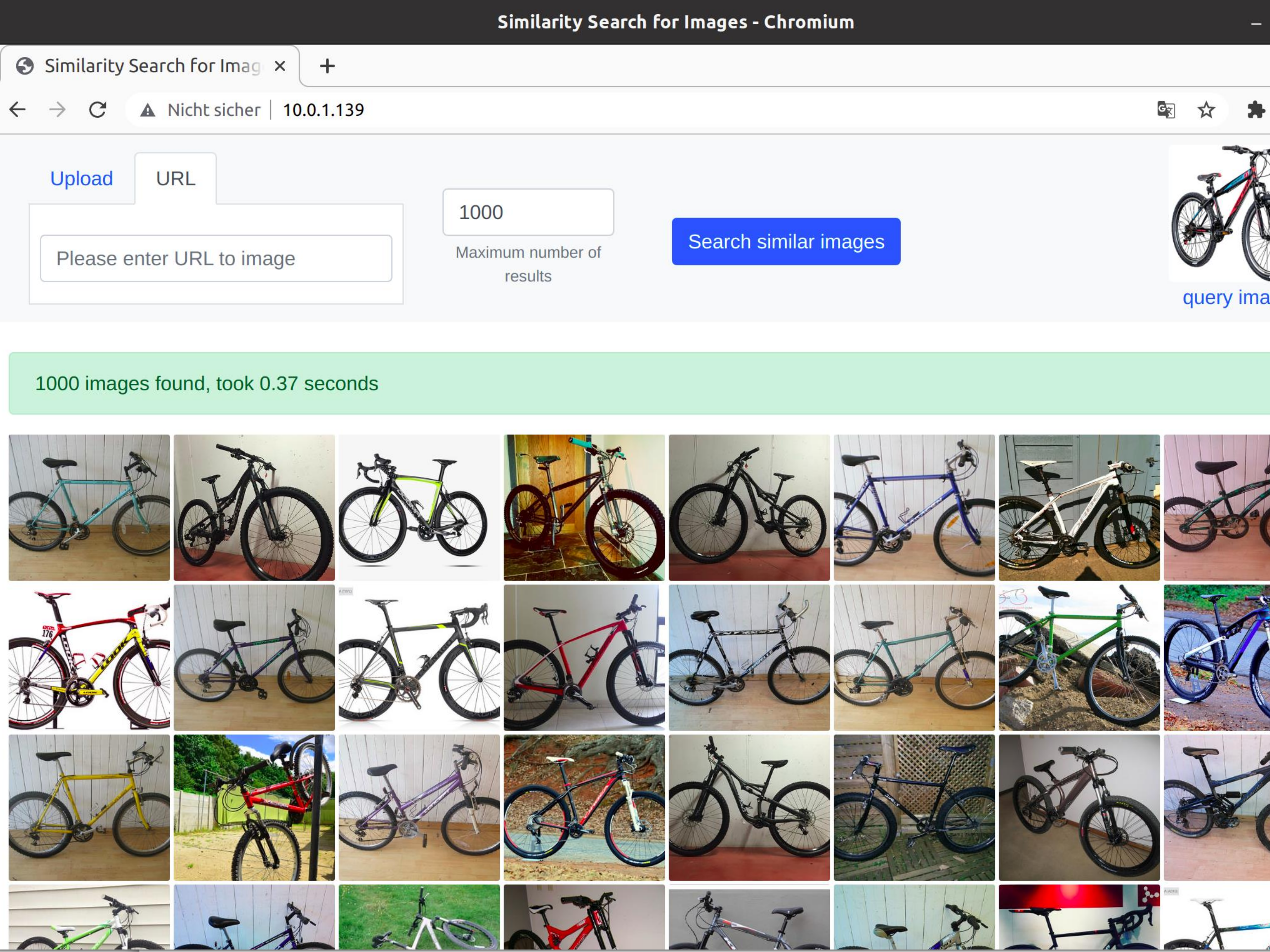
ElasticHash: Semantic Image Similarity Search by Deep Hashing with Elasticsearch

Nikolaus Korfhage
Markus Mühling
Bernd Freisleben

Department of Mathematics and Computer Science,
University of Marburg, Marburg, Germany
{korfhage,muehling,freisleb}@informatik.uni-marburg.de

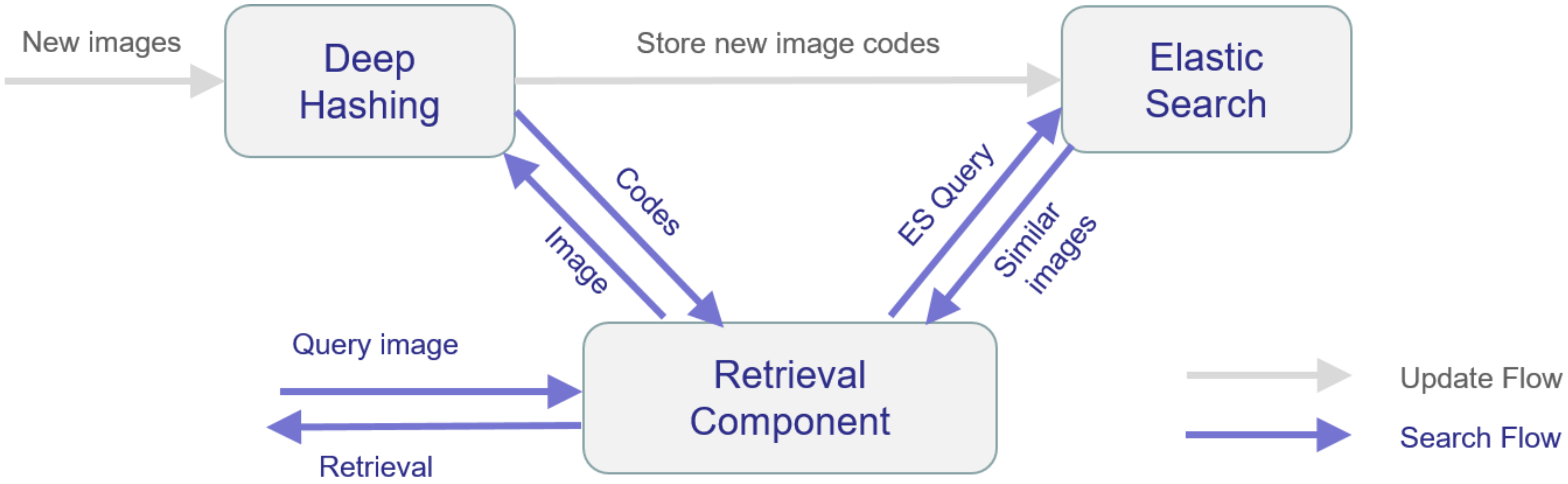
Image Similarity Search: Query by Example


Similarity Search for Images - Chromium
Similarity Search for Imag
Nicht sicher
10.0.1.139
Upload
URL
Please enter URL to image
1000
Maximum number of results
Search similar images
query ima
1000 images found, took 0.37 seconds

Motivation

- ElasticSearch is a full-text search system
- ElasticSearch advantages:
  - Fast due to inverted indices
  - Scalable to hundreds of server
  - Load balancing
  - …
- Idea:
  - Integrate Image Similarity Search into ElasticSearch
  - Multi-modal queries

Similarity Search System



Deep Hashing Model

- ImageNet-pretrained EfficientNetB3
- Trained on all ImageNet Images with >= 1000 examples and PlacesV2
- 256-bit coding layer
- Combination of Log-loss and triplet-loss

$$\mathcal{L} = \alpha \sum_{i=1}^{k} y_i \log_{p_i} + \beta \max(d(a,p) - d(a,n) + \gamma, 0)$$

Review: Multi-Index Hashing (MIH)

- Binary codes:
  - $\mathrm{h} = (\mathrm{h}^1, \dots, \mathrm{h}^\mathrm{m})$
  - $\mathrm{g} = (\mathrm{g}^1, \dots, \mathrm{g}^\mathrm{m})$
- m: number of partitions
- H: Hamming distance

- Proposition:

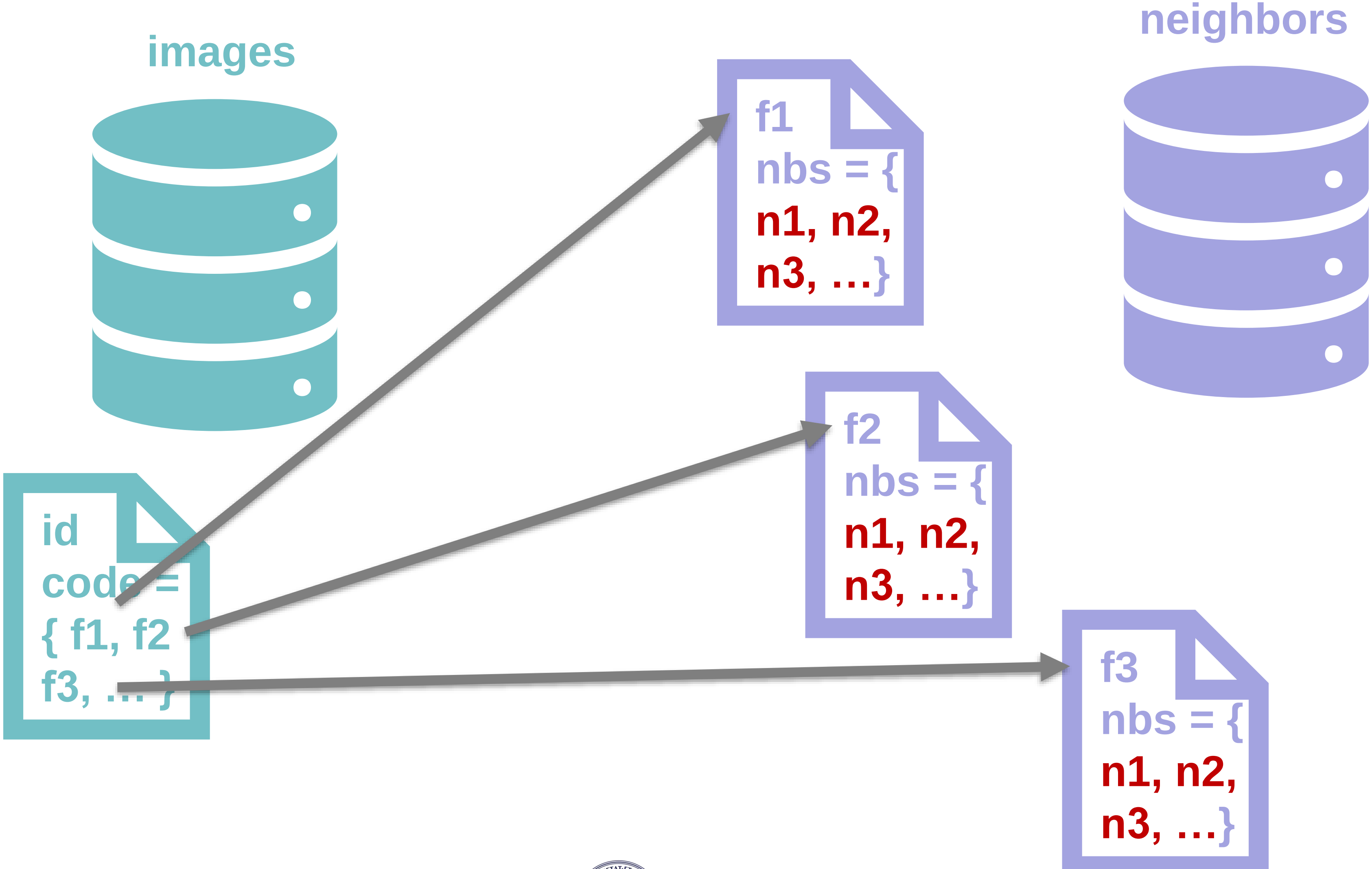
$$\|h - g\|_H \leq r \Rightarrow \exists k \in \{1..m\} \ \left\|h^k - g^k\right\|_H \leq \left\lfloor \frac{r}{m} \right\rfloor$$

- Example: 64-bit codes decomposed into $\mathrm{m} = 4$ subcodes
  Within Hamming radius $\mathrm{r} = 4$ at least one of the subcodes has distance $\mathrm{d} \leq \left\lfloor \frac{r}{m} \right\rfloor = 2$

Two-Stage Approach

- Hamming distance computation on complete data set of long hashcodes (256 bit) ➔ does not scale
- 1st Step: Filtering (64 bit codes):
  - Find all 64 bit codes with distance < 12
  - Select 64 most important bits from 256 bit codes
  - Apply Multi-Index Hashing [Norouzi et al. 2012]:
    - Partition 64 bit codes into $m = 4$ subcodes
    - Within Hamming radius $r = 11$ at least one of the subcodes has distance $d \leq \left\lfloor \frac{r}{m} \right\rfloor = 2$
  - Searching subcode radii $d = 2$ much faster (ES inverted index)
- 2nd Step: Re-Ranking (256 bit codes):
  - More accurate
  - Hamming distance computation only on small subset

Integration into ElasticSearch



Re-Ranking

```
POST _scripts/hd64

{
  "script":
   {
     "lang": "painless",
     "source":
      64-Long.bitCount(params.subcode^doc[params.field].value)
   }
}
```

Images Index

```
PUT /es-retrieval
PUT /es-retrieval/default/_mapping

{
  "properties": {
    "image": {"type": "text"},
    "f_0": {"type": "keyword"},
    "f_1": {"type": "keyword"},
    "f_2": {"type": "keyword"},
    "f_3": {"type": "keyword"},
    "r_0": {"type": "long"},
    "r_1": {"type": "long"},
    "r_2": {"type": "long"},
    "r_3": {"type": "long"}
  }
}
```

1. Filtering:
   4 x 16-bit subcodes

2. Re-ranking:
   4 x 64-bit subcodes

Neighbors index

- Created once
- Contains neighboring hashcodes for each subcode f_j
- 2^{16} documents (subcodes)
- Example: all possible neighbors of *01* are *01,10,00,11*, i.e. *1,2,0,3*

```
POST/nbs-d2/_doc/<16 bit subcode>

{
  "nbs" : [ <d2 neighbors  of 16 bit  subcode> ]
}
```

Search Query

```
GET /es-retrieval/_search

{ "query": { "function_score":  {"boost_mode": "sum", "score_mode":
"sum", "functions":
  [ ..., { "script_score": {"script": {"id": "hd64",
    "params": {
        "field": "r_<i>",
        "subscode": <64 bit subcode for re-ranking>} } },
         "weight": 1}, ... ],

  "query": {"constant_score":{"boost": 0.,
    "filter":{"bool":{"minimum_should_match": 1,"should":
    [..., {"terms":
        {"f_<j>":
          {"id": "<16 bit subcode for lookup>",
           "index": "nbs-d2",
            "path": "nbs"} } }, ... ]
} } } },} } }
```

2. Re-ranking

1. Filtering

Evaluation

- OpenImages:
  - Multi-label
  - 9.2M Flickr images
  - ~20K different labels
  - Train/val/test splits

- Evaluation of ElasticHash:
  - Available database images: ~7M
  - Query images (quality): ~120K
  - Query images (latency): 10K

- Intel Core i7-4771 CPU @ 3.50GHz and 32 GB RAM

Results: Search Quality

- Mean AP@k for 121,588 query images
- Database images: ~7M
- Query images: ~120K

Top *k*	**10**	**25**	**50**	**100**	**250**	**500**	**1000**
Short	87.94	86.08	84.44	82.54	79.41	76.44	72.86
Long	95.35	94.72	94.23	93.71	92.90	92.09	90.95
Two-stage	95.21	94.48	93.90	93.22	92.02	90.61	88.42

Results: Search Latency

- Milliseconds (ms)
- Database images: ~7M
- Query images: 10K

Top *k*		**10**	**25**	**50**	**100**	**250**	**500**	**1000**
Short	μ	23.09	23.98	24.45	25.58	28.38	33.09	42.20
	σ	4.74	4.65	4.70	4.72	4.86	5.20	6.07
Long	μ	111.83	111.58	111.99	113.05	116.77	121.98	132.60
	σ	16.50	16.58	16.72	16.54	17.04	17.13	17.99
Two-stage	μ	36.12	36.75	37.28	38.17	40.88	45.73	55.23
	σ	7.80	7.96	7.81	7.89	7.93	8.12	8.64

Conclusion

- Deep hashing based image similarity search in ElasticSearch
- Seamless integration into ES by terms queries and neighbors index
- Evaluation on OpenImages dataset
- Low search latencies: short hash codes for filtering (64 bit)
- High retrieval quality: long hash codes for re-ranking (256 bit)
- Code, indices, models etc.: https://github.com/umr-ds/ElasticHash

References

- Tan, M., & Le, Q. (2019, May). Efficientnet: Rethinking model scaling for convolutional neural networks. In International Conference on Machine Learning (pp. 6105-6114). PMLR.
- Norouzi, M., Punjani, A., & Fleet, D. J. (2012, June). Fast search in hamming space with multi-index hashing. In 2012 IEEE conference on computer vision and pattern recognition (pp. 3108-3115). IEEE.
- Kuznetsova, Alina, et al. "The open images dataset v4." International Journal of Computer Vision 128.7 (2020): 1956-1981.