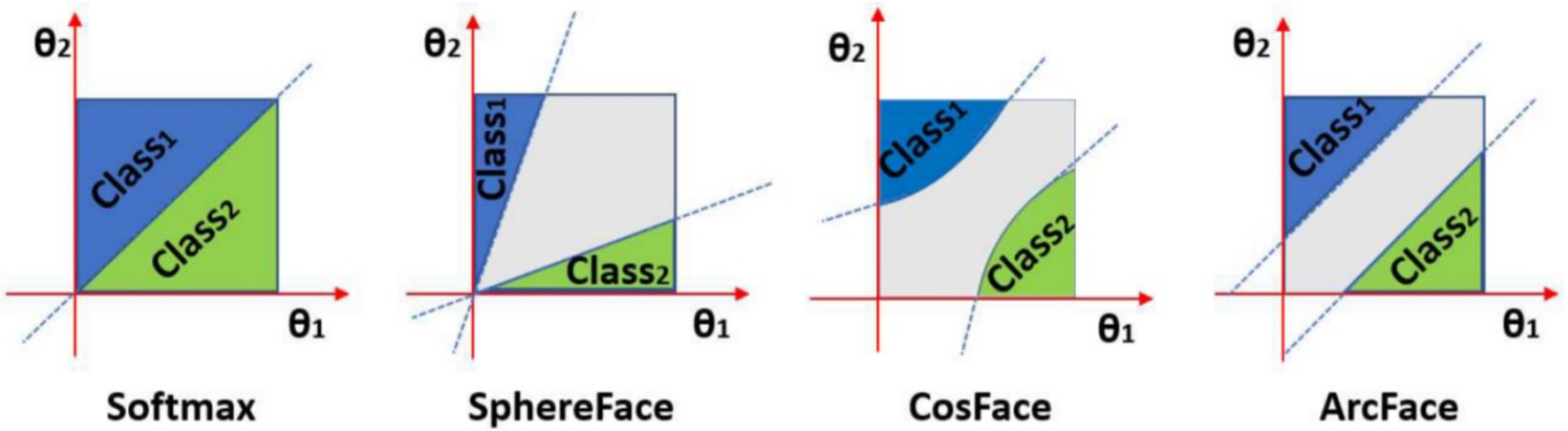
$$L_{\text{ang}}=\frac{1}{N}\sum_i-\log\Big(\frac{e^{\|\boldsymbol{x}_i\|\cos(m\theta_{y_i,i})}}{e^{\|\boldsymbol{x}_i\|\cos(m\theta_{y_i,i})}+\sum_{j\neq y_i}e^{\|\boldsymbol{x}_i\|\cos(\theta_{j,i})}}\Big)$$



Euclidean Margin Loss Modified Softmax Loss A-Softmax Loss (m⩾2)

Liu et al. SphereFace: Deep Hypersphere Embedding for Face Recognition. CVPR 2017

CosFace, ArcFace



$$L_{lmc} = \frac{1}{N}\sum_{i} -\log \frac{e^{s(\cos(\theta_{y_i,i})-m)}}{e^{s(\cos(\theta_{y_i,i})-m)} + \sum_{j\neq y_i} e^{s\cos(\theta_{j,i})}}$$

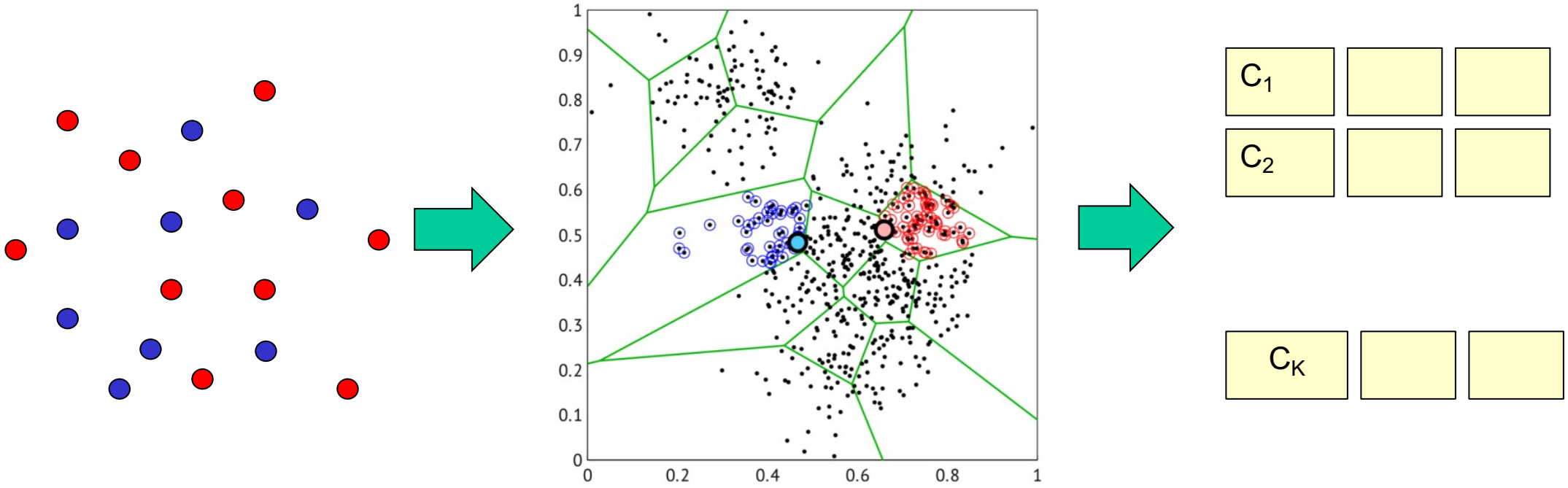
$$L_3 = -\frac{1}{N}\sum_{i=1}^{N} \log \frac{e^{s(\cos(\theta_{y_i}+m))}}{e^{s(\cos(\theta_{y_i}+m))} + \sum_{j=1,j\neq y_i}^{n} e^{s\cos\theta_j}}$$

Wang et al. CosFace: Large Margin Cosine Loss for Deep Face Recognition. CVPR 2018
Deng et al. ArcFace: Additive Angular Margin Loss for Deep Face Recognition. CVPR 2019

План лекции

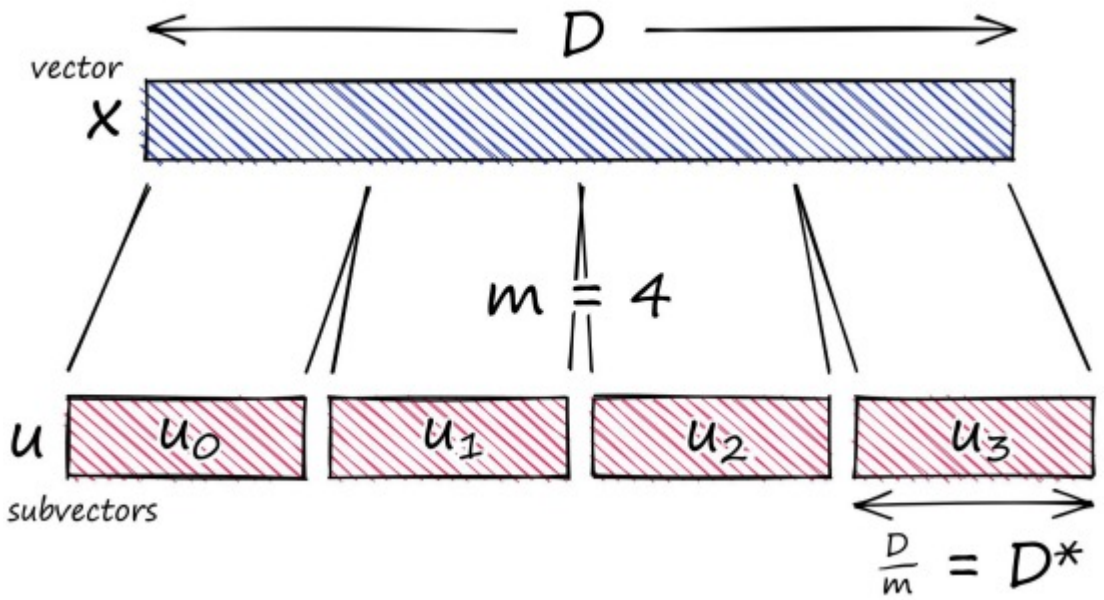
1. Постановка задачи, датасеты и метрики
2. Metric Learning
3. Эффективный поиск

Инвертированный индекс



- **Построение**: применяем k-means для разбиения изображений на k кластеров. k центроидов (codewords) образуют codebook. Храним k списков с ID изображений в RAM
- **Поиск**: по запросу q, найдем несколько ближайших codewords. Возвращаем все ID из соответвующих списков.

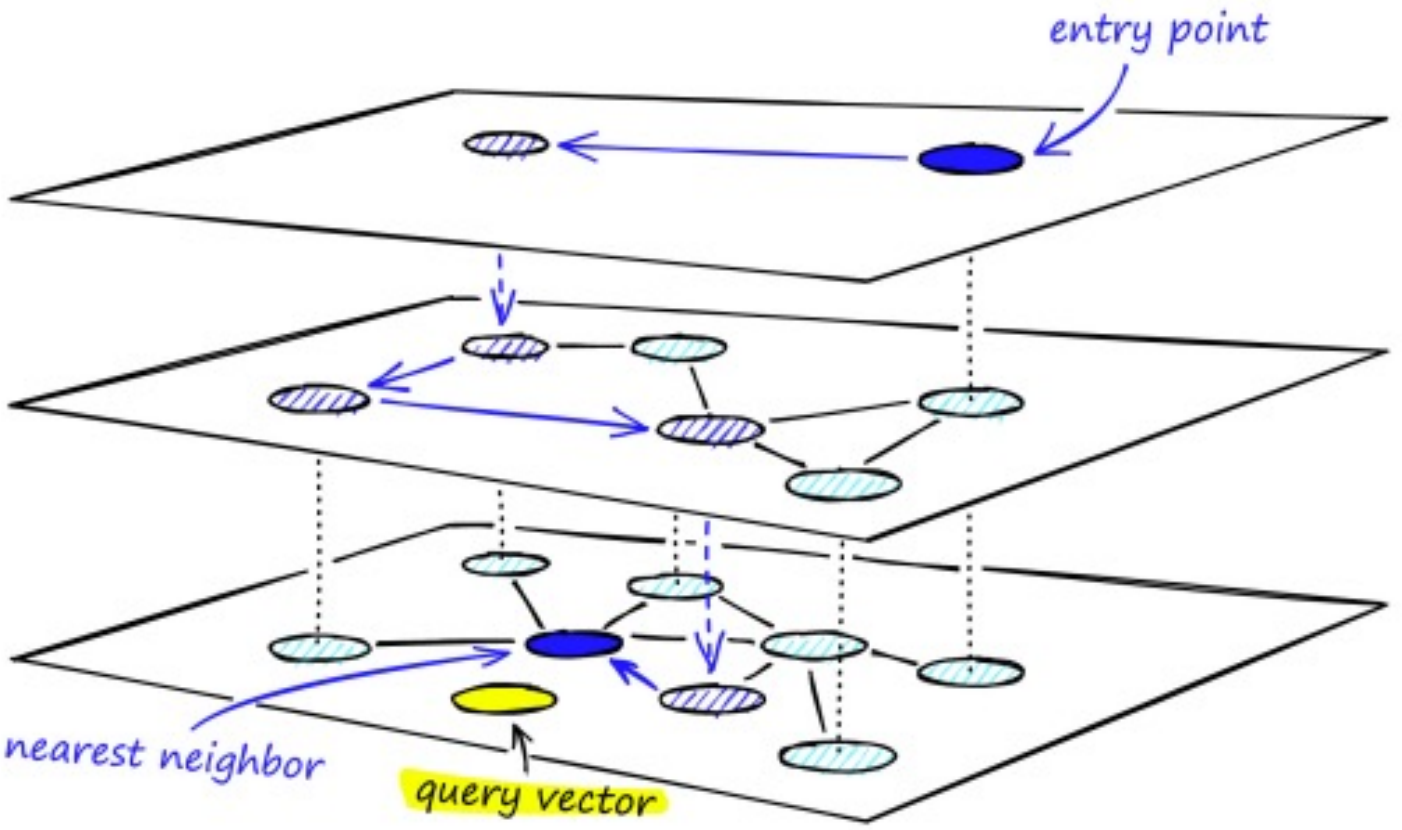
Product quantization (PQ)



- Простроение:
  - Разобьём вектор x длины D на m частей
  - Применим K-means к каждому отдельно
- Сравнение с K-means:
  - kD и $mk^*(D/m)=k^{1/m}D$
  - Обычно k*=256 (1 байт)

H.Jégou et. al. Product quantization for nearest neighbor search, PAMI 2011

Hierarchical Navigable Small World



Грубая оценка точности

Поиск за O(logN)

Построение за O(N log N)

Память: 60-450 байт/объект

Malkov, Yashunin. Efficient and robust approximate nearest neighbor search using Hierarchical Navigable Small World graphs. TPAMI 2018

Вариант общего метода

1. Строим инвертированный индекс с большим $K = 2^{20}$
2. В каждом кластере кодируют остаточные вектора с помощью PQ
3. Применяем HNSW для выбора кластеров при поиске

		DEEP1B					SIFT1B				
Method	K	R@1	R@10	R@100	t	Mem	R@1	R@10	R@100	t	Mem
O-Multi-D-OADC[24]	2^{14}	0.397	0.766	0.909	8.5	17.34	0.360	0.792	0.901	5	17.34
Multi-LOPQ[4]	2^{14}	0.41	0.79	-	20	18.68	**0.454**	**0.862**	0.908	19	19.22
GNOIMI[5]	2^{14}	0.45	0.81	-	20	19.75	-	-	-	-	-
IVFOADC+G+P	2^{20}	**0.452**	**0.832**	**0.947**	**3.3**	17.87	0.405	0.851	**0.957**	**3.5**	18

Table 4. Comparison to the previous works for 16-byte codes. The search runtimes are reported in milliseconds. We also provide the memory per point required by the retrieval systems (the numbers are in bytes and do not include 4 bytes for point ids).

Baranchuk et al. Revisiting the inverted indices for billion-scale approximate nearest neighbors. ECCV 2018

Резюме

Мы рассмотрели три компонента систем поиска:

1. Постановки задач и датасеты
2. Фукнции потерь для обучения метрик, которые заставляют признаки лежать в компактных многообразиях (manifolds)
3. Приближенные методы поиска ближайших соседей для масштабных задач поиска по картинкам