Classifications, predictions, and recommendations with features

Wednesday

Anton Antonov
Digital Humanities at Oxford Summer School 2017
July 2017

Introduction

This notebook shows the making of a classifier using image keypoints.

Here are the steps.

1. Get a collection of images; (“CIFAR-10”).
    1.1. And split into training and testing data sets.
2. Compute image keypoints descriptors for each image.
3. Cluster the descriptors into 100 clusters.
4. Find the mean descriptor of each cluster.
5. For each image make a profile that is the number of image’s descriptors that belong to each of the clusters.
6. Make a classifier over the descriptor clusters profiles.
7. Evaluate the classifier.

Remark: Instead of profiles based on cluster membership (or proximity) we can use profiles derived from dimension reduction.

ImageKeypoints Applications example

Object recognition using "bag of words" on a dataset of 5,000 images 32×32 each, belonging to 10 categories:

```
obj = ResourceObject["CIFAR-10"];
training = ResourceData[obj, "TrainingData"];
testing = Flatten[Table[training[[i + 1000 ;; i + 1099]], {i, 1, 50 000, 5000}]];
training = Flatten[Table[training[[i ;; i + 499]], {i, 1, 50 000, 5000}]];
```

```
Length[training]
```

```
5000
```

```
Length[testing]
```

```
1000
```

Compute keypoint descriptors on 256×256 images and create the codebook of visual words:

```
AbsoluteTiming[
 kp = ParallelTable[
    ImageKeypoints[ImageResize[i, 256], "Descriptor"], {i, training[[All, 1]]}
   ];]
```

```
{33.0801, Null}
```

```
Length@ImageKeypoints[ImageResize[training[[1332, 1]], 256], "Descriptor"]
```

```
203
```

```
Dimensions[kp⟦1⟧]
```

```
{113, 64}
```

The codebook contains all image descriptors of length 64:

```
codebook = Flatten[kp, 1];
Dimensions[codebook]
```

```
{698 180, 64}
```

```
RecordsSummary[codebook]
AutoCollapse[]
```

Find 100 visual codewords using *k*-means clustering:

```
AbsoluteTiming[
 clust = ClusteringComponents[codebook,
    100, 1, Method -> "KMeans", PerformanceGoal → "Speed"];]
```

```
{296.813, Null}
```

This finds the mean point of each cluster.

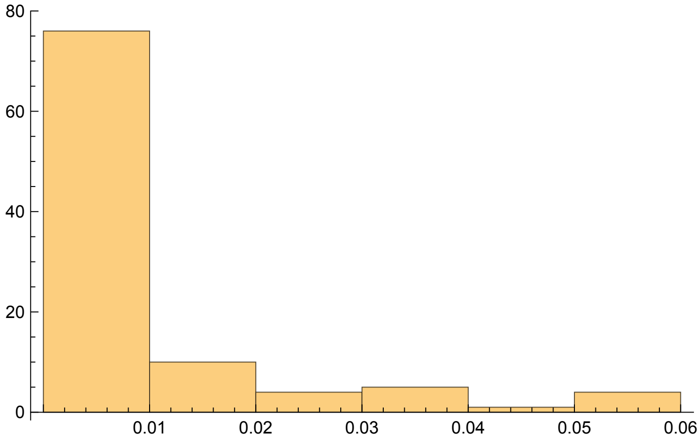
```
codewords = Table[
   p = Flatten@Position[clust, i];
   Mean[codebook[[p]]],
   {i, Max@clust}];
```

Image features are defined as the normalized counts of all the codewords:

```
nf = Nearest[codewords → Automatic];
```

```
imageCodewordsFeatureExtract[image_?ImageQ] := imageCodewordsFeatureExtract[
   ImageKeypoints[ImageResize[image, {256, 256}], "Descriptor"]];
imageCodewordsFeatureExtract[keypoints_List] := Module[
   {h},
   h = N@BinCounts[First@nf[#, 1] &/@keypoints, {1, Length@codewords + 1}];
   h / Total[h]
  ];
```

```
imageCodewordsFeatureExtract[RandomSample[training][[1, 1]]] // Histogram
```



Construct a classifier trained on extracted image features:

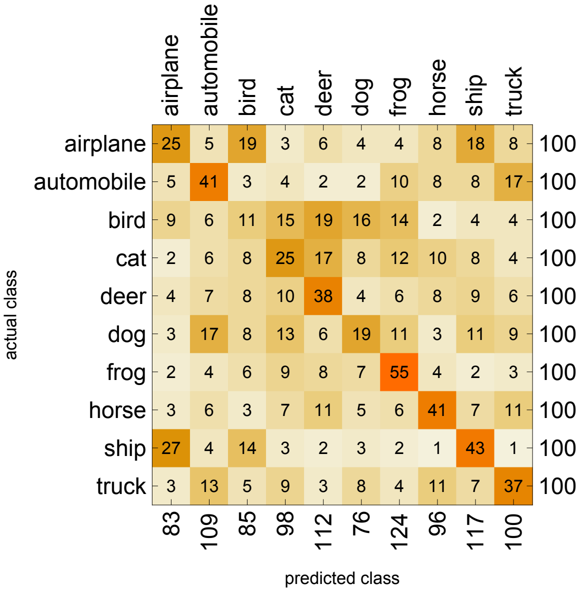
```
t = Table[imageCodewordsFeatureExtract[kp[[i]]] → training[[i, 2]],
   {i, Length@training}];
```

```
classifier = Classify[t];
```

Evaluate the classifier on test data:

```
cm = ClassifierMeasurements[classifier,
   Table[imageCodewordsFeatureExtract[First@sample] → Last@sample,
    {sample, testing}]];
```

```
cm["ConfusionMatrixPlot"]
```



	airplane	automobile	bird	cat	deer	dog	frog	horse	ship	truck	
airplane	25	5	19	3	6	4	4	8	18	8	100
automobile	5	41	3	4	2	2	10	8	8	17	100
bird	9	6	11	15	19	16	14	2	4	4	100
cat	2	6	8	25	17	8	12	10	8	4	100
deer	4	7	8	10	38	4	6	8	9	6	100
dog	3	17	8	13	6	19	11	3	11	9	100
frog	2	4	6	9	8	7	55	4	2	3	100
horse	3	6	3	7	11	5	6	41	7	11	100
ship	27	4	14	3	2	3	2	1	43	1	100
truck	3	13	5	9	3	8	4	11	7	37	100
	83	109	85	98	112	76	124	96	117	100	