Assignment 4: Object Classification

Input

airplane	butterfly	camera	helicopter

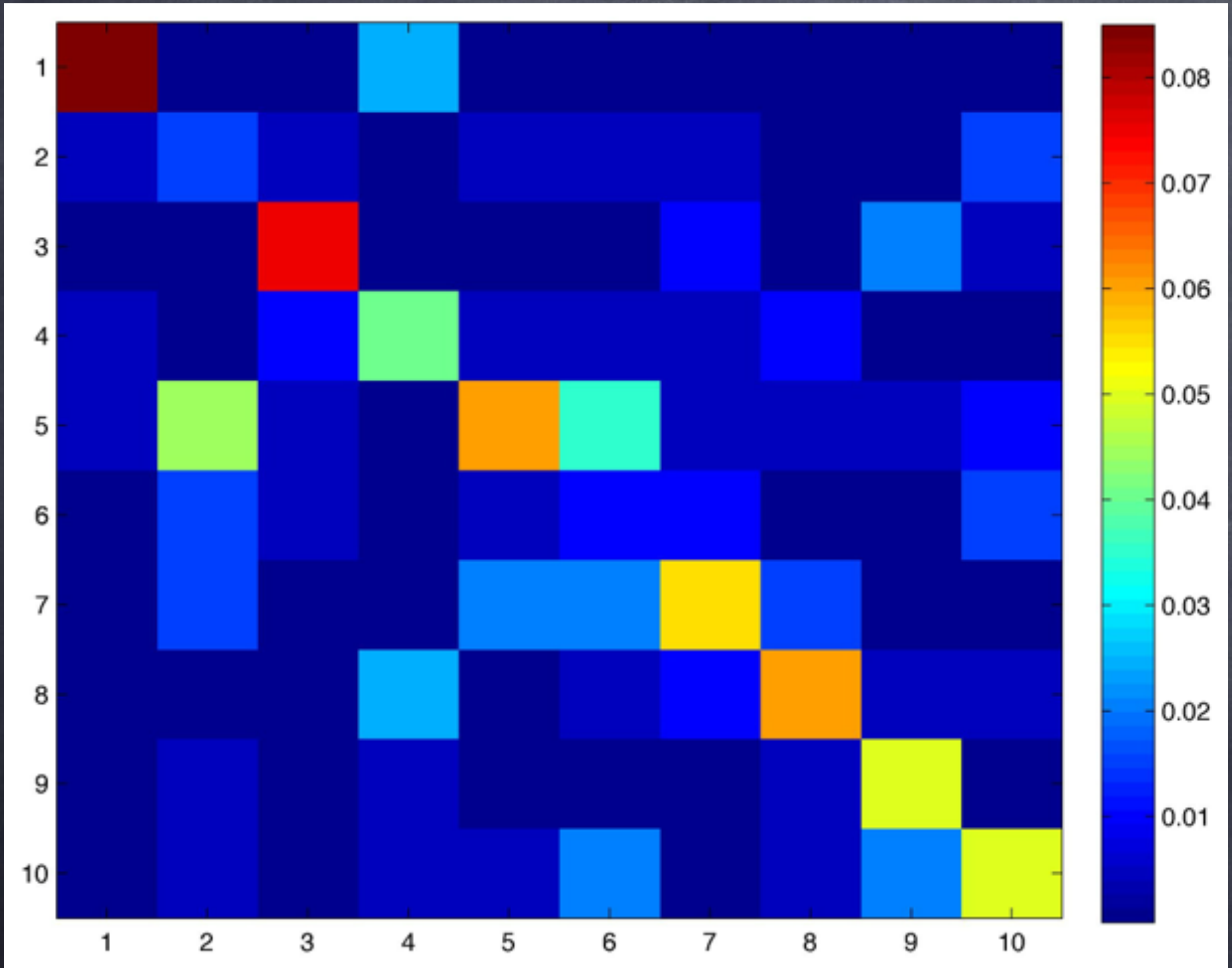
Also: lotus, panda, pizza, pyramid, snoopy, yin-yang

Your Task:

What am I?

10 categories, 10x10 train images, 10x20 test images

Results




```
Overall success rate: 0.50

Individual success rates:
airplane:       0.85
butterfly:      0.15
camera:         0.75
helicopter:     0.40
lotus:          0.60
panda:          0.10
pizza:          0.55
pyramid:        0.60
snoopy:         0.50
yin_yang:       0.50
```

How?

Train

Detect Features

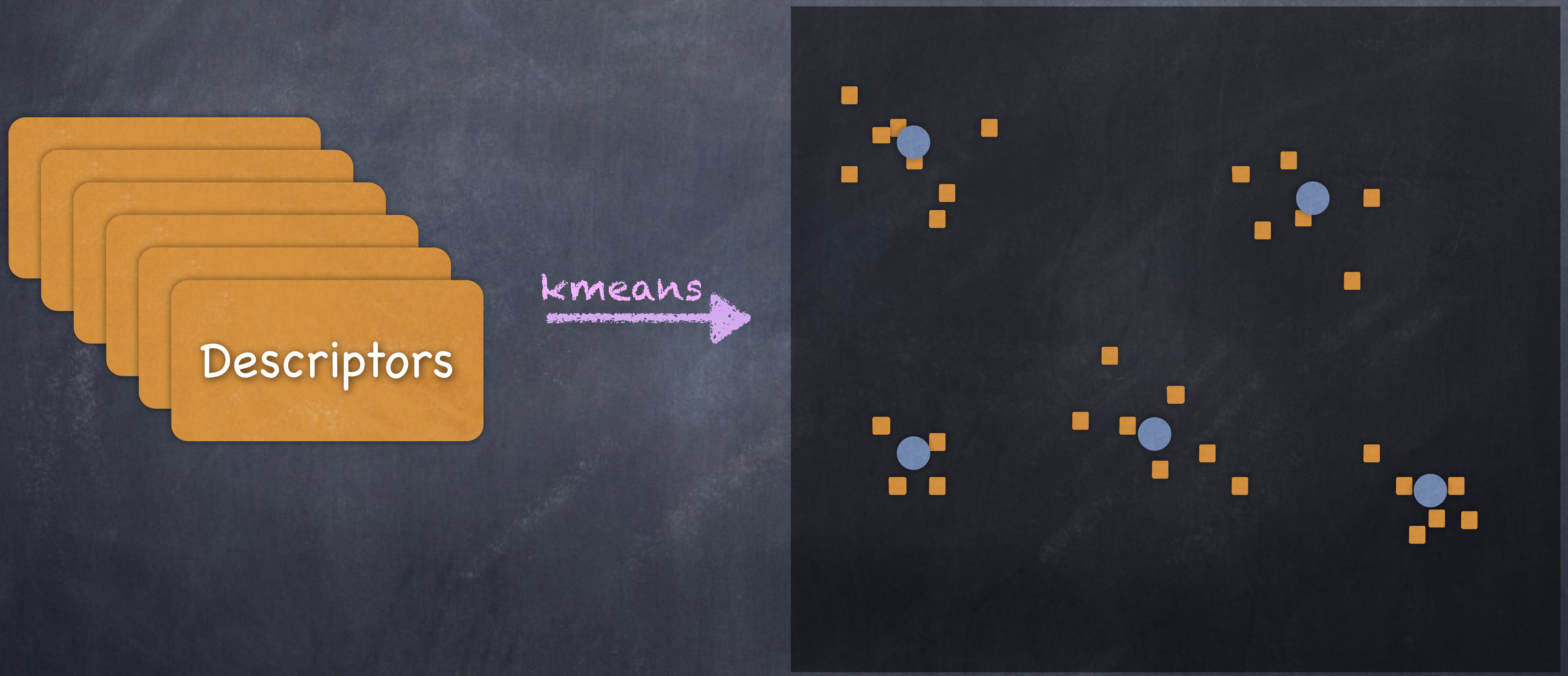
Compute Descriptors

Random
Harris
Sift
Etc.

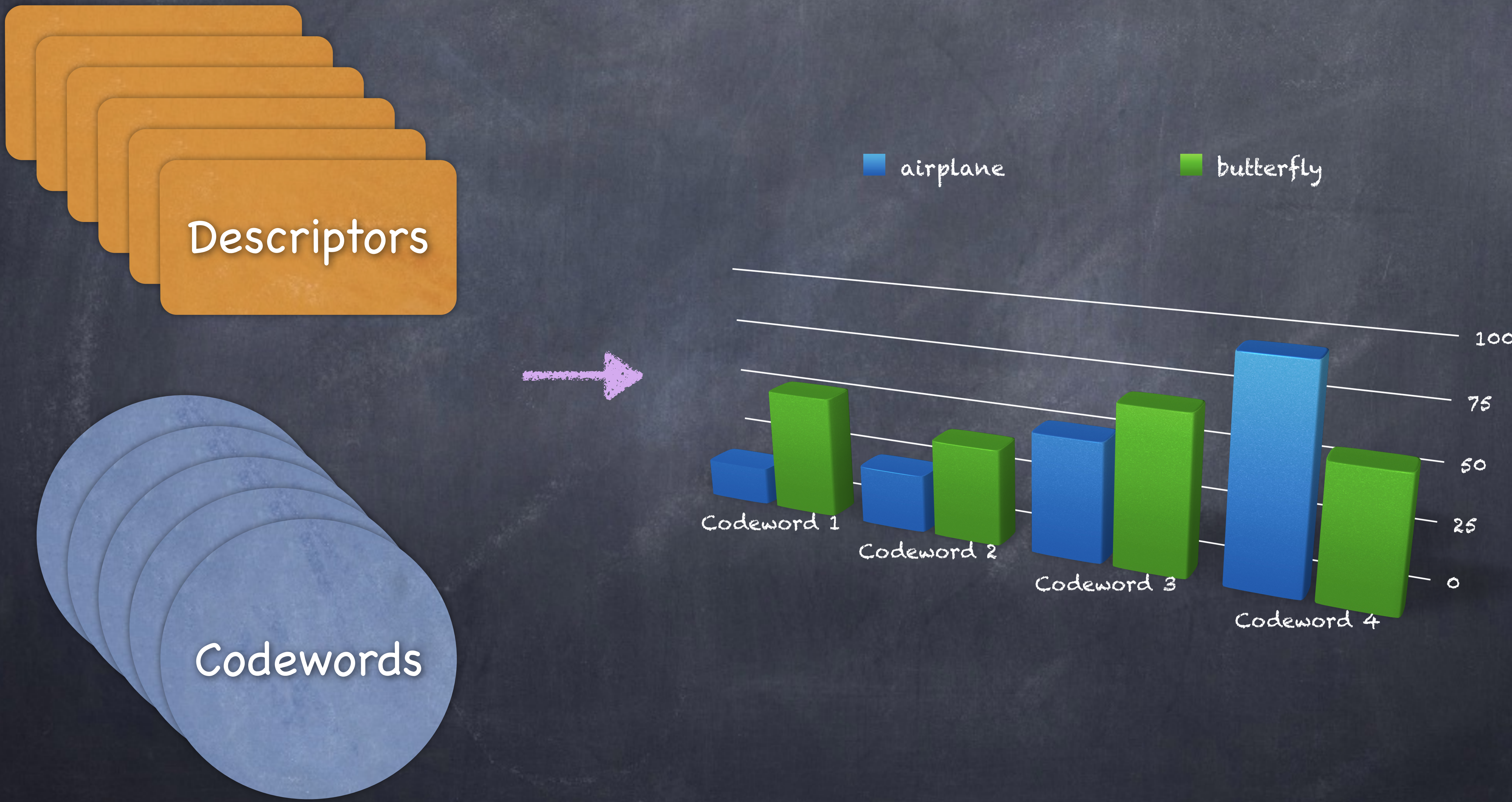
Window
Sift
Etc.

Should be familiar by now...

Train



Train


Descriptors
Codewords
airplane
butterfly
100
75
50
25
0
Codeword 1
Codeword 2
Codeword 3
Codeword 4

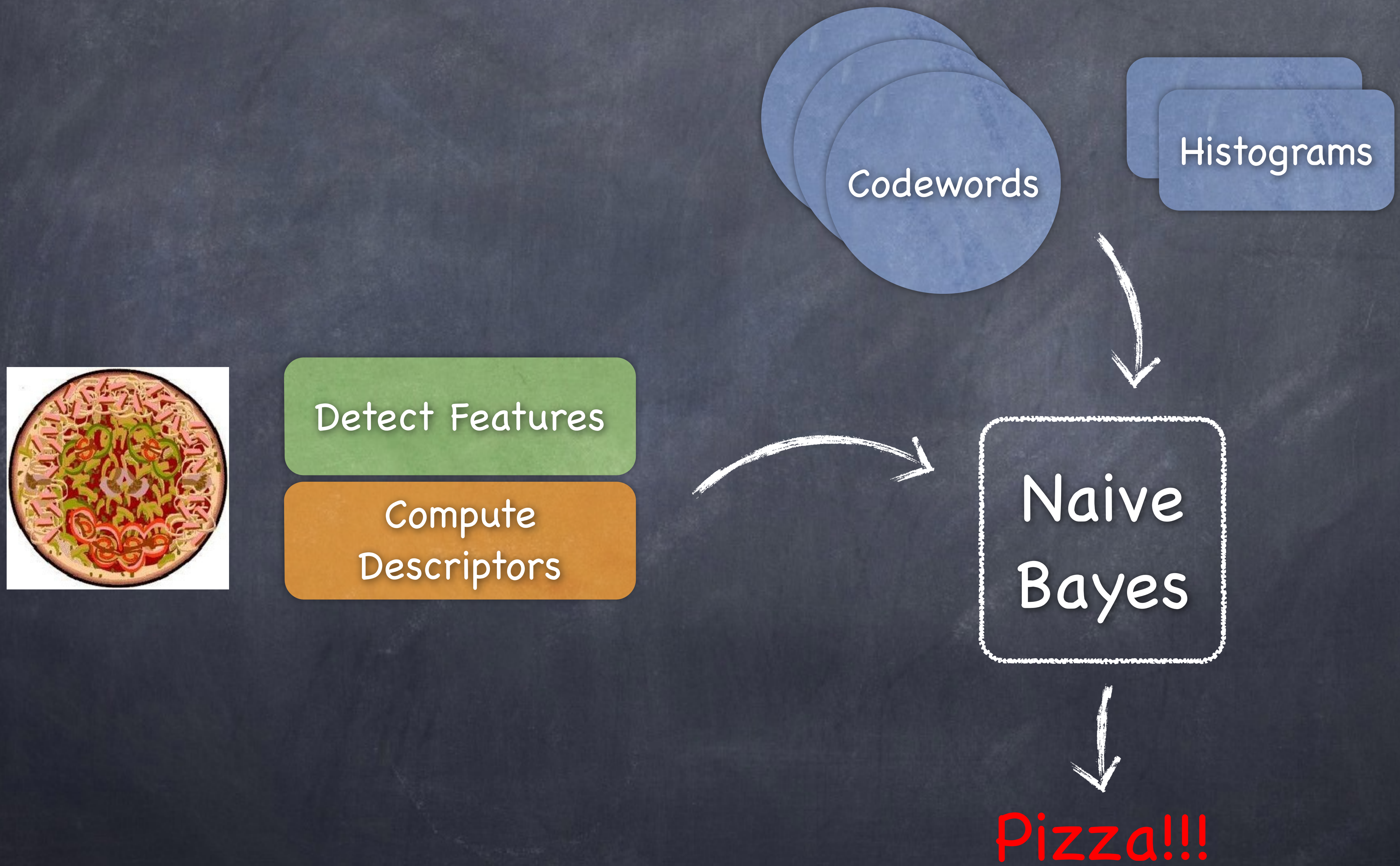
API Tips

[**IDX**, **C**] = kmeans(X, k)

[...] = kmeans(...,**param1**,**val1**,param2,val2,...)

'emptyaction'	Action to take if a cluster loses all its member observations.	
	'error'	Treat an empty cluster as an error (default).
	'drop'	Remove any clusters that become empty. kmeans sets the corresponding return values in C and D to NaN.
	'singleton'	Create a new cluster consisting of the one point furthest from its centroid.

Classify


Codewords
Histograms
Detect Features
Compute Descriptors
Naive Bayes
Pizza!!!

Naive Bayes

$$p(C_i|p_1, p_2, \ldots) = p(p_1, p_2, \ldots|C_i) \frac{p(C_i)}{p(p_1, p_2, \ldots)} \quad (1)$$

$$\propto p(p_1, p_2, \ldots|C_i) \quad (2)$$

$$= \prod_j p(p_j|C_i) \quad (3)$$

(1) - Bayes formula

(2) - Assume equal class priors, don't care about normalization.

(3) - Assume independent descriptors

Naive Bayes

$p(C_i|p_1, p_2, \ldots) \propto \prod_j p(p_j|C_i)$

Estimated class: $\arg\max_i \prod_j p(p_j|C_i)$

(using the fact that $p(p_j|C_i) \propto \text{count}\big(p_j \in \mathit{training}(C_i)\big)$)

2 more practical modifications...

Naive Bayes

Prevent zero counts

$$p(p_j|C_i) \propto \text{count}\big(p_j \in training(C_i)\big)$$

↓

$$p(p_j|C_i) \propto 1 + \text{count}\big(p_j \in training(C_i)\big)$$

Prevent overflow

Estimated class: $\arg\max_i \sum_j log(1 + \text{count}\big(p_j \in training(C_i)\big))$