

```
% 7. Refer to the Bag of Features example MATLAB
% source code provided in the classroom's classwork
% page. In your homework, pick an object category
% that would be commonly seen in any household
% (e.g. cutlery) and pick 5 object types (e.g.
% for cutlery pick spoon, fork, butter knife,
% cutting knife, ladle). Present your performance
% evaluation.
```

```
%% Load Image Dataset
imds=imageDatastore('./office_images','IncludeSubfolders',true,'LabelSource','foldernames')

% inspect the number of images per category, as well as category labels
tbl=countEachLabel(imds)
```

tbl = 5x2 table

	Label	Count
1	headphones	10
2	monitor	10
3	pen	10
4	scissors	10
5	stapler	10

```
% visualization
figure
montage(imds.Files(1:16:end))
```



```
% Prepare Training and Validation Image Sets
[trainingSet, validationSet] = splitEachLabel(imds, 0.6, 'randomize');
```

```
% Create a Visual Vocabulary and Train an Image Category Classifier
```

```
% Creating Bag-Of-Features.
bag = bagOfFeatures(trainingSet);
```

Creating Bag-Of-Features.

```
-----
* Image category 1: headphones
* Image category 2: monitor
* Image category 3: pen
* Image category 4: scissors
* Image category 5: stapler
* Selecting feature point locations using the Grid method.
* Extracting SURF features from the selected feature point locations.
** The GridStep is [8 8] and the BlockWidth is [32 64 96 128].
```

```

* Extracting features from 30 images...done. Extracted 1875000 features.
* Keeping 80 percent of the strongest features from each category.
* Creating a 500 word visual vocabulary.
* Number of levels: 1
* Branching factor: 500
* Number of clustering steps: 1

* [Step 1/1] Clustering vocabulary level 1.
* Number of features      : 1500000
* Number of clusters      : 500
* Initializing cluster centers...100.00%.
* Clustering...completed 14/100 iterations (~3.14 seconds/iteration)...converged in 14 iterations.

* Finished creating Bag-Of-Features

```

```

% Encoding images using Bag-Of-Features.
img = readimage(imds, 1);
featureVector = encode(bag, img);

```

```

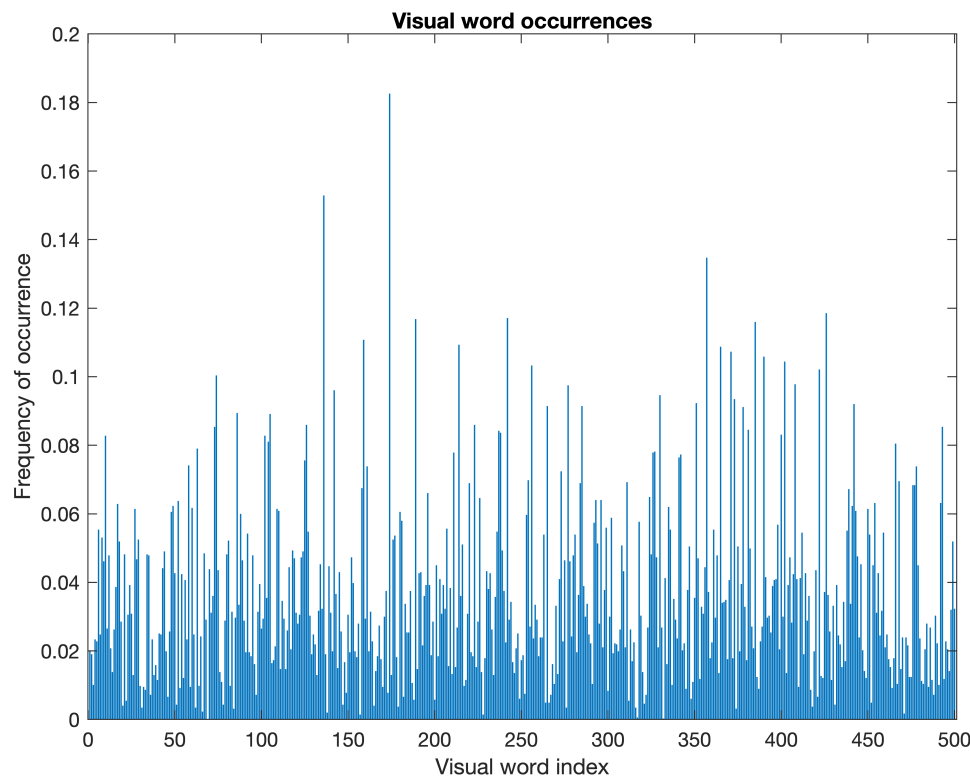
Encoding images using Bag-Of-Features.
-----
* Encoding an image...done.

```

```

% Plot the histogram of visual word occurrences
figure
bar(featureVector)
title('Visual word occurrences')
xlabel('Visual word index')
ylabel('Frequency of occurrence')

```



```
%% Training an image category classifier for 5 categories.
categoryClassifier = trainImageCategoryClassifier(trainingSet, bag);
```

Training an image category classifier for 5 categories.

```
-----
* Category 1: headphones
* Category 2: monitor
* Category 3: pen
* Category 4: scissors
* Category 5: stapler

* Encoding features for 30 images...done.

* Finished training the category classifier. Use evaluate to test the classifier on a test set.
```

```
%% Evaluate Classifier Performance
```

```
% on training set
confMatrix = evaluate(categoryClassifier, trainingSet);
```

Evaluating image category classifier for 5 categories.

```
-----
* Category 1: headphones
* Category 2: monitor
* Category 3: pen
* Category 4: scissors
* Category 5: stapler
```

```
* Evaluating 30 images...done.
* Finished evaluating all the test sets.
* The confusion matrix for this test set is:
```

KNOWN	PREDICTED				
	headphones	monitor	pen	scissors	stapler
headphones	1.00	0.00	0.00	0.00	0.00
monitor	0.00	1.00	0.00	0.00	0.00
pen	0.00	0.00	1.00	0.00	0.00
scissors	0.00	0.00	0.33	0.67	0.00
stapler	0.00	0.00	0.17	0.00	0.83

```
* Average Accuracy is 0.90.
```

```
% on validation set
confMatrix_val = evaluate(categoryClassifier, validationSet);
```

```
Evaluating image category classifier for 5 categories.
-----
```

```
* Category 1: headphones
* Category 2: monitor
* Category 3: pen
* Category 4: scissors
* Category 5: stapler

* Evaluating 20 images...done.
* Finished evaluating all the test sets.
* The confusion matrix for this test set is:
```

KNOWN	PREDICTED				
	headphones	monitor	pen	scissors	stapler
headphones	0.75	0.25	0.00	0.00	0.00
monitor	0.00	1.00	0.00	0.00	0.00
pen	0.00	0.00	1.00	0.00	0.00
scissors	0.00	0.00	0.75	0.25	0.00
stapler	0.00	0.50	0.00	0.00	0.50

```
* Average Accuracy is 0.70.
```

```
% Compute average accuracy
avg_acc = mean(diag(confMatrix_val));
```

```
%% show example
% read an image
img = imread(fullfile('./office_images', 'headphones', 'frame_0001.jpg'));
figure
imshow(img)
```



```
% run classification  
[labelIdx, scores] = predict(categoryClassifier, img);
```

Encoding images using Bag-Of-Features.

* Encoding an image...done.

```
labelName = categoryClassifier.Labels(labelIdx);  
disp(labelName)
```

```
{'headphones'}
```