Jan 12th 2016

Applications of PCA

- Data reduction
- data visualization
- data classification
- Trend analysis
- factor analysis
- noise removal.

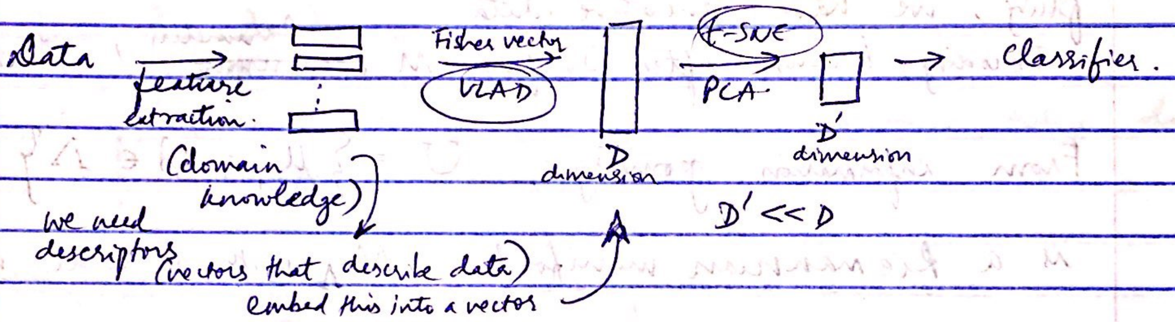
Related:

SVD : Singular Value Dec.

LDA : Linear Discriminant Analysis

(look up Kernalized PCA)

one problem: PCA is a linear approach. ∴ if you have nonlinear, then cannot be used.



we need descriptors (vectors that describe data)

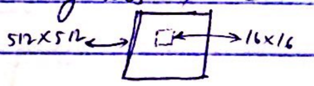
embed this into a vectors

$D' << D$

VLAD.

[Vector of Locally Aggregated Descriptors]

cuz it's visual data = features

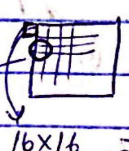


→ How to recognize objects/scenes?

(extracts features automatically: deep solution)

Visual Recognition.

Given an image I

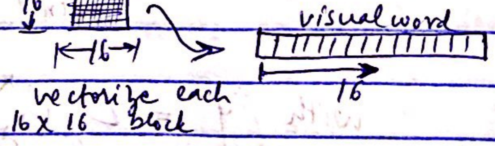


divide it into many n x n windows/cells.

16x16 (sub image)

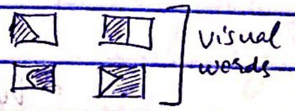
16x16 → nice size. details that matter for recognition

(David Lowe used this)

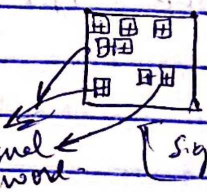


vectorize each 16 x 16 block

Visual Bag of Words



Another approach: (Keypoints)



Find keypoints (corners)

[Significant salient locations in image]

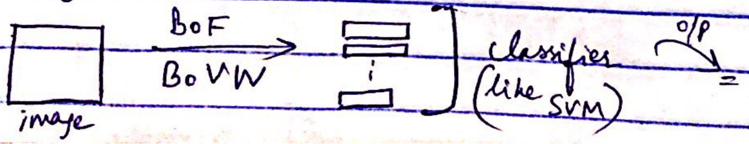
BoF / BoVW

(Bag of features) (Bag of visual words)

SIFT

SURF

~~Theory~~



= "it's a cat!"

practice
- too many vectors (curse of dimensionality)
- redundancy
- noise

General approach to visual recognition:

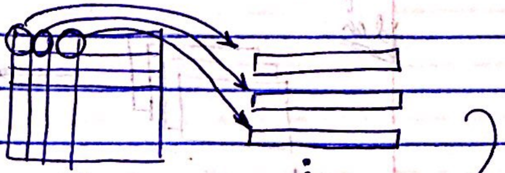
Build a codebook $C = \{c_1, c_2 \ldots c_n\}$ from $m \gg n$ feature vectors (visual words)

"codebook" = (dictionary of visual words)

m → # of features we have

n → size of dictionary

idea: use a clustering algorithm!

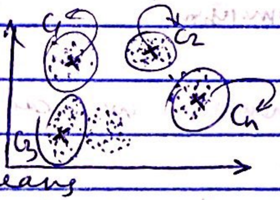


Codebook =

$C = \{c_1, c_2, \ldots c_n\}$ ~~clustered data~~ ← run clustering ← find classes.

↓ (centre of classes)

(codebook is a vector)

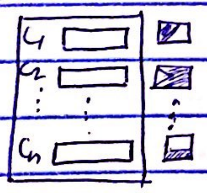


map to feature space to cluster

# of classes : # of size of dictionary

Idea of histogram:

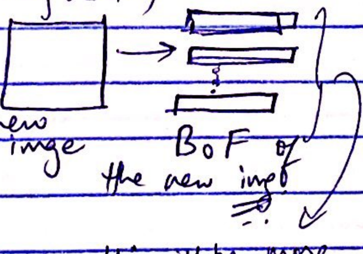
codebook.



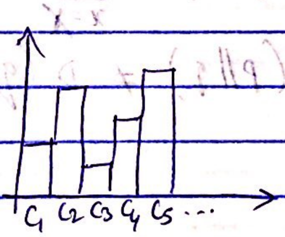
result of training, we have a codebook. These are prototypes of visual words.

~~Based on data~~

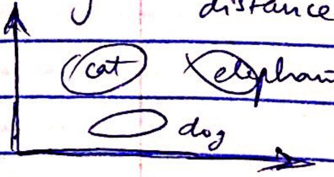
new image comes in. (not a part of training data)



this will be more # of words obviously than the # of words in codebook.



compare histograms using Euclidean distance



VLAD idea.

Accumulate for each visual word c_i, the difference $x - c_i$ of the vectors x assigned to c_i

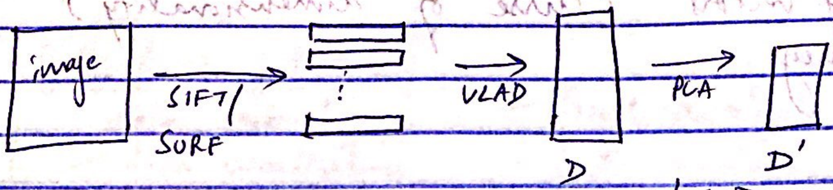
$$V_{i,j} = \sum_{NN(x)=c_i} x_j - c_{i,j}$$

→ embedding

normalization

L2 normalization

$$V := \frac{V}{\|V\|_2}$$



$D'<<D$.

VLAD & fisher vector: encoding & embedding techniques

PCA ; t-SNE : compression techniques.

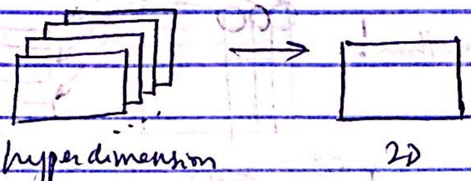
t-SNE

t-distributed Stochastic Embedding.

PCA : linear   t-SNE : non linear

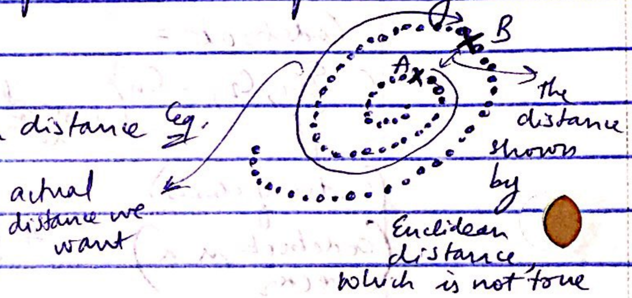
if you have hyperdimensional data & you want to ~~have~~ have a topology preserving... (Only for visualization)

idea:



Similar objects in hyper dimension correspond to close proximity in 2D.

in hyperdimension, we cannot use Euclidean distance Eg.



* Kullbuck-Leibler Divergence

$$D(p||q) = \sum_{x=X} p(x) \log \frac{p(x)}{q(x)}$$

$$D(p||q) \neq D(q \neq p)$$

divergence in hyperspace & 2D space should be same.