Lec8 - Representations for Object Categorization

CS 783 - Visual Recognition

Vinay P. Namboodiri

IIT Kanpur

29^{th} January 2019

Contents

Outline

What is Object Categorization

- In this problem we are given a database of sets of images for training. Each set contains images that denotes a single kind of object. This could be a thing or an animal. The name given to that set is termed the category label. Each set has a unique label in the case of object categorization.

What is Object Categorization

- In this problem we are given a database of sets of images for training. Each set contains images that denotes a single kind of object. This could be a thing or an animal. The name given to that set is termed the category label. Each set has a unique label in the case of object categorization.
- Given a test image, we need to predict the correct category label to which the image belongs.

What is Object Categorization

- In this problem we are given a database of sets of images for training. Each set contains images that denotes a single kind of object. This could be a thing or an animal. The name given to that set is termed the category label. Each set has a unique label in the case of object categorization.
- Given a test image, we need to predict the correct category label to which the image belongs.
- During training we learn a function $f_w(x) \rightarrow y$ that maps an instance x to the appropriate category label y using a parameter vector w that is learnt during training.

What is Object Categorization

- In this problem we are given a database of sets of images for training. Each set contains images that denotes a single kind of object. This could be a thing or an animal. The name given to that set is termed the category label. Each set has a unique label in the case of object categorization.
- Given a test image, we need to predict the correct category label to which the image belongs.
- During training we learn a function $f_w(x) \rightarrow y$ that maps an instance x to the appropriate category label y using a parameter vector w that is learnt during training.
- This learnt function $f_w(x_i)$ is then used to predict the category label y_i for a test sample x_i using the learnt parameter vector w.

Example: Object Categorization

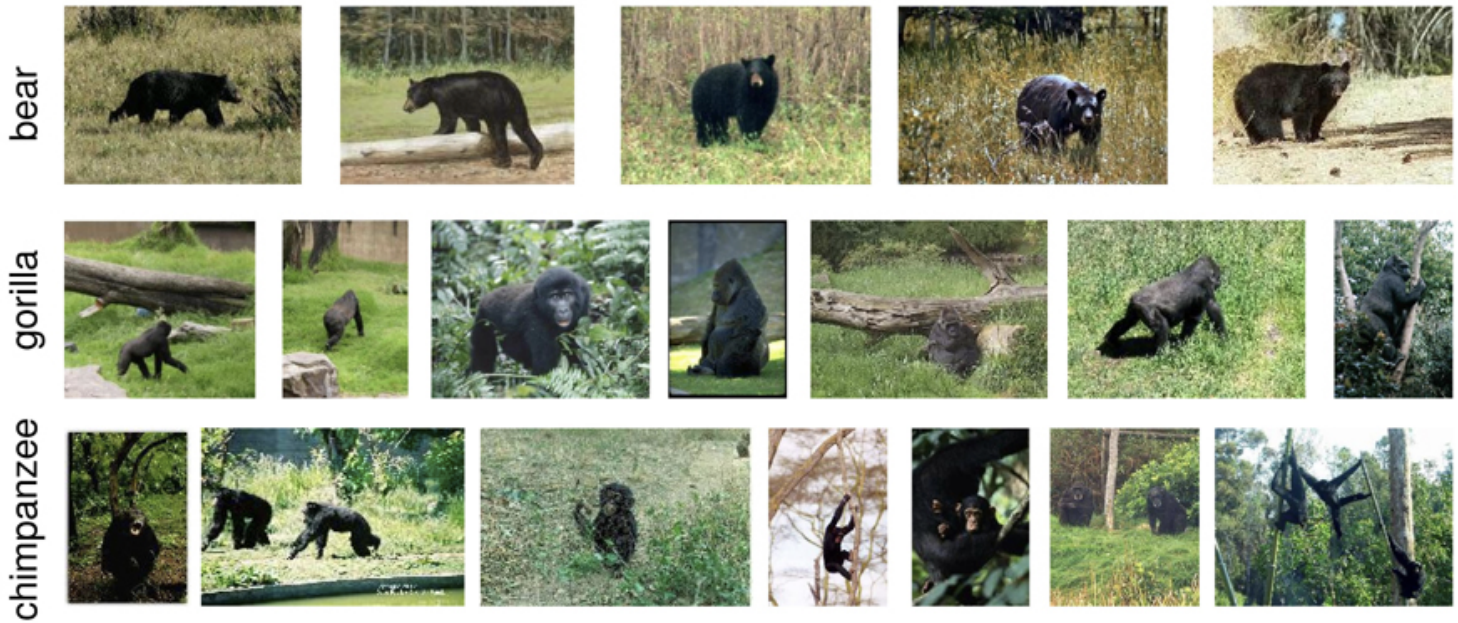


Figure: Illustration of Classification among various categories

Outline

Approach for Object Categorisation

Representation

Approach for Object Categorisation

Representation

- Given a set of images, obtain an appropriate representation for each image that captures the essence of the image.

Approach for Object Categorisation

Representation

- Given a set of images, obtain an appropriate representation for each image that captures the essence of the image.

Classification

Approach for Object Categorisation

Representation

- Given a set of images, obtain an appropriate representation for each image that captures the essence of the image.

Classification

- Learn an appropriate classifier that uses the representation and learns an appropriate parameter vector w.

Approach for Object Categorisation

Representation

- Given a set of images, obtain an appropriate representation for each image that captures the essence of the image.

Classification

- Learn an appropriate classifier that uses the representation and learns an appropriate parameter vector w.

Recognition/Inference

Approach for Object Categorisation

Representation

- Given a set of images, obtain an appropriate representation for each image that captures the essence of the image.

Classification

- Learn an appropriate classifier that uses the representation and learns an appropriate parameter vector w.

Recognition/Inference

- Use the learnt parameter vector for predicting the target label using the learnt classification parameter vector w for a test sample representation x.

Outline

Bag of Words Representation

Feature Extraction

Bag of Words Representation

Feature Extraction

- Given an image, we extract the appropriate set of features.

Bag of Words Representation

Feature Extraction

- Given an image, we extract the appropriate set of features.

Clustering

Bag of Words Representation

Feature Extraction

- Given an image, we extract the appropriate set of features.

Clustering

- We obtain a visual word vocabulary using clustering. This results in quantization of the set of words based on the centroids obtained.

Bag of Words Representation

Feature Extraction

- Given an image, we extract the appropriate set of features.

Clustering

- We obtain a visual word vocabulary using clustering. This results in quantization of the set of words based on the centroids obtained.

Representation

Bag of Words Representation

Feature Extraction

- Given an image, we extract the appropriate set of features.

Clustering

- We obtain a visual word vocabulary using clustering. This results in quantization of the set of words based on the centroids obtained.

Representation

- In this we use the visual word vocabulary to obtain a fixed size histogram representation for an image. This is obtained based on the count of words that are assigned to each centroid.

Feature extraction

Figure: Illustration of feature extraction from an image. One example is using SIFT features

Visual Vocabulary

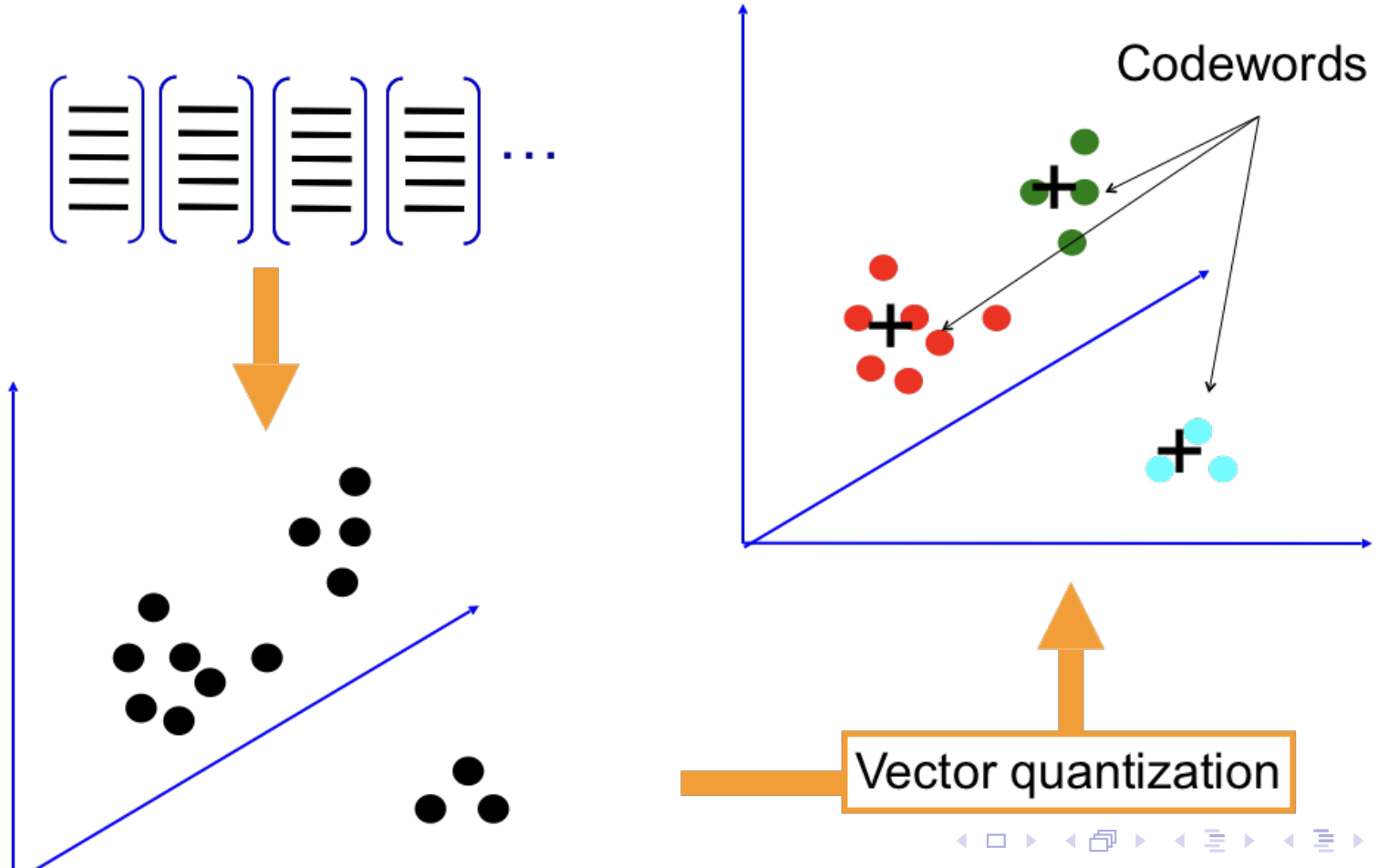
All the features extracted from the set of training images are clustered using the kmeans algorithm as follows to obtain the visual vocabulary set. The cardinality of the set of centroids can be a fixed one for instance using 4096 centroids.

Algorithm 1 Algorithm to obtain visual word vocabulary using K-Means

Result: Clustering using K-Means for obtaining Visual Word Vocabulary
Input: A set of visual words V_i over all images D
Output: A set of visual word centroids C_k that is the set of k visual vocabulary words
Initialization: Randomly initialize C_j^0 to a set of visual words that are randomly chosen
while $C_j^{t+1} - C_j^t > \epsilon$ **do**
    $\forall\ V_i$, obtain distance d_{ij} from each centroid C_j
    Assign V_i to the nearest cluster centroid C_j^t to which distance d_{ij} is minimum;
    For each cluster j obtain the new centroid by obtaining **mean**($V_i \in$ cluster j) ;
    obtain new centroid $C_j^{t+1} =$ **mean**$_j$;
end

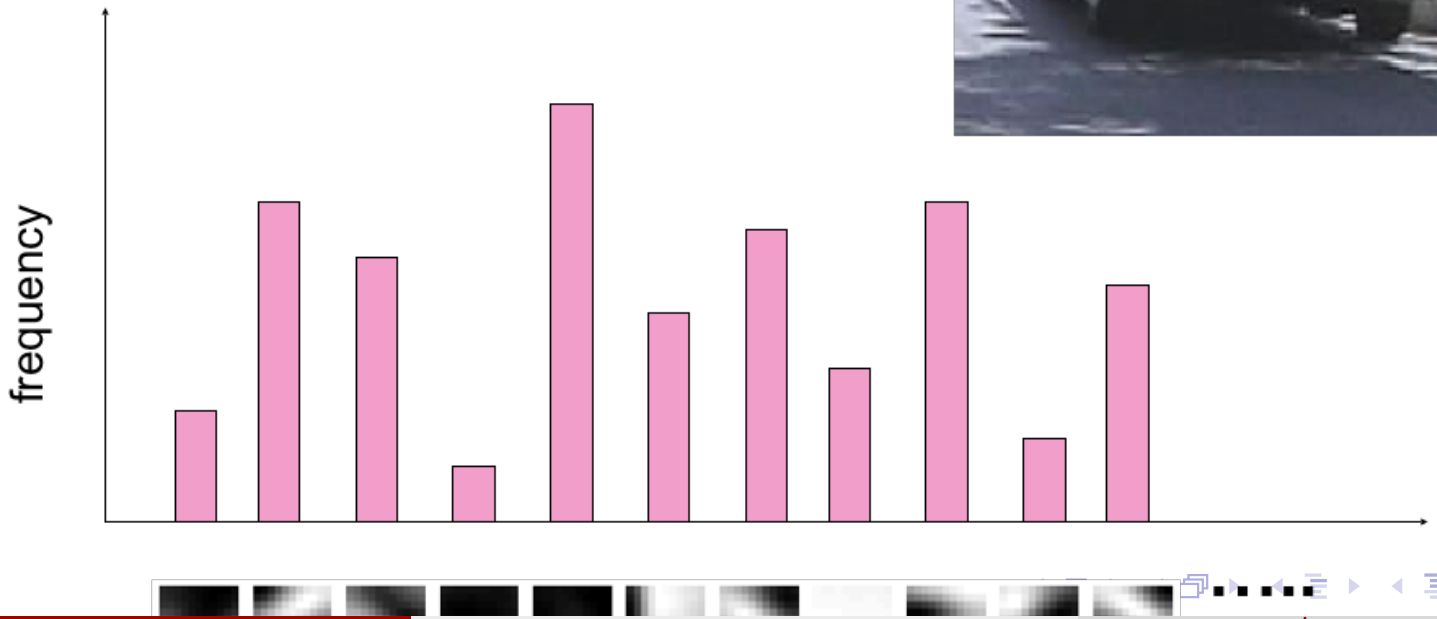
Vector Quantization

The visual vocabulary is used for vector quantization. This is the procedure where each sift word is assigned to the closest centroid from the visual vocabulary to which it is closest.



Bag of Words Representation

The quantized visual words are used to form a histogram of visual words. This representation is termed the bag of visual words representation. It results in a fixed length histogram. Typically it is normalized to have unit norm.



BoW Representation

Pros

- Simple and easy to obtain

Cons

BoW Representation

Pros

- Simple and easy to obtain
- Surprisingly effective and can be used in other settings as well such as action recognition

Cons

BoW Representation

Pros

- Simple and easy to obtain
- Surprisingly effective and can be used in other settings as well such as action recognition

Cons

- Suffers due to quantization

BoW Representation

Pros

- Simple and easy to obtain
- Surprisingly effective and can be used in other settings as well such as action recognition

Cons

- Suffers due to quantization
- Coarse approximation to the actual image representation

Outline

Pyramid Match Kernel: Motivation

- We would like to obtain the approximate an optimal matching algorithm efficiently.

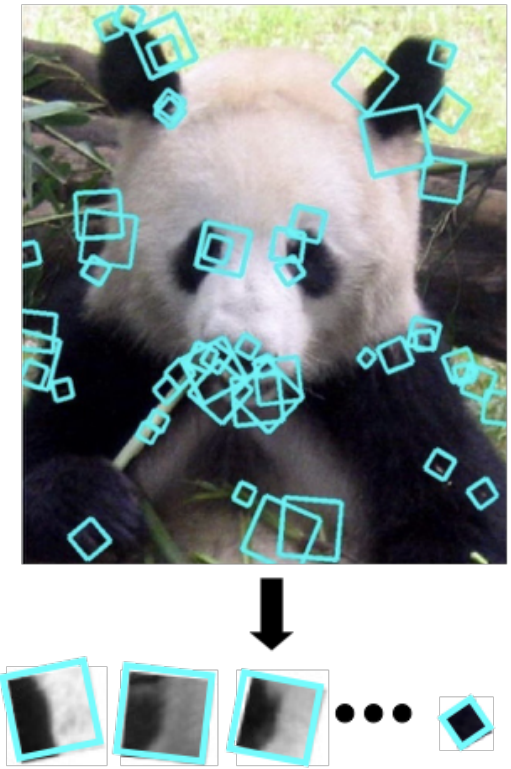
Pyramid Match Kernel: Motivation

- We would like to obtain the approximate an optimal matching algorithm efficiently.
- An optimal algorithm would be one where for a pair of images we obtain the exact correspondence between the set of visual words that match exactly.

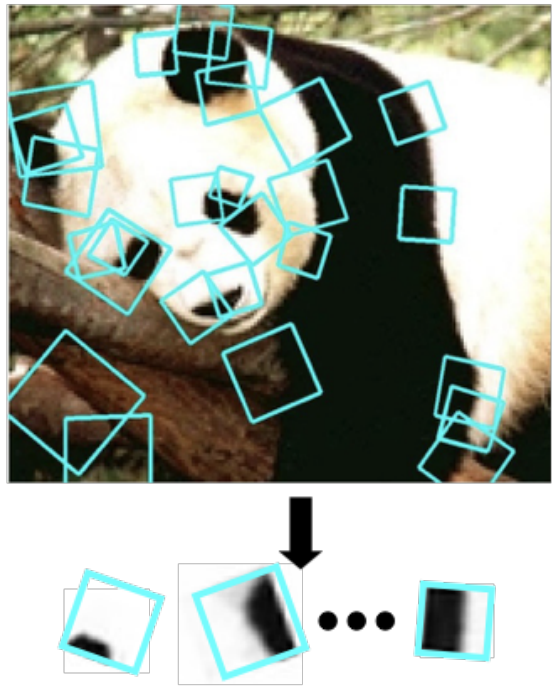
Pyramid Match Kernel: Motivation

- We would like to obtain the approximate an optimal matching algorithm efficiently.
- An optimal algorithm would be one where for a pair of images we obtain the exact correspondence between the set of visual words that match exactly.
- However, such an optimal algorithm would be very expensive and be of the order of dm^2 where d is the dimension of the feature and m is the cardinality of the number of features. Assuming 2000 points, and 128 dimensions this amounts to 512,000,000.

Pyramid Match Kernel: Idea

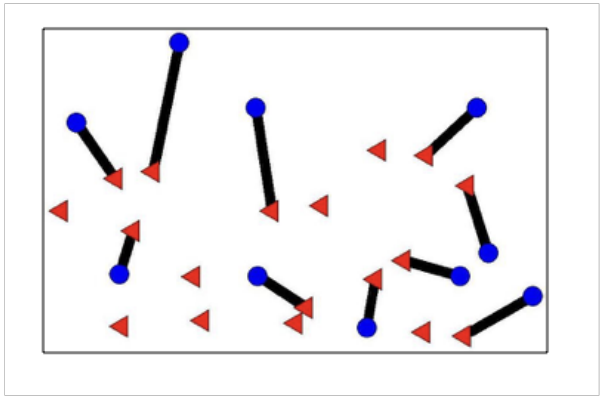


$$\mathbf{X} = \{\vec{\mathbf{x}}_1, \ldots, \vec{\mathbf{x}}_m\}$$



$$\mathbf{Y} = \{\vec{\mathbf{y}}_1, \ldots, \vec{\mathbf{y}}_n\}$$

Pyramid Match Kernel: Idea



Pyramid Match Kernel: Idea

- Pyramid match kernel measures similarity of a partial matching between two sets

Pyramid Match Kernel: Idea

- Pyramid match kernel measures similarity of a partial matching between two sets
- Place multi-dimensional, multi-resolution grid over point sets

Pyramid Match Kernel: Idea

- Pyramid match kernel measures similarity of a partial matching between two sets
- Place multi-dimensional, multi-resolution grid over point sets
- Consider points matched at finest resolution where they fall into same grid cell

Pyramid Match Kernel: Idea

- Pyramid match kernel measures similarity of a partial matching between two sets
- Place multi-dimensional, multi-resolution grid over point sets
- Consider points matched at finest resolution where they fall into same grid cell
- Approximate similarity between matched points with worst case similarity at given level

Pyramid Match Kernel: Idea

- Pyramid match kernel measures similarity of a partial matching between two sets
- Place multi-dimensional, multi-resolution grid over point sets
- Consider points matched at finest resolution where they fall into same grid cell
- Approximate similarity between matched points with worst case similarity at given level
- It involves no explicit search for matches

Pyramid Match Kernel: Idea

Number of newly matched pairs at level *i*

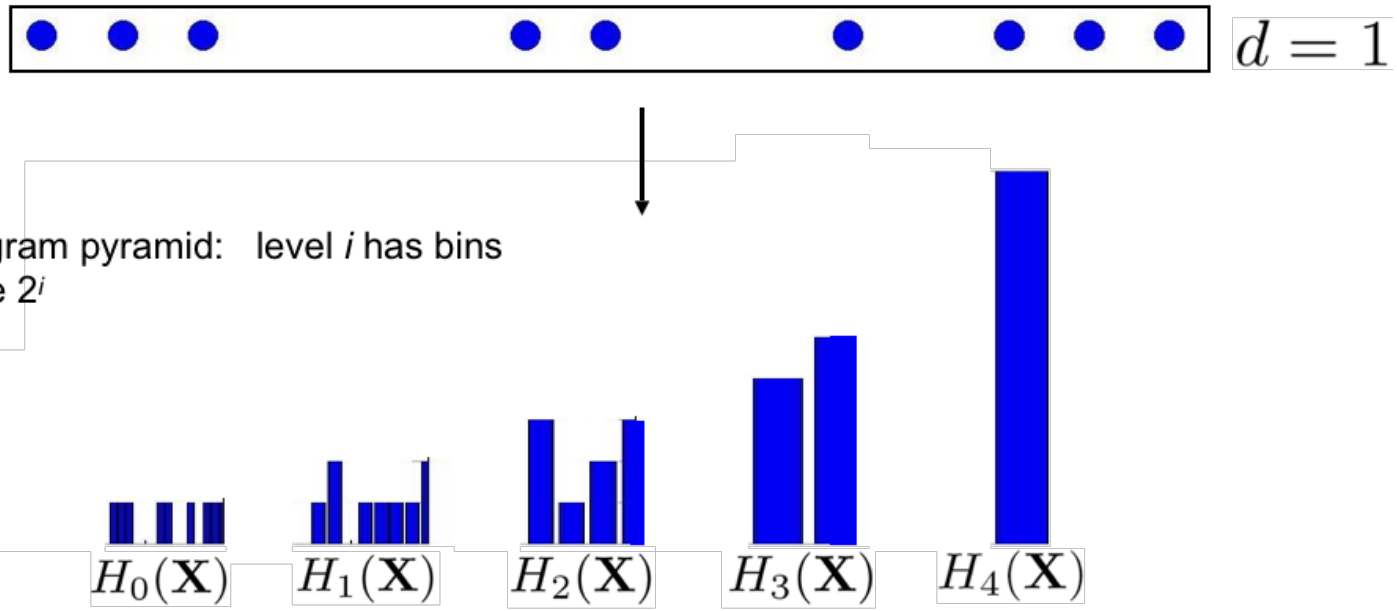
Approximate partial match similarity

$$K_\Delta = \sum_{i=0}^{L} w_i N_i$$

Measure of difficulty of a match at level *i*

Pyramid Match Kernel: Idea

$$\mathbf{X} = \{\vec{\mathbf{x}}_1, \ldots, \vec{\mathbf{x}}_m\}, \quad \vec{\mathbf{x}}_i \in \Re^d$$

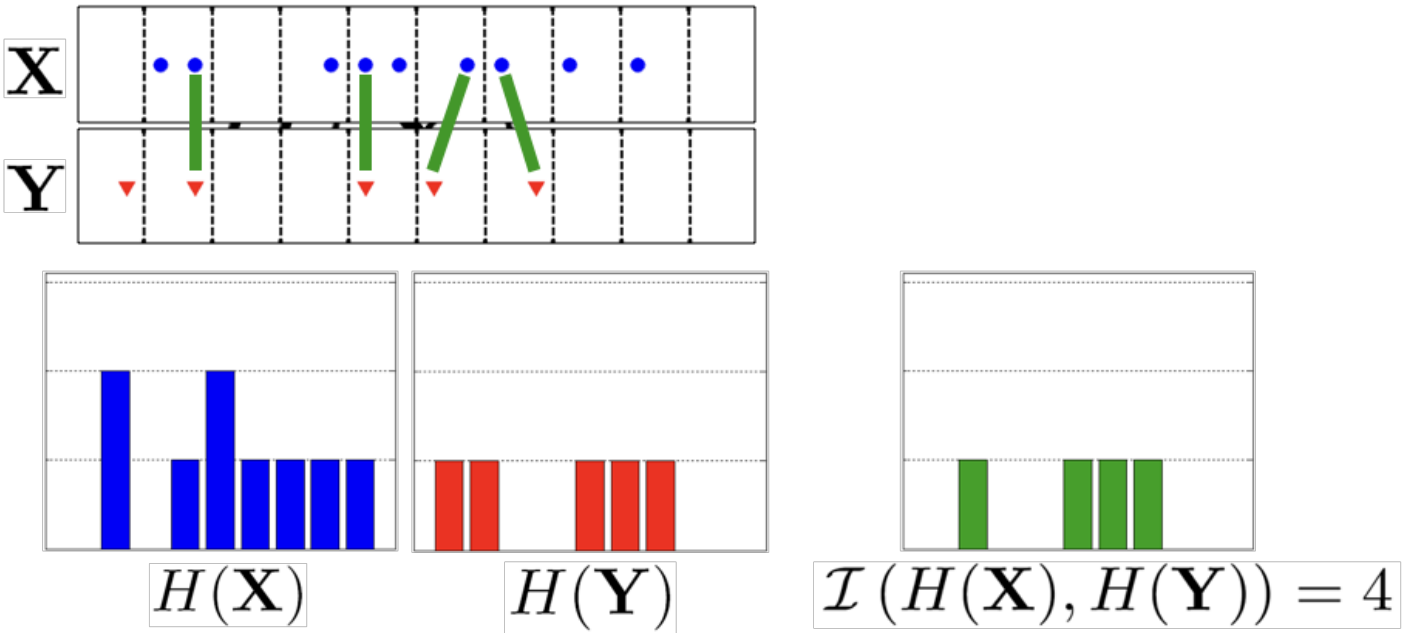


$$\Psi(\mathbf{X}) = [H_0(\mathbf{X}), \ldots, H_L(\mathbf{X})]$$

Pyramid Match Kernel: Idea

Histogram intersection

$$\mathcal{I}\left(H(\mathbf{X}), H(\mathbf{Y})\right) = \sum_{j=1}^{r} \min\left(H(\mathbf{X})_j, H(\mathbf{Y})_j\right)$$



Pyramid Match Kernel: Idea

Histogram intersection

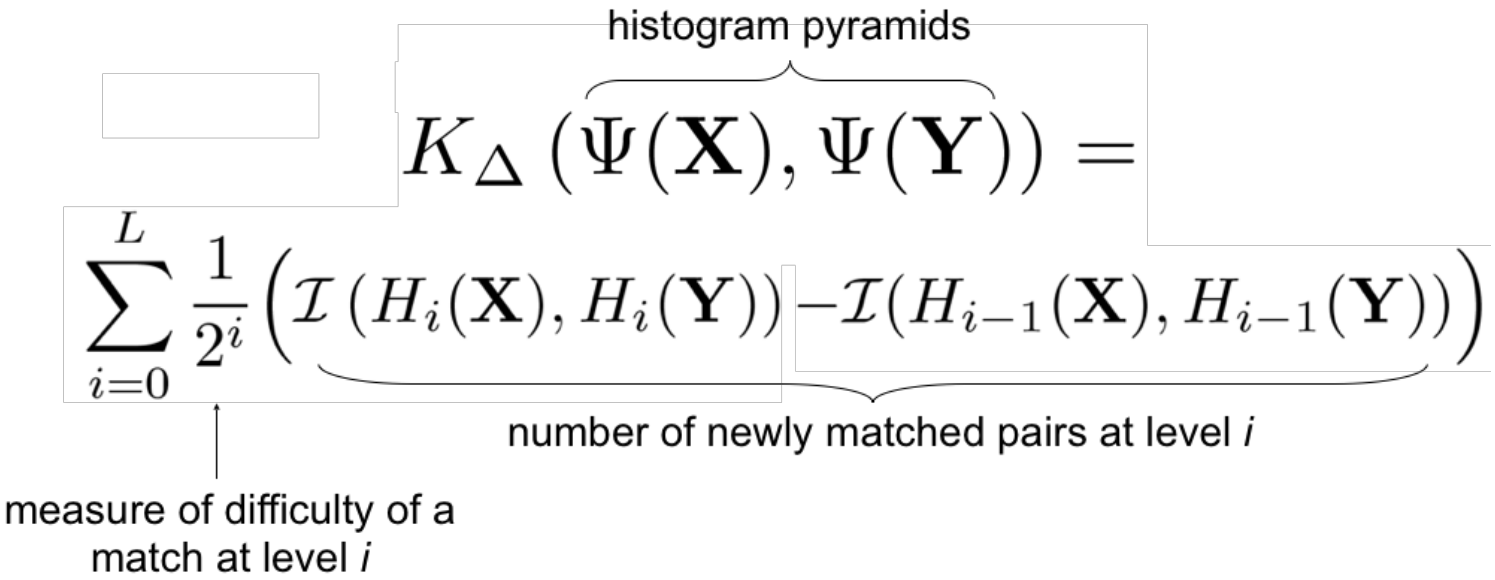
$$\mathcal{I}\left(H(\mathbf{X}), H(\mathbf{Y})\right) = \sum_{j=1}^{r} \min\left(H(\mathbf{X})_j, H(\mathbf{Y})_j\right)$$

matches at this level — matches at previous level

$$N_i = \mathcal{I}\left(H_i(\mathbf{X}), H_i(\mathbf{Y})\right) - \mathcal{I}\left(H_{i-1}(\mathbf{X}), H_{i-1}(\mathbf{Y})\right)$$

Difference in histogram intersections across levels counts *number of new pairs* matched

Pyramid Match Kernel: Idea



- Weights inversely proportional to bin size
- Normalize kernel values to avoid favoring large sets

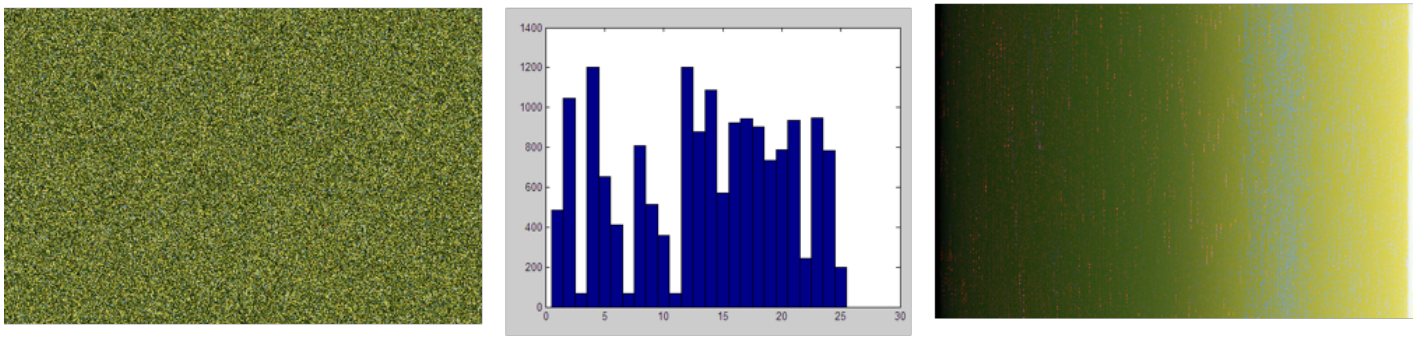
Outline

Spatial Pyramid Matching

- A limitation of the Pyramid match kernel is that it ignores the spatial layout of the features.

Spatial Pyramid Matching

- A limitation of the Pyramid match kernel is that it ignores the spatial layout of the features.



- All of these images have the same color histogram

Spatial Pyramid Matching: Motivation

Spatial pyramid is motivated by the idea of locally orderliess matching by Koenderink and Van Doorn (IJCV 1999) where the authors defined the idea of a locally orderless representation.

Inspiration: *locally orderless images* Koenderink & Van Doorn (1999)

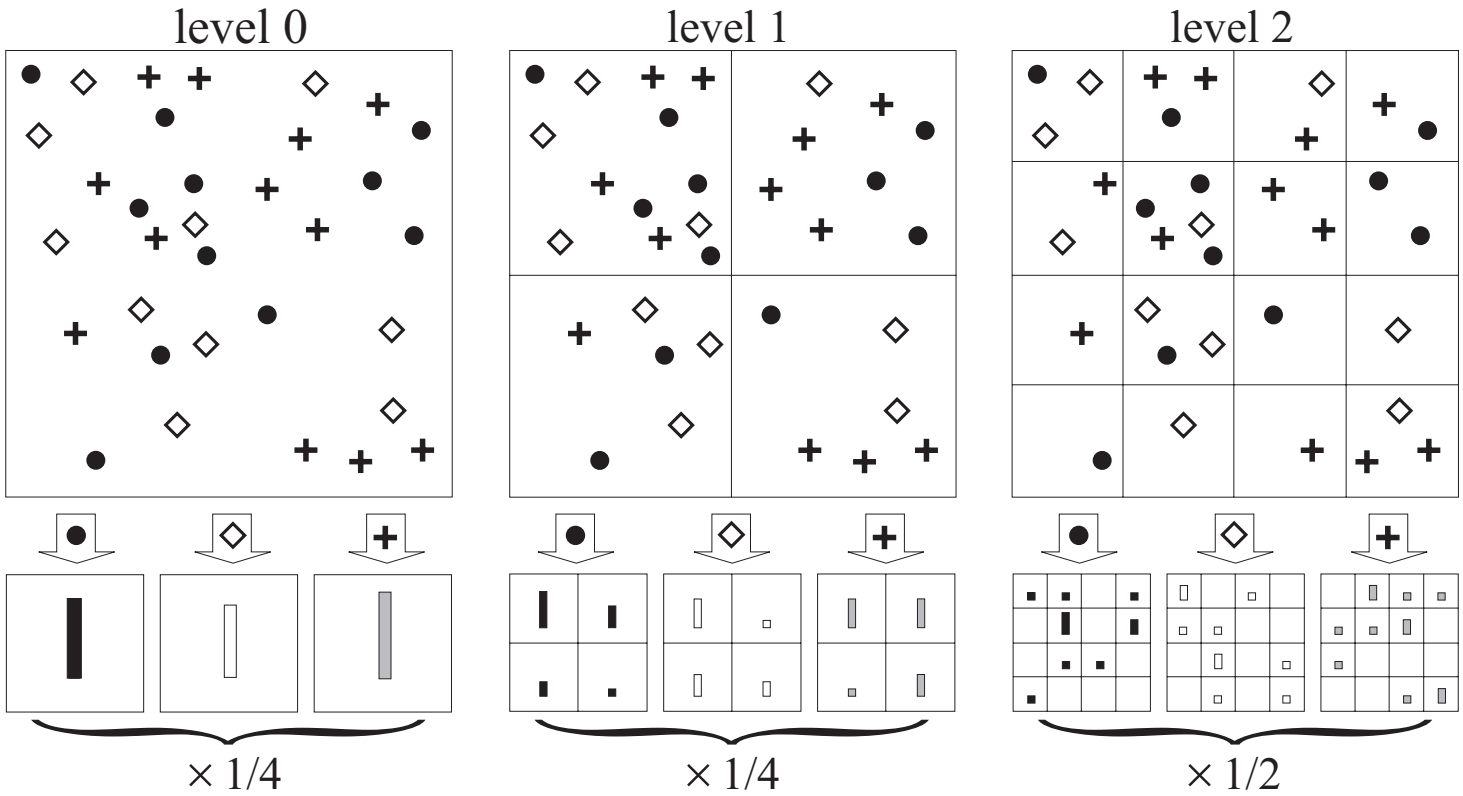
Spatial Pyramid Matching: Idea

Motivated by the idea of locally orderless images, the authors proposed the notion of using a histogram representation for the bag of words in a cell as the image representation.

Compute histogram in each spatial bin

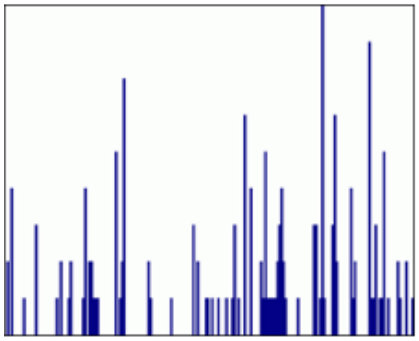
Spatial Pyramid Matching: Idea

This idea is illustrated in this toy example, where we consider the features as being different features such as cross or diamond and we obtain a pyramidal spatial layout and corresponding feature histogram representation



Spatial Pyramid Matching: Idea

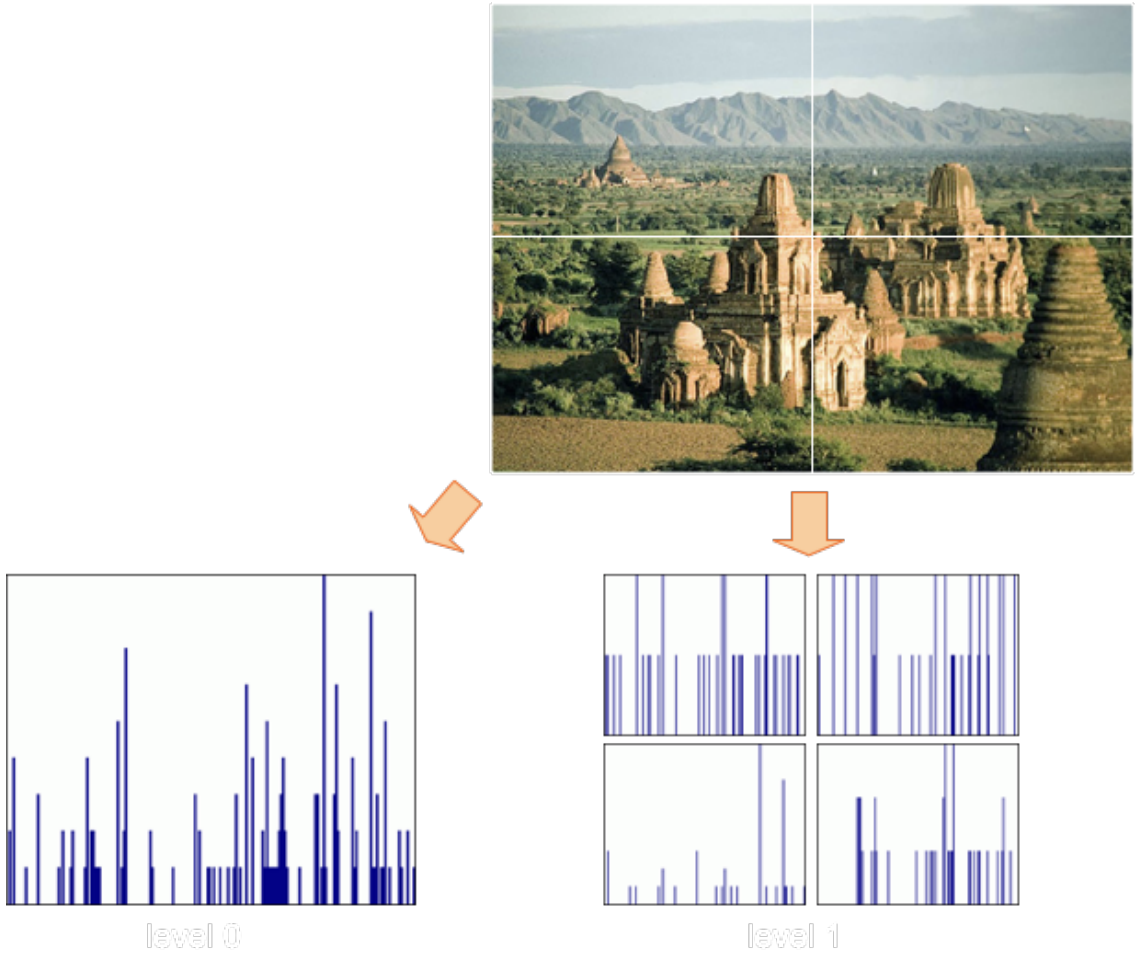
Initially for an image we would obtain a bag of words representation for the whole image



level 0

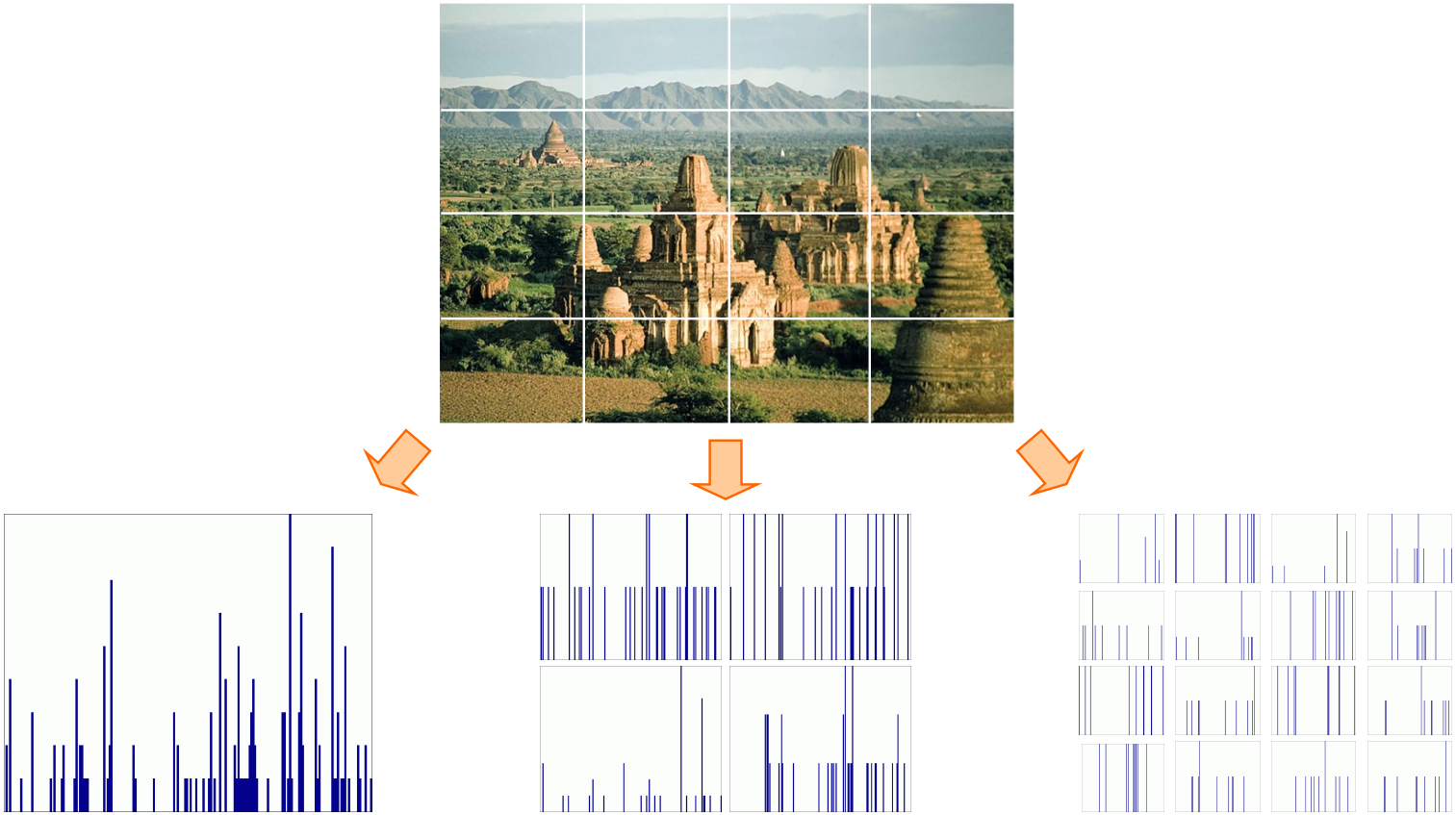
Spatial Pyramid Matching: Idea

We would then split the image into four parts to obtain the first decomposition and the bag of words in each of these parts.



Spatial Pyramid Matching: Idea

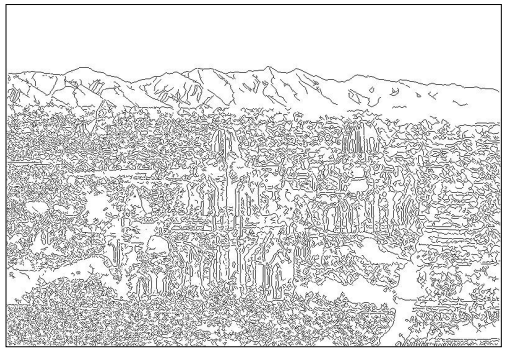
We would then split the image into 16 parts to obtain the second decomposition and the bag of words in each of these parts.



Spatial Pyramid Matching: Features

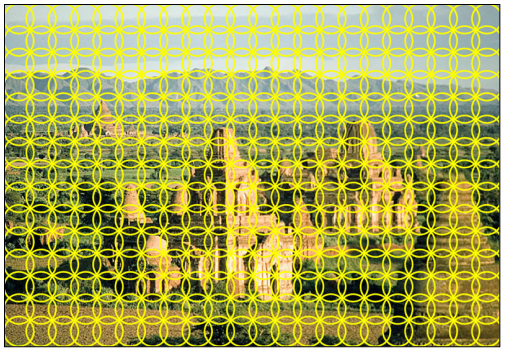
The authors evaluated two sets of features, weak features that were obtained from edge points and strong features that were obtained by regularly spaced sift descriptors.

Weak features



Edge points at 2 scales and 8 orientations (vocabulary size 16)

Strong features



SIFT descriptors of 16x16 patches sampled on a regular grid, quantized to form visual vocabulary (size 200, 400)

Spatial Pyramid Matching: Kernel

The kernel and matching approach followed exactly the pyramid matching kernel approach with the pyramid being spatial pyramids in this case and the histograms being over the same visual vocabulary of all words at each level

$$K_\Delta\left(\overbrace{\Psi(\mathbf{X}), \Psi(\mathbf{Y})}^{\text{histogram pyramids}}\right) = \sum_{i=0}^{L} \frac{1}{2^i}\Big(\underbrace{\mathcal{I}\left(H_i(\mathbf{X}), H_i(\mathbf{Y})\right) - \mathcal{I}\left(H_{i-1}(\mathbf{X}), H_{i-1}(\mathbf{Y})\right)}_{\text{number of newly matched pairs at level } i}\Big)$$

$\frac{1}{2^i}$: measure of difficulty of a match at level i

- Weights inversely proportional to bin size
- Normalize kernel values to avoid favoring large sets

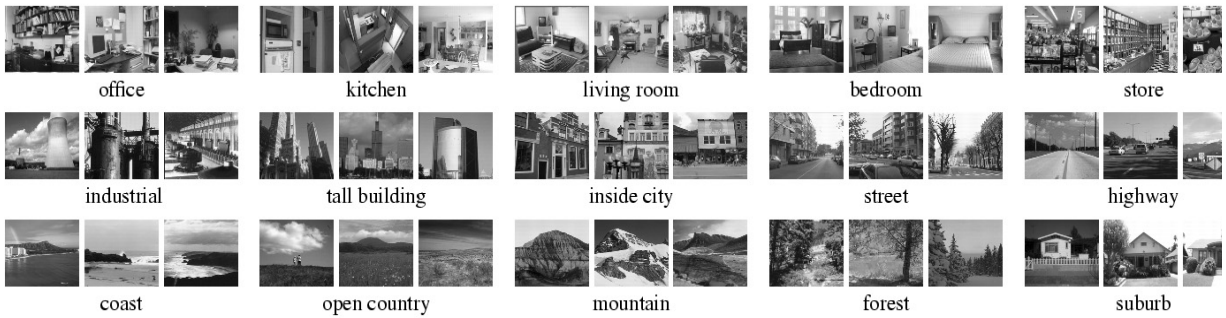
Spatial Pyramid Matching: Results

The method was evaluated on two datasets, scene category recognition and object category recognition

Scene category dataset

Fei-Fei & Perona (2005), Oliva & Torralba (2001)

`http://www-cvr.ai.uiuc.edu/ponce_grp/data`



Multi-class classification results (100 training images per class)

	Weak features (vocabulary size: 16)		Strong features (vocabulary size: 200)	
Level	Single-level	Pyramid	Single-level	Pyramid
0 (1×1)	45.3 $\pm$0.5		72.2 $\pm$0.6	
1 (2×2)	53.6 $\pm$0.3	56.2 $\pm$0.6	77.9 $\pm$0.6	79.0 $\pm$0.5
2 (4×4)	61.7 $\pm$0.6	64.7 $\pm$0.7	79.4 $\pm$0.3	**81.1** $\pm$0.3
3 (8×8)	63.3 $\pm$0.8	**66.8** $\pm$0.6	77.2 $\pm$0.4	80.7 $\pm$0.3

Fei-Fei & Perona: 65.2%

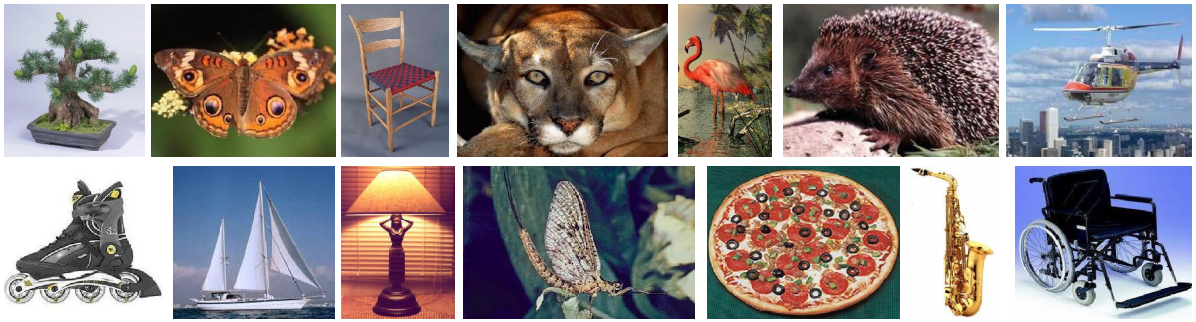
Spatial Pyramid Matching: Results

The method was evaluated on two datasets, scene category recognition and object category recognition and was observed to be better than the Pyramid match kernel approach

Caltech101 dataset

Fei-Fei et al. (2004)

http://www.vision.caltech.edu/Image_Datasets/Caltech101/Caltech101.html



Multi-class classification results (30 training images per class)

	Weak features (16)		Strong features (200)	
Level	Single-level	Pyramid	Single-level	Pyramid
0	15.5 ±0.9		41.2 ±1.2	
1	31.4 ±1.2	32.8 ±1.3	55.9 ±0.9	57.0 ±0.8
2	47.2 ±1.1	49.3 ±1.4	63.6 ±0.9	**64.6** ±0.8
3	52.2 ±0.8	**54.0** ±1.1	60.3 ±0.9	64.6 ±0.7

Outline

How to Improve Representation further

While spatial pyramid representation was performing well, however, the results for object recognition was around 65% accuracies on one of the easy datasets i.e. Caltech 101.

How to Improve Representation further

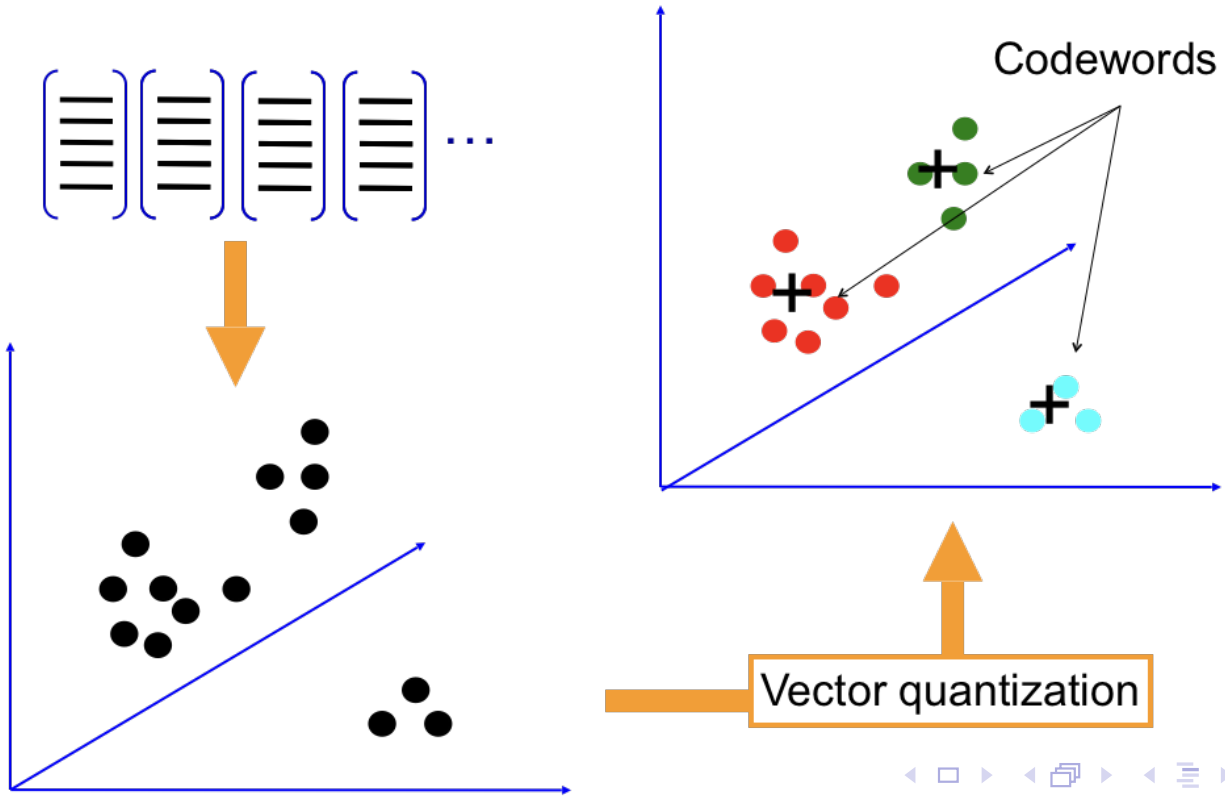
While spatial pyramid representation was performing well, however, the results for object recognition was around 65% accuracies on one of the easy datasets i.e. Caltech 101.

One of the ways to improve the accuracy was to consider the problem of vector quantization and the loss due to that.

How to Improve Representation further

While spatial pyramid representation was performing well, however, the results for object recognition was around 65% accuracies on one of the easy datasets i.e. Caltech 101.

One of the ways to improve the accuracy was to consider the problem of vector quantization and the loss due to that.

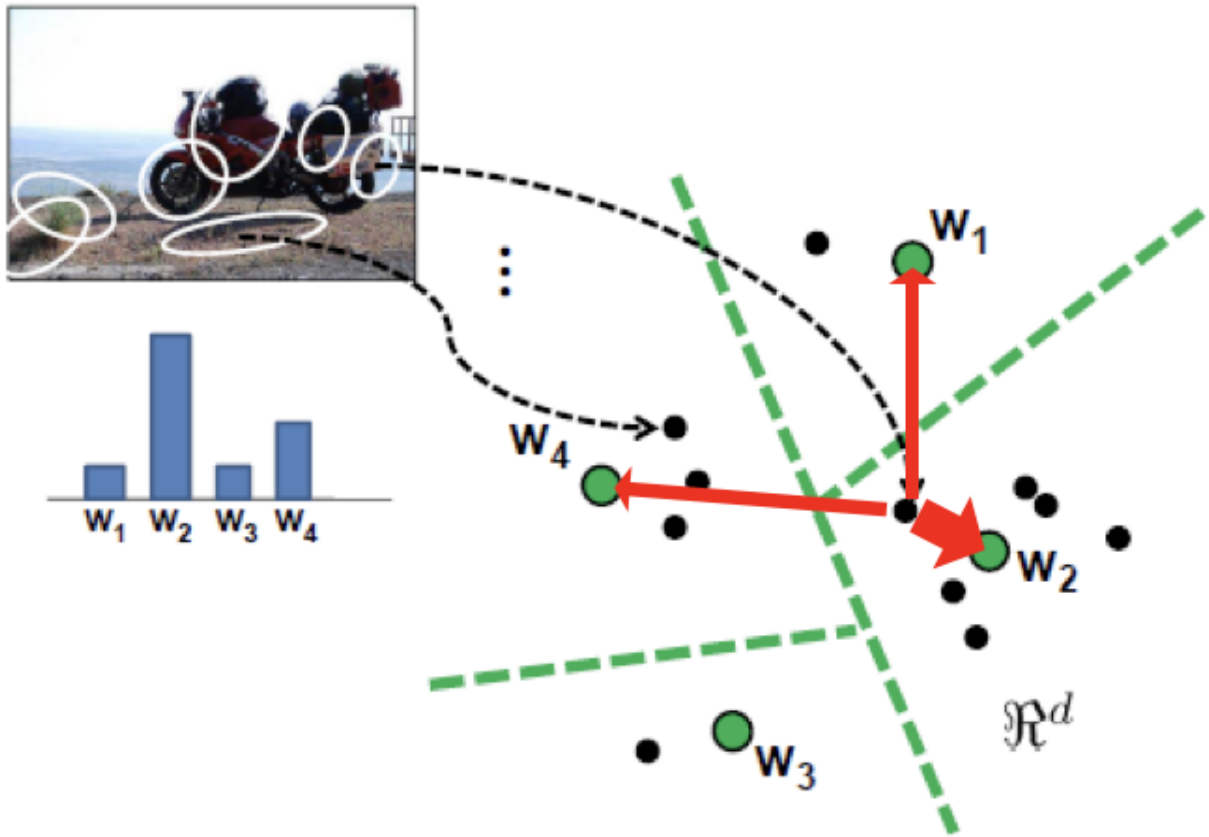


Kernel Codebook Encoding

An approach proposed to improve this was to consider weighted matching of a word to the nearest related words that allowed for soft assignment of sift features to the centroids

Kernel Codebook Encoding

An approach proposed to improve this was to consider weighted matching of a word to the nearest related words that allowed for soft assignment of sift features to the centroids



VLAD

A better approach was to consider the variance of the sift features after considering the mean. This allowed to aggregate the quantization error in each dimension and explicitly represent it

VLAD

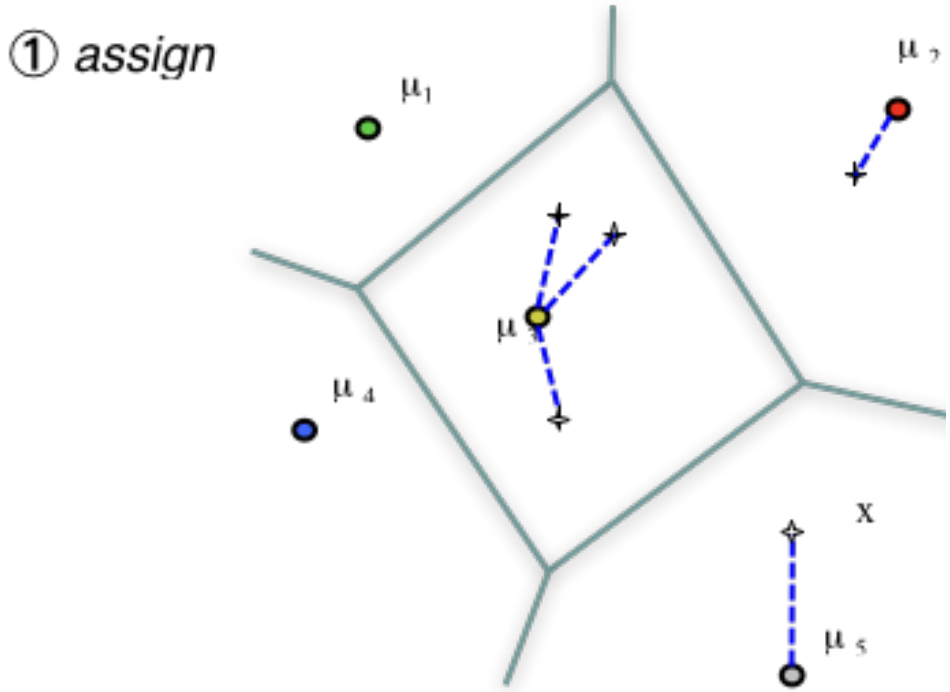
A better approach was to consider the variance of the sift features after considering the mean. This allowed to aggregate the quantization error in each dimension and explicitly represent it

It consisted of the following steps. Initially the centroids $\{\mu_i, i = 1, ..., N\}$ is estimated from the set of sift feature vectors $X = \{x_j, j = 1, ...T\}$.

VLAD

A better approach was to consider the variance of the sift features after considering the mean. This allowed to aggregate the quantization error in each dimension and explicitly represent it

It consisted of the following steps. Initially the centroids $\{\mu_i, i = 1, ..., N\}$ is estimated from the set of sift feature vectors $X = \{x_j, j = 1, ...T\}$.



VLAD

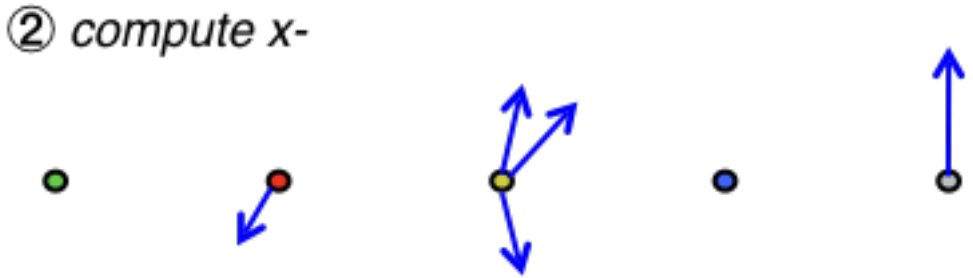
Next for each feature vector we associate the nearest neighbor centroid obtained by $NN(x_t) = argmin_{\mu_i} ||x_t - \mu_i||$

VLAD

Next for each feature vector we associate the nearest neighbor centroid obtained by $NN(x_t) = argmin_{\mu_i} ||x_t - \mu_i||$
After associating it with the centroid the difference $x_t - \mu_i$ is calculated

VLAD

Next for each feature vector we associate the nearest neighbor centroid obtained by $NN(x_t) = argmin_{\mu_i} ||x_t - \mu_i||$
After associating it with the centroid the difference $x_t - \mu_i$ is calculated

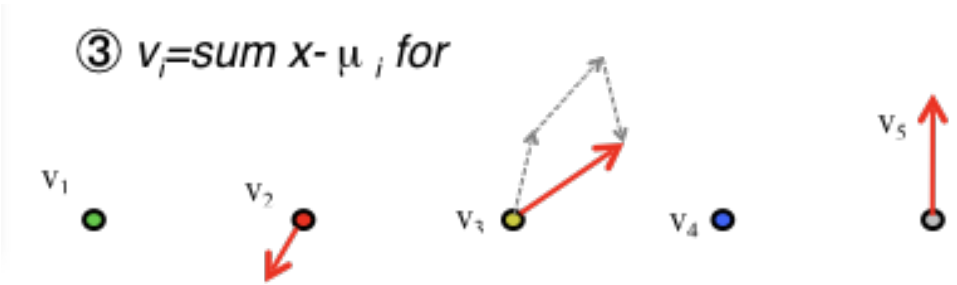


VLAD

Next for centroid the difference is aggregated around each dimension of the SIFT vector from the set of centroids mapped to the centroid. each feature vector we associate the nearest neighbor centroid obtained by $v_i = \sum_{x_t : NN(x_t) = \mu_i} x_t - \mu_i$

VLAD

Next for centroid the difference is aggregated around each dimension of the SIFT vector from the set of centroids mapped to the centroid. each feature vector we associate the nearest neighbor centroid obtained by $v_i = \sum_{x_t : NN(x_t) = \mu_i} x_t - \mu_i$



VLAD

These are the steps for VLAD summarized as follows

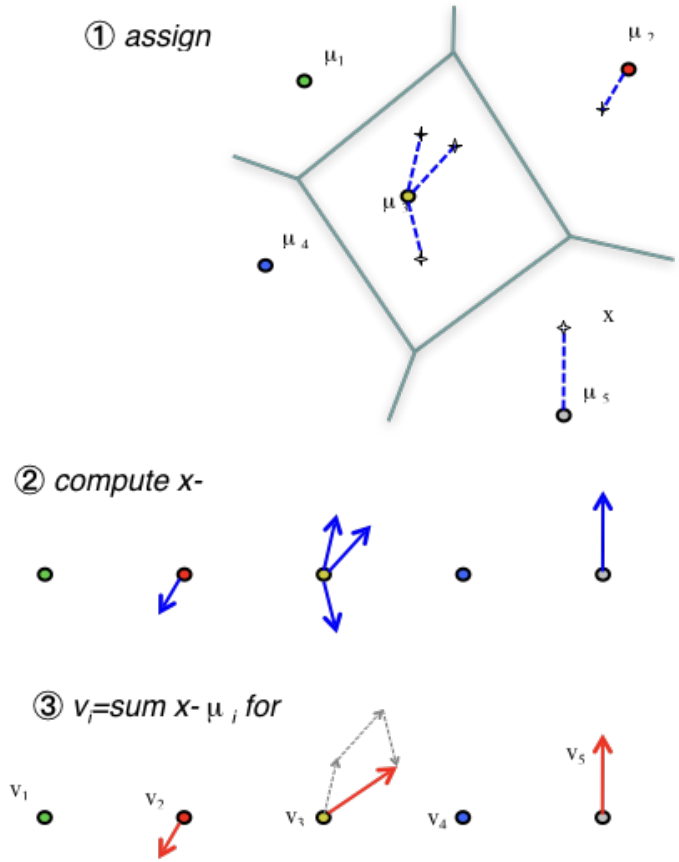
$$\{\mu_i, i = 1 \dots N\}$$

$$X = \{x_t, t = 1 \dots T\}$$

$$\text{NN}(x_t) = \arg\min_{\mu_i} ||x_t - \mu_i||$$

$$v_i = \sum_{x_t : \text{NN}(x_t) = \mu_i} x_t - \mu_i$$

$$\ell_2$$



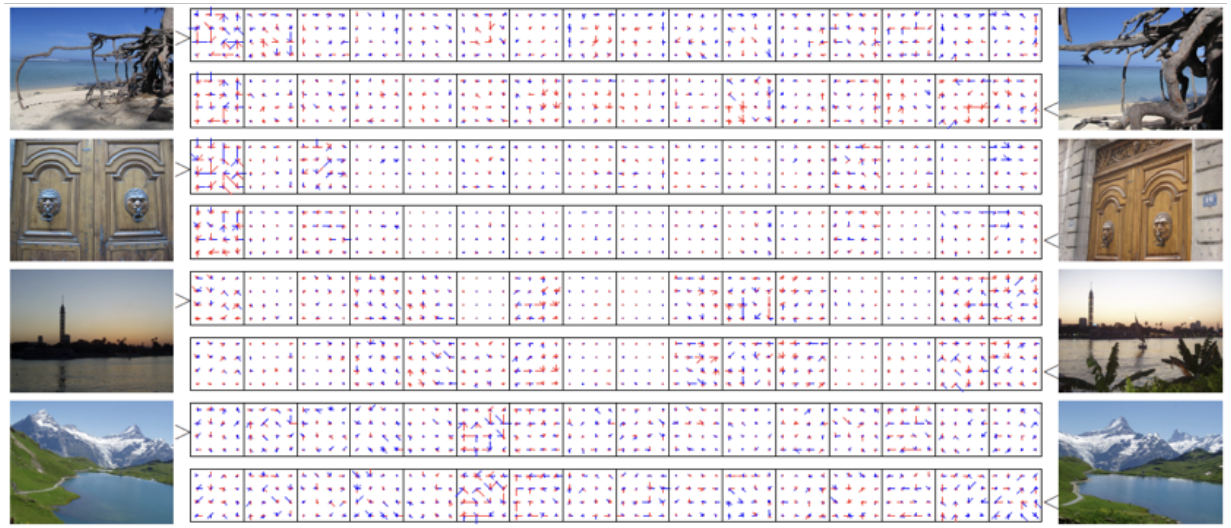
VLAD

This can be understood by visualizing the variance for the case where there are 16 centroids and we consider the variance for these 16 centroids. This bag of words representation will have a length of $16+ 16 \times 128$ as its length for the case of 16 centroids

VLAD

This can be understood by visualizing the variance for the case where there are 16 centroids and we consider the variance for these 16 centroids. This bag of words representation will have a length of 16+ 16 $\times$ 128 as its length for the case of 16 centroids

$$v_i = \sum_{x_t : \mathrm{NN}(x_t) = \mu_i} x_t - \mu_i$$

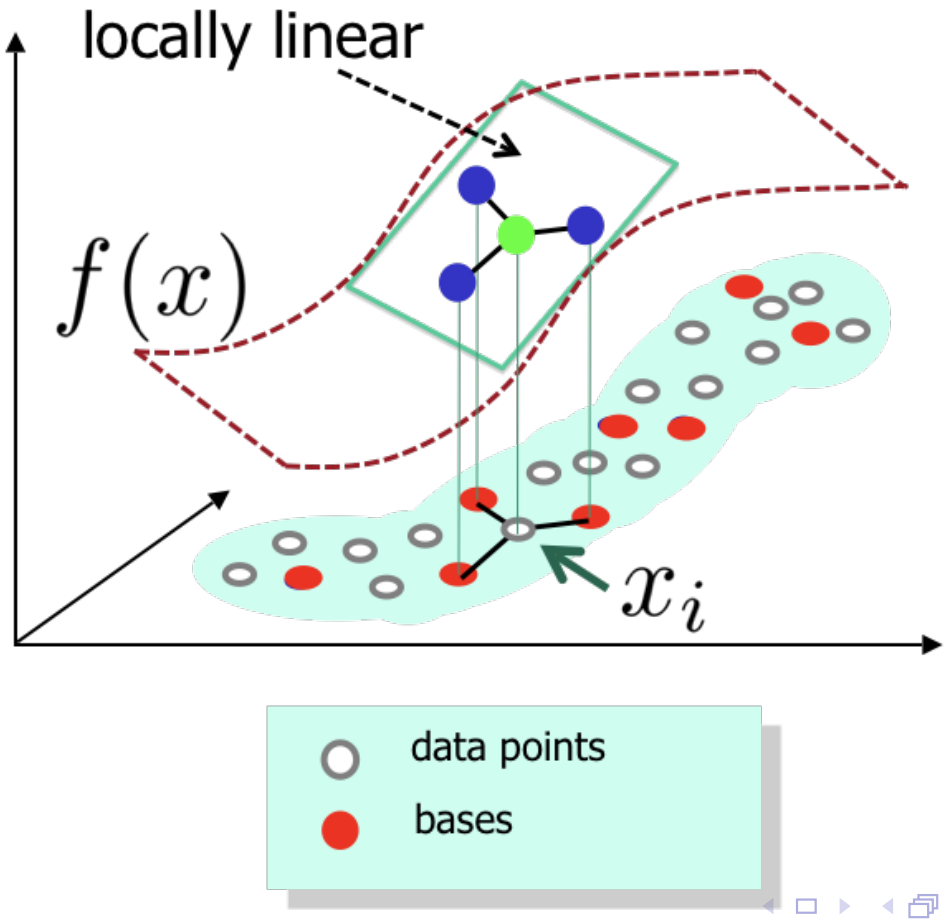


Locally linear coordinate representation

VLAD feature representation is expensive in terms of size of the feature vector. A better approach was to reduce the quantization error using locally linear coordinate representation

Locally linear coordinate representation

VLAD feature representation is expensive in terms of size of the feature vector. A better approach was to reduce the quantization error using locally linear coordinate representation



Locally linear coordinate formulation

The formulation for loally linear coordinate coding is given as follows

Locally linear coordinate formulation

The formulation for loally linear coordinate coding is given as follows

- Coding for x, to obtain its sparse representation a

  Step 1 – ensure locality: find the K nearest bases

$$[\phi_j]_{j \in J(x)}$$

  Step 2 – ensure low coding error:

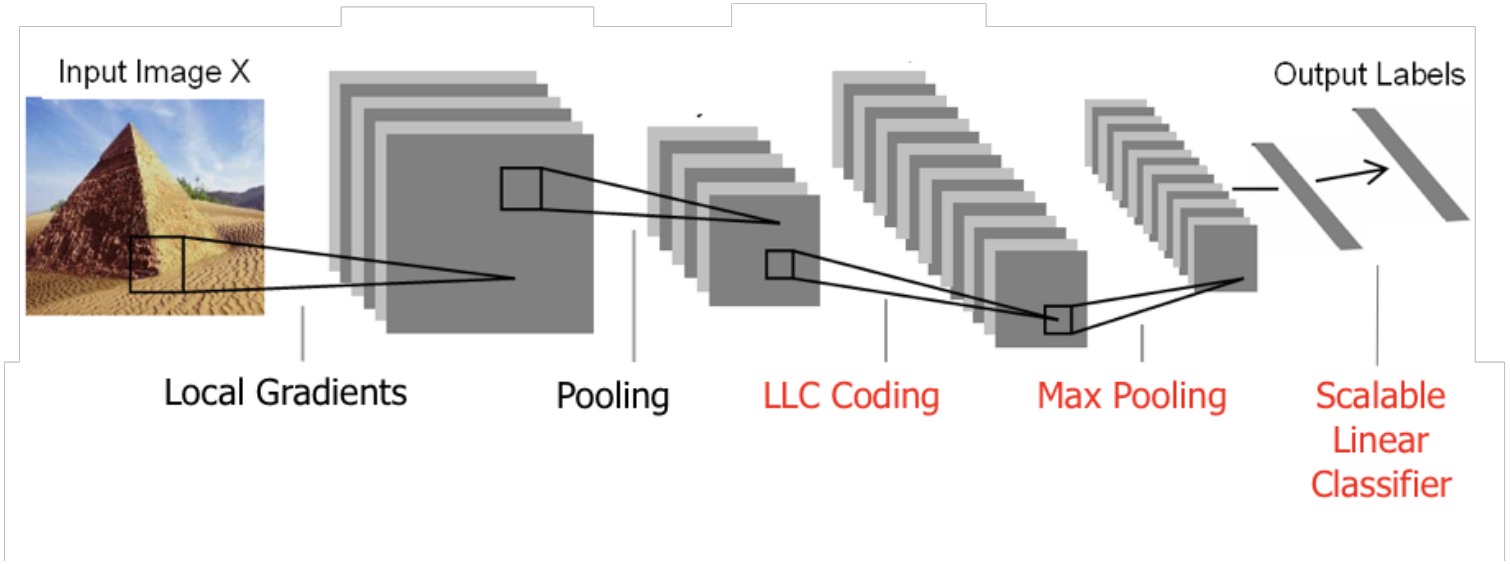
$$\min_a \left\| x - \sum_{j \in J(x)} a_{i,j} \phi_j \right\|^2, \quad \text{s.t.} \sum_{j \in J(x)} a_{i,j} = 1$$

Locally linear coordinate Overview

The pipeline for the whole framework is presented as follows. This appears to be very similar to the conventional deep learning framework that consists of convolution and pooling operations

Locally linear coordinate Overview

The pipeline for the whole framework is presented as follows. This appears to be very similar to the conventional deep learning framework that consists of convolution and pooling operations



The End