**Problem Statement: Generic Instance Search from 1 Example**

**Goal:**
Find all examples of an instance in a dataset, given 1 image of the instance and a bounding box. The instance can be any type, e.g., buildings, logos and specific scenes.

**Proposal:** Search local boxes in database images.
- selective search \([\text{van de Sande11iccv}]\) to sample ~1000 boxes per database image

**Problem:** VLAD/Fisher encodings for all boxes are too expensive to store or too slow to compute on-the-fly.

**Solution:**
- Decompose the VLAD/Fisher encoding of a box into a set of point-indexed representations. The number of points in an image is fixed, independent of the number of boxes.
- Decompose the cosine similarity of two VLAD/Fisher encodings into a sum of point-wise similarities.

\[
S_{\text{cos}}(x_1, x_2) = \frac{1}{\|v_1\|\|v_2\|} \sum_{x \in x_1} \sum_{y \in x_2} (c(x) \cdot c(y))d_x d_y
\]

- Evaluate on-the-fly all boxes by summing inside-box point scores and dividing the pre-computed norms. Point scores are computed once.

**Contribution I. Locality in the Image**

**Proposal:** Search local boxes in database images.
- selective search \([\text{van de Sande11iccv}]\) to sample ~1000 boxes per database image

**Problem:** VLAD/Fisher encodings for all boxes are too expensive to store or too slow to compute on-the-fly.

**Solution:**
- Decompose the VLAD/Fisher encoding of a box into a set of point-indexed representations. The number of points in an image is fixed, independent of the number of boxes.
- Decompose the cosine similarity of two VLAD/Fisher encodings into a sum of point-wise similarities.

\[
S_{\text{cos}}(x_1, x_2) = \frac{1}{\|v_1\|\|v_2\|} \sum_{x \in x_1} \sum_{y \in x_2} (c(x) \cdot c(y))d_x d_y
\]

- Evaluate on-the-fly all boxes by summing inside-box point scores and dividing the pre-computed norms. Point scores are computed once.

**Contribution II. Locality in the Feature Space**

**Proposal:** Search locally in the feature space by two tactics:

1. **Large vocabulary** which can suppress false matches and enlarge local distinction in the feature space.
   - Point-indexed representation requires constant memory, independent of vocabulary size.

2. **Exponential similarity** which gives disproportionately high similarity score when descriptors get close in the feature space, emphasizing local search.

\[
\exp(\alpha \cdot \frac{d_x d_y}{|d_x| |d_y|})
\]

* The thresholded polynomial similarity in the concurrent work \([\text{Tolias13iccv}]\) employs a similar idea.

With the large vocabulary and exponential similarity, the true correspondences pop up.

**Experiments**

- **The influence of spatial locality**
  - **global:** evaluate the entire database image \([\text{Jegou12pami}]\)
  - **tFV:** truncated Fisher vector (assign to 2 Gaussians)

- **Generic:** average mAP over the three datasets
- **vocabulary size:** 256

**The local search also provides a reliable localization of the instance.**

**State-of-the-art comparison**

- **Oxford51S**
  - **Belgian15:**
  - **Holiday:**
  - **Generic**

**The local search also provides a reliable localization of the instance.**

**Hard example**

- **query**
- **top 5 results**

**Two database images**
- 256 vocabulary
- 20k vocabulary
- 20k vocabulary & exponential similarity

**low ranked positive examples due to large variation**