Image Description: Bag of Visual Words

Alexandre Xavier Falcão

Institute of Computing - UNICAMP

afalcao@ic.unicamp.br

Bag of Words (BoW)

Let A, B, and C be examples of texts about different subjects.

A

I love pets. At home, I have one dog and two cats. My cats like dog food. I have to hide the dog's food from the cats every time I feed the dog.

B

Here, it is the business card of John. He is the veterinary who owns a pet shop in downtown and takes care of my dogs and cats. The dogs and cats love him.

C

Here, it is my business card if you are still interested in buying the coffee shop I own in downtown.

A Bag of Words (BoW) is a dictionary of keywords identified as the most frequent ones in texts from different subjects.

Bag of Words (BoW)

A

I love pets. At home, I have one dog and two cats. My cats like dog food. I have to hide the dog's food from the cats every time I feed the dog.

B

Here, it is the business card of John. He is the veterinary who owns a pet shop in downtown and takes care of my dogs and cats. The dogs and cats love him.

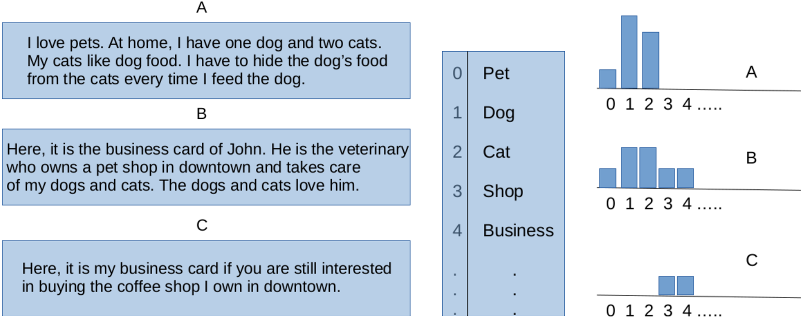
C

Here, it is my business card if you are still interested in buying the coffee shop I own in downtown.

0	Pet
1	Dog
2	Cat
3	Shop
4	Business
.	.
.	.
.	.

By using the dictionary on the right to determine the frequency of its words in A, B, and C, each text is represented by the following histograms.

Bag of Words (BoW)



We expect higher similarity between the histograms of A and B then between any of them and the histogram of C.

Agenda

- This lecture extends the concept of BoW to Bag of Visual Words (BoVW) – a methodology to describe images from local texture (color) features.

Agenda

- This lecture extends the concept of BoW to Bag of Visual Words (BoVW) – a methodology to describe images from local texture (color) features.
- BoVW consists of three main steps.

Agenda

- This lecture extends the concept of BoW to Bag of Visual Words (BoVW) – a methodology to describe images from local texture (color) features.
- BoVW consists of three main steps.
  - Local feature extraction.

Agenda

- This lecture extends the concept of BoW to Bag of Visual Words (BoVW) – a methodology to describe images from local texture (color) features.
- BoVW consists of three main steps.
  - Local feature extraction.
  - Dictionary construction (codebook generation).

Agenda

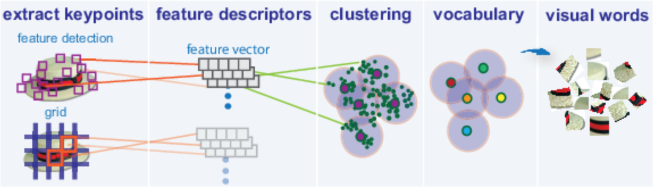
- This lecture extends the concept of BoW to Bag of Visual Words (BoVW) – a methodology to describe images from local texture (color) features.
- BoVW consists of three main steps.
  - Local feature extraction.
  - Dictionary construction (codebook generation).
  - Coding (vector quantization, pooling).

Agenda

- This lecture extends the concept of BoW to Bag of Visual Words (BoVW) – a methodology to describe images from local texture (color) features.
- BoVW consists of three main steps.
  - Local feature extraction.
  - Dictionary construction (codebook generation).
  - Coding (vector quantization, pooling).
- The result is one feature vector per image.

Bag of Visual Words (BoVW)

The main steps in BoVW can be represented as follows.

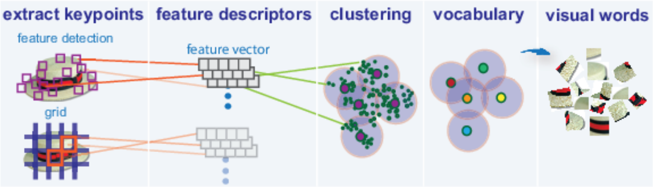


from MathWorks.com

- Feature extraction involves key-point detection and subimage (patch) description at each key point from all training images.

Bag of Visual Words (BoVW)

The main steps in BoVW can be represented as follows.



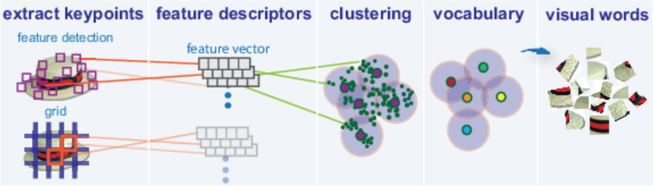
from MathWorks.com

- Feature extraction involves key-point detection and subimage (patch) description at each key point from all training images.

- Codebook generation requires patch clustering identifying one visual word per group.

Bag of Visual Words (BoVW)

The main steps in BoVW can be represented as follows.



from MathWorks.com

- Feature extraction involves key-point detection and subimage (patch) description at each key point from all training images.

- Codebook generation requires patch clustering identifying one visual word per group.

- Coding explores similarities between patch descriptors and visual words.

Feature extraction

- Some visual words might be common for different categories, but be discriminative when used together with other words.

Feature extraction

- Some visual words might be common for different categories, but be discriminative when used together with other words.
- The discriminative power of such words must be determined by some feature selection technique.

Feature extraction

- Some visual words might be common for different categories, but be discriminative when used together with other words.
- The discriminative power of such words must be determined by some feature selection technique.
- Key points may be estimated by sampling techniques such as SIFT, SURF, GRID, Random, or as superpixel centers [1, 2].

Feature extraction

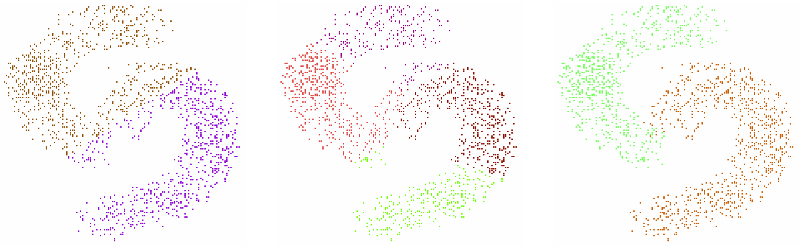
- Some visual words might be common for different categories, but be discriminative when used together with other words.
- The discriminative power of such words must be determined by some feature selection technique.
- Key points may be estimated by sampling techniques such as SIFT, SURF, GRID, Random, or as superpixel centers [1, 2].
- BIC, LBP, HoG and other descriptors can then be extracted from patches at each key point.

Codebook generation

- Visual words are usually defined by the k-means clustering algorithm, where k defines the size of the dictionary.

- Visual words are usually defined by the k-means clustering algorithm, where k defines the size of the dictionary.

- The k-means algorithm assumes that the clusters are hyper-spheres.



$k = 2$ groups $k = 4$ groups 2 categories

Codebook generation by k-means

Let X be the $n \times m$ feature matrix of m samples from a dataset $\mathcal{Z} = \{s_j\}_{j=1}^{m}$ – i.e., each row in X is a feature vector $\mathbf{x}(s_j)$. The k-means algorithm finds k groups $\{G_i\}_{i=1}^{k}$ (clusters) by assigning each sample $s \in \mathcal{Z}$, $m \gg k$, to one group, such that

$$\sum_{i=1}^{k} \sum_{s \in G_i} \|\mathbf{x}(s) - \mu_i\|^2$$

is minimized and

$$\mu_i = \frac{1}{|G_i|} \sum_{s \in G_i} \mathbf{x}(s)$$

is the centroid of group G_i. The algorithm works as follows.

Codebook generation by k-means

Input : A dataset $(\mathcal{Z}, X)$.
Output: A label map $L\colon \mathcal{Z} \rightarrow \{i\}_{i=1}^{k}$ (i.e., $L(s) = i \Rightarrow s \in G_i$).

1. Select k random centroids $\{\mu_i\}_{i=1}^{k}$ from $\{\mathbf{x}(s_j)\}_{j=1}^{m}$.
2. For each iteration $t = 1, 2, \ldots, T$ do.
3. For each sample $s \in \{s_j\}_{j=1}^{m}$ do.
4. Set $L(s) \leftarrow \arg\min_{i=1,2,\ldots,k}\{\|\mathbf{x}(s) - \mu_i\|^2\}$.
5. For each group G_i, $i = 1, 2, \ldots, k$, do.
6. Update $\mu_i \leftarrow \frac{1}{|G_i|}\sum_{s\in\mathcal{Z}|L(s)=i}\mathbf{x}(s)$.

- The algorithm may be interrupted, when the differences between previous and current centroids are negligible.

- The representative $\mathbf{x}(s)$ of group G_i can also be selected as the observation closest to the others in G_i.

$$\mathbf{x}(s) \quad \leftarrow \quad \underset{\mathbf{x}(s'):s',t\in\mathcal{Z}|L(t)=L(s')=i}{\arg\min} \|\mathbf{x}(t)-\mathbf{x}(s')\|^2.$$

  The observation $\mathbf{x}(s)$ is called medoid and the method becomes k-medoids.

Coding

Once the codebook is defined, for every training/test image,

- key-points are detected and local descriptors are extracted from patches.

Coding

Once the codebook is defined, for every training/test image,

- key-points are detected and local descriptors are extracted from patches.

- A distance function (e.g., Euclidean) is used to compare patch descriptors and visual words.

Coding

Once the codebook is defined, for every training/test image,

- key-points are detected and local descriptors are extracted from patches.
- A distance function (e.g., Euclidean) is used to compare patch descriptors and visual words.
- The images are usually coded by either
  - hard assigment: each patch counts for its closest visual word only.

Coding

Once the codebook is defined, for every training/test image,

- key-points are detected and local descriptors are extracted from patches.
- A distance function (e.g., Euclidean) is used to compare patch descriptors and visual words.
- The images are usually coded by either
  - hard assigment: each patch counts for its closest visual word only.
  - soft assignment: each patch counts for all visual words with count directly proportional to the similarity between patch and word.

Coding

- Hard assignment results one visual-word histogram per image – a feature vector of size k.

Coding

- Hard assignment results one visual-word histogram per image – a feature vector of size k.

- Soft assignment results a feature vector of size $k \times n$, where n is the number of patches (key points) per image.

- Hard assignment results one visual-word histogram per image – a feature vector of size k.
- Soft assignment results a feature vector of size $k \times n$, where n is the number of patches (key points) per image.
- The former loses spatial information (localization) of the visual words while the latter maintains that information.

Coding

- Hard assignment results one visual-word histogram per image – a feature vector of size k.
- Soft assignment results a feature vector of size $k \times n$, where n is the number of patches (key points) per image.
- The former loses spatial information (localization) of the visual words while the latter maintains that information.
- When all pixels (dense sampling) are used as key points, soft assignment resembles convolution with a kernel bank [3].

Challenges and opportunities

- In the next lectures, we will perceive some similarities between visual words and kernels of convolutional neural networks (CNNs), codebook and kernel bank, coding by soft assignment and convolution with a kernel bank.

Challenges and opportunities

- In the next lectures, we will perceive some similarities between visual words and kernels of convolutional neural networks (CNNs), codebook and kernel bank, coding by soft assignment and convolution with a kernel bank. Is there a relation between BoVW and a convolutional layer?

Challenges and opportunities

- In the next lectures, we will perceive some similarities between visual words and kernels of convolutional neural networks (CNNs), codebook and kernel bank, coding by soft assignment and convolution with a kernel bank. Is there a relation between BoVW and a convolutional layer?

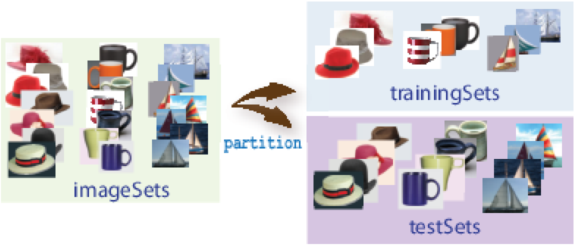
- BoVW uses an unsupervised technique to create the codebook.

Challenges and opportunities

- In the next lectures, we will perceive some similarities between visual words and kernels of convolutional neural networks (CNNs), codebook and kernel bank, coding by soft assignment and convolution with a kernel bank. Is there a relation between BoVW and a convolutional layer?

- BoVW uses an unsupervised technique to create the codebook. Can we explore class information to improve the codebook? [1, 2].

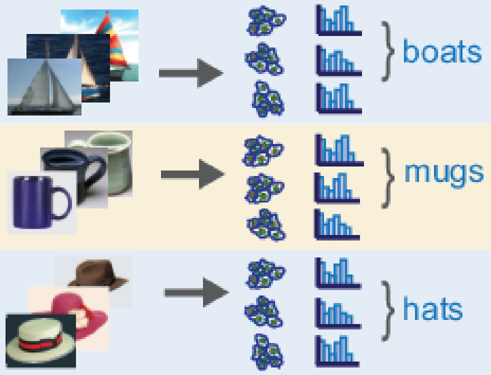
Challenges and opportunities



from MathWorks.com

One can separate training images from each category, build and merge the codebooks, code the training images, and train a classifier, but....

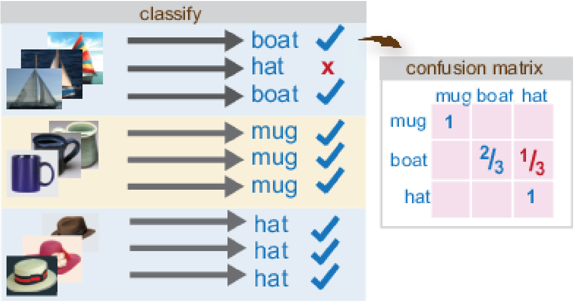
Challenges and opportunities



from MathWorks.com

One can separate training images from each category, build and merge the codebooks, code the training images, and train a classifier, but....

Challenges and opportunities



from MathWorks.com

One can separate training images from each category, build and merge the codebooks, code the training images, and train a classifier, but....

Challenges and opportunities

- How do we choose the key patches, local features, clustering technique, similarity function, image coding rule?

Challenges and opportunities

- How do we choose the key patches, local features, clustering technique, similarity function, image coding rule?
- Can we identify and select the most discriminative visual words, when merging codebooks from multiple classes?

Challenges and opportunities

- How do we choose the key patches, local features, clustering technique, similarity function, image coding rule?
- Can we identify and select the most discriminative visual words, when merging codebooks from multiple classes?
- Can we select visual words from markers (strokes) drawn by an expert on image regions that discriminate classes?

Challenges and opportunities

- How do we choose the key patches, local features, clustering technique, similarity function, image coding rule?
- Can we identify and select the most discriminative visual words, when merging codebooks from multiple classes?
- Can we select visual words from markers (strokes) drawn by an expert on image regions that discriminate classes?
- Is the BoVW based on image markers an effective descriptor?

[1] C. Castelo-Fernandez and A.X. Falcão.
Learning visual dictionaries from class-specific superpixel segmentation.
In *18th Computer Analysis of Images and Patterns (CAIP)*, volume 11678, pages 171–182, 2019.

[2] C. Castelo-Fernandez and A.X. Falcão.
Improving supervised superpixel-based codebook representations by local convolutional features.
In *24th European Conference on Artificial Intelligence (ECAI)*, volume 325 of *Frontiers in Artificial Intelligence and Applications*, pages 1055–1062, 2020.

[3] Ian Goodfellow, Yoshua Bengio, and Aaron Courville.
Deep Learning.
MIT Press, 2016.
http://www.deeplearningbook.org.