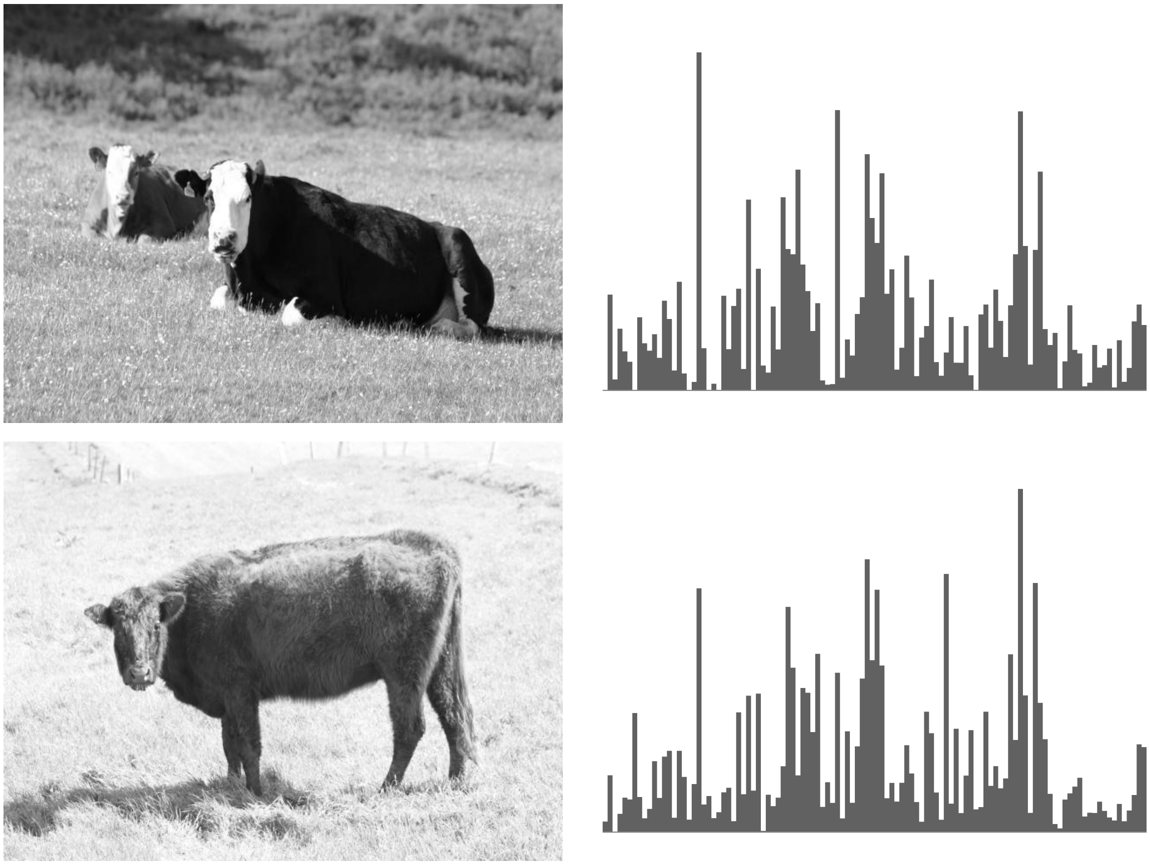
University of Cambridge
Engineering Part IB
Paper 8 Information Engineering

Handout 4: Visual Words



Roberto Cipolla and Matthew Johnson

May 2008

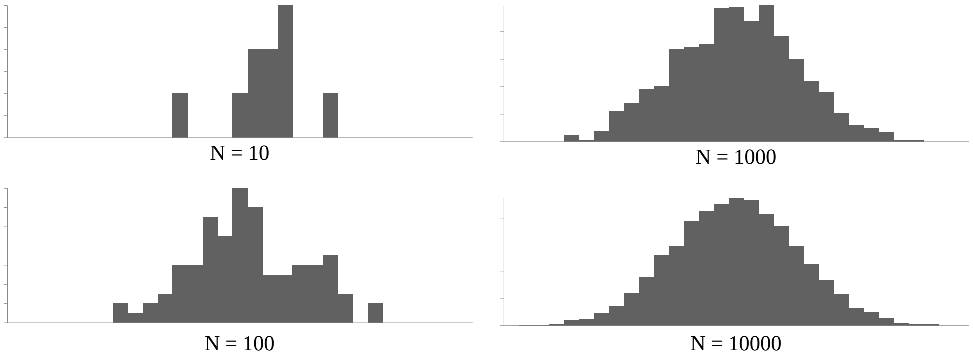
A Thousand Words

The old adage that an image is worth a thousand words is actually quite accurate. In this lecture we are going to discuss one of the practical uses for the detectors and descriptors you have been learning about in the past two weeks: category recognition.

Category recognition is the task of looking at something (in our case, an image) and determining what category it belongs to. We're going to discuss how turning an image into a thousand words makes that possible. Before we can talk about that though, we have to learn about histograms.

The Humble Histogram

A histogram consists of a discrete number of “bins”, which count occurrences of a kind of data sample within a population. For example, if you roll 6 dice and treat the sum of the pips as the data sample, you receive a progression of histograms like the following:



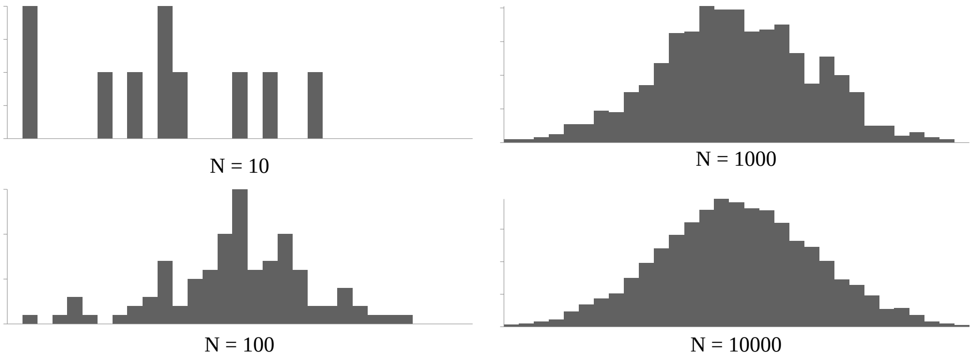
N = 10

N = 1000

N = 100

N = 10000

They are useful method for summarizing data that is continuous as well. In those cases, the bins typically correspond to a range of data values. Determining the right range is a bit of a black art. Let’s take, for example, average heights for men in the United Kingdom and pretend this data is normally distributed around its mean, 175.1 cm. If we had only a few samples (let’s say, the men in this classroom) then we’d use large bin sizes, but if we had the entire population, we can afford to have more bins.



N = 10

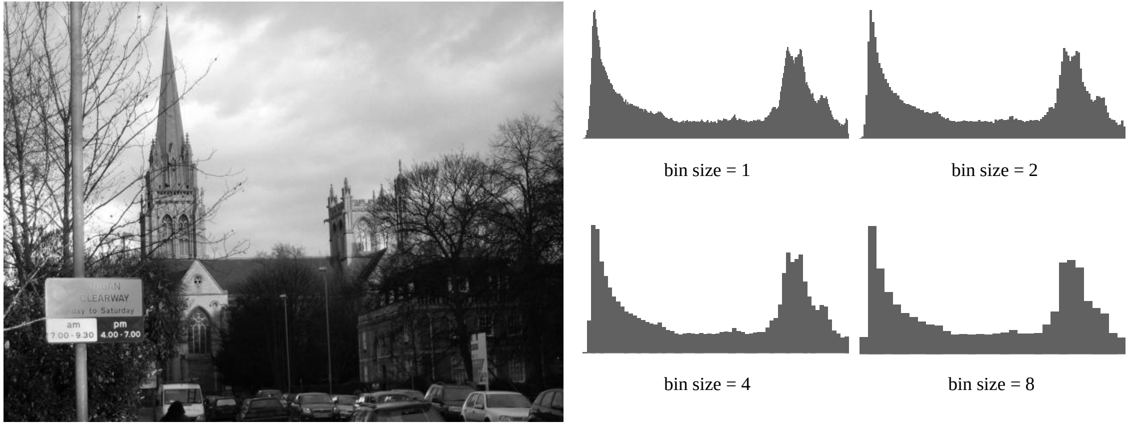
N = 1000

N = 100

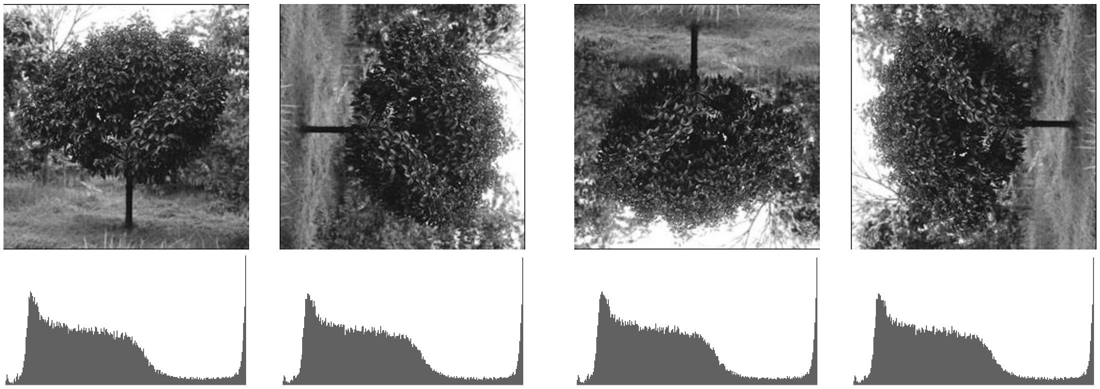
N = 10000

Intensity Histograms

For images, the most common kind of histogram is the intensity histogram. A gray-scale image, where each pixel is stored as a value from 0 to 255 (where 0 is black and 255 is white), can be described by a histogram with 255 bins, or fewer bins which cover the range.

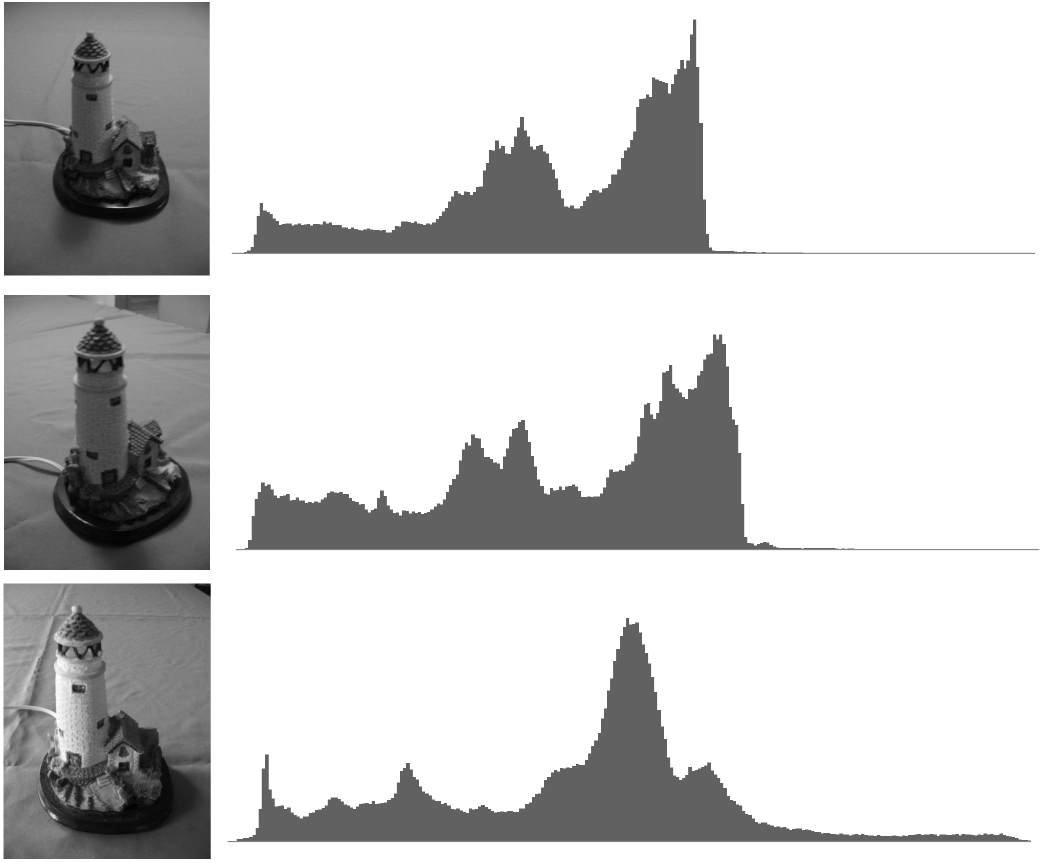


The intensity histogram has its uses as an image descriptor. It is invariant to rotation, for one thing, and is very easy to compute.



Intensity Histograms, cont.

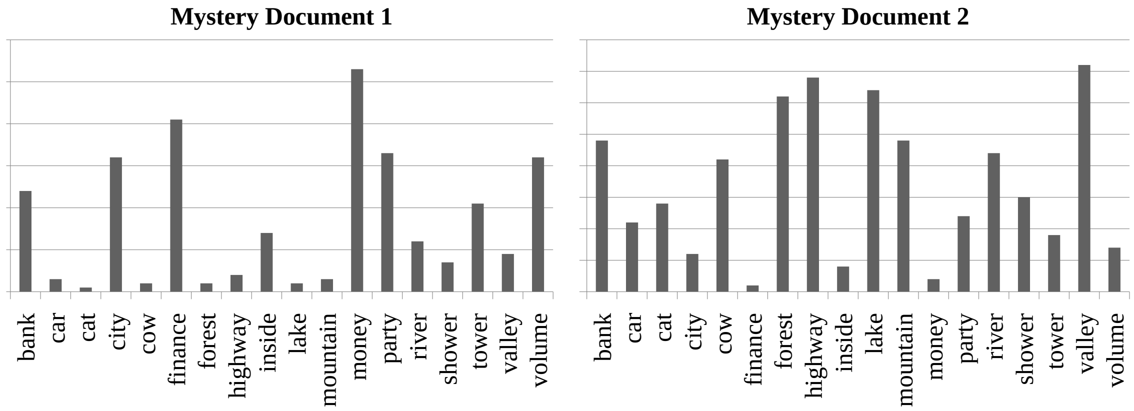
Rotation is just about all it is invariant to, however. Any number of other common image changes create very different histograms.



While there are techniques which help with some of these problems, histograms in computer vision were relegated to a set of specific uses and otherwise passed over in favor of newer, better descriptors and methods.

Word Histograms

Meanwhile, researchers in textual analysis and data mining were faced with a problem. They needed a way to represent documents so that similar documents could be easily retrieved from a large database. Doing full-text searches was out of the question, and so they needed a compact representation of a document which could be easily stored and matched against a query. Most importantly, whose size was independent of document length. In short, they needed histograms.



By building a word histogram which counted the occurrences of common words in a document, they found a solution to their problem. Not only did this allow the matching of similar documents, it also allowed documents to be sorted into categories by looking at which words were most prominent.

TF/IDF

There were two unresolved issues, however. The first was that documents had vastly different lengths and thus varying count magnitudes for each bin, and the second was that particularly common terms (such as "person", "place" or "thing") were dominating the histograms and drowning out words whose presence was more meaningful. The solution to the first was **term frequency**, and the solution to the second was **inverse document frequency**. Together, they made word histograms even more useful.

Term frequency is the normalized occurrence of a term, t_i in a document.

$$tf_i = \frac{n_i}{\sum_k n_i}$$

where n_i is the number of times the word occurs. Inverse document frequency, on the other hand, measures how many times a word is used in a set of documents, D.

$$idf_i = \log \frac{|D|}{|\{d : t_i \in d\}|}$$

By multiplying the two together ($tfidf_i = tf_i \times idf_i$) the resulting term is invariant to document length and stresses rare terms over common terms. With this refinement, word histograms showed remarkable success for document categorization and retrieval.

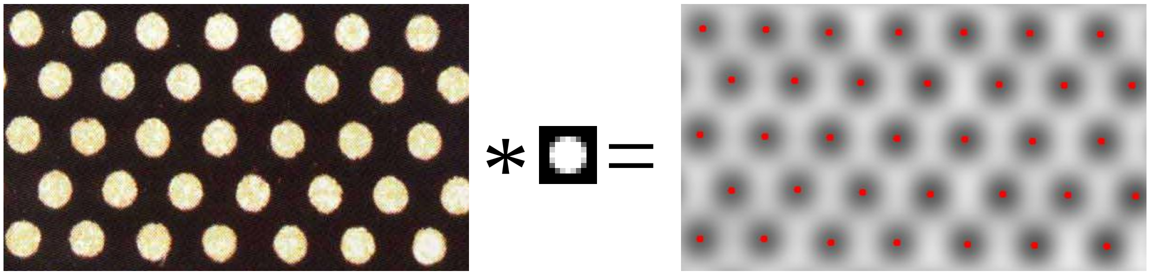
Visual Words

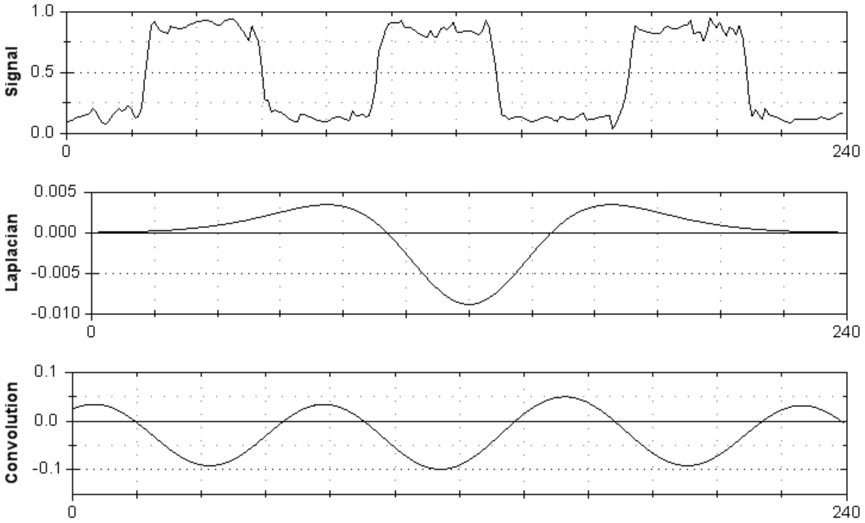
In the grand tradition of academic research, computer vision scientists wanted to find a way to steal this concept wholesale for use in image categorization and retrieval. The problem, however, lay in finding a way of turning an image into a thousand words. Thus, the concept of the visual word was born. What is a visual word?

A word consists of two parts: a symbol, and a definition. The symbol is the collection of letters which acts as a marker within a document that points to its definition. When we determine which words are similar, we do not compare their symbols. Rather, we compare their definitions. What two things have you learned about in the past few lectures which correspond to symbol and definition?

The Symbol

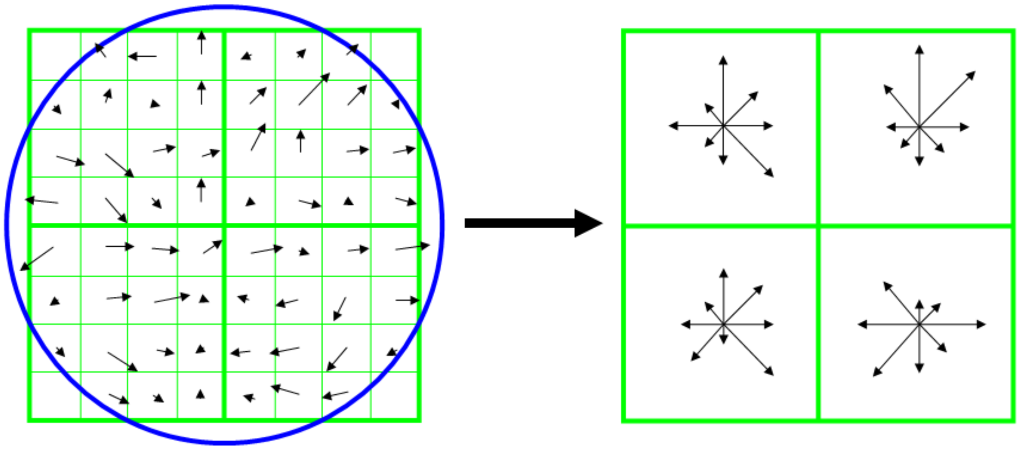
Feature detectors play a role in images which is similar to the role played by groups of letters on a page, in that they are interesting markers which can be found in many images. In the most common implementation of visual words, the detector used is a blob detector. The beauty of a blob detector is that it fires at the centers of objects at an appropriate scale for that object.





The Definition

Feature descriptors play the role of definition to the interest points detected by feature detectors. They describe the area around an interest point in a way which is determined by the contents of the image and which is similar in different images of the same thing. For example, a detector which fired at bicycle wheels (such as a blob detector) would have a similar descriptor in different images of bicycles. The descriptor most commonly used is the SIFT descriptor.



The SIFT (**S**cale-**I**nvariant **F**eature **T**ransform) descriptor is an orientation histogram-based descriptor, and is based upon describing the edges in the neighborhood of an interest point in a way which is robust to translation, rotation and scale.

Building a Dictionary

Now that we have a way to extract visual words from an image, we need a dictionary. In textual analysis, this was the easy part: buy a dictionary. We have a different problem, in that we need to compile a dictionary given the raw language. When building a text dictionary, you first do a survey of all the words in use and then find a public consensus of their definition. If we have a database of images, then we have the survey; what we need is to determine the consensus of their definitions.

To do this, these words first have to be grouped into sets of synonyms which have the same definition. Once we have these groups, we need a method of determining which group a new word belongs to. All we have to work with are the feature descriptors, which take the form of real-valued vectors which (we assume) can be compared using Euclidean distance. We need to divide these vectors into groups which are similar, essentially cluster them together. This is a classic problem in machine learning, and we are going to use a classic solution.

K Means

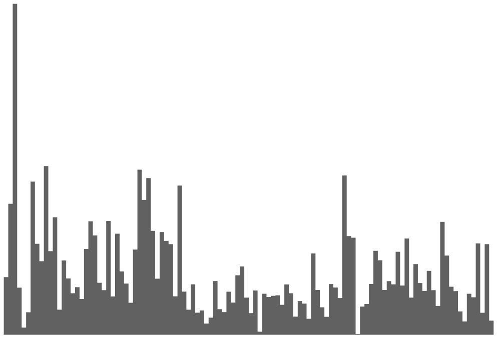
K Means is the simplest, most efficient, and easiest to understand algorithm for clustering together real-valued vectors that can be compared using a Euclidean distance metric. This results in it being the most commonly used algorithm for the task. We start with a collection of data vectors, D, which have each been randomly assigned to one of K clusters. The algorithm consists of two steps, which are repeated until no vector changes membership.

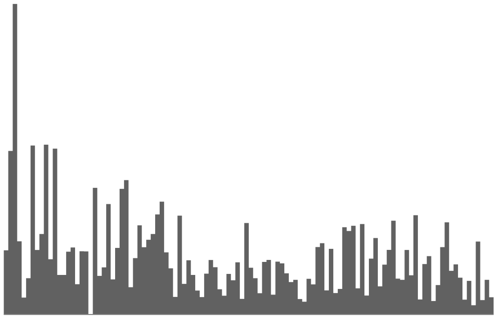
1. Compute a cluster center, c for each cluster as the mean of the cluster members.
2. Reassign each data point to the cluster whose center is nearest.

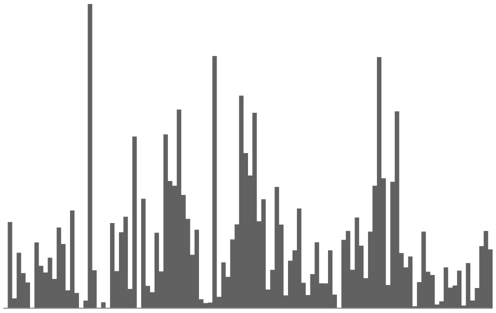
This is guaranteed to converge and the result is a set of data clusters and their associated mean values. For our purposes, these data clusters are the synonym sets and their centers are the consensus definition. The cluster centers are now a visual word dictionary.

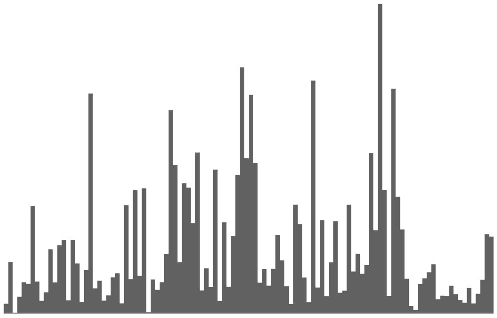
Example Histograms

Here are some bicycle images and their associated visual word histograms, using this technique.









Learning Categories

Now that we can create word histograms for individual images, we can begin to examine what makes them similar. We could cluster the histograms again, but that is only one of the many options we have available. The problem of determining the class or category of a training example is the most common one addressed in machine learning, and there are many options. They all share certain characteristics. First, each technique requires training data from the categories it must learn. Second, there is an initial process of learning which enables the technique to then categorize new, unseen images. Some of the commonly used techniques are:

- Nearest Neighbor
- Naïve Bayes
- Support Vector Machines
- Boosting
- Probabilistic Latent Semantic Analysis