Subhankari Mishra

16 December 2015

VLAD for Image Retrieval

Outline

- Introduction
- Pipeline
- Implementation
- Experiments

Introduction

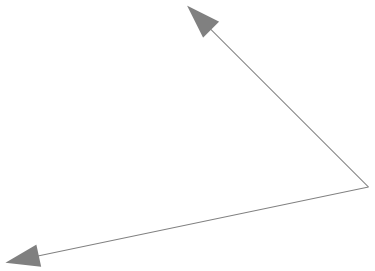
The problem can be divided into three sub-problems associated with image representation

- Should be invariant and robust to lighiting,occulsion, view-point
- Should be informative
- Efficient computation

Introduction

The problem can be divided into three sub-problems associated with image representation

- Should be invariant and robust to lighiting,occulsion, view-point
- Should be informative
- Efficient computation



Introduction

The problem can be divided into three sub-problems associated with image representation

- Should be invariant and robust to lighiting,occulsion, view-point
- Should be informative

SIFT descriptors

- Efficient computation

VLAD / Bag of Feature representation

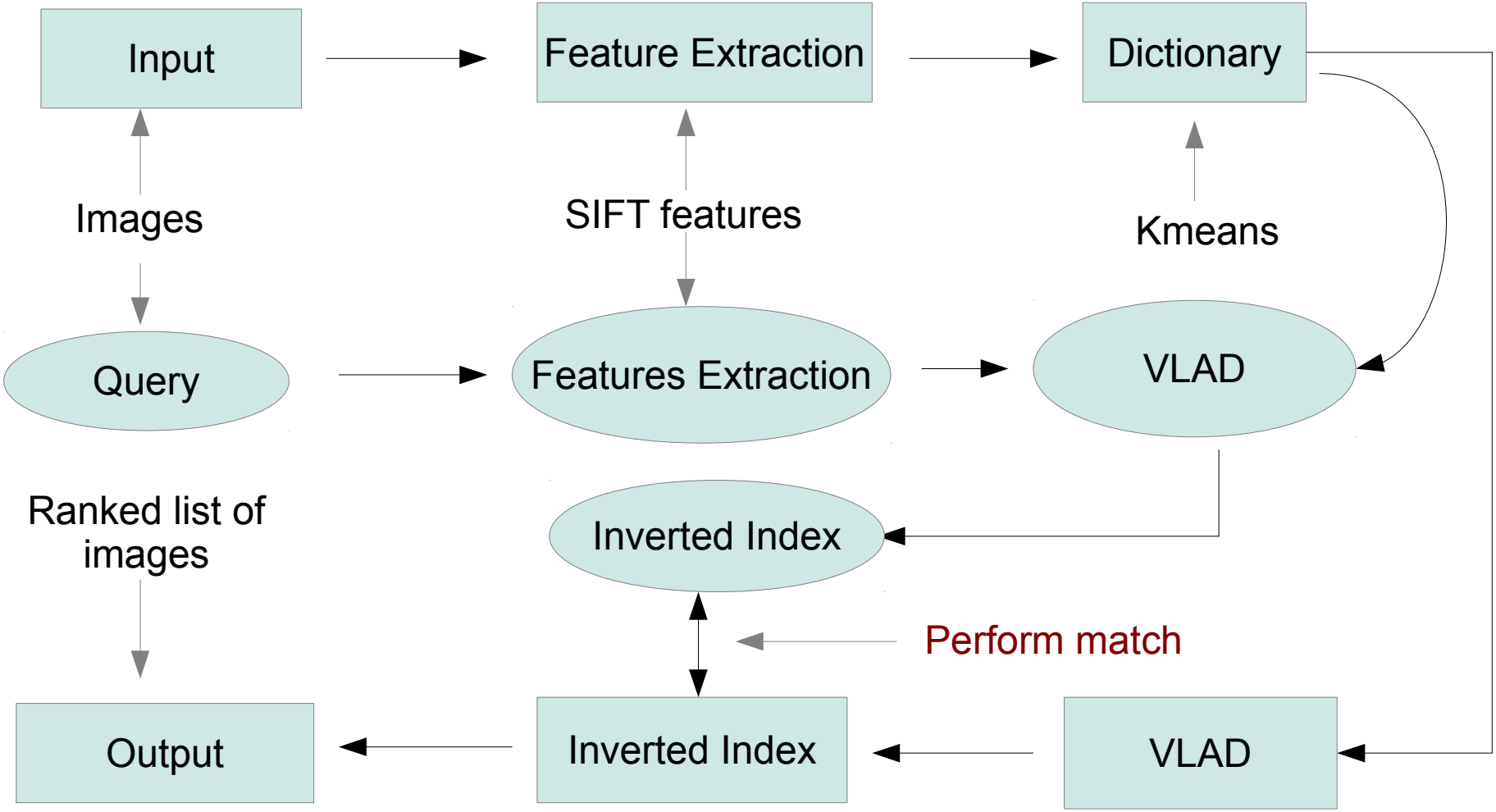
Introduction

Input: JPEG images

Query: JPEG images

Output: Ranked list of images

Pipeline


Input
Feature Extraction
Dictionary
Images
SIFT features
Kmeans
Query
Features Extraction
VLAD
Ranked list of images
Inverted Index
Perform match
Output
Inverted Index
VLAD

Implementation

Tools and languages:

- Python
- MATLAB
- Scikit-learn
- VLfeat
- Holidays evaluation package

Implementation

Dataset:

- INRIA Holiday Dataset
- Data size: 2.7 GB
- 1491 images
- 500 groups
- test and train set – random split at 50%

Implementation

Feature extraction: SIFT Features

- Vlfeat SIFT implementation in MATLAB
- RGB image ⟶ Gray scale single precision
- Output: 128D local descriptors
- Observed dimension: approx. 15000 X 128 D per image
- Resulting features matrix ⟵ approx. 12MB
- Image representaion size = 3 * image size
- Computation time = 50 seconds per image inlcuding storage of feature vectors
- Power normalize the features

Implementation

Principal Component Analysis:

- Scikit-learn's PCA
- Input: random 50 features of each image = 74550 features
- Compute the number of features that preserve at least 95% of variance
- Identified number of components to be 75
- Project the data using these 74550 features.
- Computation time:
  - Extracting random points: 6 hours
  - Applying PCA to the data set and storage: 8 hours with two jobs in parallel

Implementation

Building Dictionary: Kmeans Clustering

- Scikit-learn's Online Mini Batch Kmeans clustering.
- Max number of iterations = 1000 Batch size = 100
- Input: Randomly selected 500 local features from each image of traning set
- Output: Centroids of the clusters
- Computation time: 7200 seconds for 350000 local features

Implementation

VLAD: Vector of Locally Aggregated Descriptors

- Given a dictionary learned with k-means: $\{\mu_i, i = 1, ..., N\}$
- A set of local descriptors: $X = \{x_t, t = 1, ..., T\}$
- Assign the nearest neighbor: $NN(x_t) = \text{argmin}_{\mu_i} \parallel x_t - \mu_i \parallel$
- Distance from the cluster center: $v_i = \sum_{x_t : NN(x_t) = \mu_i} \parallel x_t - \mu_i \parallel$
- Concatenate all v_i and l_2 normalize the data
- Output: k x D dimensional representation of image
- Image representation reduced from 12MB to few Bytes.
- Computation time: 36 seconds per image including storage

Implementation

Inverted Index,Scoring and Ranking:

- More optimization towards memory consumption: Store VLADs as inverted index
- Power normalize inverted indices
- Score: Dot(query, training set)
- Rank: Sort score deseding
- Computation time: 1200 seconds for 745 queries againts 745 training images.
- Ten query takes 0.13 seconds .

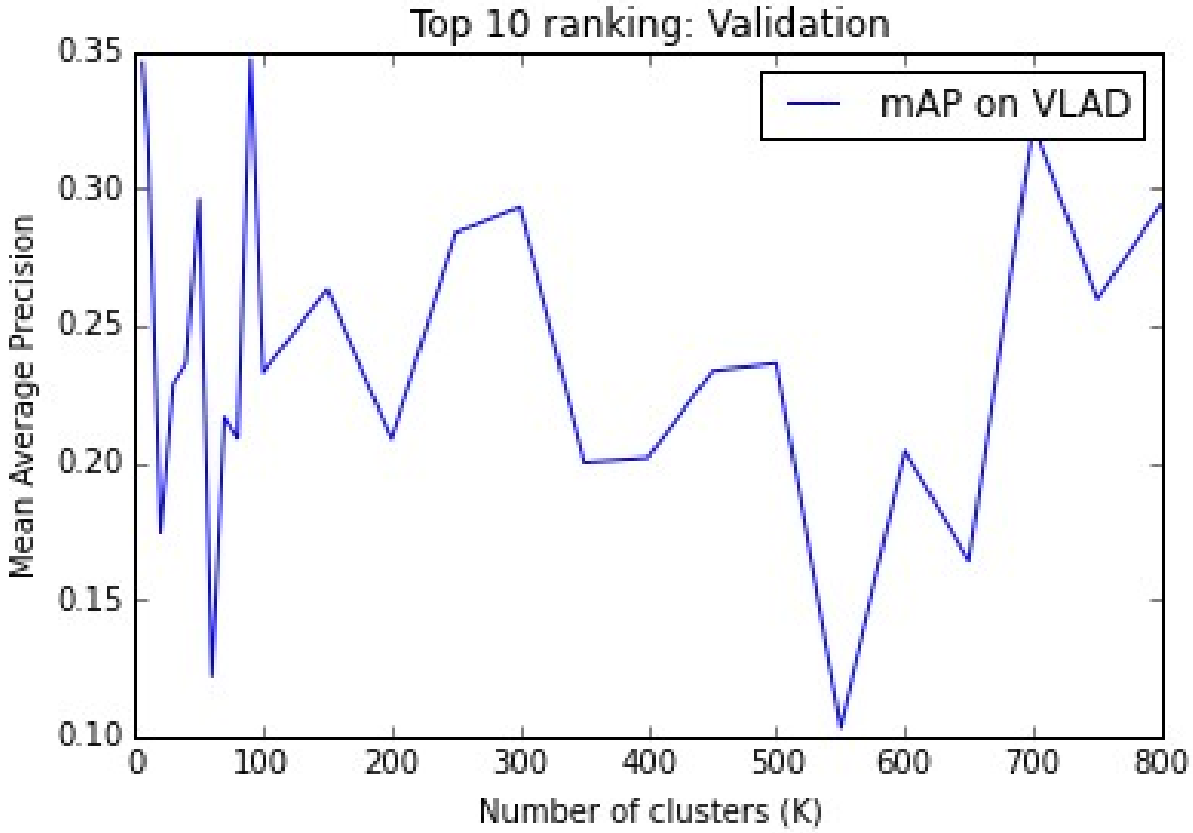
Evaluation

- 500 groups of images
- Group identified from image name
- Correct result: first 4 digits of result image = first 4 digits of query image
- Evaluation criteria used mean average precision

  Precision = fraction of retrieved documents that are relevant
- Evaluation package: Holidays evaluation package customized for the experiments

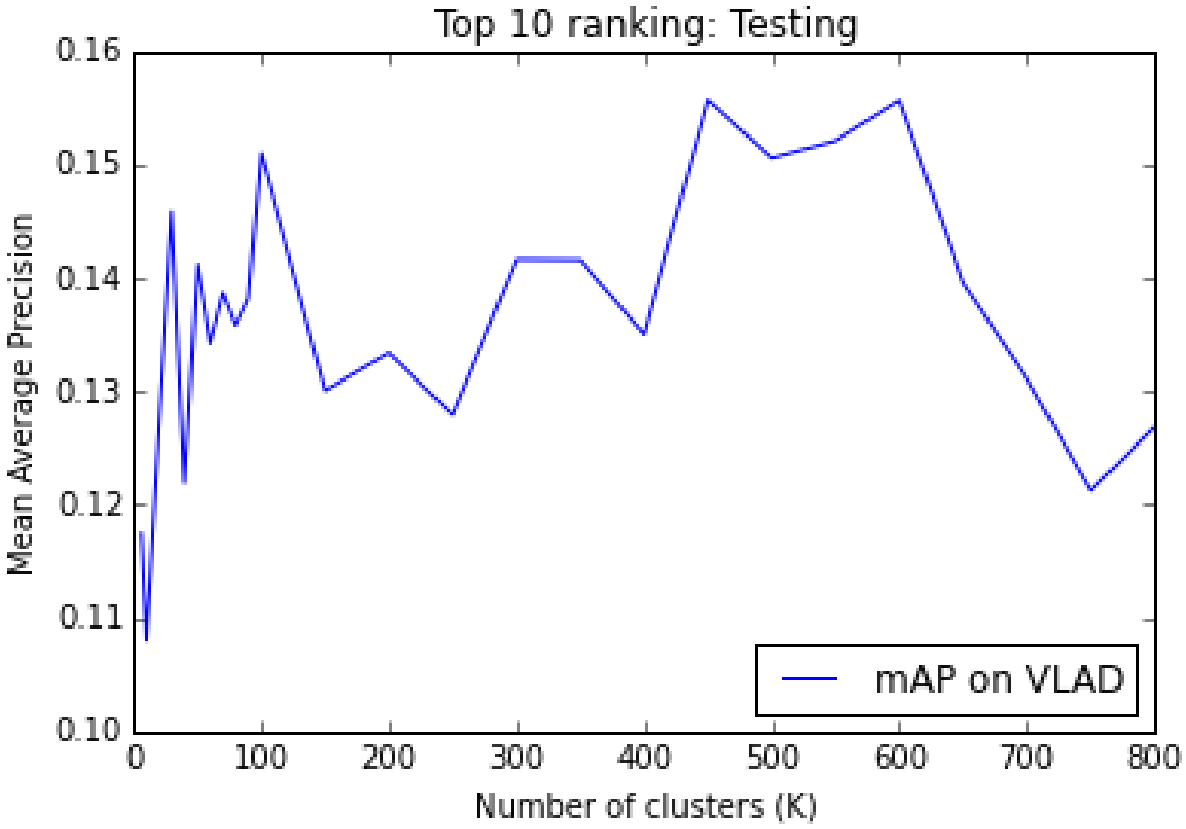
Experiments

- 745 images in traning set and test set.
- Build VLAD vectors for the training set for clusters ranging from 6 to 800.
- Cross Validation:
  - 10 random VLAD vectors/images from the training set used as query image
  - K-Means batch size = 100
  - K-Means max_iter = 1000
  - Max mean average precision achieved = 35 % for K close to 90 and 6.



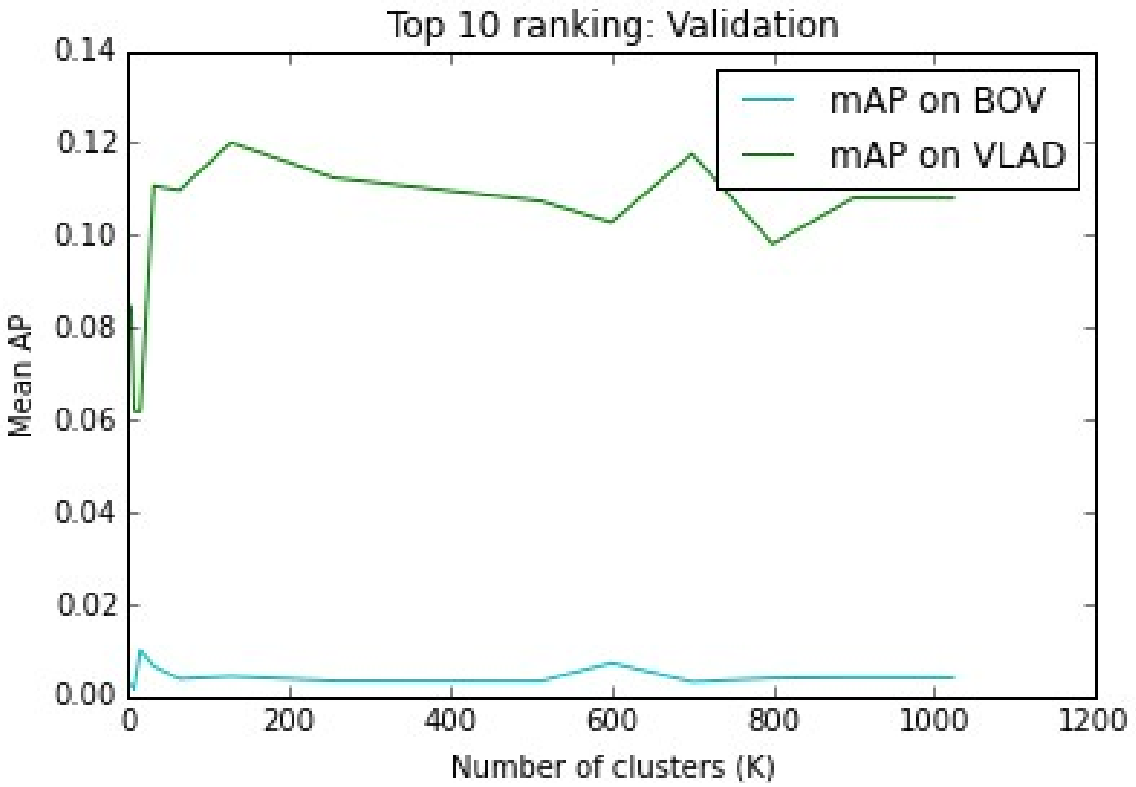
Experiments

- 745 images in traning set and test set.
- Build VLAD vectors for the test set for clusters ranging from 6 to 800.
- Test results:
  - 745 queries on 745 training set.
  - Max mean average precision achieved = 15.5% at 450 clusters.
  - K-Means batch size = 100
  - K-Means max_iter = 1000
  - Very low compared to 55% achieved in the reference.



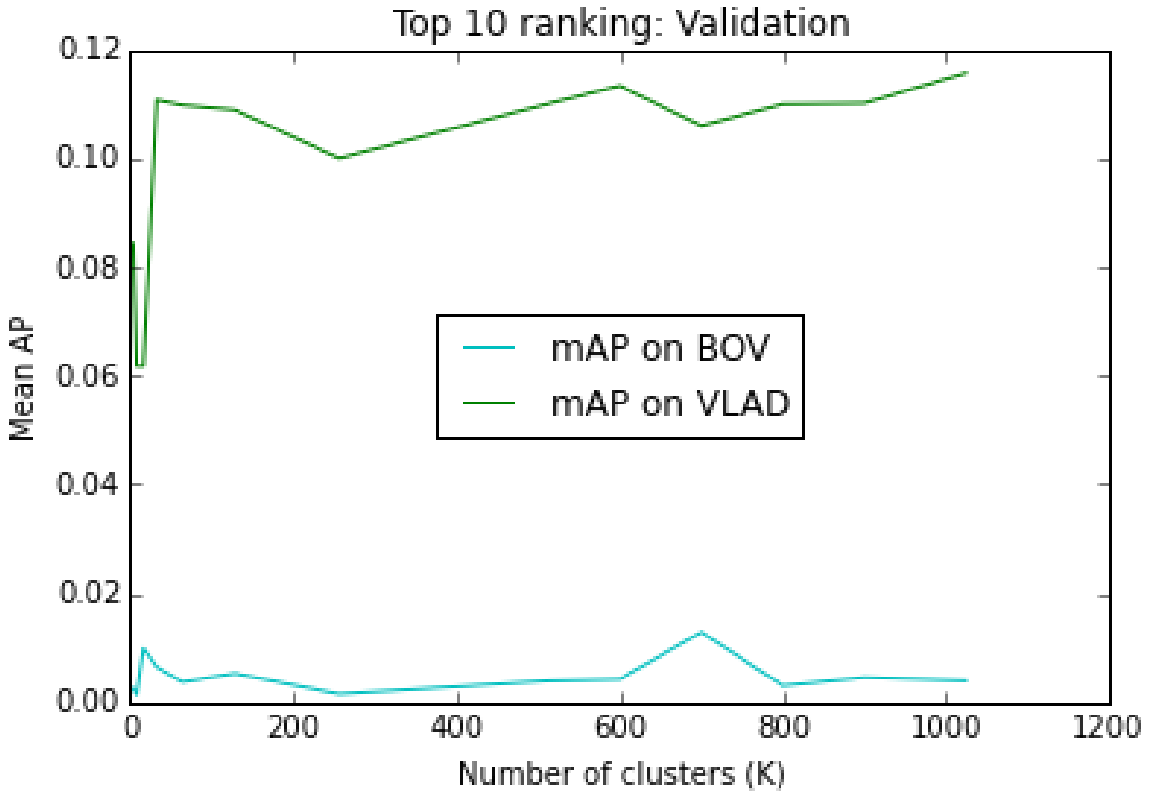
Experiments

- 500 query images 991 test images.
- Build VLAD vectors and BOV for the data with clusters ranging from 4 to 1024.
- Test results:
  - 500 queries on 991 training set.
  - K-Means batch size = default
  - K-Means max_iter = 10000
  - Max mean average precision achieved = 11% at 700 clusters.
  - Very low compared to 55% achieved in the reference.



Experiments

- 500 query images 991 test images.
- Build VLAD vectors and BOV for the data with clusters ranging from 4 to 1024.
- Test results:
  - 500 queries on 991 training set.
  - K-Means batch size = default
  - K-Means max_iter = default
  - Max mean average precision achieved = 12% at 1024 clusters.
  - Very low compared to 55% achieved in the reference.



Thank You !!