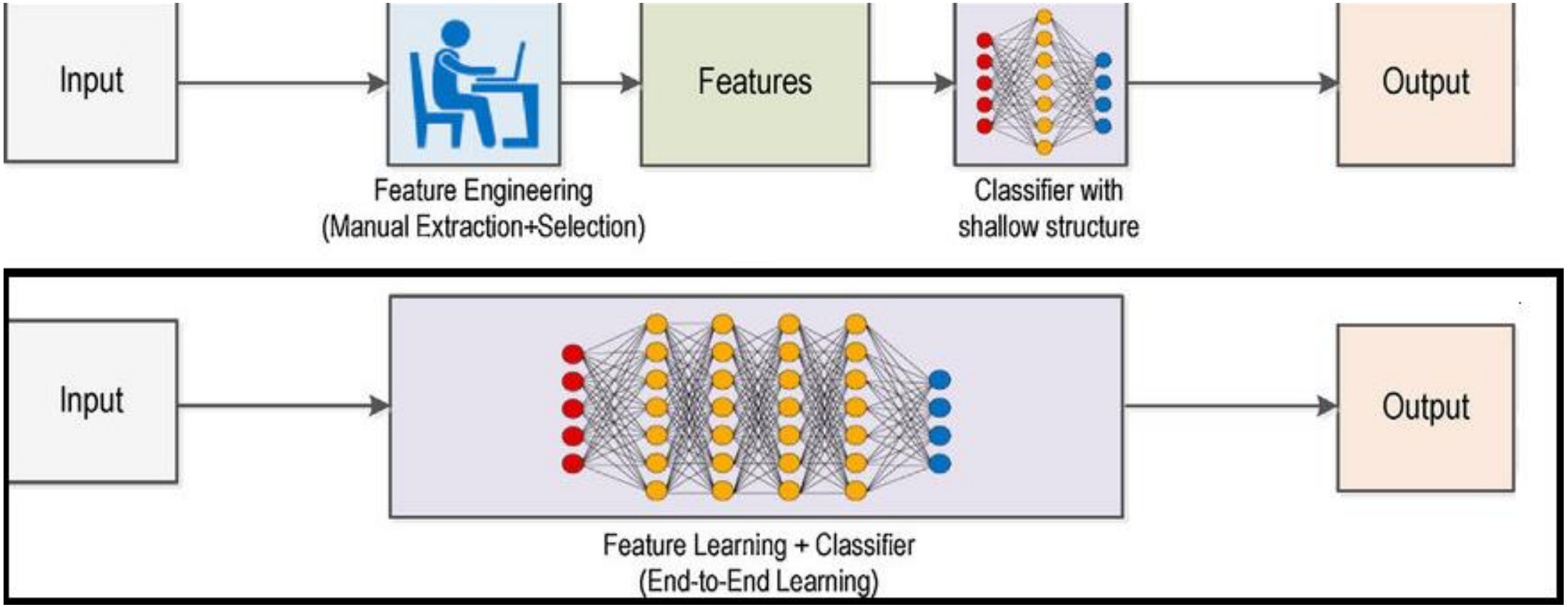
Feature extraction from Images

Prof. Biplab Banerjee

Traditional ML vs Deep Learning (Recap)



Feature learning

Object classification accuracy on ImageNet (ILSVRC)

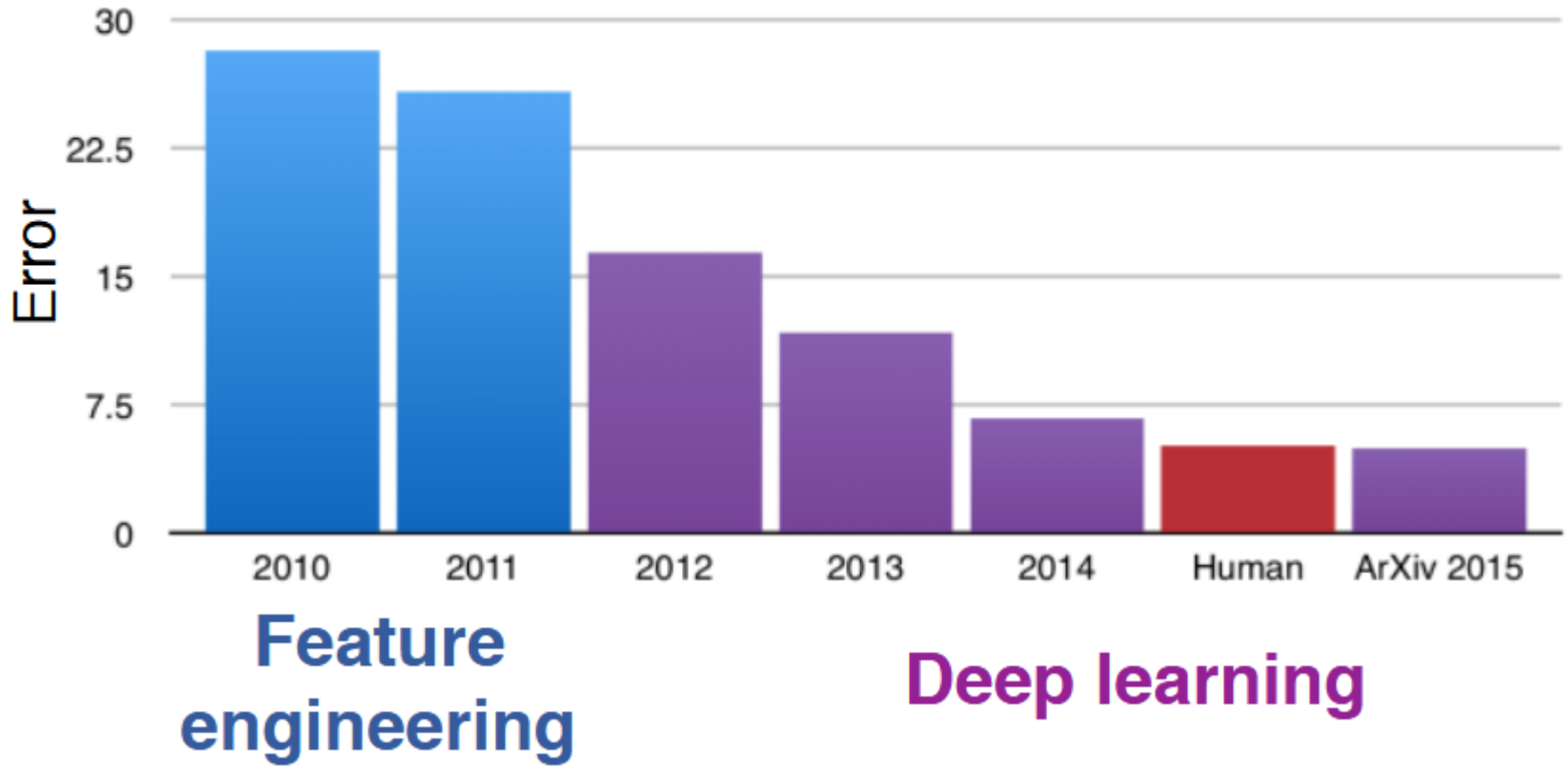
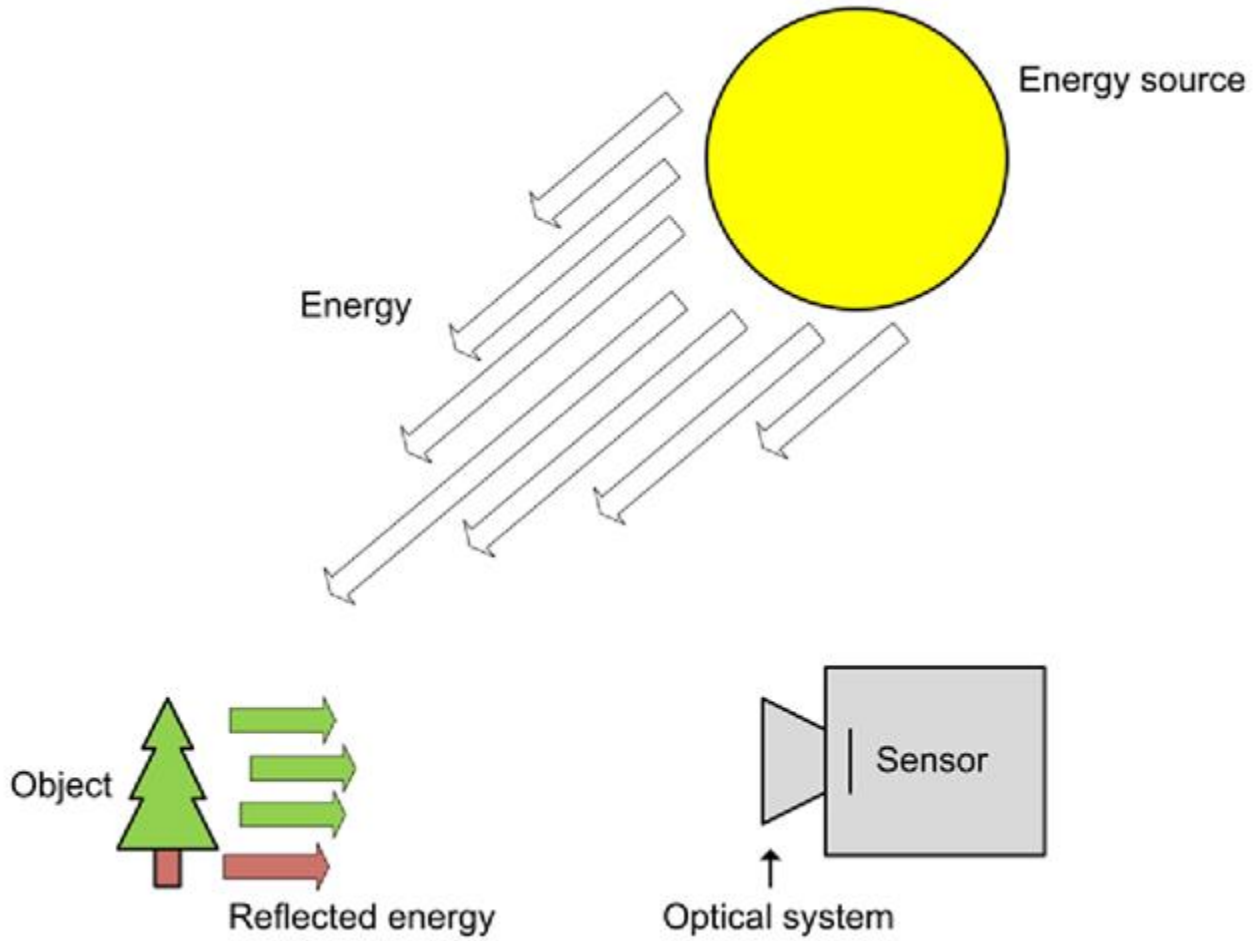
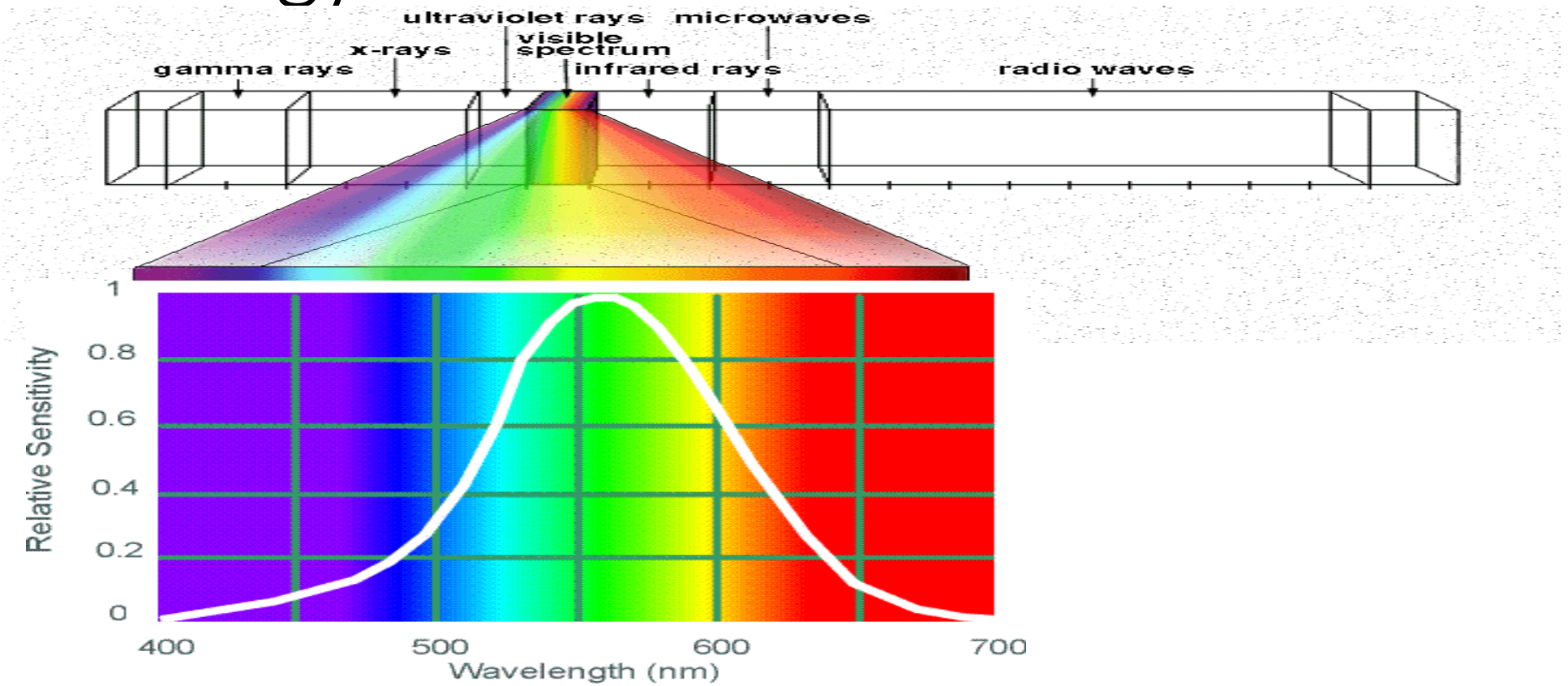


Image acquisition

$$z = f(x, y)$$



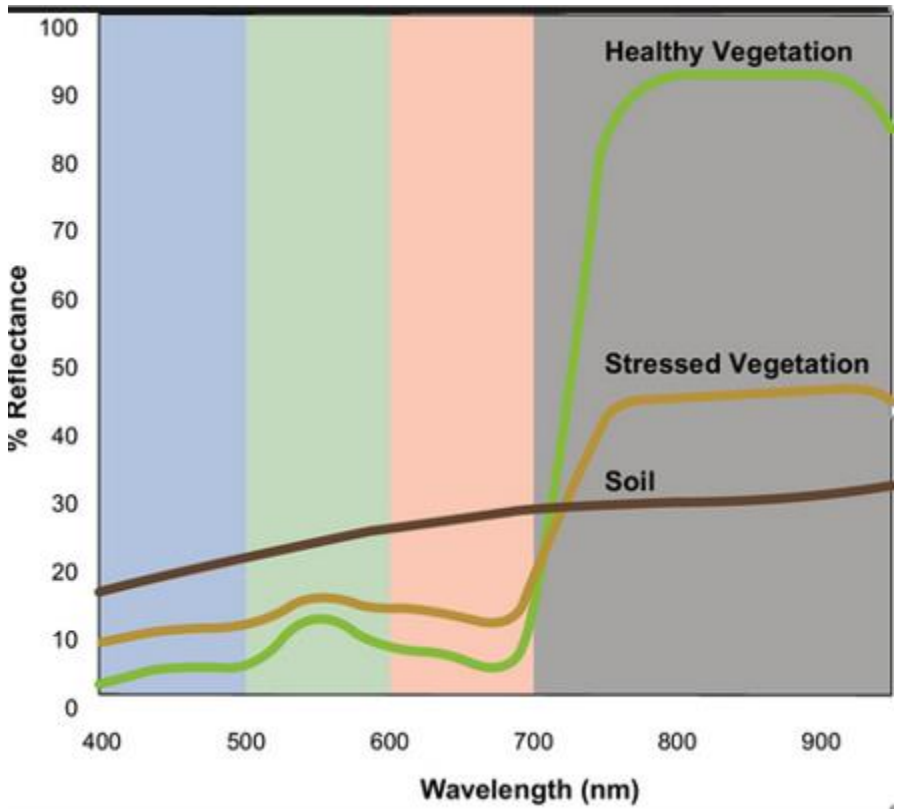
Electromagnetic Spectrum – characterizing the energy



Human Luminance Sensitivity Function

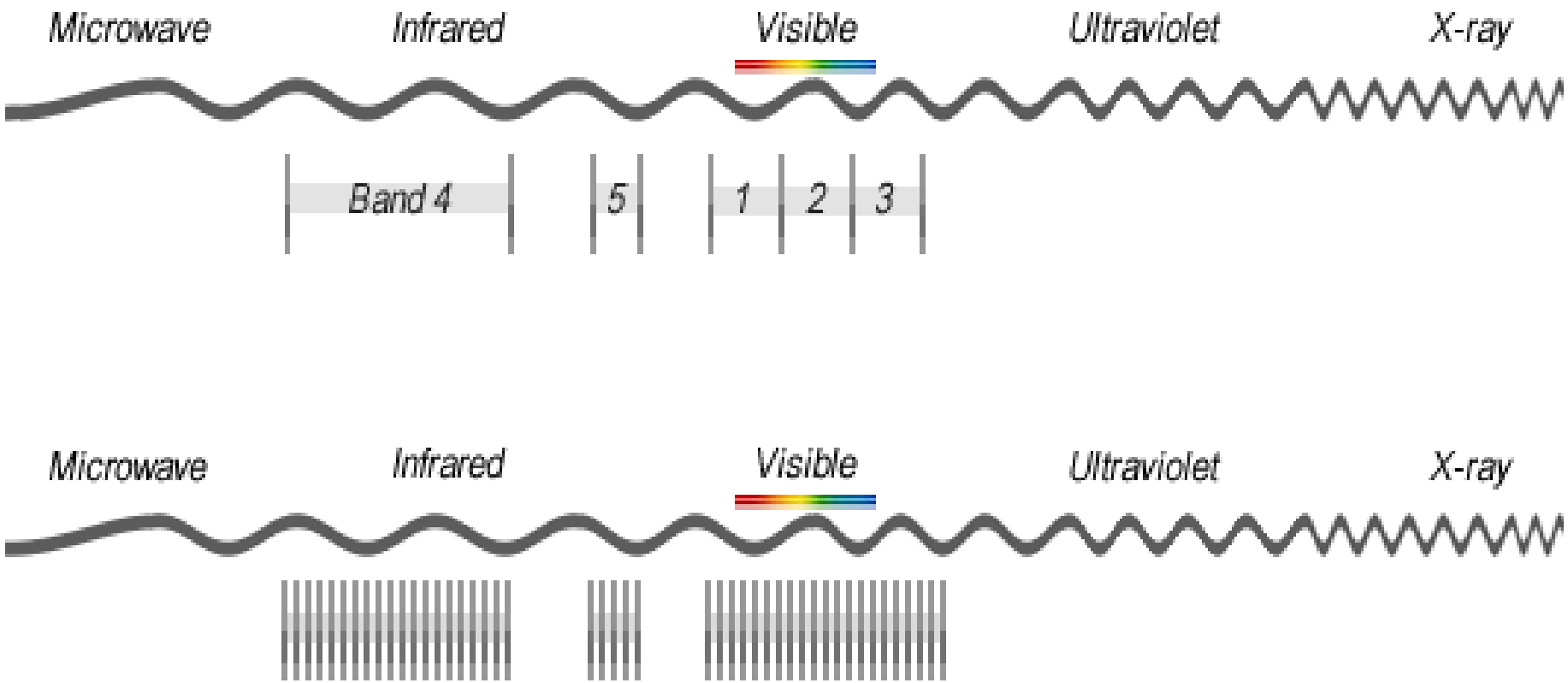
http://www.yorku.ca/eye/photopik.htm

Beyond visible spectra – satellite images

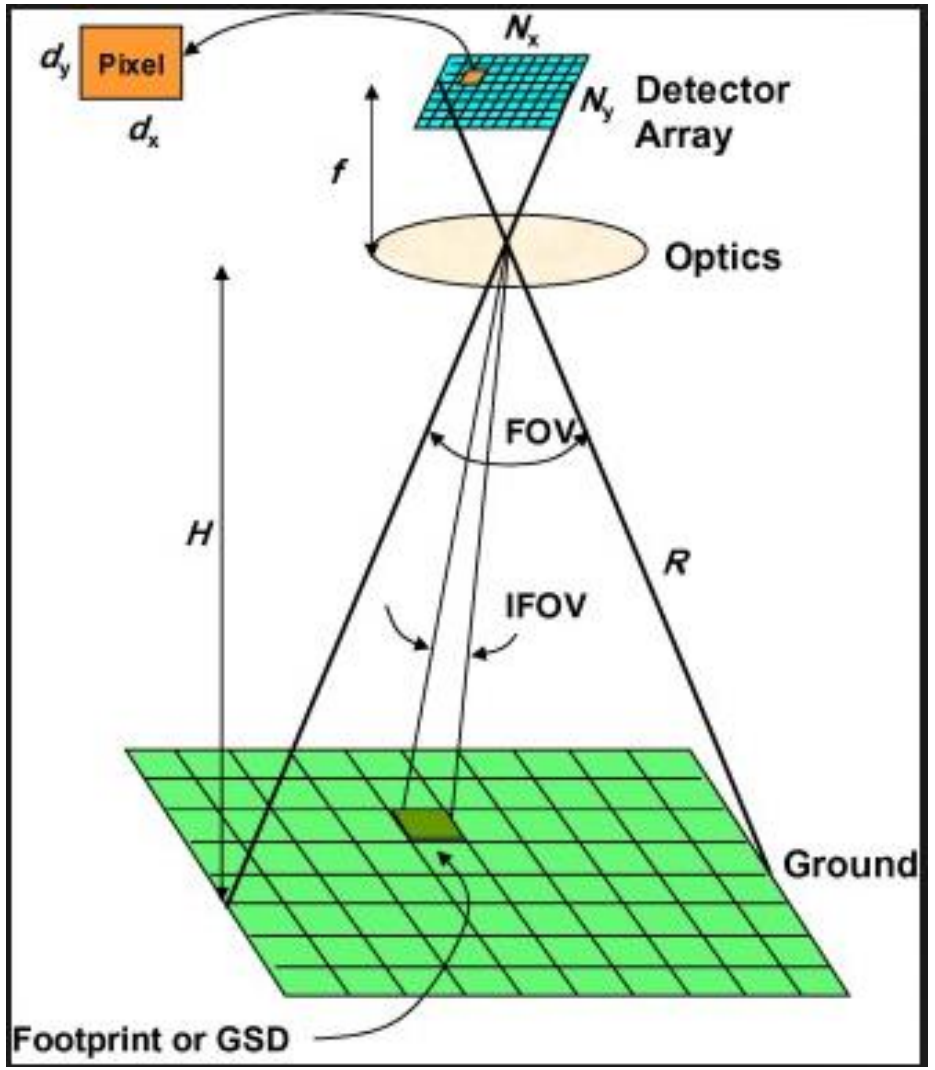


The Infra-red region is better in characterizing the classes that the visible region

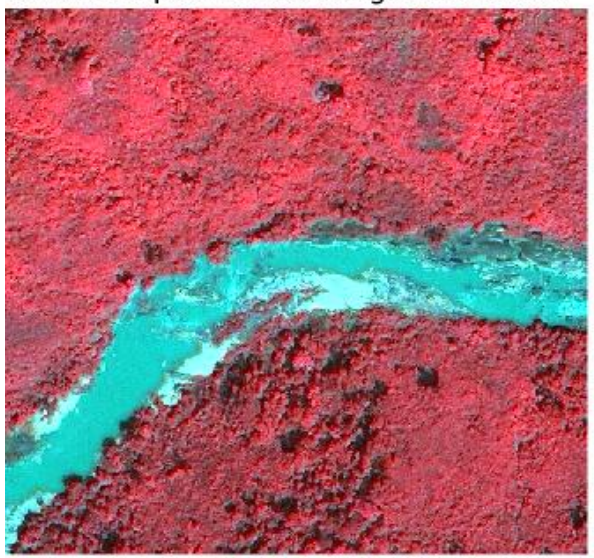
Multi-spectral and hyper-spectral



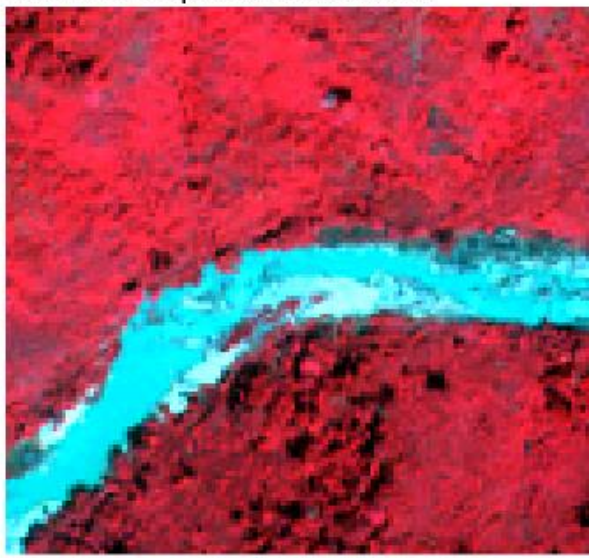
Spatial resolution

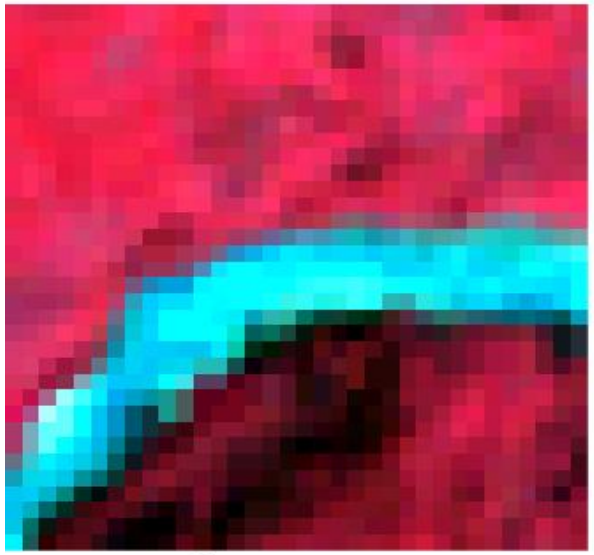


IKONOS 1m panchromatic merge 1m resolution

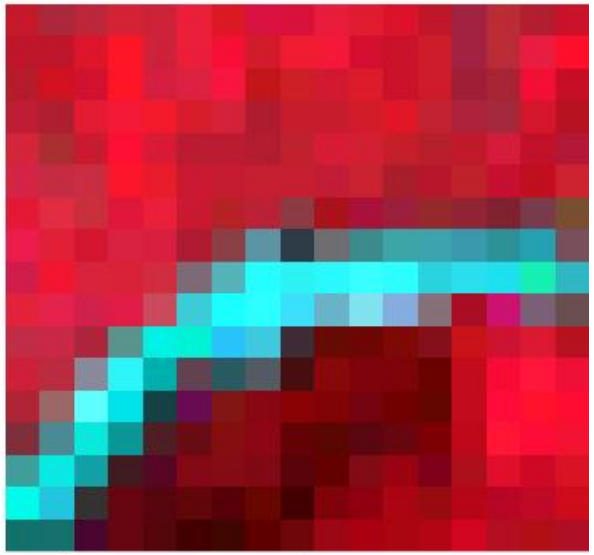


IKONOS multispectral 4m resolution

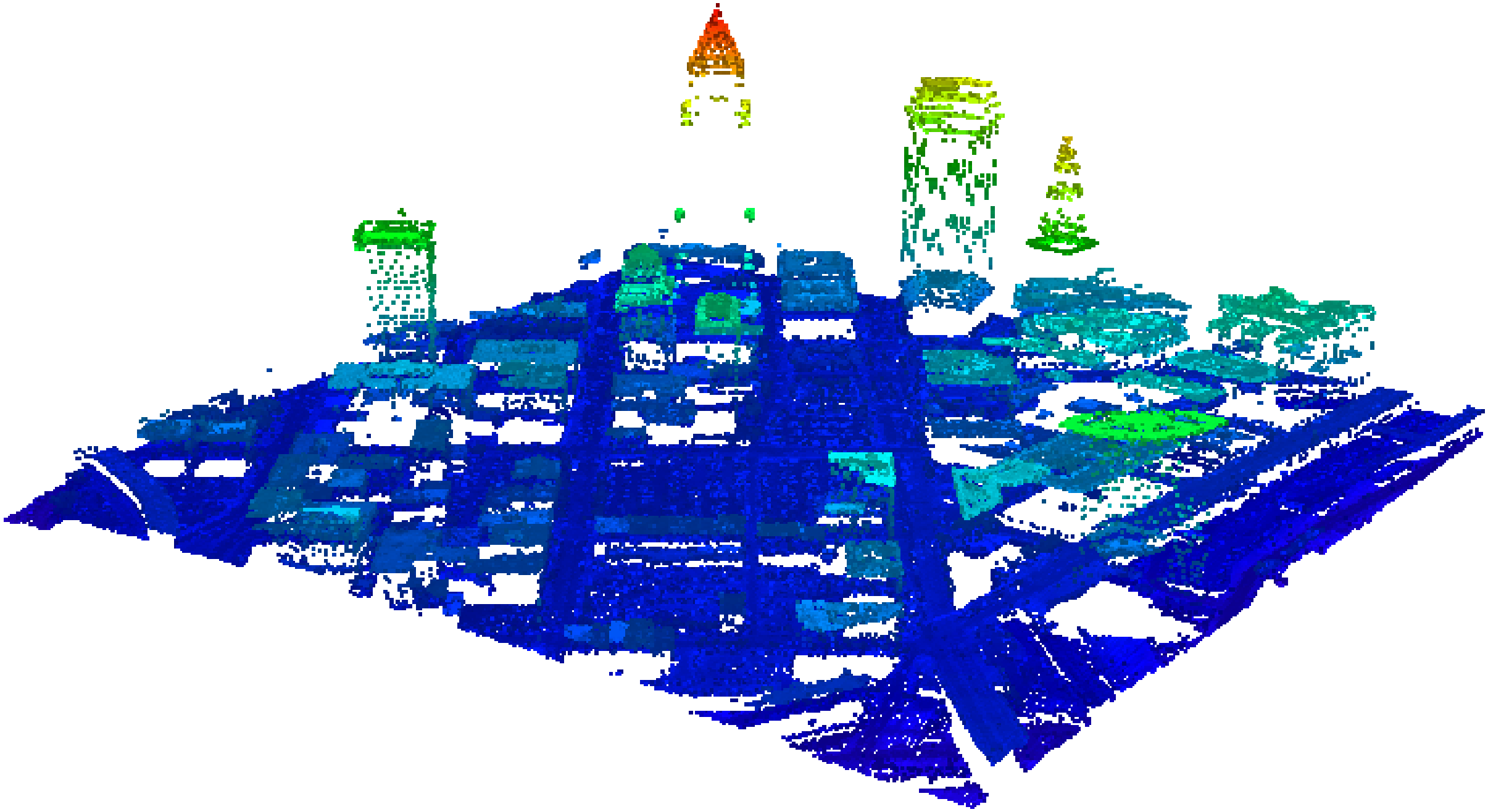




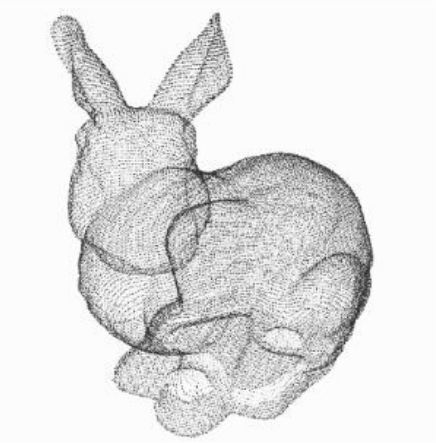
ASTER 15m resolution



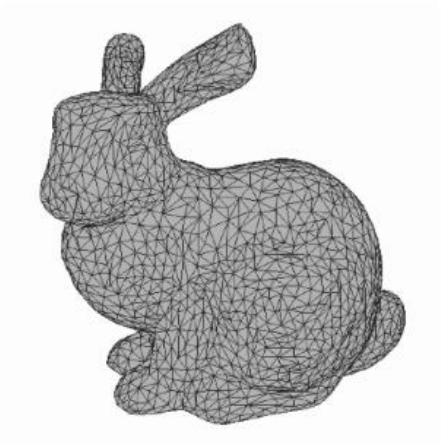
Landsat ETM+ 30m resolution



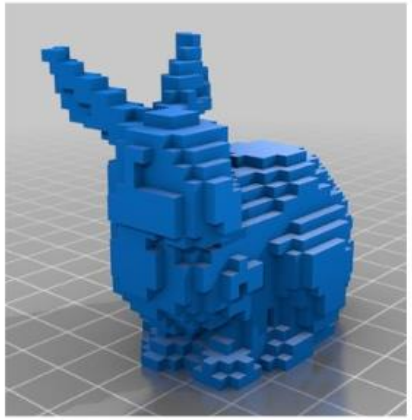
Different rep. of 3D information in images



Point Cloud



Mesh



Volumetric

Projected View
RGB(D)

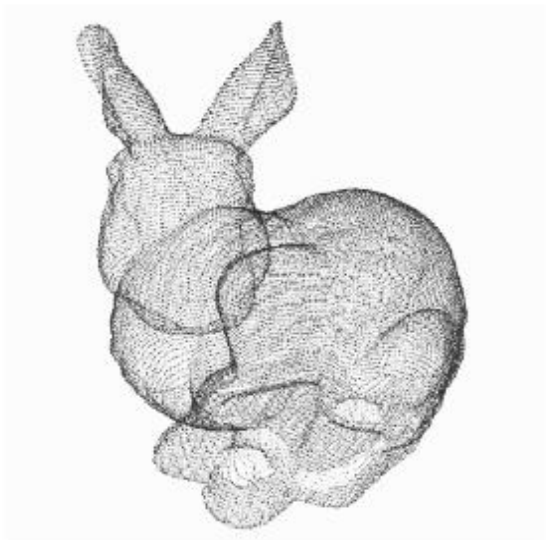
...

Point cloud is close to raw sensor data

Point cloud is canonical

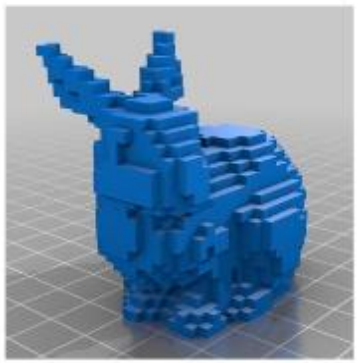
LiDAR

Depth Sensor



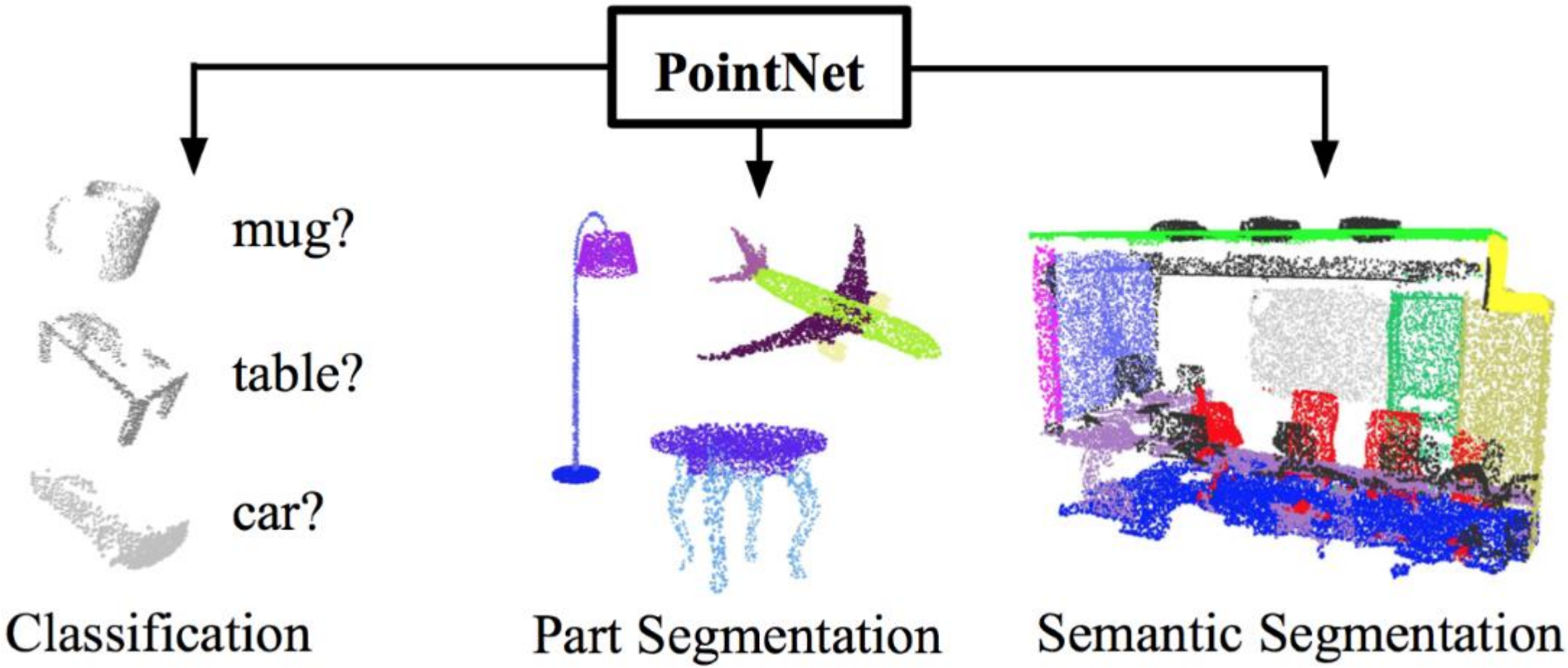
Point Cloud

Mesh

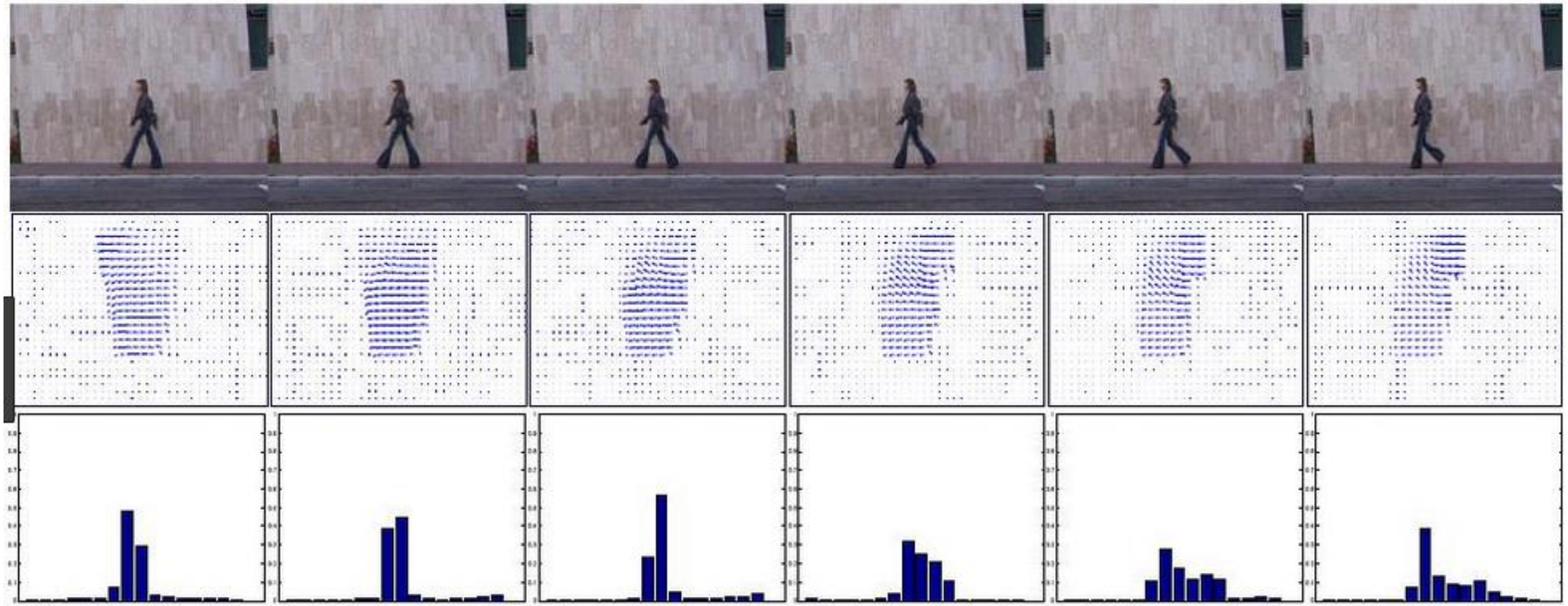


Volumetric

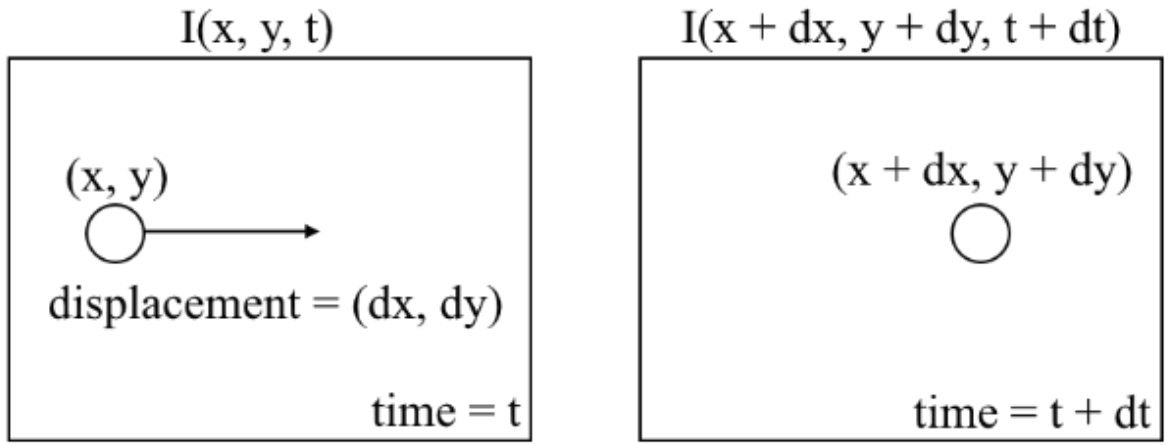
Depth Map


PointNet
mug?
table?
car?
Classification
Part Segmentation
Semantic Segmentation

Video data – appearance + motion (optical flow)



Optical flow - a quick review



$$I(x, y, t) = I(x + \delta x, y + \delta y, t + \delta t)$$

$$I(x + \delta x, y + \delta y, t + \delta t) = I(x, y, t) + \frac{\partial I}{\partial x}\delta x + \frac{\partial I}{\partial y}\delta y + \frac{\partial I}{\partial t}\delta t + \dots$$

$$\Rightarrow \frac{\partial I}{\partial x}\delta x + \frac{\partial I}{\partial y}\delta y + \frac{\partial I}{\partial t}\delta t = 0$$

$$\frac{\partial I}{\partial x}u + \frac{\partial I}{\partial y}v + \frac{\partial I}{\partial t} = 0$$

Color coding the optical flow

What are Image Features?

Image

0.22
0.30
0.13
0.24
0.31
0.15
0.35
0.48

Color histogram?

Maximum color on sub-areas of the image?

Any statistics on the input image?

The output of some image processing on the input image?

Why are they useful?

Image

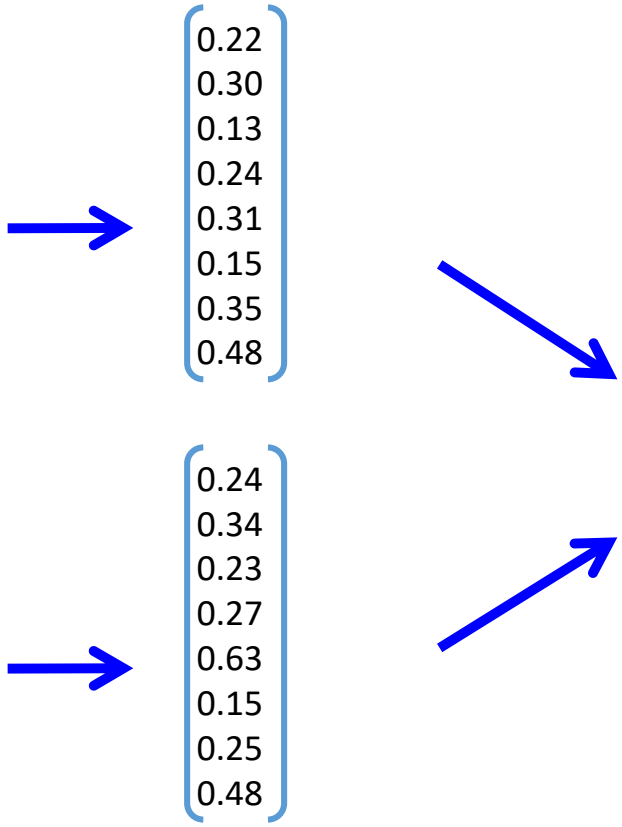
0.22
0.30
0.13
0.24
0.31
0.15
0.35
0.48

Machine Learning Model

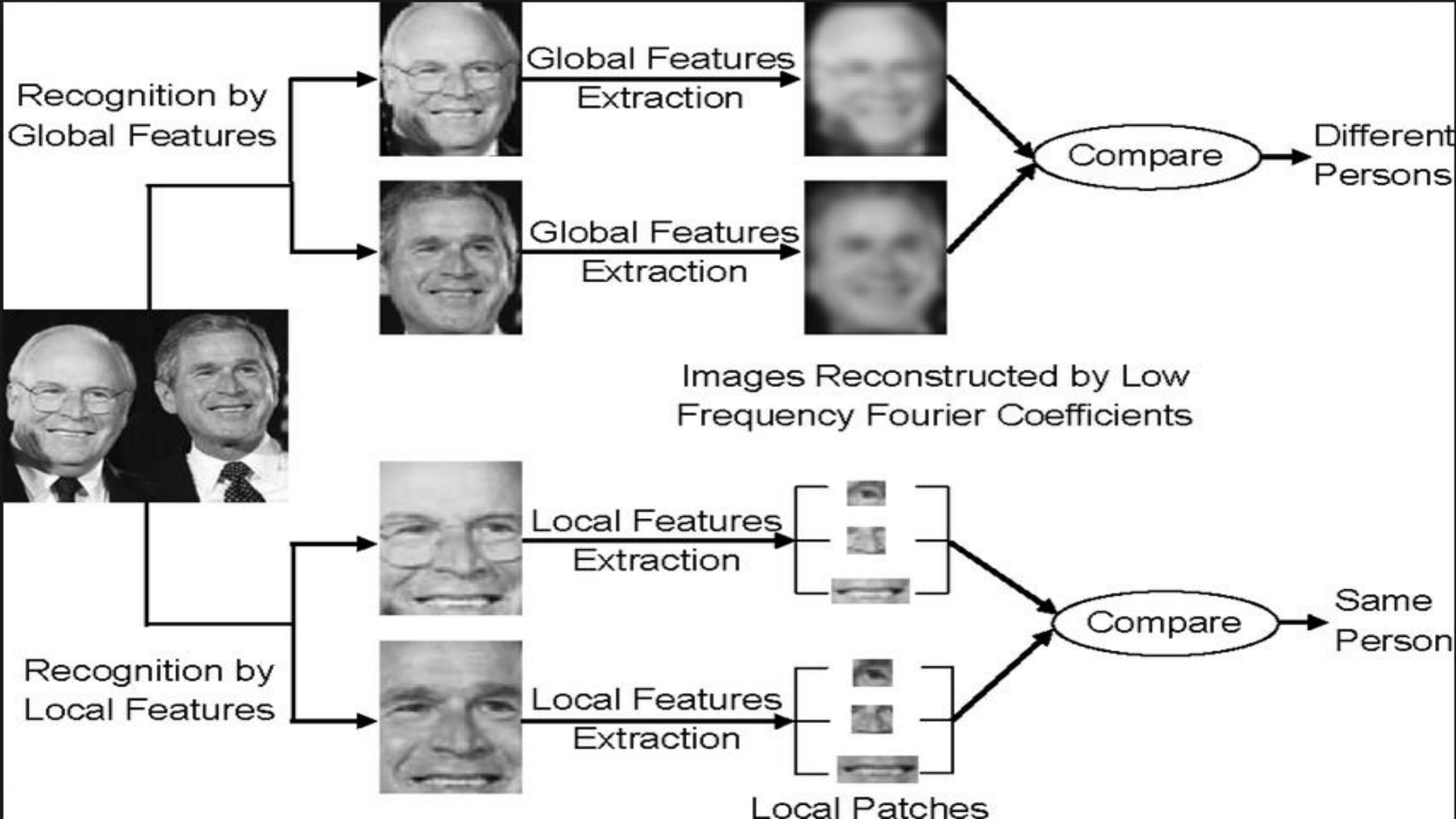
Predictions

As inputs to a machine learning model

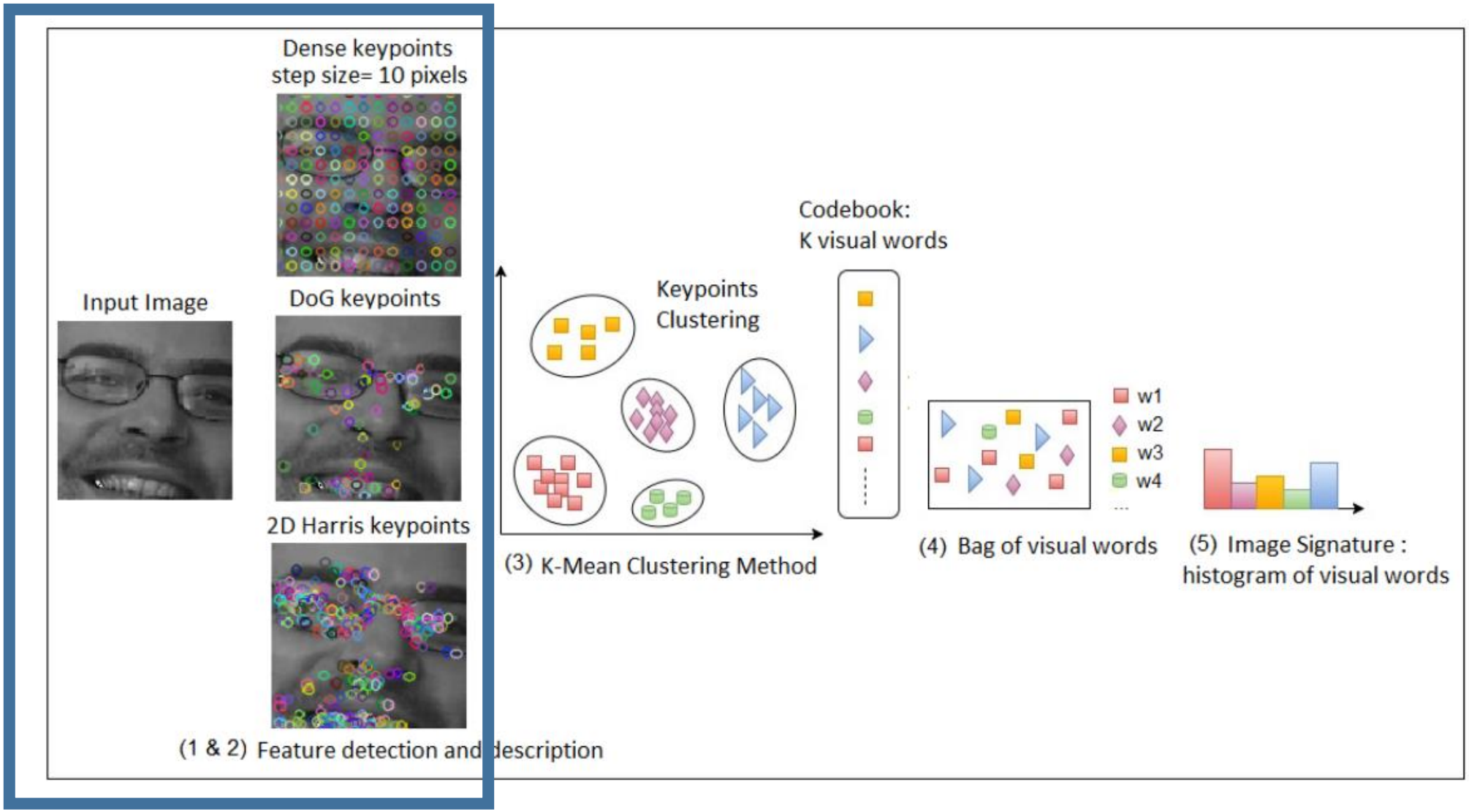
Why are they useful?



To compare images (i.e. retrieve similar images)

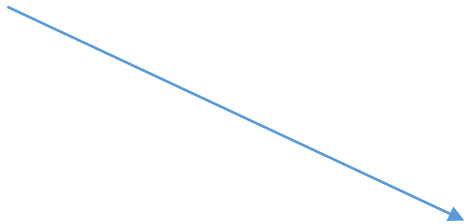

Recognition by
Global Features
Global Features
Extraction
Global Features
Extraction
Compare
Different
Persons
Images Reconstructed by Low
Frequency Fourier Coefficients
Recognition by
Local Features
Local Features
Extraction
Local Features
Extraction
Compare
Same
Person
Local Patches

Feature detection and description



Visual features

- Color histogram
- Edge & boundary feature
- Shape feature
- Texture feature
- Interests points based feature
- **Deep features**



Invariant to transformations
- Scale
- Rotation
- Translation
- Affine transformation
- Illumination change
- Some more

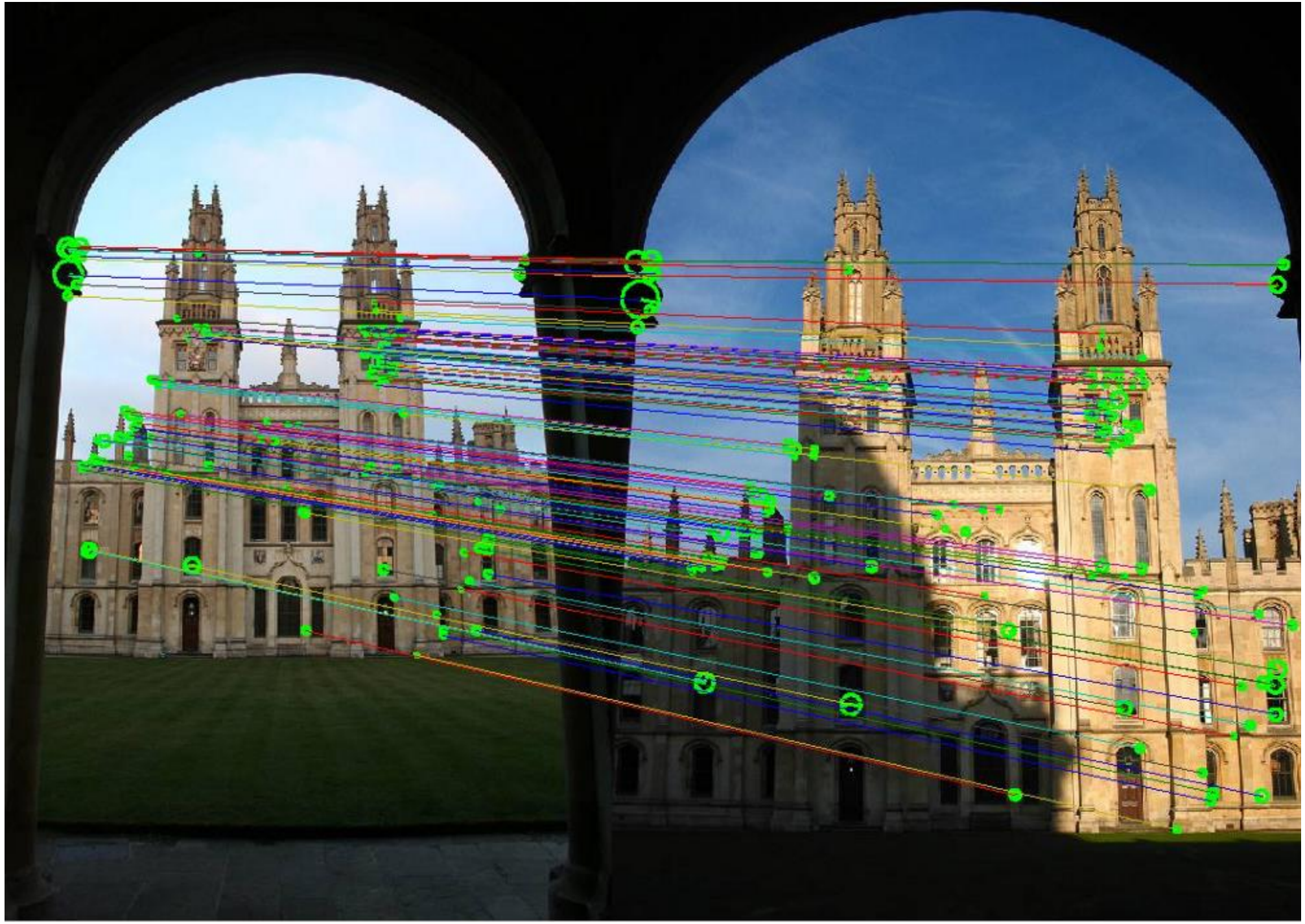


Image Features: Color

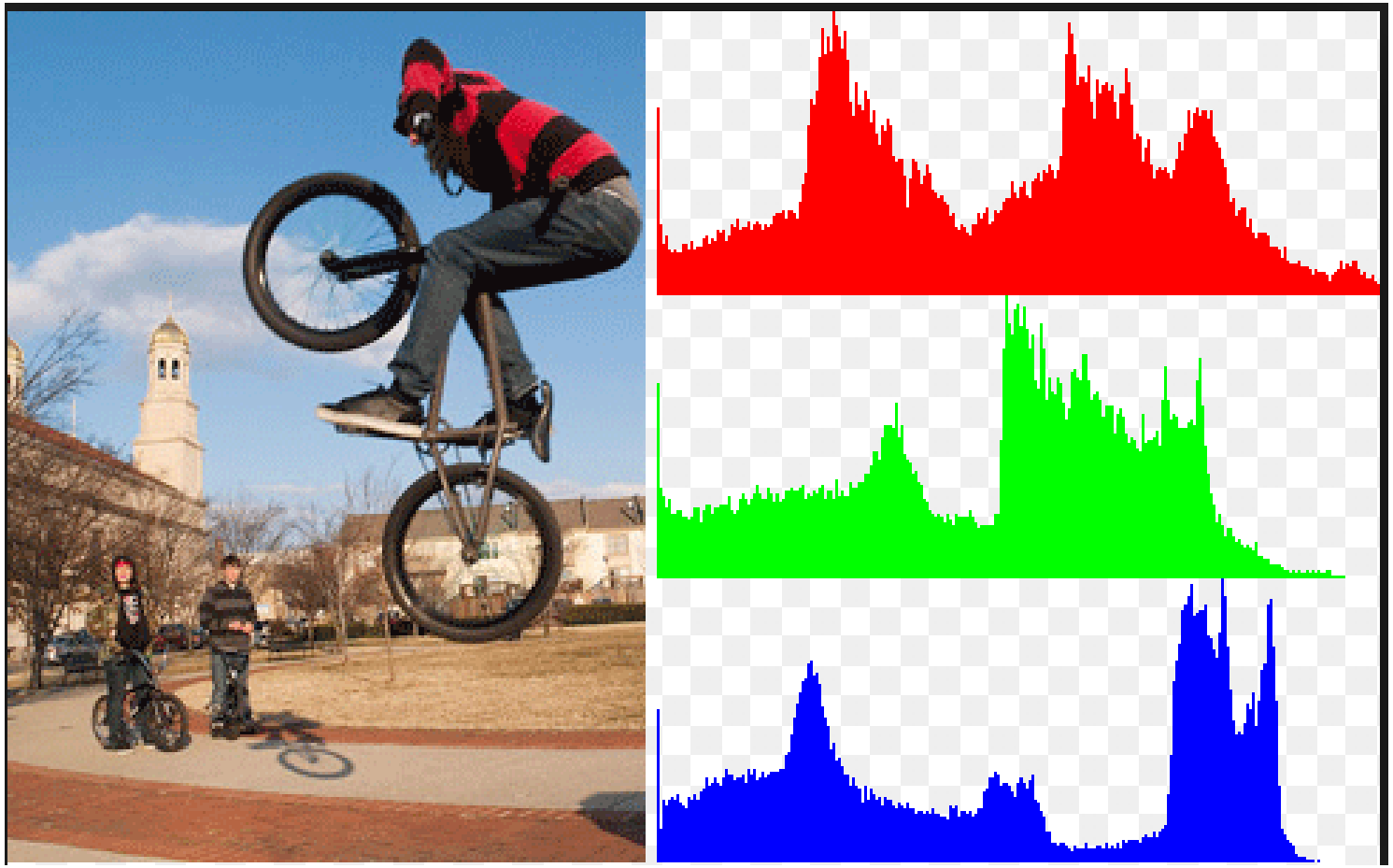


Image Features: Color

80 million tiny images: a large dataset for non-parametric object and scene recognition

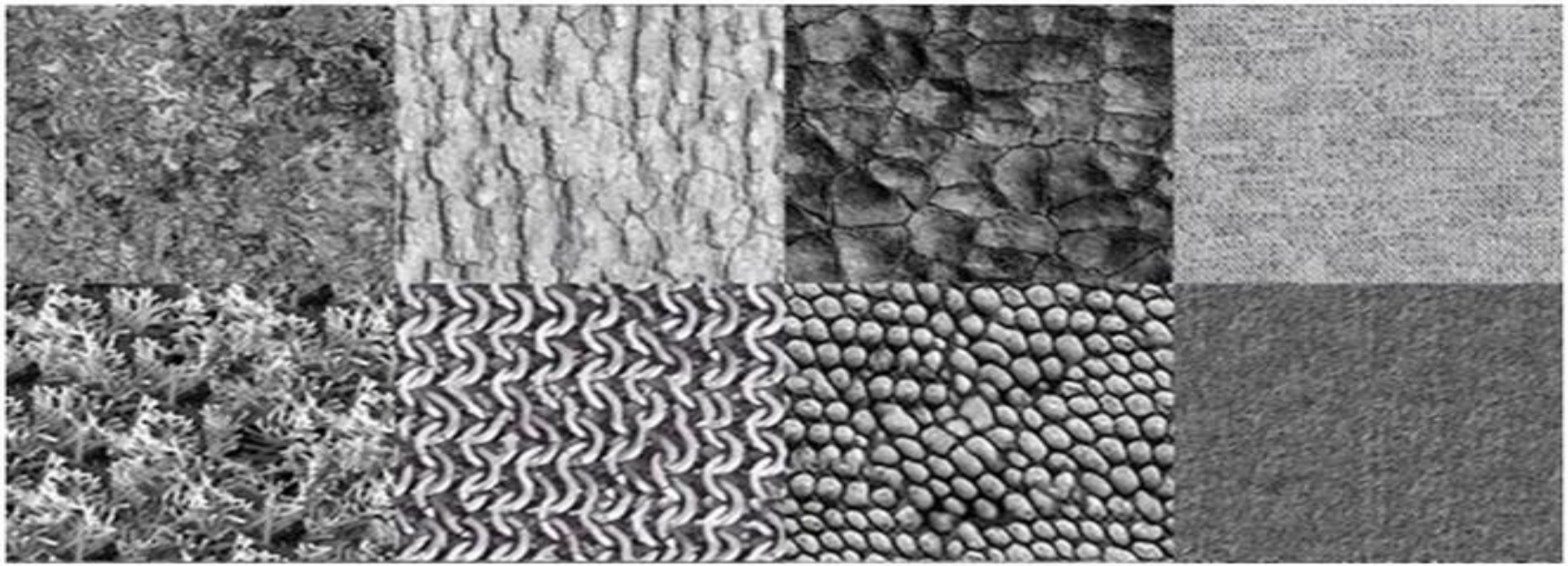
Antonio Torralba, Rob Fergus and William T. Freeman

Color often not a powerful feature

However, these are all images of people but the colors in each image are very different.

Texture feature

Texture is a description of the spatial arrangement of color or intensities in an image or a selected region of an image.



But textures are not always easy to quantify

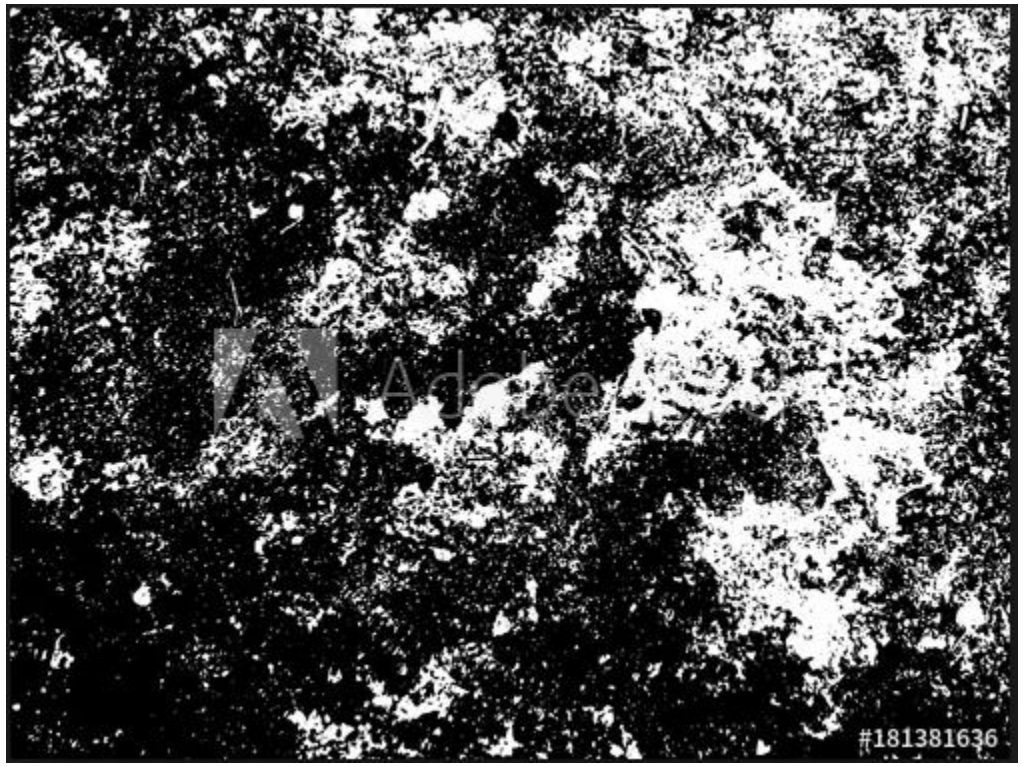
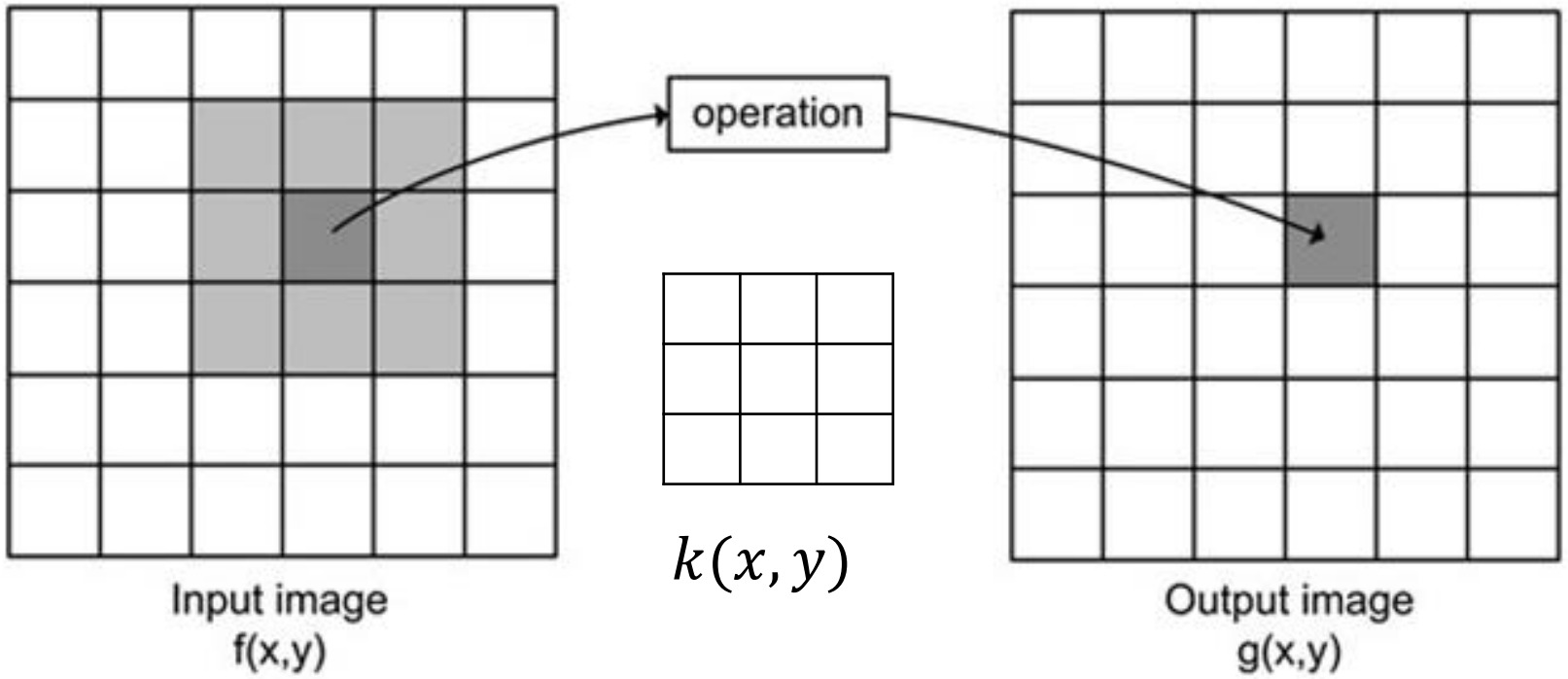
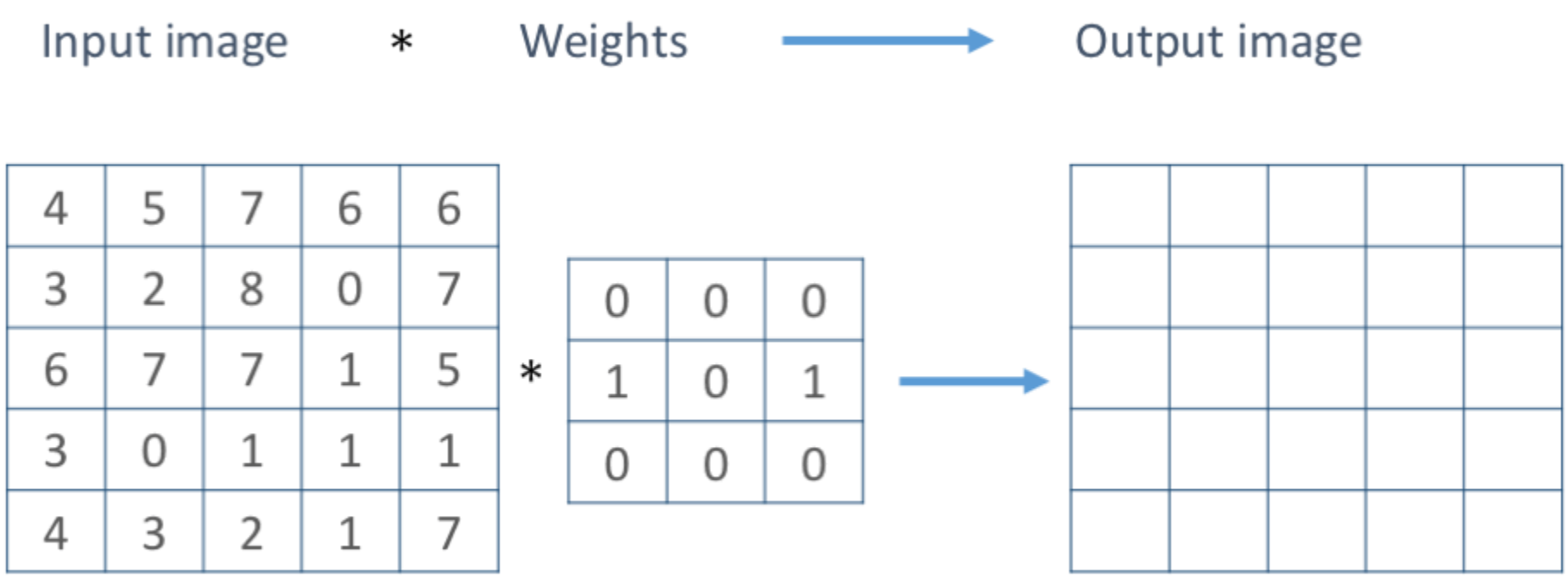


Image filtering: Convolution operator



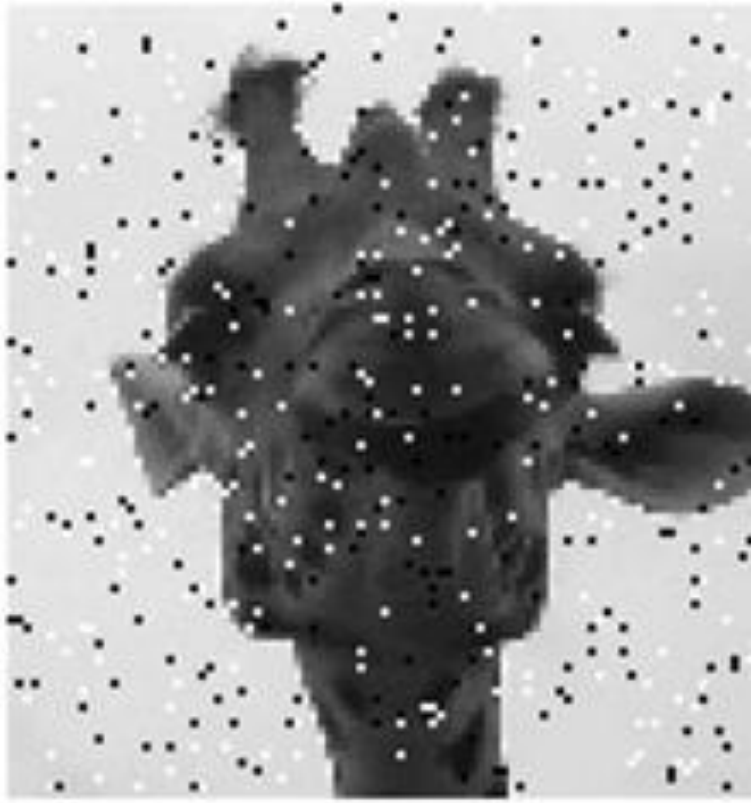
$$g(x,y) = \sum_v \sum_u k(u,v) f(x - u, y - v)$$

Image Credit: http://what-when-how.com/introduction-to-video-and-image-processing/neighborhood-processing-introduction-to-video-and-image-processing-part-1/



http://www.cs.virginia.edu/~vicente/recognition/animation.gif

Image filtering: e.g. Mean Filter



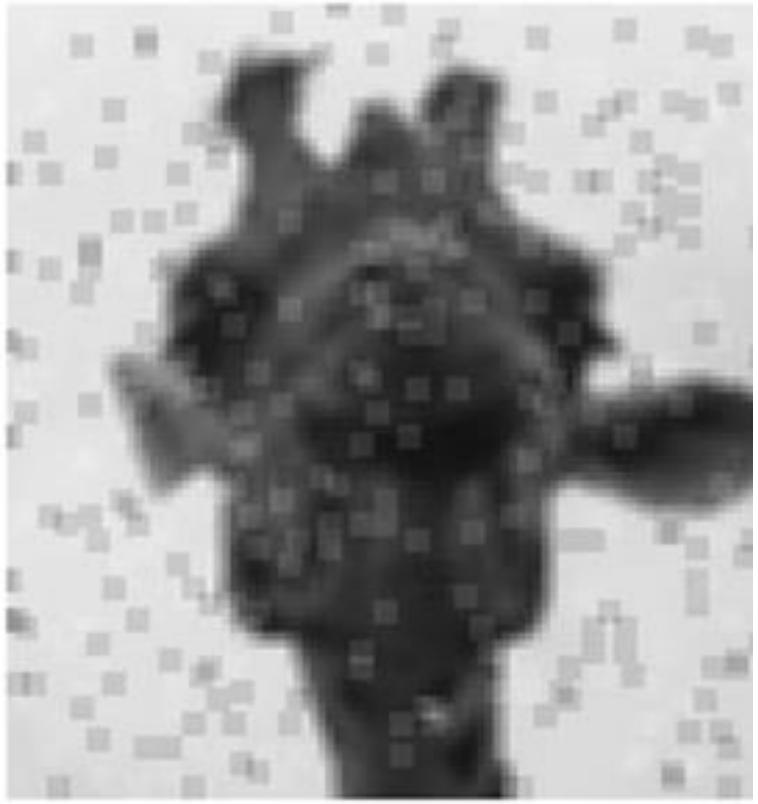
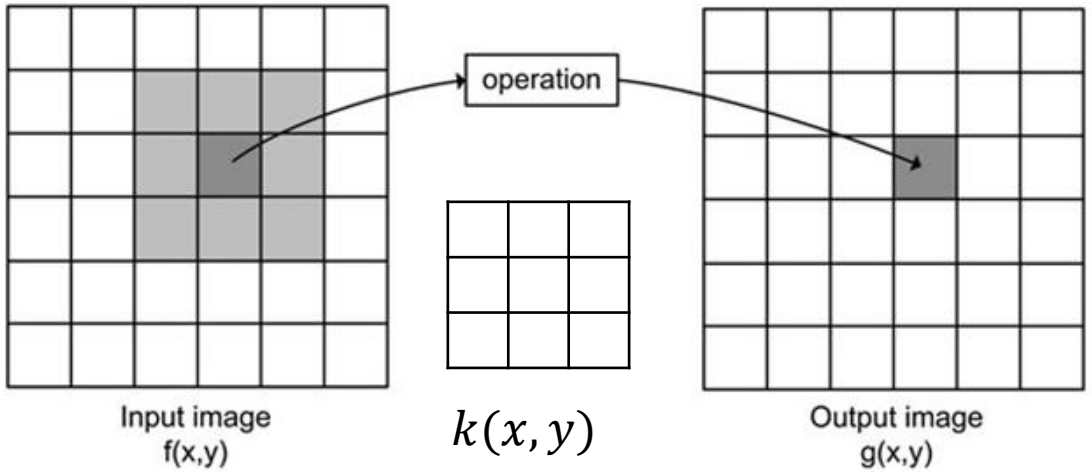
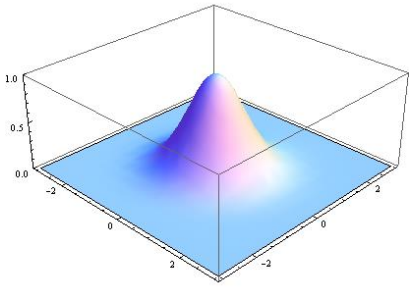


Image filtering: Convolution operator
Important filter: gaussian filter (gaussian blur)



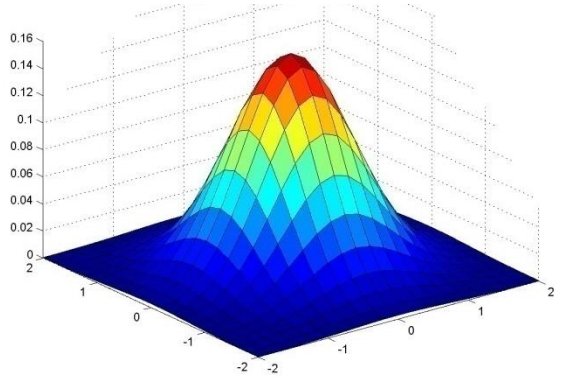
$k(x,y) =$

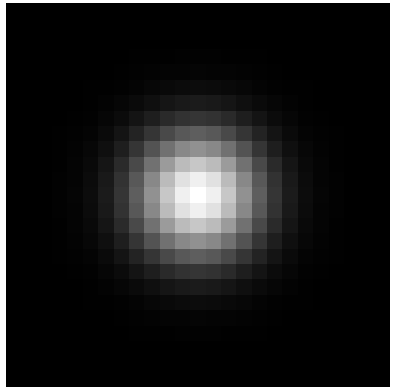
1/16	1/8	1/16
1/8	1/4	1/8
1/16	1/8	1/16



Important filter: Gaussian

- Weight contributions of neighboring pixels by nearness





0.003	0.013	0.022	0.013	0.003
0.013	0.059	0.097	0.059	0.013
0.022	0.097	0.159	0.097	0.022
0.013	0.059	0.097	0.059	0.013
0.003	0.013	0.022	0.013	0.003

5 x 5, $\sigma = 1$

$$G_\sigma = \frac{1}{2\pi\sigma^2} e^{-\frac{(x^2+y^2)}{2\sigma^2}}$$

Slide credit: Christopher Rasmussen

Image filtering: Convolution operator e.g. gaussian filter (gaussian blur)

Image Credit: http://what-when-how.com/introduction-to-video-and-image-processing/neighborhood-processing-introduction-to-video-and-image-processing-part-1/

Practice with linear filters

Original

0	0	0
0	1	0
0	0	0

?

Source: D. Lowe

Practice with linear filters

Original

0	0	0
0	1	0
0	0	0

Filtered
(no change)

Source: D. Lowe

Practice with linear filters

Original

0	0	0
0	0	1
0	0	0

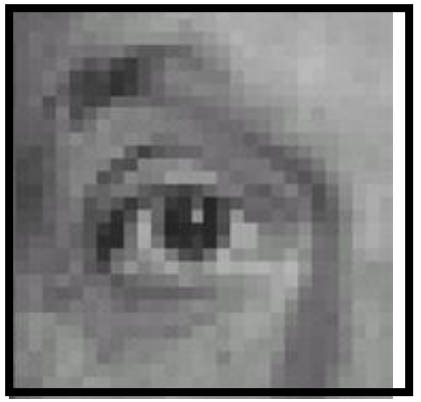
?

Source: D. Lowe

Practice with linear filters

Original

0	0	0
0	0	1
0	0	0



Shifted left
By 1 pixel

Source: D. Lowe

Practice with linear filters

Original

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 0 \end{bmatrix} - \frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad ?$$

(Note that filter sums to 1)

Source: D. Lowe

Practice with linear filters

Original

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 0 \end{bmatrix} - \frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Sharpening filter

- Accentuates differences with local average

Source: D. Lowe

Key properties of linear filters

Linearity:

$\texttt{imfilter(I, } f_1 + f_2\texttt{)} =$

$\texttt{imfilter(I,} f_1\texttt{)} + \texttt{imfilter(I,} f_2\texttt{)}$

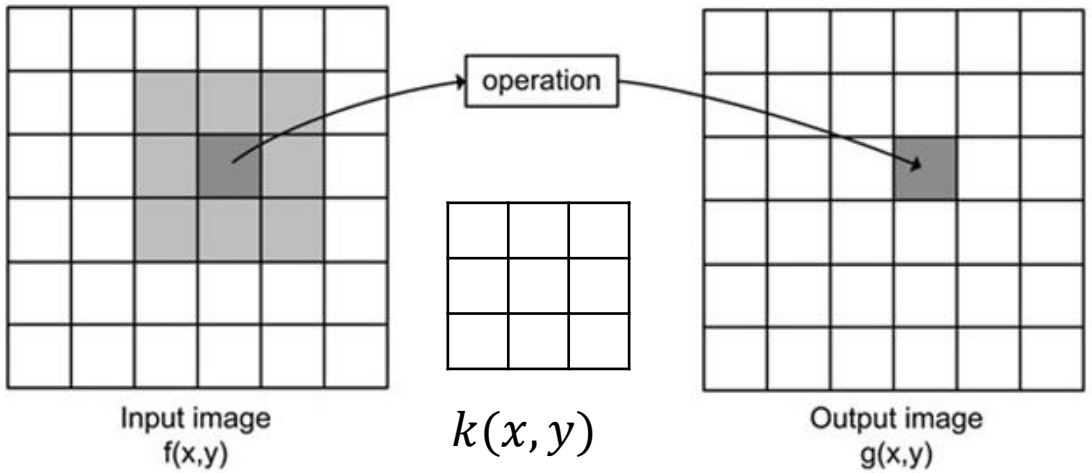
Shift invariance: same behavior regardless of pixel location

`imfilter(I,shift(f)) = shift(imfilter(I,f))`

Any linear, shift-invariant operator can be represented as a convolution

Source: S. Lazebnik

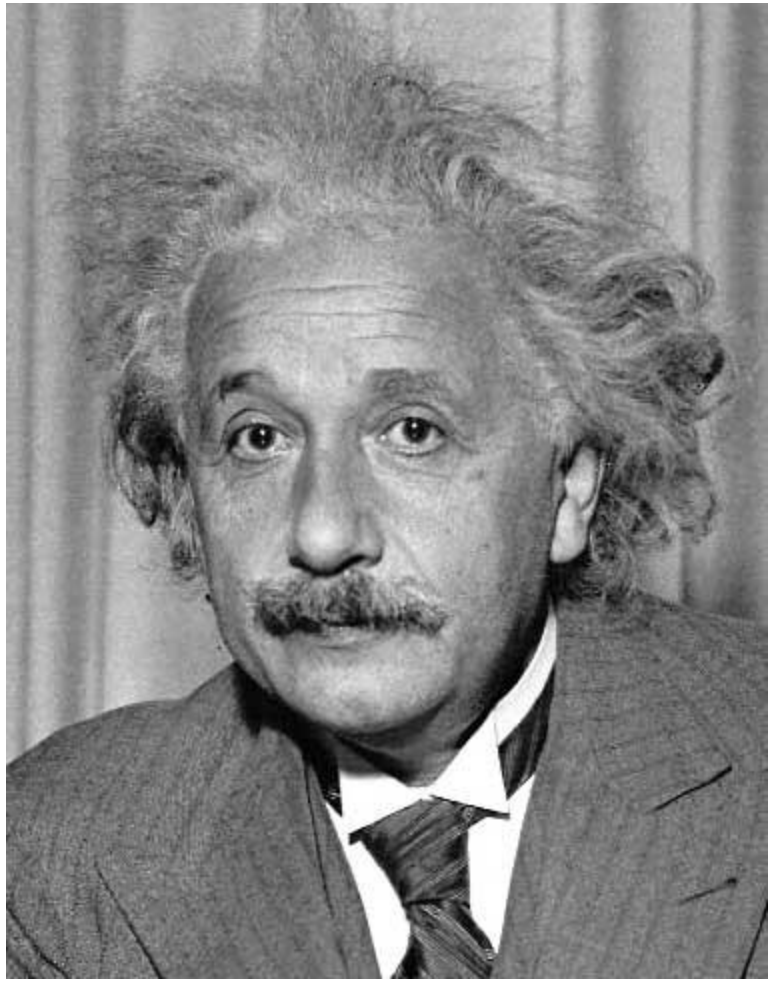
Image filtering: Convolution operator
Important Filter: Sobel operator



$k(x,y) =$

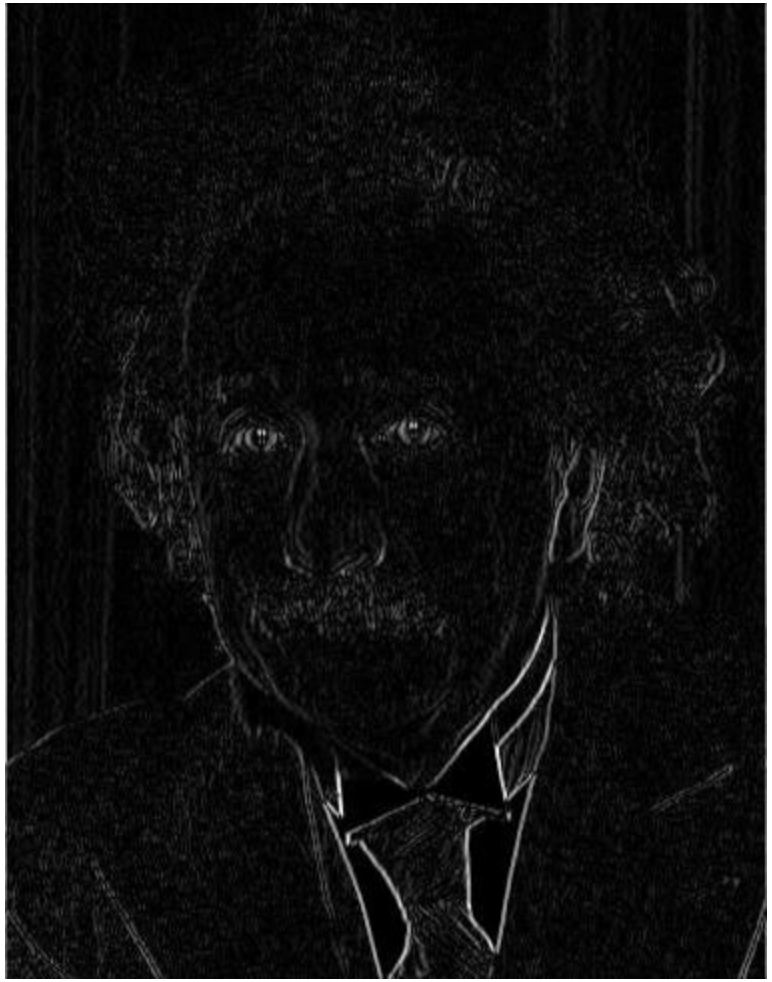
1	0	-1
2	0	-2
1	0	-1

Image Credit: http://what-when-how.com/introduction-to-video-and-image-processing/neighborhood-processing-introduction-to-video-and-image-processing-part-1/



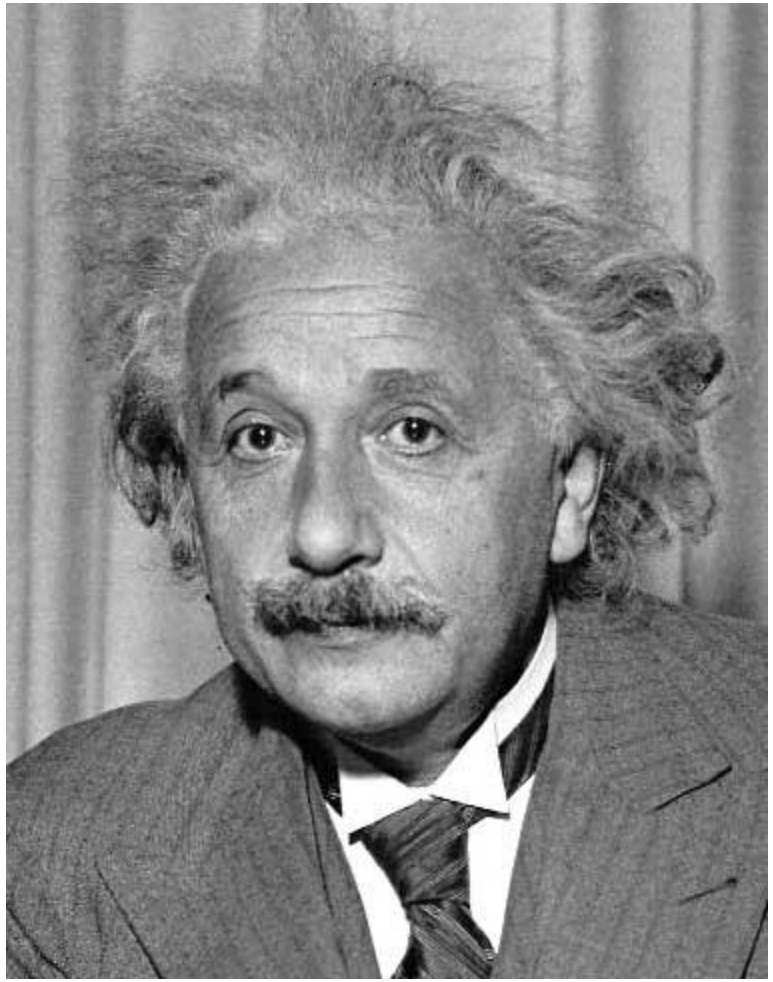
1	0	-1
2	0	-2
1	0	-1

Sobel



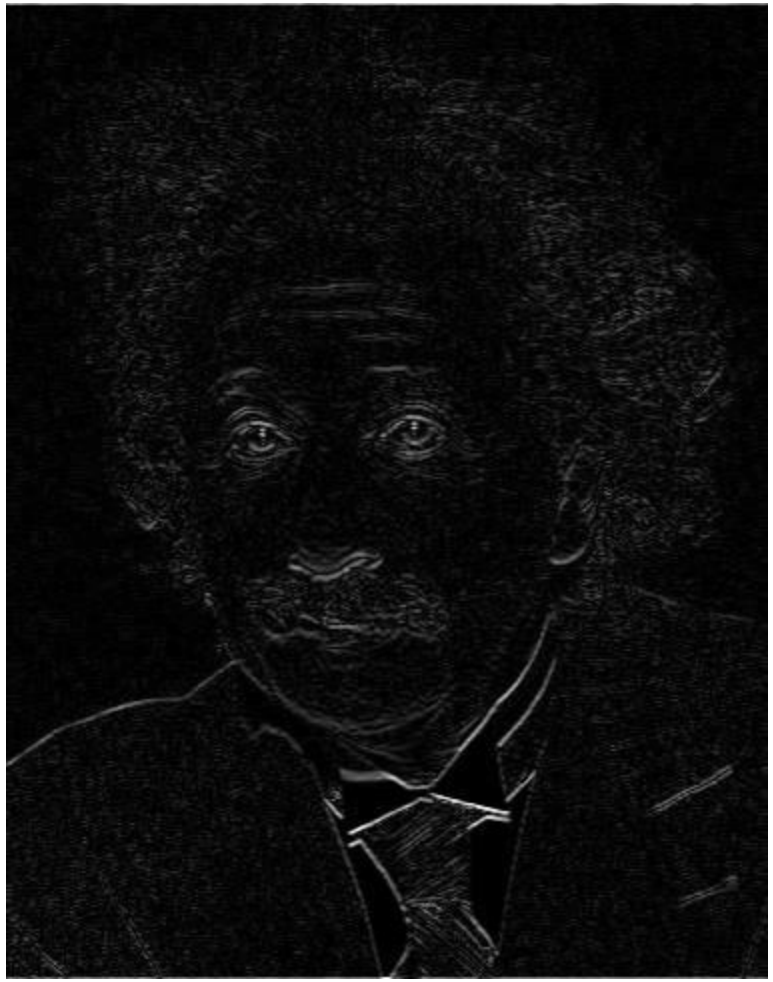
Vertical Edge
(absolute value)

Slide by James Hays



1	2	1
0	0	0
-1	-2	-1

Sobel



Horizontal Edge
(absolute value)

Slide by James Hays

Sobel operators are equivalent to 2D partial derivatives of the image

- Vertical sobel operator – Partial derivative in X (width)
- Horizontal sobel operator – Partial derivative in Y (height)
- Can compute magnitude and phase at each location
- Useful for detecting edges

Sobel filters are (approximate) partial derivatives of the image

Let $f(x, y)$ be your input image, then the partial derivative is:

$$\frac{\partial f(x,y)}{\partial x} = \lim_{h \to 0} \frac{f(x+h, y) - f(x, y)}{h}$$

Also:

$$\frac{\partial f(x,y)}{\partial x} = \lim_{h \to 0} \frac{f(x+h, y) - f(x-h, y)}{2h}$$

But digital images are not continuous, they are discrete

Let $f[x, y]$ be your input image, then the partial derivative is:

$$\Delta_x f[x, y] = f[x + 1, y] - f[x, y]$$

Also: $$\Delta_x f[x, y] = f[x + 1, y] - f[x - 1, y]$$

But digital images are not continuous, they are discrete

Let $f[x, y]$ be your input image, then the partial derivative is:

$$\Delta_x f[x, y] = f[x + 1, y] - f[x, y]$$

k(x, y) =

-1	1

Also: $\Delta_x f[x, y] = f[x + 1, y] - f[x - 1, y]$

k(x, y) =

-1	0	1

Sobel Operators Smooth in Y and then Differentiate in X

$k(x, y) =$

1
2
1

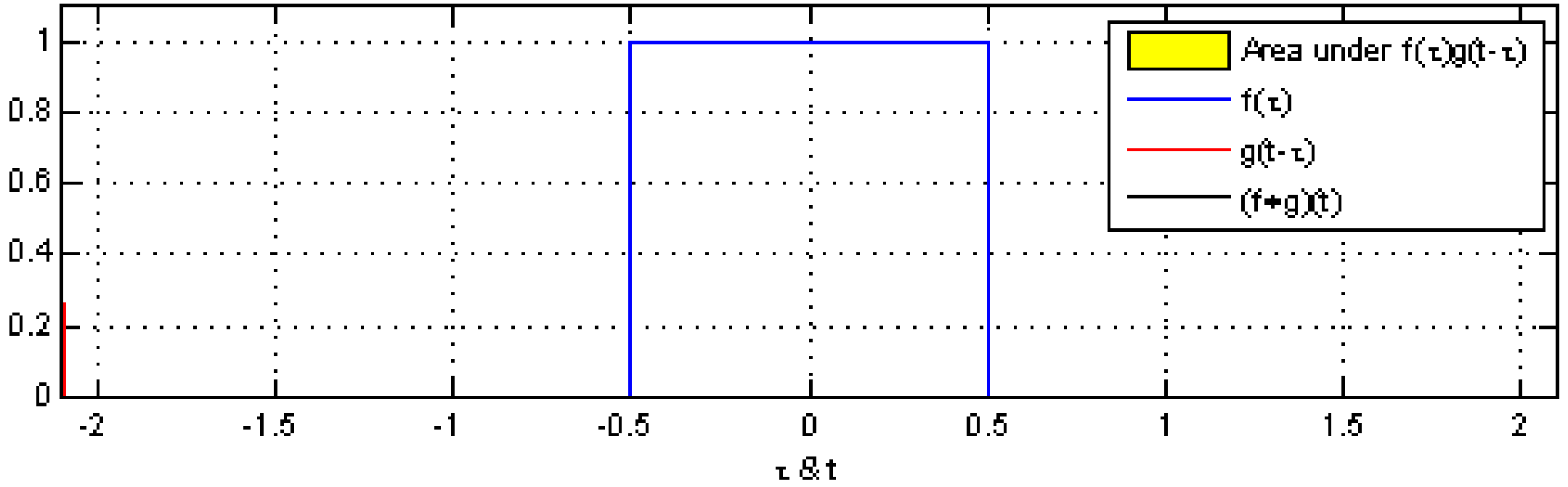
$*$

1	0	-1

$=$

1	0	-1
2	0	-2
1	0	-1

Similarly to differentiate in Y



From Wikipedia

Gabor filters

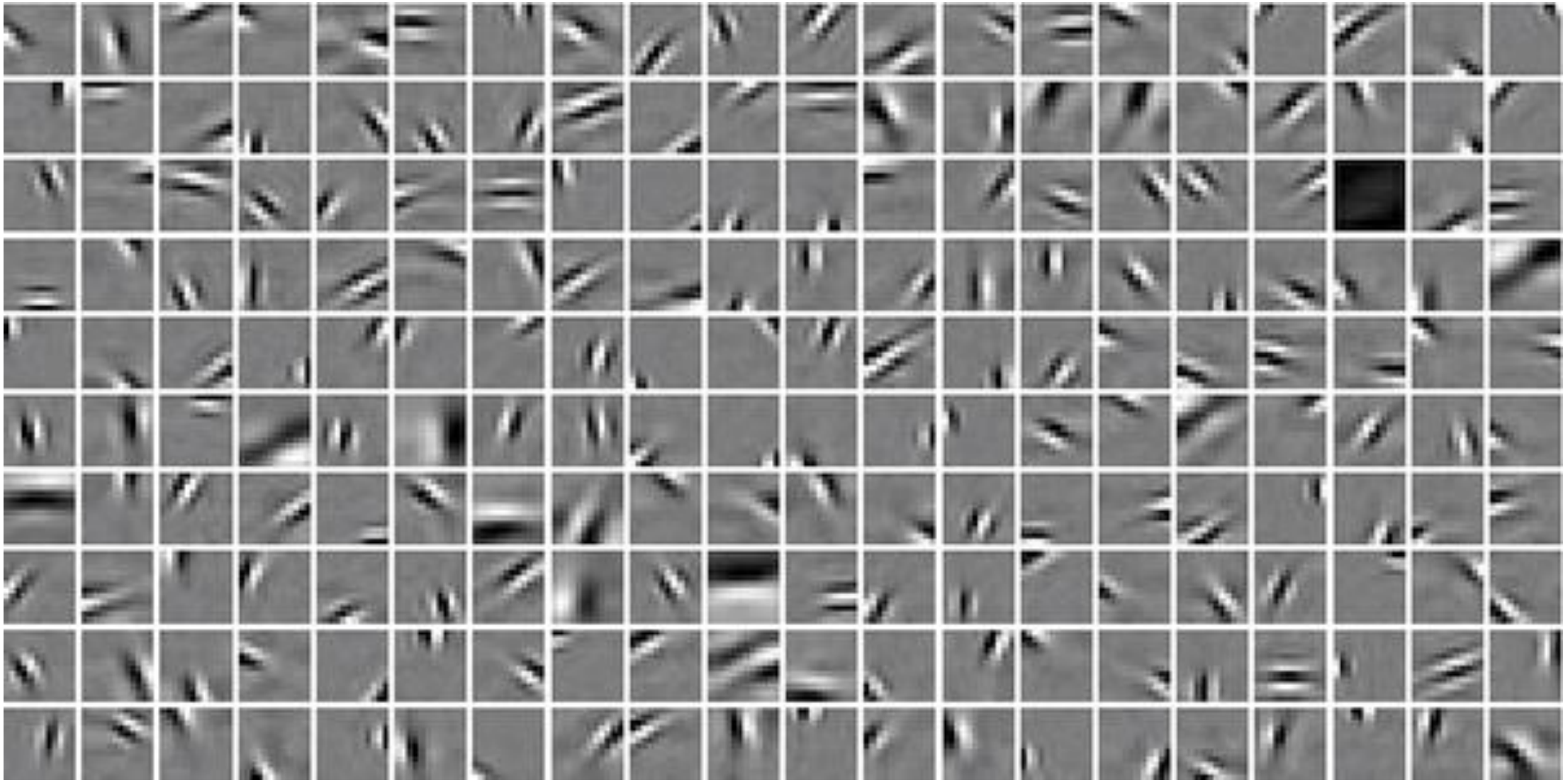
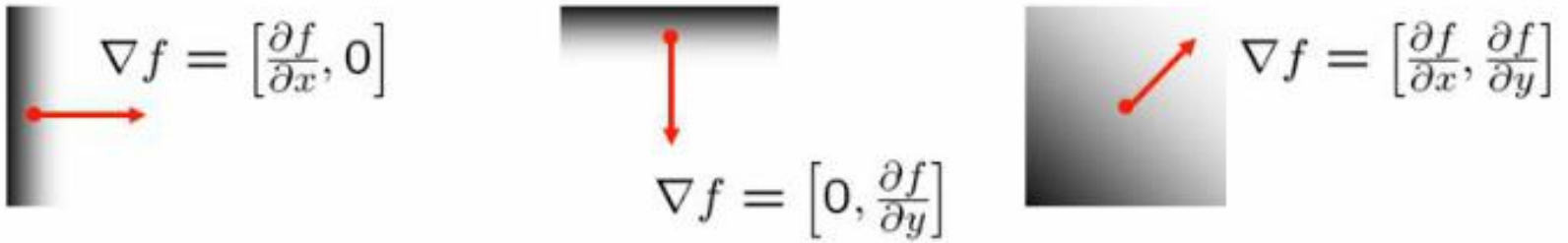


Image gradient in a nuttshell

The gradient of an image:

$$\nabla f = \left[\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\right]$$

The gradient points in the direction of most rapid change in intensity



The gradient direction (orientation of edge normal) is given by:

$$\theta = \tan^{-1}\left(\frac{\partial f}{\partial y} / \frac{\partial f}{\partial x}\right)$$

The *edge strength* is given by the **gradient magnitude**

$$\|\nabla f\| = \sqrt{\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2}$$

Gradient histogram

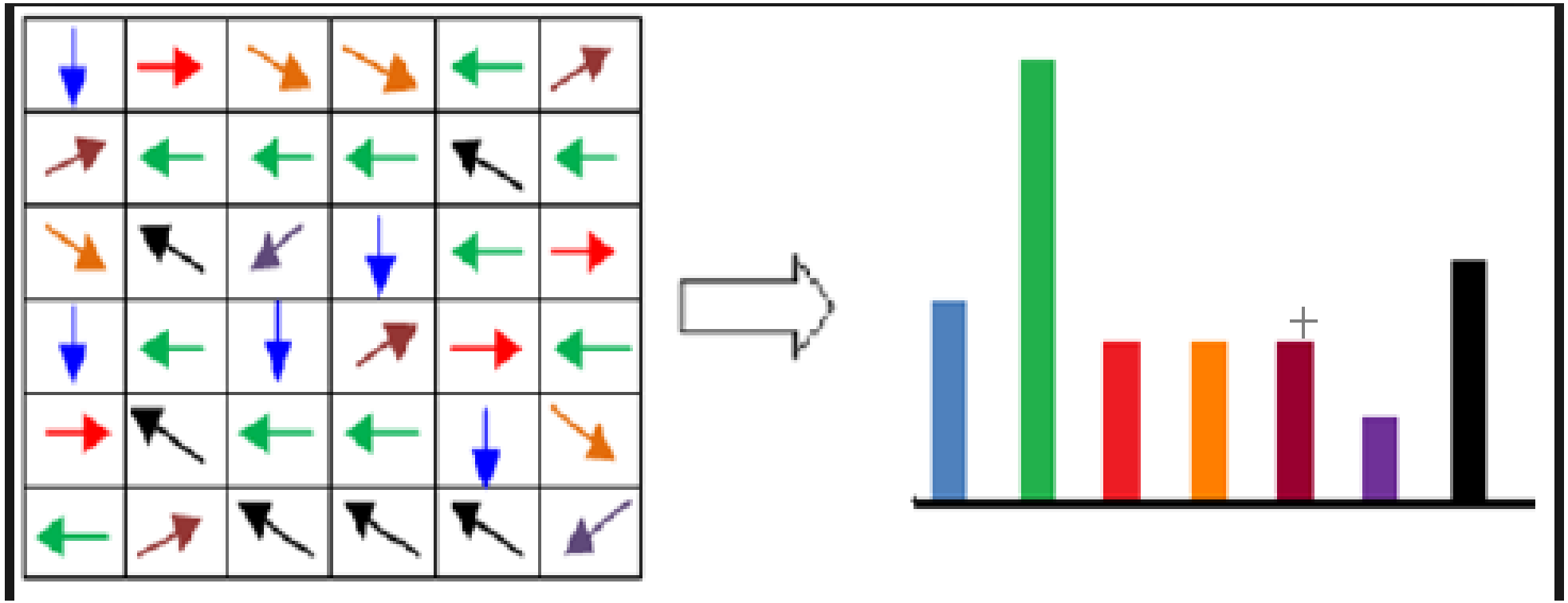
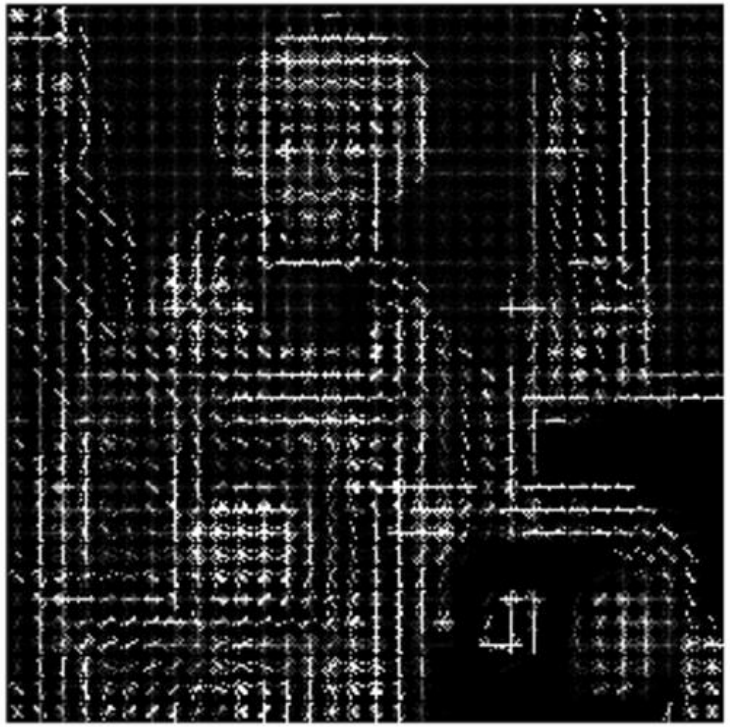


Image Features: HoG

Input image

Histogram of Oriented Gradients



Paper by Navneet Dalal & Bill Triggs presented at CVPR 2005 for detecting people.

Image Features: HoG

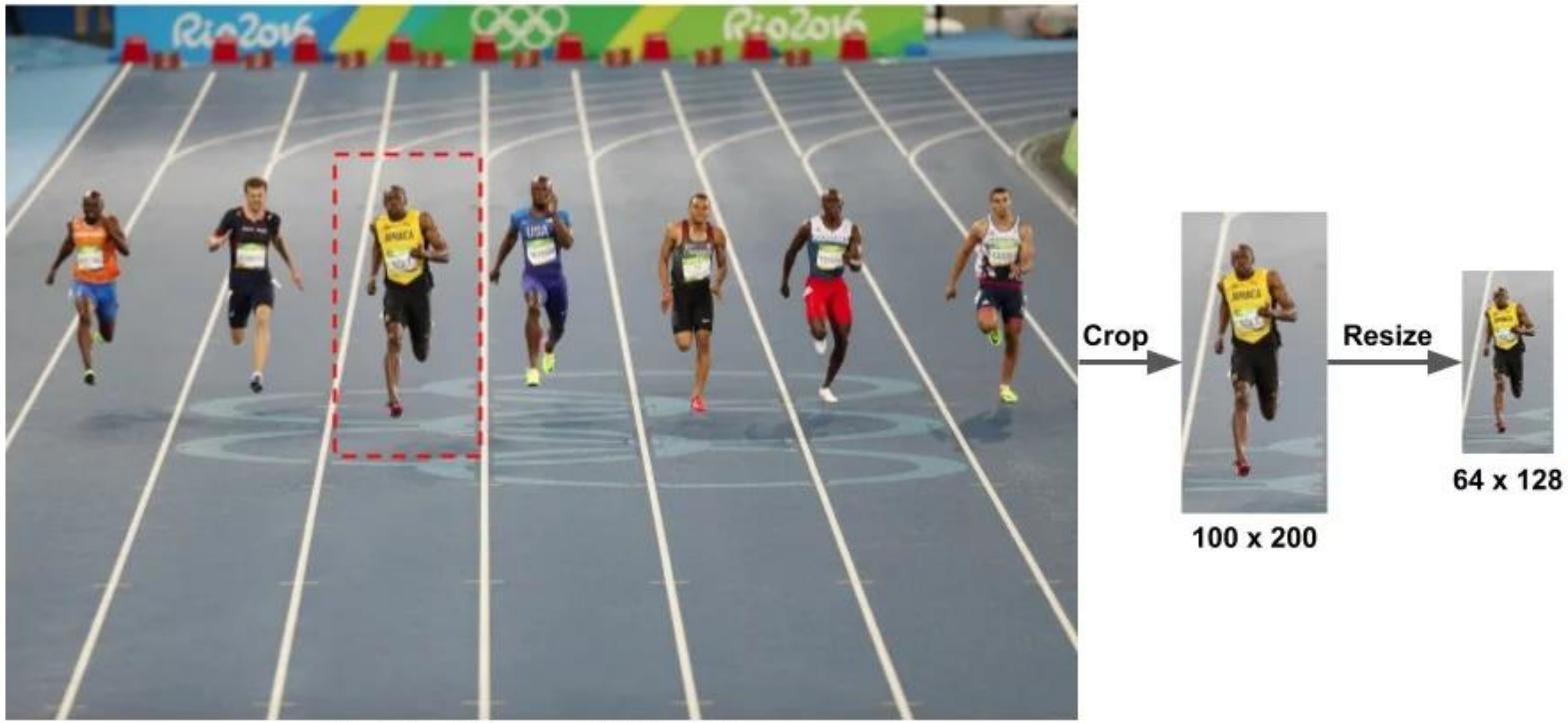


Image Features: HoG

Compute gradients

$*$ | -1 | 0 | 1 |

-1
0
1

$\rightarrow$

I_x

I_y

$\sqrt{I_x^2 + I_y^2}$

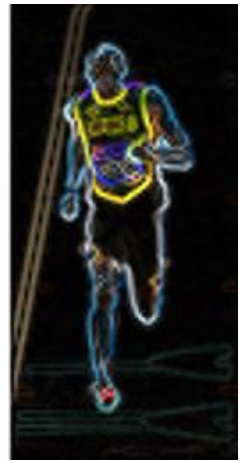
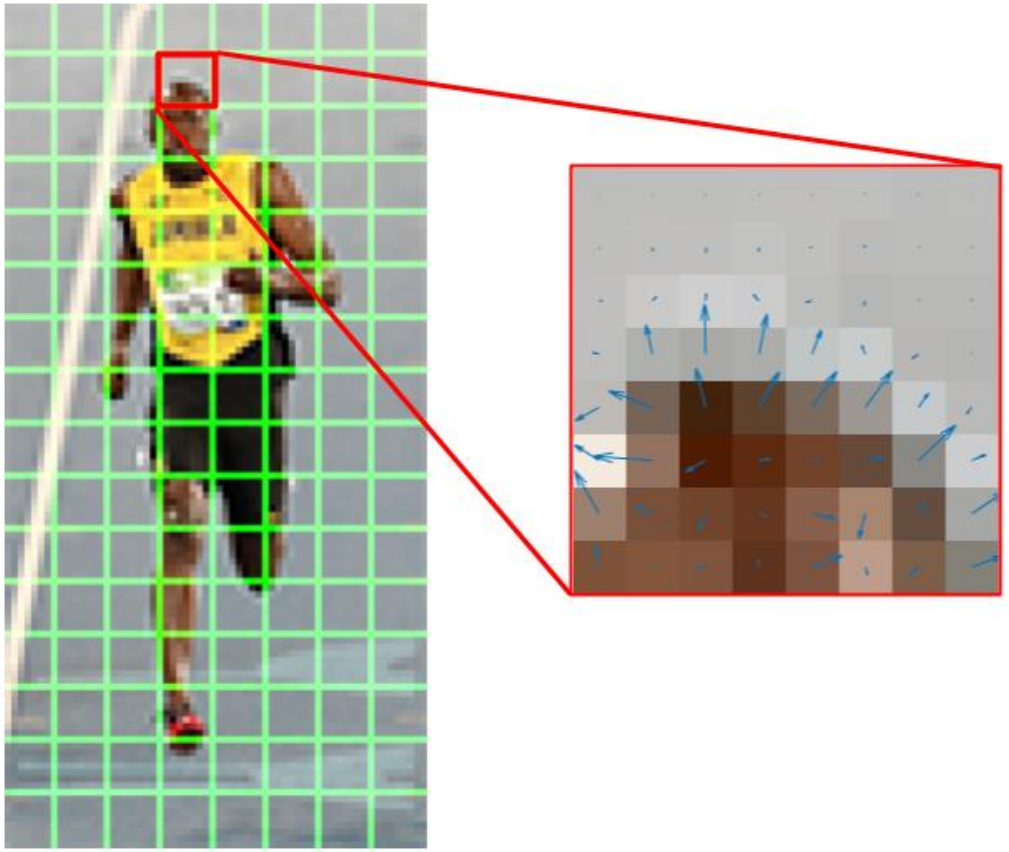


Image Features: HoG

We will aggregate gradient magnitude and directions in 8x8 pixel regions



2	3	4	4	3	4	2	2
5	11	17	13	7	9	3	4
11	21	23	27	22	17	4	6
23	99	165	135	85	32	26	2
91	155	133	136	144	152	57	28
98	196	76	38	26	60	170	51
165	60	60	27	77	85	43	136
71	13	34	23	108	27	48	110

Gradient Magnitude

80	36	5	10	0	64	90	73
37	9	9	179	78	27	169	166
87	136	173	39	102	163	152	176
76	13	1	168	159	22	125	143
120	70	14	150	145	144	145	143
58	86	119	98	100	101	133	113
30	65	157	75	78	165	145	124
11	170	91	4	110	17	133	110

Gradient Direction

Image Features: HoG

Compute a histogram with 9 bins for angles from 0 to 180

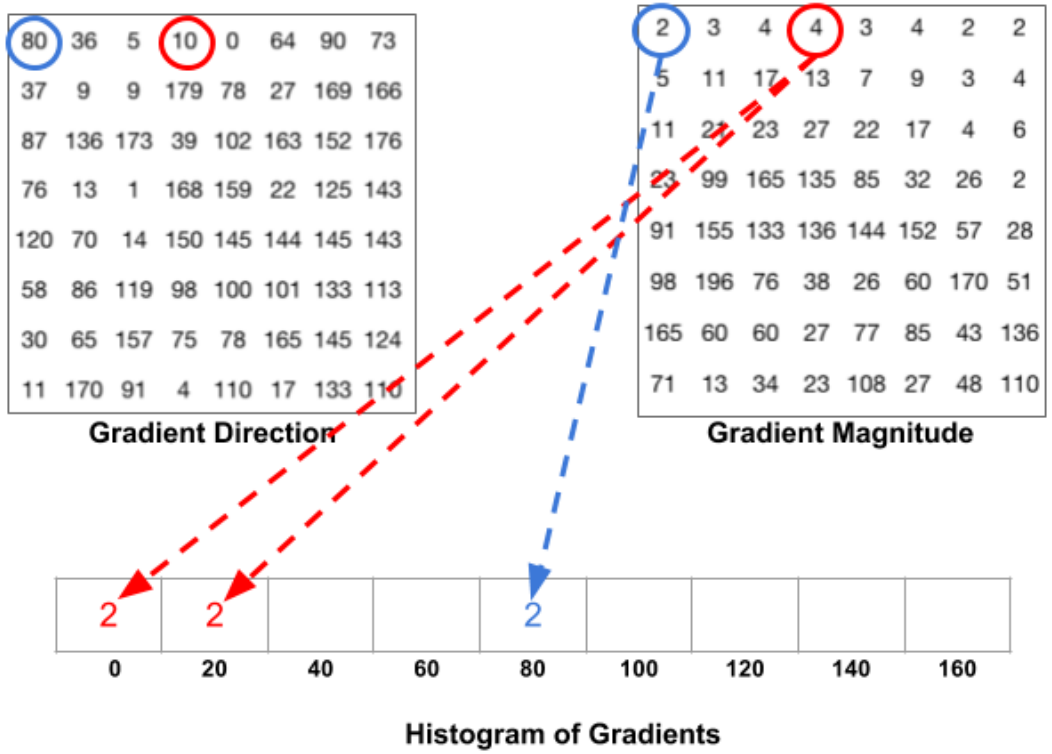


Image Features: HoG

Normalize histograms with respect to histograms of adjacent neighbors.

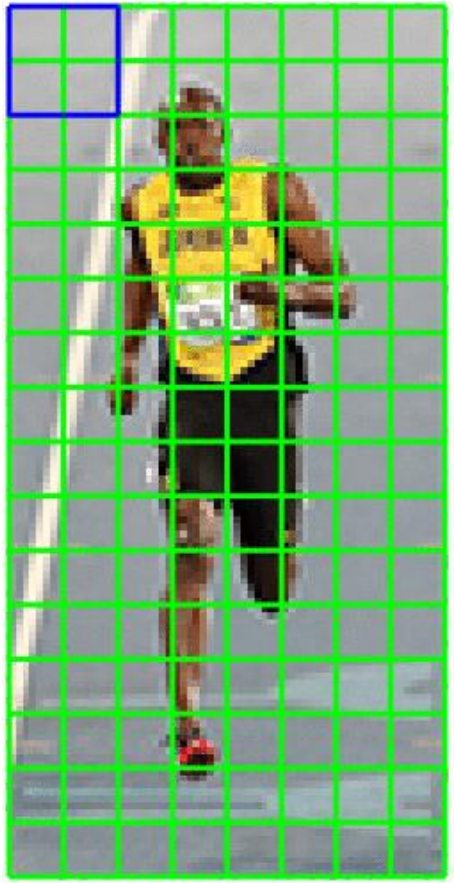


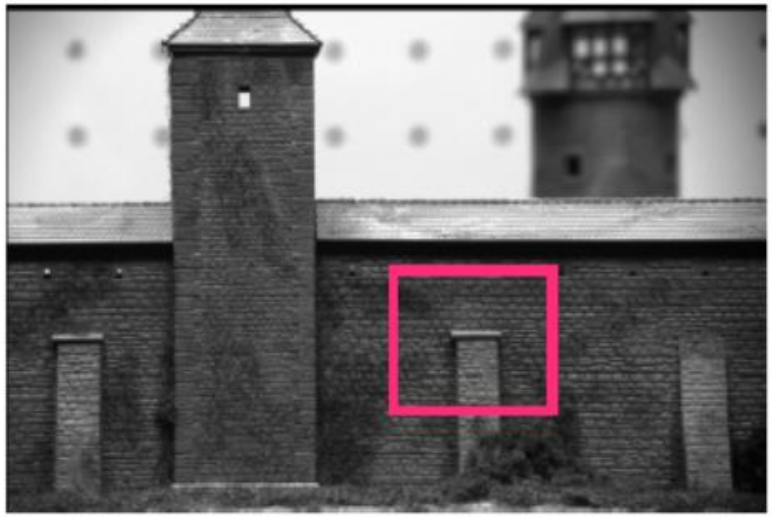
Image Features: HoG

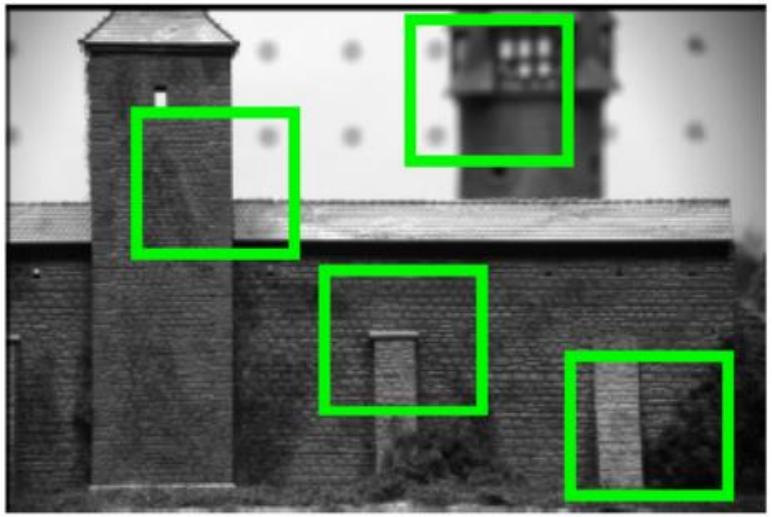
Image (or image region) represented by a vector containing all the histograms.

In this case how long is that vector?

Interest points - motivation

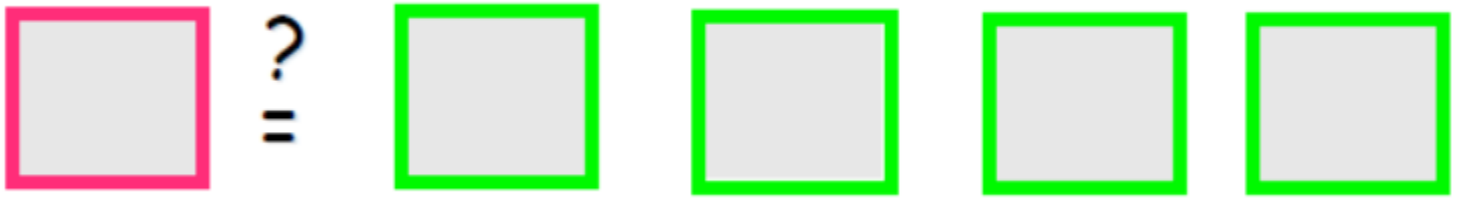
Elements to be matched are image patches of fixed size



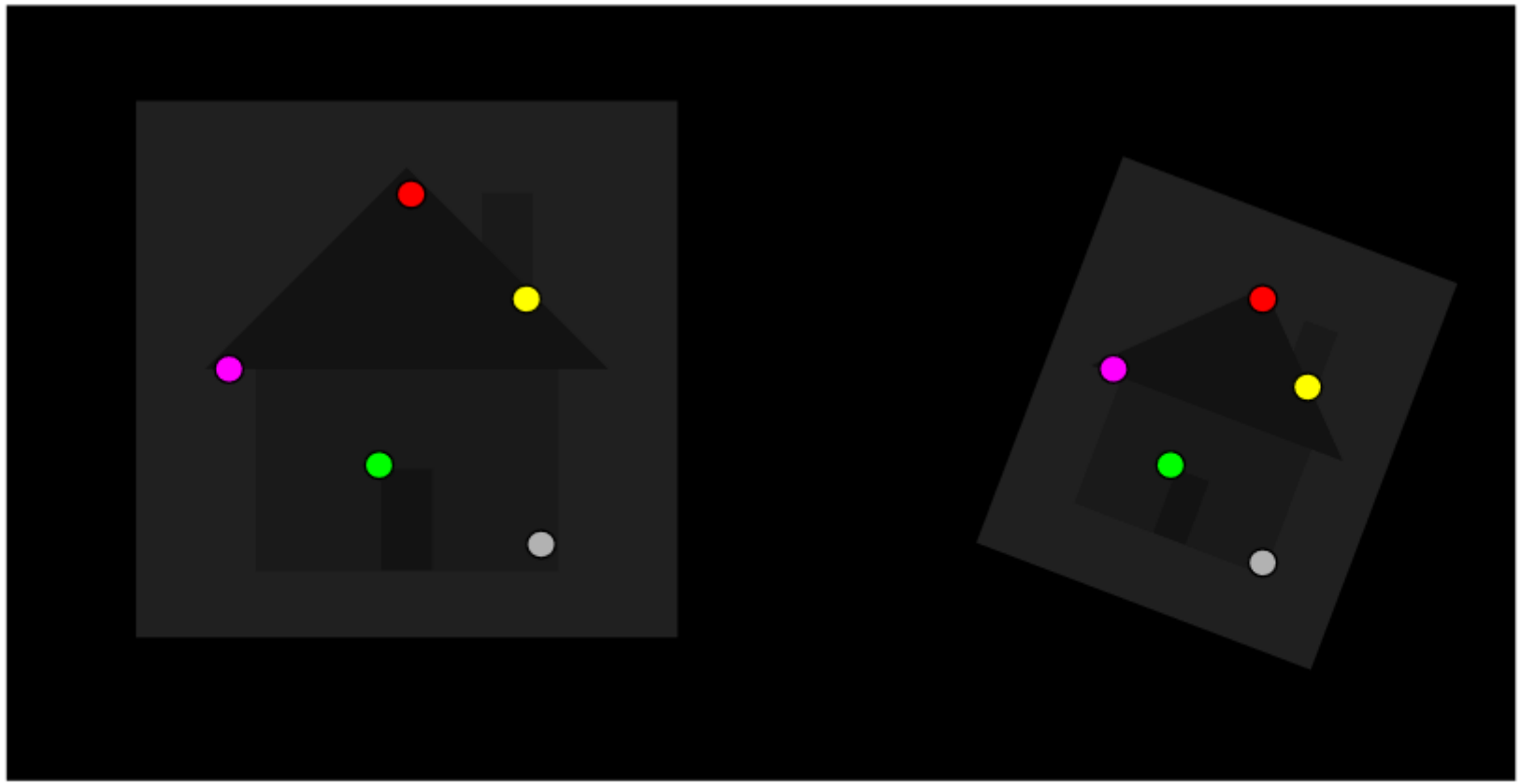


Task: find the best (most similar) patch in a second image

Interest points - motivation

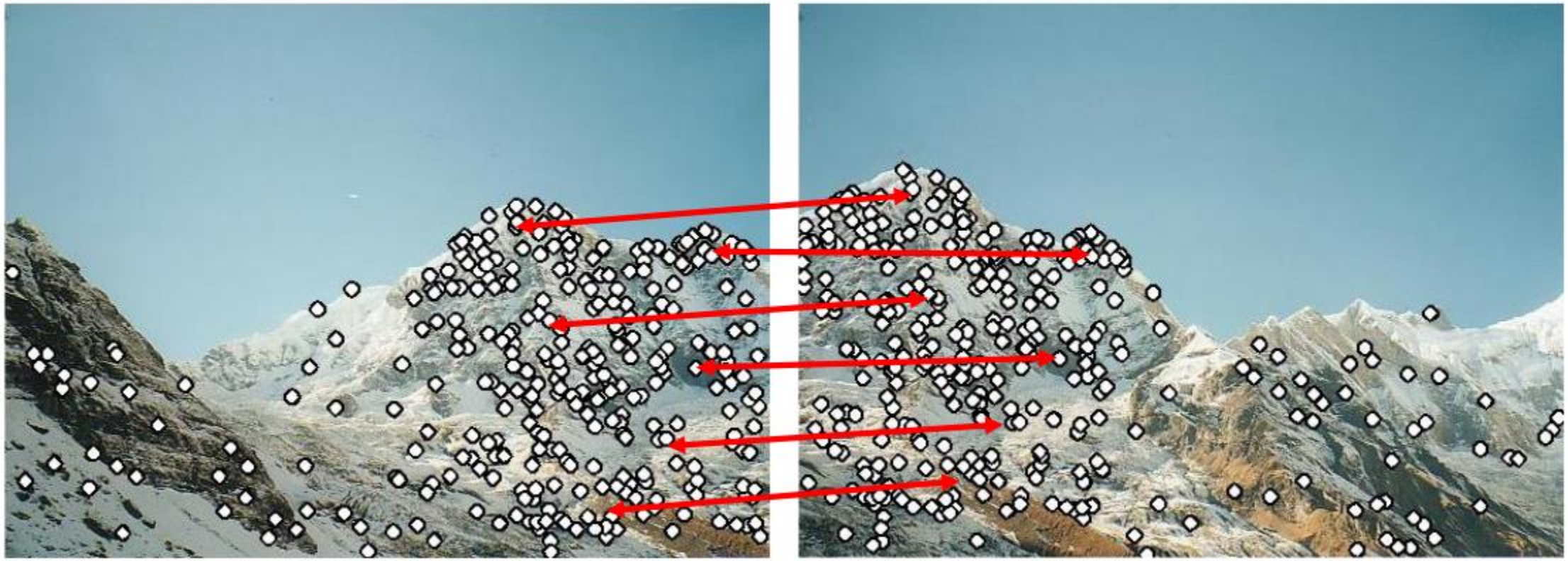


Corresponding points (or features) between images enable the estimation of parameters describing geometric transforms between the images.

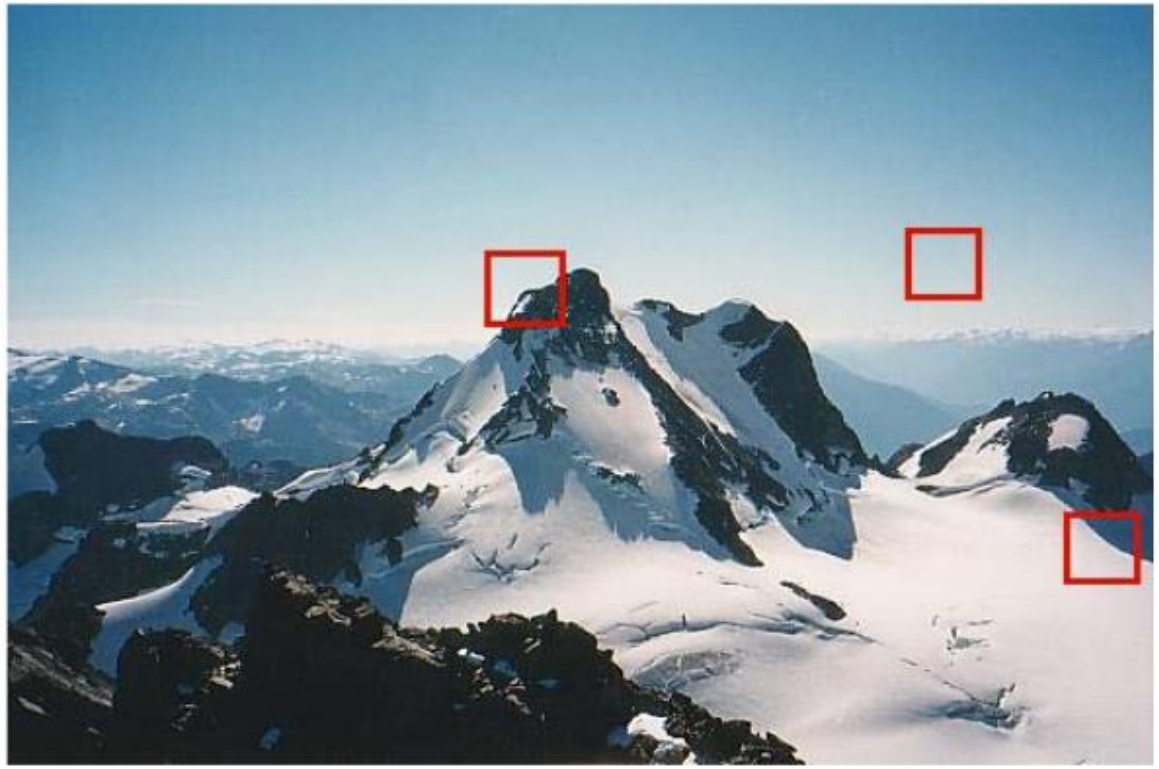


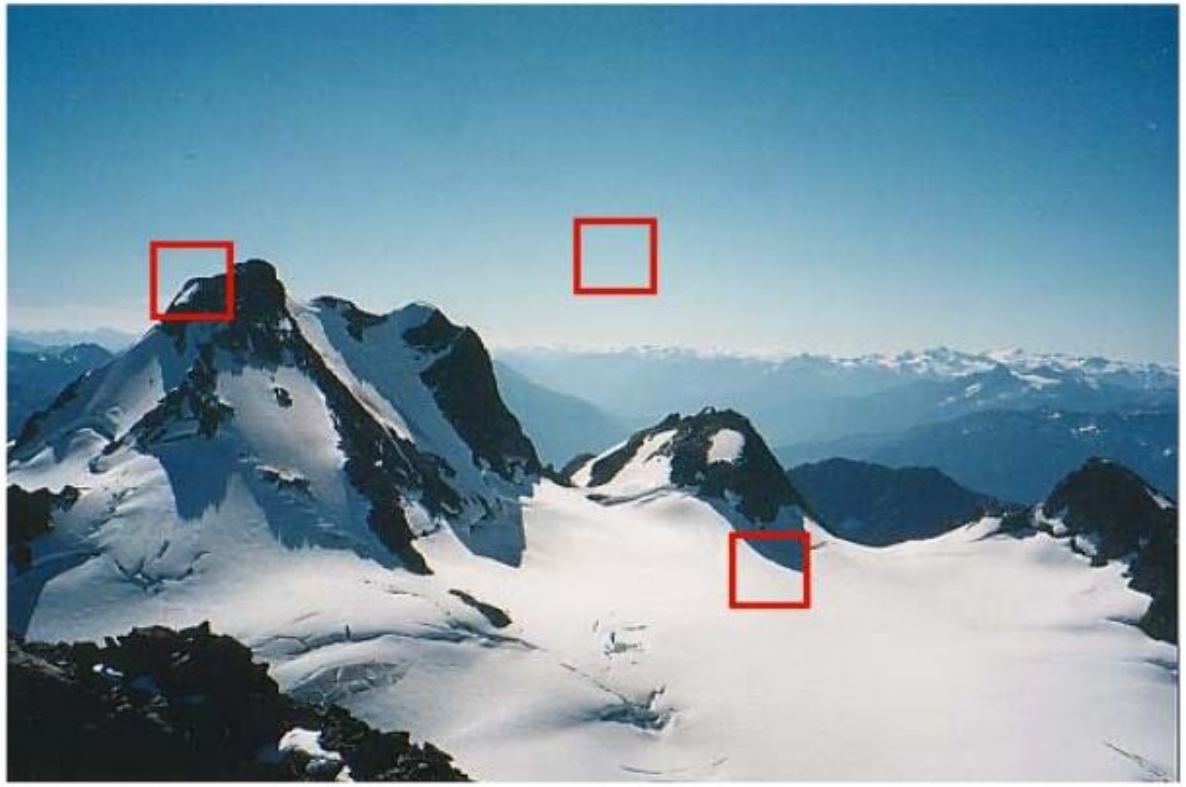
- How do we combine these two images?

Panorama stitching



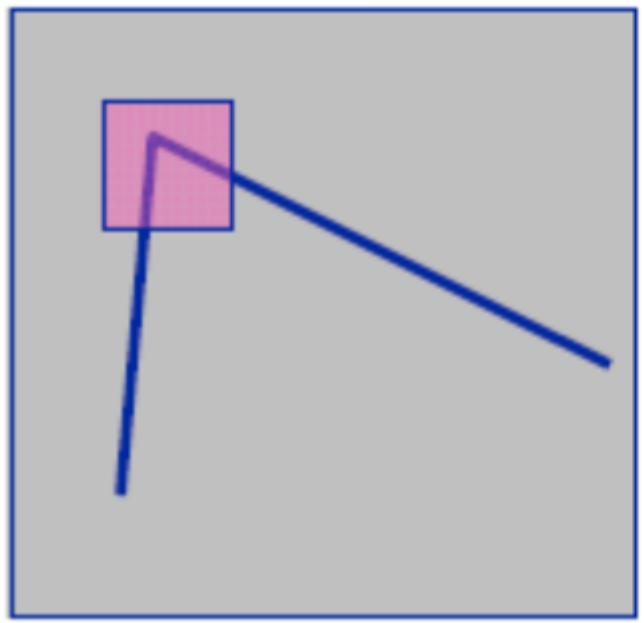
Corners

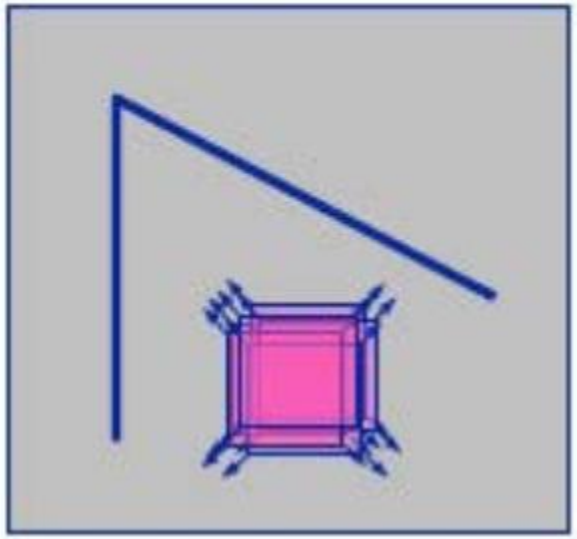


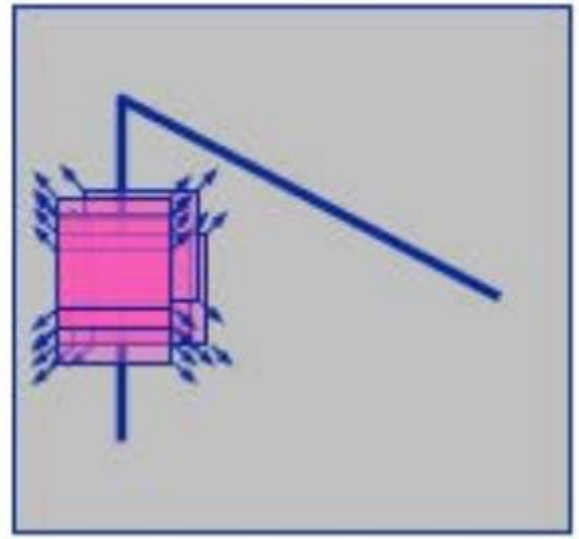


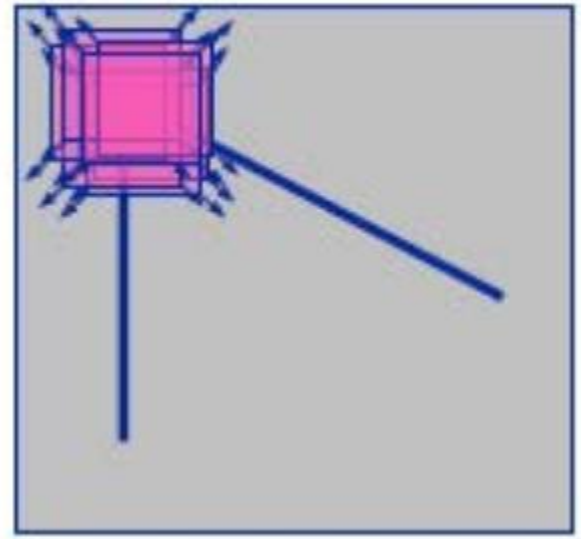
We should easily recognize the point by looking through a small window

Shifting a window in *any direction* should give *a large change* in response









"flat" region:
no change in
all directions

"edge":
no change along
the edge direction

"corner":
significant change
in all directions

Change of intensity for the shift $[u,v]$:

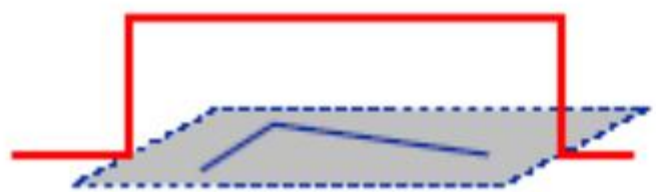
$$E(u,v) = \sum_{x,y} w(x,y)\left[I(x+u, y+v) - I(x,y)\right]^2$$

Window function

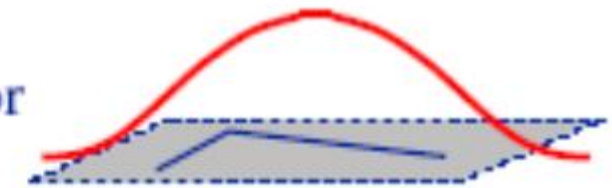
Shifted intensity

Intensity

Window function $w(x,y)$ =



or



$$E(u,v) = \sum_{x,y} w(x,y)[I(x+u,y+v) - I(x,y)]^2$$

$$E(u,v) \approx \sum_{x,y} [I(x,y) + uI_x + vI_y - I(x,y)]^2$$

$$E(u,v) \approx \sum_{x,y} u^2 {I_x}^2 + 2uvI_xI_y + v^2 {I_y}^2$$

$$E(u,v) \approx \begin{bmatrix} u & v \end{bmatrix} \left(\sum \begin{bmatrix} {I_x}^2 & I_xI_y \\ I_xI_y & {I_y}^2 \end{bmatrix} \right) \begin{bmatrix} u \\ v \end{bmatrix}$$

$$M = \sum w(x,y) \begin{bmatrix} {I_x}^2 & I_xI_y \\ I_xI_y & {I_y}^2 \end{bmatrix}$$

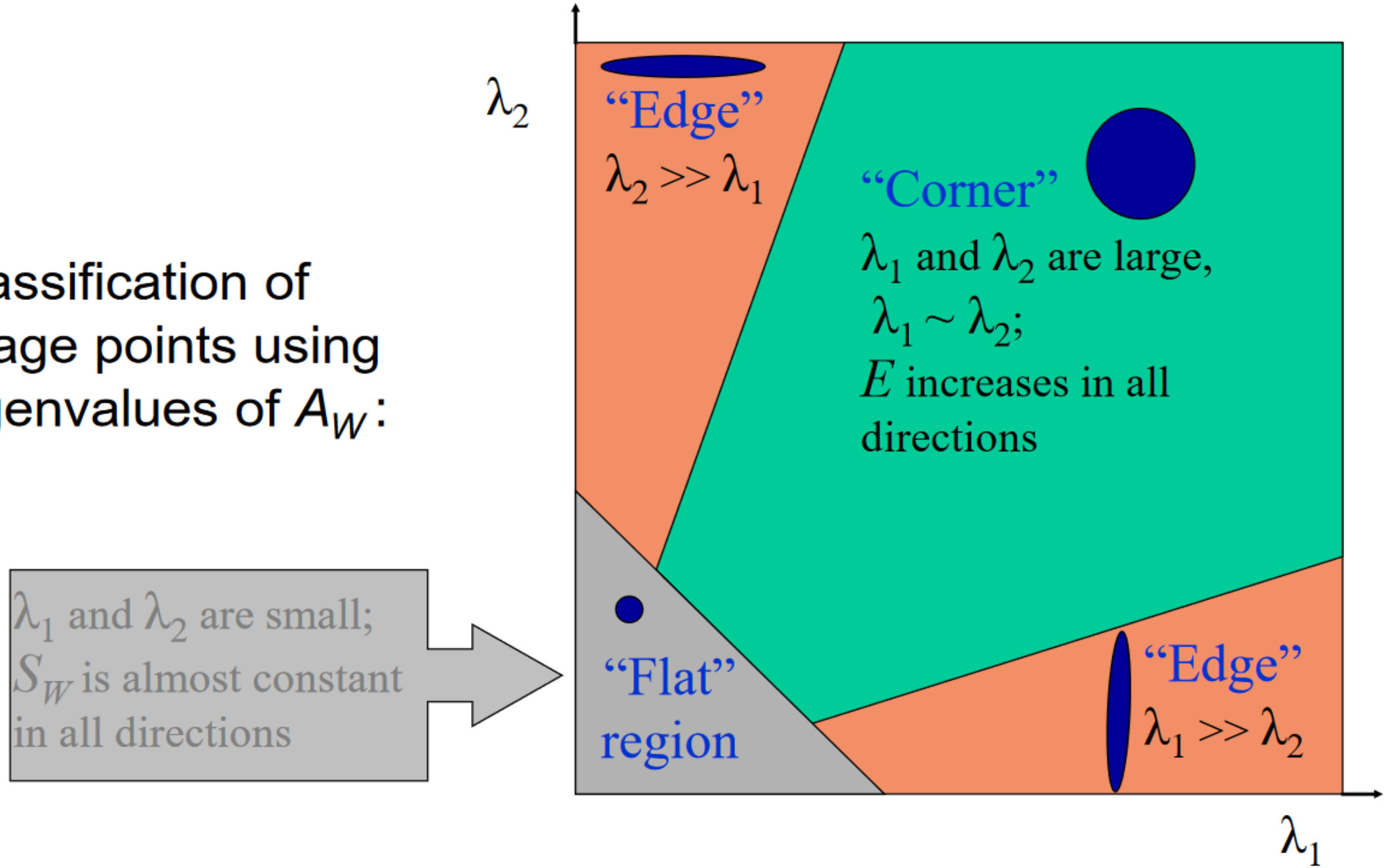
$$R = \det M - k(trace\, M)^2$$

$$\det M = \lambda_1 \lambda_2$$

$$trace\, M = \lambda_1 + \lambda_2$$

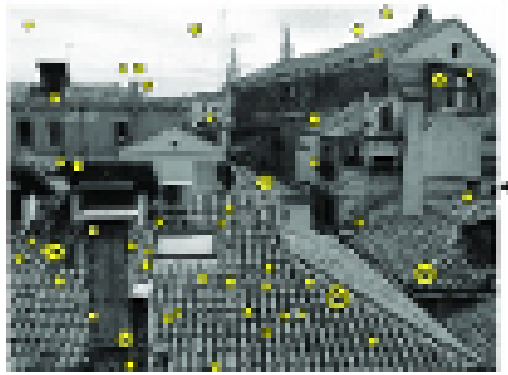
Cornerness

Classification of image points using eigenvalues of A_W:



The state of the art interest point detector - SIFT

Scale Space Construction → Difference of Gaussian → Extreme Point Extraction → Keypoint Localization → Orientation Assignment → Descriptor Generation



Won’t cover in the class – Pl refer to David Lowe paper for details

Given:

- positive training images containing an object class, and



- negative training images that don't



Classify:

- a test image as to whether it contains the object class or not

?

Visual words

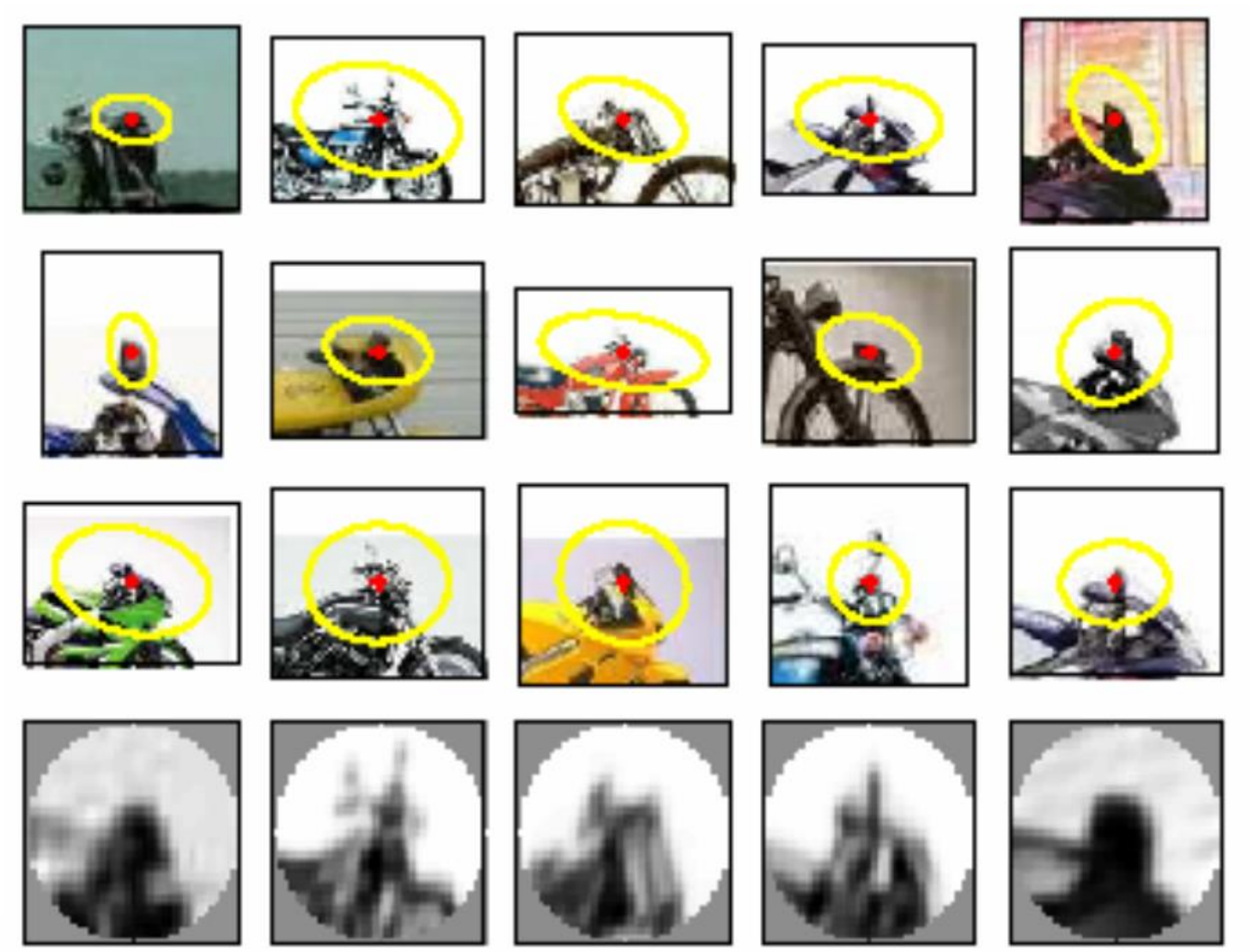
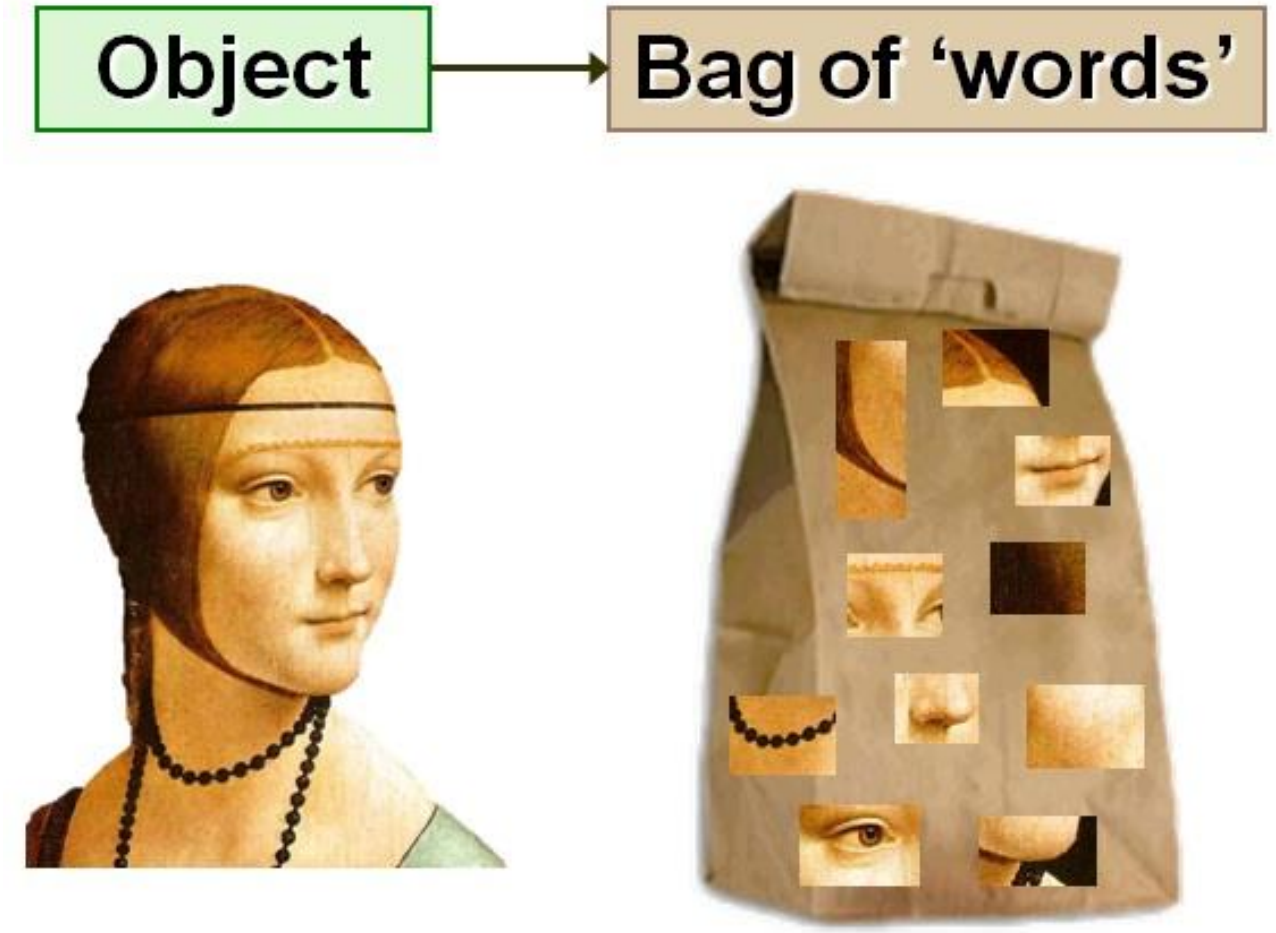


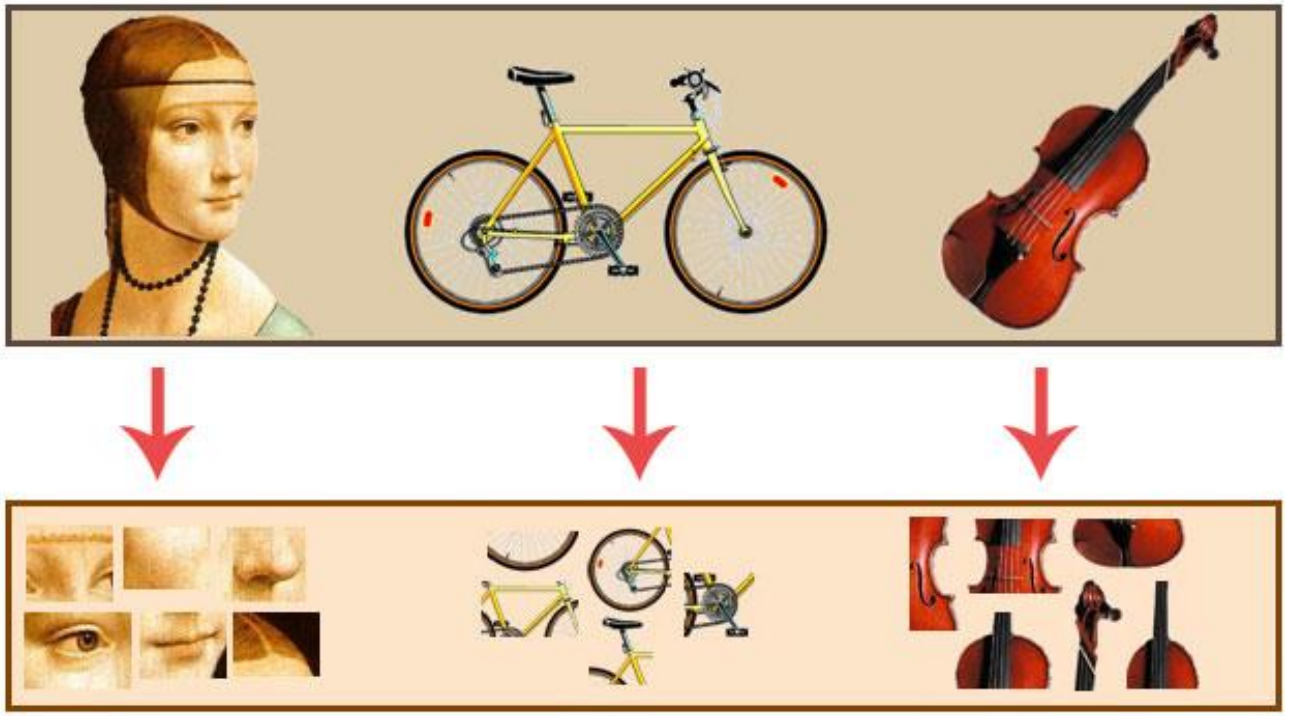
Image Features: Bag of (Visual) Words Representation



Bag of Features (read more in Gabriela Csurka's ECCV 2004 paper)

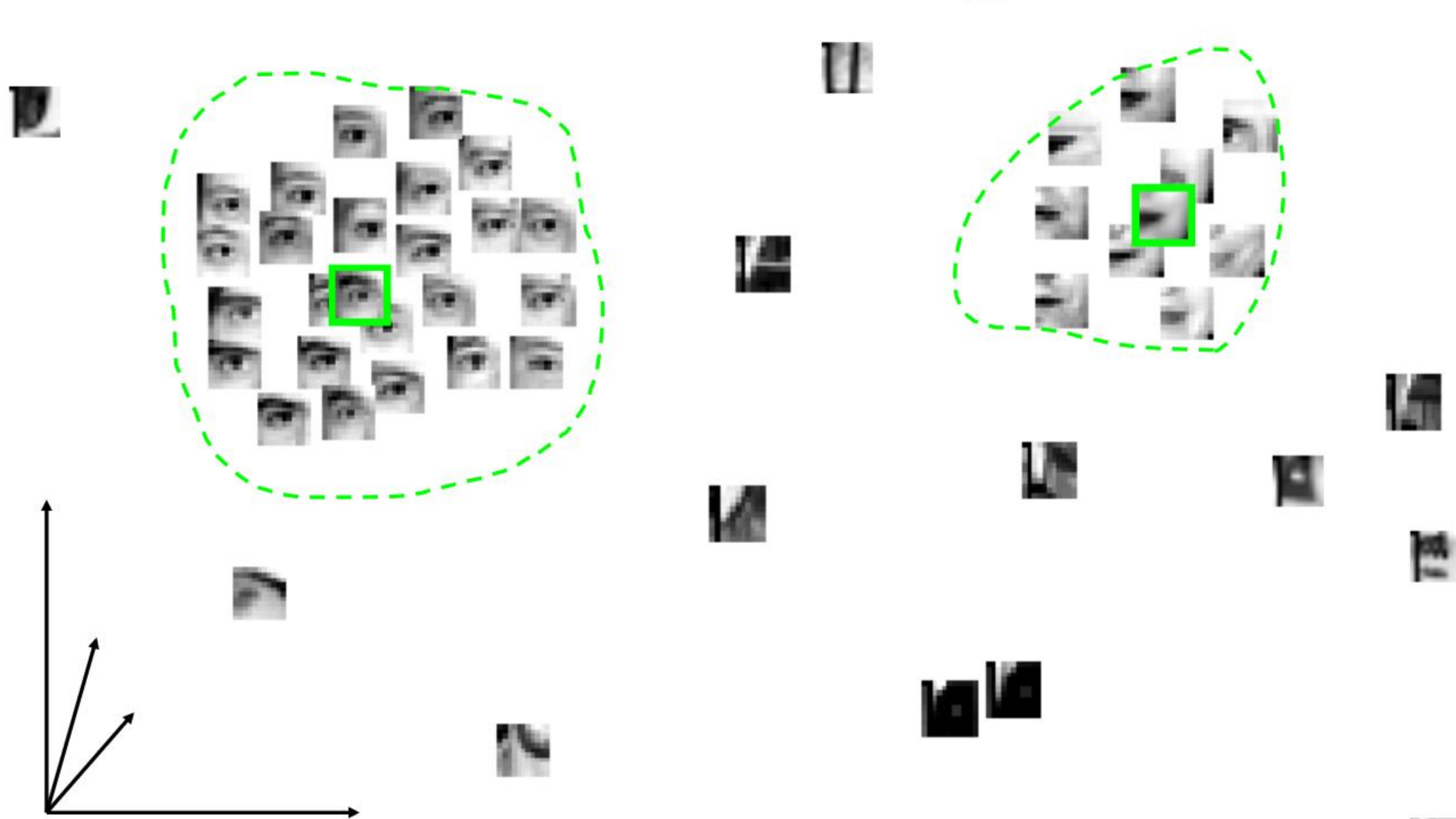
slide by Fei-fei Li

Extract SIFT Feature Descriptors

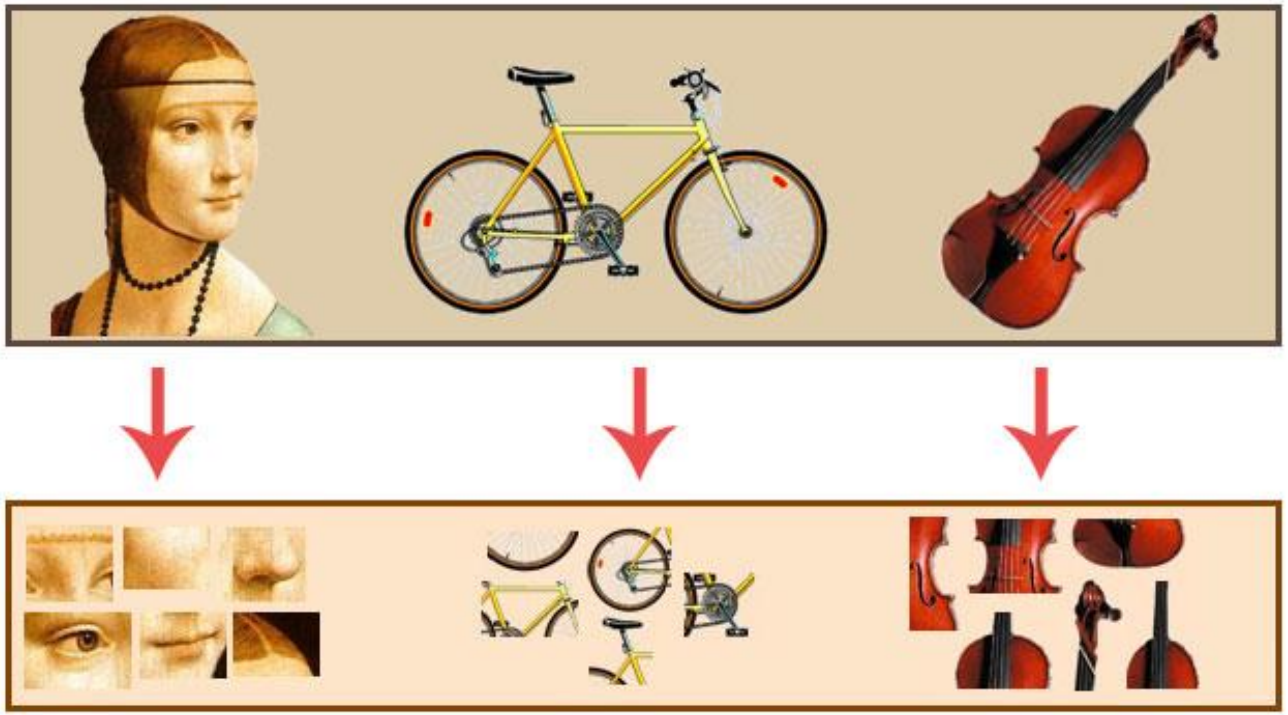


Intuition: sample a bunch of pieces (“words”) from various parts of the image. Images of the same object are more likely to share similar pieces

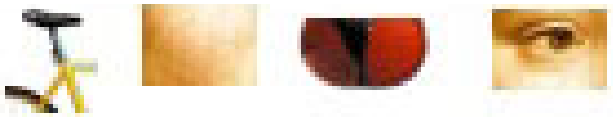
slide by Fei-fei Li



Extract SIFT Feature Descriptors



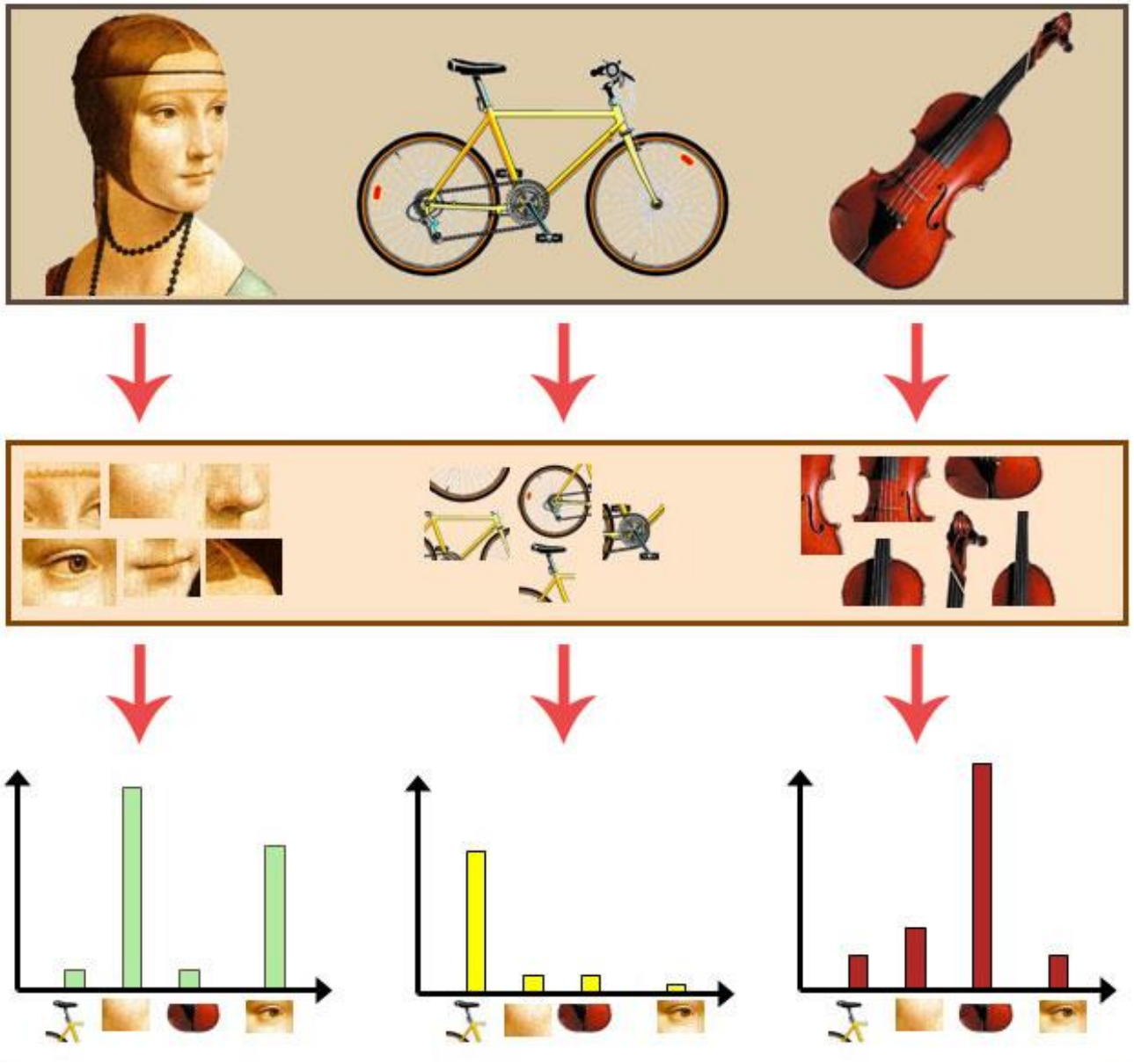
Create Dictionary of Distinctive SIFT Features



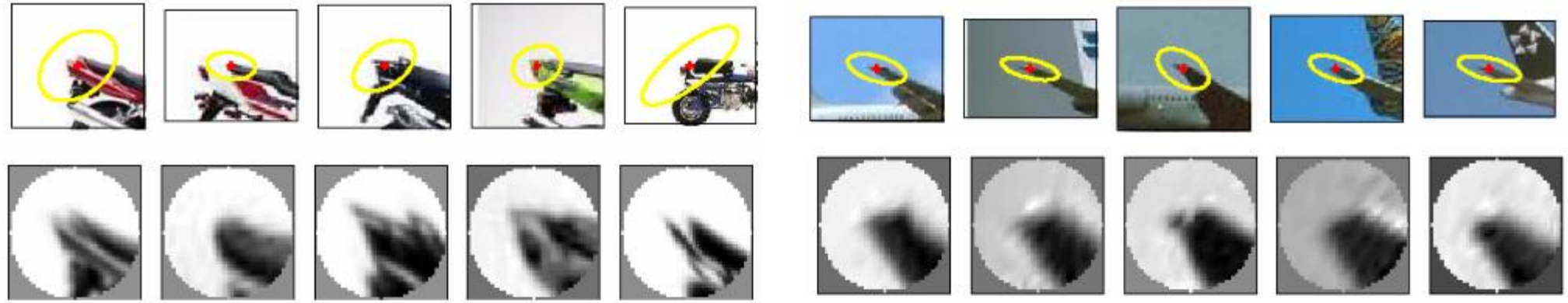
slide by Fei-fei Li

Extract SIFT Feature Descriptors

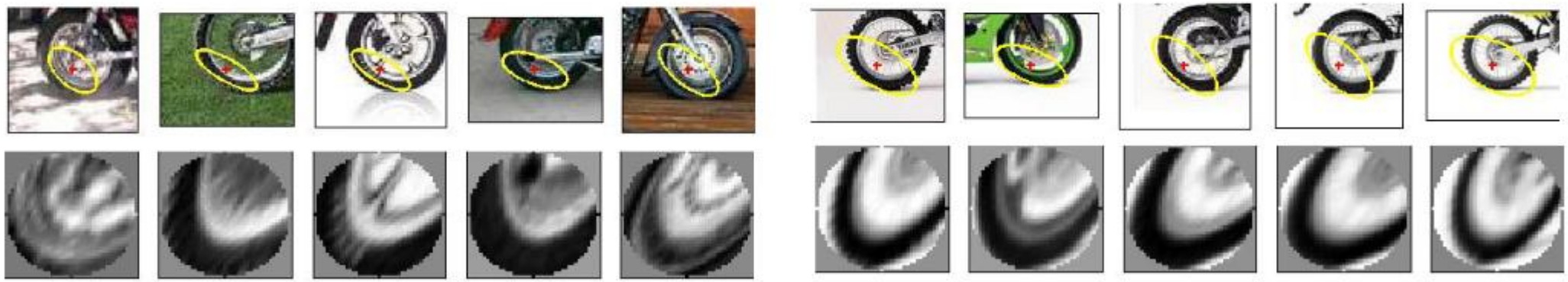
Compute Histograms of Features



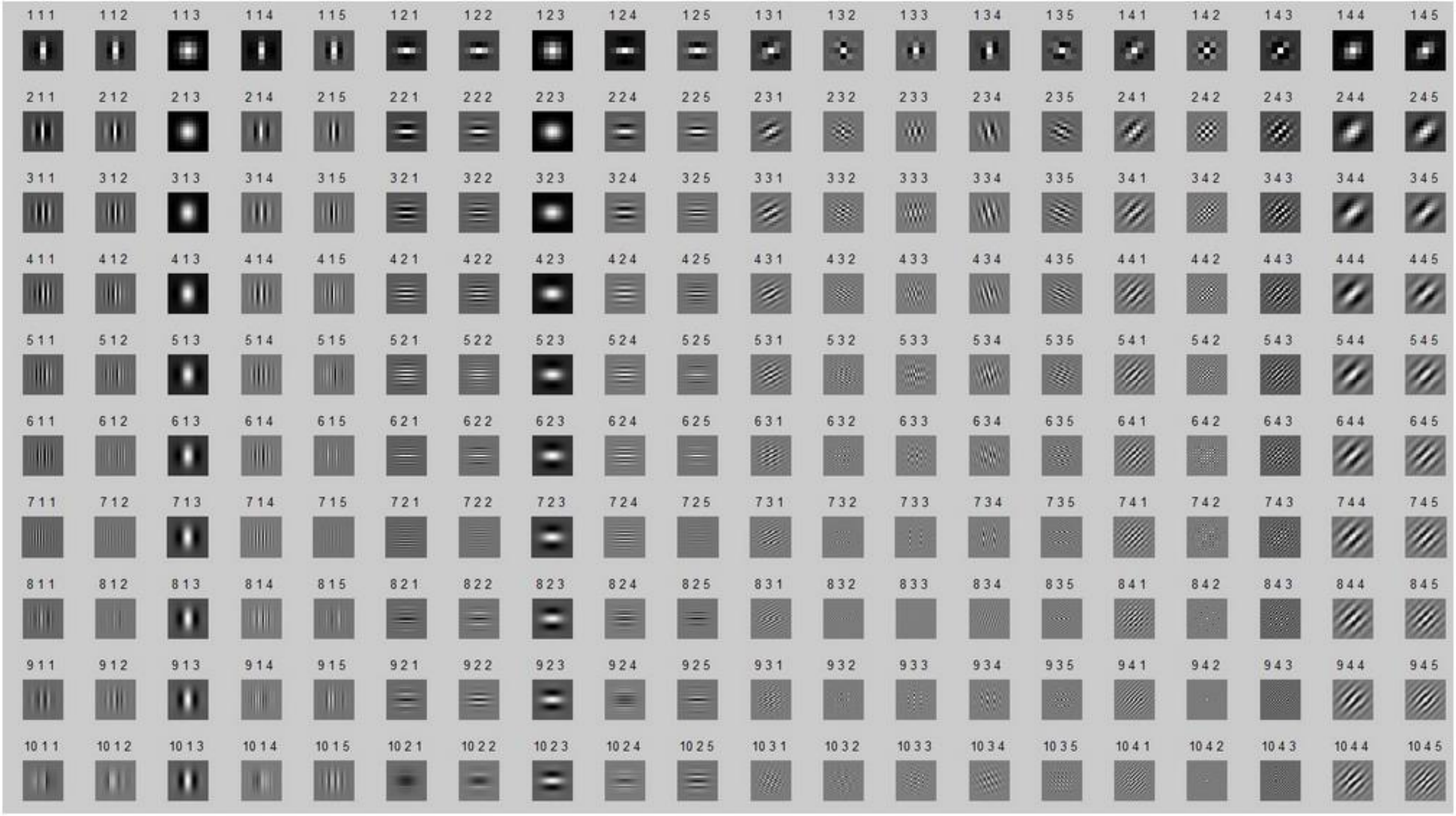
slide by Fei-fei Li



Visual Polysemy: Single visual word occurring on different (but locally similar) parts on different object categories.

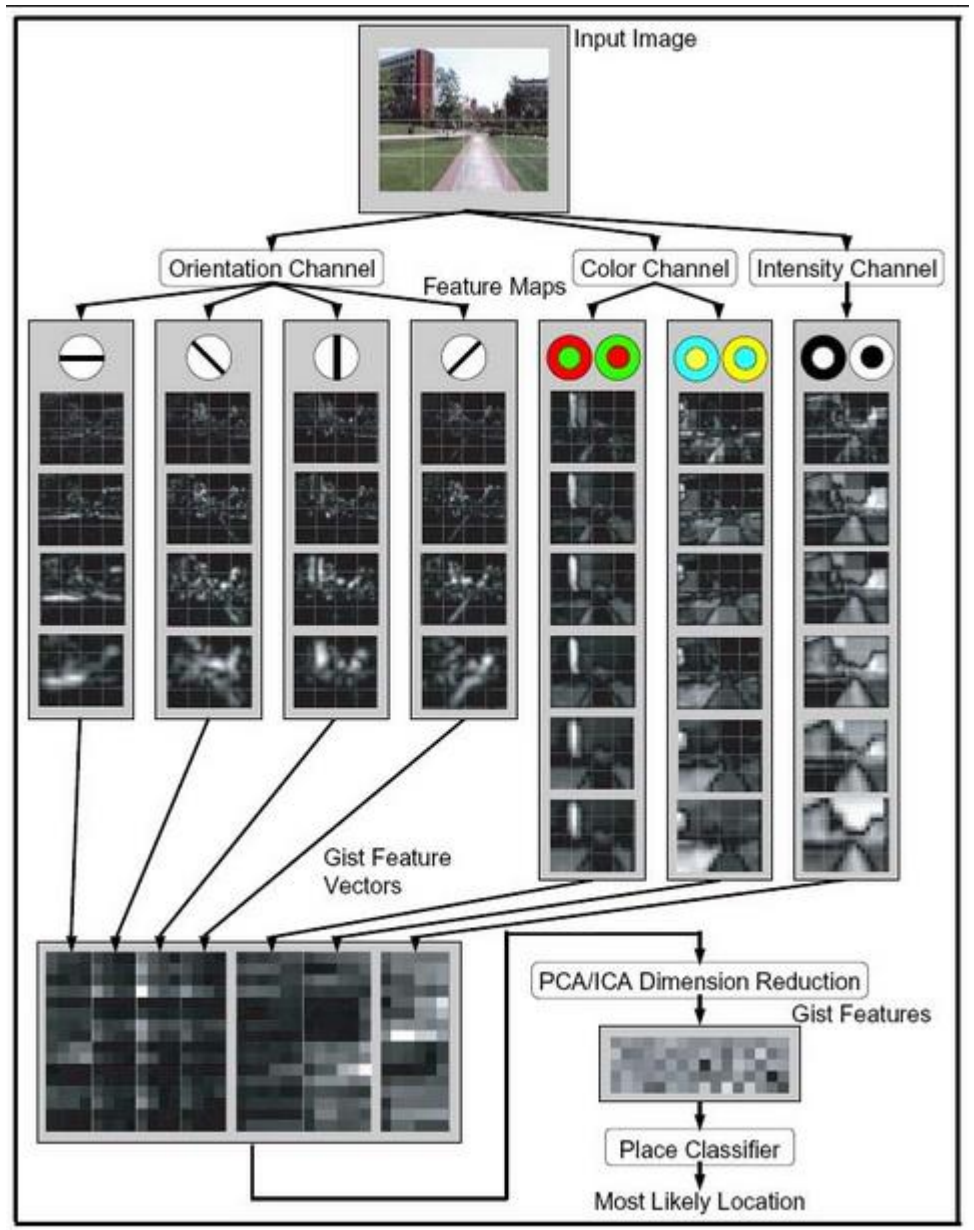


Visual Synonyms: Two different visual words representing a similar part of an object (wheel of a motorbike).


111 112 113 114 115 121 122 123 124 125 131 132 133 134 135 141 142 143 144 145
211 212 213 214 215 221 222 223 224 225 231 232 233 234 235 241 242 243 244 245
311 312 313 314 315 321 322 323 324 325 331 332 333 334 335 341 342 343 344 345
411 412 413 414 415 421 422 423 424 425 431 432 433 434 435 441 442 443 444 445
511 512 513 514 515 521 522 523 524 525 531 532 533 534 535 541 542 543 544 545
611 612 613 614 615 621 622 623 624 625 631 632 633 634 635 641 642 643 644 645
711 712 713 714 715 721 722 723 724 725 731 732 733 734 735 741 742 743 744 745
811 812 813 814 815 821 822 823 824 825 831 832 833 834 835 841 842 843 844 845
911 912 913 914 915 921 922 923 924 925 931 932 933 934 935 941 942 943 944 945
1011 1012 1013 1014 1015 1021 1022 1023 1024 1025 1031 1032 1033 1034 1035 1041 1042 1043 1044 1045

GIST

For more, see http://ilab.usc.edu/siagian/Research/Gist/Gist.html



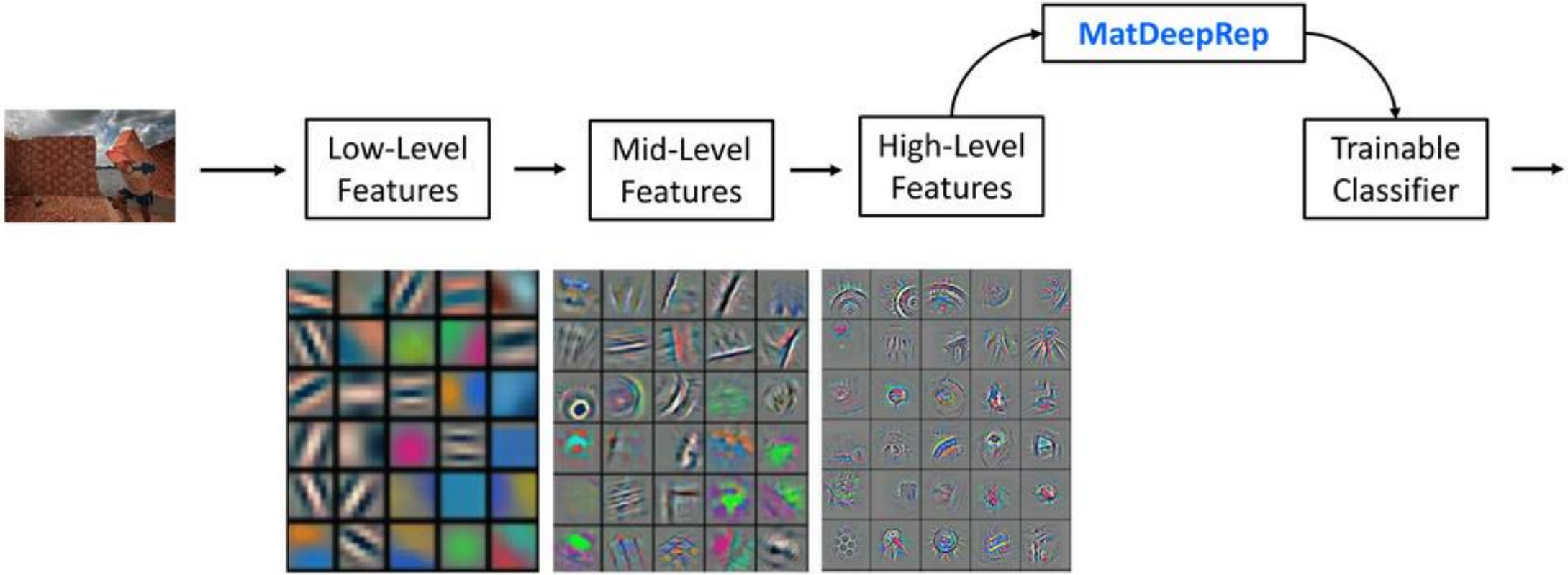
Idea of low, mid and high level features

- Edge, point, line – low level
- Visual words – mid level
- Object feature – high level (more semantic meanings)

- ✓ However, we have different set of techniques for extracting such features before deep learning
- ✓ Those techniques are not optimized
- ✓ No idea on which feature is good for which data/task

But, this gives the notion of hierarchical features!

In deep learning



Suggested reading

1940 PROCEEDINGS OF THE IRE November

What the Frog's Eye Tells the Frog's Brain*

J. Y. LETTVIN†, H. R. MATURANA‡, W. S. McCULLOCH‖, SENIOR MEMBER, IRE,
AND W. H. PITTS‖

Suggested readings

Recent Advances in Features Extraction and Description Algorithms: A Comprehensive Survey

Ehab Salahat, *Member, IEEE*, and Murad Qasaimeh, *Member, IEEE*