Object Categorisation

CS 783: Visual Recognition

Review of main ideas of SIFT

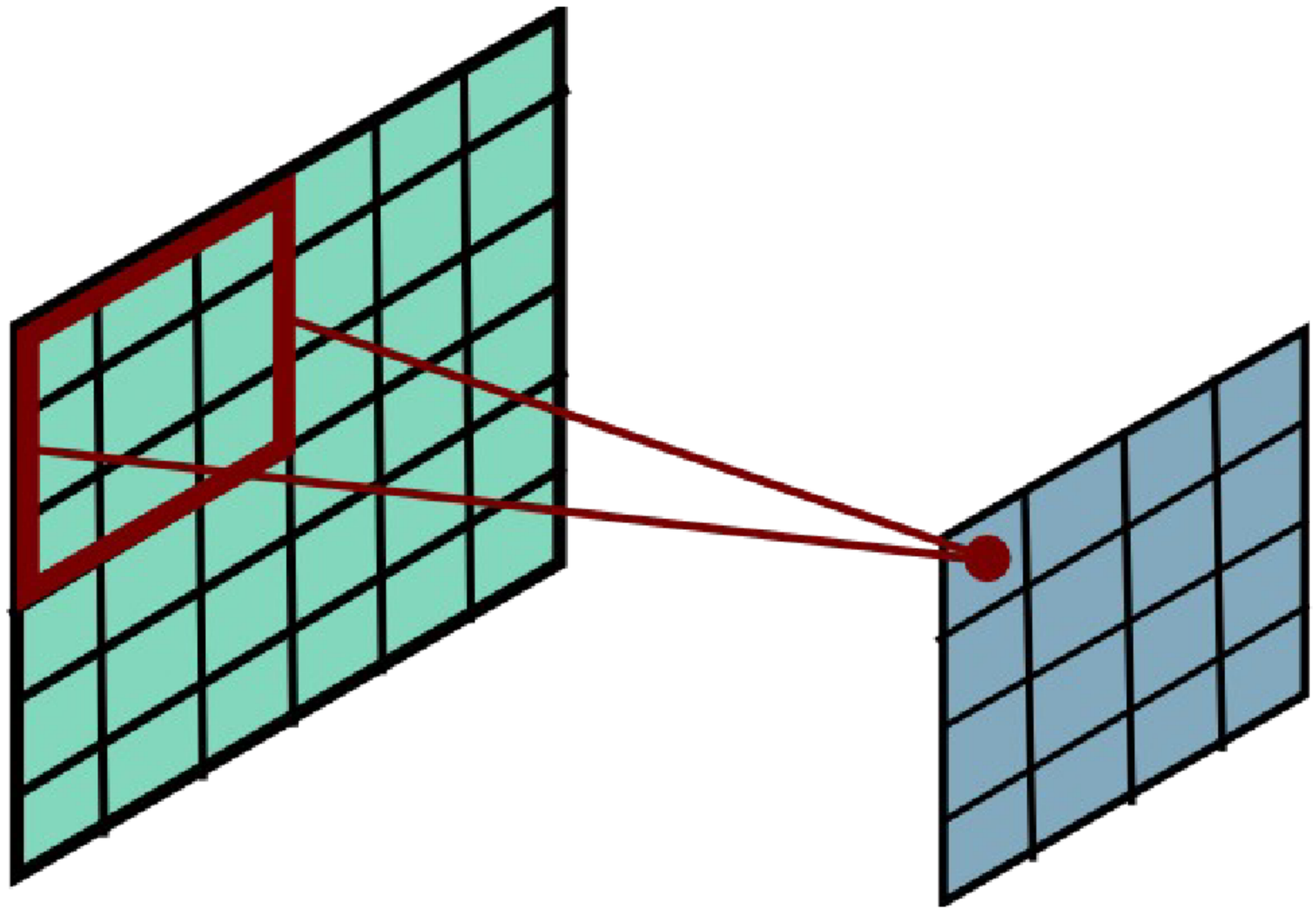
- Step 1: Convolve input image with a number of Gaussian filters
- Sample Gaussian filter:
-

This is a sample matrix, produced by sampling the Gaussian filter kernel (with σ = 0.84089642) at the midpoints largest value, decreasing symmetrically as distance from the center increases.

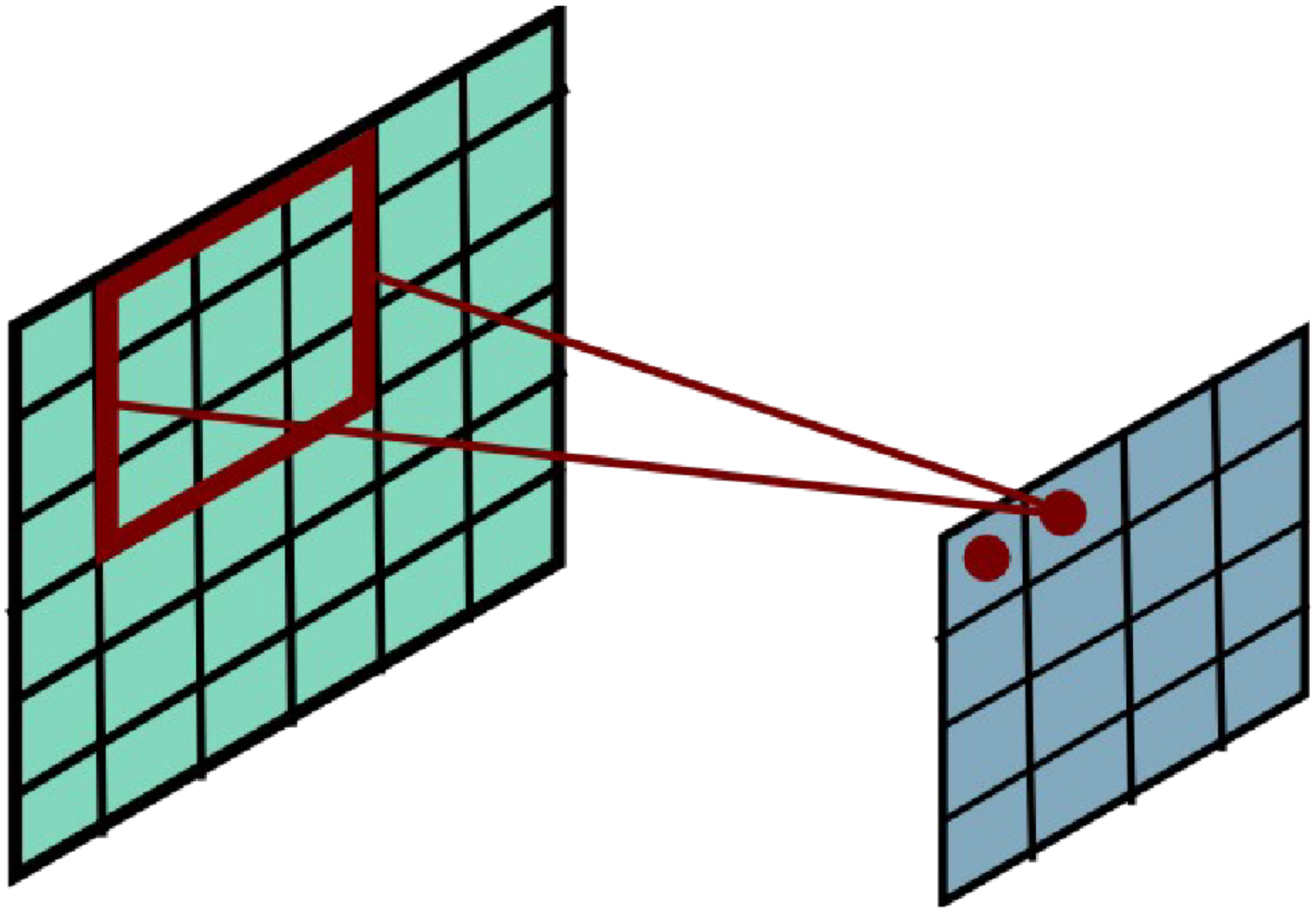
0.00000067	0.00002292	**0.00019117**	0.00038771	**0.00019117**	0.00002292	0.00000067
0.00002292	0.00078633	0.00655965	0.01330373	0.00655965	0.00078633	0.00002292
0.00019117	0.00655965	0.05472157	0.11098164	0.05472157	0.00655965	**0.00019117**
0.00038771	0.01330373	0.11098164	**0.22508352**	0.11098164	0.01330373	0.00038771
0.00019117	0.00655965	0.05472157	0.11098164	0.05472157	0.00655965	**0.00019117**
0.00002292	0.00078633	0.00655965	0.01330373	0.00655965	0.00078633	0.00002292
0.00000067	0.00002292	**0.00019117**	0.00038771	**0.00019117**	0.00002292	0.00000067

Note that 0.22508352 (the central one) is 1177 times larger than 0.00019117 which is just outside 3σ.

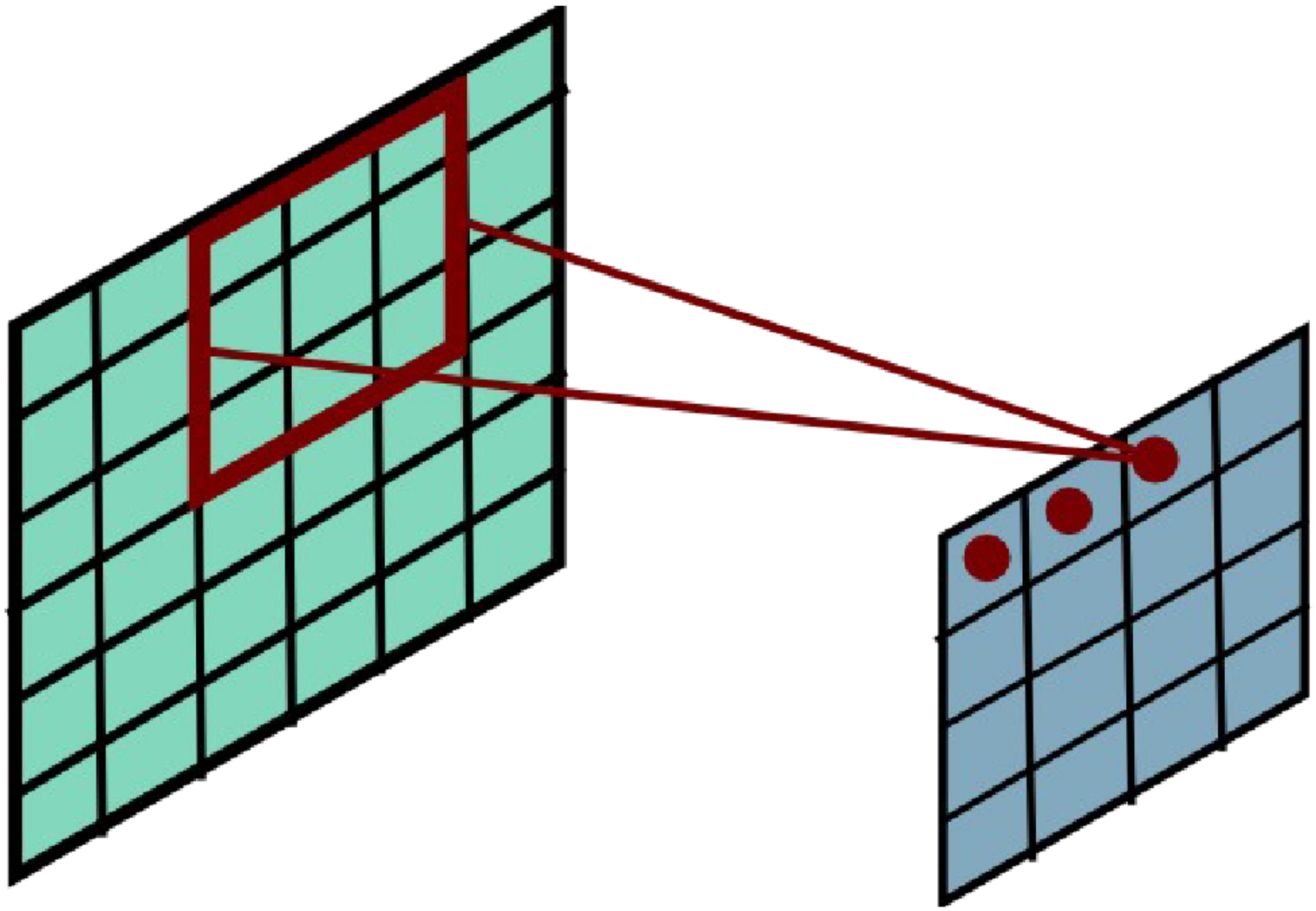
Convolutional Layer



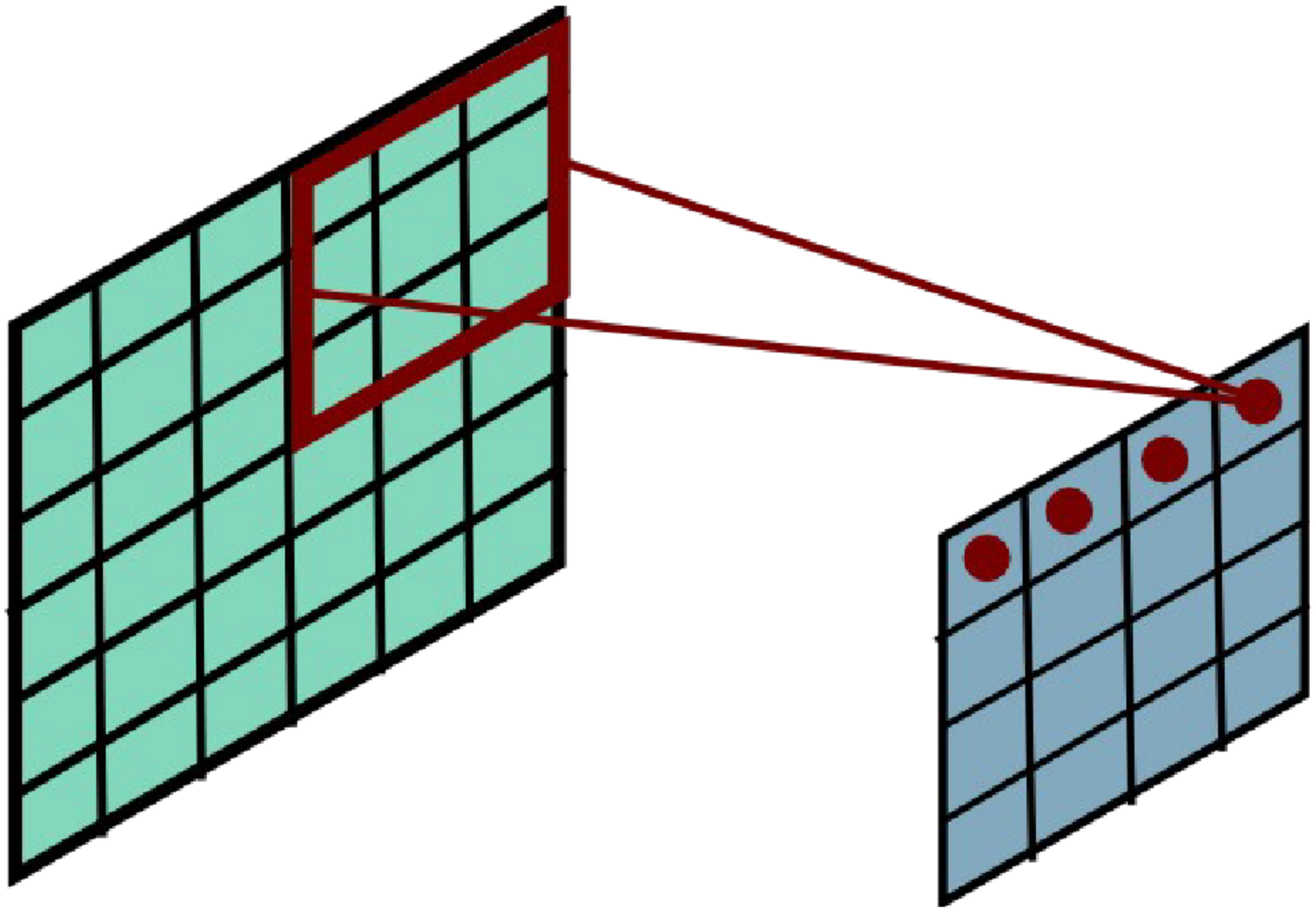
Convolutional Layer



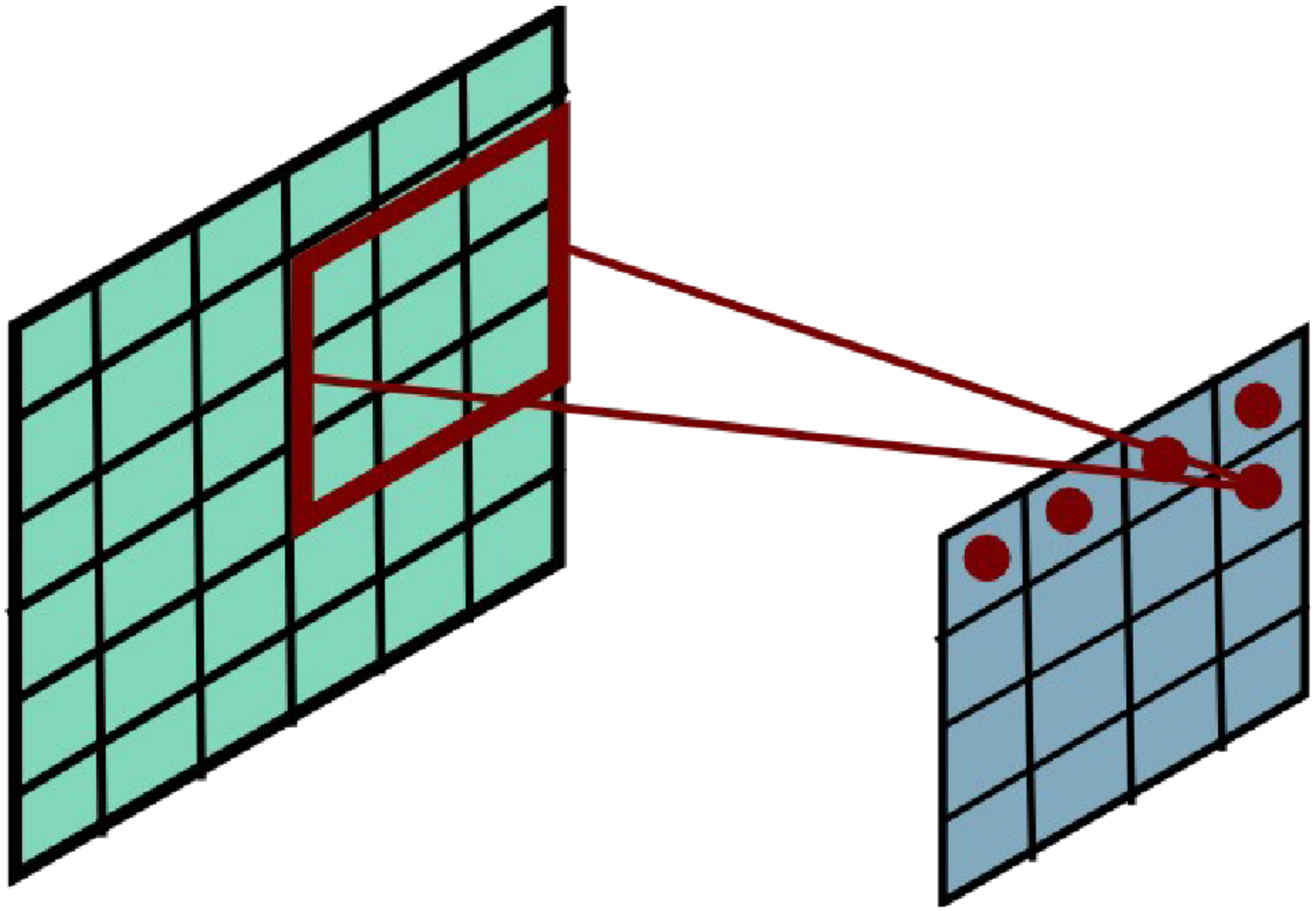
Convolutional Layer



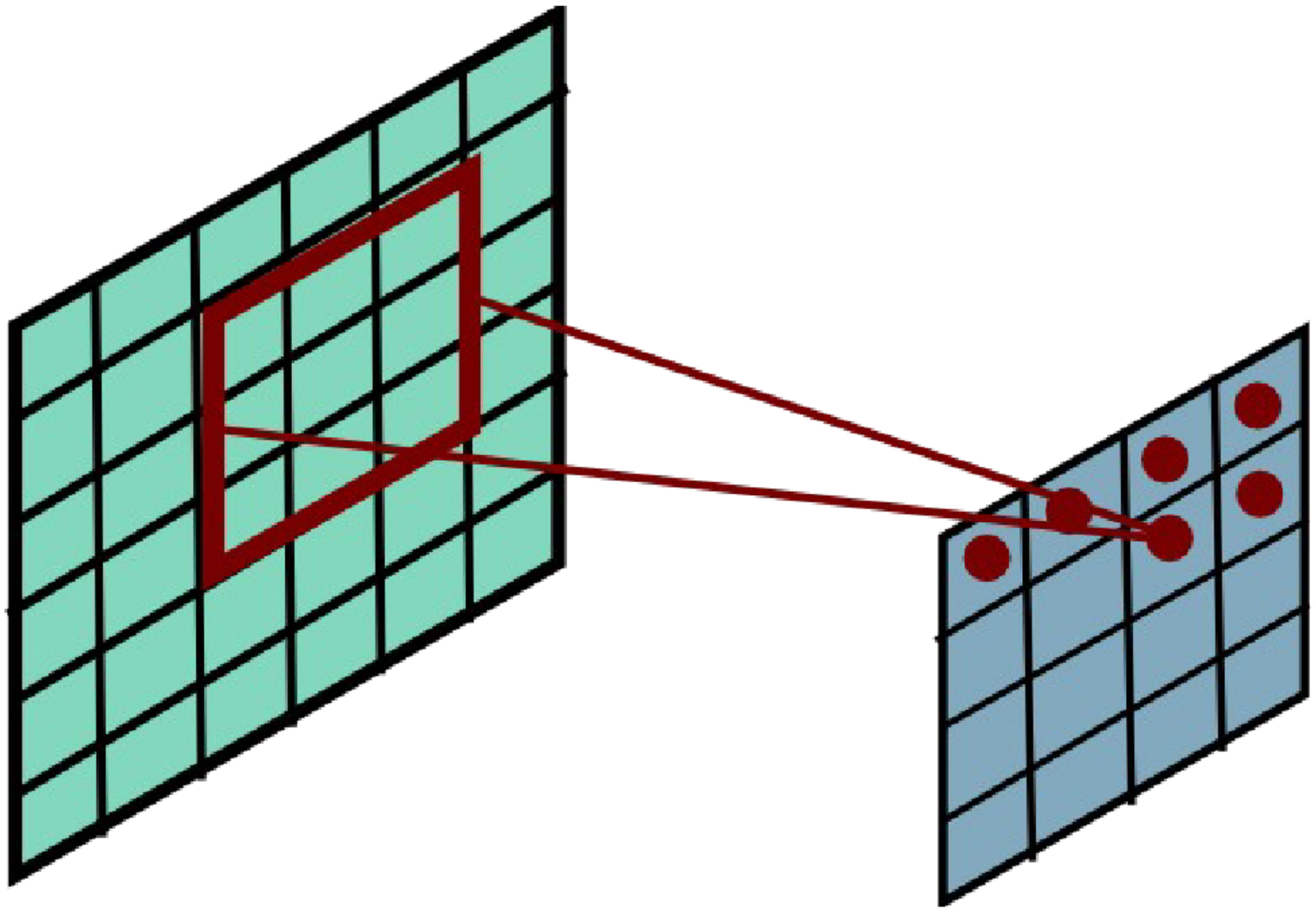
Convolutional Layer



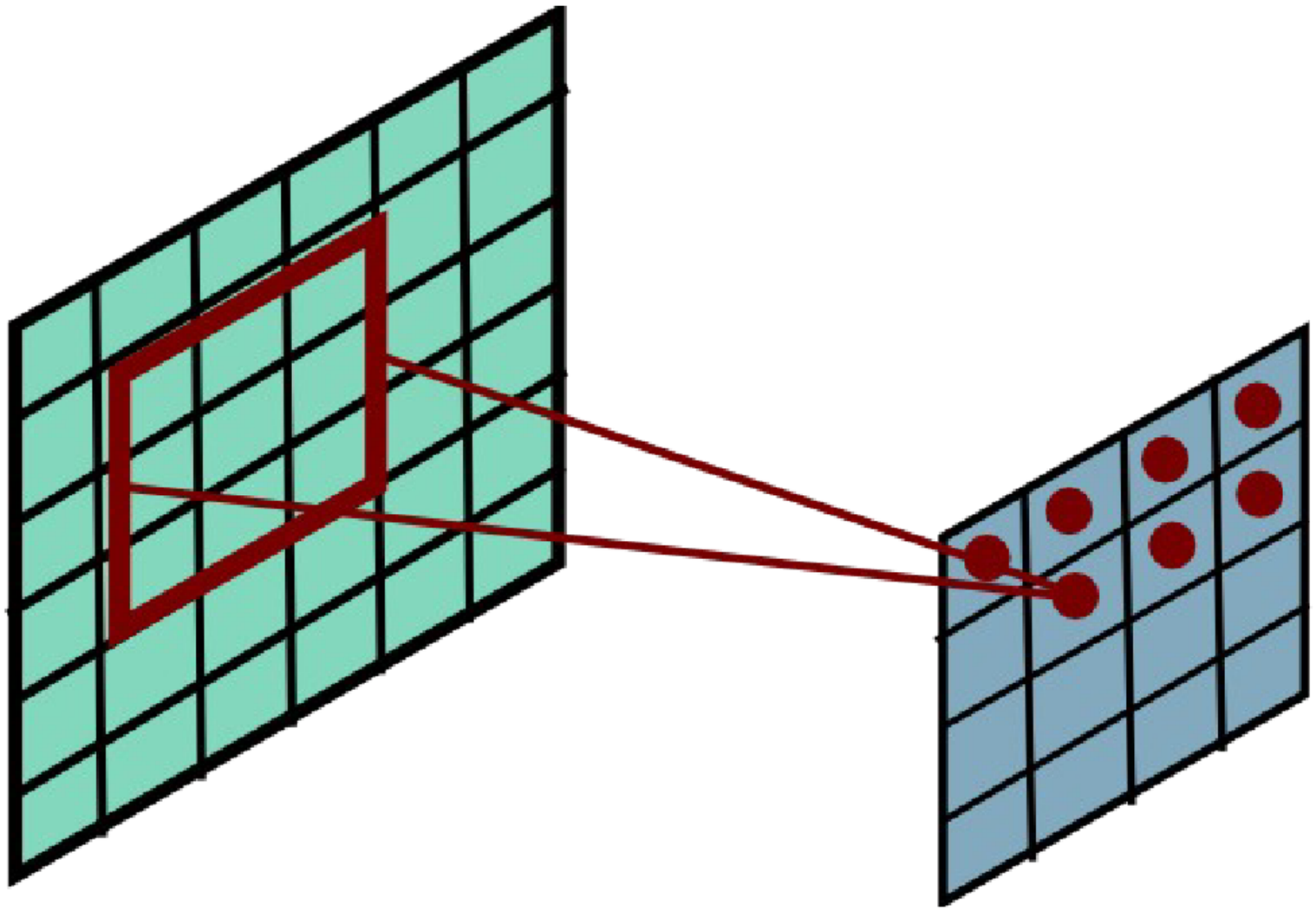
Convolutional Layer



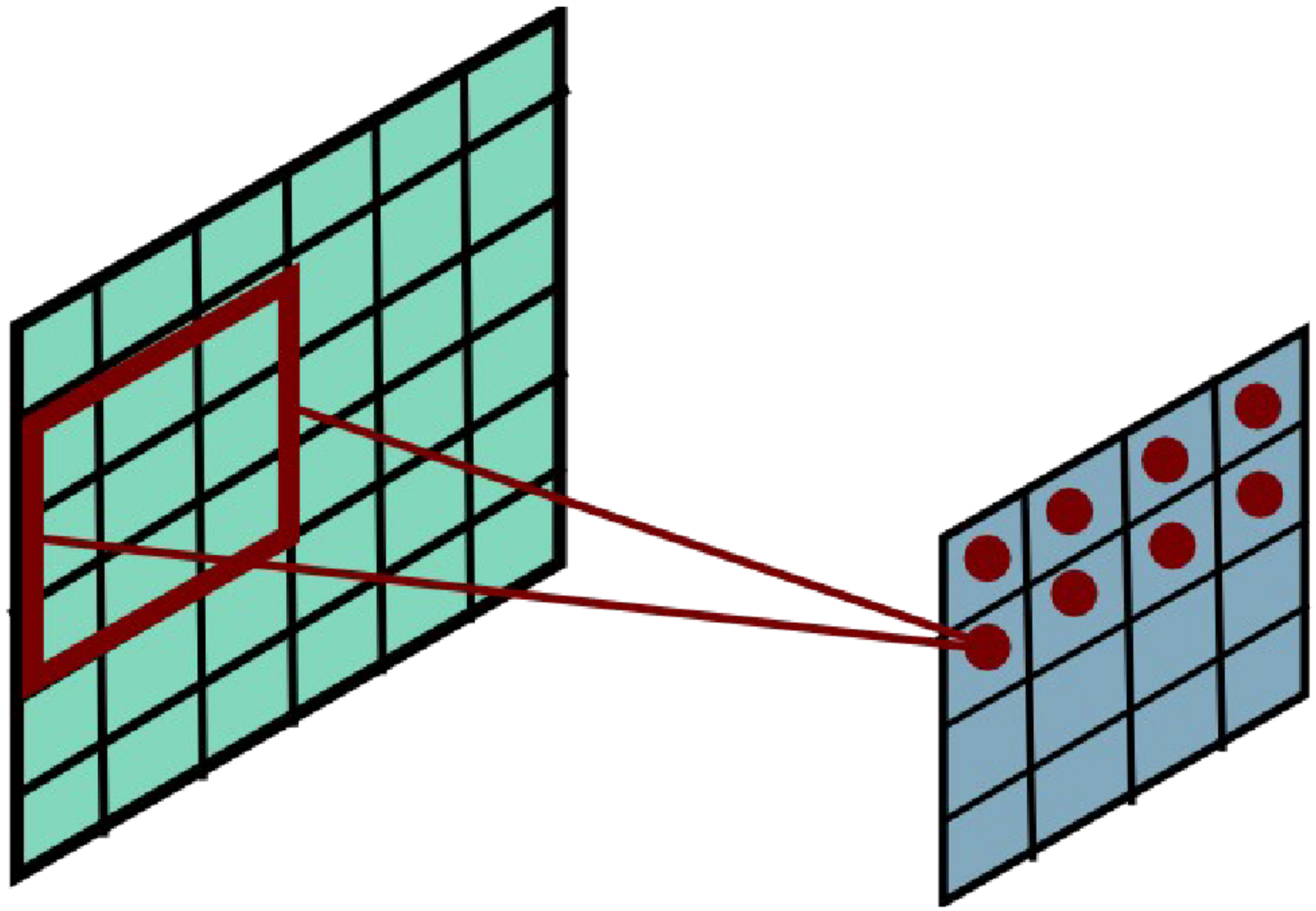
Convolutional Layer



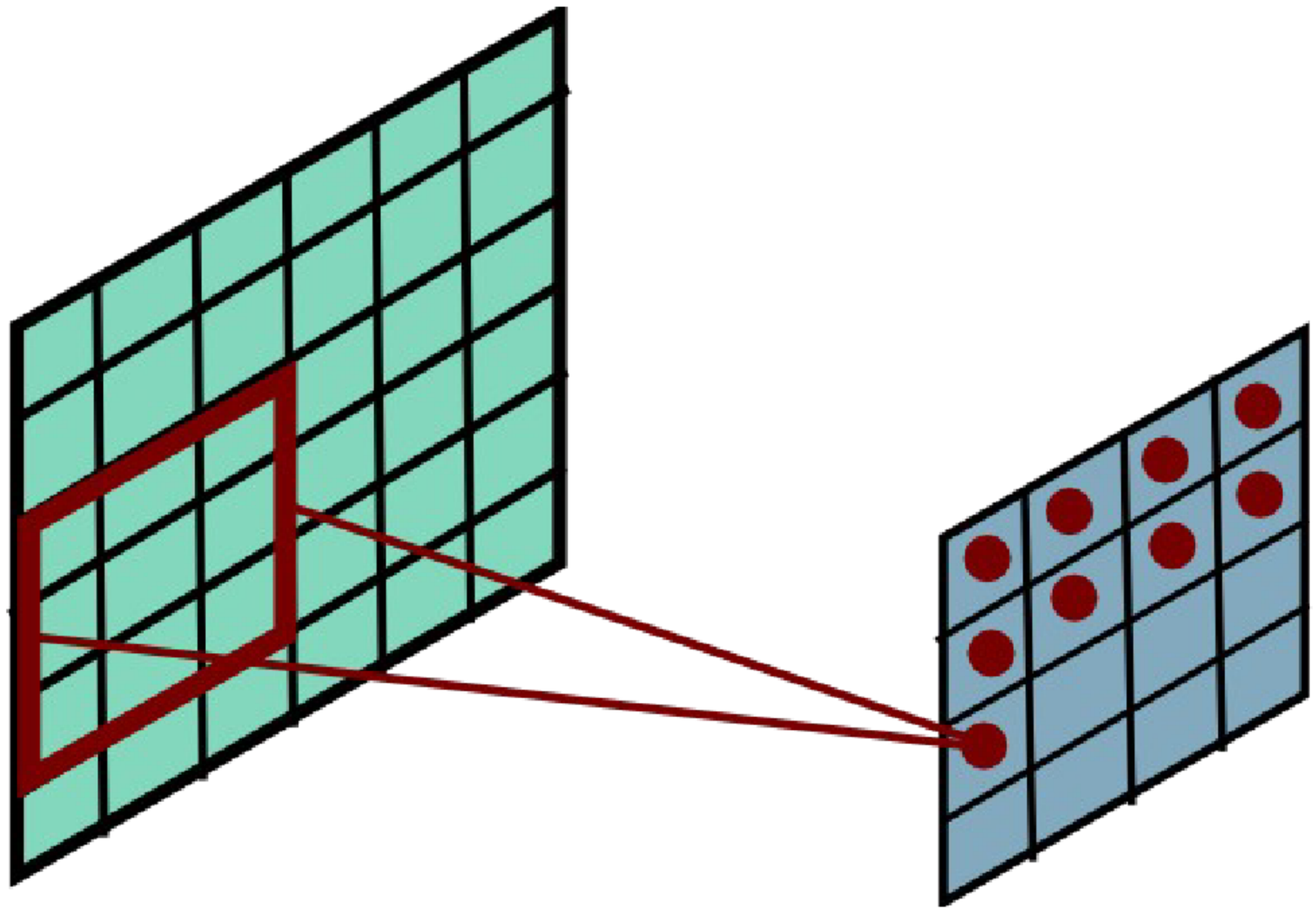
Convolutional Layer



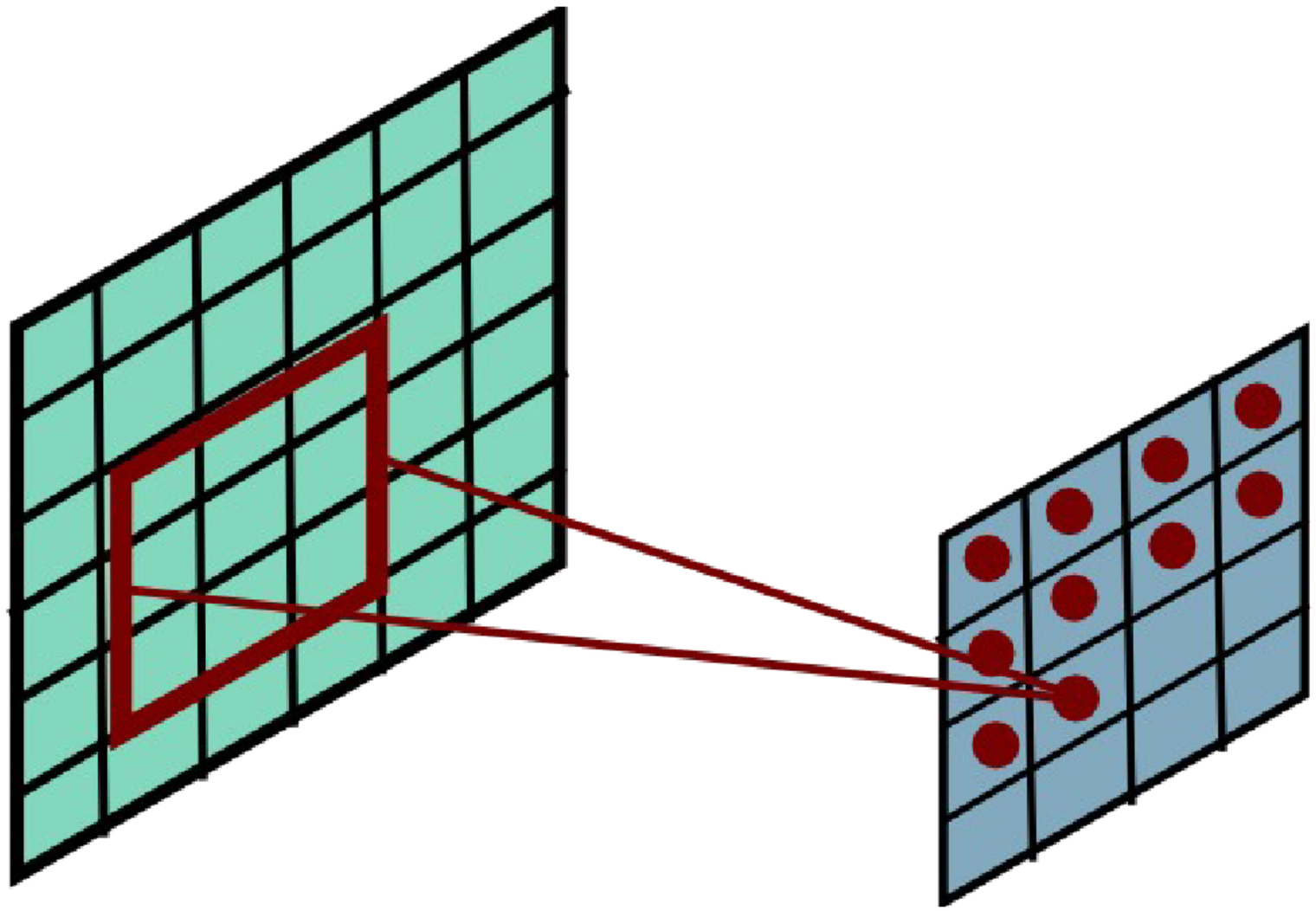
Convolutional Layer



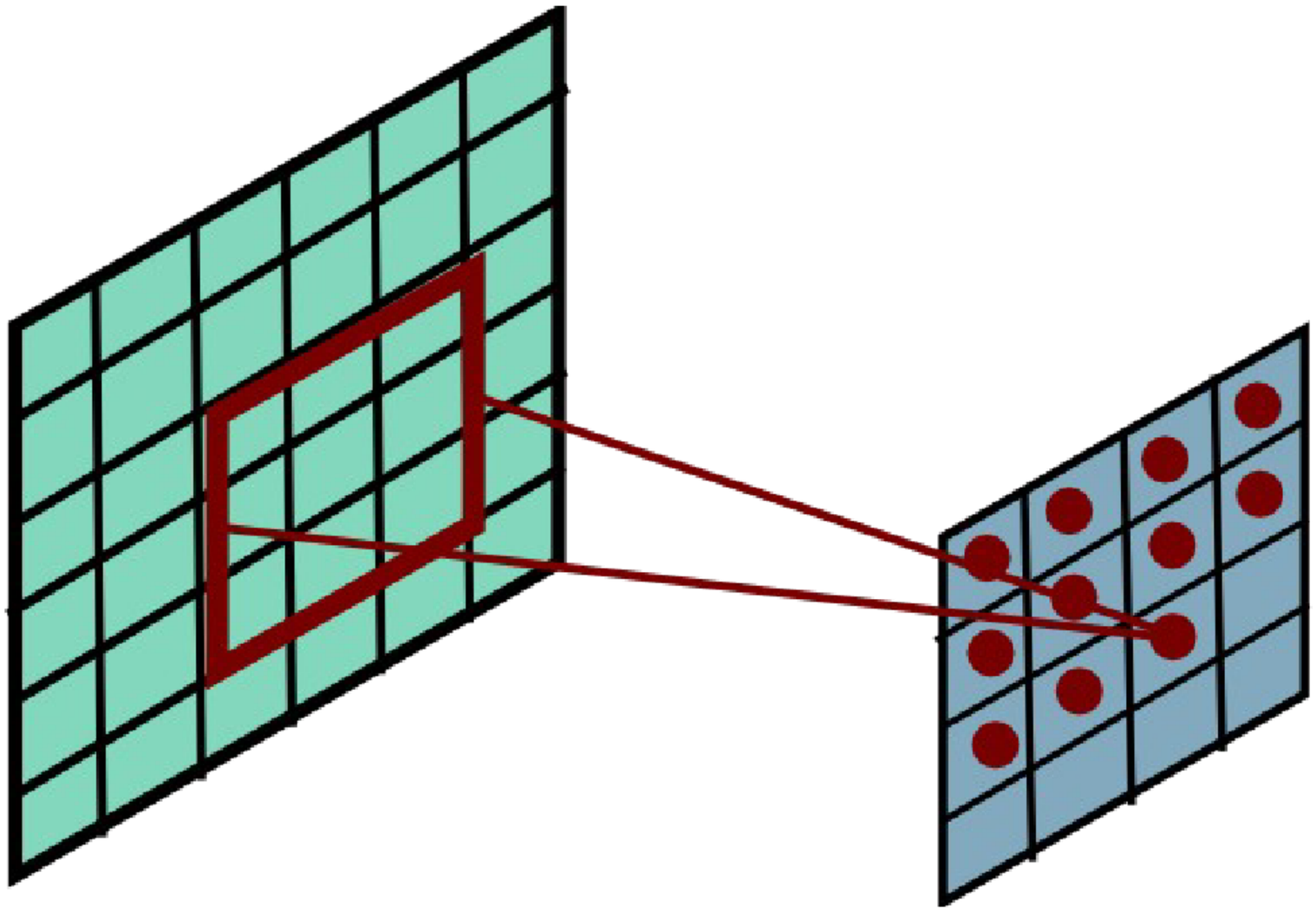
Convolutional Layer



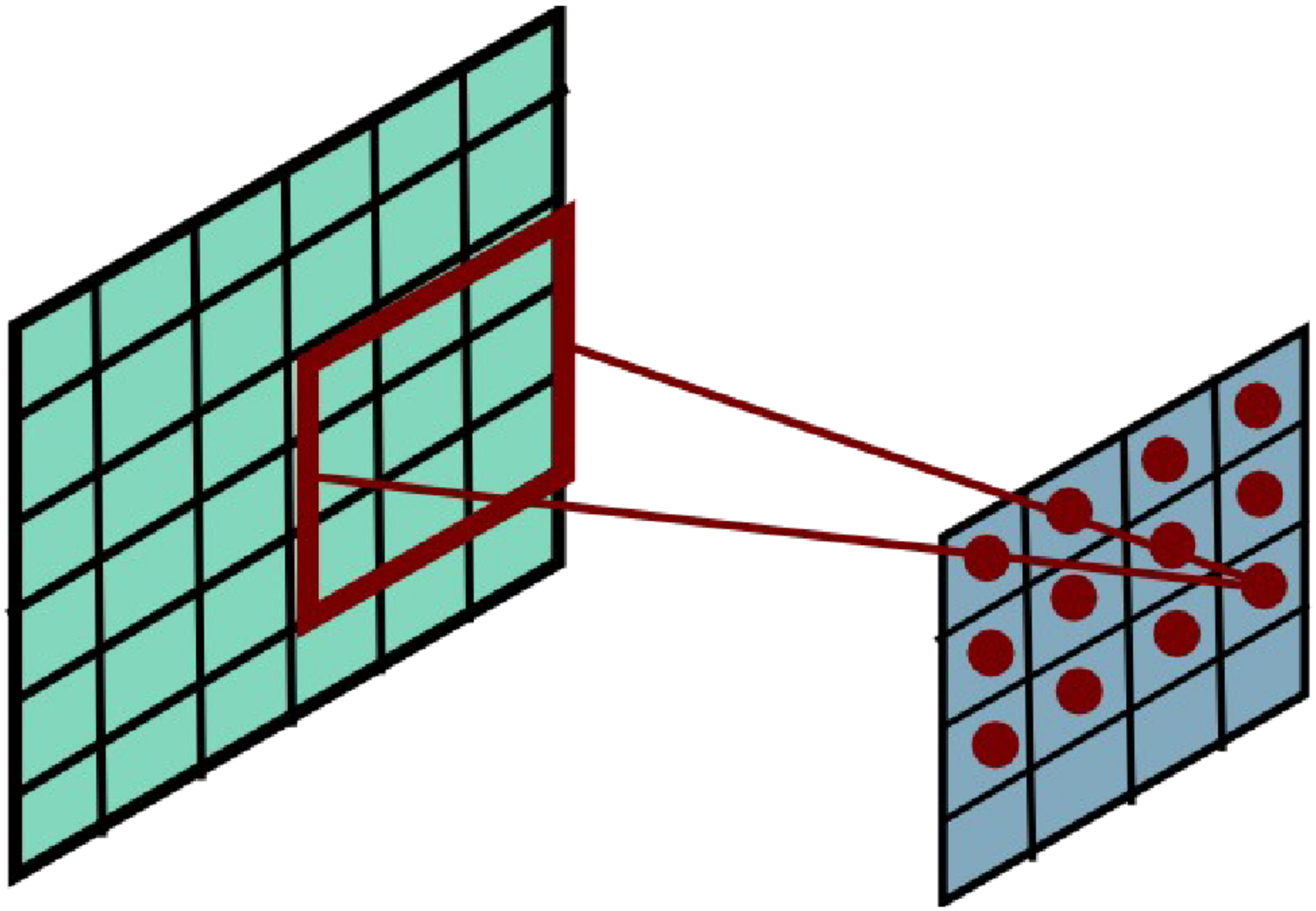
Convolutional Layer



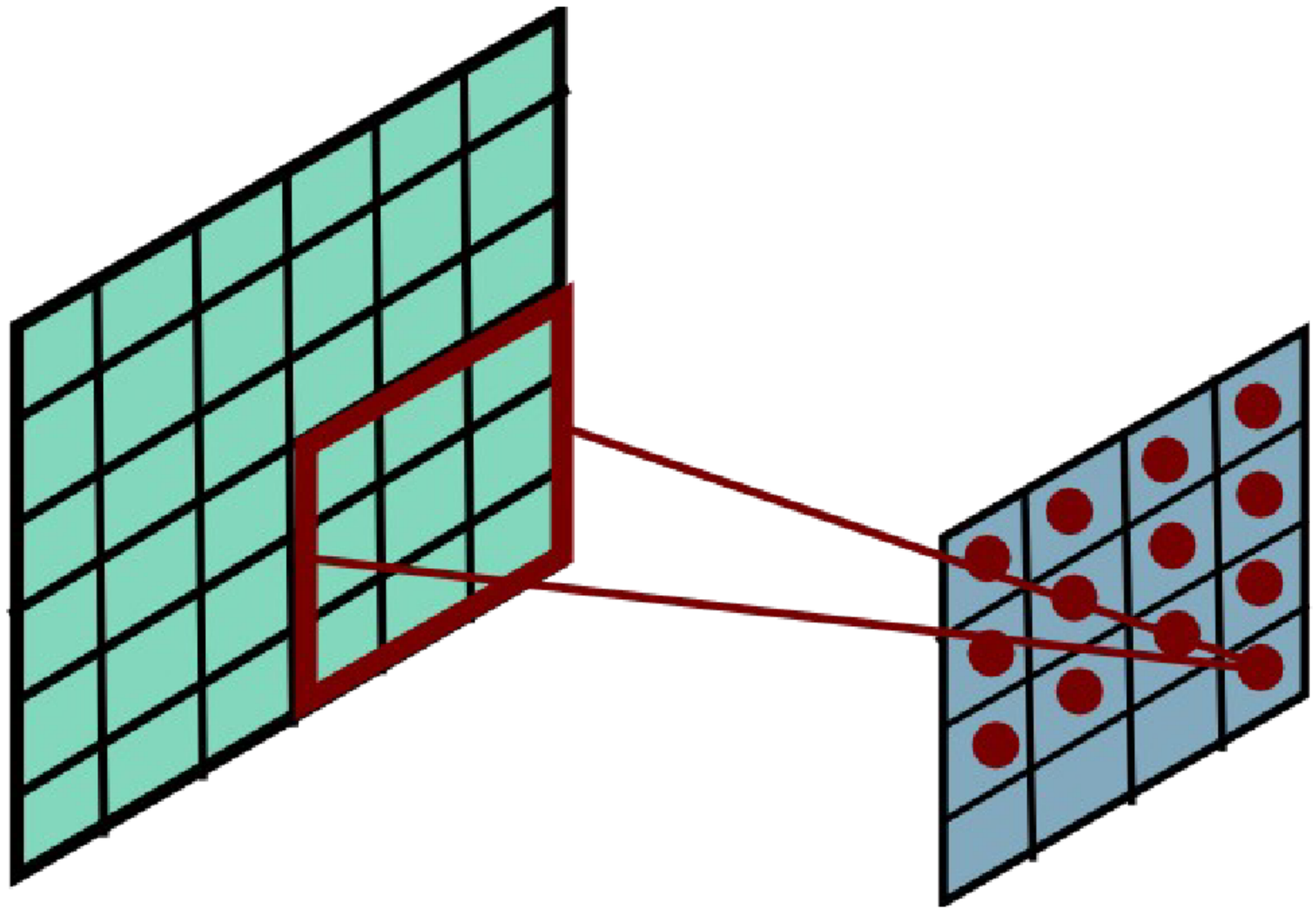
Convolutional Layer



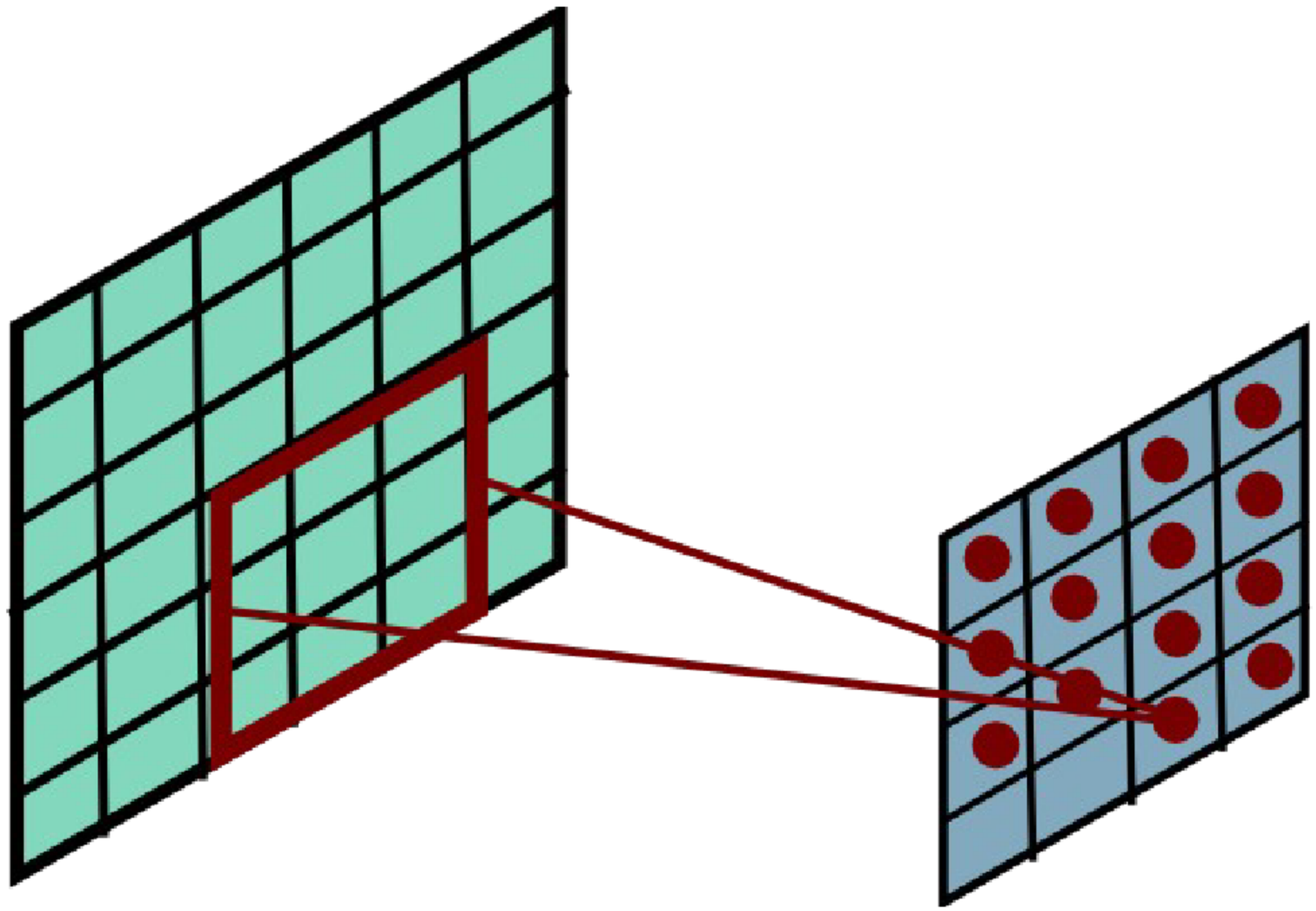
Convolutional Layer



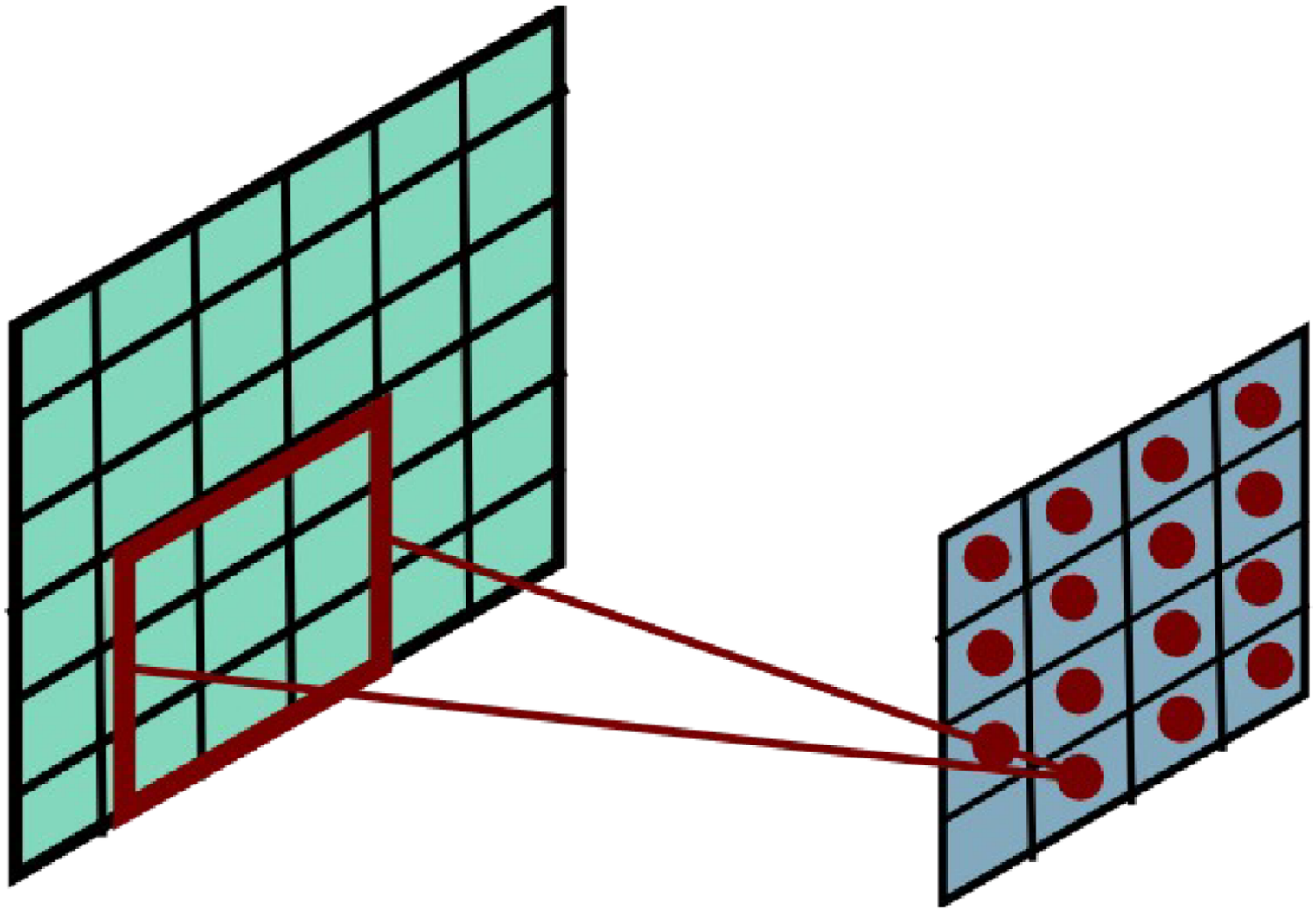
Convolutional Layer



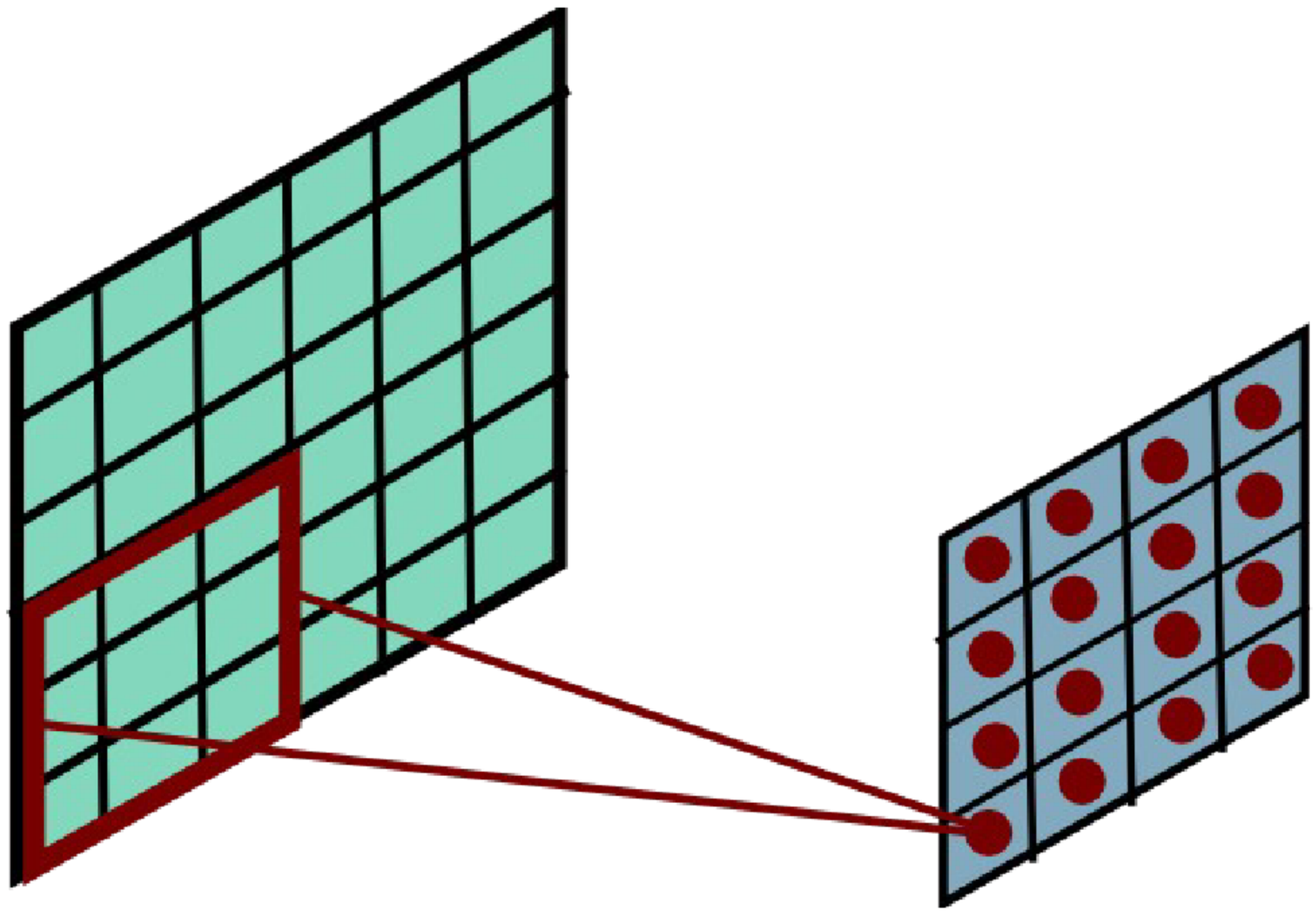
Convolutional Layer



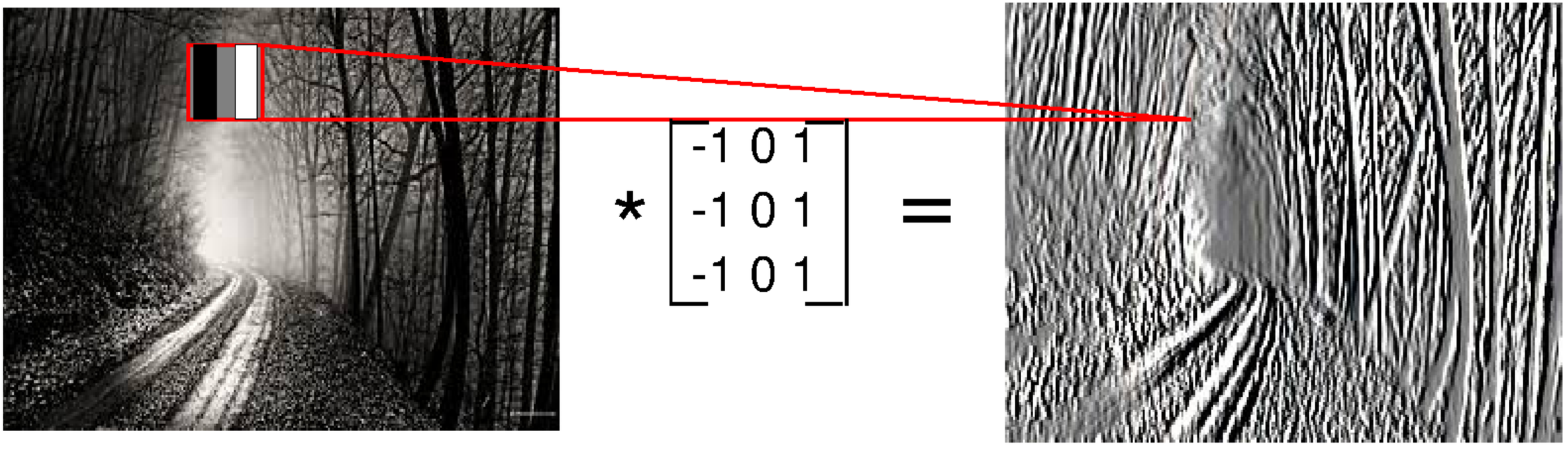
Convolutional Layer



Convolutional Layer



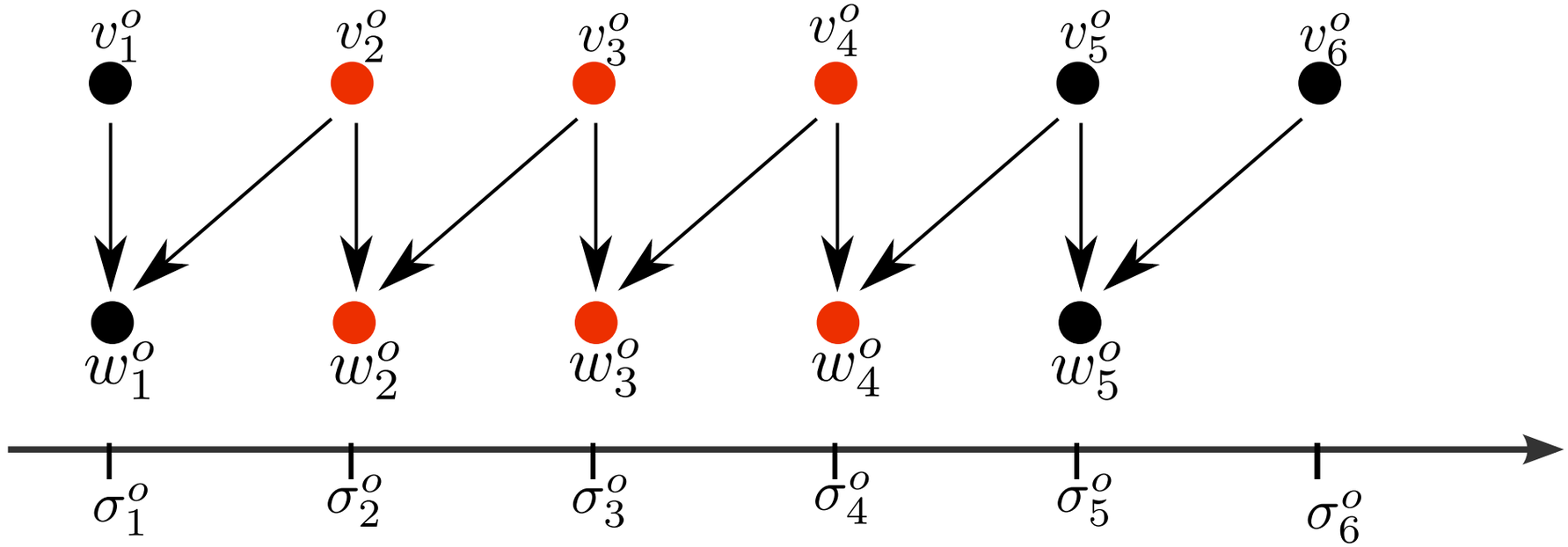
Convolutional Layer



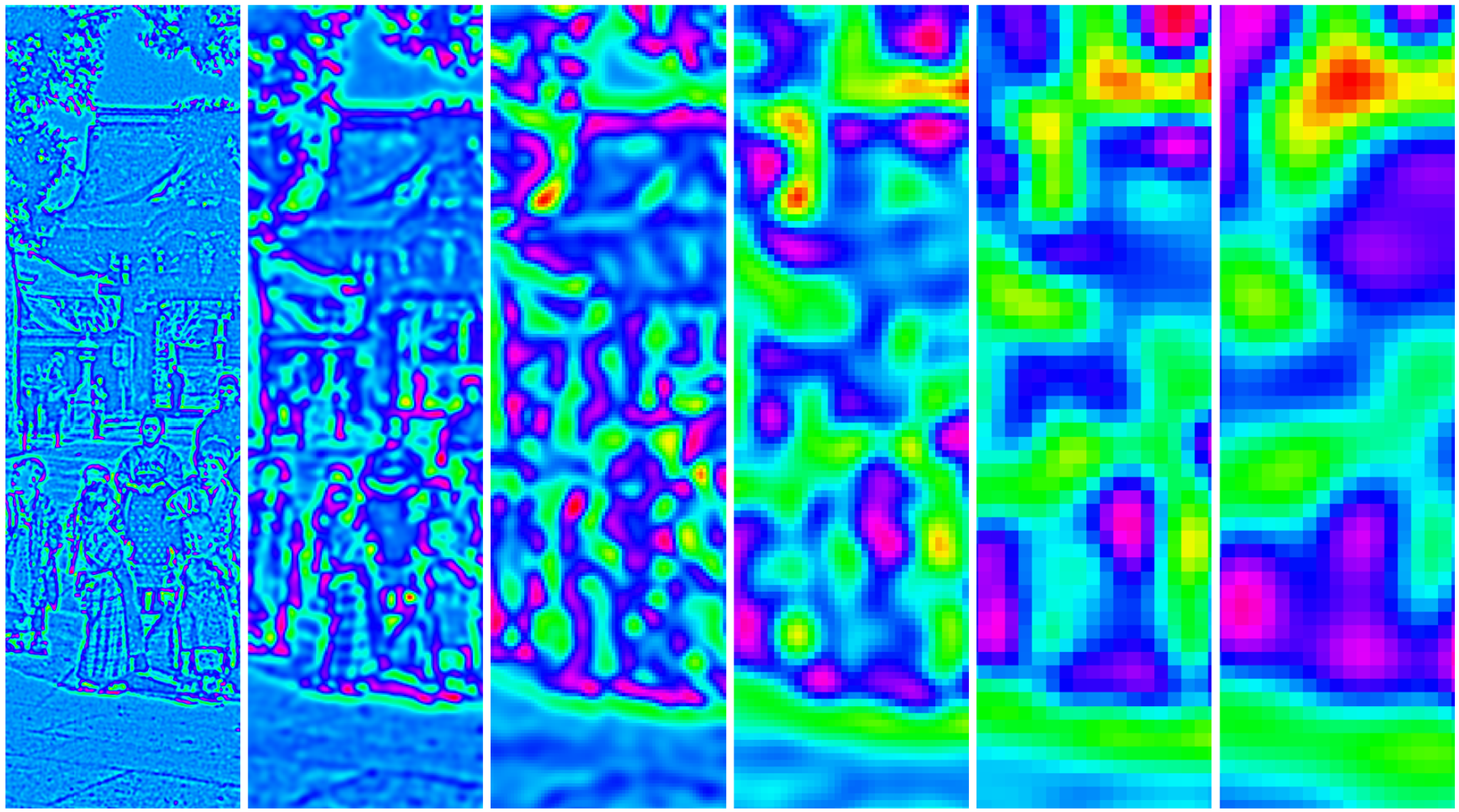
Scale Space Generation

$\mathbf{v}_1^1$	$\mathbf{v}_2^2$	$\mathbf{v}_2^3$	$\mathbf{v}_2^4$	$\mathbf{v}_3^5$	$\mathbf{v}_5^5$
$\delta_1 = 0.5$	$\delta_2 = 1.0$	$\delta_3 = 2.0$	$\delta_4 = 4.0$	$\delta_5 = 8.0$	$\delta_5 = 8.0$
$\sigma_1^1 = 1.0$	$\sigma_2^2 = 2.5$	$\sigma_2^3 = 5.1$	$\sigma_2^4 = 10.2$	$\sigma_3^5 = 25.6$	$\sigma_5^5 = 40.6$

Scale Space DoG



Scale Space DoG



$\mathbf{w}_1^1$
$\delta_1 = 0.5$
$\sigma_1^1 = 1.0$

$\mathbf{w}_2^2$
$\delta_2 = 1.0$
$\sigma_2^2 = 2.5$

$\mathbf{w}_2^3$
$\delta_3 = 2.0$
$\sigma_2^3 = 5.1$

$\mathbf{w}_2^4$
$\delta_4 = 4.0$
$\sigma_2^4 = 10.2$

$\mathbf{w}_1^5$
$\delta_5 = 8.0$
$\sigma_1^5 = 16.1$

$\mathbf{w}_3^5$
$\delta_5 = 8.0$
$\sigma_3^5 = 25.6$

Scale Space Extrema Extraction

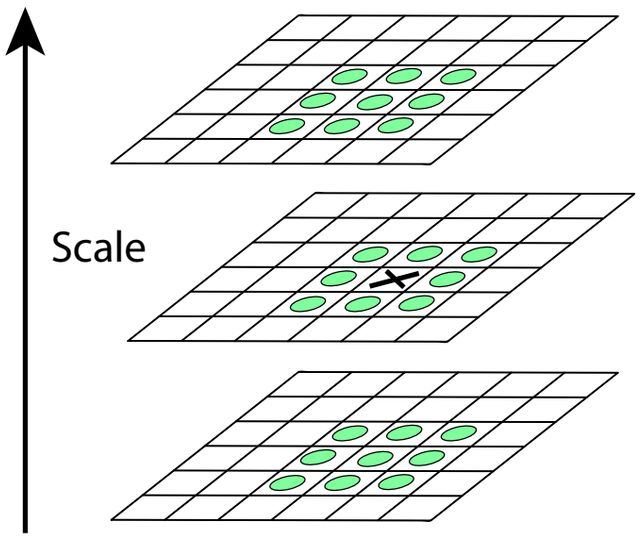
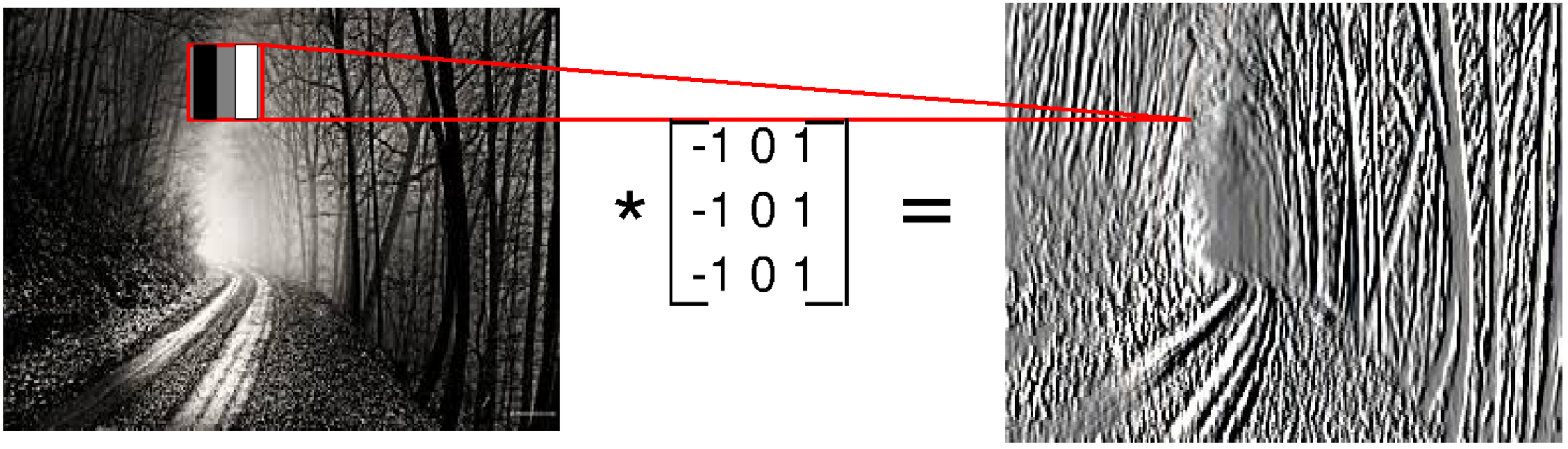


Figure 2: Maxima and minima of the difference-of-Gaussian images are detected by comparing a pixel (marked with X) to its 26 neighbors in 3x3 regions at the current and adjacent scales (marked with circles).

Convolutional Layer



Keypoint Description

- The Gaussian smoothed image L with the appropriate scale is chosen
- At this scale the gradient magnitude and orientation are computed

$$m(x,y) = \sqrt{(L(x+1,y) - L(x-1,y))^2 + (L(x,y+1) - L(x,y-1))^2}$$

$$\theta(x,y) = \tan^{-1}((L(x,y+1) - L(x,y-1))/(L(x+1,y) - L(x-1,y)))$$

Keypoint description

- The key point description is obtained by creating orientation histograms over 4x4 sample regions

- Each orientation histogram has 8 bins

- Each sample is weighted by a circularly symmetric Gaussian around the centre so that gradients far from the centre have less effect

- The resultant 4x4x8 length feature vector is obtained as the description of the key point

Keypoint description

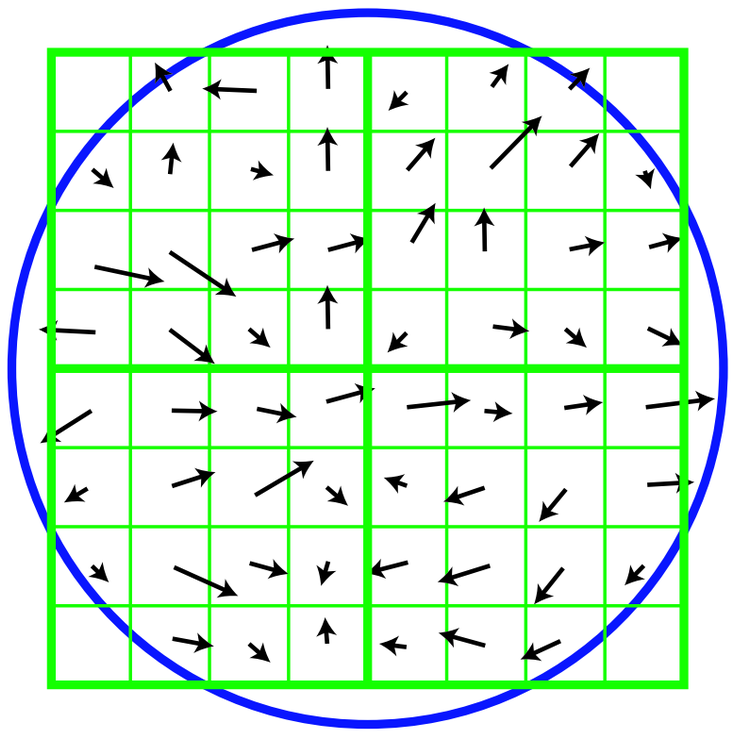
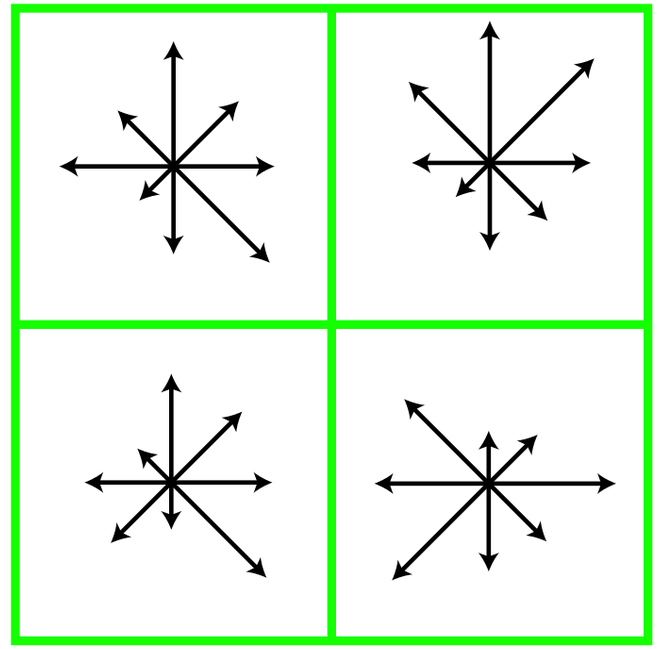


Image gradients

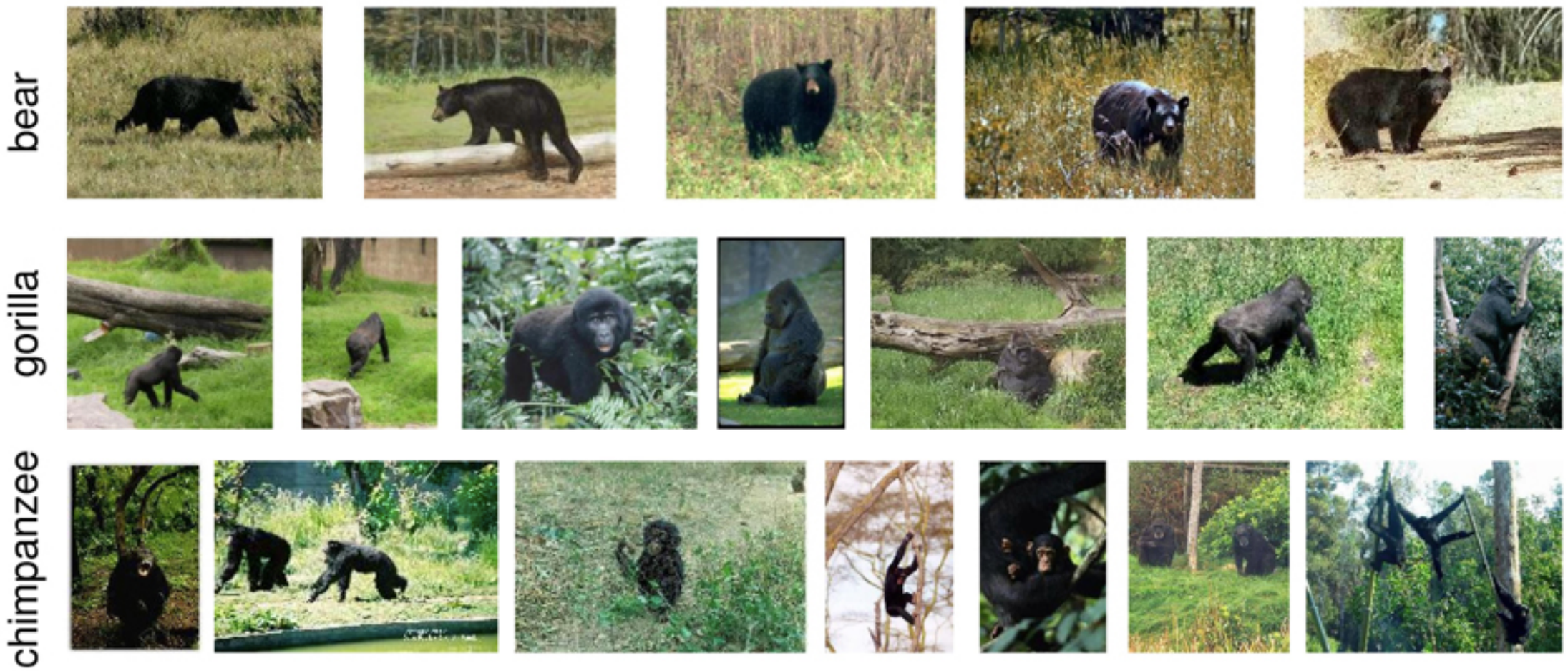


Keypoint descriptor

Object Categorisation

CS 783: Visual Recognition

Object Categorisation



http://www.robots.ox.ac.uk/~vgg/research/caltech/C256_ambiguities.html

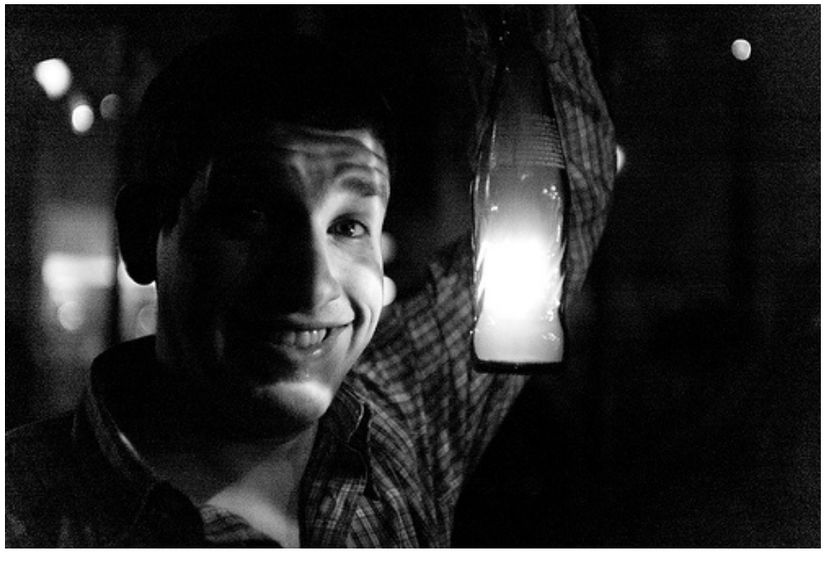
Object Categorisation

Object Categorisation

canoe

kayac

Challenges



Illumination

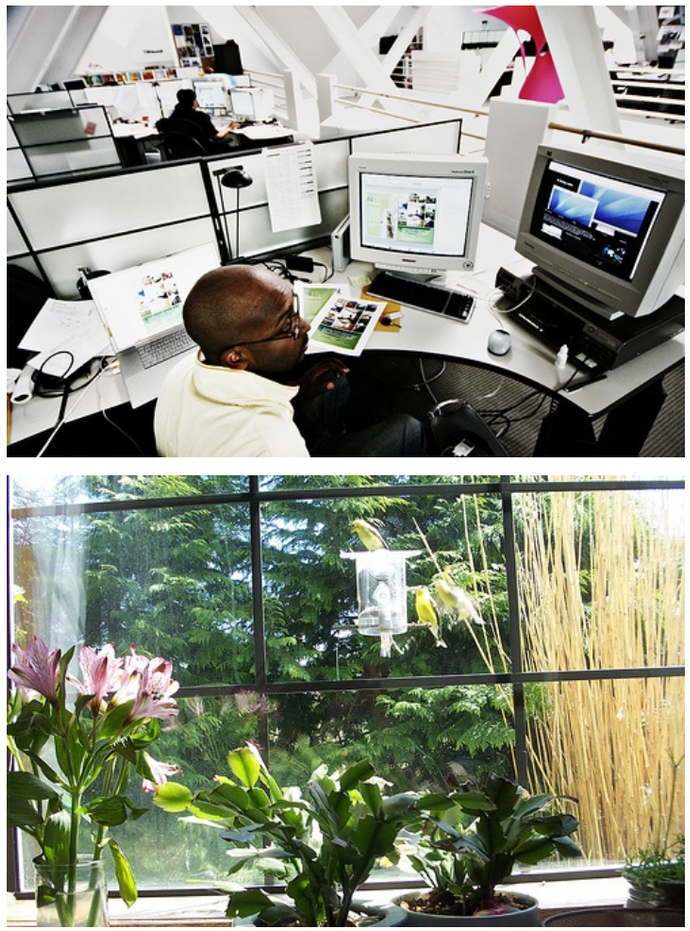
image credit: Hakan Bilen

Challenges

Viewpoint

image credit: Hakan Bilen

Challenges



Background clutter

image credit: Hakan Bilen

Challenges

Occlusion

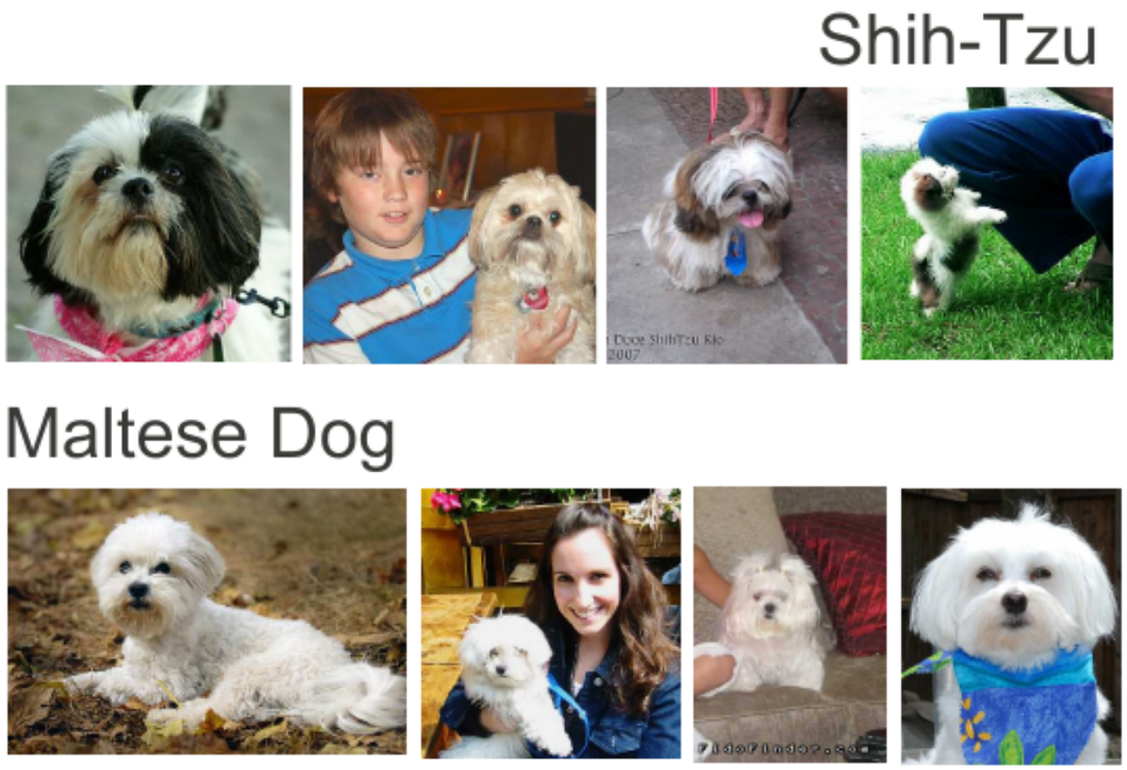
image credit: Hakan Bilen

Challenges

Intra-class variation

image credit: Hakan Bilen

Challenges



Intra-class vs inter-class variability

image credit: Hakan Bilen

Problem

- Given: Set of positive training images that contain images from a particular object class

- And a set of negative images that do not contain the particular object class

- Predict given a test image whether the image contains the class or not

Approach

Approach

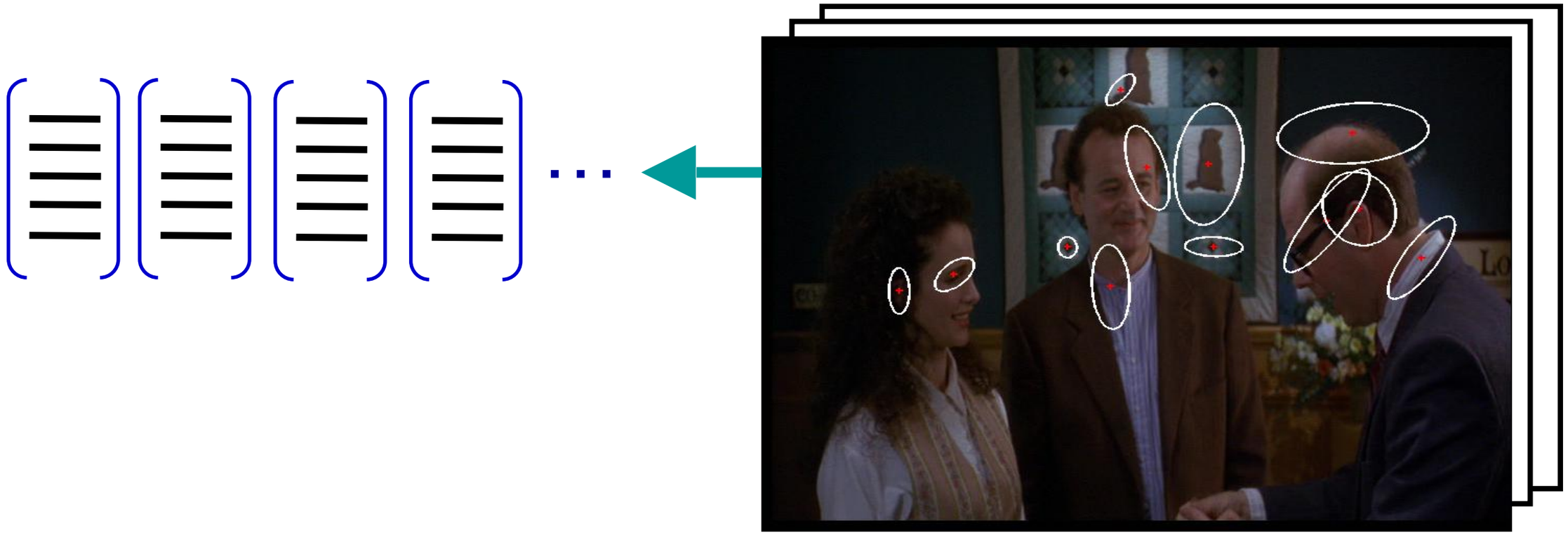
- Obtain a representation of an image
- Learn a classifier based on the representation for a set of training data with ground truth label
- Use classifier to infer class label at test time

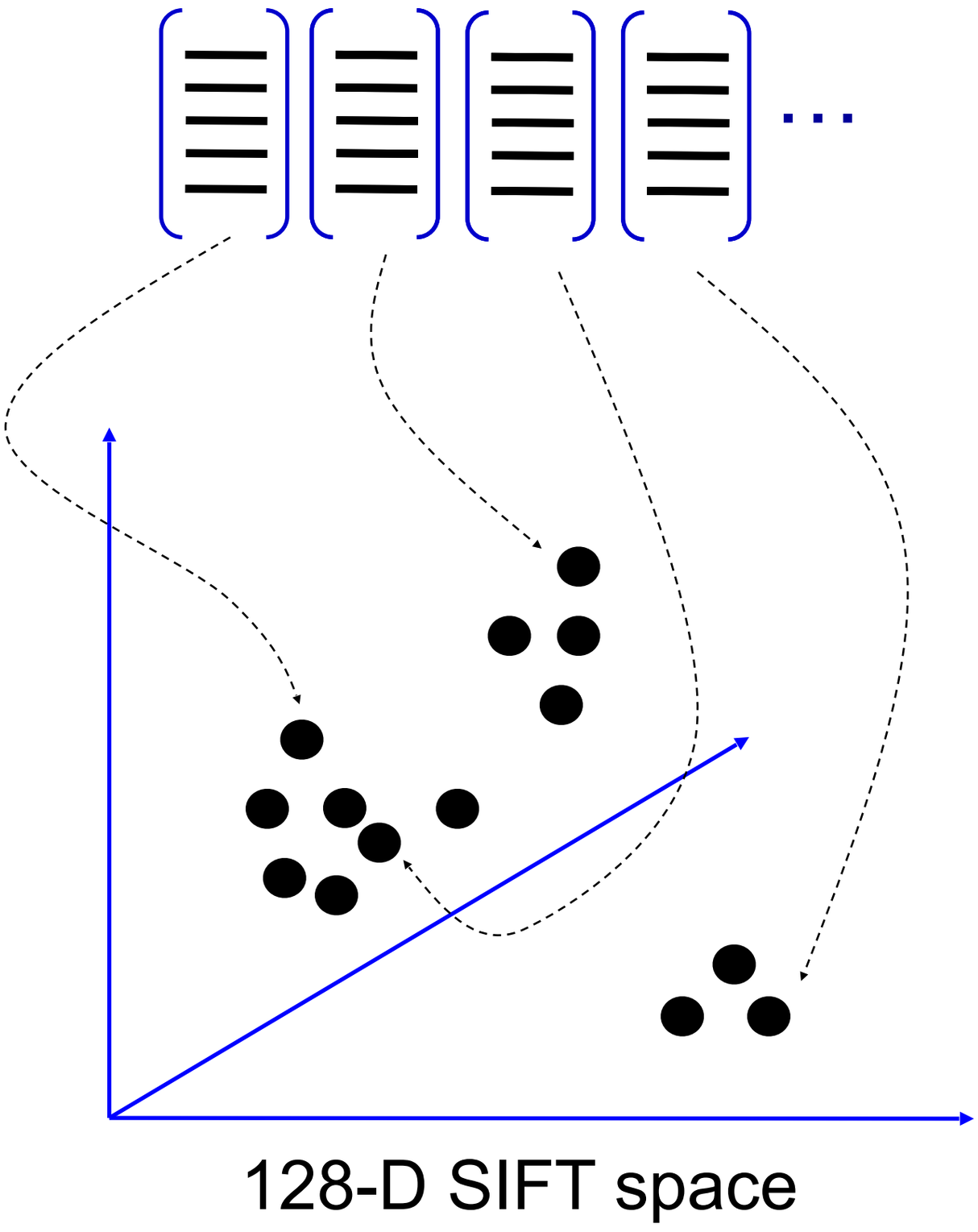
Approach

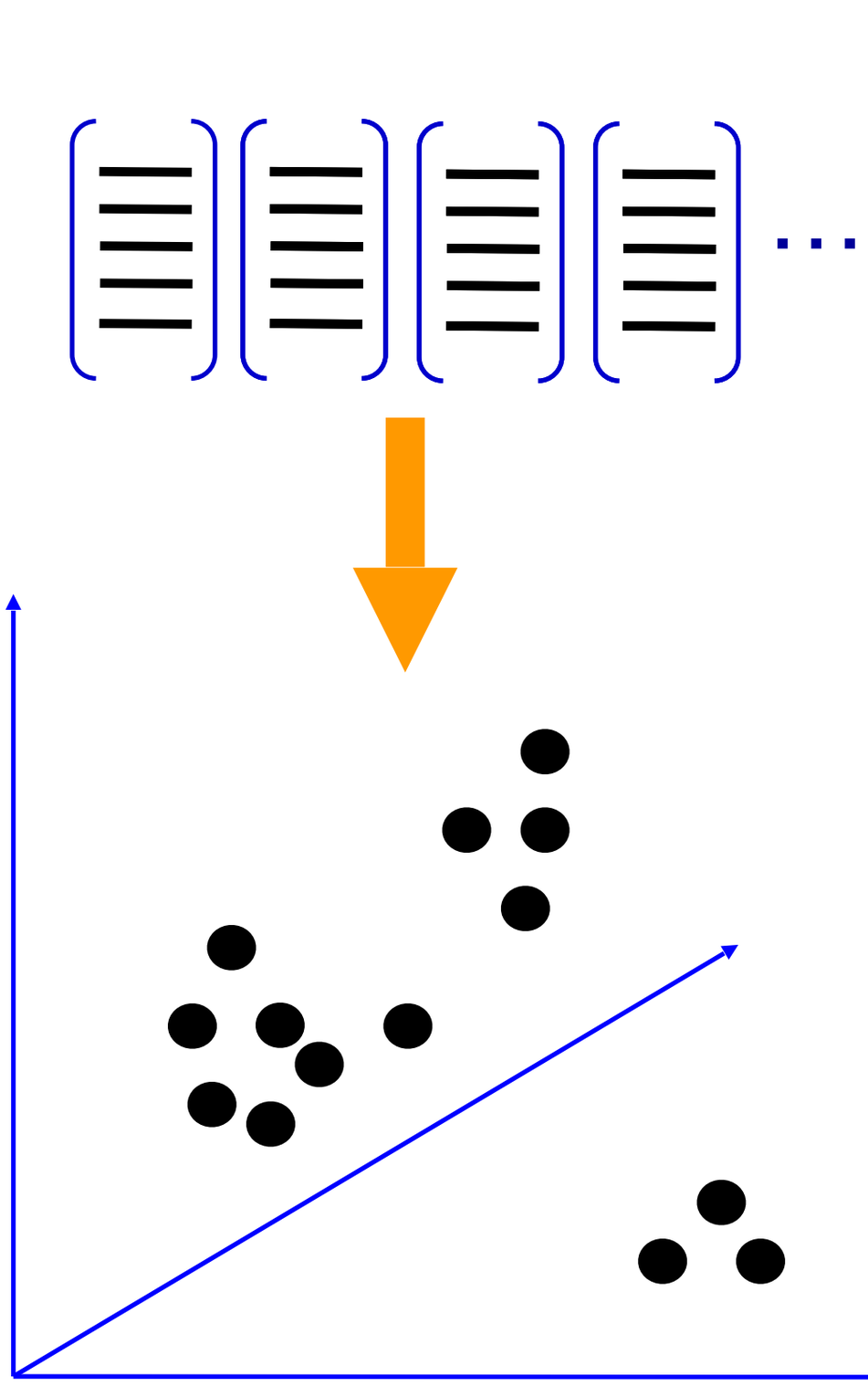
- Represent each image as a bag of visual words to obtain a feature
- Train a classifier to classify the images
- Use the trained model to predict the test image

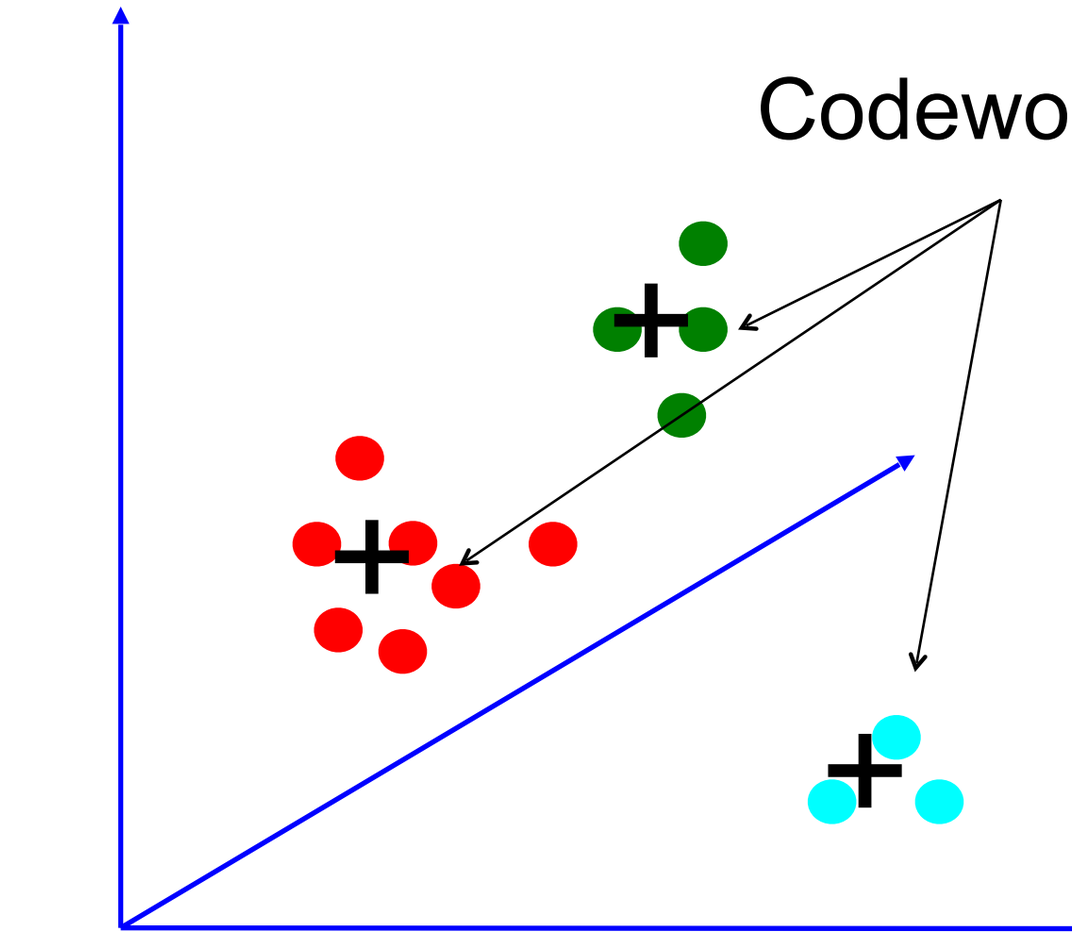
Object → Bag of ‘words’

1.Feature detection and representation




128-D SIFT space

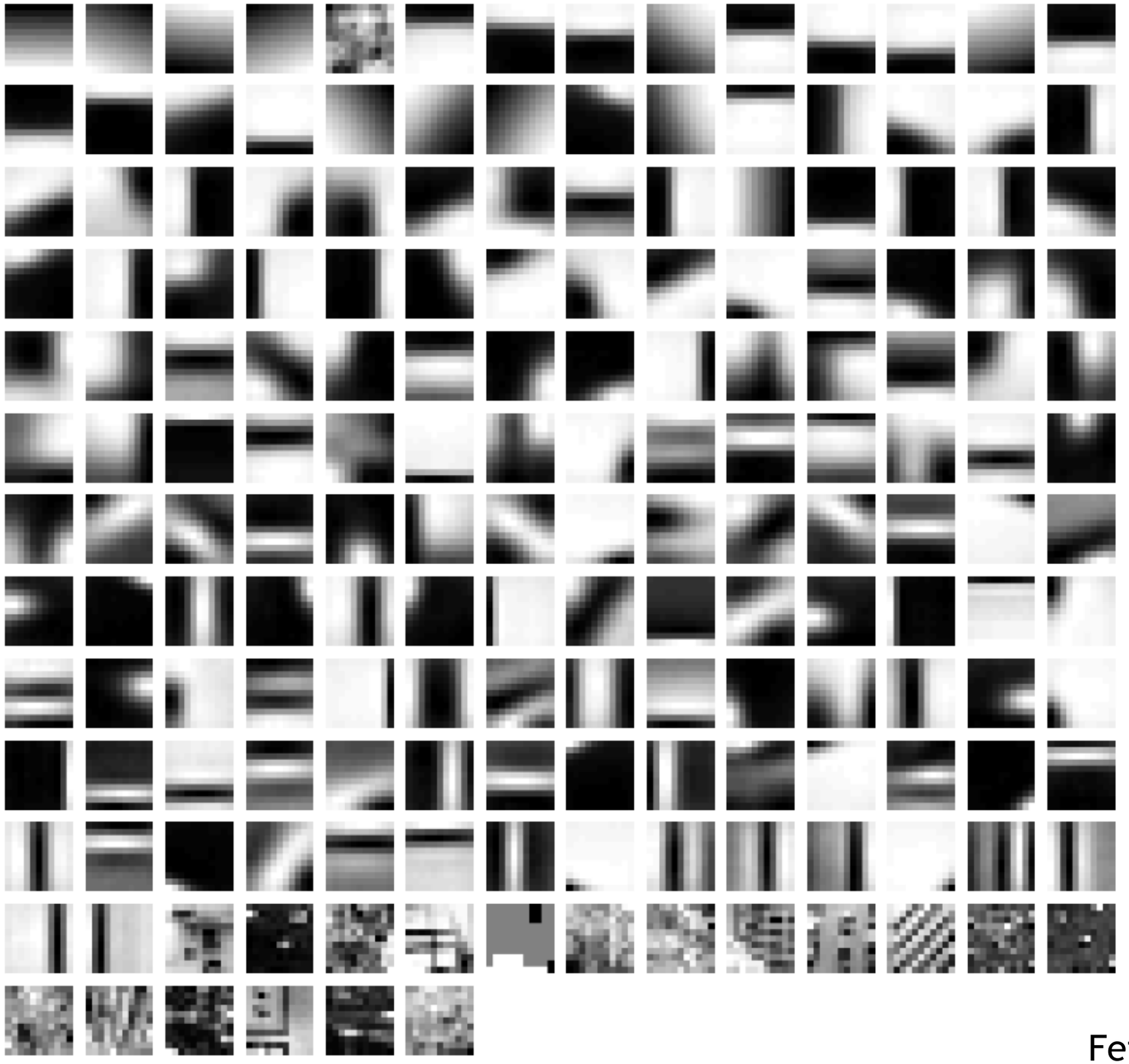




Vector quantization

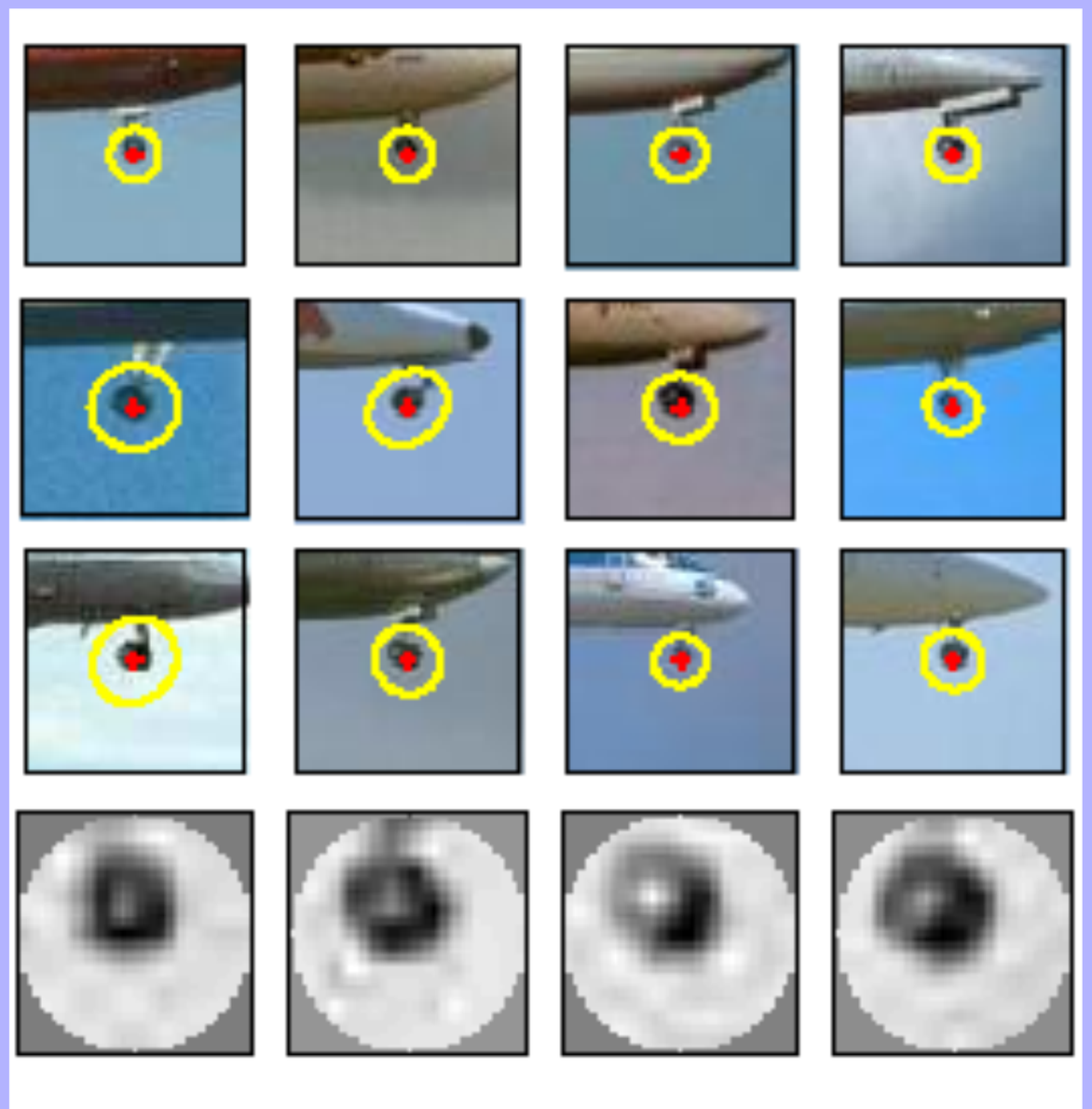
Slide credit: Josef Sivic

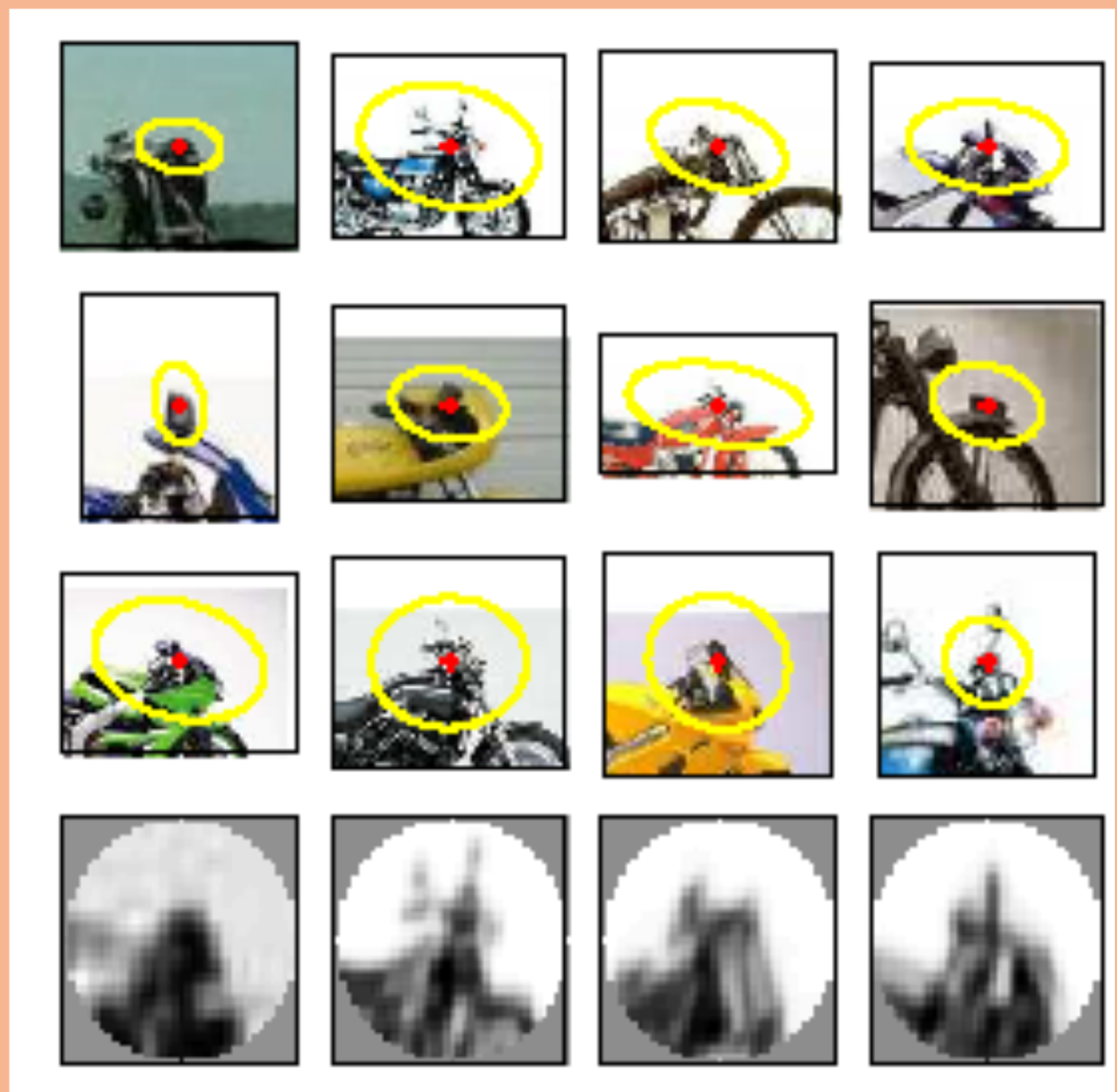
Codewords dictionary formation



Fei-Fei et al. 2005

Image patch examples of codewords

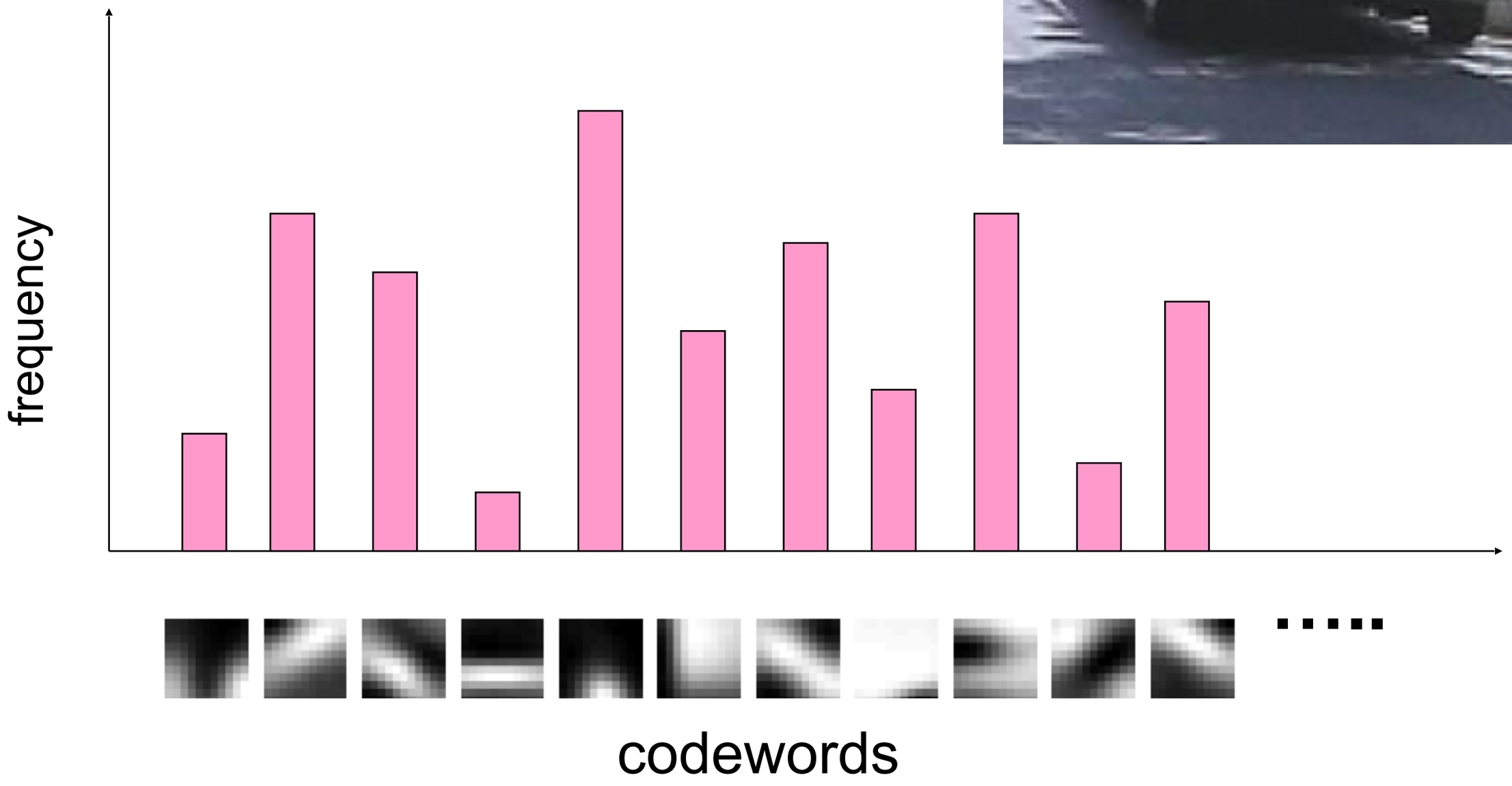




Sivic et al. 2005

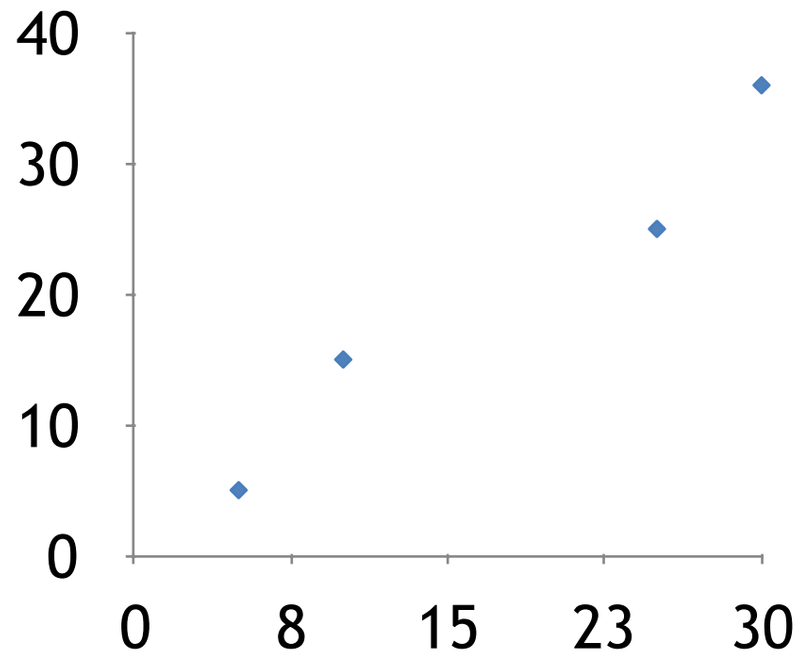
Image representation

Histogram of features
assigned to each cluster

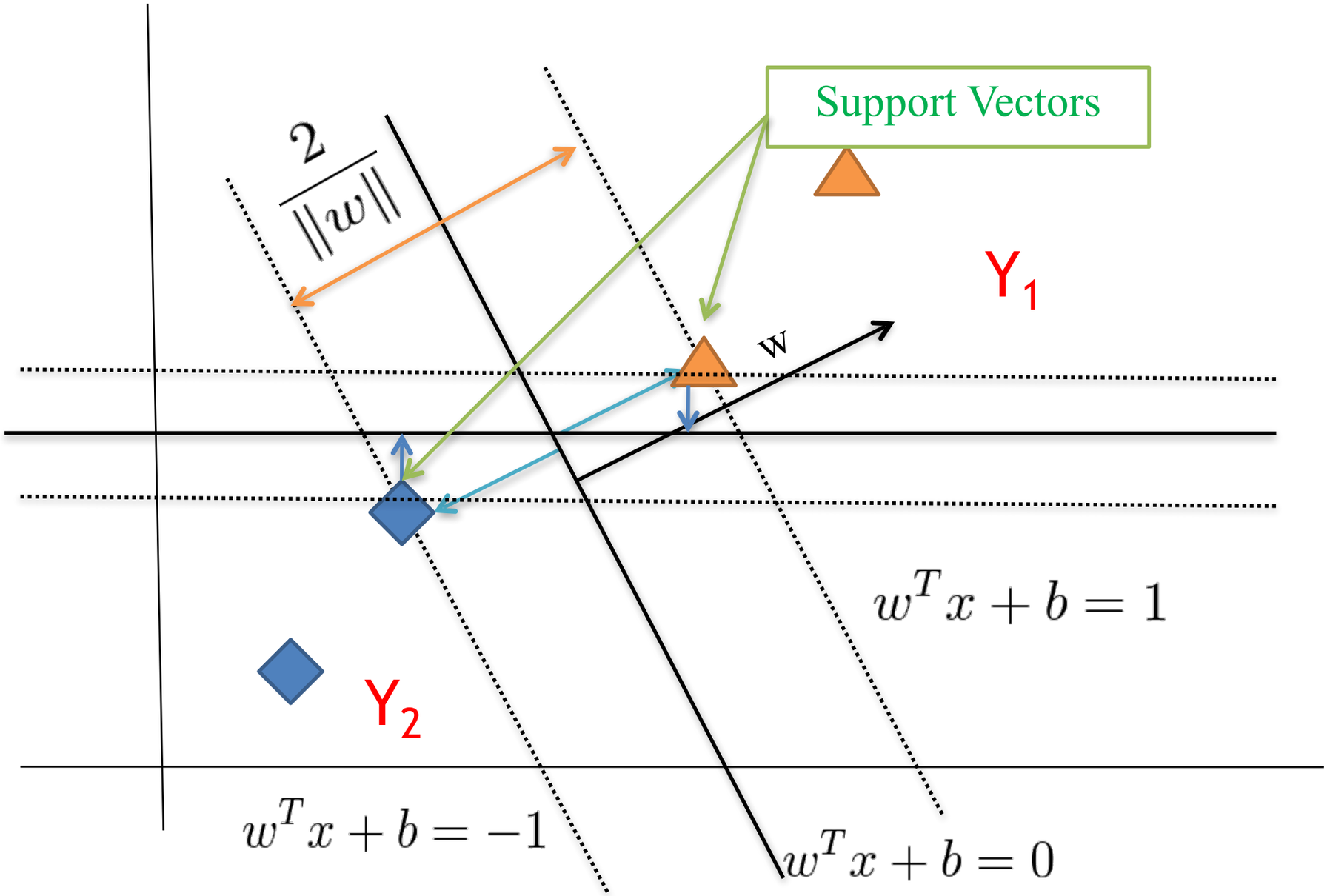


Basics
Machine Learning

Blood Pressure	Sugar	Hyper Tension
10	15	No
5	5	No
25	25	Yes
30	36	Yes



Basics
Machine Learning



Objective Function of SVM

$$\max_{w} \frac{2}{\|w\|}$$

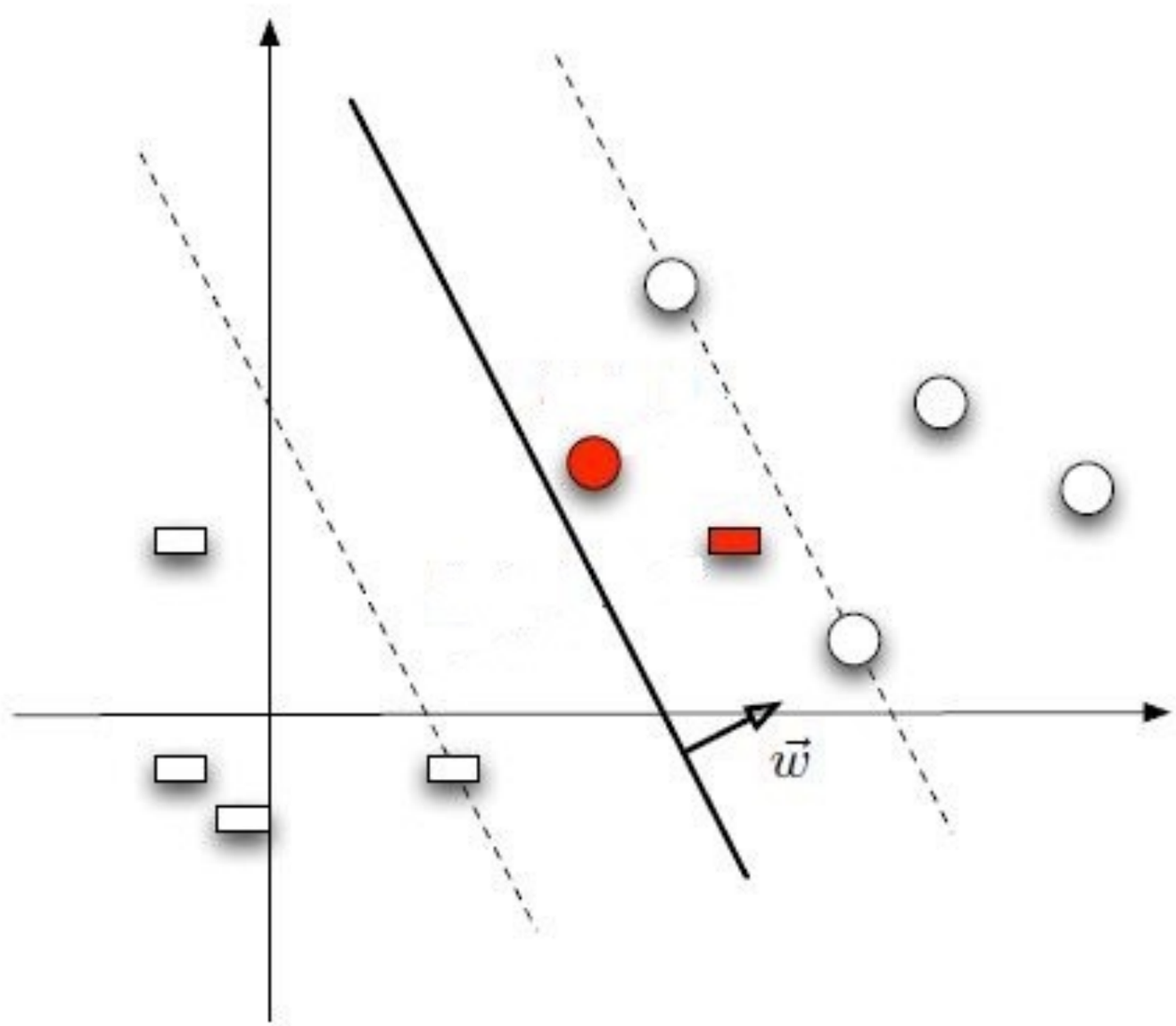
s.t.
$$if \;\; y_i = Y_1, \quad w^T x_i + b \geq 1$$
$$if \;\; y_i = Y_2, \quad w^T x_i + b \leq -1$$

Objective Function of SVM

$$\min_{w} \frac{\|w\|^2}{2}$$

s.t. $$if \;\; y_i = Y_1, \quad w^T x_i + b \geq 1$$
$$if \;\; y_i = Y_2, \quad w^T x_i + b \leq -1$$

Soft-Margin SVM



Soft-Margin SVM

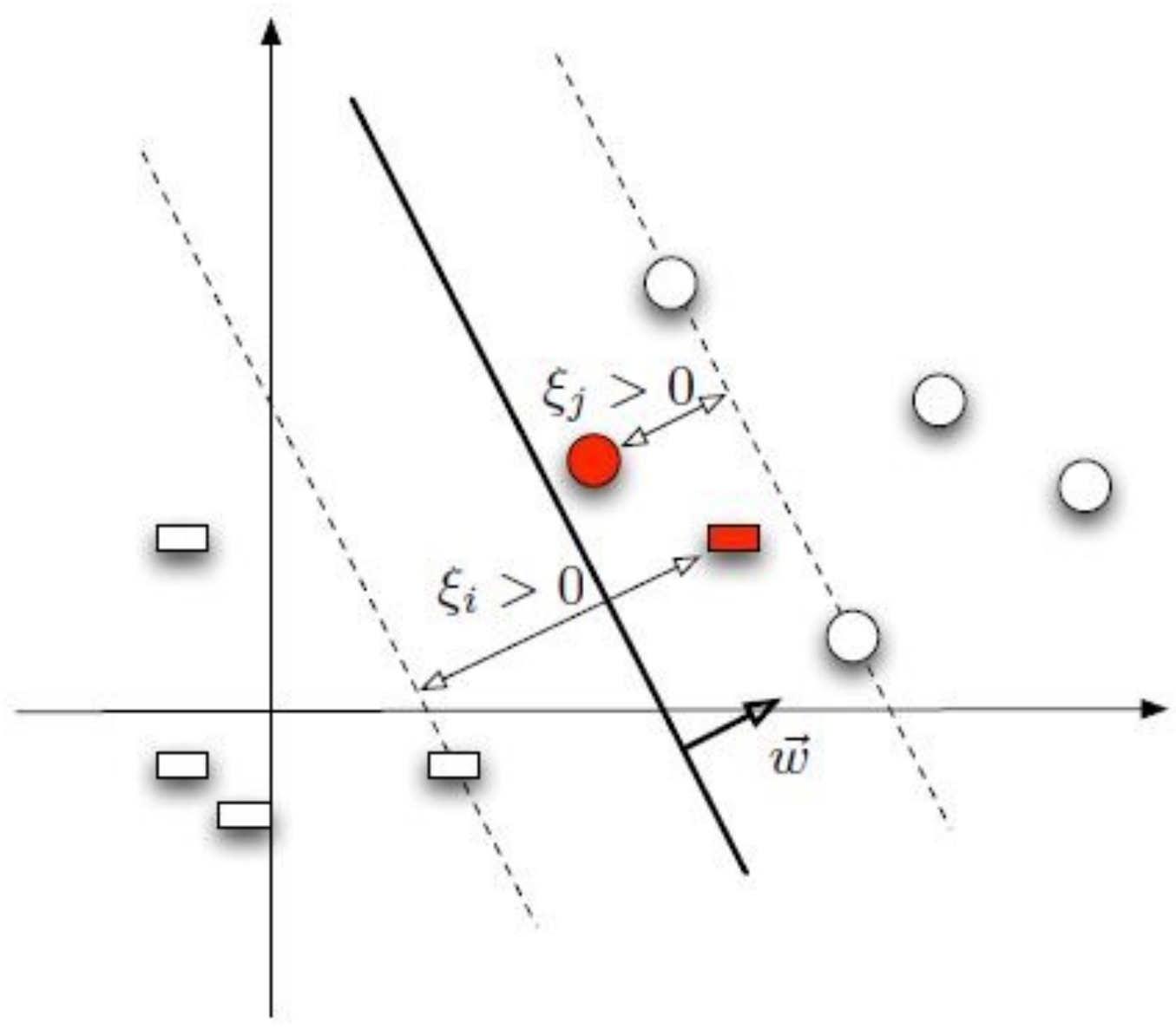
$$\min_{w,\xi} \frac{\|w\|^2}{2} + C \sum_i \xi_i$$

s.t.

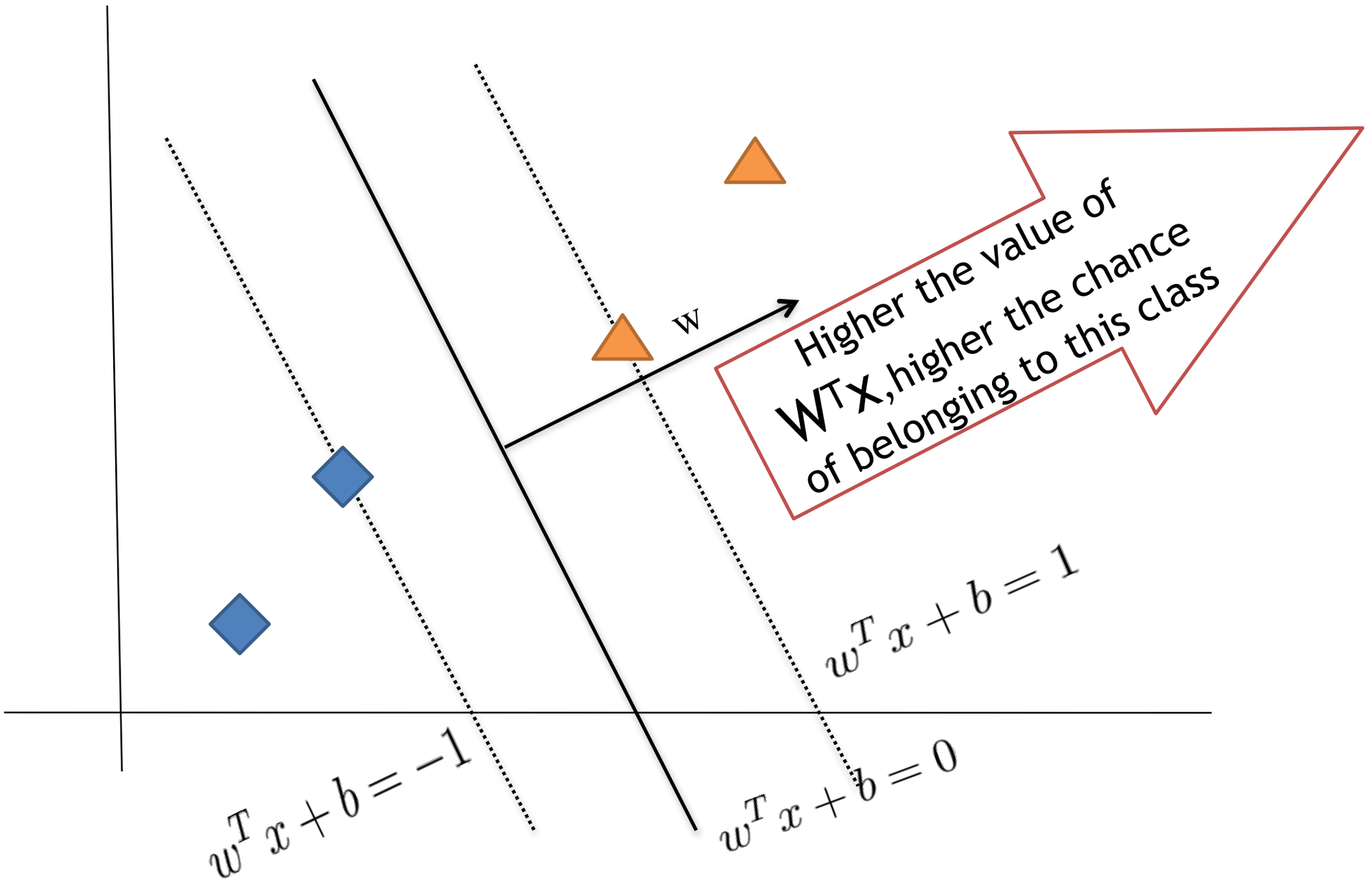
$$if \;\; y_i = Y_1, \quad w^T x_i + b \geq 1 - \xi_i$$
$$if \;\; y_i = Y_2, \quad w^T x_i + b \leq -(1 - \xi_i)$$

$$\forall i, \; \xi_i \geq 0$$

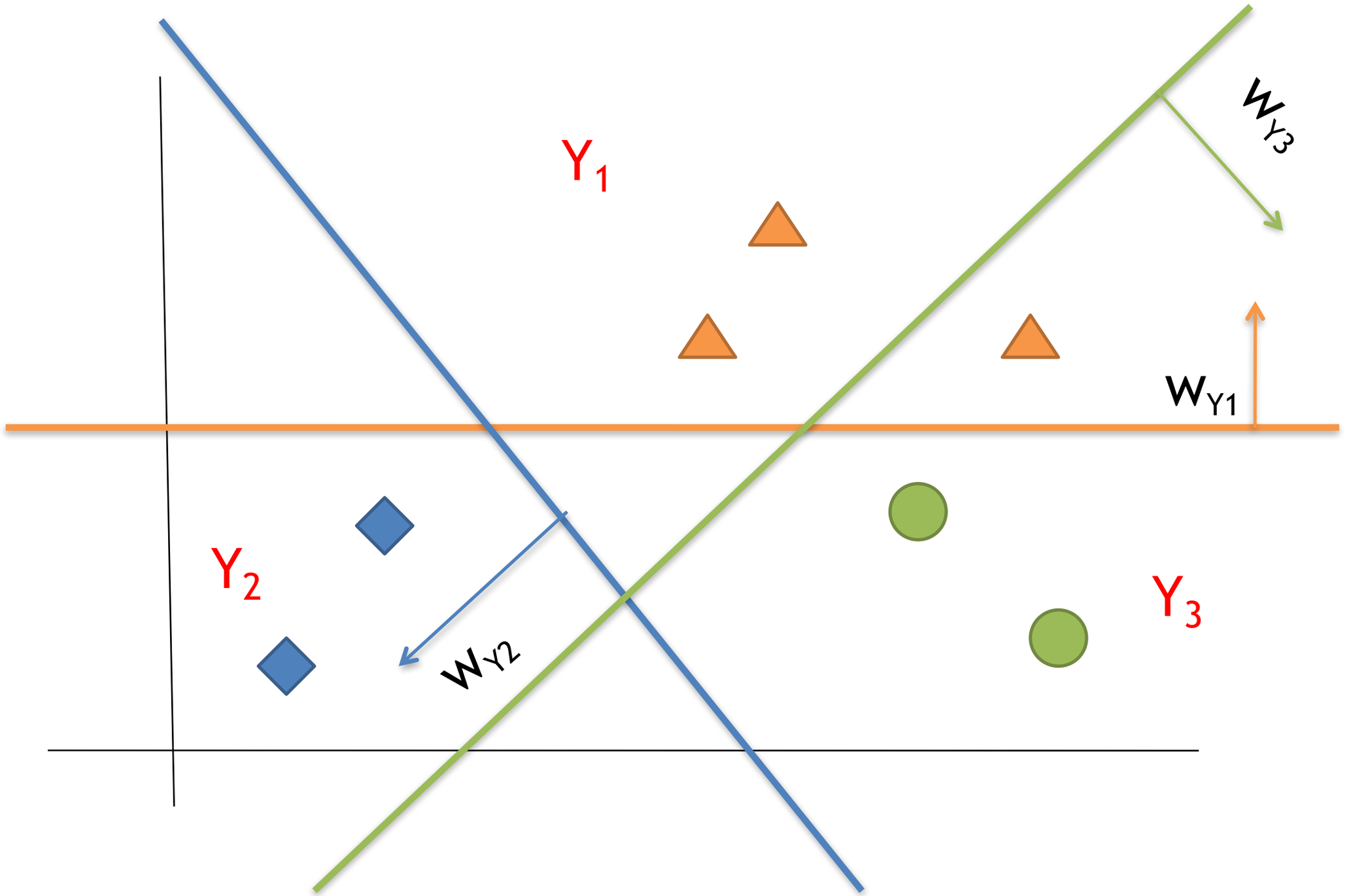
Soft-Margin SVM



Score


w
Higher the value of
$W^T X$, higher the chance
of belonging to this class
$w^T x + b = 1$
$w^T x + b = 0$
$w^T x + b = -1$

Multiclass SVM



Predicted Class: $\max_{Y_i} \quad w_{Y_i} \cdot x$

Multi-Class SVM

$$\min_{w,\xi} \frac{\|w\|^2}{2} + C \sum_i \xi_i$$

s.t.

$$w_{Y_i} \cdot x_i - w_{\hat{Y} \neq Y_i} \cdot x_i \geqslant 1 - \xi_i, \forall \ (x_i, Y_i)$$

$$\forall i, \hat{y} \neq y_i, \xi_i \geqslant 0$$

here, $w = \begin{pmatrix} w_{Y_1} \\ w_{Y_2} \\ w_{Y_3} \end{pmatrix}$

Multi-Class SVM

$$\min_{w,\xi} \frac{\|w\|^2}{2} + C \sum_i \xi_i$$

s.t.

$$w_{Y_i} \cdot x_i - w_{\hat{Y} \neq Y_i} \cdot x_i \geqslant \Delta(Y_i, Y) - \xi_i, \forall\, (x_i, Y_i)$$

$$\forall i, \hat{y} \neq y_i, \xi_i \geqslant 0$$

Multi-Class SVM

$$\min_{w,\xi} \frac{\|w\|^2}{2} + C \sum_{.} \xi_i$$

s.t.

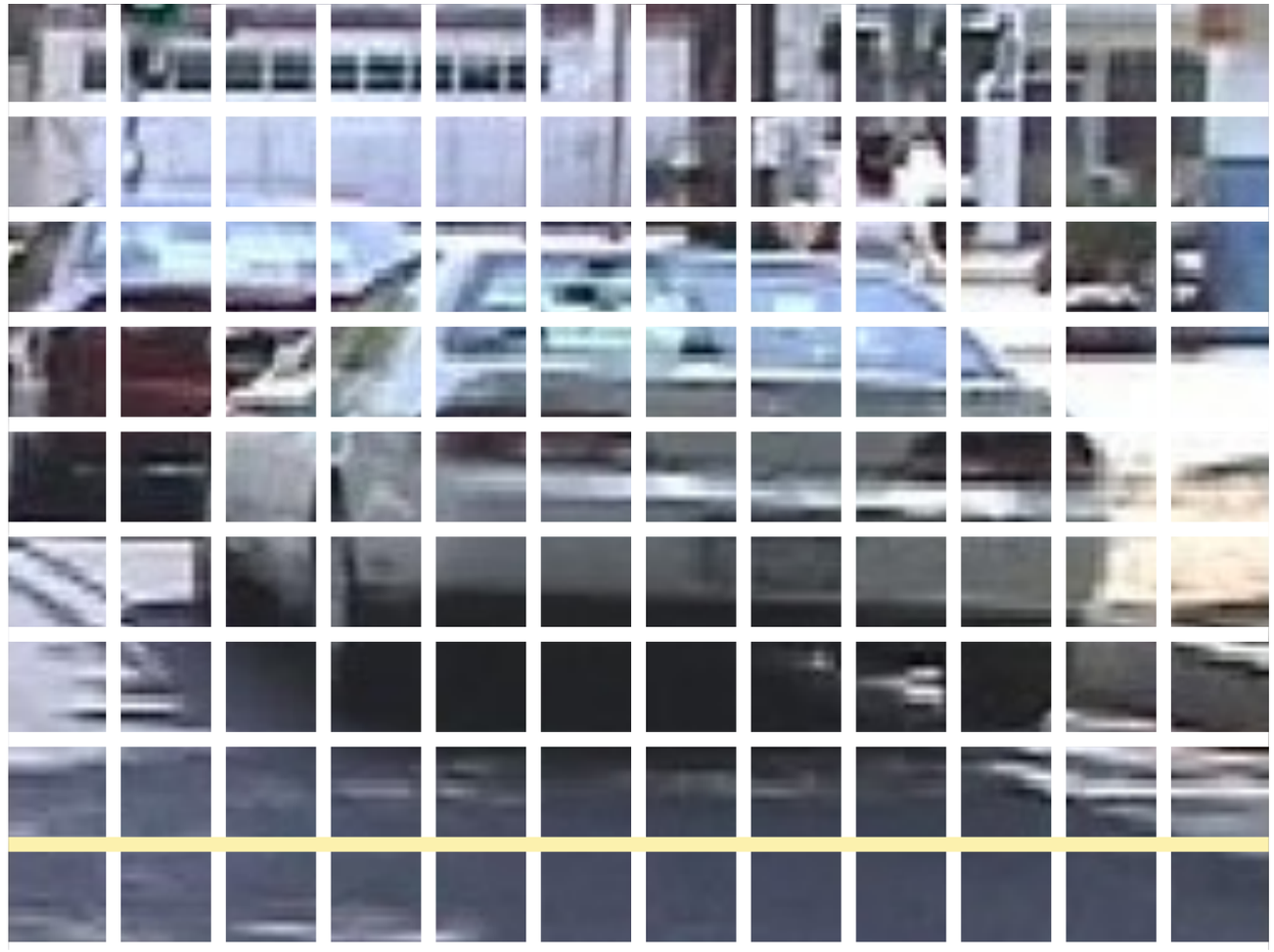
$$\boldsymbol{w} \cdot \boldsymbol{\Phi}(x_i, y_i) - \boldsymbol{w} \cdot \boldsymbol{\Phi}(x_i, \bar{y}) \geq \Delta(y_i, \bar{y}) - \xi_i$$

$$\forall i, \hat{y} \neq y_i, \xi_i \geqslant 0$$

$$w = \begin{pmatrix} \\ w_{Y_1} \\ w_{Y_2} \\ . \\ . \\ . \\ w_{Y_{k-1}} \\ w_{Y_k} \\ \end{pmatrix} \quad \Phi(x, y) = \begin{pmatrix} 0 \\ . \\ . \\ x \\ . \\ . \\ 0 \end{pmatrix}$$

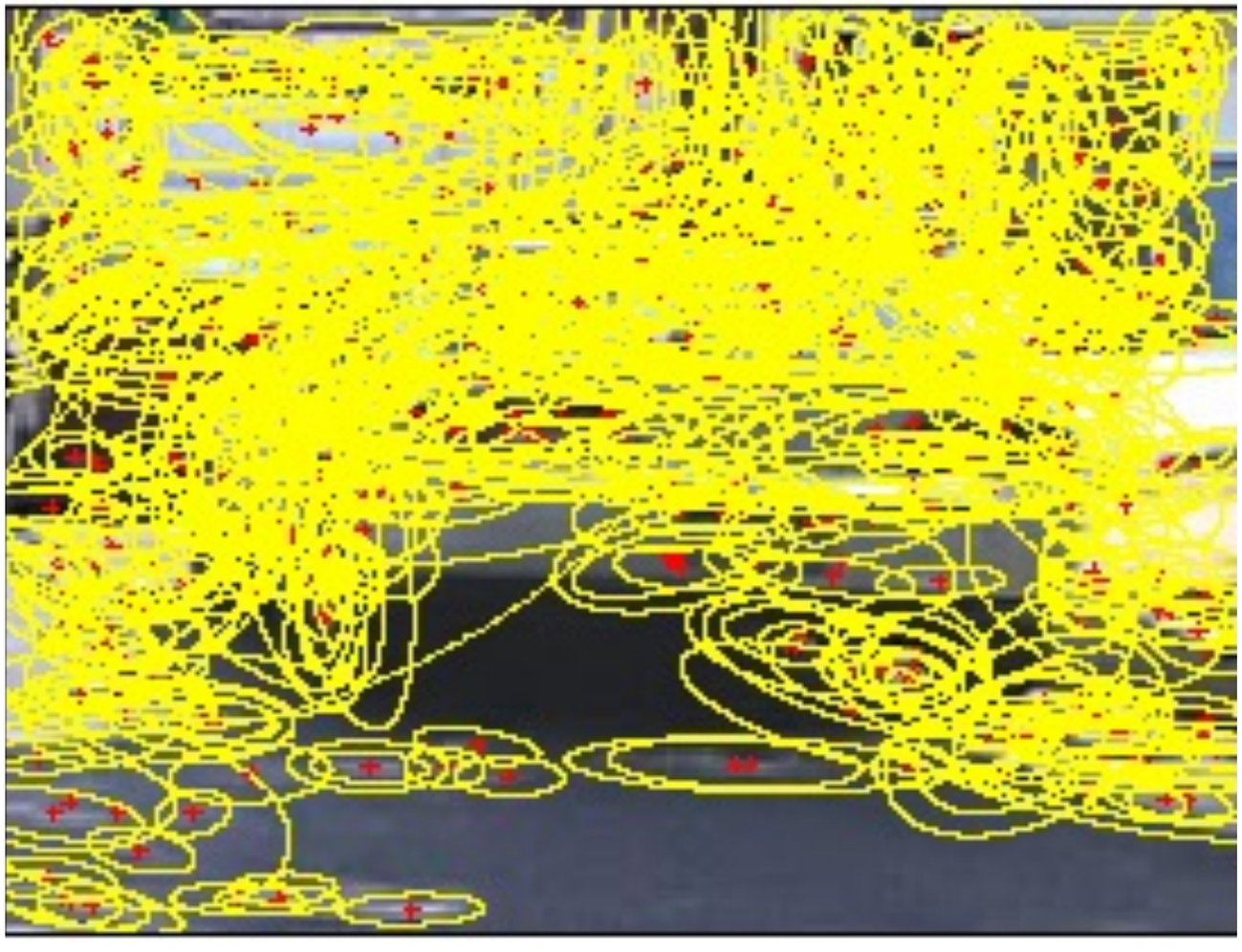
1.Feature detection and representation

- Regular grid
  - Vogel et al. 2003
  - Fei-Fei et al. 2005



1.Feature detection and representation

- Regular grid
  - Vogel et al. 2003
  - Fei-Fei et al. 2005
- Interest point detector
  - Csurka et al. 2004
  - Fei-Fei et al. 2005
  - Sivic et al. 2005



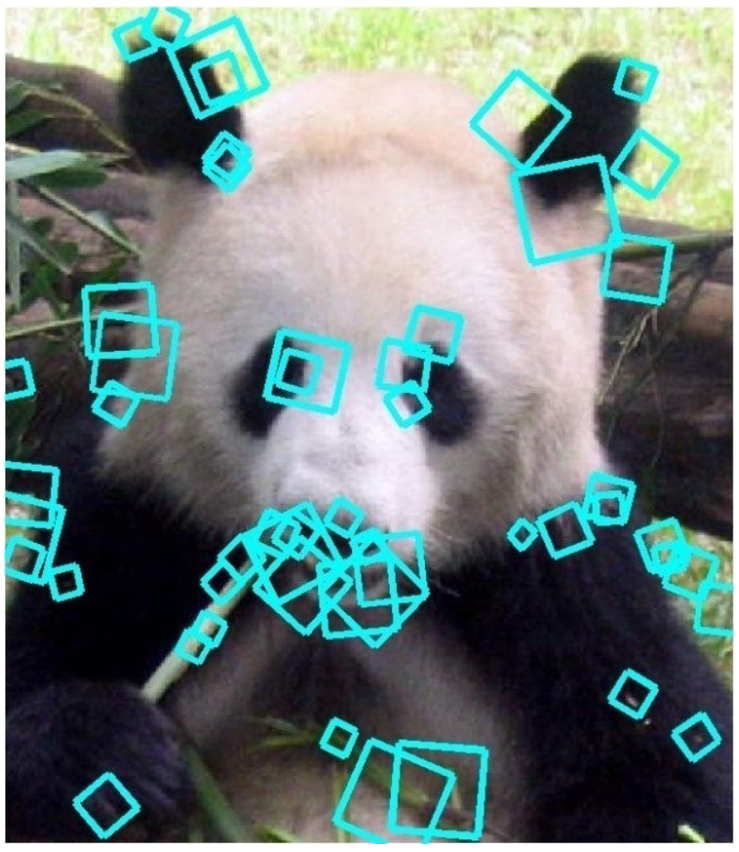
1.Feature detection and representation

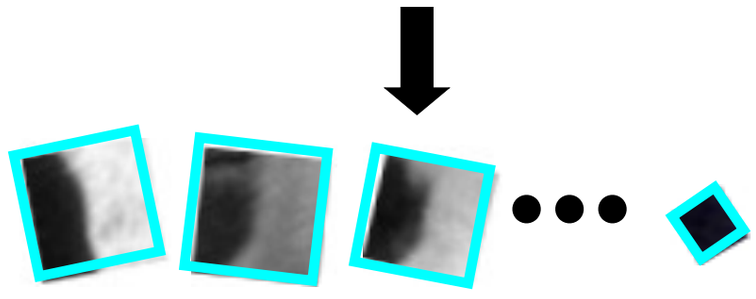
- Regular grid
  - Vogel et al. 2003
  - Fei-Fei et al. 2005
- Interest point detector
  - Csurka et al. 2004
  - Fei-Fei et al. 2005
  - Sivic et al. 2005
- Other methods
  - Random sampling (Ullman et al. 2002)

The Pyramid Match Kernel: Discriminative Classification with Sets of Image Features

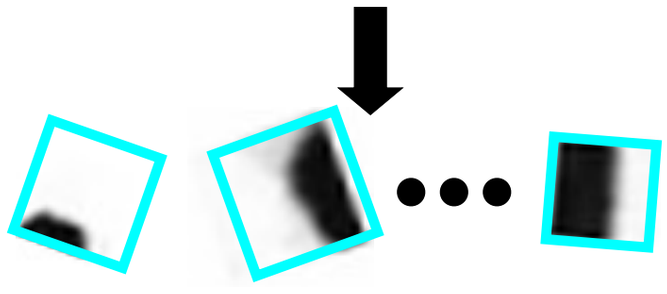
Kristen Grauman
Trevor Darrell
MIT

Sets of features





$$\mathbf{X} = \{\vec{\mathbf{x}}_1, \ldots, \vec{\mathbf{x}}_m\}$$



$$\mathbf{Y} = \{\vec{\mathbf{y}}_1, \ldots, \vec{\mathbf{y}}_n\}$$

Problem

How to build a **discriminative classifier** using the set representation?

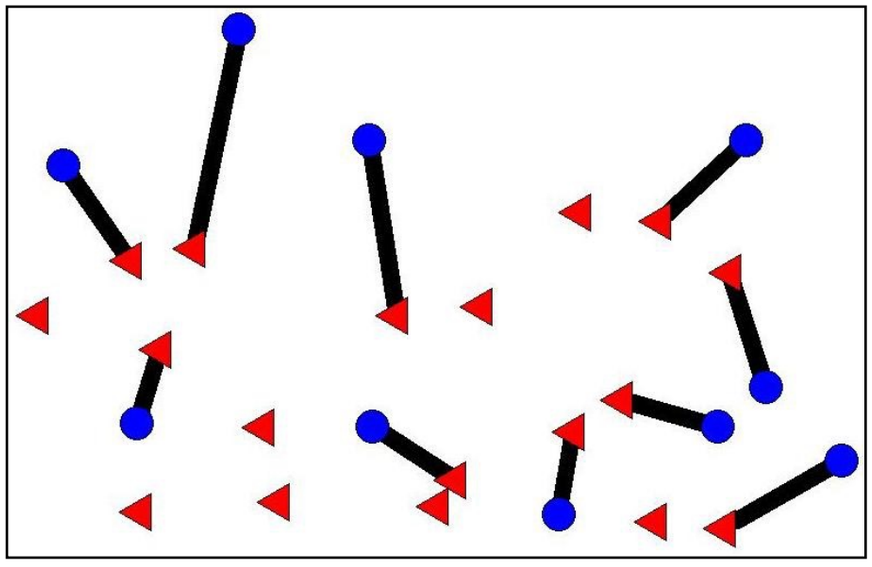
Kernel-based methods (e.g. SVM) are appealing for efficiency and generalization power…

But what is an appropriate kernel?

- Each instance is unordered set of vectors
- Varying number of vectors per instance

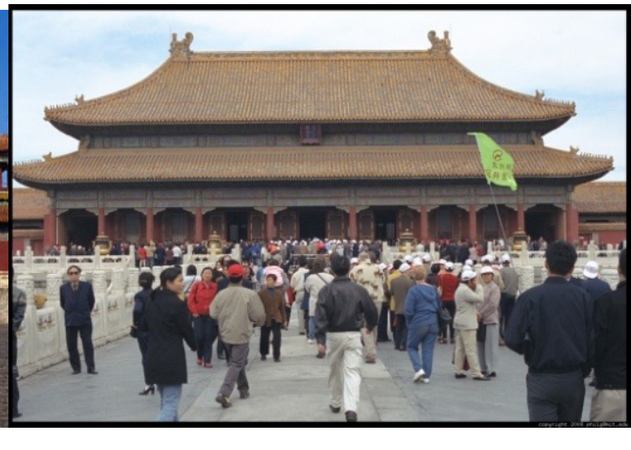
Partial matching for sets of features

Compare sets by computing a *partial matching* between their features.



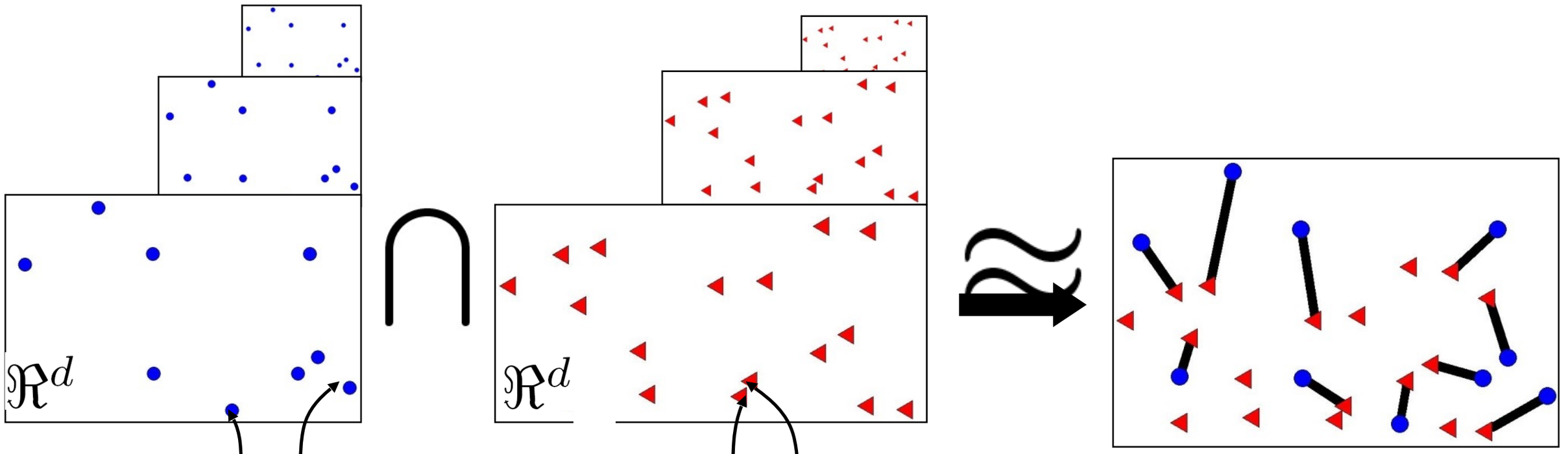




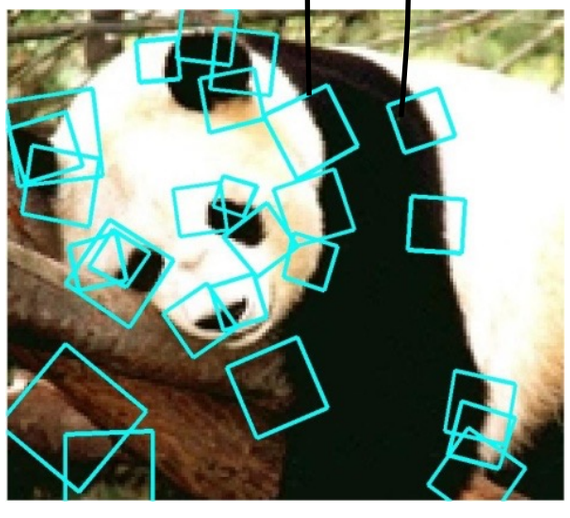



Robust to clutter, segmentation errors, occlusion…

Pyramid match



optimal partial matching



$$\mathbf{X} = \{\vec{\mathbf{x}}_1, \ldots, \vec{\mathbf{x}}_m\}$$
$$\vec{\mathbf{x}}_i \in \Re^d$$

$$\mathbf{Y} = \{\vec{\mathbf{y}}_1, \ldots, \vec{\mathbf{y}}_n\}$$
$$\vec{\mathbf{y}}_i \in \Re^d$$

$$\max_{\pi:\mathbf{X}\to\mathbf{Y}} \sum_{\mathbf{x}_i \in \mathbf{X}} \mathcal{S}(\mathbf{x}_i, \pi(\mathbf{x}_i))$$