Deep Learning for Computer Vision

From Points to Images: Bag-of-Words and VLAD Representations

Vineeth N Balasubramanian

Department of Computer Science and Engineering
Indian Institute of Technology, Hyderabad

Acknowledgements

- Most of this lecture's slides are based on lectures of **Deep Learning for Vision** course taught by Prof Yannis Avrithis at Inria Rennes-Bretagne Atlantique

Review

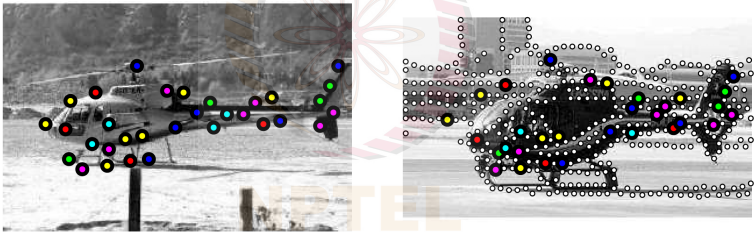
So far:

- Descriptors for matching features between different views of the same scene/object
  - Used in image stitching, image retrieval, etc.

Source: Yannis Avrithis, Deep Learning for Vision, Spring 2019 SIF

Review

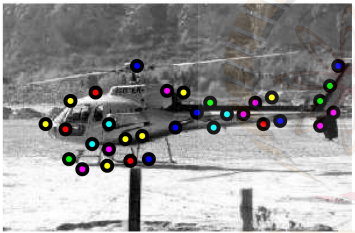
- Can the same descriptors be used for matching different instances of the object?
  - Used in image classification, detection, etc.

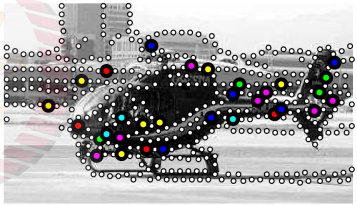


Source: Yannis Avrithis, Deep Learning for Vision, Spring 2019 SIF

Review

- Can the same descriptors be used for matching different instances of the object?
  - Used in image classification, detection, etc.



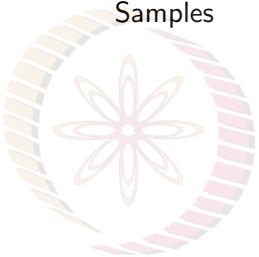


Rigid transformations may not work here. Can we discard geometry for now, and bring it back in other ways?

Source: Yannis Avrithis, Deep Learning for Vision, Spring 2019 SIF

Our First Attempt: Bag-of-Words (BoW)

Samples



Our First Attempt: Bag-of-Words (BoW)

Samples

Form Vocabulary

Our First Attempt: Bag-of-Words (BoW)

Our First Attempt: Bag-of-Words (BoW)

1) John likes to watch movies. Mary likes movies too.

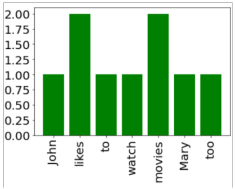
2) Mary also likes to watch football games..

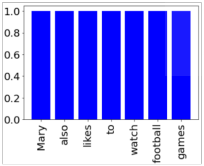
Samples

"John","likes","to","watch","movies","Mary","likes","movies","too"

"Mary","also","likes","to","watch","football","games"

Form Vocabulary





Histogram

Our First Attempt: Bag-of-Words (BoW)

1) John likes to watch movies. Mary likes movies too.

2) Mary also likes to watch football games..

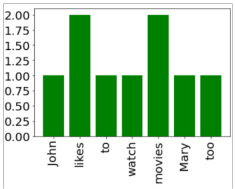
Samples

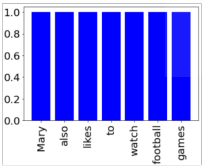
Form Vocabulary

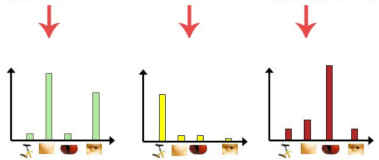
"John","likes","to","watch","movies","Mary","likes","movies","too"

"Mary","also","likes","to","watch","football","games"

Histogram







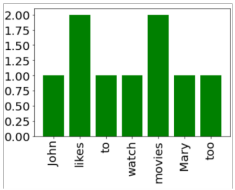
Our First Attempt: Bag-of-Words (BoW)

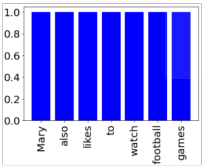
1) John likes to watch movies. Mary likes movies too.

2) Mary also likes to watch football games..

Samples

Uses of Bag-of-Words?





Histogram

Our First Attempt: Bag-of-Words (BoW)

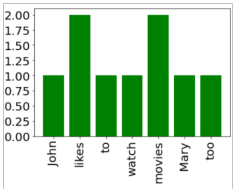
1) John likes to watch movies. Mary likes movies too.

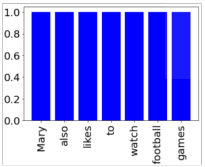
2) Mary also likes to watch football games..

Samples

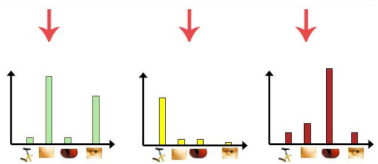
Uses of Bag-of-Words?

- Retrieval
- Classification





Histogram



BoW for Retrieval[1]

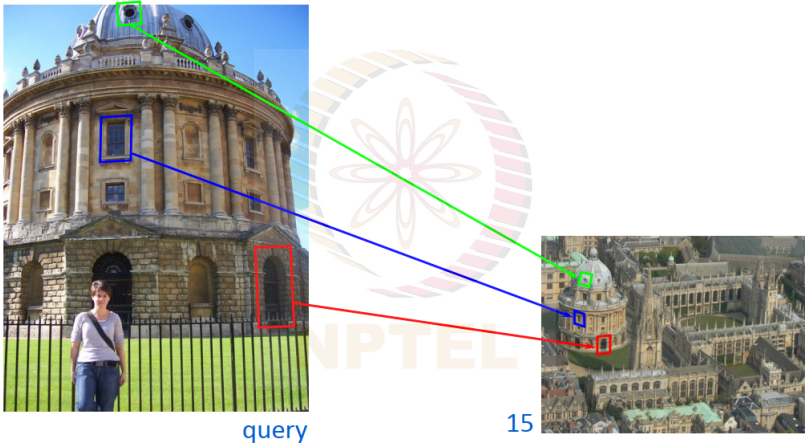
query

15

Query vs dataset image

[1] Sivic and Zisserman, Video Google: A Text Retrieval Approach to Object Matching in videos, ICCV 2003

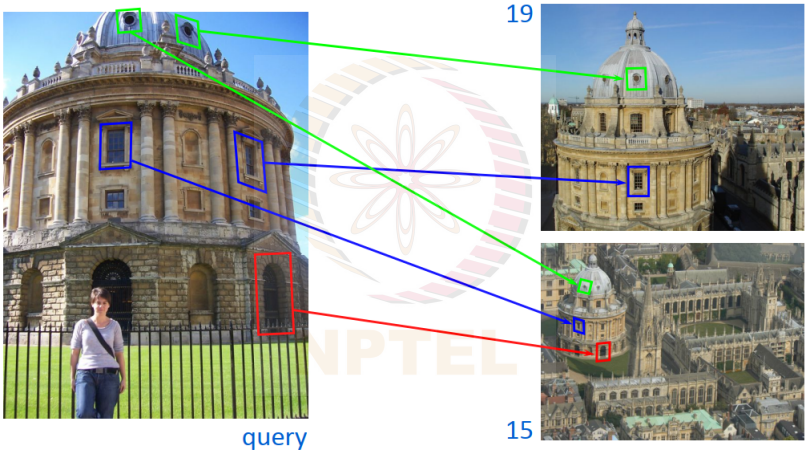
BoW for Retrieval[1]



Pairwise descriptor matching

[1] Sivic and Zisserman, Video Google: A Text Retrieval Approach to Object Matching in videos, ICCV 2003

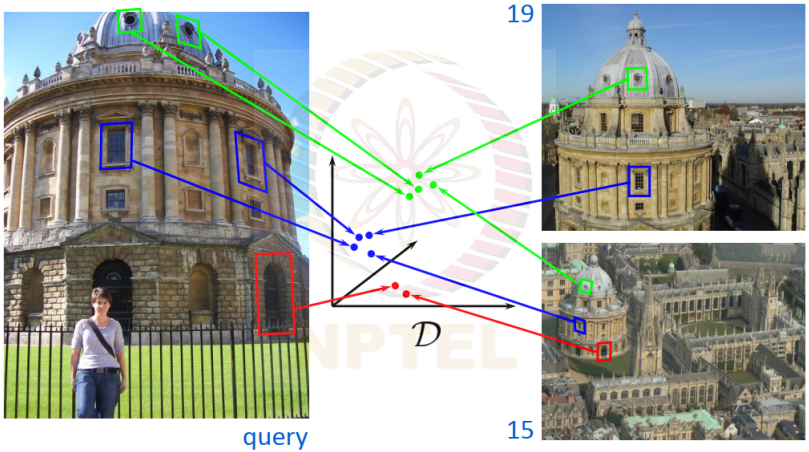
BoW for Retrieval[1]



Pairwise descriptor matching for every dataset image

[1] Sivic and Zisserman, Video Google: A Text Retrieval Approach to Object Matching in videos, ICCV 2003

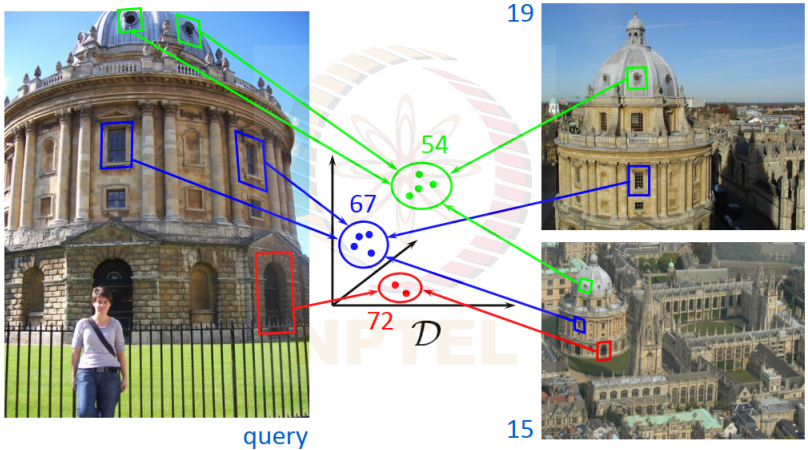
BoW for Retrieval[1]



Similar descriptors should all be nearby in the descriptor space

[1] Sivic and Zisserman, Video Google: A Text Retrieval Approach to Object Matching in videos, ICCV 2003

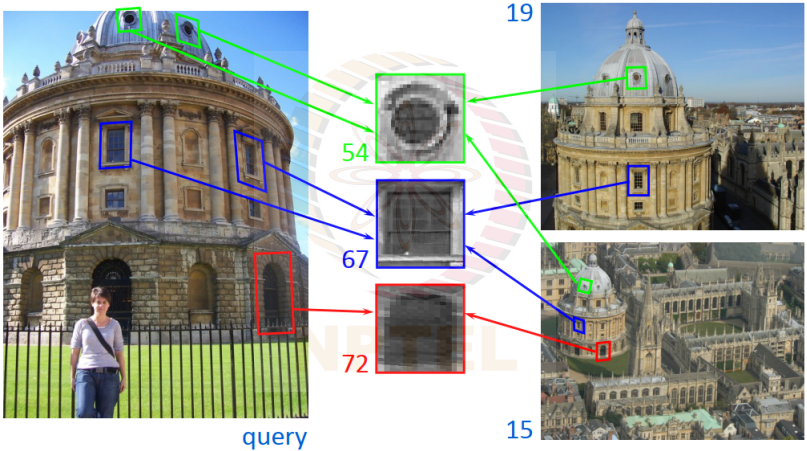
BoW for Retrieval[1]



Quantize them into visual words

[1] Sivic and Zisserman, Video Google: A Text Retrieval Approach to Object Matching in videos, ICCV 2003

BoW for Retrieval[1]



Now visual words act as a proxy. No pairwise matching needed

[1] Sivic and Zisserman, Video Google: A Text Retrieval Approach to Object Matching in videos, ICCV 2003

BoW for Retrieval

- Each image is represented by a vector $\mathbf{z} \in \mathbb{R}^k$, where k is size of codebook
  - Each element $z_i = w_i n_i$ where w_i is a fixed weight per visual word and n_i number of occurrences of this word in the image

BoW for Retrieval

- Each image is represented by a vector $\mathbf{z} \in \mathbb{R}^k$, where k is size of codebook
  - Each element $z_i = w_i n_i$ where w_i is a fixed weight per visual word and n_i number of occurrences of this word in the image
- Given a set of n images represented by matrix $Z \in \mathbb{R}^{k \times n}$ (each image as a column) and query image $\mathbf{q}$, we need a vector of similarities:

$$s = S_{\text{BoW}}(Z, \mathbf{q}) := Z^\top \mathbf{q}$$

and then sort s by descending order

- Note: With L_2-normalization, this is equivalent to measuring Euclidean distance: for vectors $\mathbf{z}$ and $\mathbf{q}$, $||\mathbf{z} - \mathbf{q}||^2 = 2(1 - \mathbf{z}^T\mathbf{q})$

BoW for Retrieval

- Each image is represented by a vector $\mathbf{z} \in \mathbb{R}^k$, where k is size of codebook
  - Each element $z_i = w_i n_i$ where w_i is a fixed weight per visual word and n_i number of occurrences of this word in the image
- Given a set of n images represented by matrix $Z \in \mathbb{R}^{k \times n}$ (each image as a column) and query image $\mathbf{q}$, we need a vector of similarities:

$$s = S_{\text{BoW}}(Z, \mathbf{q}) := Z^\top \mathbf{q}$$

  and then sort s by descending order
  - Note: With L_2-normalization, this is equivalent to measuring Euclidean distance: for vectors $\mathbf{z}$ and $\mathbf{q}$, $||\mathbf{z} - \mathbf{q}||^2 = 2(1 - \mathbf{z}^T\mathbf{q})$
- When $k \gg p$, where p is the number of features per image on average, Z and $\mathbf{q}$ are sparse

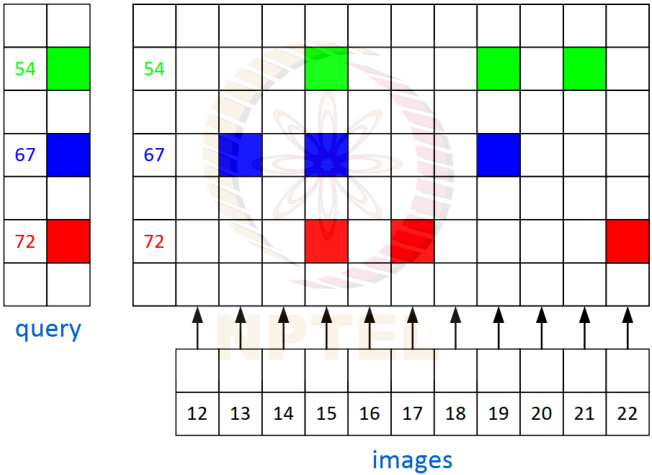
BoW for Retrieval

- Each image is represented by a vector $\mathbf{z} \in \mathbb{R}^k$, where k is size of codebook
  - Each element $z_i = w_i n_i$ where w_i is a fixed weight per visual word and n_i number of occurrences of this word in the image
- Given a set of n images represented by matrix $Z \in \mathbb{R}^{k \times n}$ (each image as a column) and query image $\mathbf{q}$, we need a vector of similarities:

$$s = S_{\text{BoW}}(Z, \mathbf{q}) := Z^\top \mathbf{q}$$

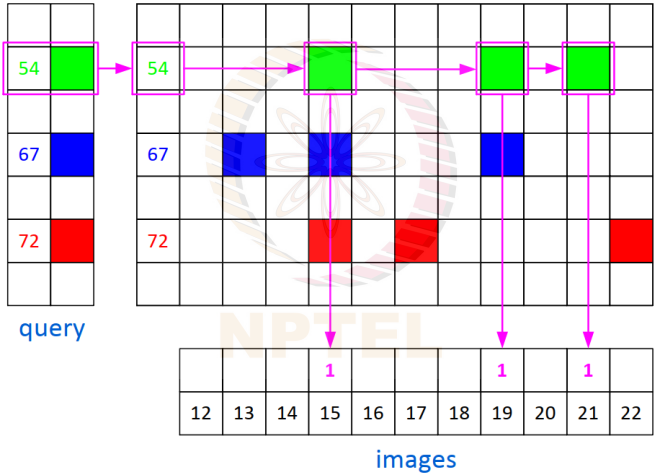
  and then sort s by descending order
  - Note: With L_2-normalization, this is equivalent to measuring Euclidean distance: for vectors $\mathbf{z}$ and $\mathbf{q}$, $||\mathbf{z} - \mathbf{q}||^2 = 2(1 - \mathbf{z}^T\mathbf{q})$
- When $k \gg p$, where p is the number of features per image on average, Z and $\mathbf{q}$ are sparse
- Rather than check whether a word is contained in an image, check which images contain a given word

BoW for Retrieval: Inverted File Index[2]



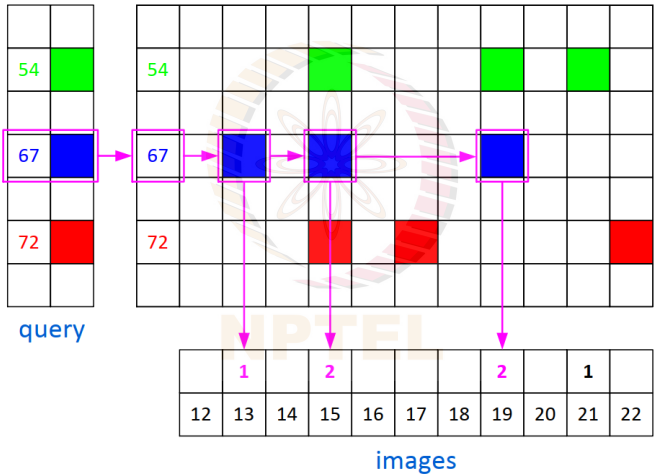
[2]Sivic and Zisserman, Video Google: A Text Retrieval Approach to Object Matching in videos, ICCV 2003

BoW for Retrieval: Inverted File Index[2]



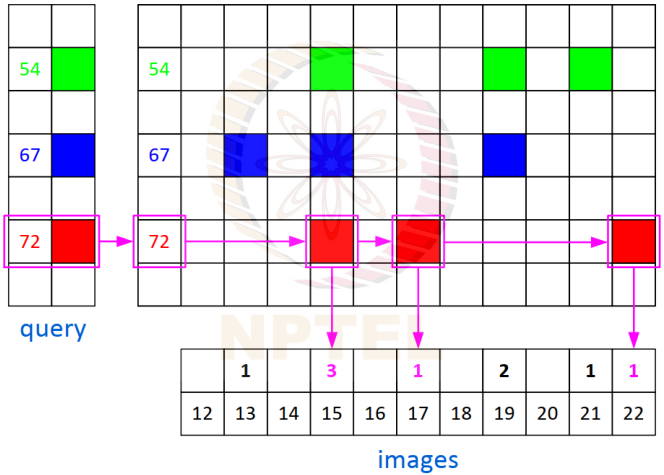
[2]Sivic and Zisserman, Video Google: A Text Retrieval Approach to Object Matching in videos, ICCV 2003

BoW for Retrieval: Inverted File Index[2]



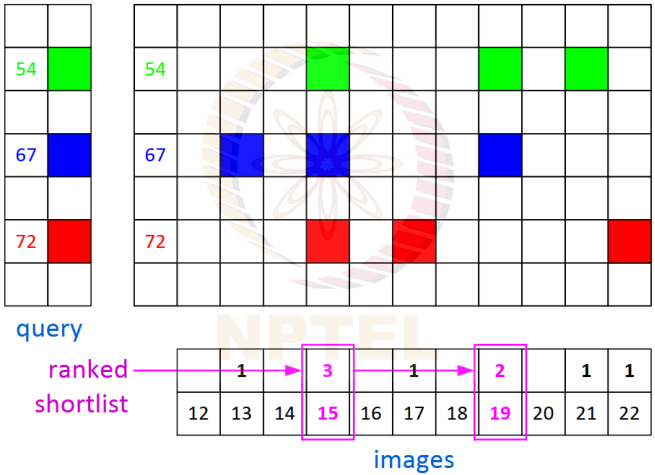
[2]Sivic and Zisserman, Video Google: A Text Retrieval Approach to Object Matching in videos, ICCV 2003

BoW for Retrieval: Inverted File Index[2]



[2]Sivic and Zisserman, Video Google: A Text Retrieval Approach to Object Matching in videos, ICCV 2003

BoW for Retrieval: Inverted File Index[2]



[2]Sivic and Zisserman, Video Google: A Text Retrieval Approach to Object Matching in videos, ICCV 2003

BoW for Classification

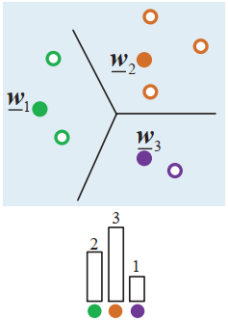
BoW for Classification

- Each image represented by $\mathbf{z} \in \mathbb{R}^k$; each element z_i the number of occurrences of visual word c_i in the image.

BoW for Classification

- Each image represented by $\mathbf{z} \in \mathbb{R}^k$; each element z_i the number of occurrences of visual word c_i in the image.
- **Naive Bayes:** Chose maximum posterior probability of class $\mathbf{C}$ given image $\mathbf{z}$ assuming features are independent $\rightarrow$ linear classifier with parameters estimated by visual word statistics on training set
- **Support Vector Machine (SVM):** Images $\mathbf{z}_1, \mathbf{z}_2$ compared using a kernel function $\phi(\cdot)$; if $\phi(\mathbf{z}_1)^T \phi(\mathbf{z}_2) = \mathbf{z}_1^T \mathbf{z}_2$, this is again a linear classifier

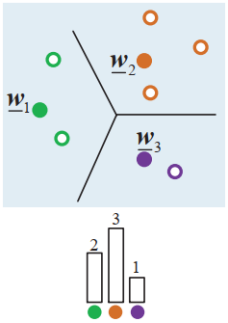
Extension of BoW: Vector of Locally Aggregated Descriptors (VLAD)



(BoW)

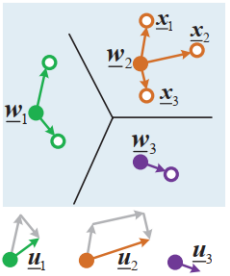
- Yields a scalar frequency
- Limited information

Extension of BoW: Vector of Locally Aggregated Descriptors (VLAD)



(BoW)

- Yields a scalar frequency
- Limited information

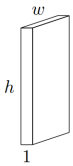


(VLAD)

- Yields a vector per visual word
- Comparatively more information, resulting in better discrimination by classifier

Credit: Li Liu

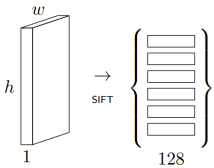
BoW Vs VLAD



- 3-channel RGB input $\rightarrow$ 1-channel gray-scale

Credit: Yannis A

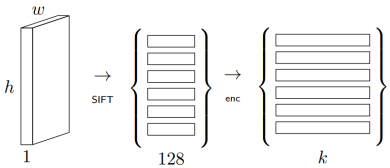
BoW Vs VLAD



- 3-channel RGB input $\rightarrow$ 1-channel gray-scale
- Set of $\sim$ 1000 features $\times$ 128-dim SIFT descriptors

Credit: Yannis A

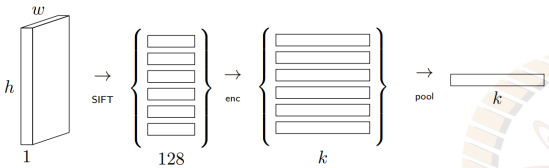
BoW Vs VLAD



- 3-channel RGB input $\rightarrow$ 1-channel gray-scale
- Set of $\sim$ 1000 features $\times$ 128-dim SIFT descriptors
- Element-wise encoding (hard assignment) on $k \sim 100$ visual words

Credit: Yannis A

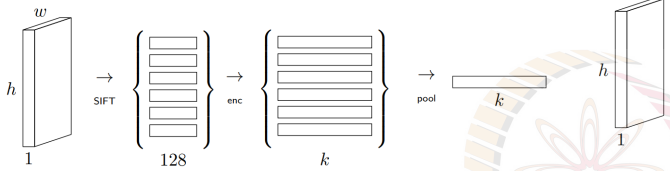
BoW Vs VLAD



- 3-channel RGB input $\rightarrow$ 1-channel gray-scale
- Set of $\sim$ 1000 features $\times$ 128-dim SIFT descriptors
- Element-wise encoding (hard assignment) on $k \sim 100$ visual words
- Global sum pooling, L_2 normalization.

Credit: Yannis A

BoW Vs VLAD

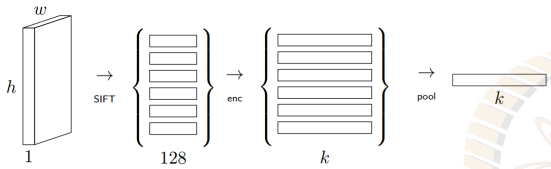


- 3-channel RGB input $\rightarrow$ 1-channel gray-scale
- Set of $\sim$ 1000 features $\times$ 128-dim SIFT descriptors
- Element-wise encoding (hard assignment) on $k \sim 100$ visual words
- Global sum pooling, L_2 normalization.

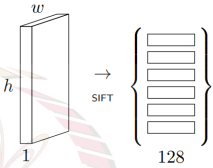
- 3-channel RGB input $\rightarrow$ 1-channel gray-scale

Credit: Yannis A

BoW Vs VLAD



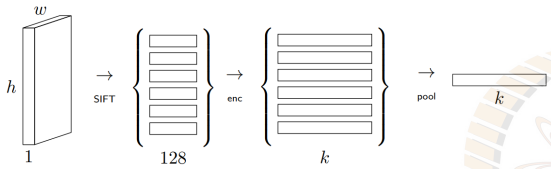
- 3-channel RGB input $\rightarrow$ 1-channel gray-scale
- Set of $\sim$ 1000 features $\times$ 128-dim SIFT descriptors
- Element-wise encoding (hard assignment) on $k \sim 100$ visual words
- Global sum pooling, L_2 normalization.



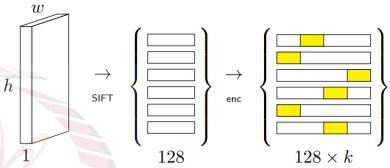
- 3-channel RGB input $\rightarrow$ 1-channel gray-scale
- Set of $\sim$ 1000 features $\times$ 128-dim SIFT descriptors

Credit: Yannis A

BoW Vs VLAD



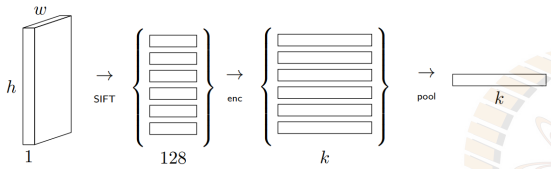
- 3-channel RGB input $\rightarrow$ 1-channel gray-scale
- Set of $\sim$ 1000 features $\times$ 128-dim SIFT descriptors
- Element-wise encoding (hard assignment) on $k \sim 100$ visual words
- Global sum pooling, L_2 normalization.



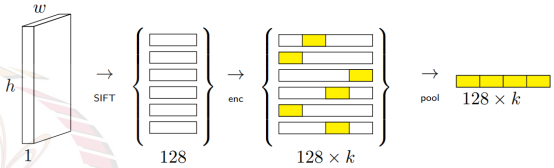
- 3-channel RGB input $\rightarrow$ 1-channel gray-scale
- Set of $\sim$ 1000 features $\times$ 128-dim SIFT descriptors
- Element-wise encoding (hard assignment) on $k \sim 100$ visual words. Yields a residual vector rather than a scalar vote

Credit: Yannis A

BoW Vs VLAD



- 3-channel RGB input $\rightarrow$ 1-channel gray-scale
- Set of $\sim$ 1000 features $\times$ 128-dim SIFT descriptors
- Element-wise encoding (hard assignment) on $k \sim 100$ visual words
- Global sum pooling, L_2 normalization.



- 3-channel RGB input $\rightarrow$ 1-channel gray-scale
- Set of $\sim$ 1000 features $\times$ 128-dim SIFT descriptors
- Element-wise encoding (hard assignment) on $k \sim 100$ visual words. Yields a residual vector rather than a scalar vote
- Global sum pooling, L_2 normalization.

Credit: Yannis A

Homework

Readings

- Chapter 14.3, 14.4, Szeliski, *Computer Vision: Algorithms and Applications*

Questions

- What is the connection of BoW to the k-means clustering algorithm?
- How would you now use k-means variants such as hierarchical k-means and approximate k-means to extend BoW? (*Hint: Look up vocabulary trees!*)

References


J Sivic and A Zisserman. "Video Google: A text retrieval approach to object matching in videos". In: ICCV. 2003.


Gabriella Csurka et al. "Visual categorization with bags of keypoints". In: *Workshop on statistical learning in computer vision, ECCV*. Vol. 1. 1-22. 2004, pp. 1–2.


Richard Szeliski. *Computer Vision: Algorithms and Applications*. Texts in Computer Science. London: Springer-Verlag, 2011.