Machine Learning Lab

Poster for Image Classification with Bag of Visual Words using MATLAB

Use the Computer Vision System Toolbox functions for image category classification by creating a bag of visual words. The process generates a histogram of visual word occurrences that represent an image. These histograms are used to train an image category classifier. The steps below describe how to setup your images, create the bag of visual words, and then train and apply an image category classifier.

Step 1: Set Up Image Category Sets

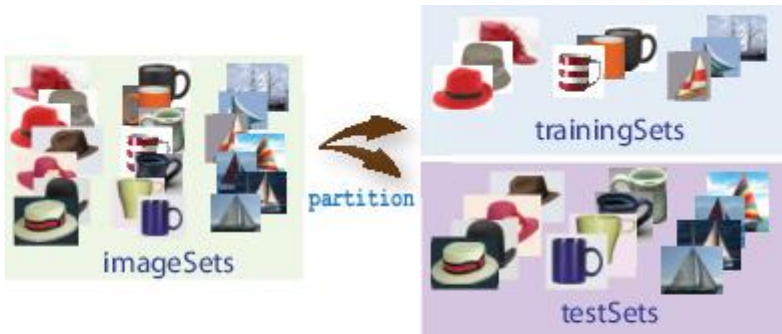
Organize and partition the images into training and test subsets. Use the `imageDatastore` function to store images to use for training an image classifier. Organizing images into categories makes handling large sets of images much easier. You can use the `splitEachLabel` function to split the images into training and test data.

Read the category images and create image sets.

```
setDir  = fullfile(toolboxdir('vision'),'visiondata','imageSets');
imds = imageDatastore(setDir,'IncludeSubfolders',true,'LabelSource',...'foldernames');
```

Separate the sets into training and test image subsets. In this example, 30% of the images are partitioned for training and the remainder for testing.

```
[trainingSet,testSet] = splitEachLabel(imds,0.3,'randomize');
```

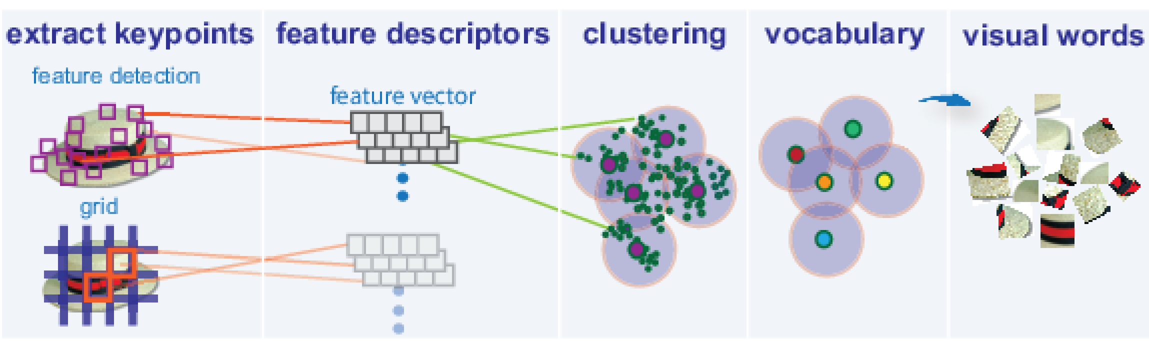


Step 2: Create Bag of Features

Create a visual vocabulary, or bag of features, by extracting feature descriptors from representative images of each category.

The `bagOfFeatures` object defines the features, or visual words, by using the k-means clustering (Statistics and Machine Learning Toolbox) algorithm on the feature descriptors extracted from `trainingSets`. The algorithm iteratively groups the descriptors into *k* mutually exclusive clusters. The resulting clusters are compact and separated by similar characteristics. Each cluster center represents a feature, or visual word.

You can extract features based on a feature detector, or you can define a grid to extract feature descriptors. The grid method may lose fine-grained scale information. Therefore, use the grid for images that do not contain distinct features, such as an image containing scenery, like the beach. Using speeded up robust features (or SURF) detector provides greater scale invariance. By default, the algorithm runs the `'grid'` method.

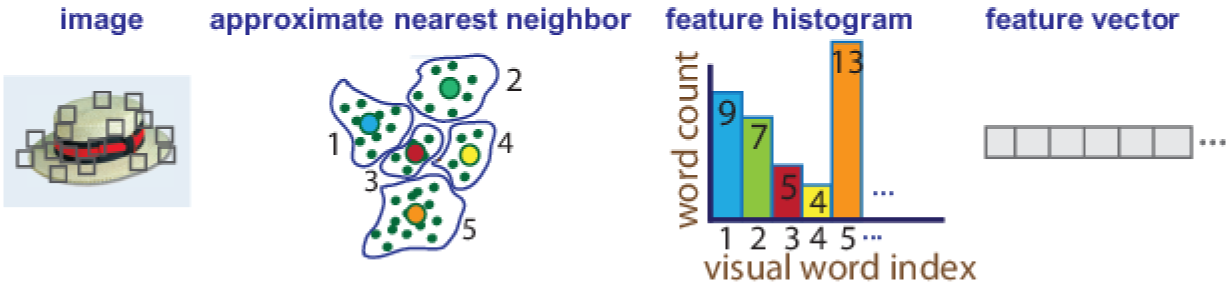


This algorithm workflow analyzes images in their entirety. Images must have appropriate labels describing the class that they represent. For example, a set of car images could be labeled cars. The workflow does not rely on spatial information nor on marking the particular objects in an image. The bag-of-visual-words technique relies on detection without localization.

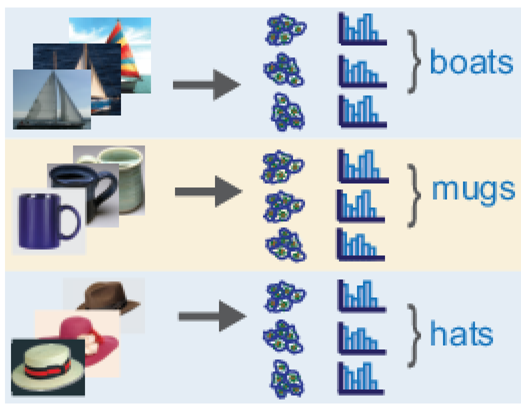
Step 3: Train an Image Classifier with Bag of Visual Words

The `trainImageCategoryClassifier` function returns an image classifier. The function trains a multiclass classifier using the error-correcting output codes (ECOC) framework with binary support vector machine (SVM) classifiers. The `trainImageCategoryClassfier` function uses the bag of visual words returned by the `bagOfFeatures` object to encode images in the image set into the histogram of visual words. The histogram of visual words are then used as the positive and negative samples to train the classifier.

1. Use the `bagOfFeatures` `encode` method to encode each image from the training set. This function detects and extracts features from the image and then uses the approximate nearest neighbor algorithm to construct a feature histogram for each image. The function then increments histogram bins based on the proximity of the descriptor to a particular cluster center. The histogram length corresponds to the number of visual words that the `bagOfFeatures` object constructed. The histogram becomes a feature vector for the image.



2. Repeat step 1 for each image in the training set to create the training data.



3. Evaluate the quality of the classifier. Use the `imageCategoryClassifier` `evaluate` method to test the classifier against the validation image set. The output confusion matrix represents the analysis of the prediction. A perfect classification results in a normalized matrix containing 1s on the diagonal. An incorrect classification results fractional values.