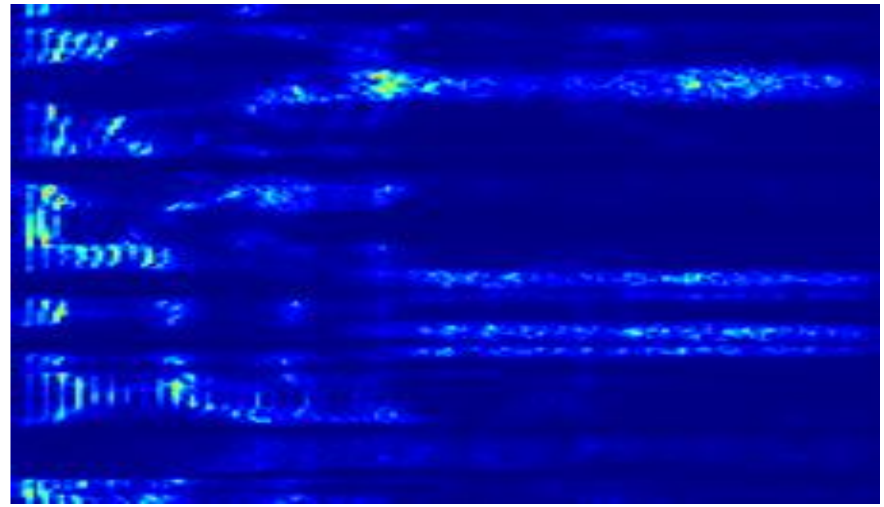
DAY 4: APPLICATIONS OF MACHINE VISION

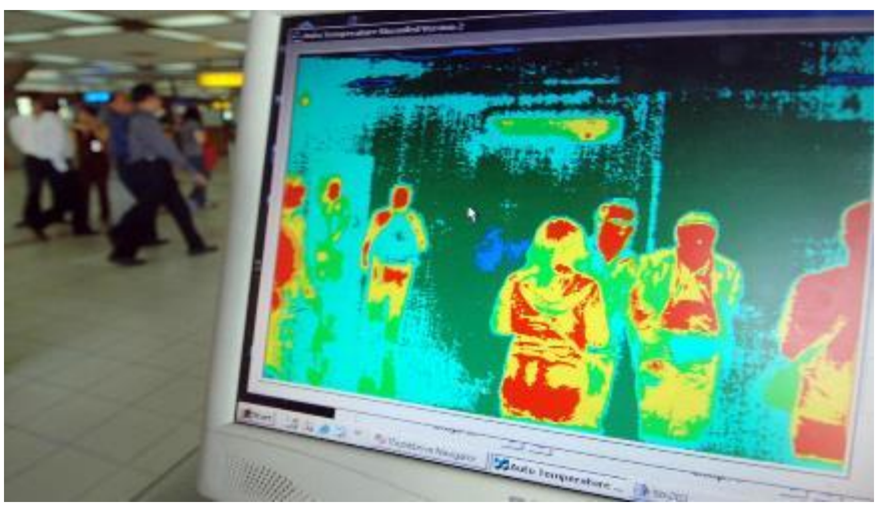
KE5108: DEVELOPING INTELLIGENT SYSTEMS FOR PERFORMING BUSINESS ANALYTICS

Dr TIAN Jing

tianjing@nus.edu.sg

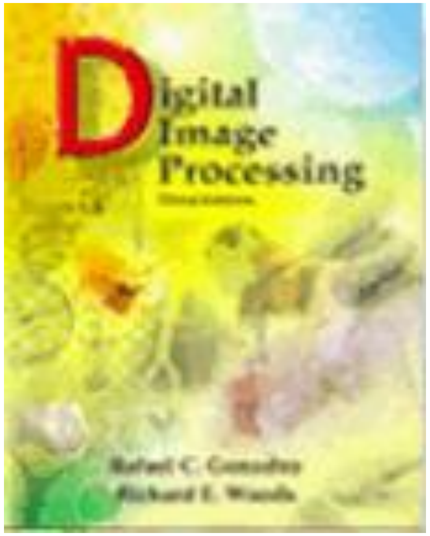
Image data

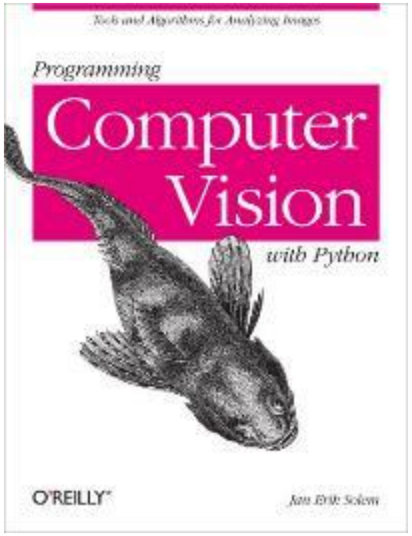




Reference

- R. C. Gonzalez and R. E. Woods, ***Digital Image Processing***, http://www.imageprocessingplace.com/
- **Computer Vision Crash Course**, Jia-Bin Huang, https://filebox.ece.vt.edu/~jbhuang/
- **Computer Vision: Algorithms and Applications**, Richard Szeliski, http://szeliski.org/Book/
- Programming Computer Vision with Python, http://programmingcomputervision.com/





Outline

- Introduction
- Feature representation: Motion
- Feature representation: Frequency-domain
- Classification and object detection

INTRODUCTION

Computer vision tasks


Face Detection/Recognition

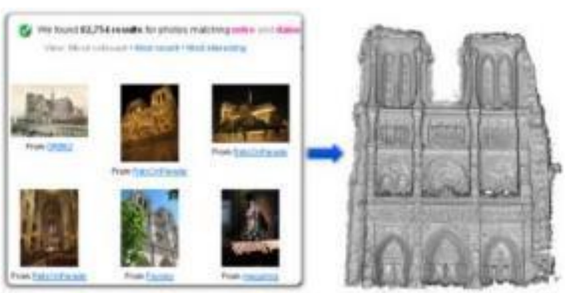

Object Detection


Sports


Object Tracking


Human Pose


Autonomous vehicle


Multi-view Geometry

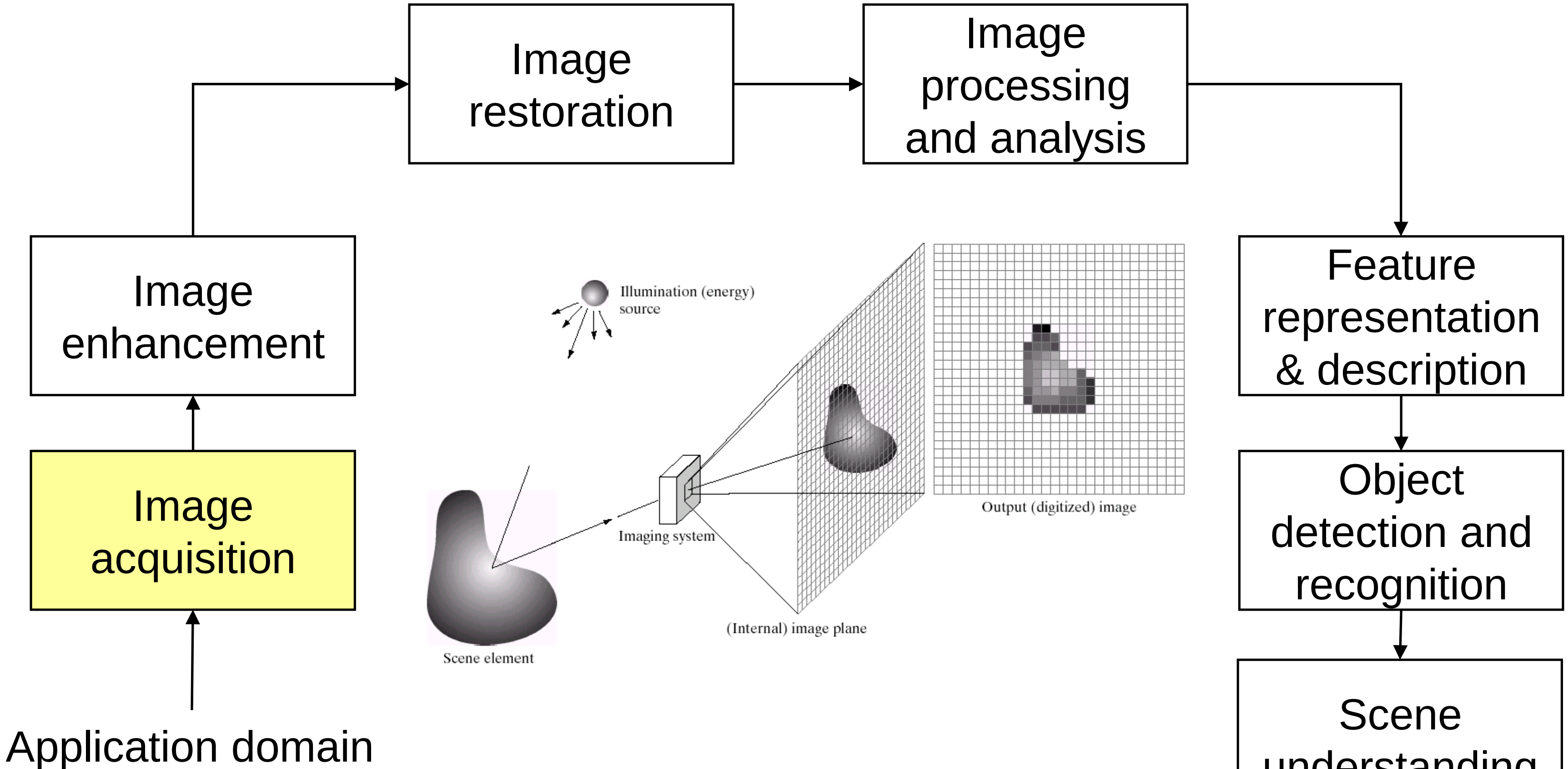
3D Scene


Vision for Robots

Outline

Low Level Task	Mid Level Task	High Level Task
Input: Image **Output:** Image **Examples:** Noise removal, image sharpening	**Input:** Image **Output:** Attributes **Examples:** Object recognition, segmentation	**Input:** Attributes/Image **Output:** Understanding **Examples:** Scene understanding, autonomous navigation

Typical image analytics pipeline



Typical image analytics pipeline

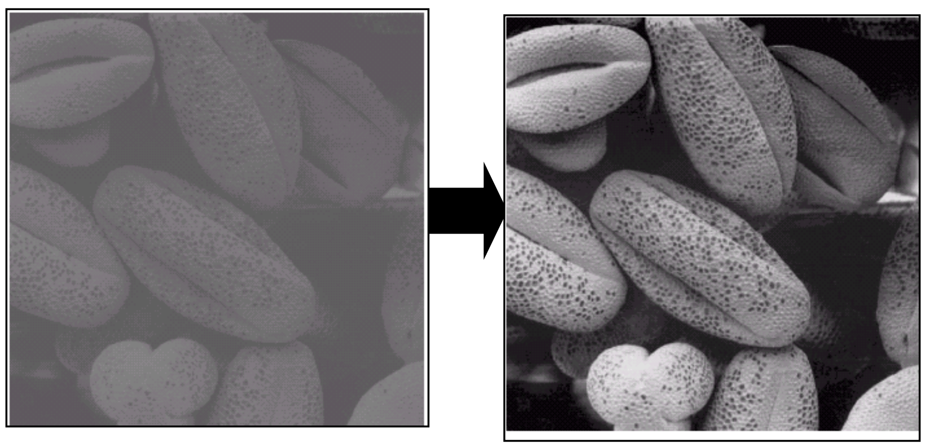
Image restoration

Image processing and analysis

Image enhancement

Image acquisition

Application domain

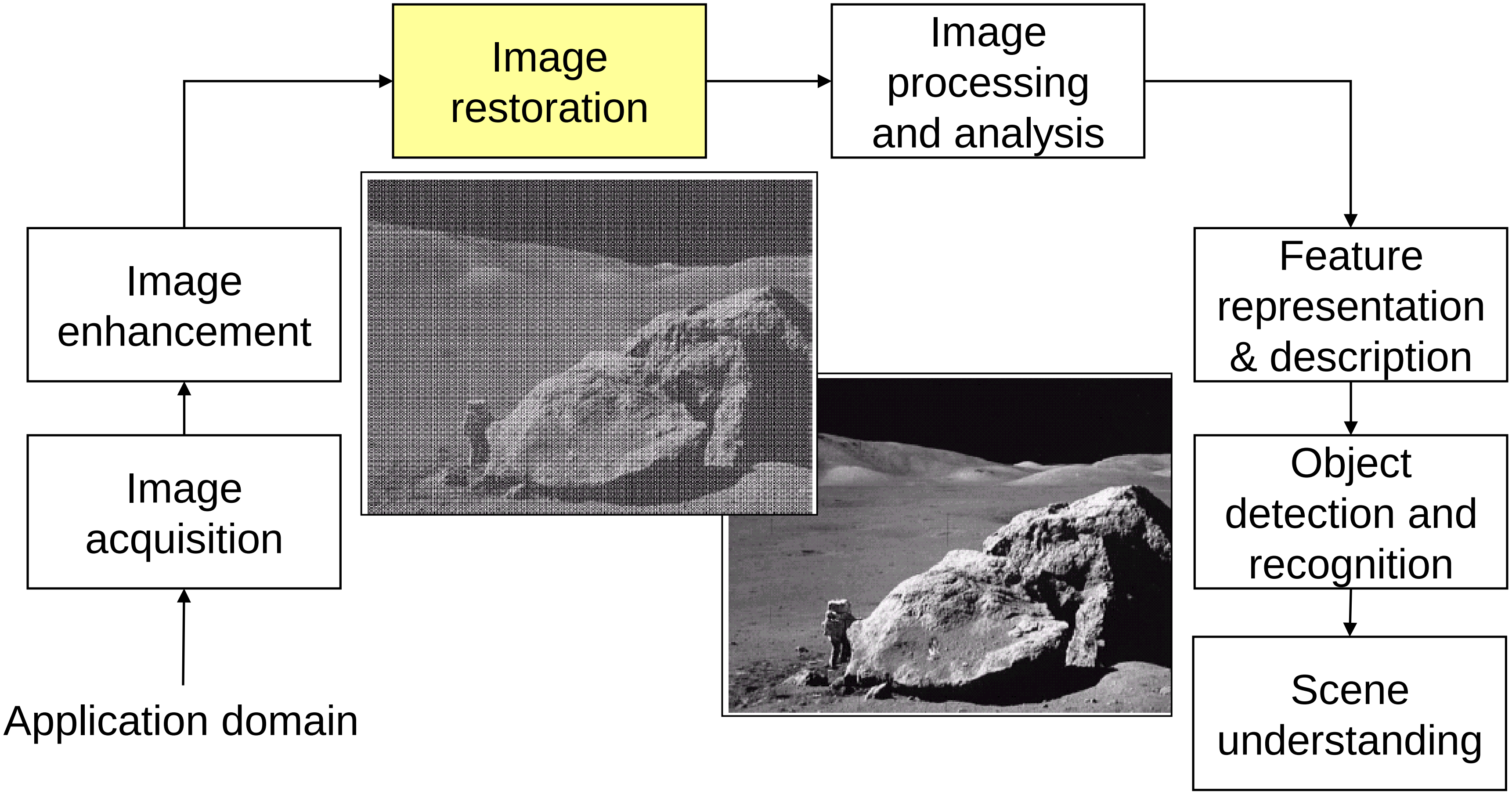


Feature representation & description

Object detection and recognition

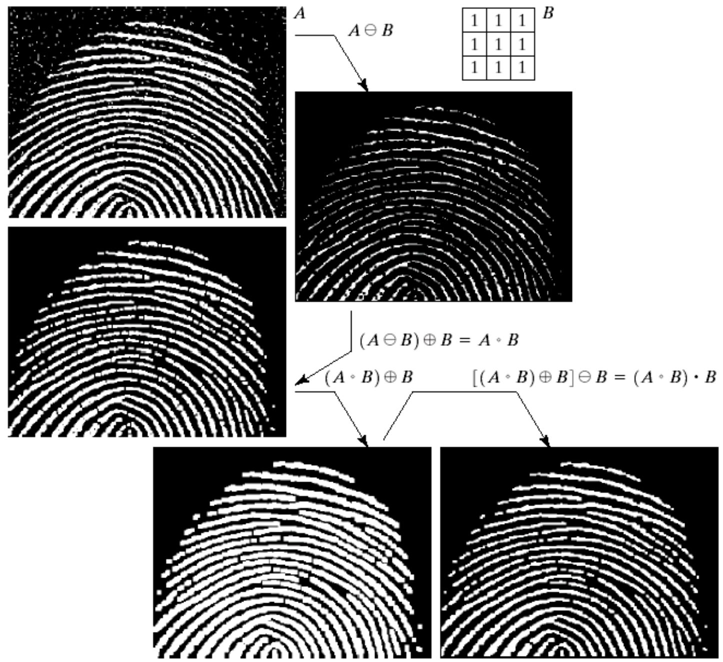
Scene understanding

Typical image analytics pipeline



Typical image analytics pipeline

Application domain → Image acquisition → Image enhancement → Image restoration → **Image processing and analysis** → Feature representation & description → Object detection and recognition → Scene understanding

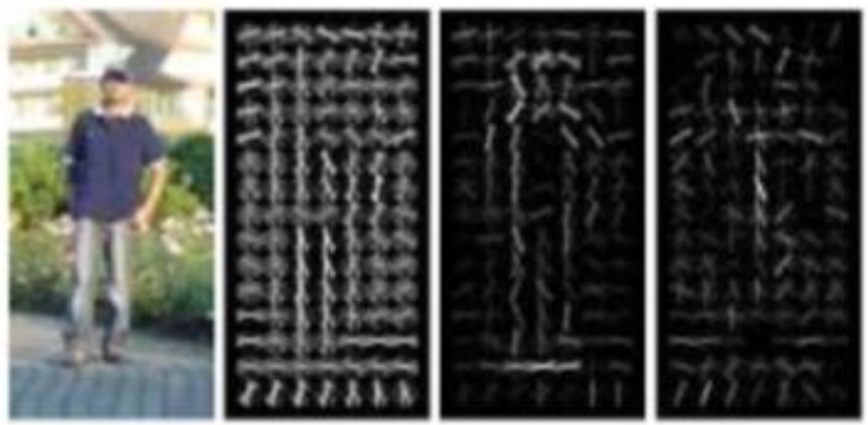


Typical image analytics pipeline

Image restoration

Image processing and analysis

Image enhancement



Feature representation & description

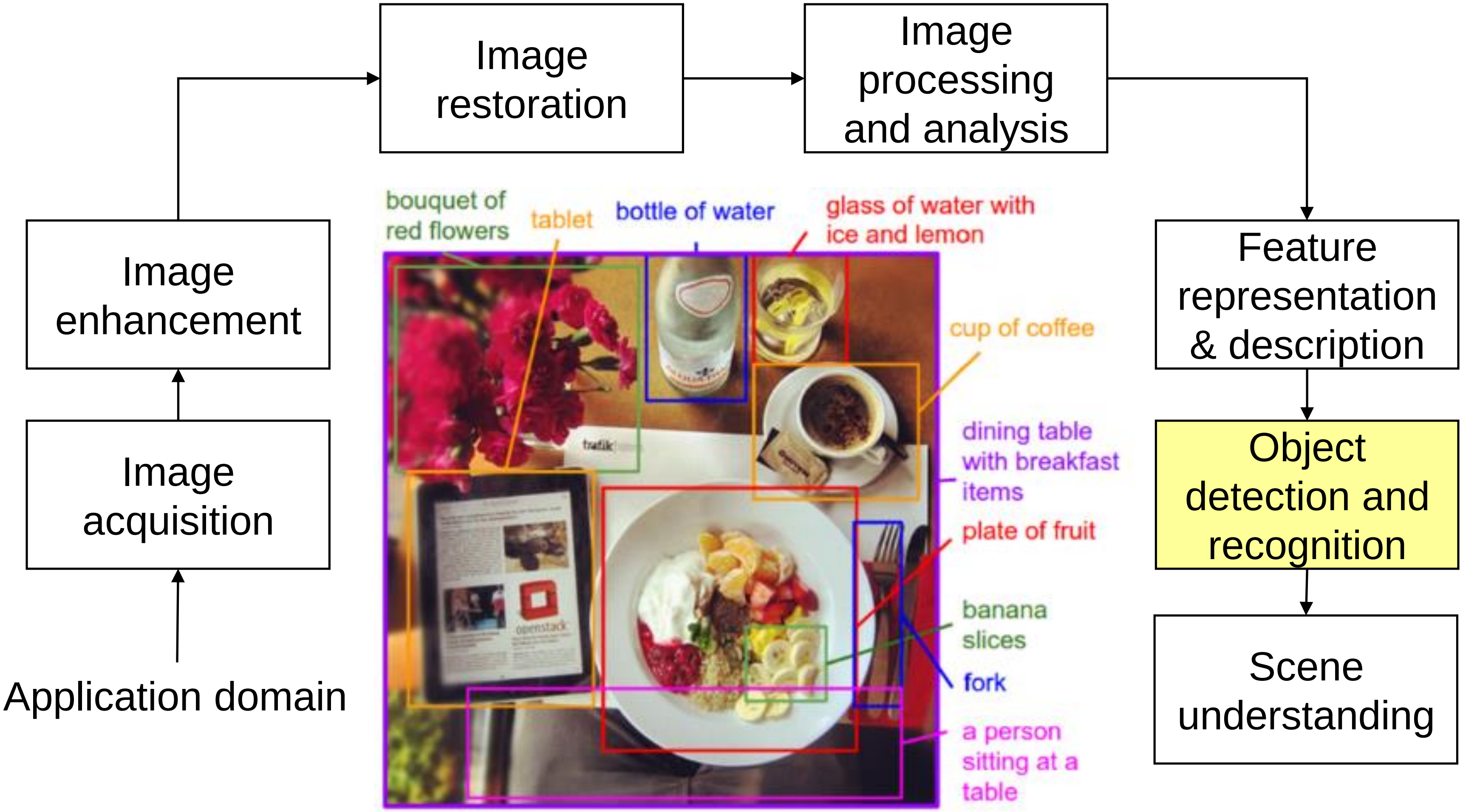
Image acquisition

Object detection and recognition

Application domain

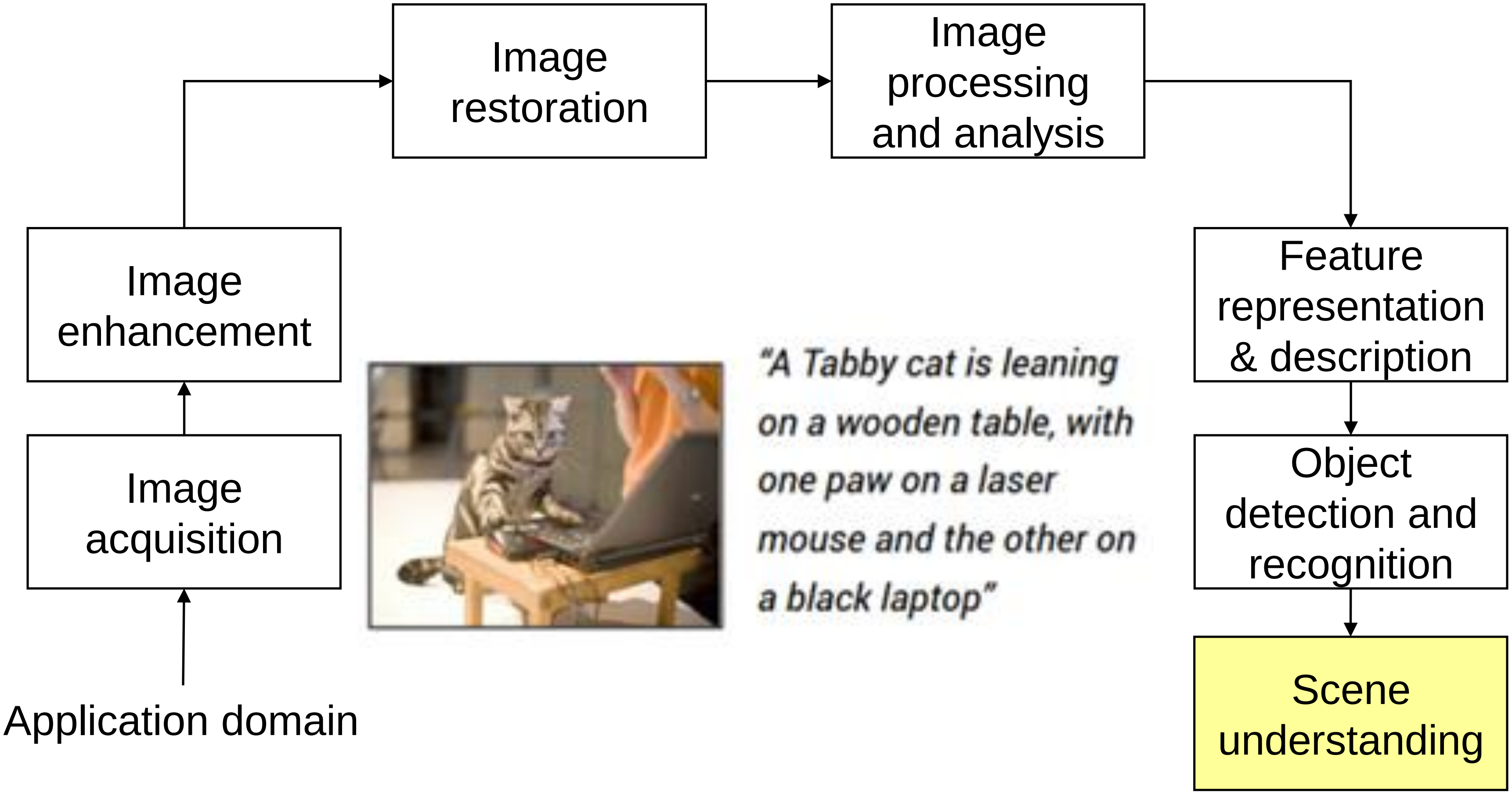
Scene understanding

Typical image analytics pipeline



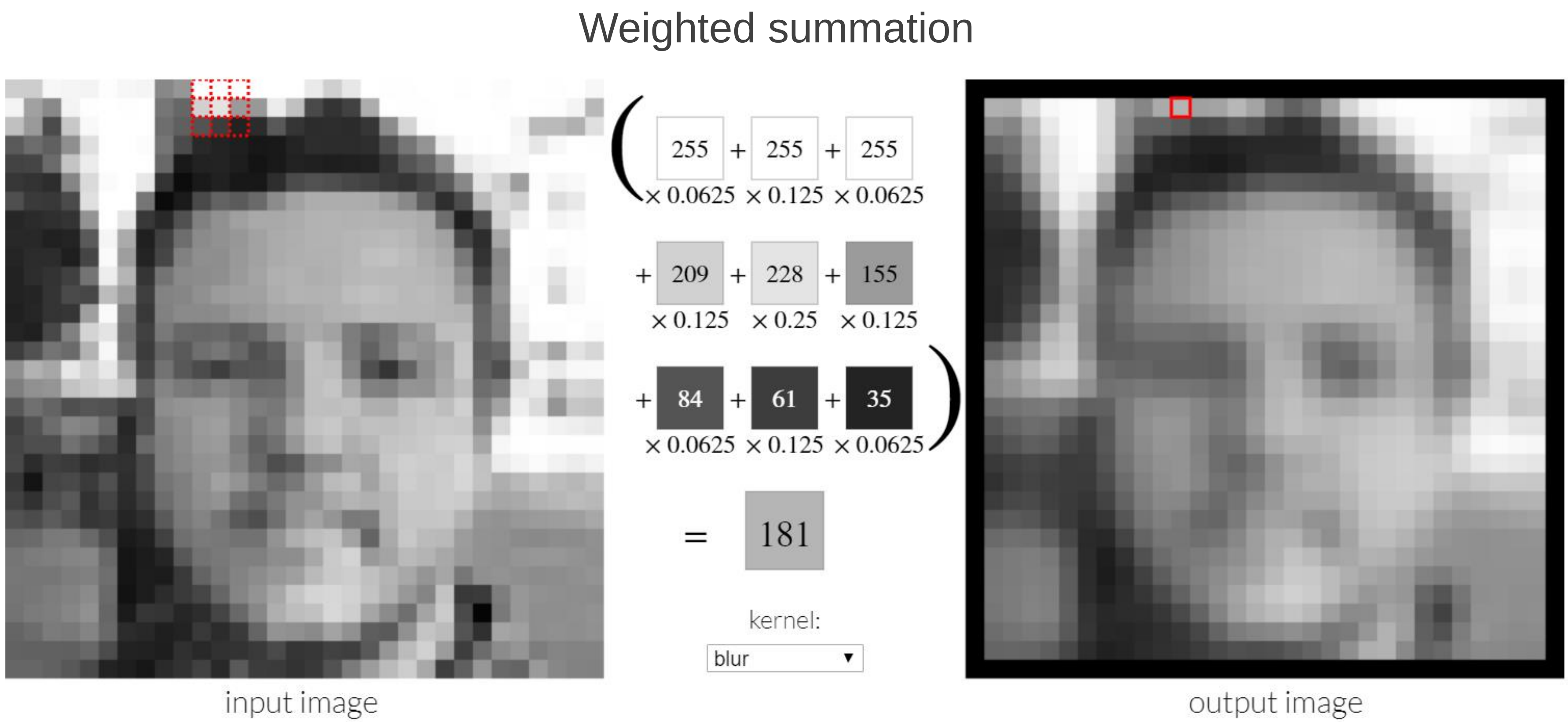
Source: https://adeshpande3.github.io/assets/Caption.png

Typical image analytics pipeline



FEATURE: FREQUENCY-DOMAIN

Recap: Image filtering



Source: http://setosa.io/ev/image-kernels/

Review: questions

* Is the filtering operator

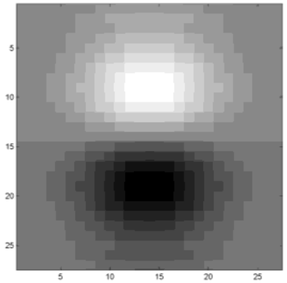
Example question:

Fill in the blanks

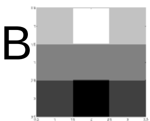
a) _ = D * B

b) F = D * _

A



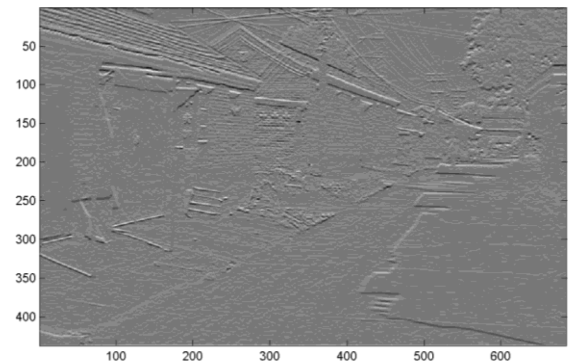
B



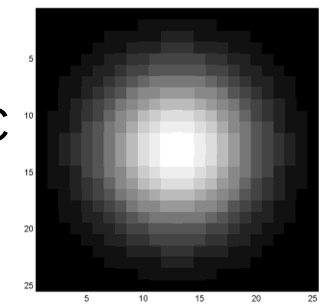
E

F

G

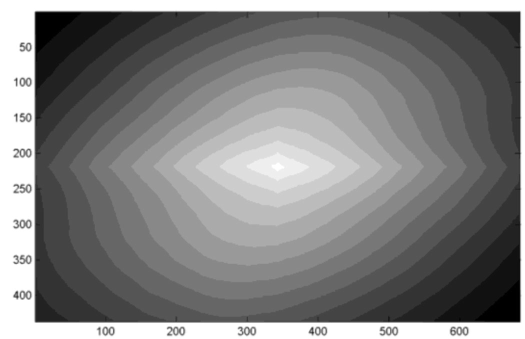


C



H

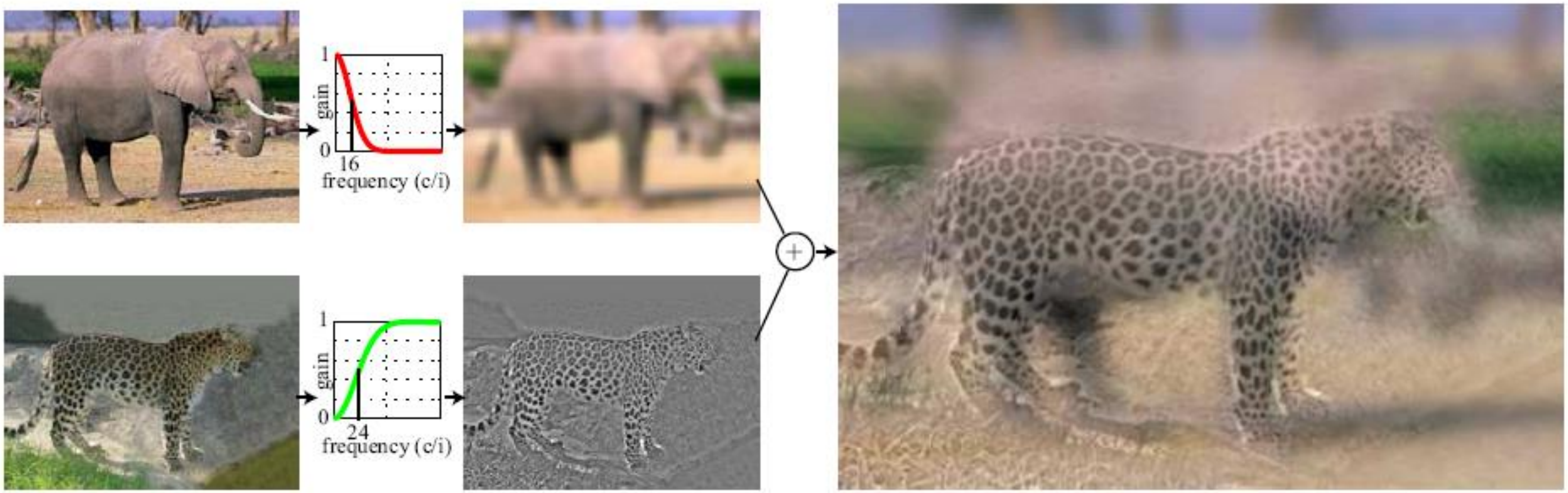
I



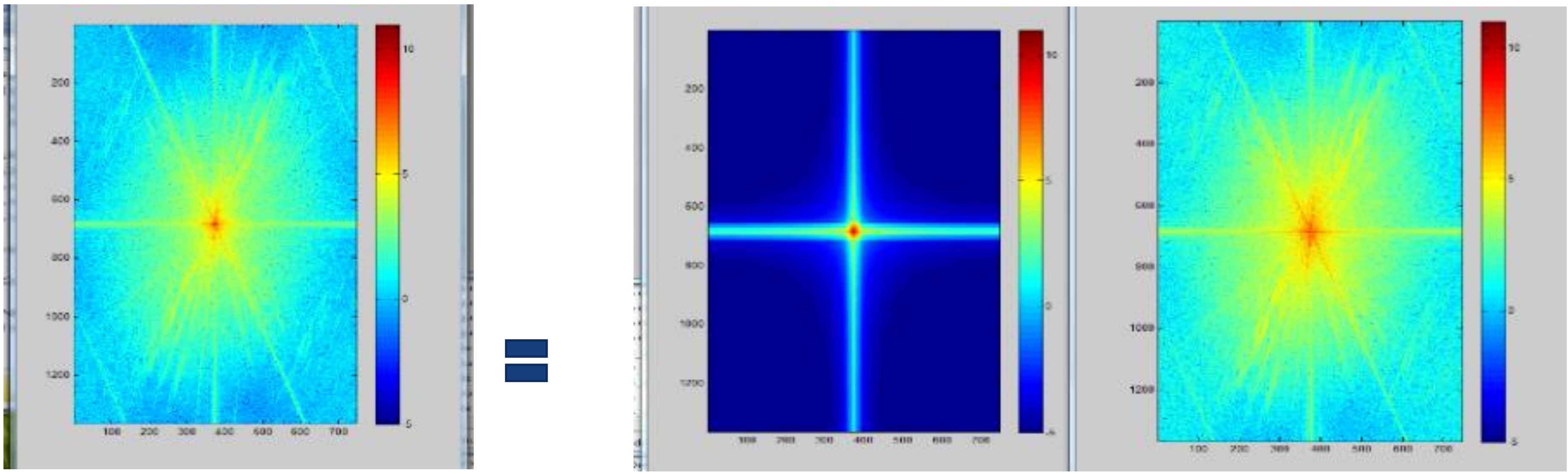
D

What do you see?

Why

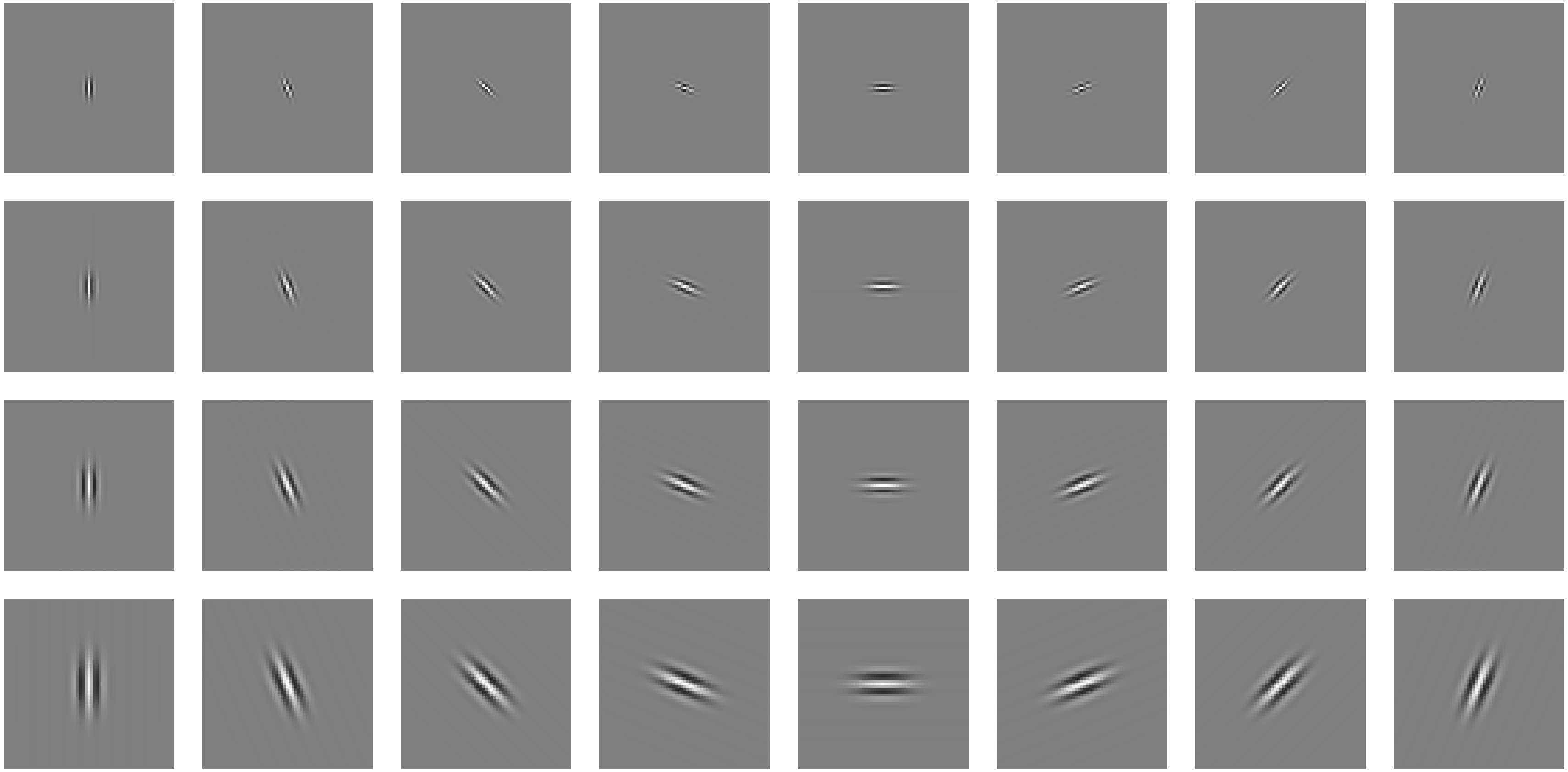


Hybrid Image   Low-passed Image + High-passed Image



Inspiration from human perception

- Early processing in humans perception filters for orientations and scales of frequency.



Early visual processing: Multi-scale edge and blob filters

Convolution theorem

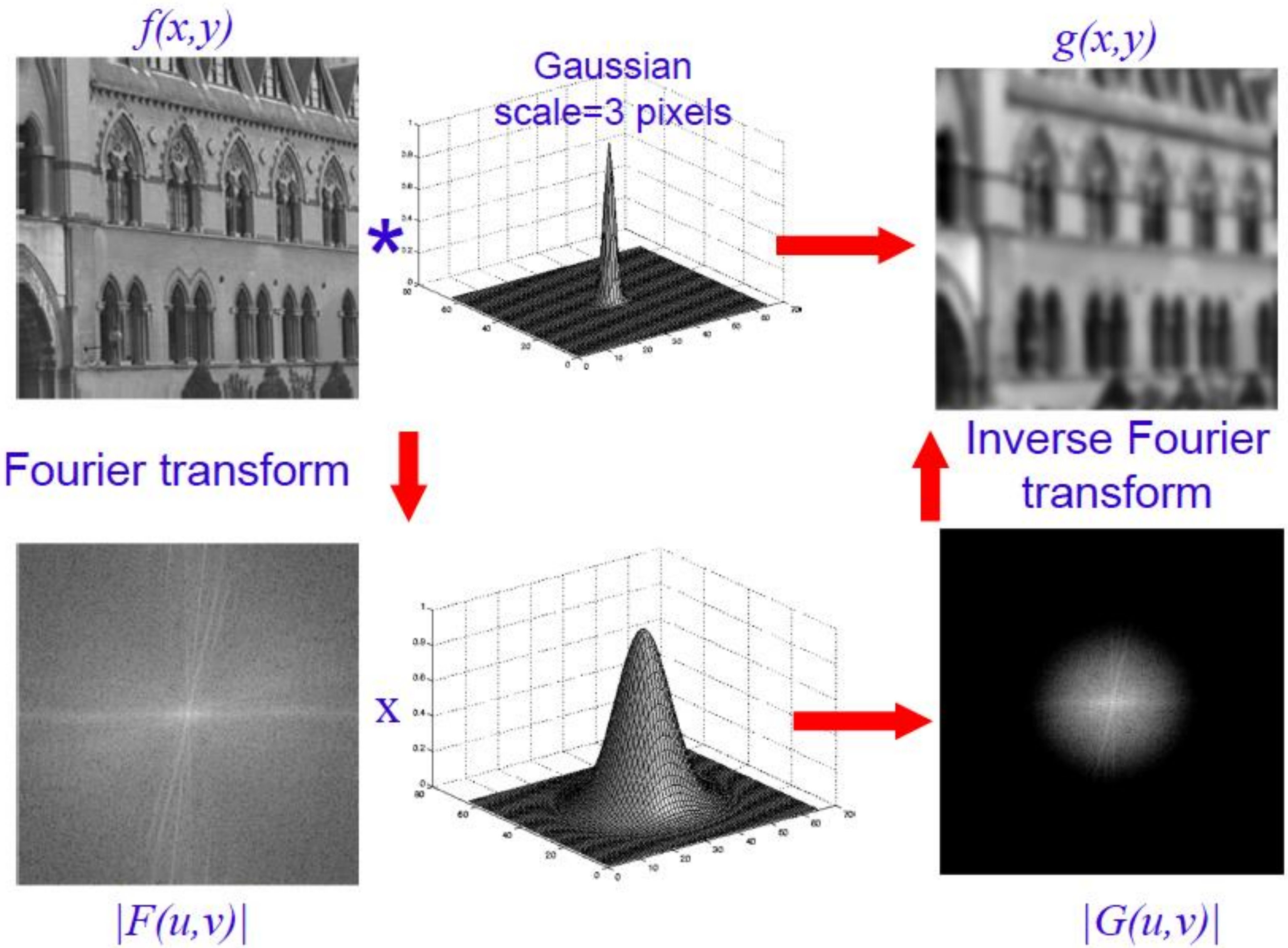
Example: f and g are functions defined in spatial domain, while F and G are their corresponding functions defined in Fourier domain

$$f(x,y) * g(x,y) \Leftrightarrow F(u,v)G(u,v)$$

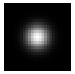
In words: the Fourier transform of the convolution of two functions is the product of their individual Fourier transforms

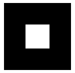
Because linear filtering operations can be carried out by simple multiplications in the Fourier domain

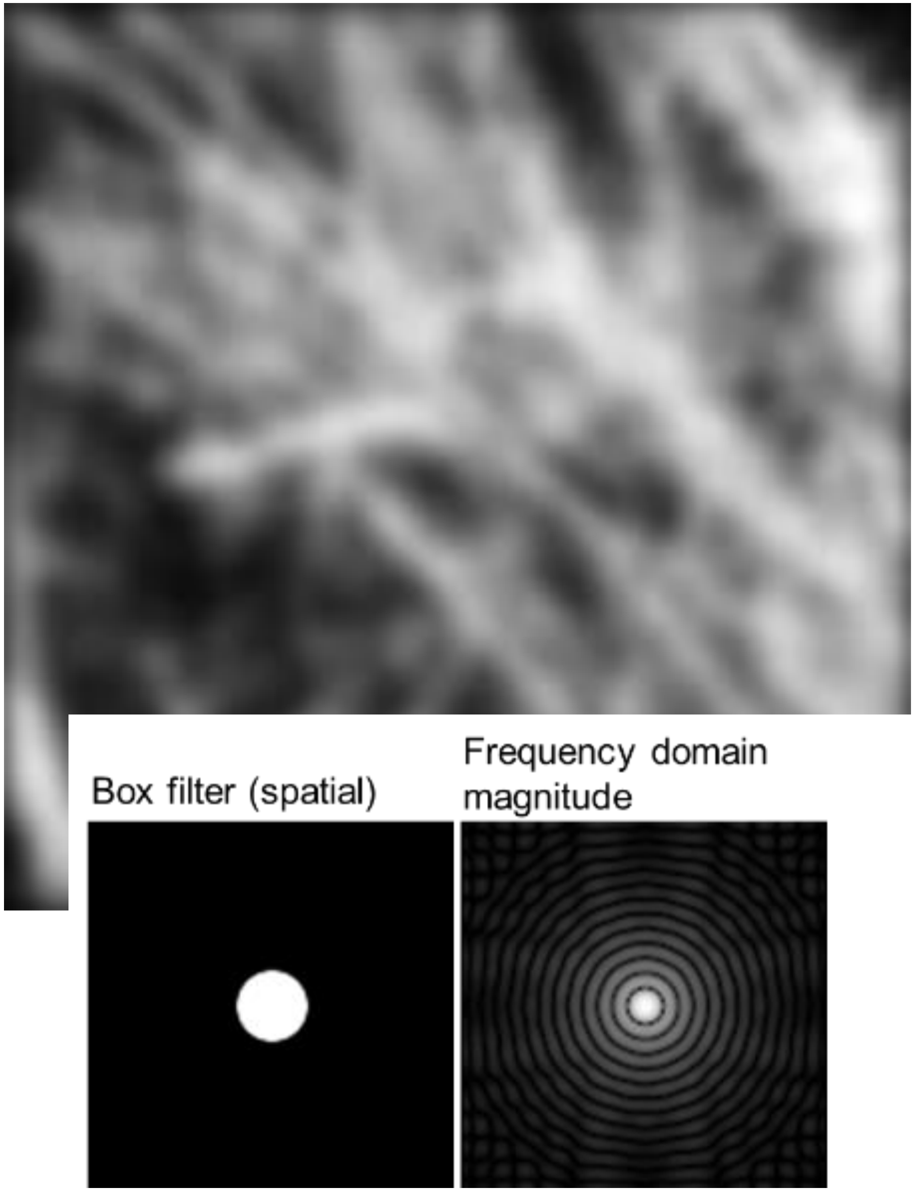
Convolution theorem

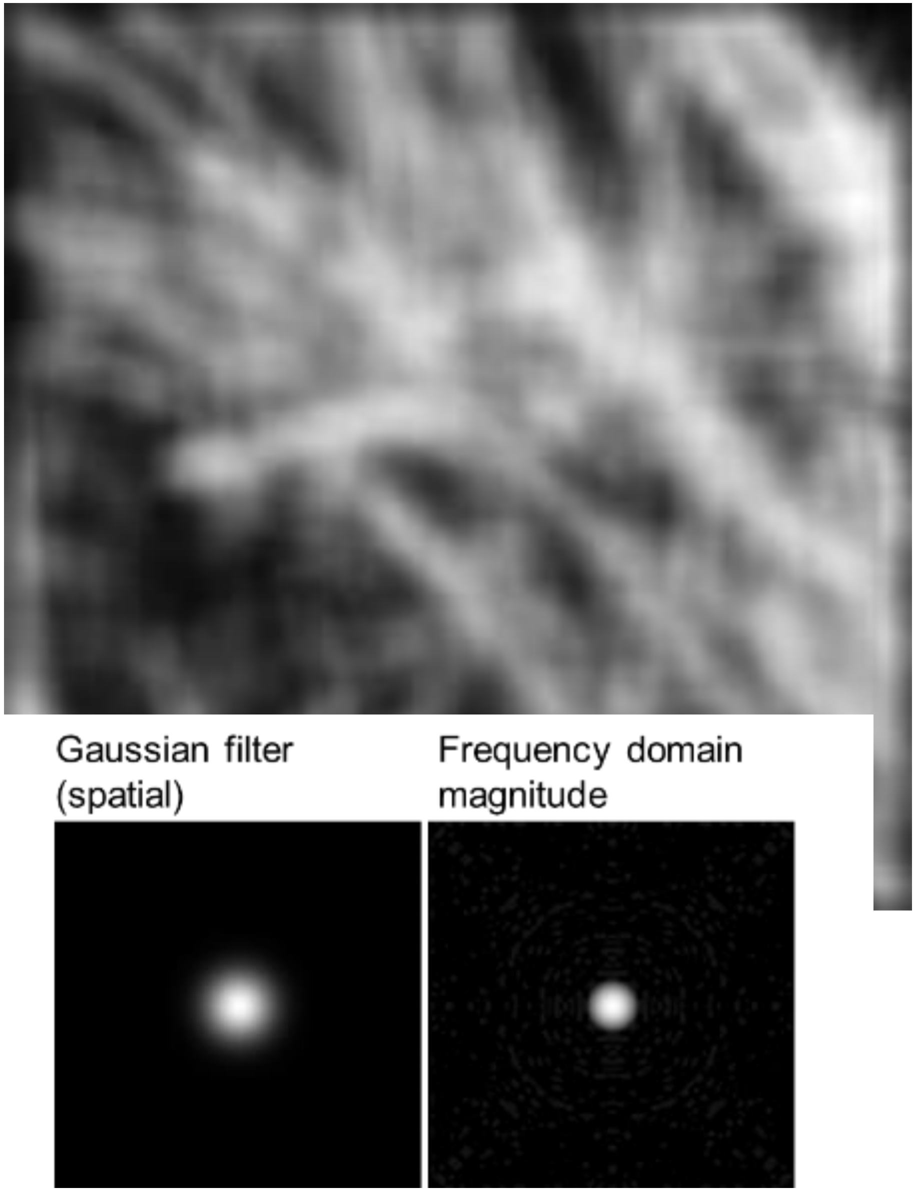


Example

Gaussian 

Box filter 

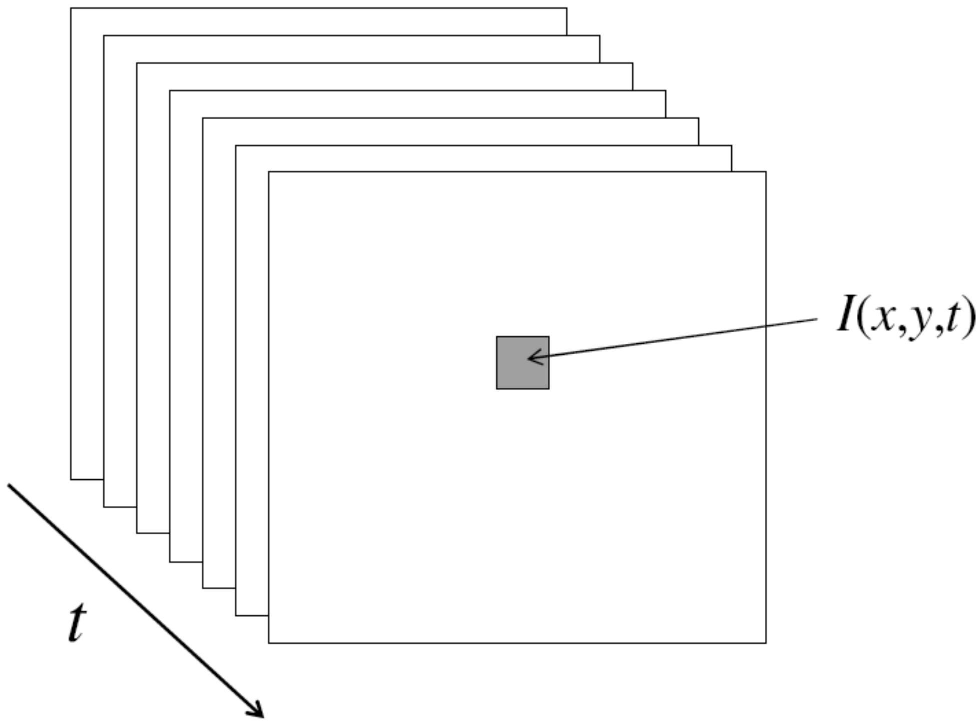




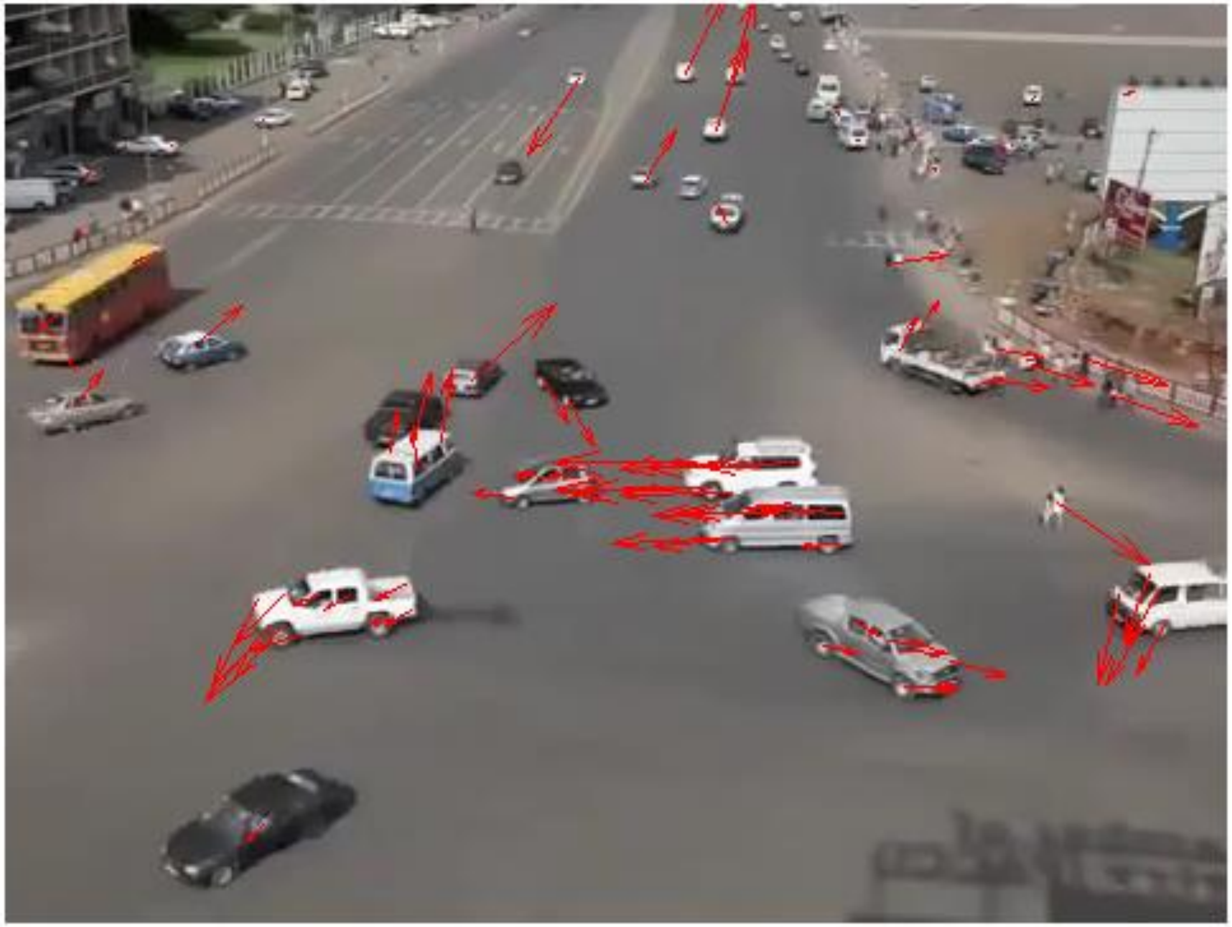
FEATURE: MOTION

From image to video

- A video is a sequence of frames captured over time
- Now our image data is a function of space (x, y) and time (t)



Why is motion useful?

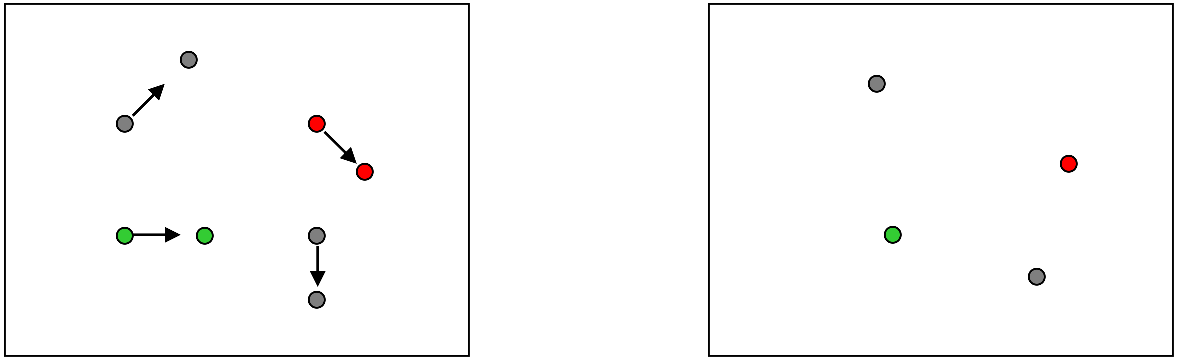


Optical flow

- Definition: optical flow is the apparent motion of brightness patterns in the image
- Note: apparent motion can be caused by lighting changes without any actual motion

GOAL: Estimate image motion at each pixel from optical flow.

Estimating optical flow

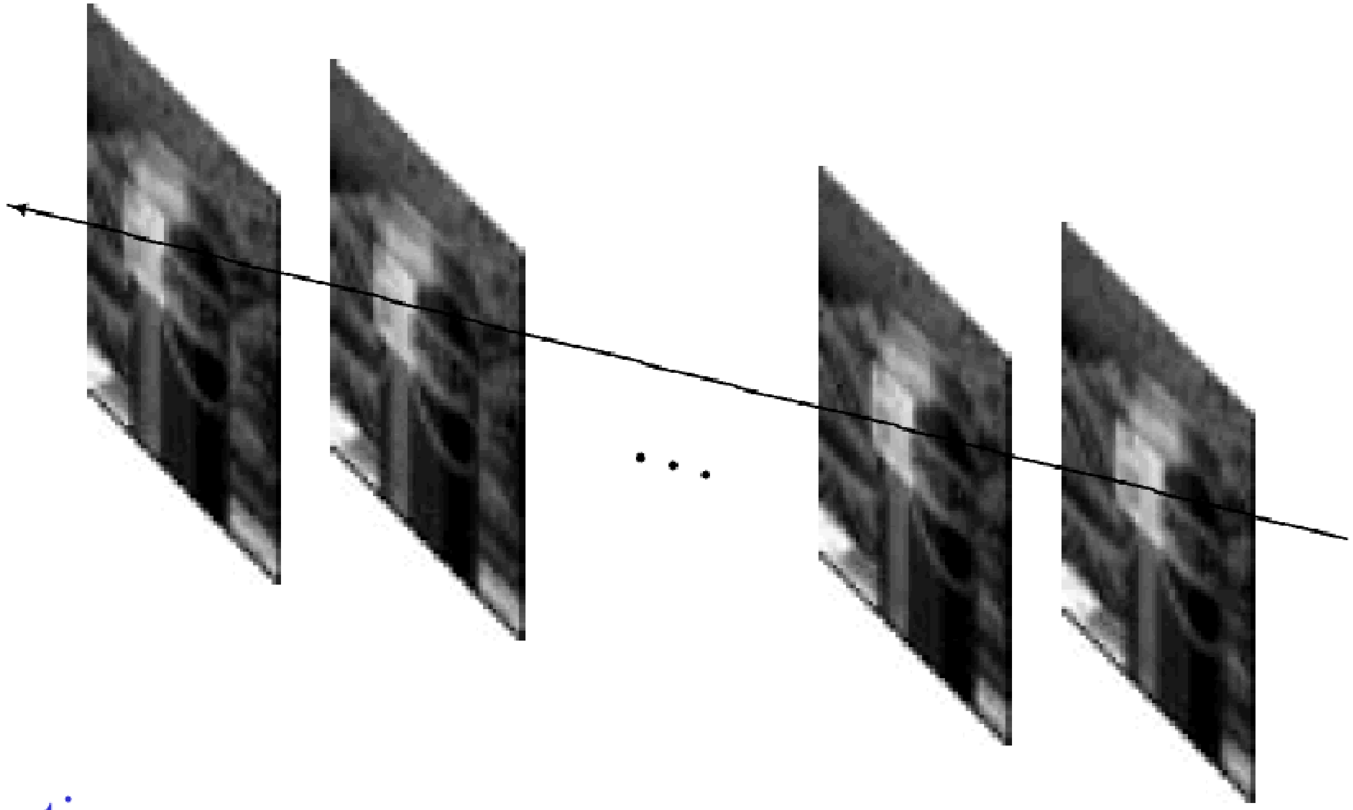


$I(x,y,t-1)$ $I(x,y,t)$

- Given two subsequent frames, estimate the apparent motion vector field $u(x,y)$, $v(x,y)$ between them

- Key assumptions
  - **Brightness constancy:** projection of the same point looks the same in every frame
  - **Small motion:** points do not move very far; the length of the vector $u(x,y)$, $v(x,y)$ are small.

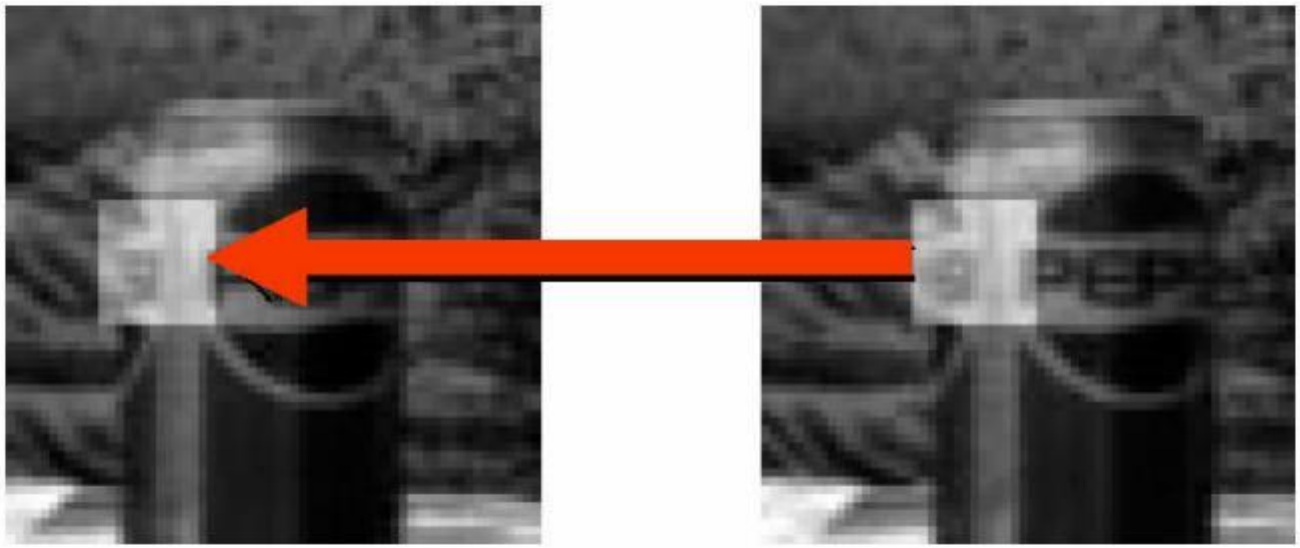
Small motion



Assumption:

The image motion of a surface patch changes gradually over time.

Brightness constancy



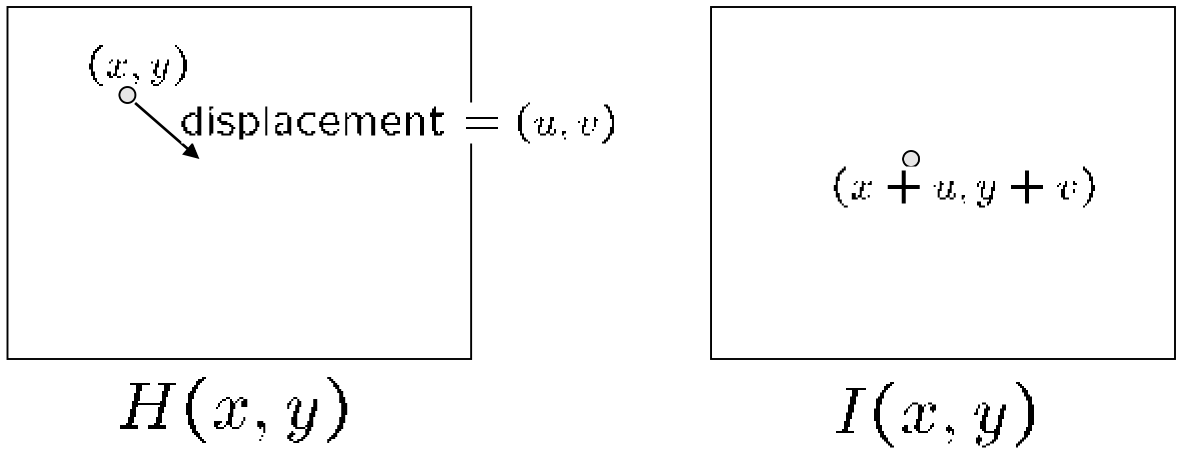
Assumption

Image measurements (e.g. brightness) in a small region remain the same although their location may change.

$$I(x+u, y+v, t+1) = I(x, y, t)$$

(assumption)

Optical flow constraints



- brightness constancy: H(x,y)=I(x+u, y+v)

- small motion: suppose we take the Taylor series expansion of I:

$$I(x+u, y+v) = I(x, y) + \frac{\partial I}{\partial x}u + \frac{\partial I}{\partial y}v + \boxed{\text{higher order terms}}$$
$$\approx I(x, y) + \frac{\partial I}{\partial x}u + \frac{\partial I}{\partial y}v$$

Optical flow equation

Combining these two equations

shorthand: $I_x = \frac{\partial I}{\partial x}$

$$
\begin{aligned}
0 &= I(x+u, y+v) - H(x,y) \\
&\approx I(x,y) + I_x u + I_y v - H(x,y) \\
&\approx (I(x,y) - H(x,y)) + I_x u + I_y v \\
&\approx I_t + I_x u + I_y v \\
&\approx I_t + \nabla I \cdot [u \;\; v]
\end{aligned}
$$

In the limit as u and v go to zero, this becomes exact

$$0 = I_t + \nabla I \cdot [\tfrac{\partial x}{\partial t} \;\; \tfrac{\partial y}{\partial t}]$$

Find solution

- How to get more equations for a pixel?
- **Spatial coherence constraint:**
- Assume the pixel's neighbors have the same (u,v)
  - If we use a 5x5 window, that gives us 25 equations per pixel

$$0 = I_t(\mathbf{p_i}) + \nabla I(\mathbf{p_i}) \cdot [u \;\; v]$$

$$\begin{bmatrix} I_x(\mathbf{p}_1) & I_y(\mathbf{p}_1) \\ I_x(\mathbf{p}_2) & I_y(\mathbf{p}_2) \\ \vdots & \vdots \\ I_x(\mathbf{p}_{25}) & I_y(\mathbf{p}_{25}) \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = - \begin{bmatrix} I_t(\mathbf{p}_1) \\ I_t(\mathbf{p}_2) \\ \vdots \\ I_t(\mathbf{p}_{25}) \end{bmatrix}$$

B. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pp. 674–679, 1981.

Lucas-Kanade flow

- Overconstrained linear system

$$\begin{bmatrix} I_x(\mathbf{p}_1) & I_y(\mathbf{p}_1) \\ I_x(\mathbf{p}_2) & I_y(\mathbf{p}_2) \\ \vdots & \vdots \\ I_x(\mathbf{p}_{25}) & I_y(\mathbf{p}_{25}) \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = - \begin{bmatrix} I_t(\mathbf{p}_1) \\ I_t(\mathbf{p}_2) \\ \vdots \\ I_t(\mathbf{p}_{25}) \end{bmatrix} \qquad \underset{25\text{x}2}{A} \ \underset{2\text{x}1}{d} = \underset{25\text{x}1}{b}$$

Least squares solution for *d* given by $(A^T A) \ d = A^T b$

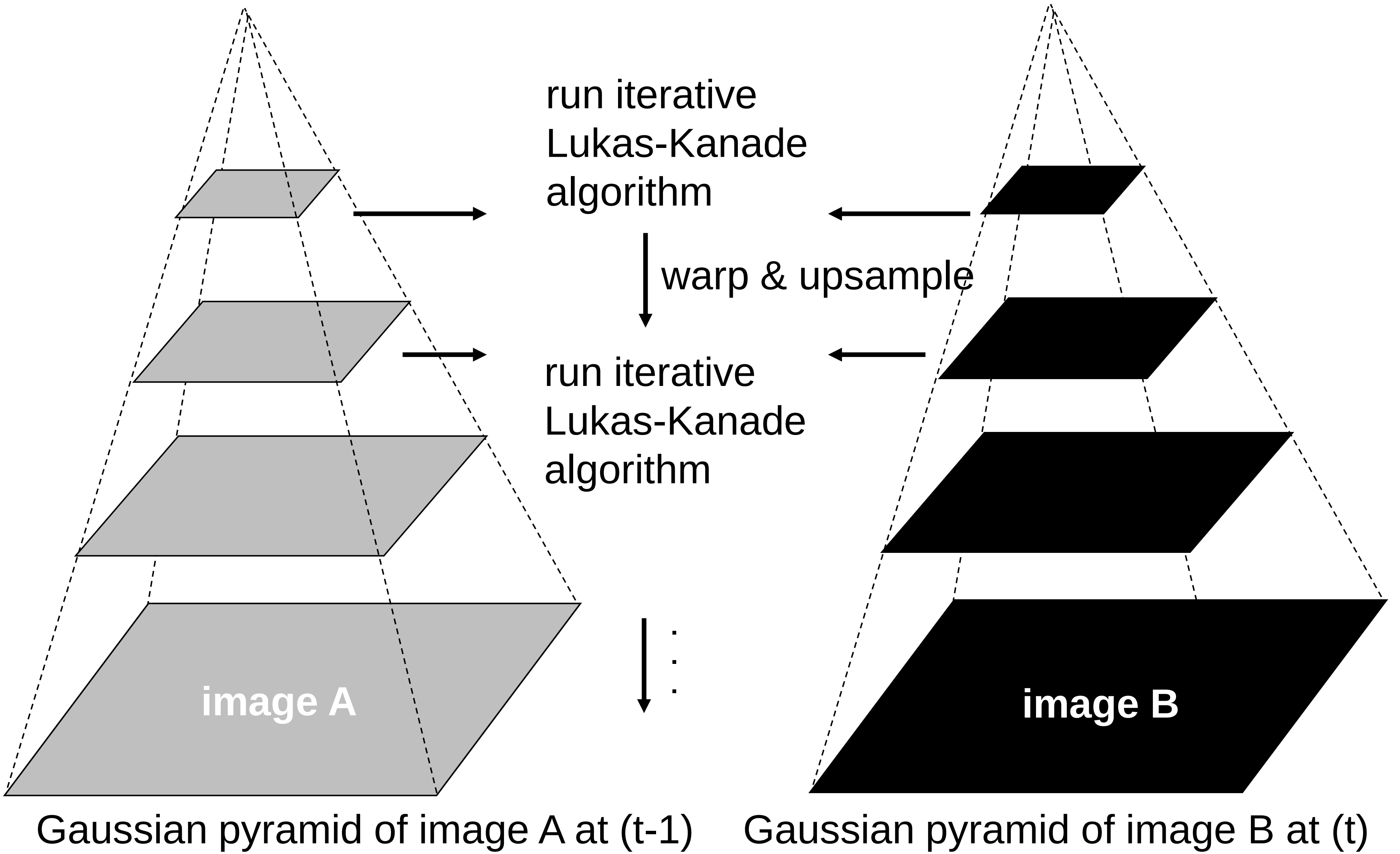
$$\underbrace{\begin{bmatrix} \sum I_x I_x & \sum I_x I_y \\ \sum I_x I_y & \sum I_y I_y \end{bmatrix}}_{A^T A} \begin{bmatrix} u \\ v \end{bmatrix} = - \underbrace{\begin{bmatrix} \sum I_x I_t \\ \sum I_y I_t \end{bmatrix}}_{A^T b}$$

Iterative refinement at same scale

- Iterative Lukas-Kanade Algorithm
  1. Estimate velocity at each pixel by solving Lucas-Kanade equations
  2. Warp I(t-1) towards I(t) using the estimated flow field

     - use image warping techniques
  3. Repeat until convergence

Coarse-to-fine optical flow estimation



When optical flow fails?

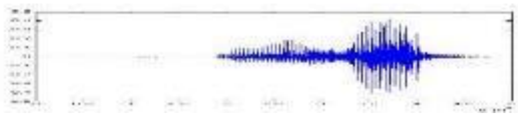
- In other words, in what situations does the displacement of pixel patches not represent physical movement of points in space?
- A uniform rotating
  - nothing seems to move, yet it is rotating
- Changing directions or intensities of lighting can make things seem to move
  - for example, if the specular highlight on a rotating sphere moves.

MACHINE LEARNING

Machine learning

Objective: Looking for a function!

- Speech Recognition

$f($  $) =$ "How are you"

- Image Recognition

$f($  $) =$ "Cat"

- Playing Go

$f($  $) =$ "5-5" (next move)

- Chatbot

$f($ "Hi" $) =$ "Hello"

(what the user said) (system response)

Machine learning tasks in vision

- Object **Classification**
  what object ?




http://pascallin.ecs.soton.ac.uk/challenges/VOC/

- Object **Detection**
  object or no-object ?

{people | vehicle | … intruder ….}

- Instance **Recognition** ?
  who (or what) is it ?


{face | vehicle plate| gait …. → biometrics}

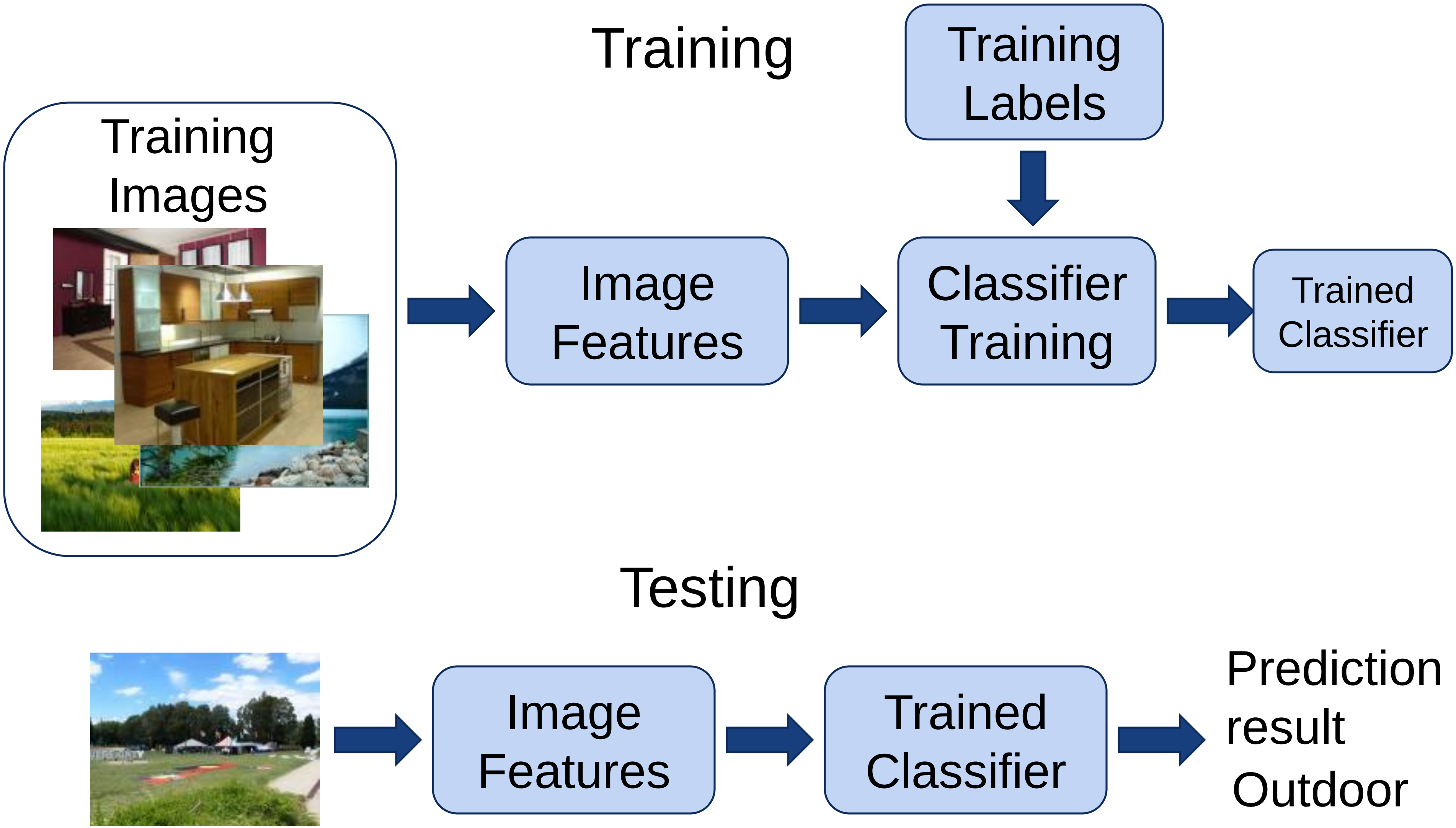
- **Sub-category** analysis
  which object type ?




{gender | type | species | age ……}

- Sequence **{ Recognition | Classification }** ?
  what is happening / occurring ?


Image classification



Test Image

Classical machine learning pipeline

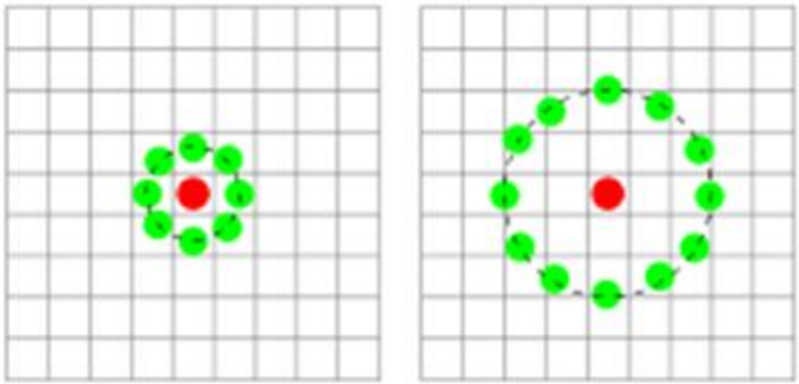
1. Select / develop features: SURF, HoG, SIFT, …
2. Add on top of this Machine Learning for multi-class recognition and train classifier

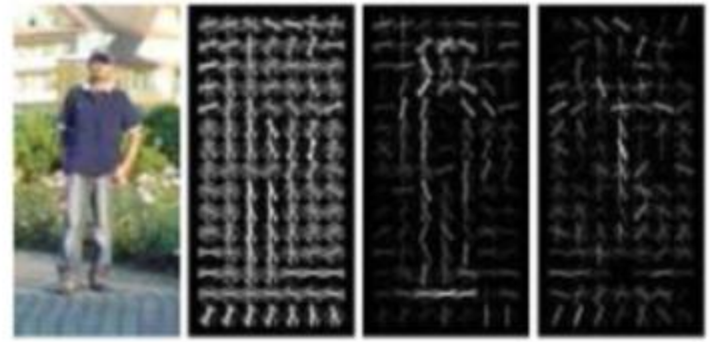
Feature Extraction: SIFT, HoG...

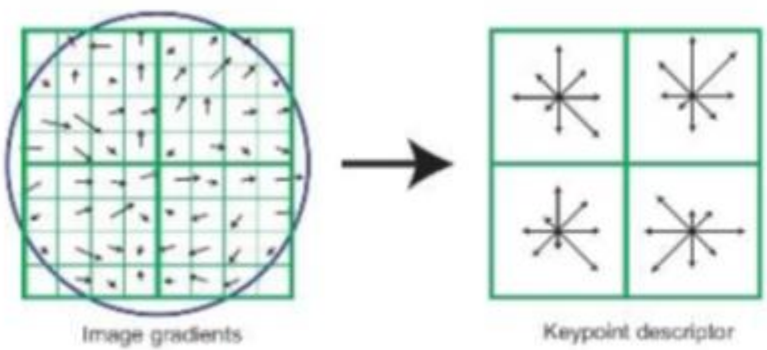
Detection, Classification Recognition

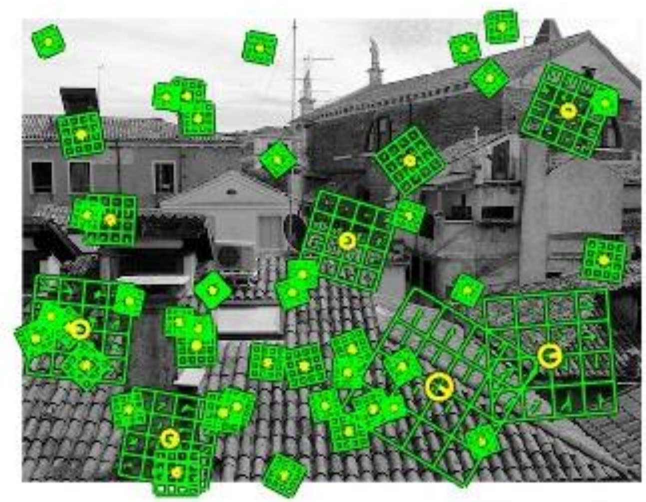
Classical computer vision feature definition is domain-specific and time-consuming

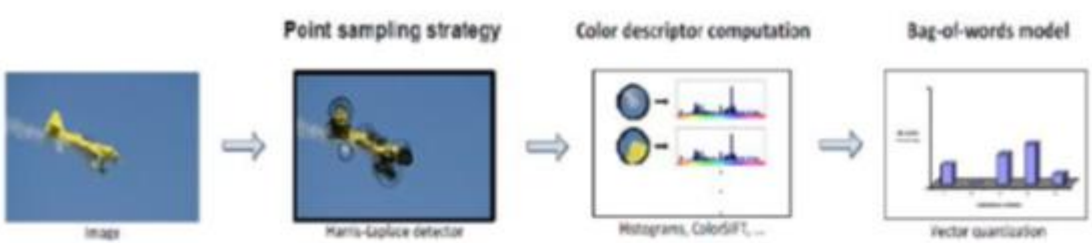
What are the right features?











What are the right features?

- Object: shape
  - Local shape info, shading, shadows, texture
- Scene: geometric layout
  - linear perspective, gradients, line segments
- Material properties: Color, texture
- Action: motion
  - Optical flow, tracked points



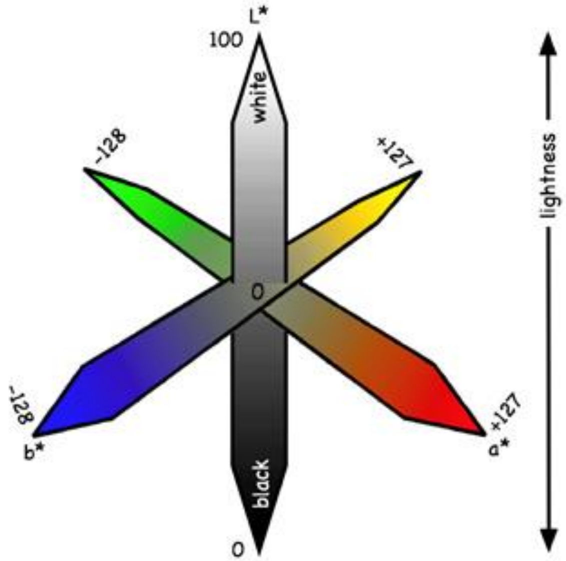
Histogram

- Histogram of an image provides the frequency of the brightness (intensity) value in the image.

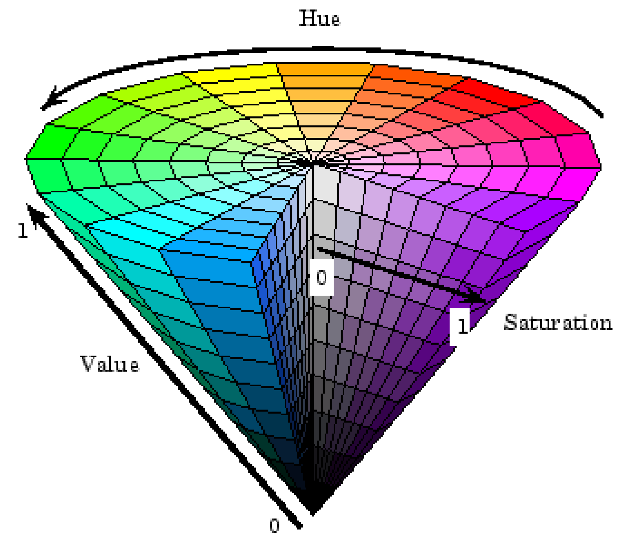
```
def histogram(im):
    h = np.zeros(255)
    for row in im.shape[0]:
        for col in im.shape[1]:
            val = im[row, col]
            h[val] += 1
```

Histogram

- Color

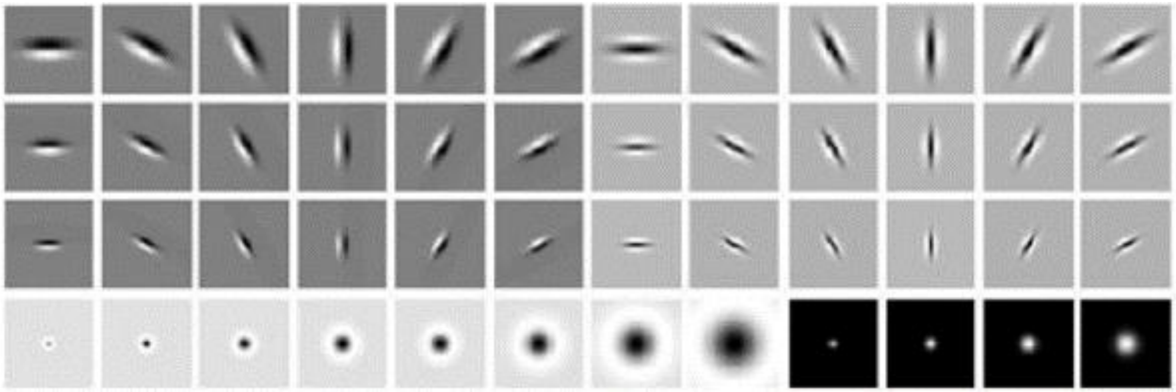


L*a*b* color space



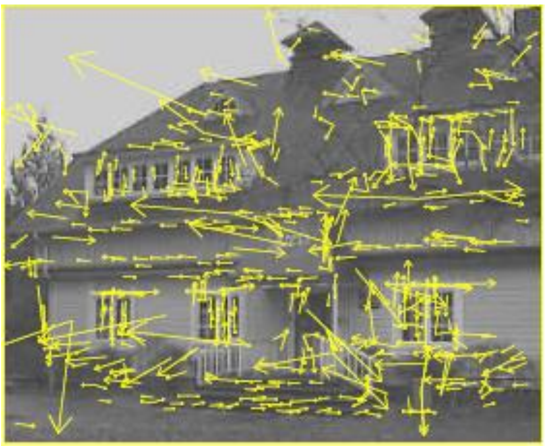
HSV color space

- Texture (filter banks)



Histogram

Gradients



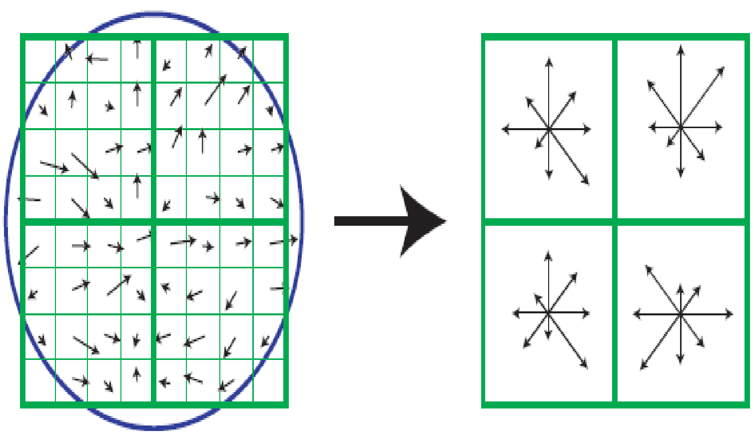
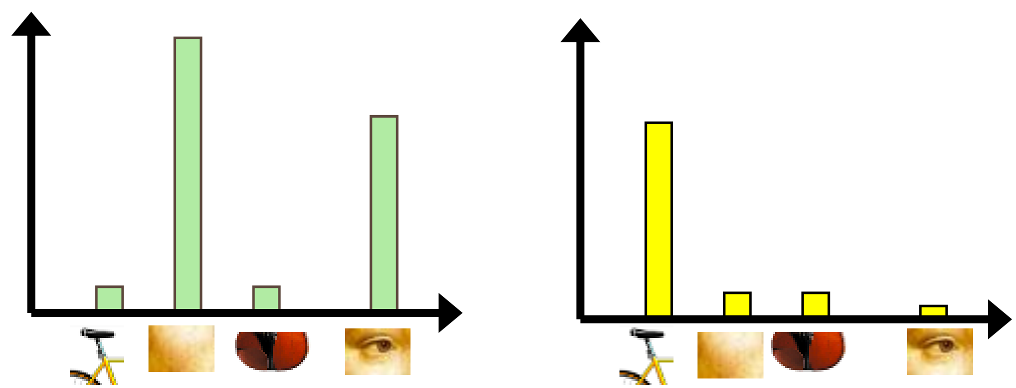


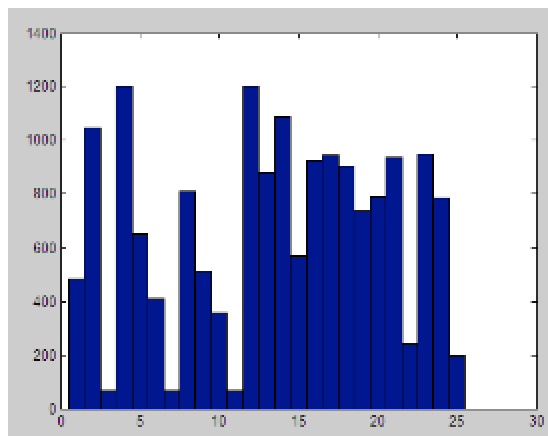
Image gradients

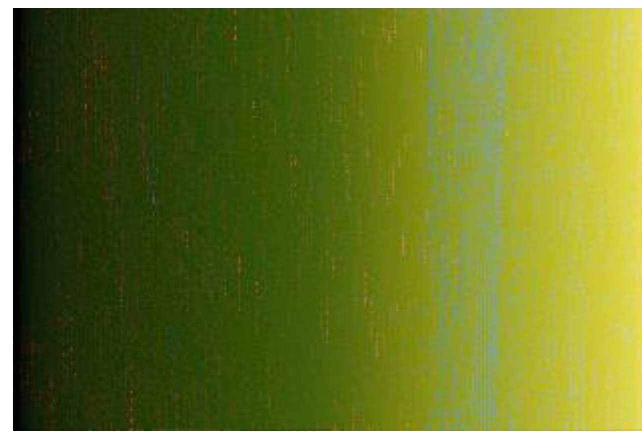
Keypoint descriptor

“Bag of visual words”



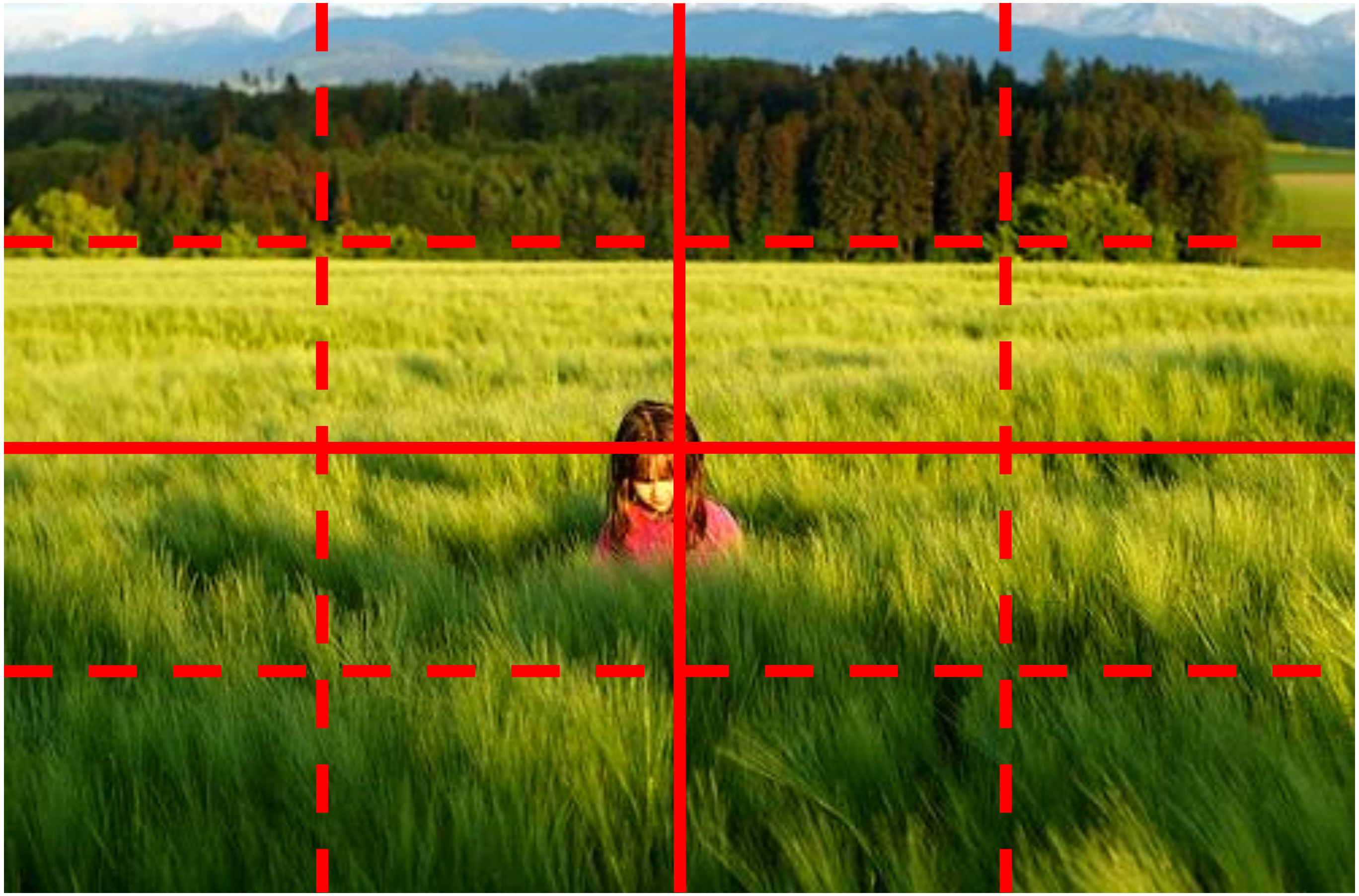
Spatial layout





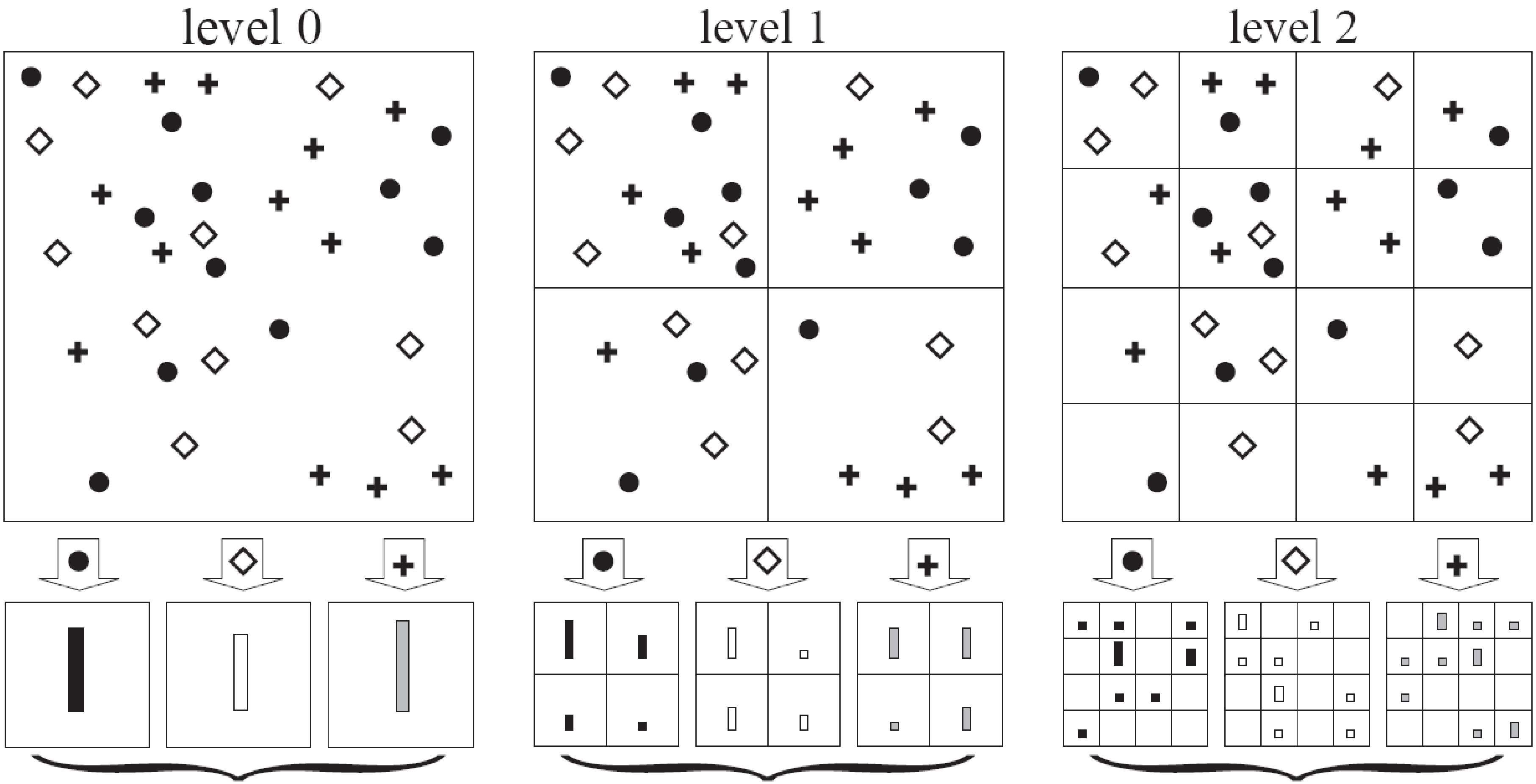
All of these images have the same color histogram

Spatial pyramid



Compute histogram in each spatial bin

Spatial pyramid

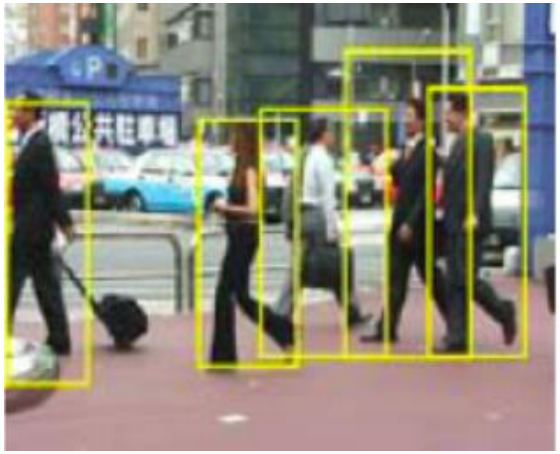


OBJECT DETECTION

Category vs. instance recognition

Category:

- Find all the people
- Find all the buildings
- Often within a single image
- Often ‘sliding window’

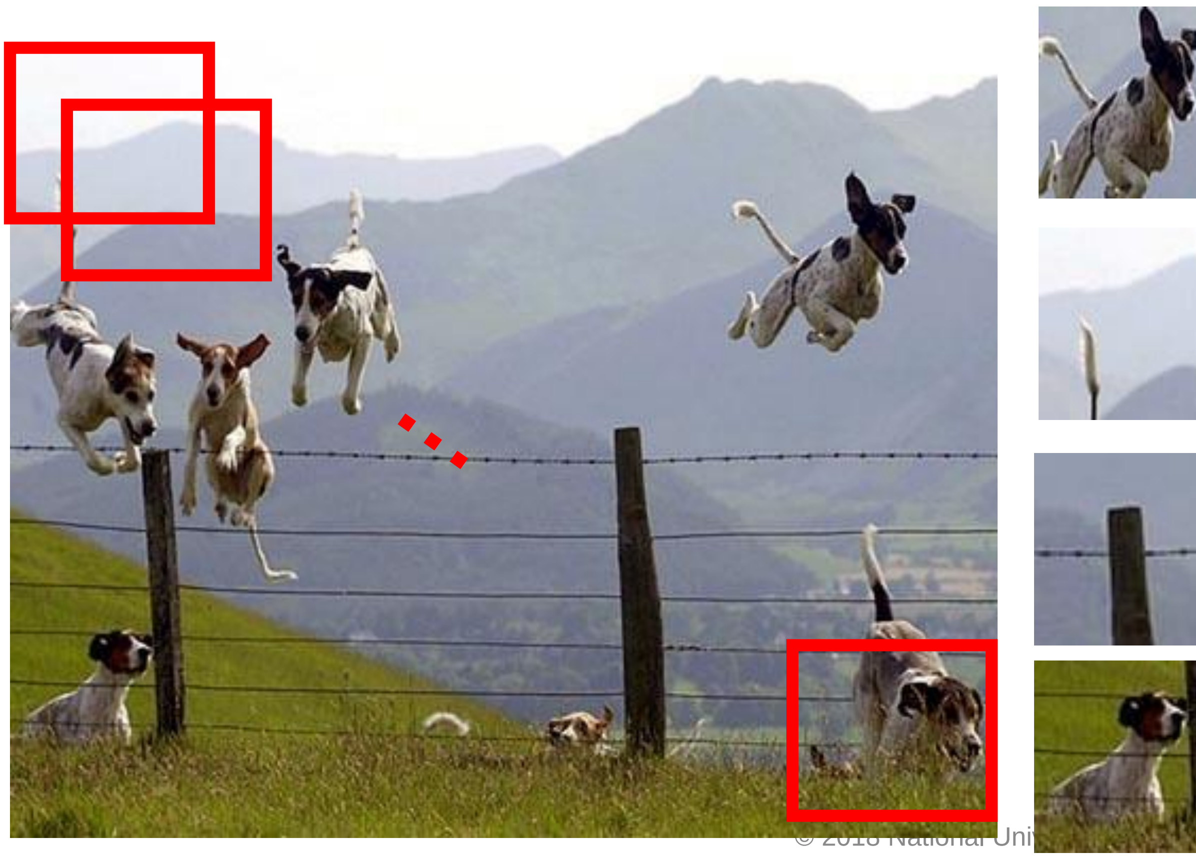


Instance:

- Is this face James?
- Find this specific famous building
- Often within a database of images

Object category detection

- Focus on object search: “Where is it?”
- Build templates that quickly differentiate object patch from background patch



Object or **Non-Object?**

Challenges in object detection

Illumination

Object pose

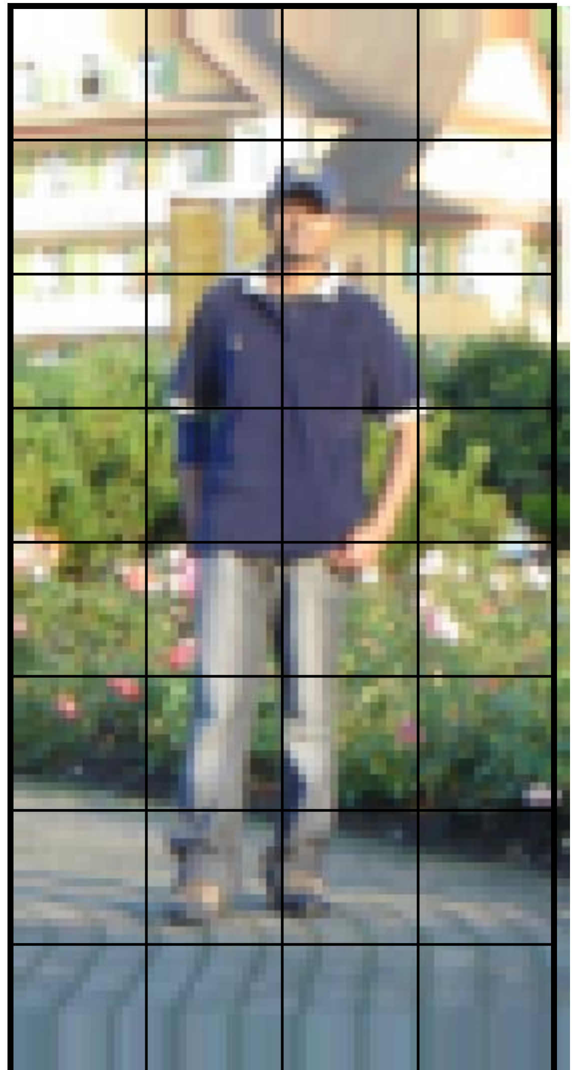
‘Clutter’

Occlusions

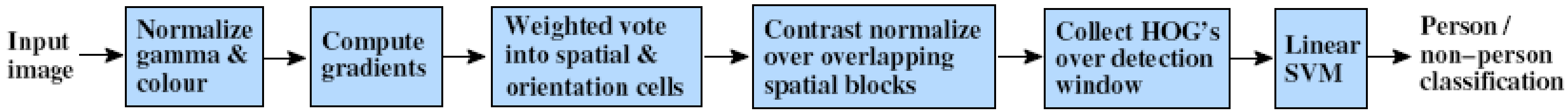
Intra-class appearance

Viewpoint

Input image → Normalize gamma & colour → Compute gradients → Weighted vote into spatial & orientation cells → Contrast normalize over overlapping spatial blocks → Collect HOG's over detection window → Linear SVM → Person / non–person classification



Navneet Dalal and Bill Triggs, Histograms of Oriented Gradients for Human Detection, CVPR05

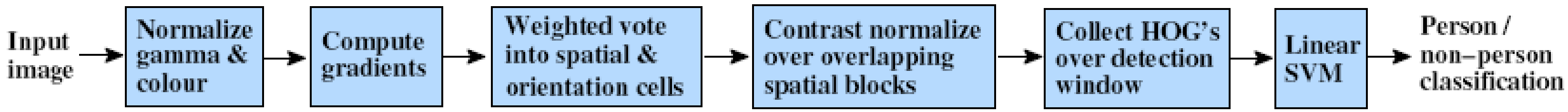


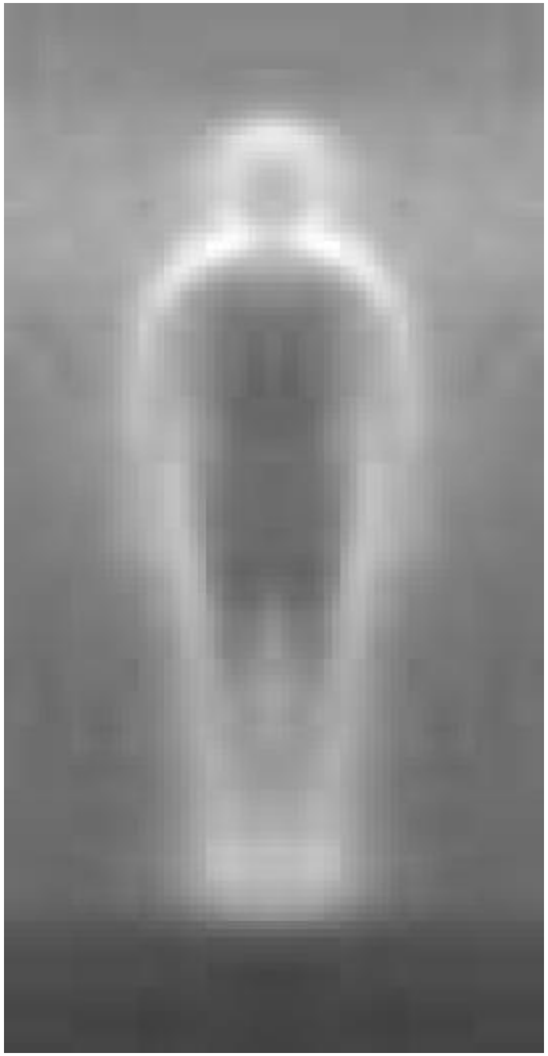
- Tested with
  - RGB
  - LAB
  - Grayscale

Slightly better performance vs. grayscale

- Gamma Normalization and Compression
  - Square root
  - Log

Very slightly better performance vs. no adjustment





-1	0	1

centered

-1	1

uncentered

1	-8	0	8	-1

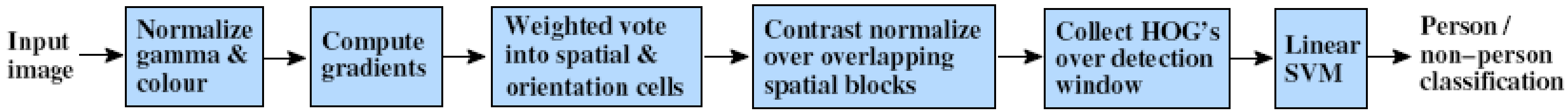
cubic-corrected

0	1
-1	0

diagonal

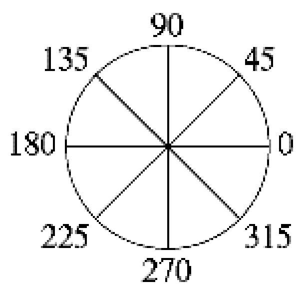
-1	0	1
-2	0	2
-1	0	1

Sobel

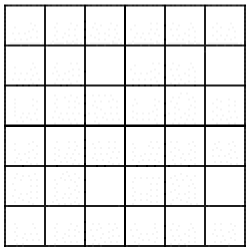


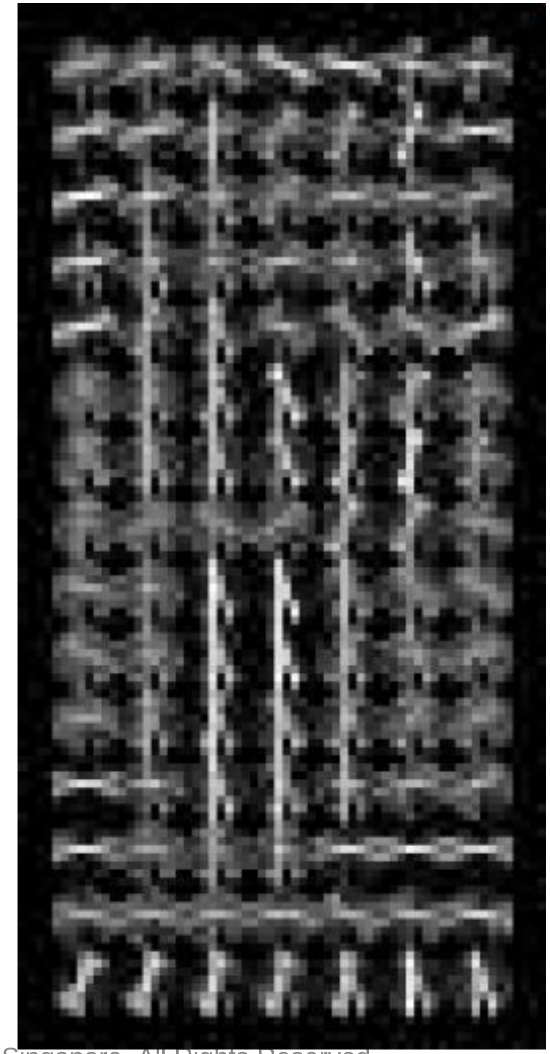
- Histogram of gradient orientations

Orientation: 9 bins (for unsigned angles)

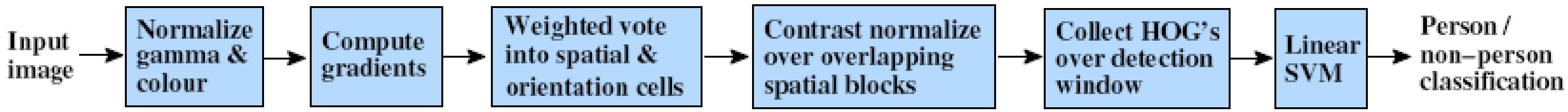


Histograms in 8x8 pixel cells

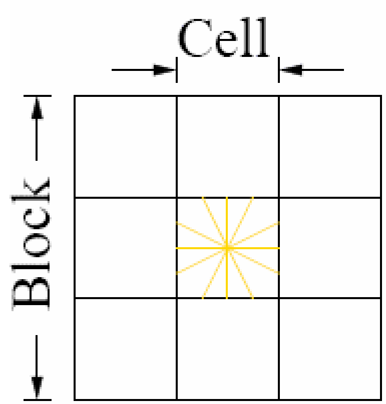




- Votes weighted by magnitude

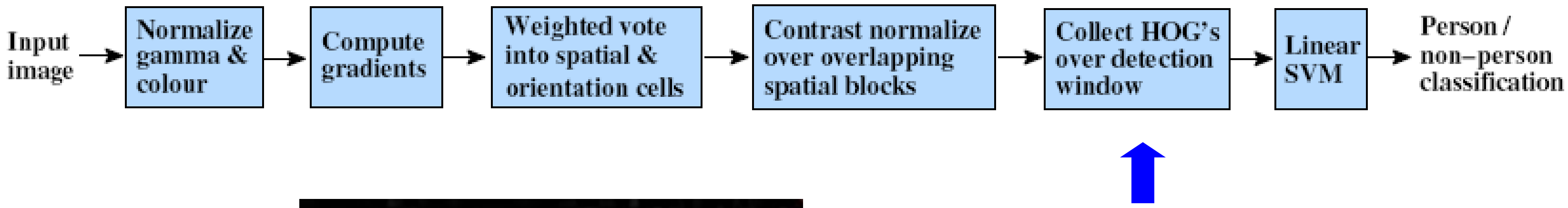


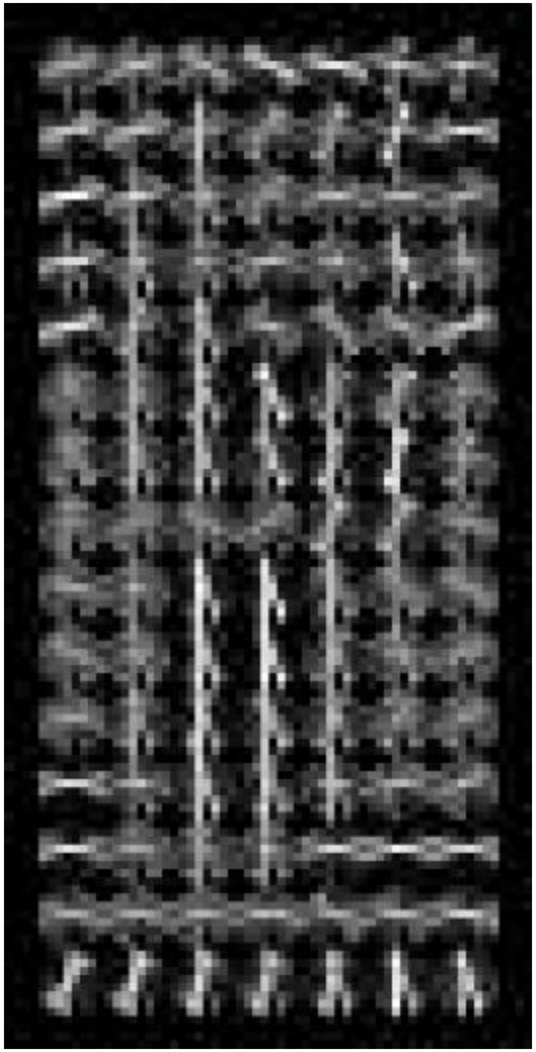
R-HOG



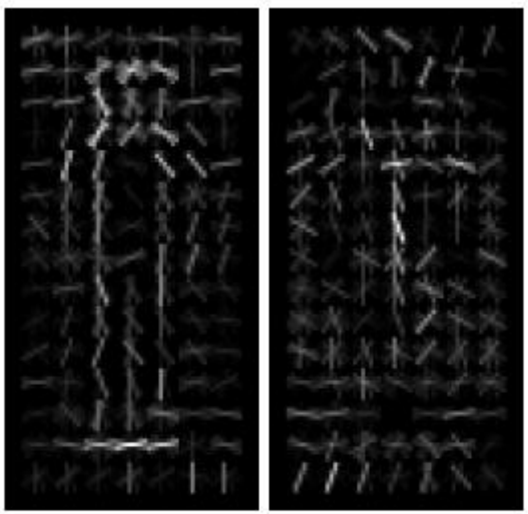
Normalize with respect to surrounding cells

$$L2 - norm : v \longrightarrow v / \sqrt{||v||_2^2 + \epsilon^2}$$


Input image
Normalize gamma & colour
Compute gradients
Weighted vote into spatial & orientation cells
Contrast normalize over overlapping spatial blocks
Collect HOG's over detection window
Linear SVM
Person / non–person classification

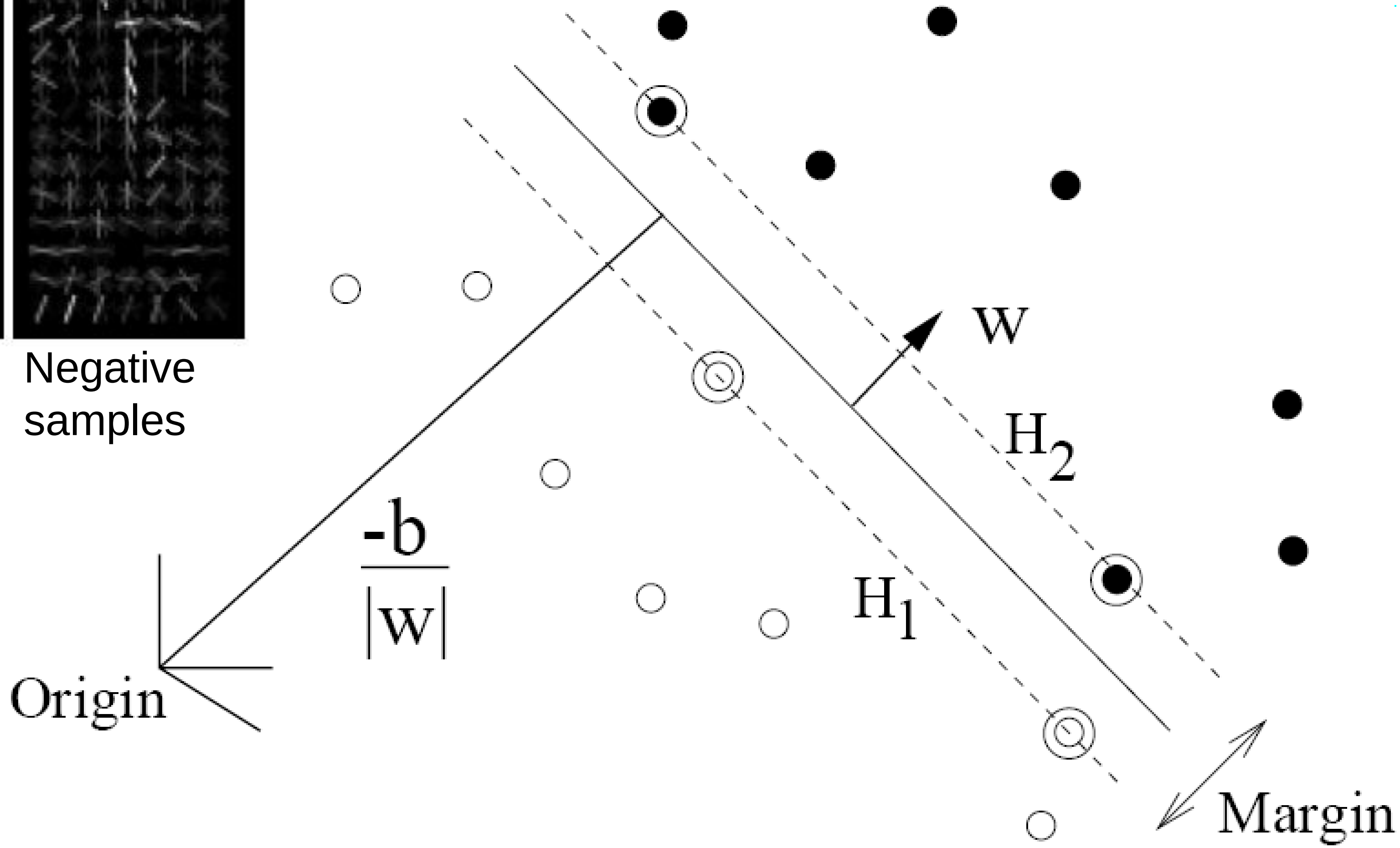


Input image → Normalize gamma & colour → Compute gradients → Weighted vote into spatial & orientation cells → Contrast normalize over overlapping spatial blocks → Collect HOG's over detection window → Linear SVM → Person / non−person classification



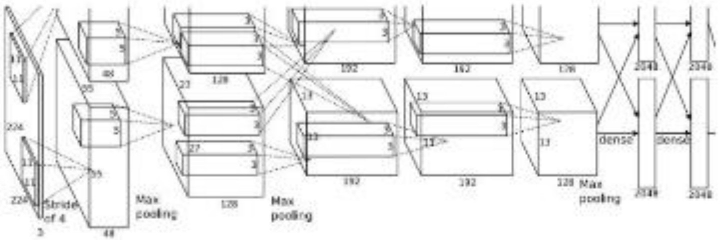
Positive samples

Negative samples



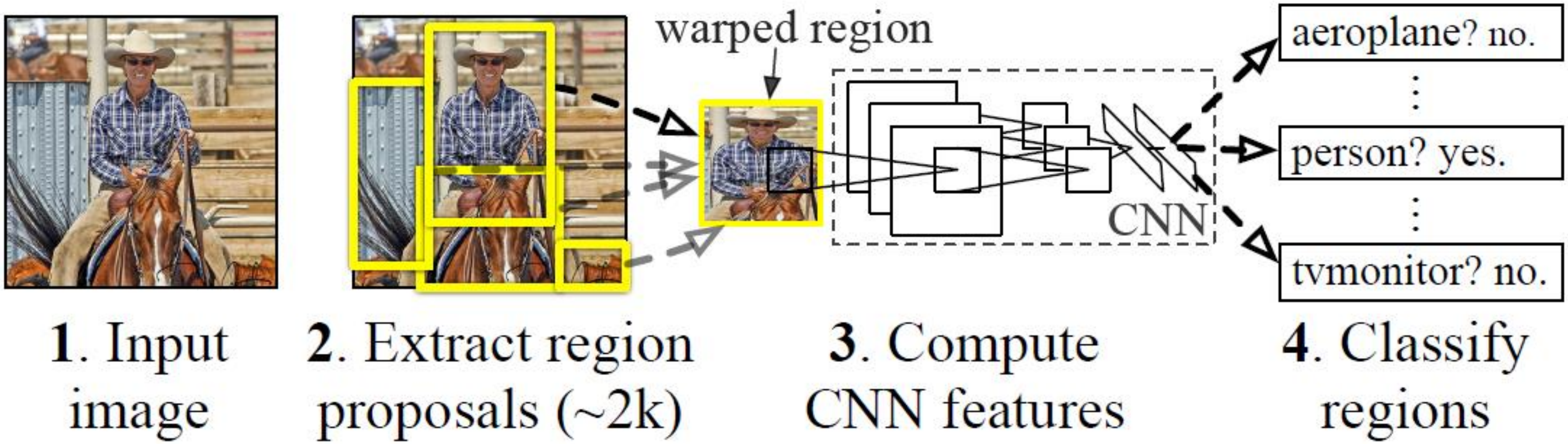
CNN as feature extractor

Apply a CNN to many different crops of the image, CNN classifies each crop as object or background



Dog? NO
Cat? NO
Background? YES

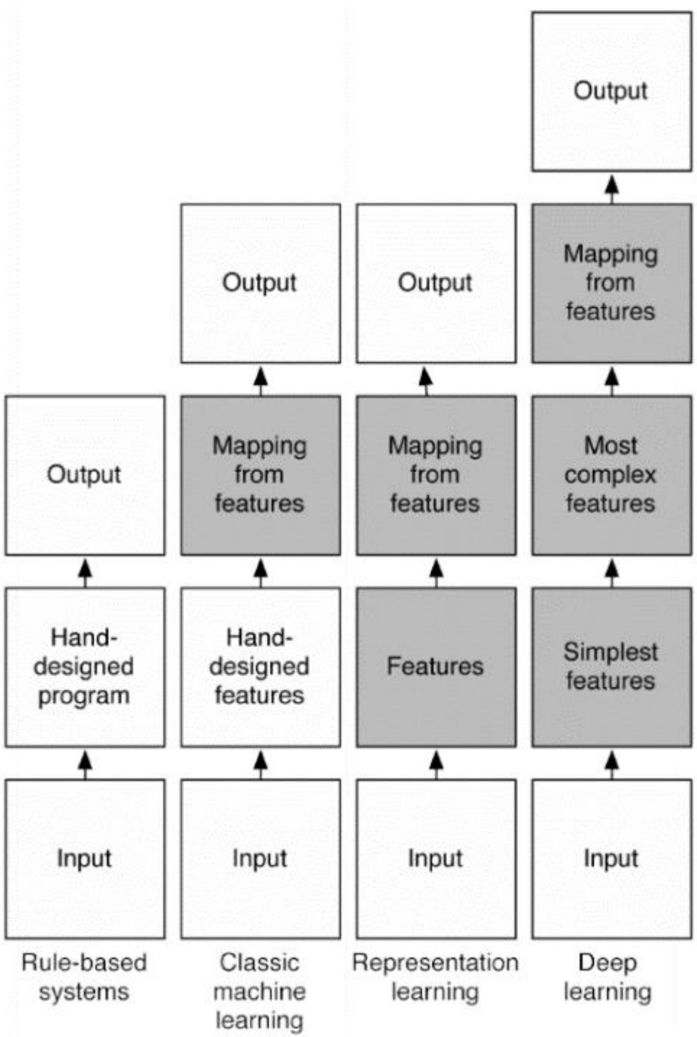
CNN as feature extractor

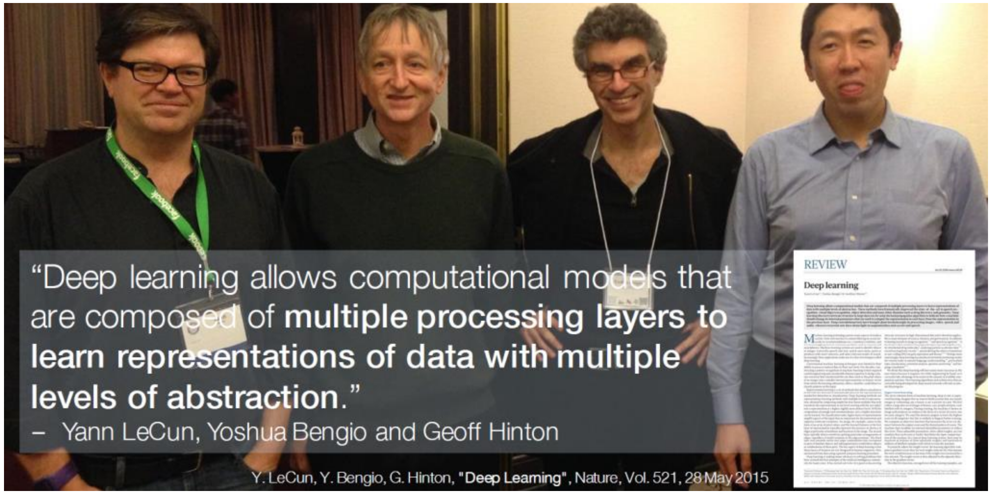


- Replace sliding windows with “selective search” region proposals
- Extract rectangles around regions
- Extract features with fine-tuned CNN (that was initialized with network trained on ImageNet before training)
- Classify last layer of network features with SVM, refine bounding box localization (bbox regression) simultaneously

http://arxiv.org/pdf/1311.2524.pdf

Deep learning pipeline





"having had countless ConvNet papers rejected, published and ignored, and occasionally paid attention to, for over 15 years"
-- Yann Lecun

Summary

- Introduction
- Feature representation: Motion
- Feature representation: Frequency-domain
- Classification and object detection

Thank You!

Dr TIAN Jing
Email: tianjing@nus.edu.sg