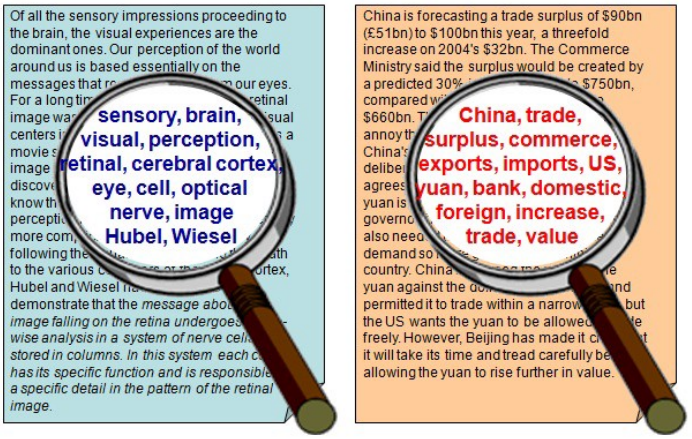
# Bag of Visual Words in a Nutshell

## The art of choosing important features

Bethea Davida Follow
Jul 3, 2018 · 3 min read

Bag-of-visual-words (BOVW)

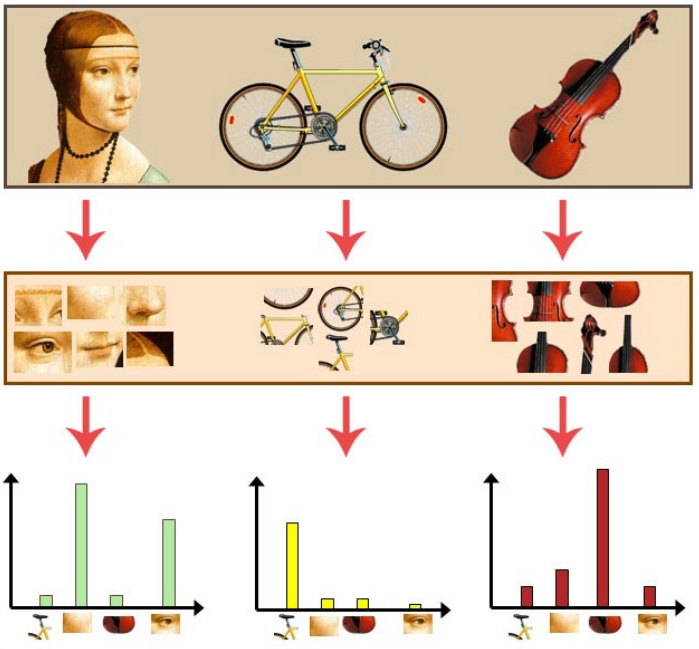
Bag of visual words (BOVW) is commonly used in image classification. Its concept is adapted from information retrieval and NLP's bag of words (BOW). In bag of words (BOW), we count the number of each word appears in a document, use the frequency of each word to know the keywords of the document, and make a frequency histogram from it. We treat a document as a bag of words (BOW). We have the same concept in bag of visual words (BOVW), but instead of words, we use image features as the "words". Image features are unique pattern that we can find in an image.




Keywords in documents

## What is bag of visual words (BOVW)?

The general idea of bag of visual words (BOVW) is to represent an image as a set of features. Features consists of keypoints and descriptors. Keypoints are the “stand out” points in an image, so no matter the image is rotated, shrink, or expand, its keypoints will always be the same. And descriptor is the description of the keypoint. We use the keypoints and descriptors to construct vocabularies and represent each image as a frequency histogram of features that are in the image. From the frequency histogram, later, we can find another similar images or predict the category of the image.

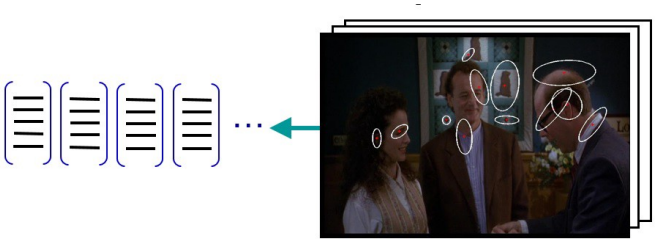


Histogram of visual words

## How to build a bag of visual words (BOVW)?

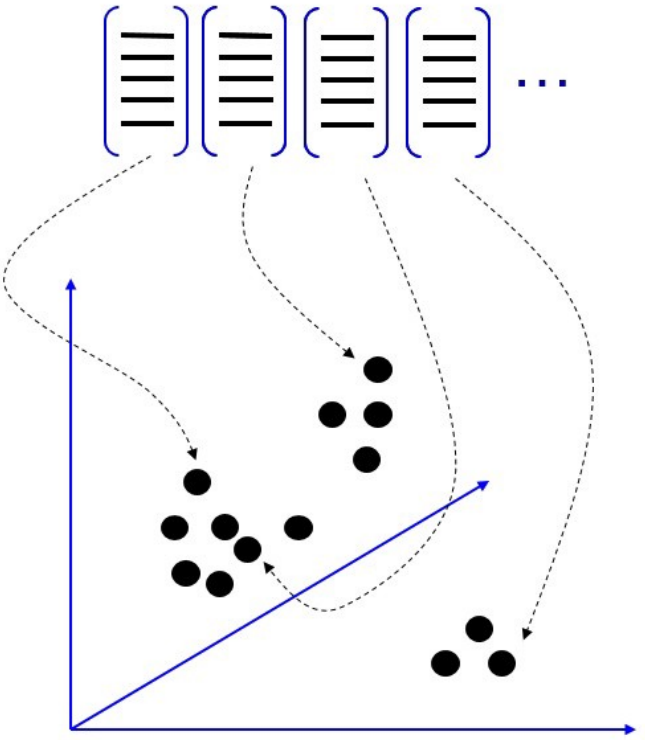
We detect features, extract descriptors from each image in the dataset, and build a visual dictionary. Detecting features and extracting descriptors in an image can be done by using feature extractor algorithms (for example, SIFT, KAZE, etc).

```
import cv2

# defining feature extractor that we want to use
extractor = cv2.xfeatures2d.SIFT_create()

def features(image, extractor):
```



Detecting features and extracting descriptor

Next, we make clusters from the descriptors (we can use K-Means, DBSCAN or another clustering algorithm). The center of each cluster will be used as the visual dictionary's vocabularies.



Descriptors clustering

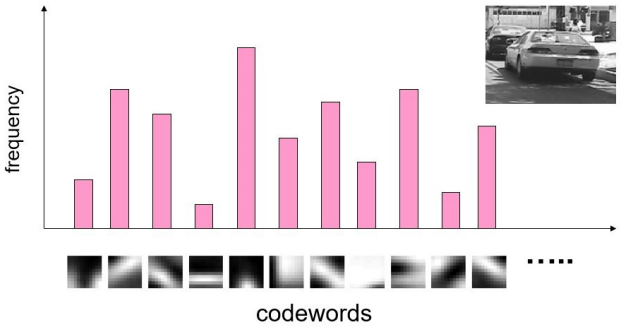
Finally, for each image, we make frequency histogram from the vocabularies and the frequency of the vocabularies in the image. Those histograms are our bag of visual words (BOVW).

```
from sklearn.cluster import KMeans

kmeans = KMeans(n_clusters = 800)
kmeans.fit(descriptor_list)

preprocessed_image = []
for image in images:
      image = gray(image)
```

## I have an image and I want to find another 20 similar images from the dataset. How can I do that?

Given another image (whether from the dataset or not), as before, we detect features in the image, extract descriptors from the image, cluster the descriptors, and build histogram with the same length with previous histogram. By using bag of visual words representation from our dataset, we can compute this image's nearest neighbors. We can do it by using nearest neighbors algorithm or another algorithm.

```python
from sklearn.neighbors import NearestNeighbors

data = cv2.imread(image_path)
data = gray(data)
keypoint, descriptor = features(data, extractor)
histogram = build_histogram(descriptor, kmeans)
neighbor = NearestNeighbors(n_neighbors = 20)
```




Image's histogram

## Reference :

Recognizing and Learning Object Categories

Awarded the Best Short Course Prize at ICCV 2005 Recognizing and Learning Object Categories Li F...

people.csail.mit.edu