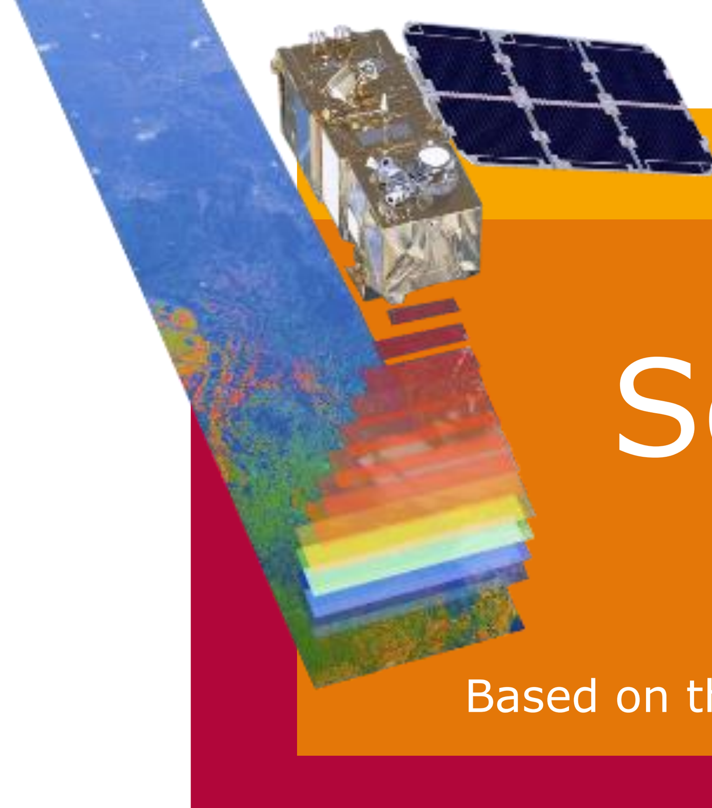


Scalable Earth Observation (EO) Image Indexing and Retrieval

Based on the lecure: “Deep Earth Query: Advances in Satellite Image Indexing from Massive Archives” by Prof.Dr. Begüm Demir

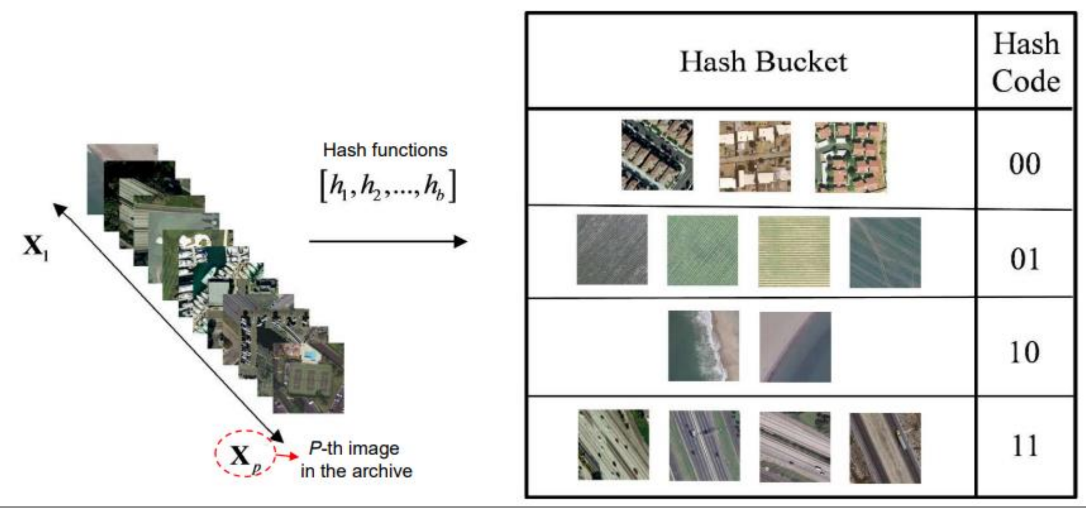
Problem Statement

A common problem of Information Discovery in large archives of EO satellite images is content based image retrieval where the archive is scanned for images that show semantic similarity to a given query image. E.g. if a satellite records an image of burning forest, how can other regions in all over the world be found that suffer from burning forest as well? Since EO image archives can contain petabytes of data, scanning the whole database would be very inefficient and therefore content based image indexing is required to enable efficient querying. This poster presents different techniques for building such indices.

Hash Indices for Images

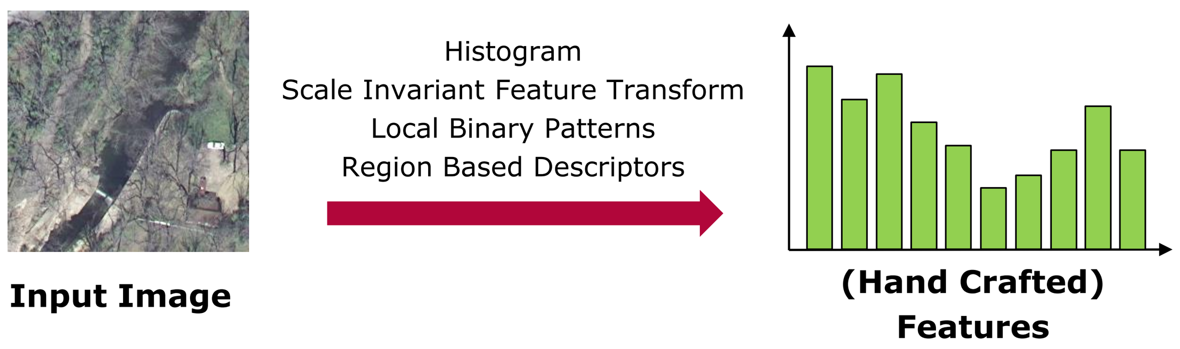
can be used to group images into a number of hash buckets that can be accessed efficiently. The goal is to assign similar images to the same bucket and dissimilar images to different buckets such that only one bucket needs to be considered when answering a query. Hashing can significantly reduce the time needed for processing a query while at the same time maintaining a high accuracy in comparison to an exhaustive search.

Similarity between images can be described by comparing content based descriptors of the images. The retrieved images can then be ranked by their similarity to the input.



Conventional Hashing Methods

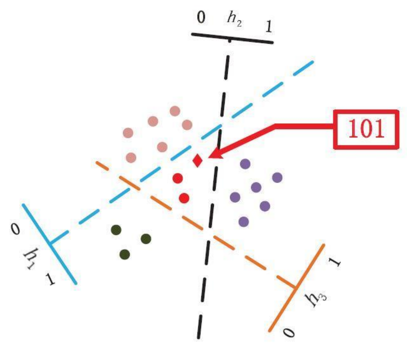
Preprocessing



Before applying the hash function, images are transformed into a vector representation where images with similar contents are represented by similar features. The hash function is then applied on these features.

Locality Sensitive Hashing (LSH)

This unsupervised method randomly creates n hyperplanes in the feature space that separate it into distinct hash-buckets. Similar images have a high probability to be contained in the same bucket and dissimilar images have a high probability to be contained in different buckets.



Feature Space with three hyperplanes. Source: [2]

♦ query image

$$h_r(X_t) = \begin{cases} 1, & if\ v_r^T X_t \geq 0 \\ 0, & otherwise \end{cases}$$

v_r is a random vector in feature space

$r = 1, 2, \ldots b$

Hash function:

$$h(X_t) = h_1(X_t) h_2(X_t) \ldots h_r(X_t)$$

Advantages

- Fast, scalable, data-independent

Disadvantages

- Data might not be linearly separable → bad accuracy
- Features might not represent image contents ideally

Kernel Based LSH (KLSH)

This method uses a kernel to perform LSH and can therefore deal with data that is nonlinearly separable. It can be done in an unsupervised or supervised way.

$$h_r(X_t) = sign\left(\sum_{i=1}^{m} \omega_r(i) K(X_t, X_i)\right)$$

K: Kernel function
ω_r: Weight function
m: Number of images

Unsupervised KLSH

The weights $\omega_r(i), i = 1 \ldots m$ are calculated based only on unlabeled images.

Supervised KLSH

The weights $\omega_r(i), i = 1 \ldots m$ are optimized on annotated images to make the hash function more distinctive.

Advantages

- Fast, scalable
- Able to deal with nonlinearly separable data

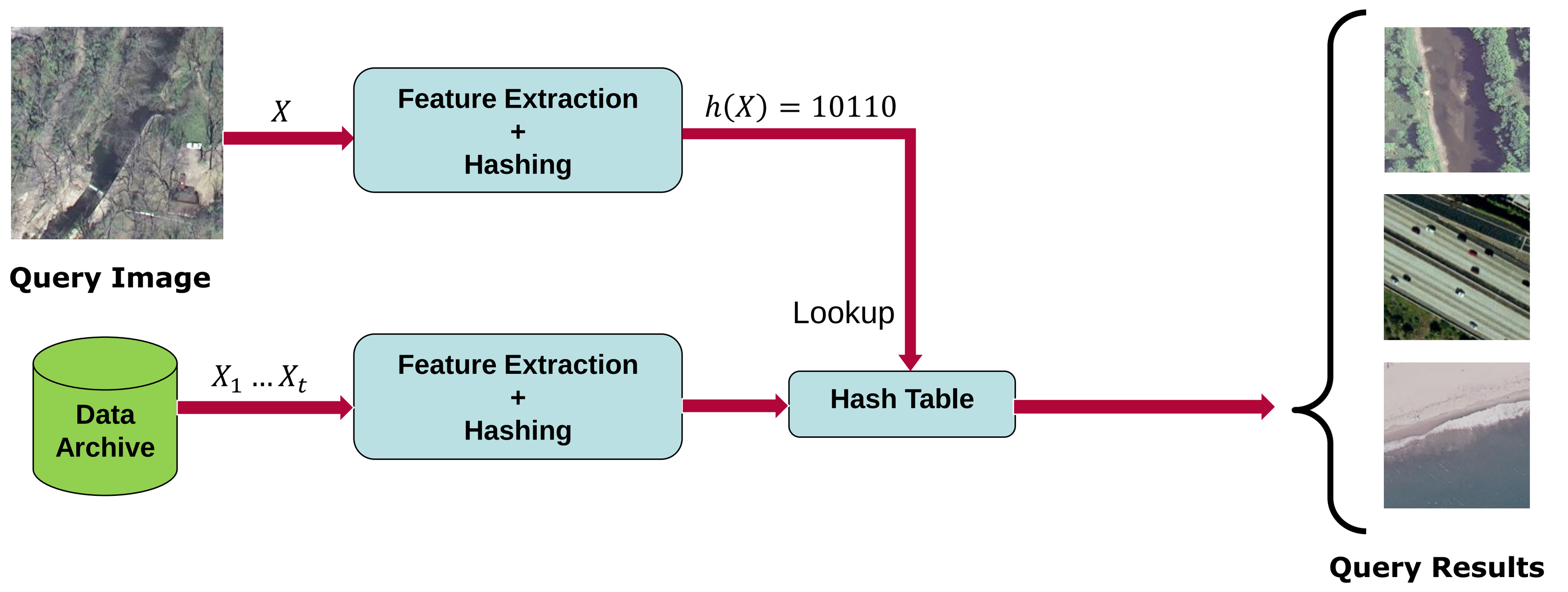
Disadvantages

- Features might not represent image contents ideally

System Layout

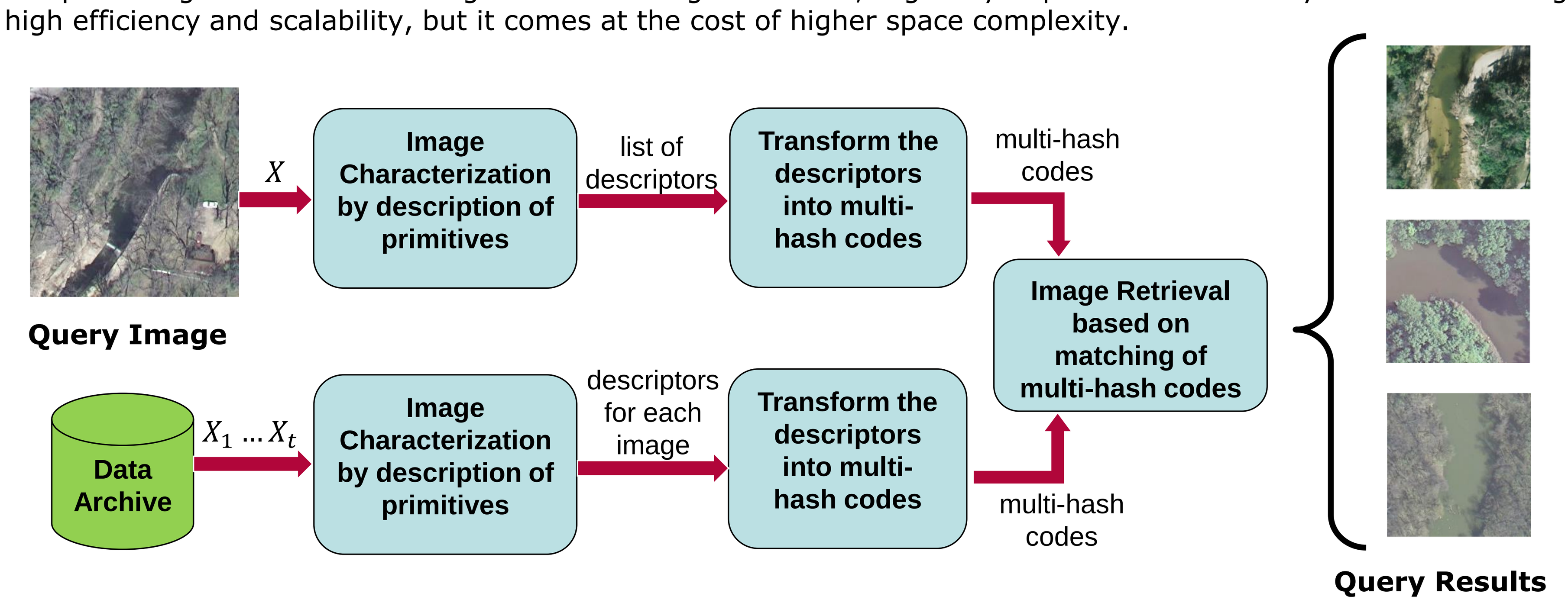
Single-Code Hashing

Single Code Hashing uses exactly one hash per image that is used as an index in the hash table.



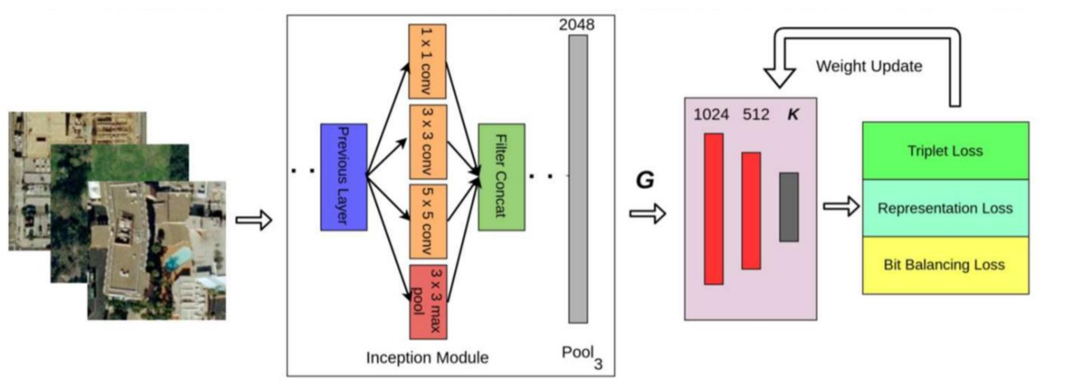
Multi-Code Hashing

Multi-Code Hashing describes different primitives in the image with different hash codes. A list of descriptors is retrieved from each image and then transformed into multiple hashes. This method is better capable of describing complex image contents than Single-Code Hashing. Therefore, it greatly improves the accuracy while maintaining high efficiency and scalability, but it comes at the cost of higher space complexity.



Deep Hashing

More recent approaches are aiming at learning a hash directly from an image using deep neural networks. This method is better capable of representing the contents of images in a hash because it does not require hand crafted features and instead learns what makes images similar on its own. The following picture shows the network architecture presented in [1].



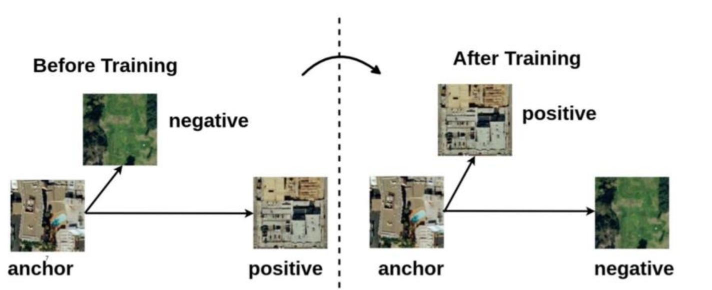
Advantages

- Fast, scalable
- Very accurate

Disadvantages

- Requires data specific training and a large amount of labeled training data

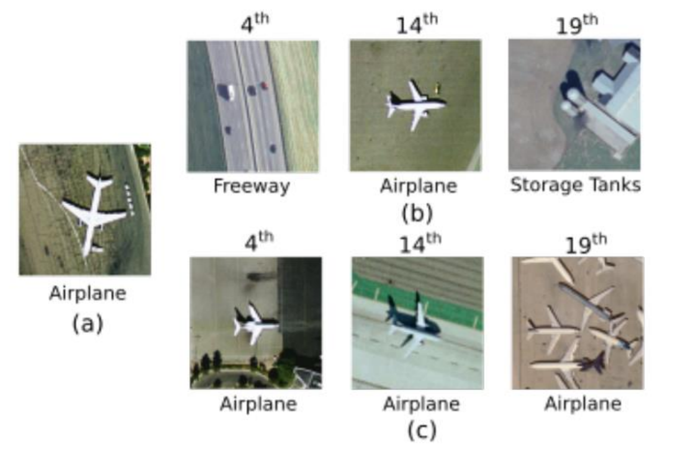
The network consists of an Inception Net Module, pretrained on the ImageNet dataset, and a smaller, fully connected network at the end that is fine tuned on EO images and outputs a hash value. One suitable way of training such a network to group similar images into the same hash bucket is Triplet Loss. Here, the network is presented triplets of data: one “anchor” image, one image similar to the “anchor” and one image dissimilar to the “anchor”. In each step, the networks weights are updated such that the positive sample moves closer to the anchor and the negative sample moves further away.



The concept of Triplet Loss: Every step the positive sample is moved closer to the anchor. Source: [1]

Comparison to Supervised Kernel LSH

The following experimental results show that Deep Hashing achieves a much higher precision than Kernel LSH while being equally fast.



(a) Is the query image, (b) are the query results with Supervised Kernel LSH and (c) are the query results with Deep Hashing. The images in (c) are more similar to (a) than the images in (b). Source: [1]

Method	Number of Hash Bits (K)					
	K=16		K=24		K=32	
	mAP	Time	mAP	Time	mAP	Time
Kernel LSH	0.557	25.3ms	0.594	25.5ms	0.630	25.6
Deep Hashing	0.875	25.3ms	0.890	25.5ms	0.904	25.6

Comparison of the mean average precision (mAP) and query execution time for Deep Hashing and Kernel LSH on the UCMD Data Set for different numbers of hash bits K. Deep hashing outperforms Kernel LSH while being equally fast. Source: [1]

Leon Schiller
B.Sc IT-Systems Engineering Student

Practical Data Engineering, Winter Term 2019/20
Hasso Plattner Institute, Potsdam, Germany

E-Mail: leon.schiller@student.hpi.de

References

[1] S. Roy, E. Sangineto, B. Demir, N. Sebe, "Metric-Learning based Deep Hashing Network for Content Based Retrieval of Remote Sensing Images", IEEE Geoscience and Remote Sensing Letters

[2] H. Li, T. Zhao, N. Li, Q. Cai, J. Du, " Feature Matching of Multi-View 3D Models based on Hash Binary Encoding", DOI:10.14311/nnw.2017.27.005