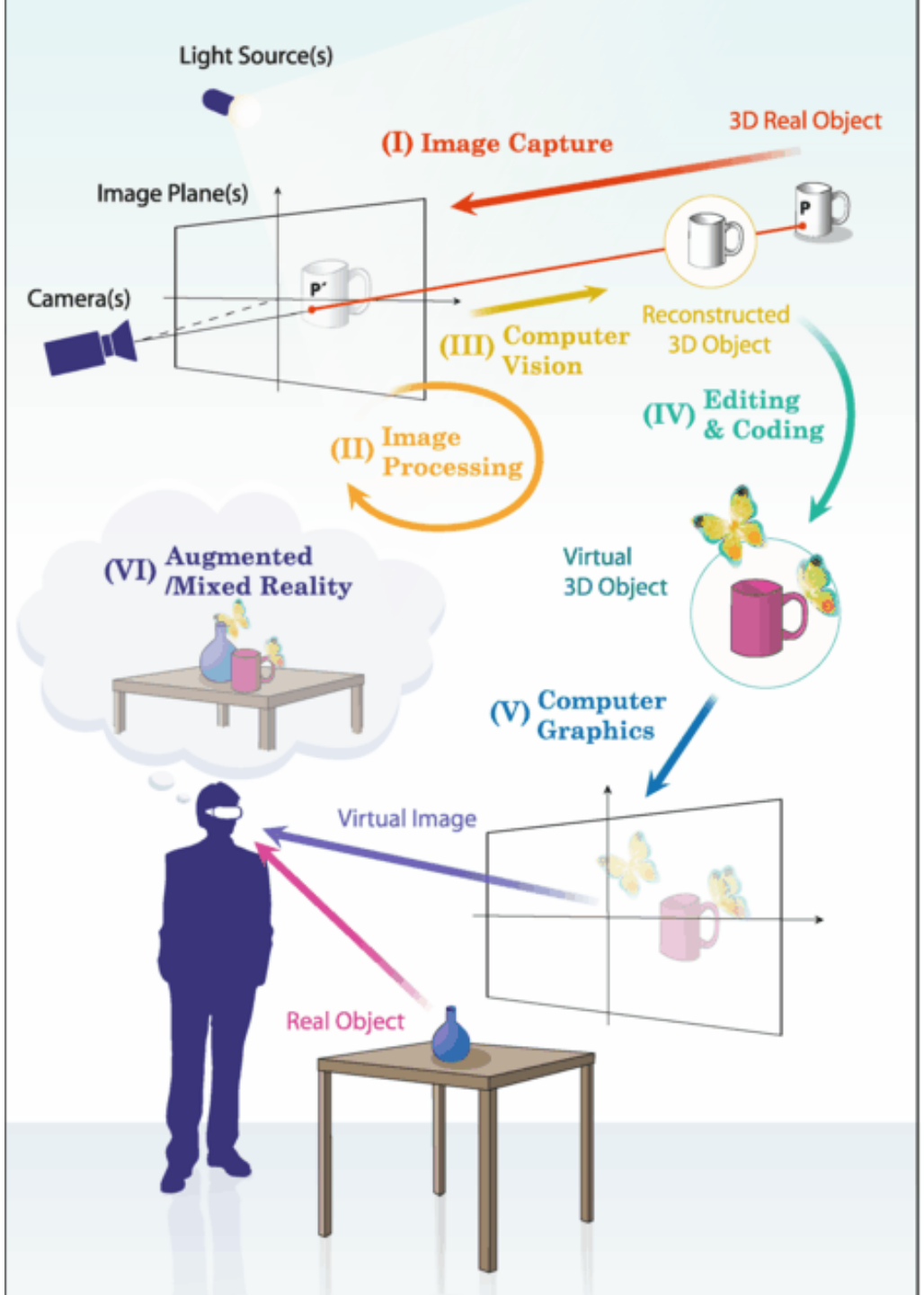
Visión Computacional en Robótica

Rodrigo Verschae, Universidad de O'Higgins

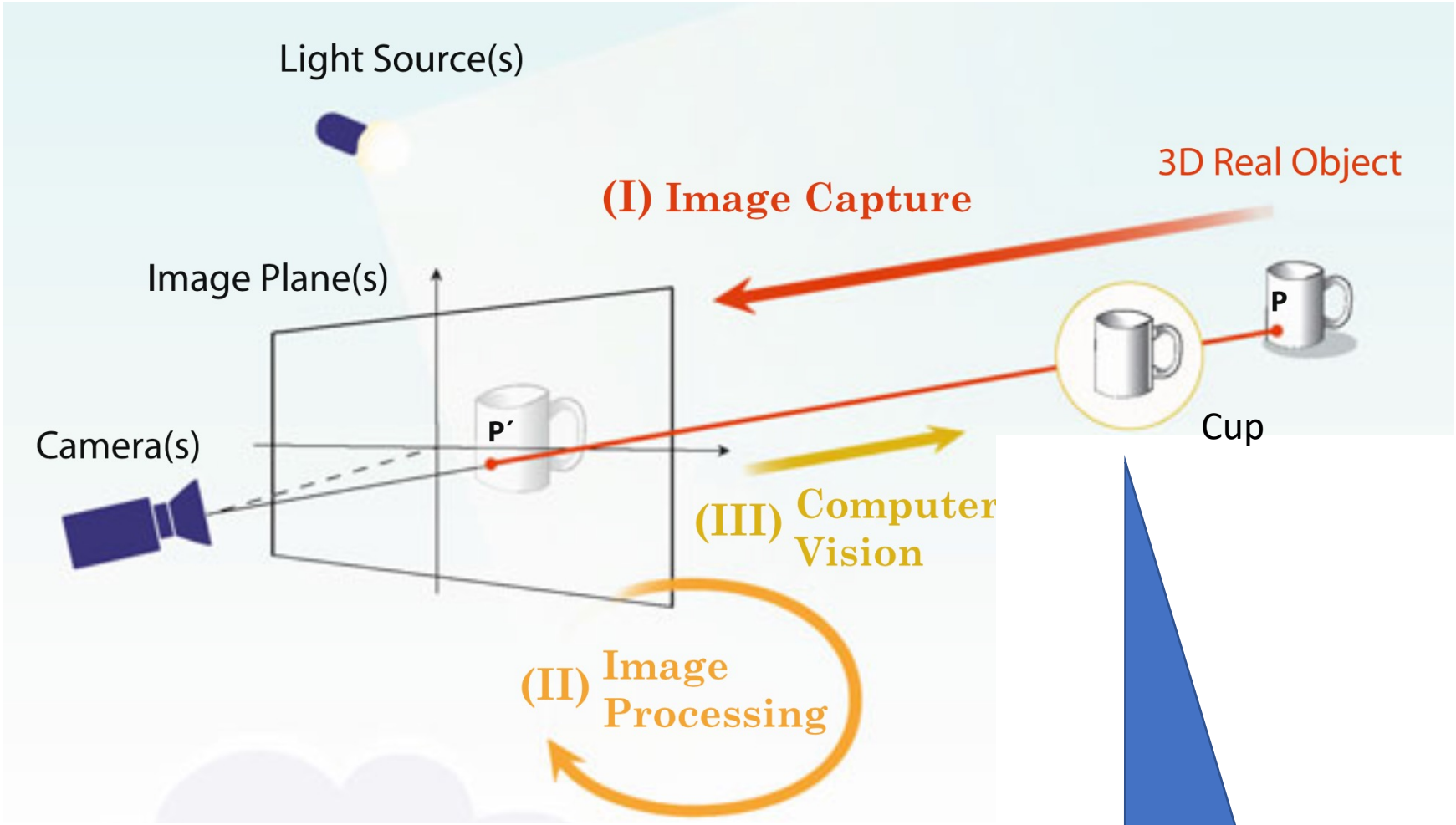
Email: rodrigo@verschae.org; rodrigo.verschae@uoh.cl

Web: rodrigo.verschae.org

World of visual information media



World of visual information media



Scene Understating

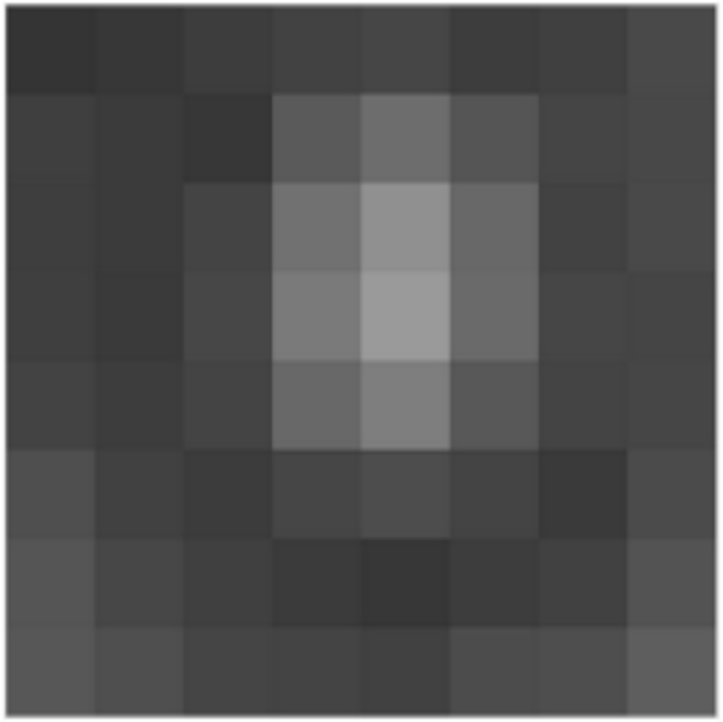
¿Que es una imagen?

Que es una imagen?

Pixel

Que es una imagen?

Pixel
(Ejemplo, escala de grices 256 valores)

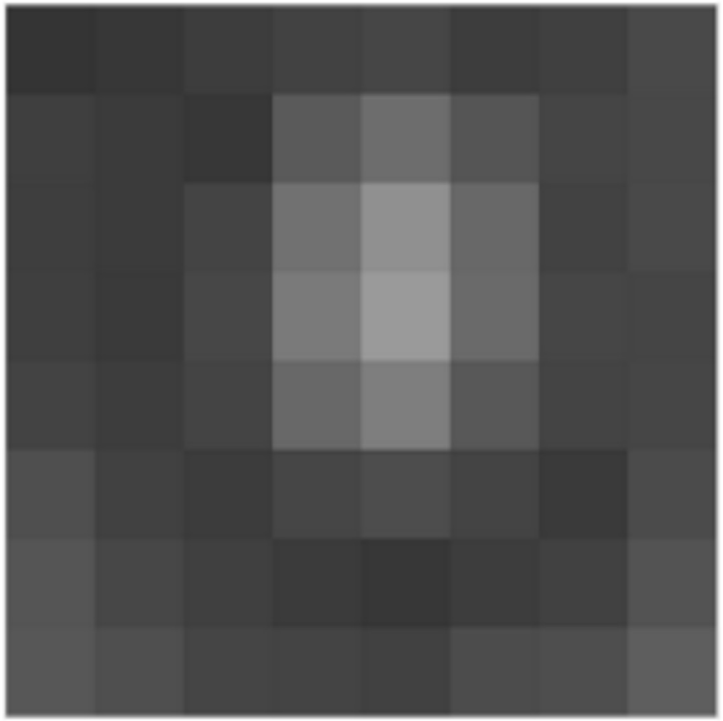


$$\begin{bmatrix} 52 & 55 & 61 & 66 & 70 & 61 & 64 & 73 \\ 63 & 59 & 55 & 90 & 109 & 85 & 69 & 72 \\ 62 & 59 & 68 & 113 & 144 & 104 & 66 & 73 \\ 63 & 58 & 71 & 122 & 154 & 106 & 70 & 69 \\ 67 & 61 & 68 & 104 & 126 & 88 & 68 & 70 \\ 79 & 65 & 60 & 70 & 77 & 68 & 58 & 75 \\ 85 & 71 & 64 & 59 & 55 & 61 & 65 & 83 \\ 87 & 79 & 69 & 68 & 65 & 76 & 78 & 94 \end{bmatrix}$$

Cuantas imágenes hay?

Imagen de 8x8 pixeles, escala de grices (256 valores):

$$256*256*\ldots*256 = 256^{8*8} \sim 10^{150}$$



$$\begin{bmatrix} 52 & 55 & 61 & 66 & 70 & 61 & 64 & 73 \\ 63 & 59 & 55 & 90 & 109 & 85 & 69 & 72 \\ 62 & 59 & 68 & 113 & 144 & 104 & 66 & 73 \\ 63 & 58 & 71 & 122 & 154 & 106 & 70 & 69 \\ 67 & 61 & 68 & 104 & 126 & 88 & 68 & 70 \\ 79 & 65 & 60 & 70 & 77 & 68 & 58 & 75 \\ 85 & 71 & 64 & 59 & 55 & 61 & 65 & 83 \\ 87 & 79 & 69 & 68 & 65 & 76 & 78 & 94 \end{bmatrix}$$

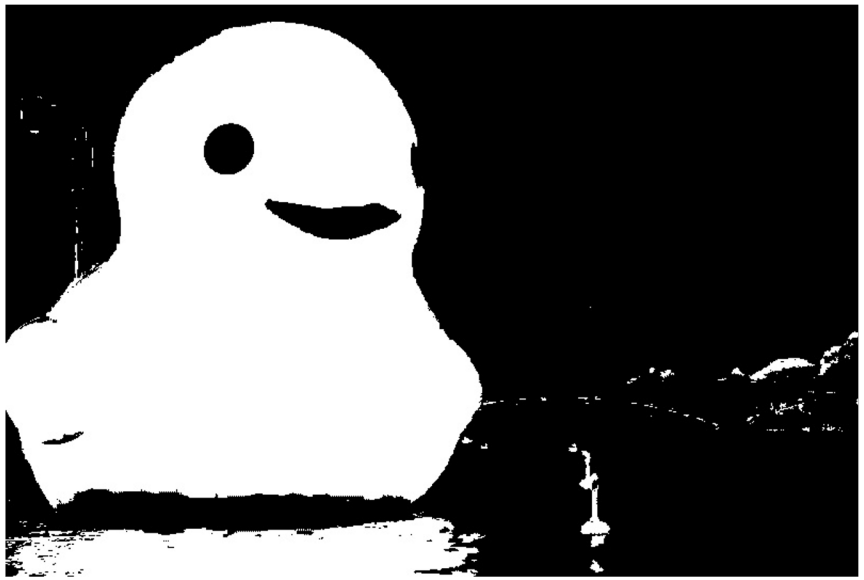
Que es una imagen?

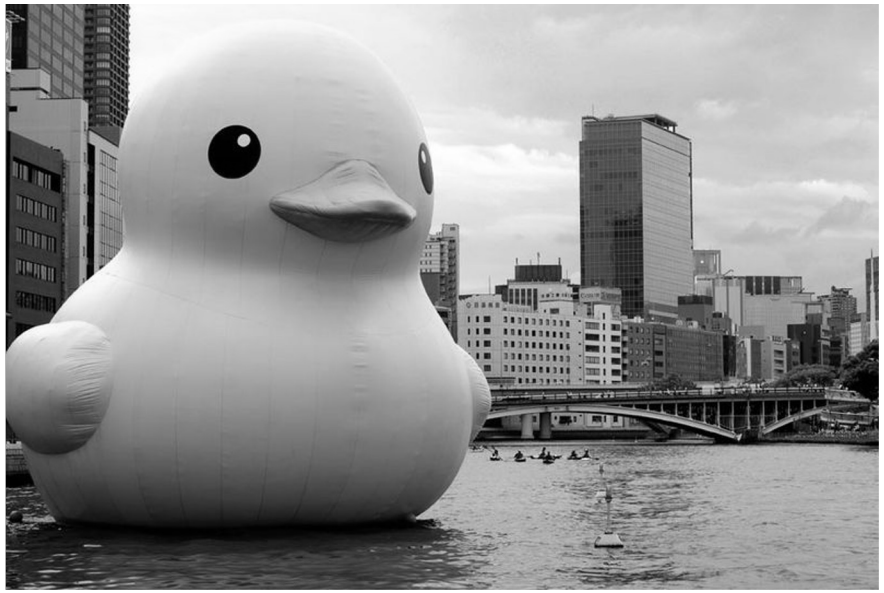
Pixel
(Ejemplo, RGB, 256 valores cada canal)
R = Red, G = Green, B = Blue

54	58	255	8	0
45	0	78	51	100	74
85	47	34	185	207	21	36
22	20	148	52	24	147	123
52	36	250	74	214	278	41
	158	0	78	51	247	255
		72	74	136	251	74

Tipos de imagen

- Binaria
- Escala de grises
- Color
- ...





Espacios de colores

- RGB (Red, Green, Blue)

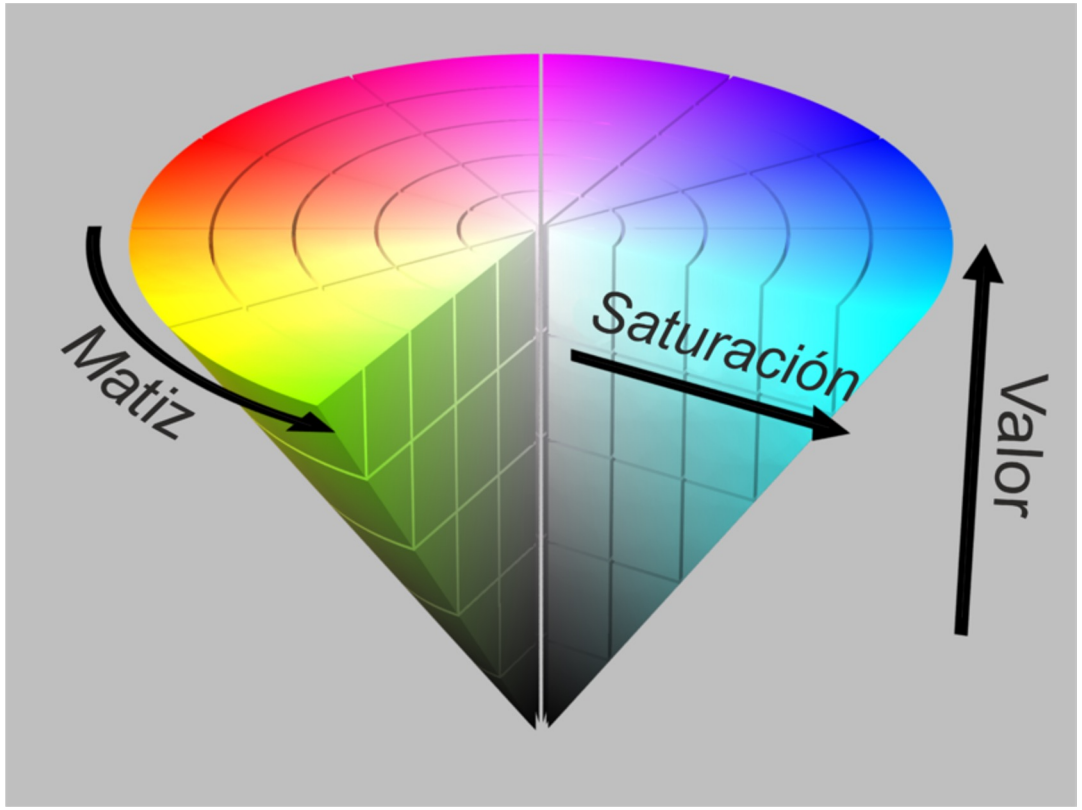
24 bits (8 bit o 256 niveles de color por canal)
256×256×256 ≈ 16,7 millones de colores

- (0, 0, 0) es negro
- (255, 255, 255) es blanco
- (255, 0, 0) es rojo
- (0, 255, 0) es verde
- (0, 0, 255) es azul
- (255, 255, 0) es amarillo

Espacios de colores

- HSV
  - Matiz -> Color (0 a 360)

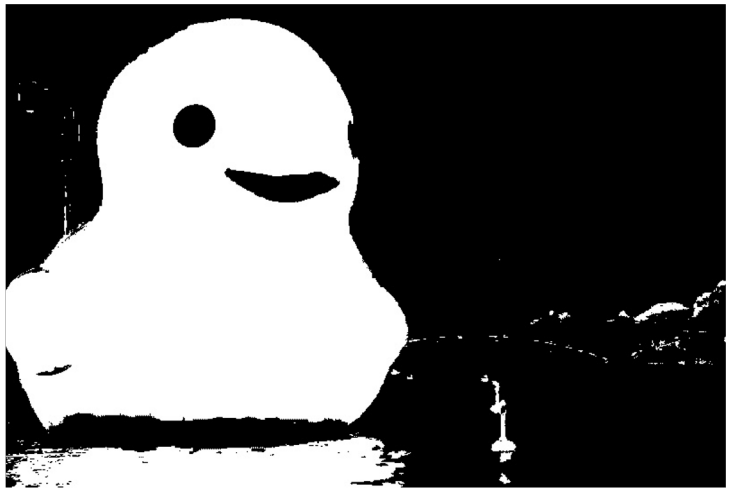
matiz 100% puro	75% de saturación	saturación media	25% de saturación	0 de saturación



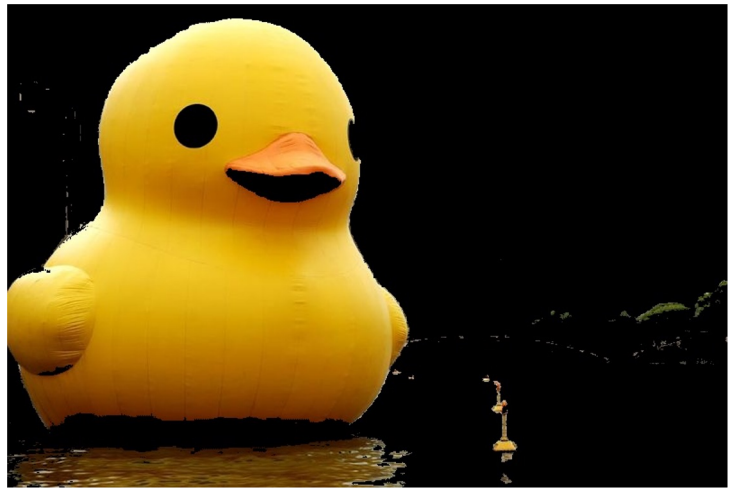
Comando simple: Mascara

```
image_out = cv2.bitwise_and(image, image, mask= mask)
```

image



mask

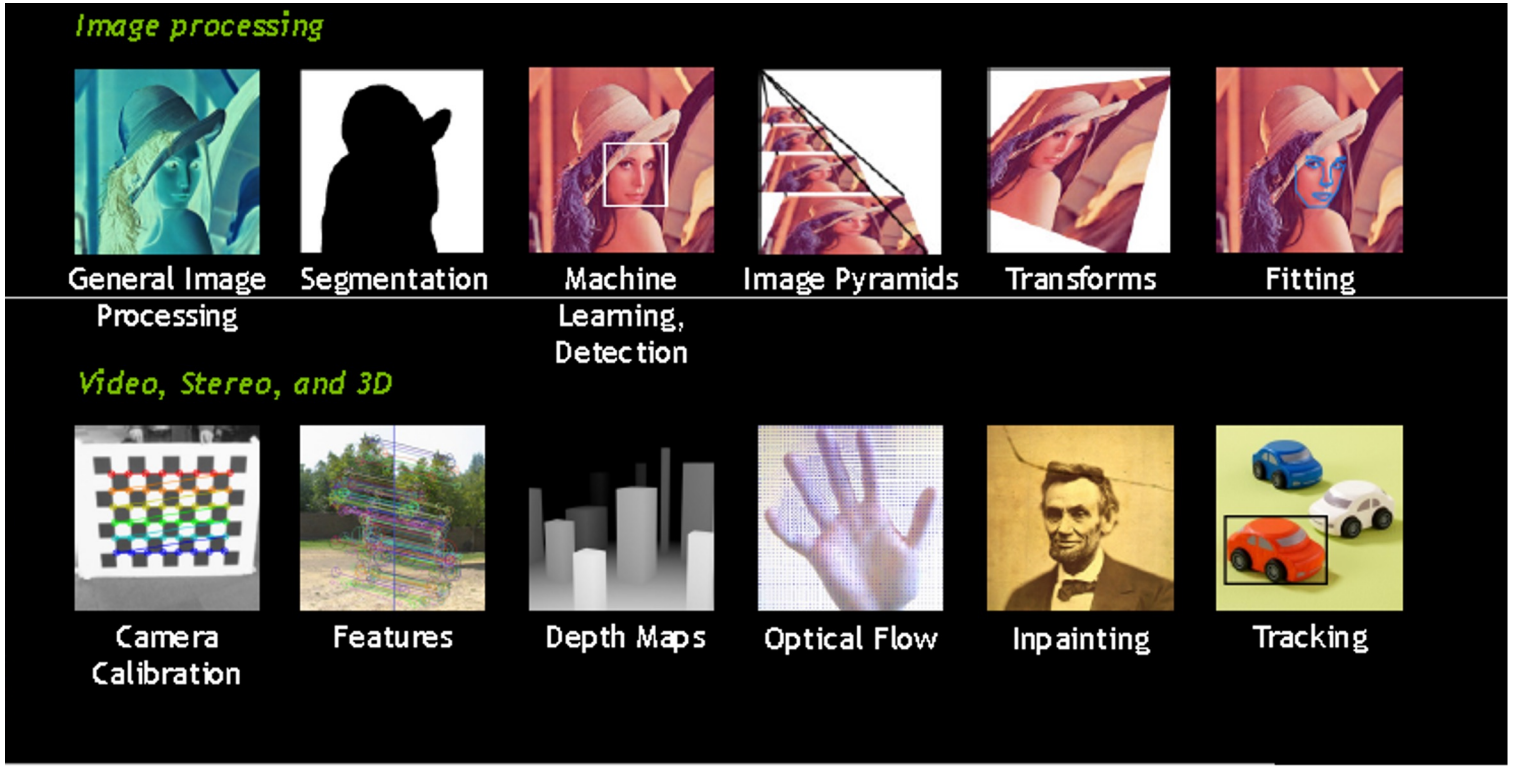


image_out

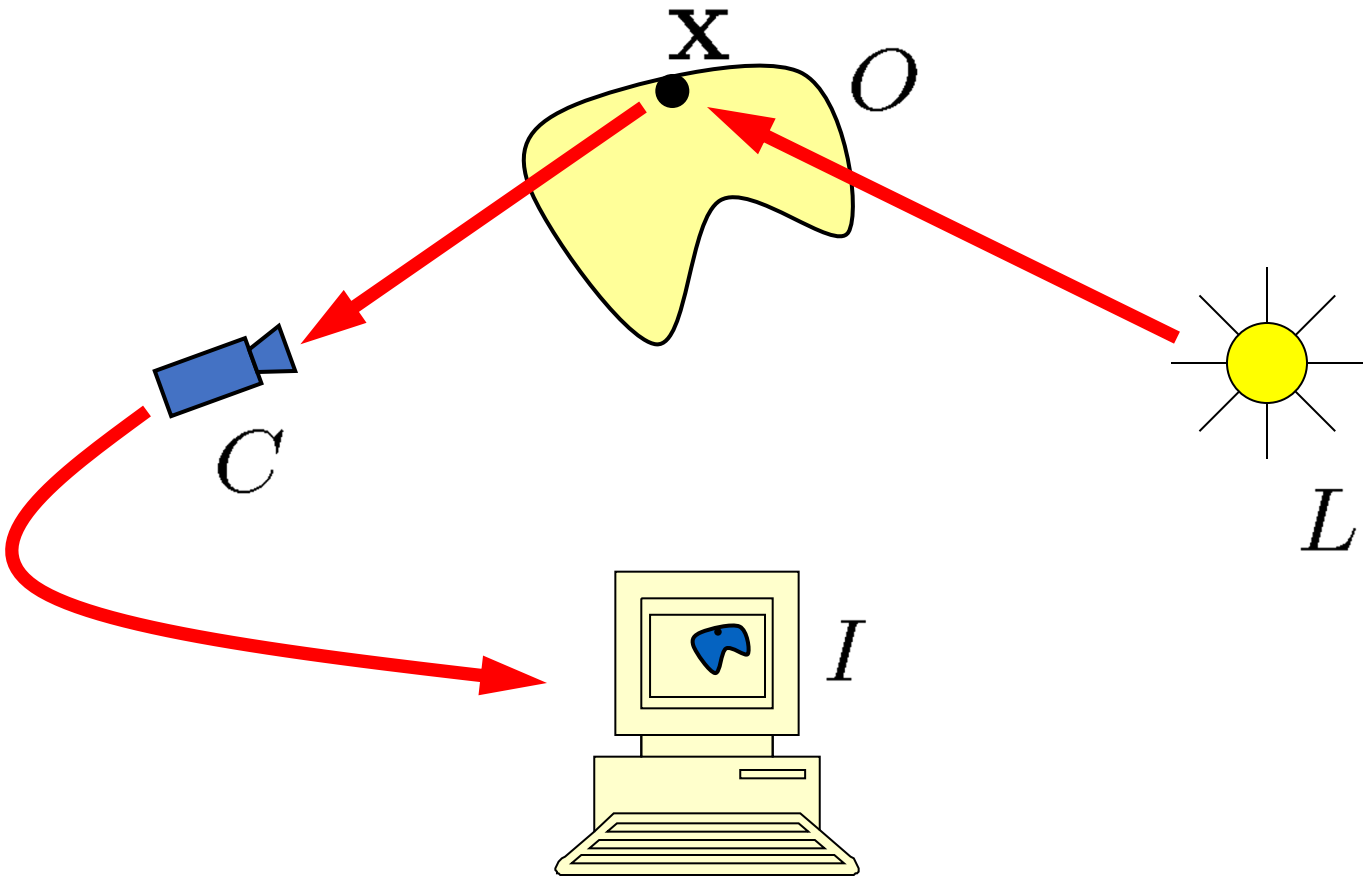
OpenCV

Librería de visión por computador de código abierto

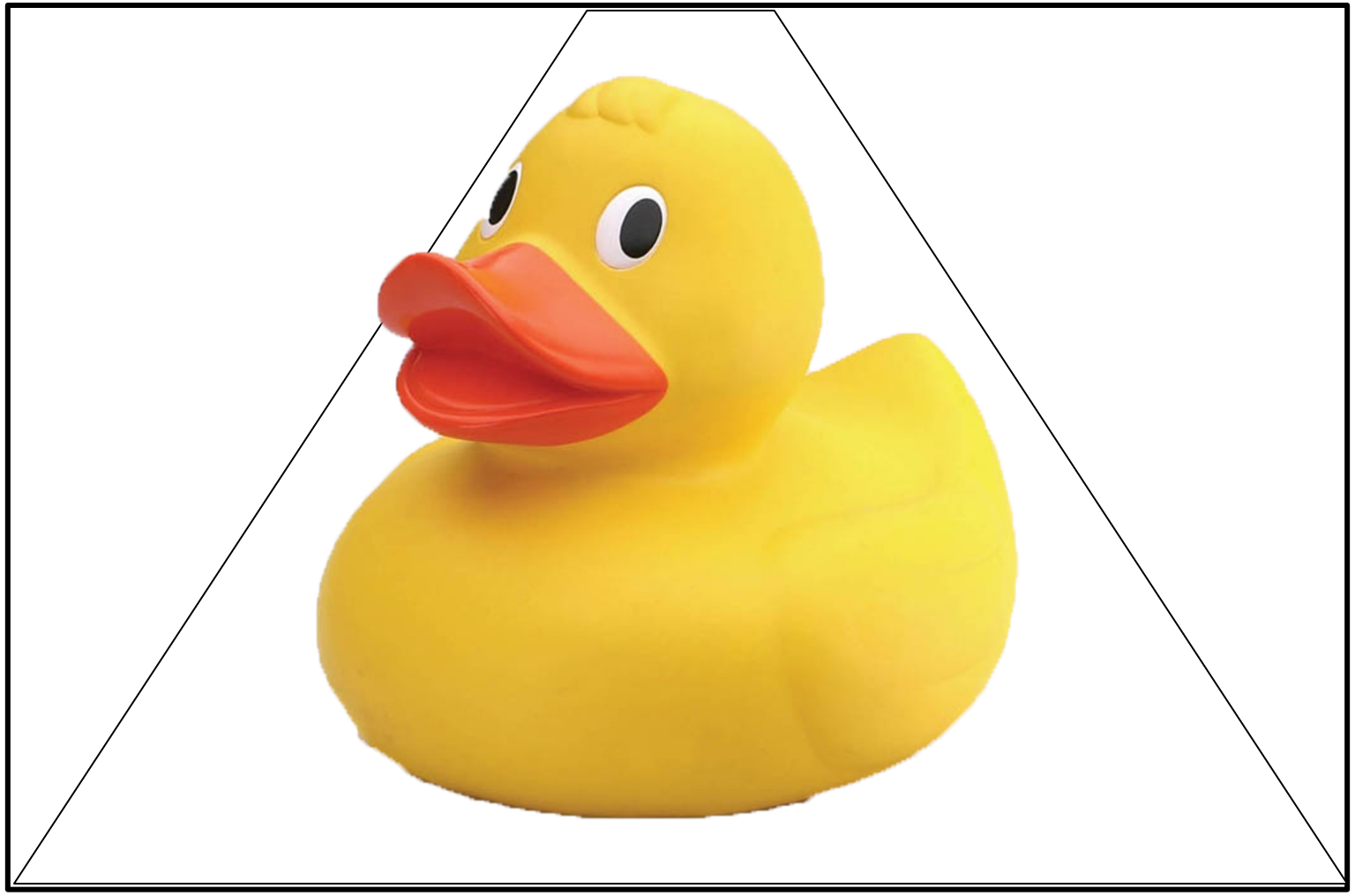
Lenguajes : C, C++, Python, Ruby, Matlab
Compatible con Linux, Windows y Mac OS X.

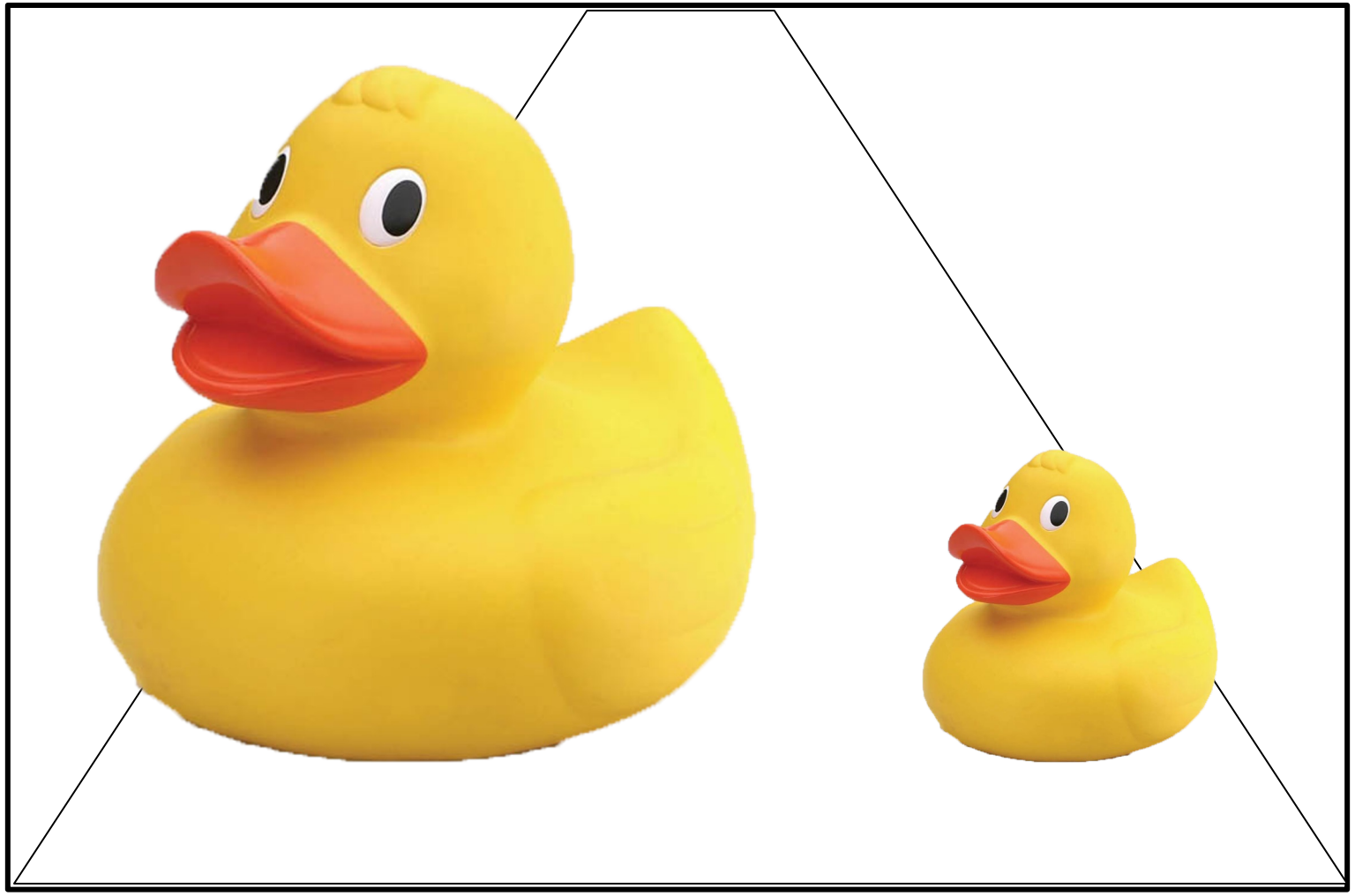


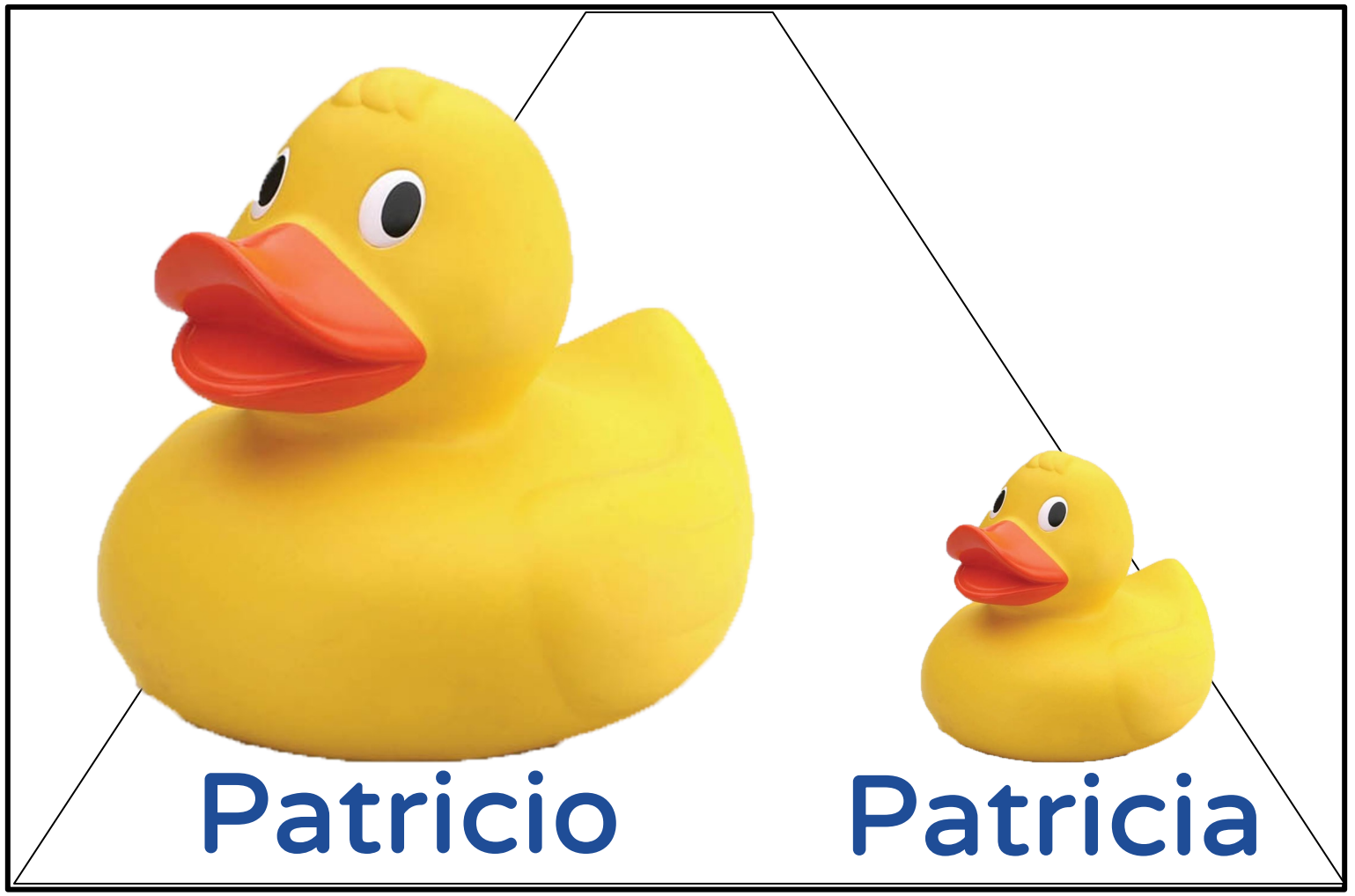
Formación de una imagen



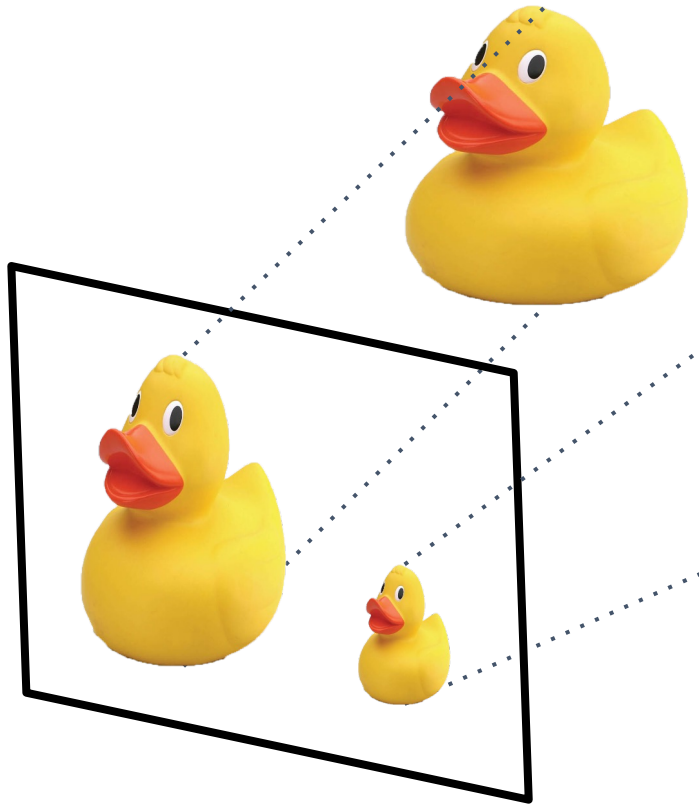
Cámaras

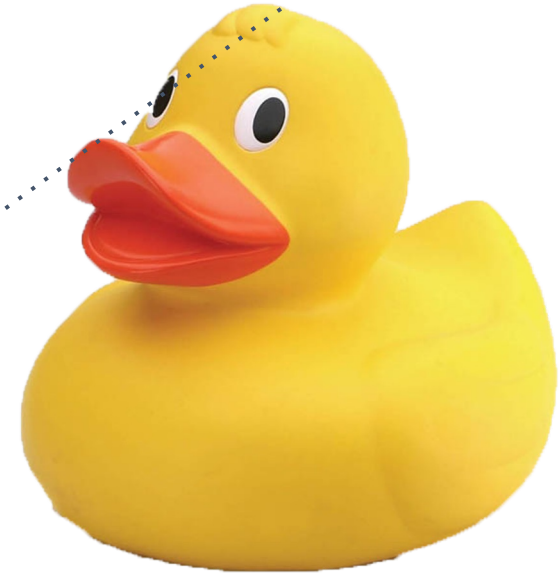





Patricio
Patricia

Patricio

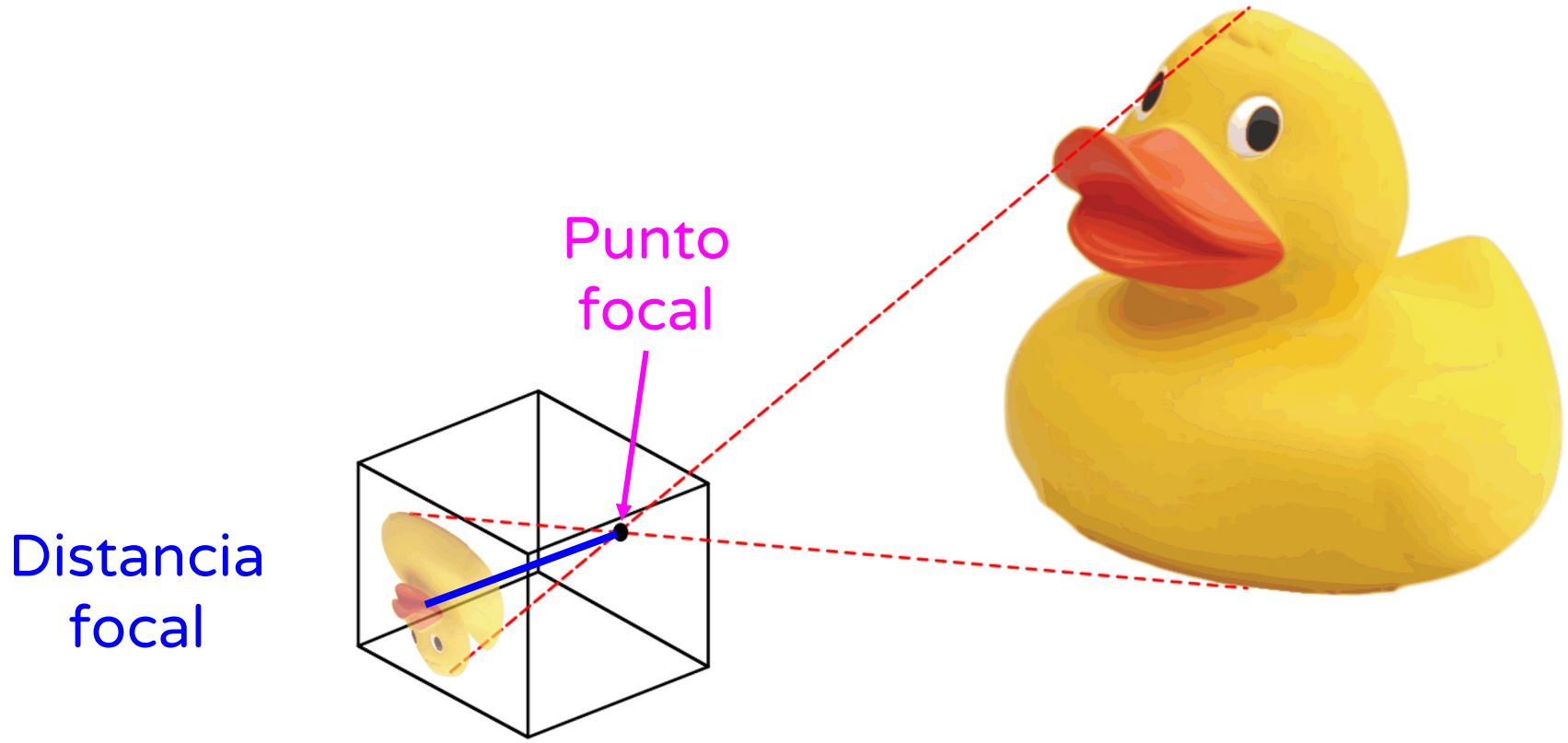




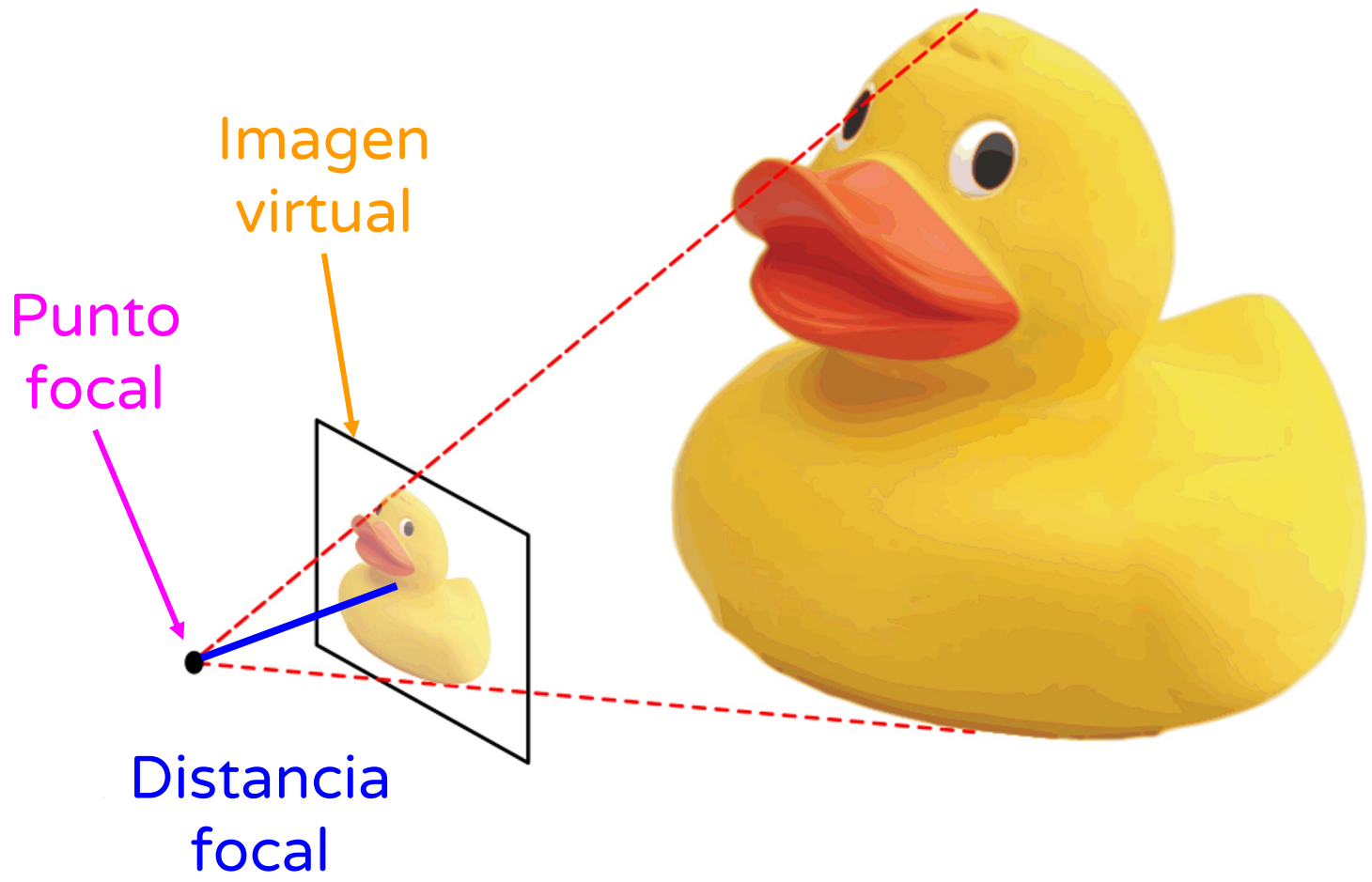
Patricia

Cámara pinhole

También se conoce como cámara oscura

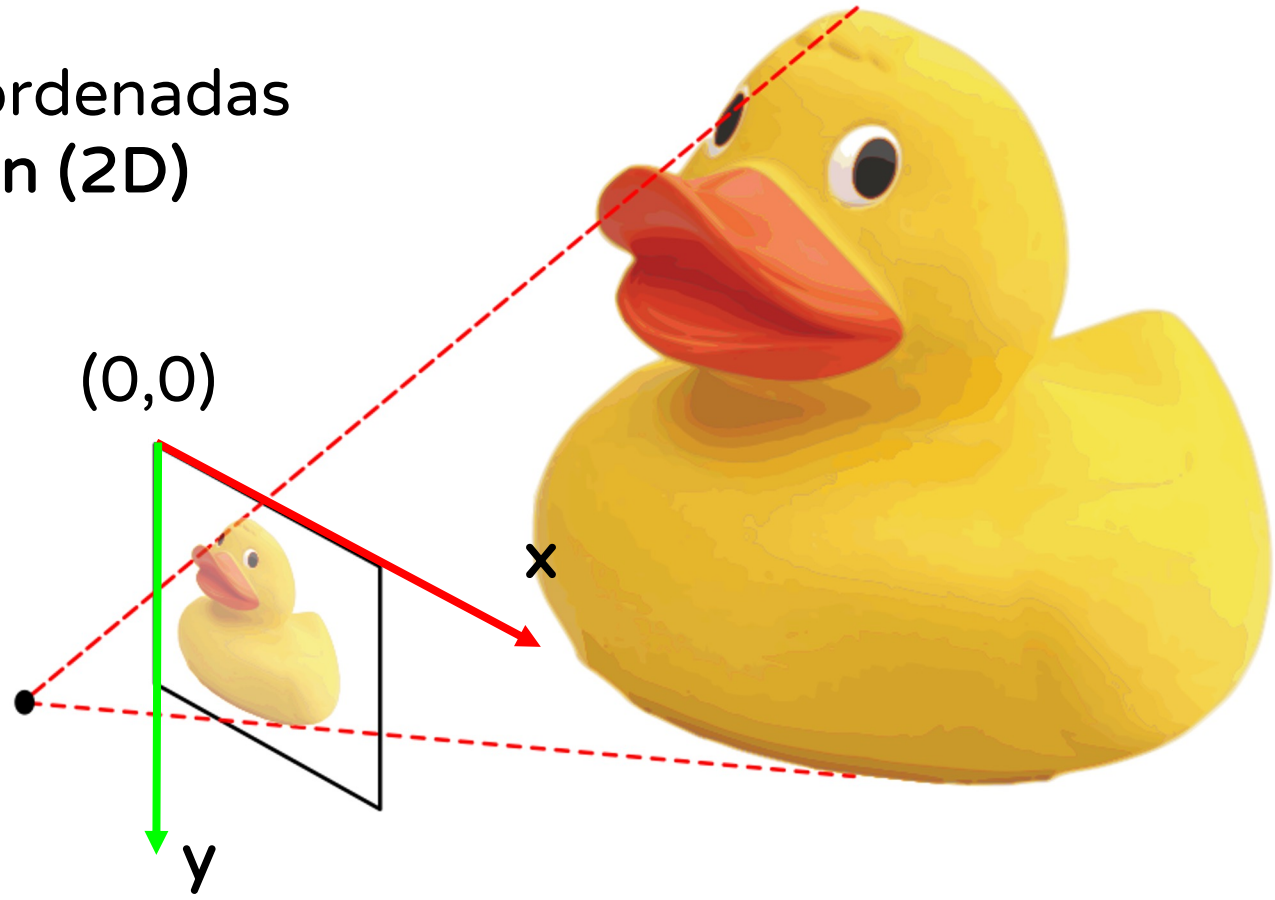


Cámara pinhole: cómo de verdad se usa



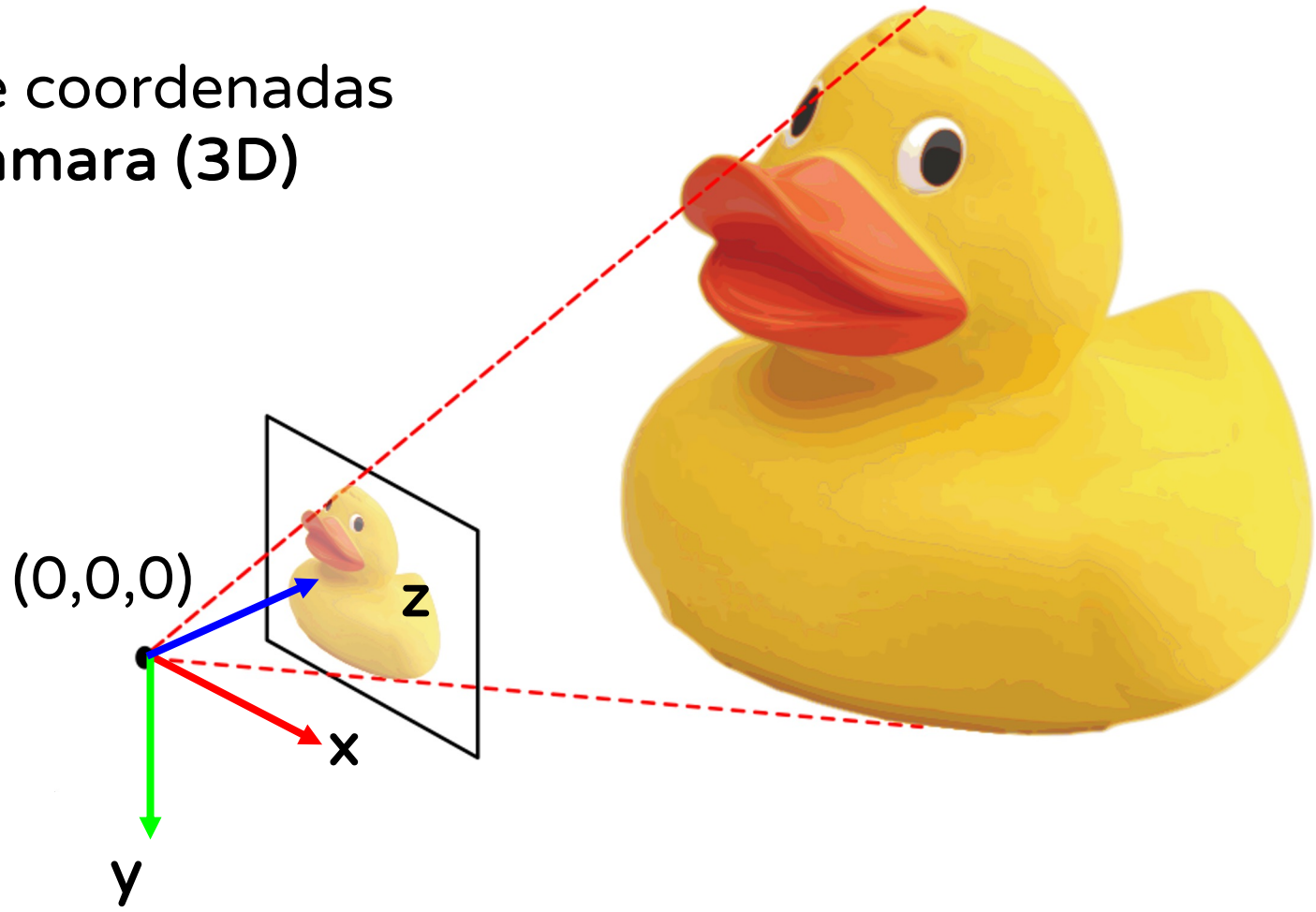
Cámara pinhole: ahora con matemáticas

Sistema de coordenadas
de la **imagen (2D)**



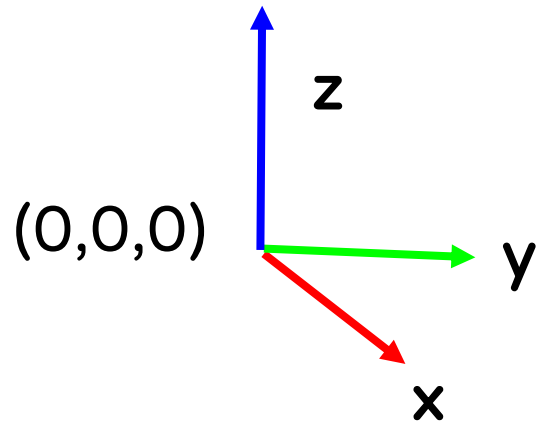
Cámara pinhole: ahora con matemáticas

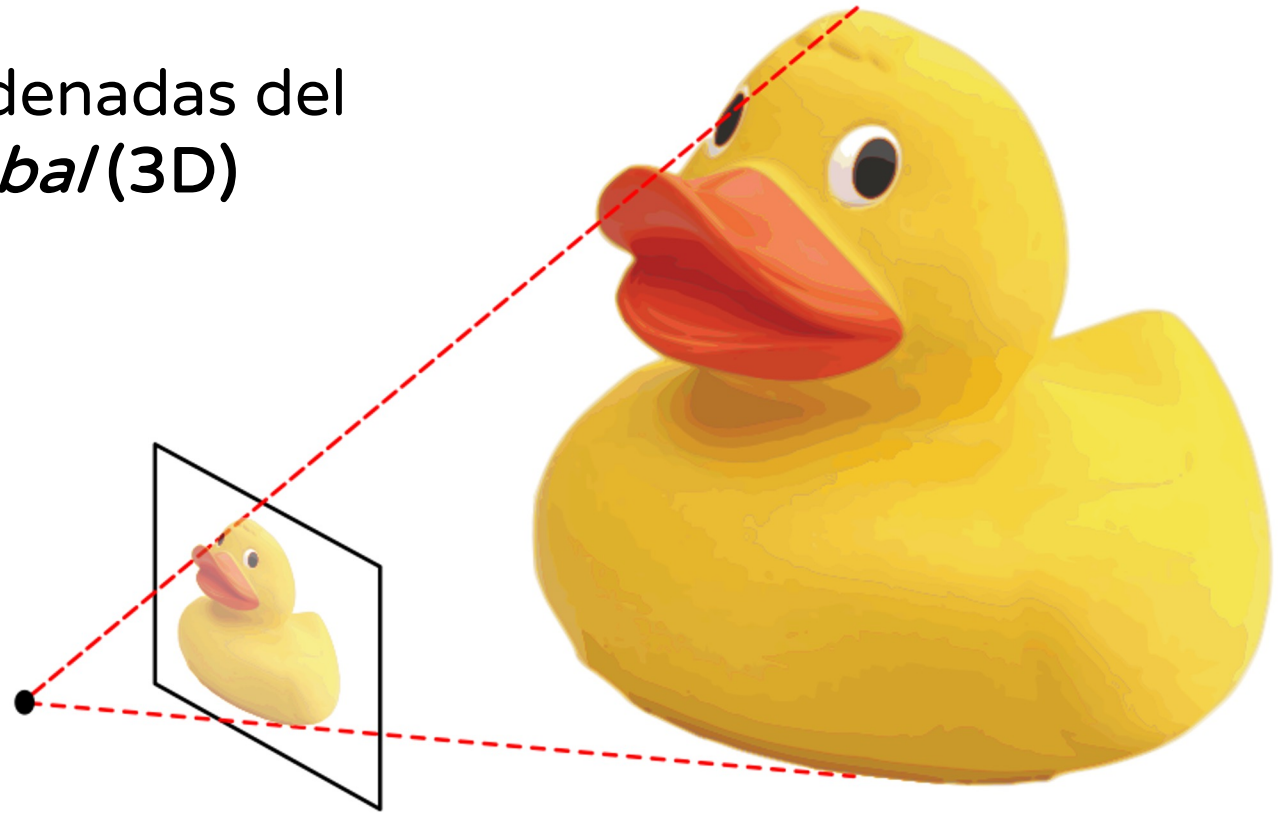
Sistema de coordenadas
de la **cámara (3D)**



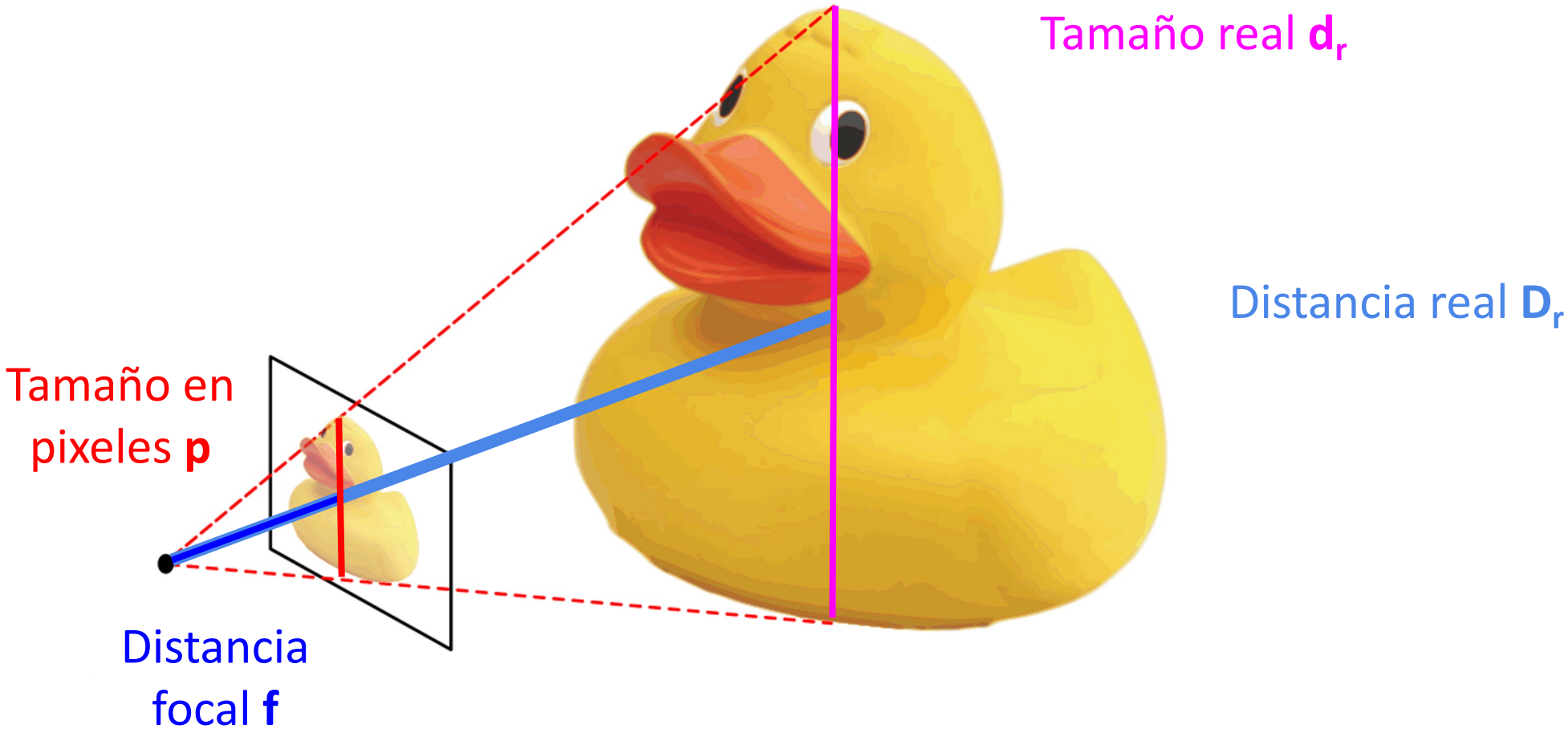
Cámara pinhole: ahora con matemáticas

Sistema de coordenadas del mundo o *global* (3D)



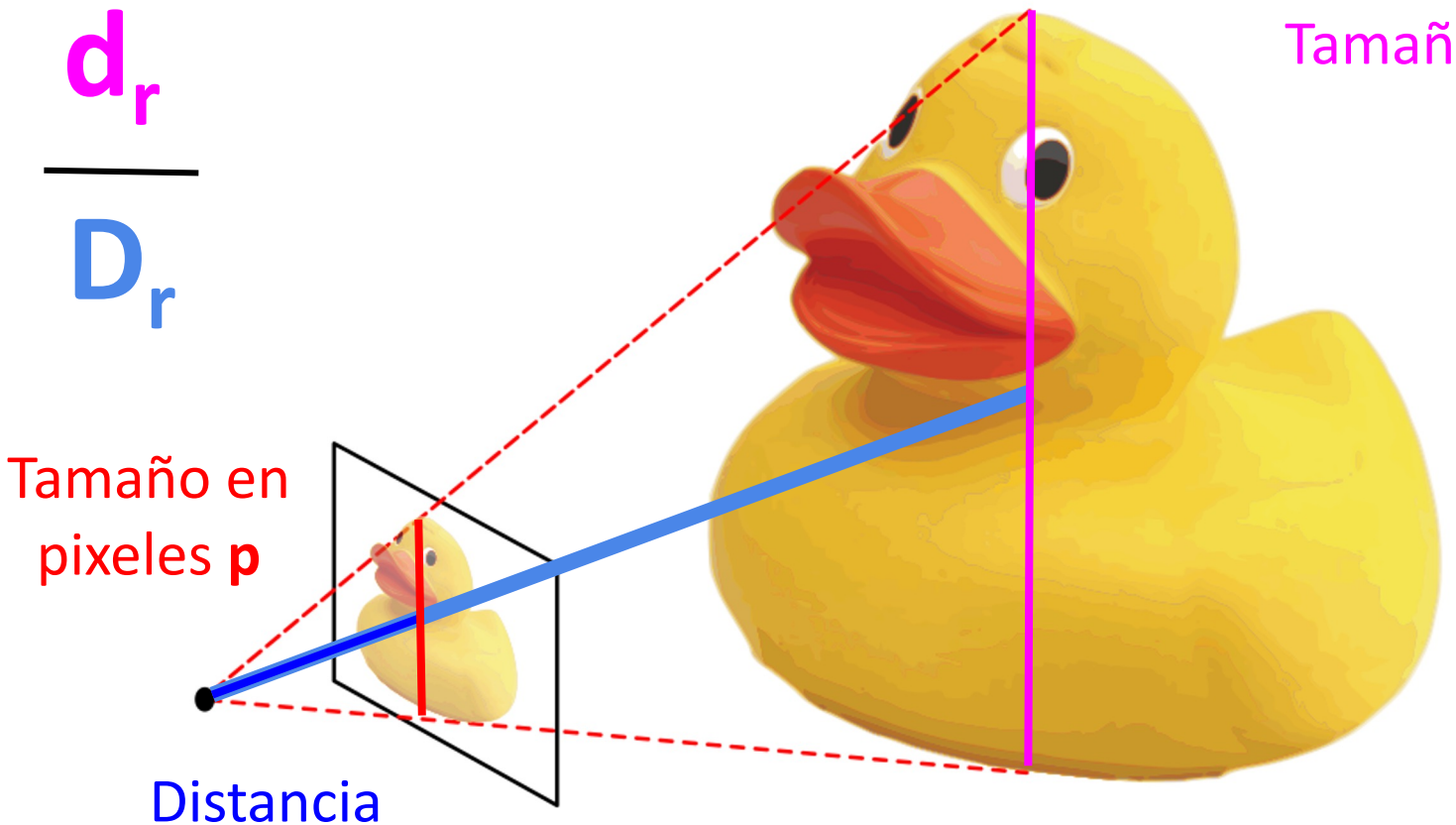


Cámara pinhole: ahora con matemáticas



Cámara pinhole: ahora con matemáticas

$$\frac{p}{f} = \frac{d_r}{D_r}$$



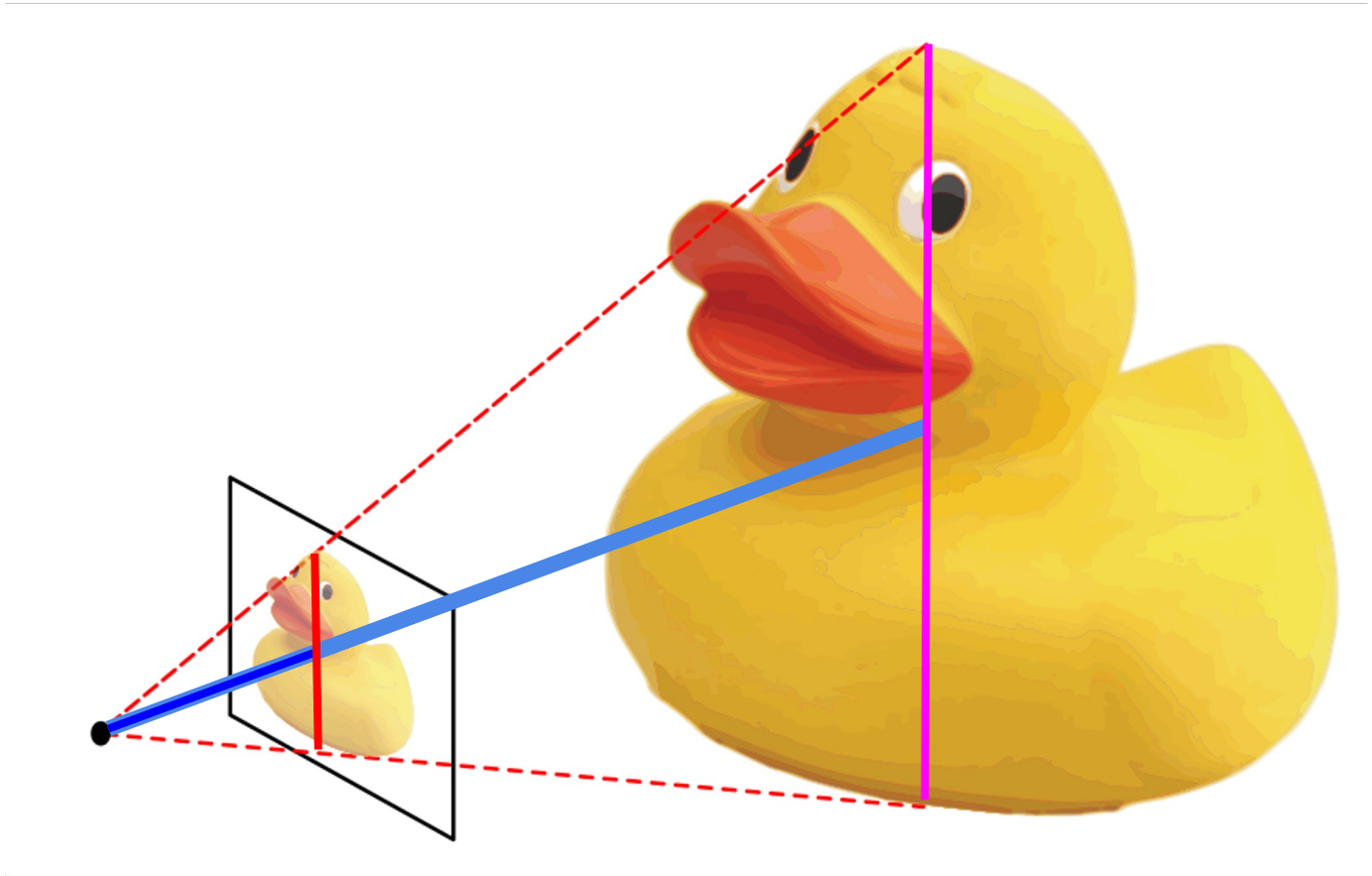
¿Qué problemas hay?

Hay incógnitas **f**, $\mathbf{d_r}$, $\mathbf{D_r}$

¿Qué problemas hay?

Hay incógnitas **f**, d_r, D_r

El objeto debe estar “centrado”



¿Qué problemas hay?

Hay incógnitas **f**, d_r, D_r

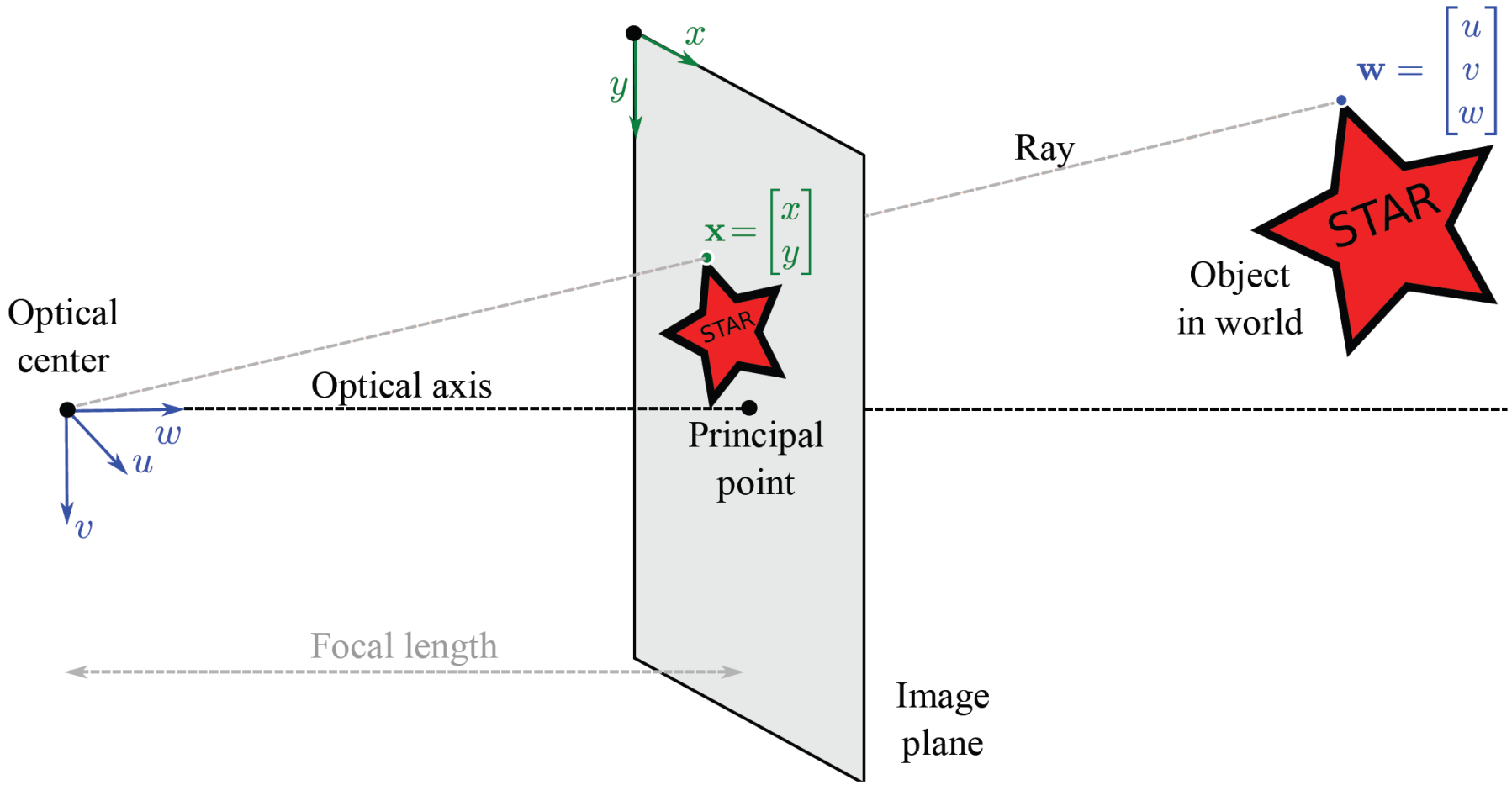
El objeto debe estar “centrado”

La cámara es ideal, no hay distorsión

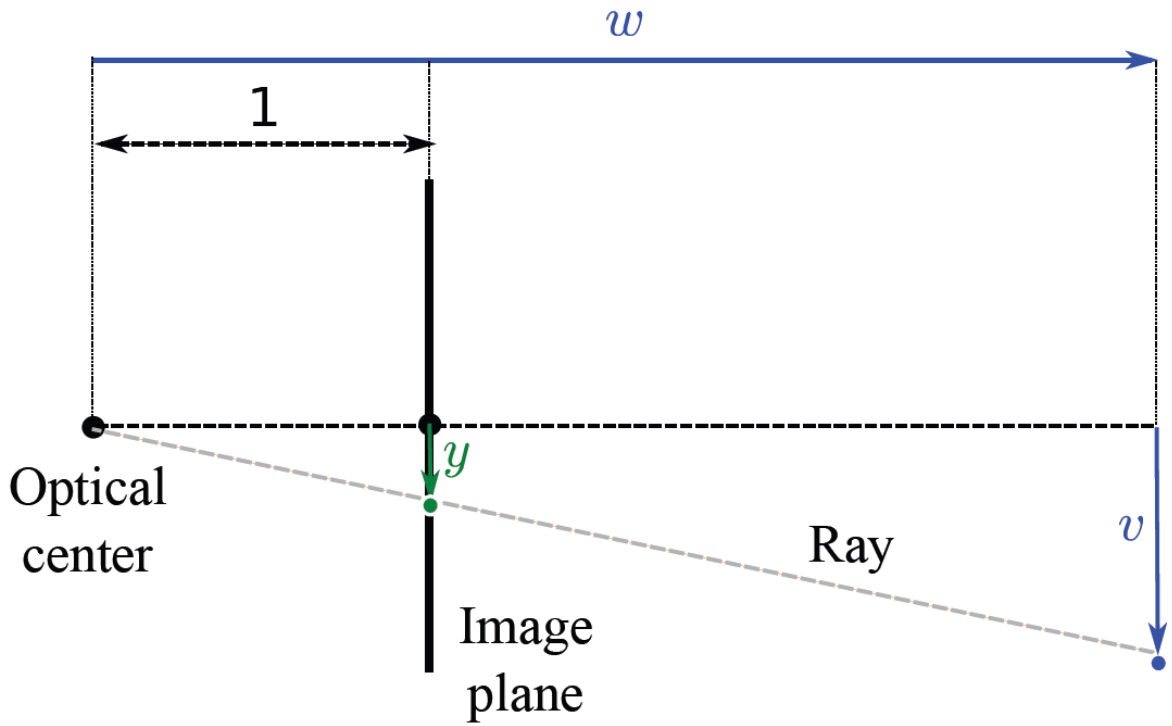
IRENEMIRANDAWORK.BLOGSPOT.COM

NYC

Pinhole camera: Terminología / Notación



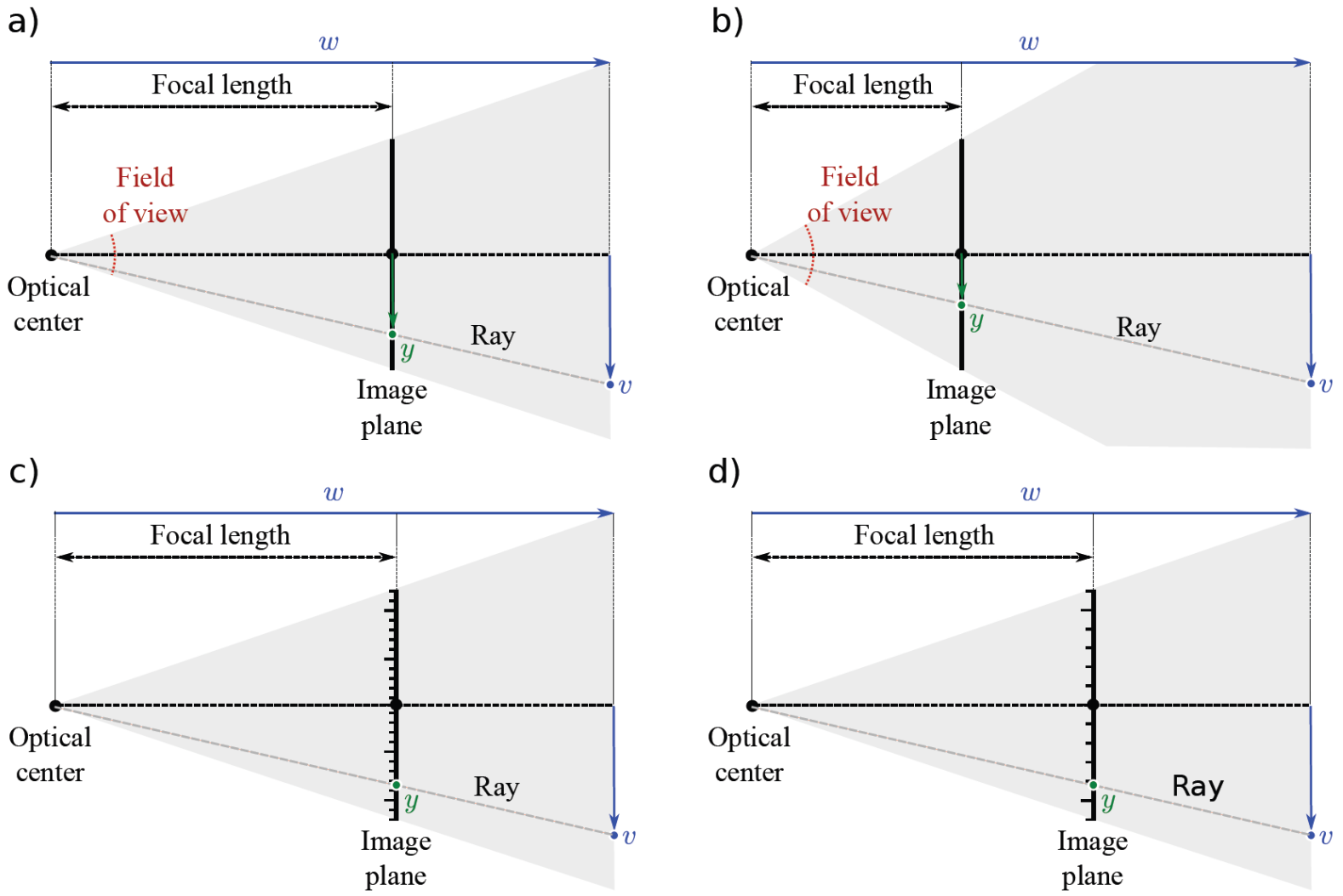
Camera Normalizada



By similar triangles:

$$x = \frac{u}{w} \qquad y = \frac{v}{w}$$

Parámetros & Distancia Focal



Parámetros & Distancia Focal

Puede modelar ambos

- El efecto de la distancia al plano focal
- la densidad de los receptores

con un solo parámetro de distancia focal f

$$x = \frac{\phi u}{w} \qquad y = \frac{\phi v}{w}$$

En la práctica, los receptores pueden no ser cuadrados:

$$x = \frac{\phi_x u}{w} \qquad y = \frac{\phi_y v}{w}$$

Por lo tanto, se usan diferentes parámetros de distancia focal para dims x e y

Parámetros de Offset

- El modelo previo supone que el píxel (0,0) es donde el rayo principal incide en el plano de la imagen (es decir, el centro)
- Offset al centro de la imagen:

$$x = \frac{\phi_x u}{w} + \delta_x$$

$$y = \frac{\phi_y v}{w} + \delta_y$$

Parametro de Skew

- Da cuenta de que el plano de la imagen no es exactamente perpendicular al rayo principal

$$
\begin{aligned}
x &= \frac{\phi_x u + \gamma v}{w} + \delta_x \\
y &= \frac{\phi_y v}{w} + \delta_y
\end{aligned}
$$

Posición y orientación de la cámara.

- La posición $\mathbf{w}=(u,v,w)^T$ del punto en el mundo generalmente no se expresa en el marco de referencia de la cámara.
- Transformación 3D

o

$$\begin{bmatrix} u' \\ v' \\ w' \end{bmatrix} = \begin{bmatrix} \omega_{11} & \omega_{12} & \omega_{13} \\ \omega_{21} & \omega_{22} & \omega_{23} \\ \omega_{31} & \omega_{32} & \omega_{33} \end{bmatrix} \begin{bmatrix} u \\ v \\ w \end{bmatrix} + \begin{bmatrix} \tau_x \\ \tau_y \\ \tau_z \end{bmatrix}$$

$$\mathbf{w}' = \mathbf{\Omega w} + \boldsymbol{\tau}$$

Punto en el marco de referencia de la cámara

Punto en el marco de referencia del mundo

Pinhole camera: Modelo completo

$$
\begin{aligned}
x &= \frac{\phi_x(\omega_{11}u + \omega_{12}v + \omega_{13}w + \tau_x) + \gamma(\omega_{21}u + \omega_{22}v + \omega_{23}w + \tau_y)}{\omega_{31}u + \omega_{32}v + \omega_{33}w + \tau_z} + \delta_x \\
y &= \frac{\phi_y(\omega_{21}u + \omega_{22}v + \omega_{23}w + \tau_y)}{\omega_{31}u + \omega_{32}v + \omega_{33}w + \tau_z} + \delta_y.
\end{aligned}
$$

- Parámetros Intrínsecos

$$\{\phi_x, \phi_y, \gamma, \delta_x, \delta_y\}$$

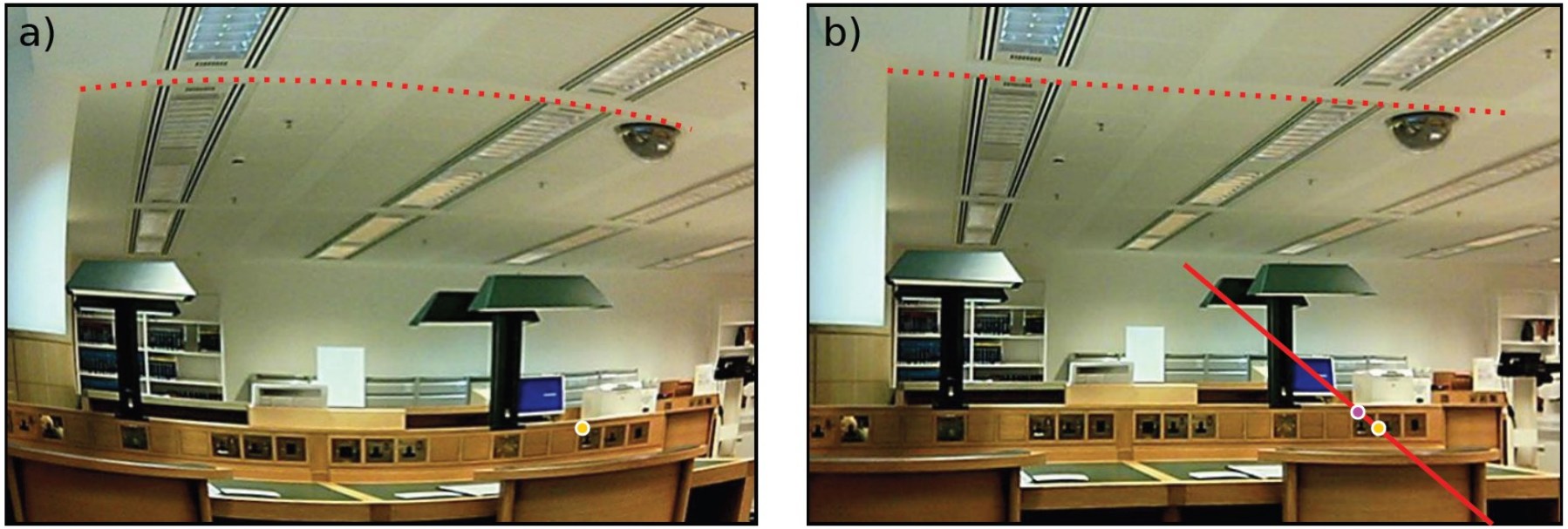
- Parámetros Extrínsecos

$$\{\boldsymbol{\Omega}, \boldsymbol{\tau}\}$$

(como matriz intrínseca)

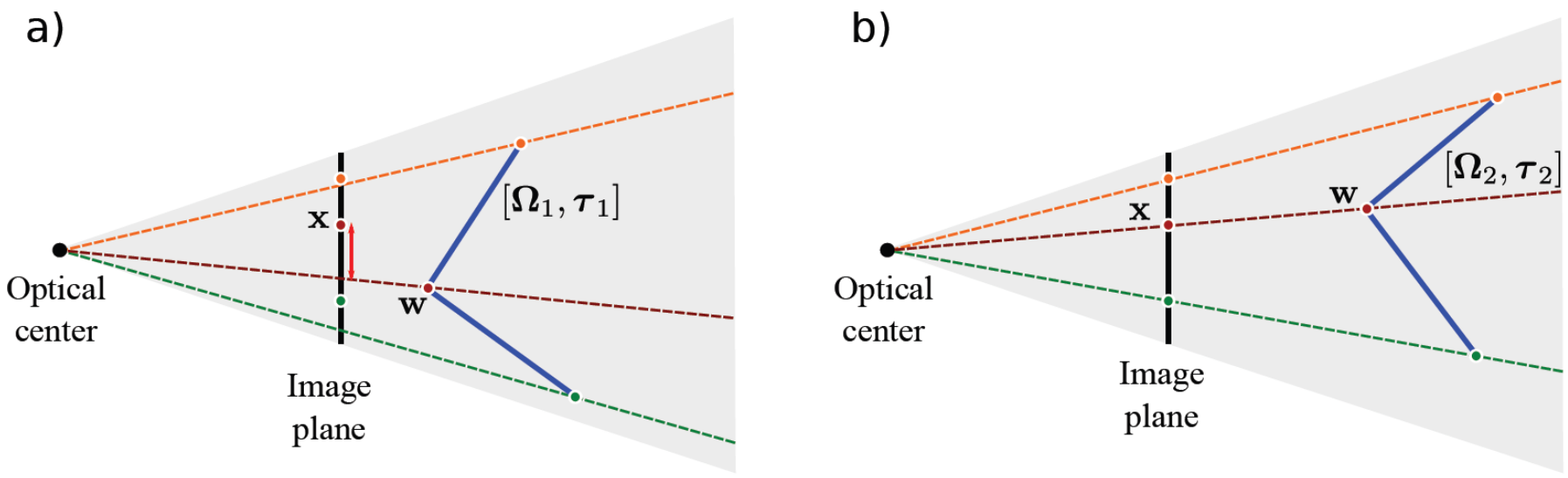
$$\boldsymbol{\Lambda} = \begin{bmatrix} \phi_x & \gamma & \delta_x \\ 0 & \phi_y & \delta_y \\ 0 & 0 & 1 \end{bmatrix}$$

Distorcion Radial



$$
\begin{aligned}
x' &= x(1 + \beta_1 r^2 + \beta_2 r^4) \\
y' &= y(1 + \beta_1 r^2 + \beta_2 r^4)
\end{aligned}
$$

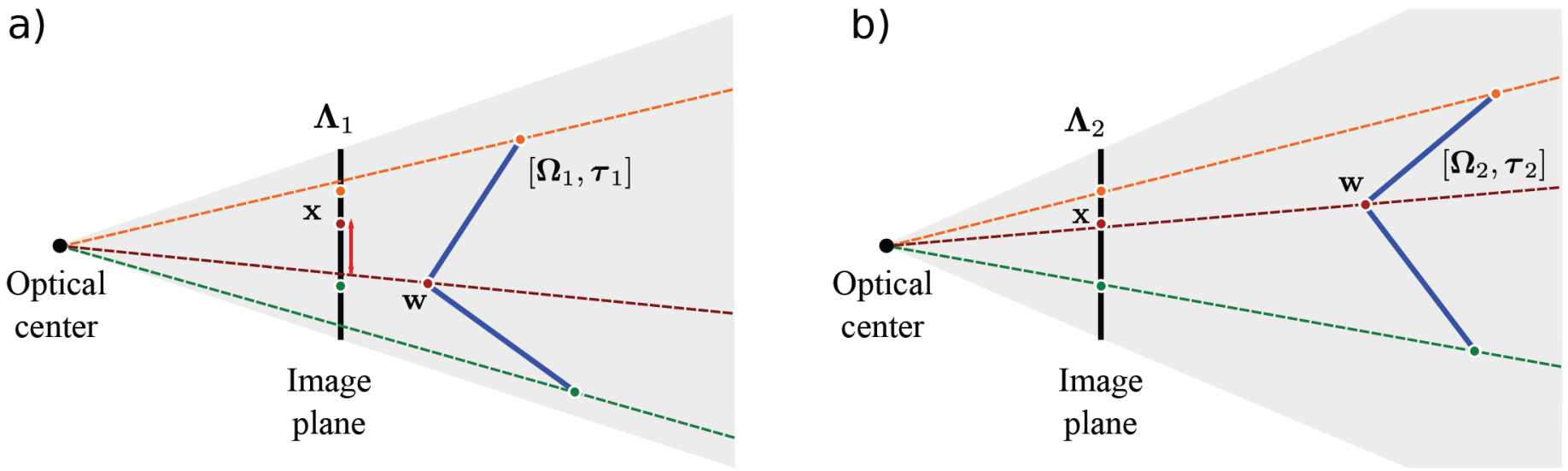
Problema 1: Aprendizaje de parámetros extrínsecos (orientación exterior)



Usar máxima verosimilitud (maximum likelihood):

$$\hat{\Omega}, \hat{\tau} = \underset{\Omega, \tau}{\operatorname{argmax}} \sum_{i=1}^{I} \log \left[Pr(\mathbf{x}_i | \mathbf{w}_i, \Lambda, \Omega, \tau) \right]$$

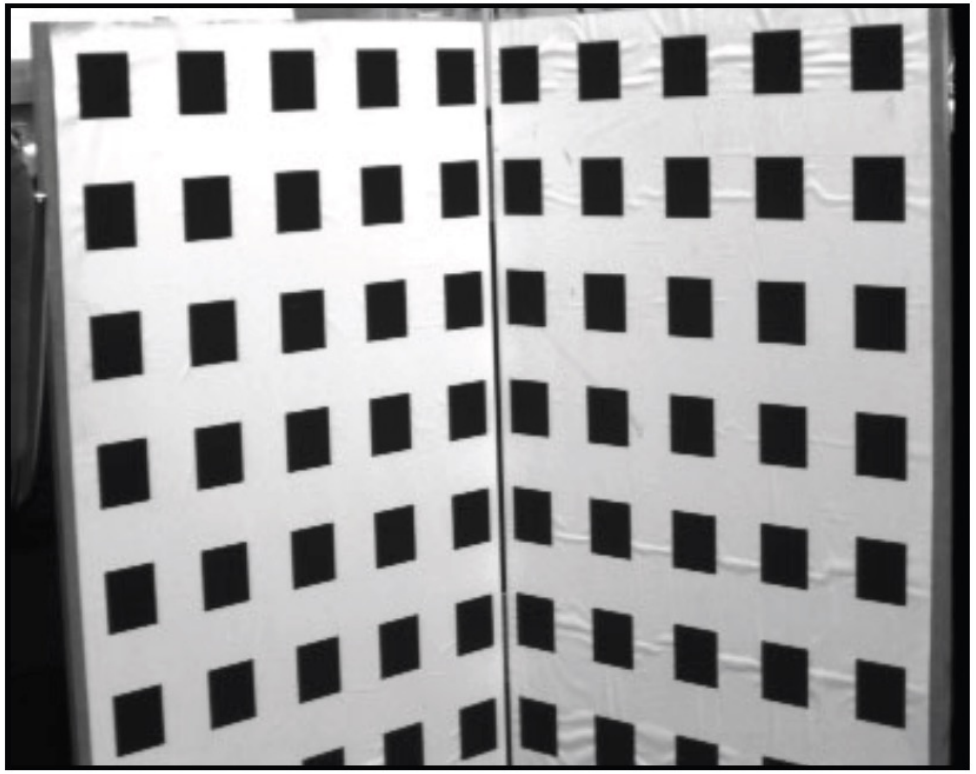
Problema 2: Aprendizaje de parámetros intrínsecos (calibración)



Usar máxima verosimilitud (maximum likelihood):

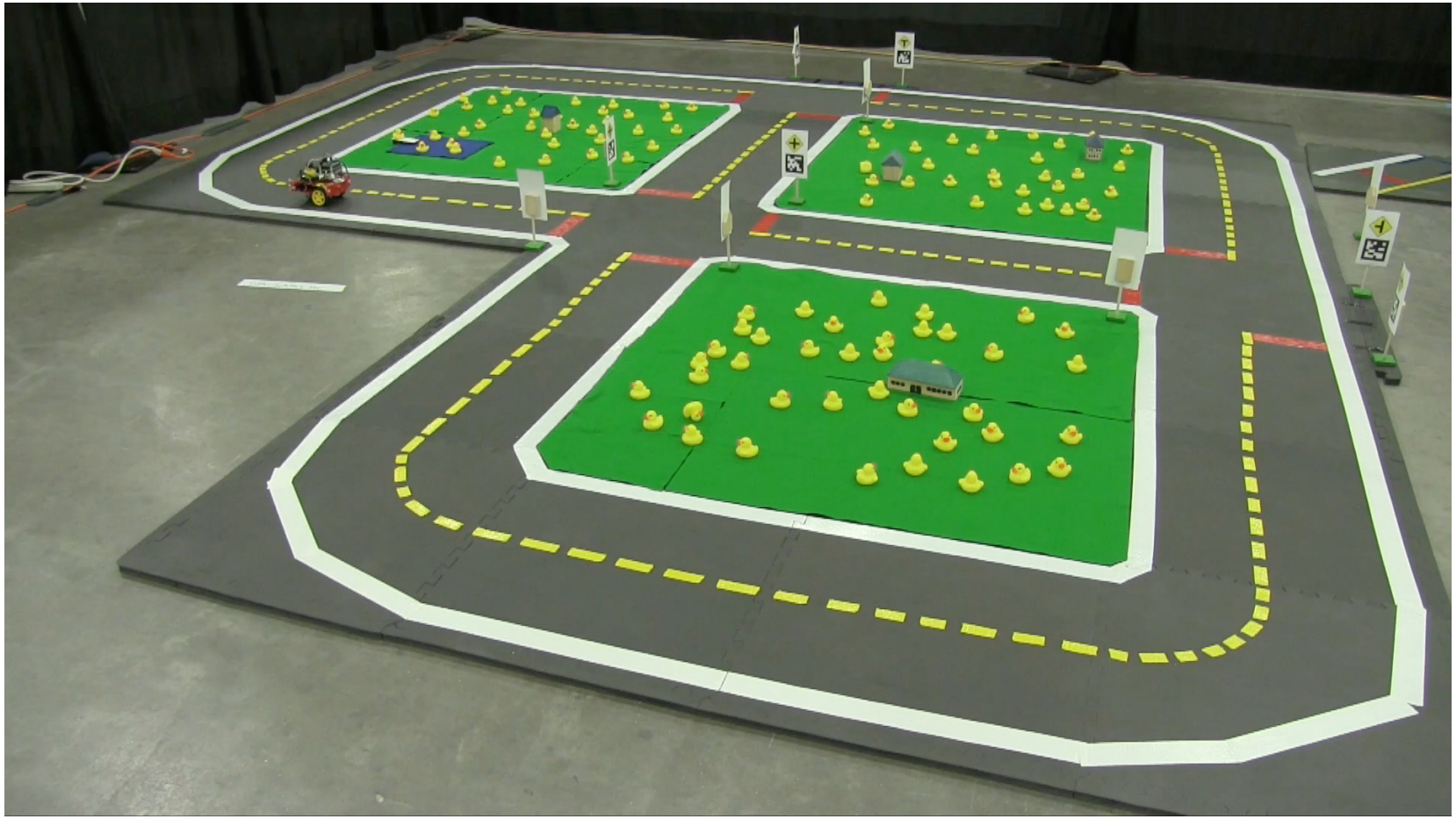
$$\hat{\mathbf{\Lambda}} = \underset{\mathbf{\Lambda}}{\operatorname{argmax}} \left[\max_{\boldsymbol{\Omega}, \boldsymbol{\tau}} \sum_{i=1}^{I} \log \left[Pr(\mathbf{x}_i | \mathbf{w}_i, \mathbf{\Lambda}, \boldsymbol{\Omega}, \boldsymbol{\tau}) \right] \right]$$

Calibracion



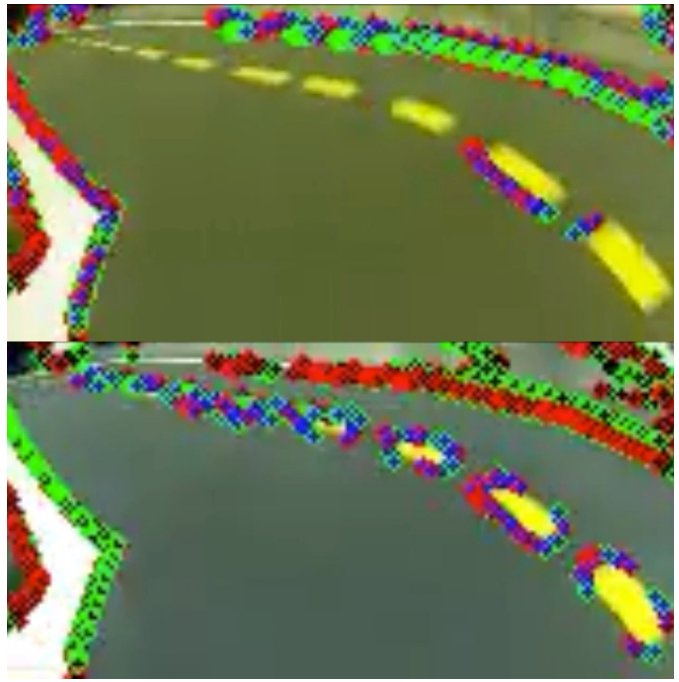
- Usar objetivo 3D con puntos 3D conocidos

Objetivo:



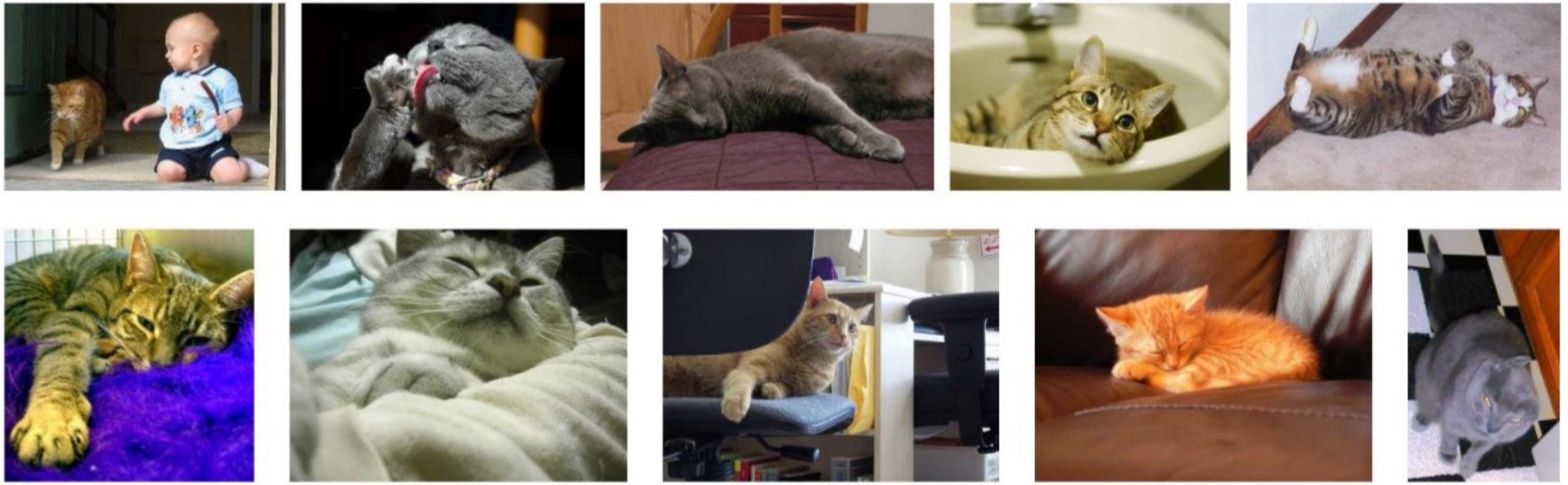
Computer Vision Basics

- **Objectives:**
  - Illumination invariance
  - Line detection
  - Feature representations
  - Place recognition



Computer Vision
~~Deep Learning~~ is Hard

Pose Variability



Computer Vision
~~Deep Learning~~ is Hard

Intra-Class Variability


Abyssinian


Bengal


Bombay


Persian


Egyptian


Ragdoll


Eng. Setter


Boxer


Keeshond


Chihuahua


Great Pyrenees


German Shorthaired

Parkhi et al. "Cats and dogs." 2012.

Computer Vision
~~Deep Learning~~ is Hard

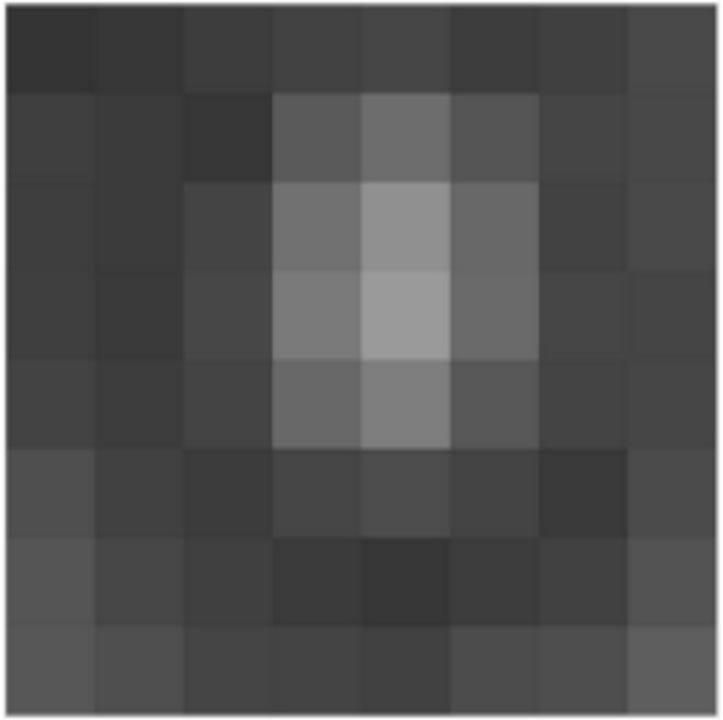
Illumination Variability

Computer Vision
~~Deep Learning~~ is Hard

Occlusion

Cuantas imágenes hay?

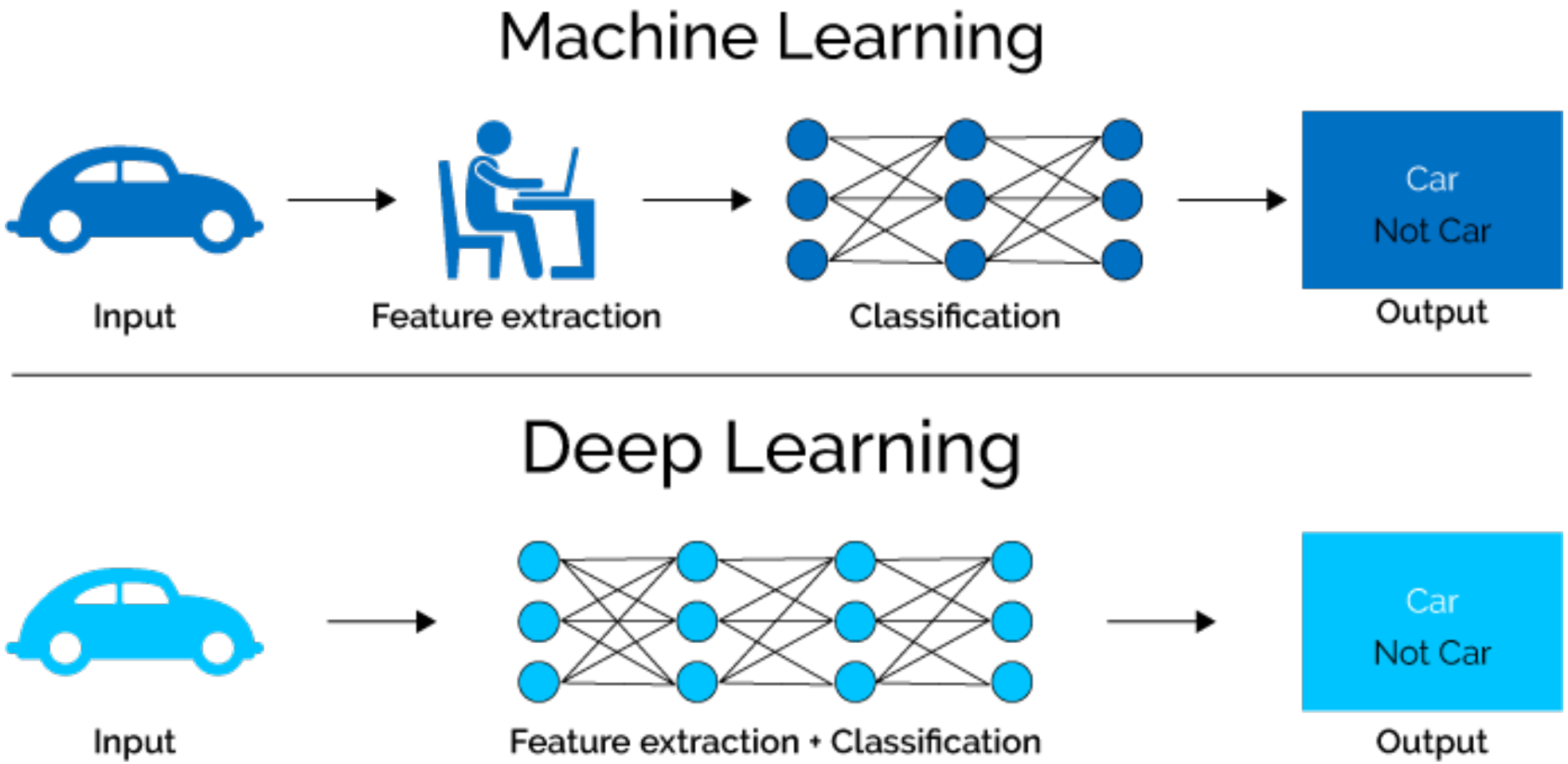
Imagen de 8x8 pixeles, escala de grices (256 valores):



$$256*256*\ldots*256 = 256^{8*8} \sim 10^{150}$$

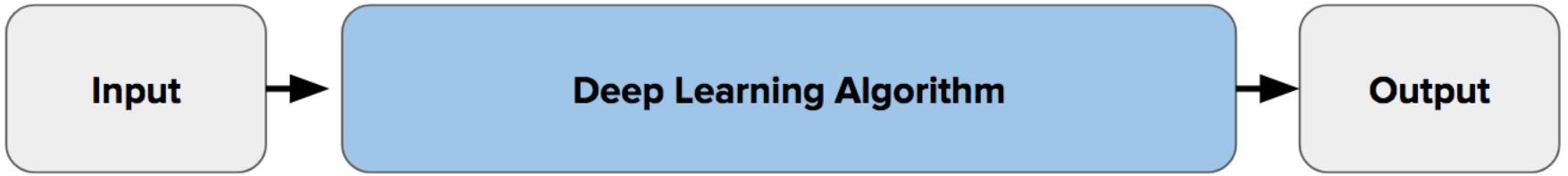
$$\begin{bmatrix} 52 & 55 & 61 & 66 & 70 & 61 & 64 & 73 \\ 63 & 59 & 55 & 90 & 109 & 85 & 69 & 72 \\ 62 & 59 & 68 & 113 & 144 & 104 & 66 & 73 \\ 63 & 58 & 71 & 122 & 154 & 106 & 70 & 69 \\ 67 & 61 & 68 & 104 & 126 & 88 & 68 & 70 \\ 79 & 65 & 60 & 70 & 77 & 68 & 58 & 75 \\ 85 & 71 & 64 & 59 & 55 & 61 & 65 & 83 \\ 87 & 79 & 69 & 68 & 65 & 76 & 78 & 94 \end{bmatrix}$$

Approaches to Computer Vision & Machine Learning

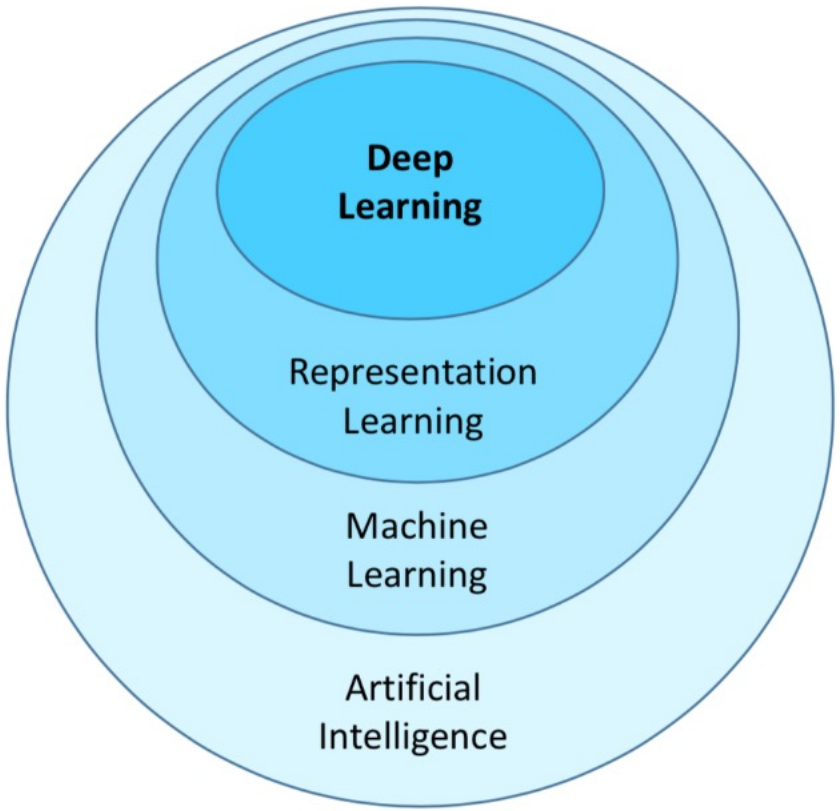


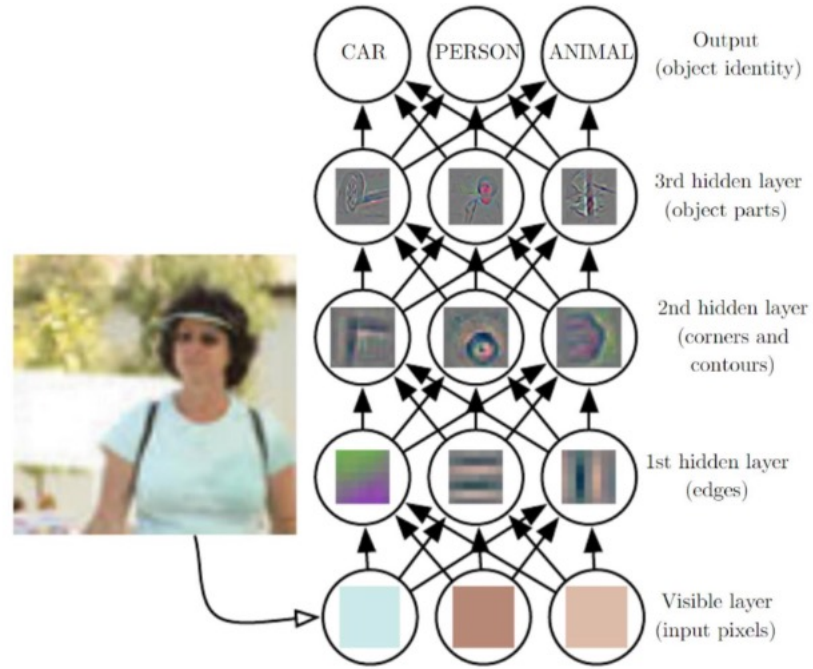
Approaches to Computer Vision & Machine Learning

Traditional Machine Learning Flow



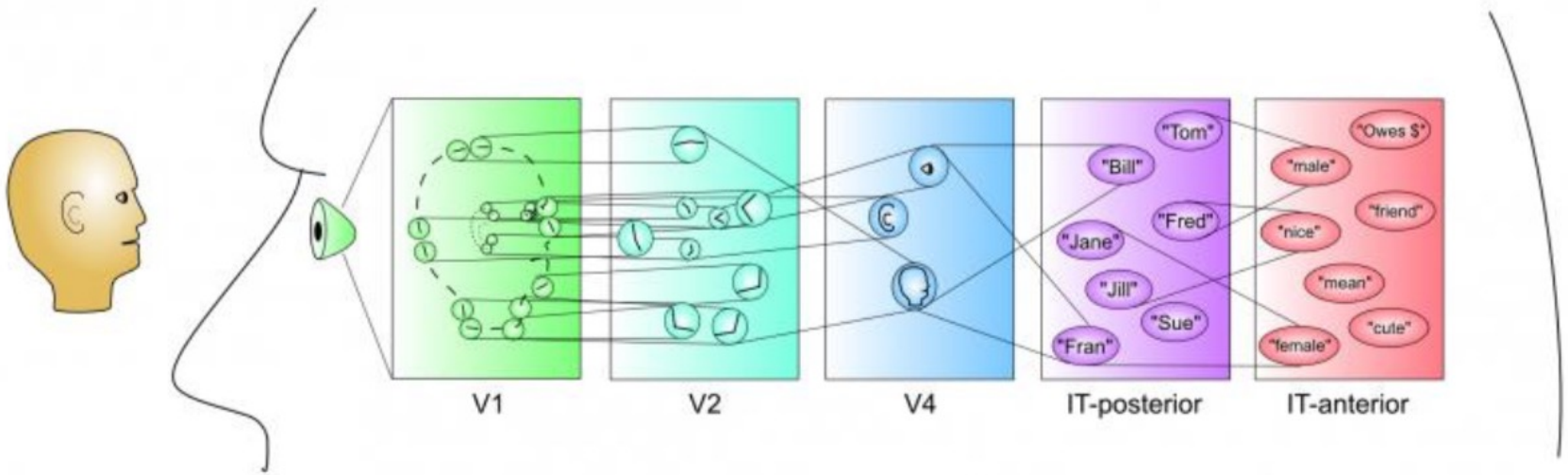
Deep Learning Flow

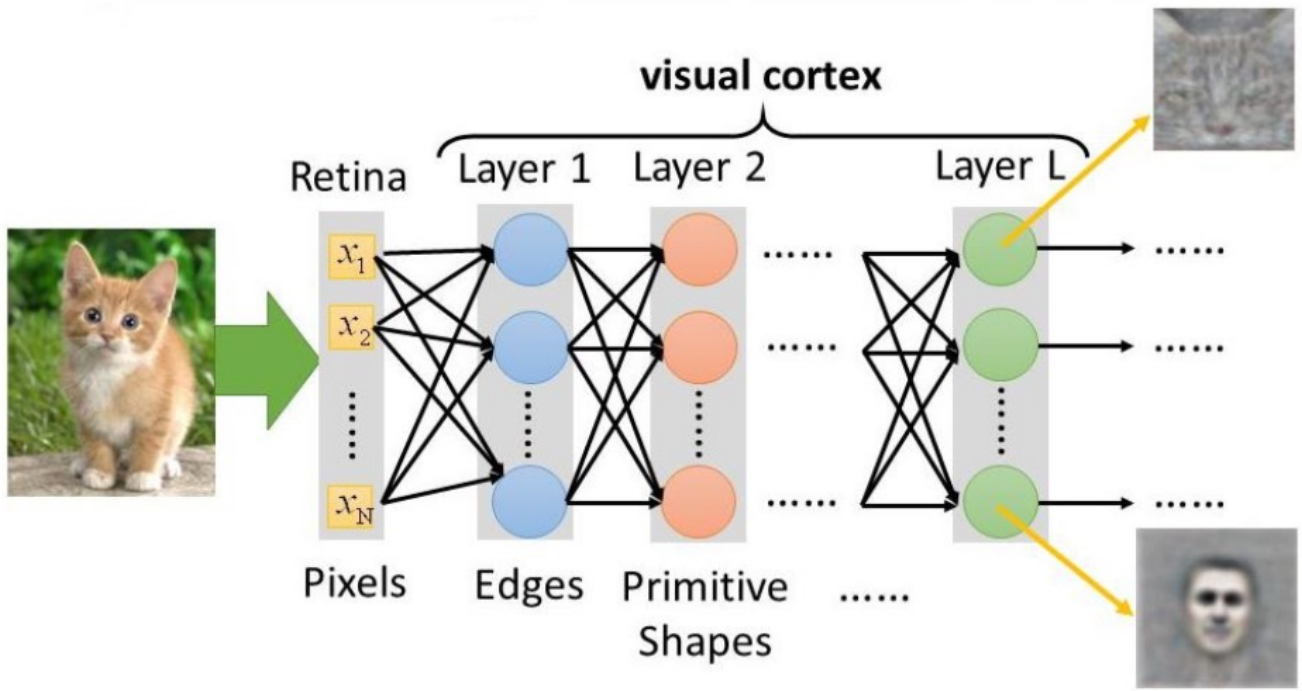

Deep
Learning
Representation
Learning
Machine
Learning
Artificial
Intelligence


CAR
PERSON
ANIMAL
Output
(object identity)
3rd hidden layer
(object parts)
2nd hidden layer
(corners and
contours)
1st hidden layer
(edges)
Visible layer
(input pixels)

Visual Cortex

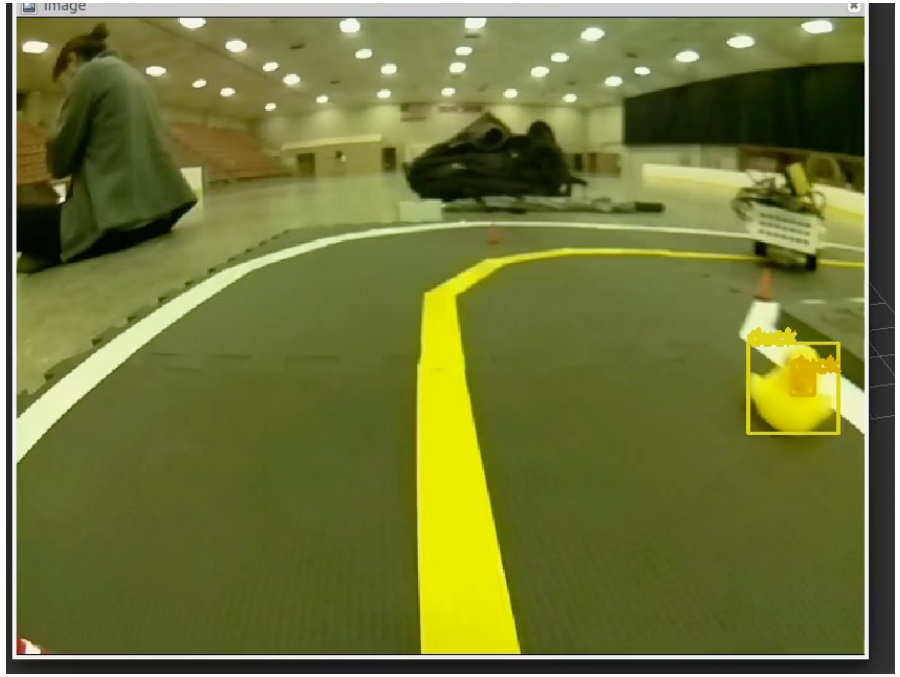
(Its Structure is Instructive and Inspiring)





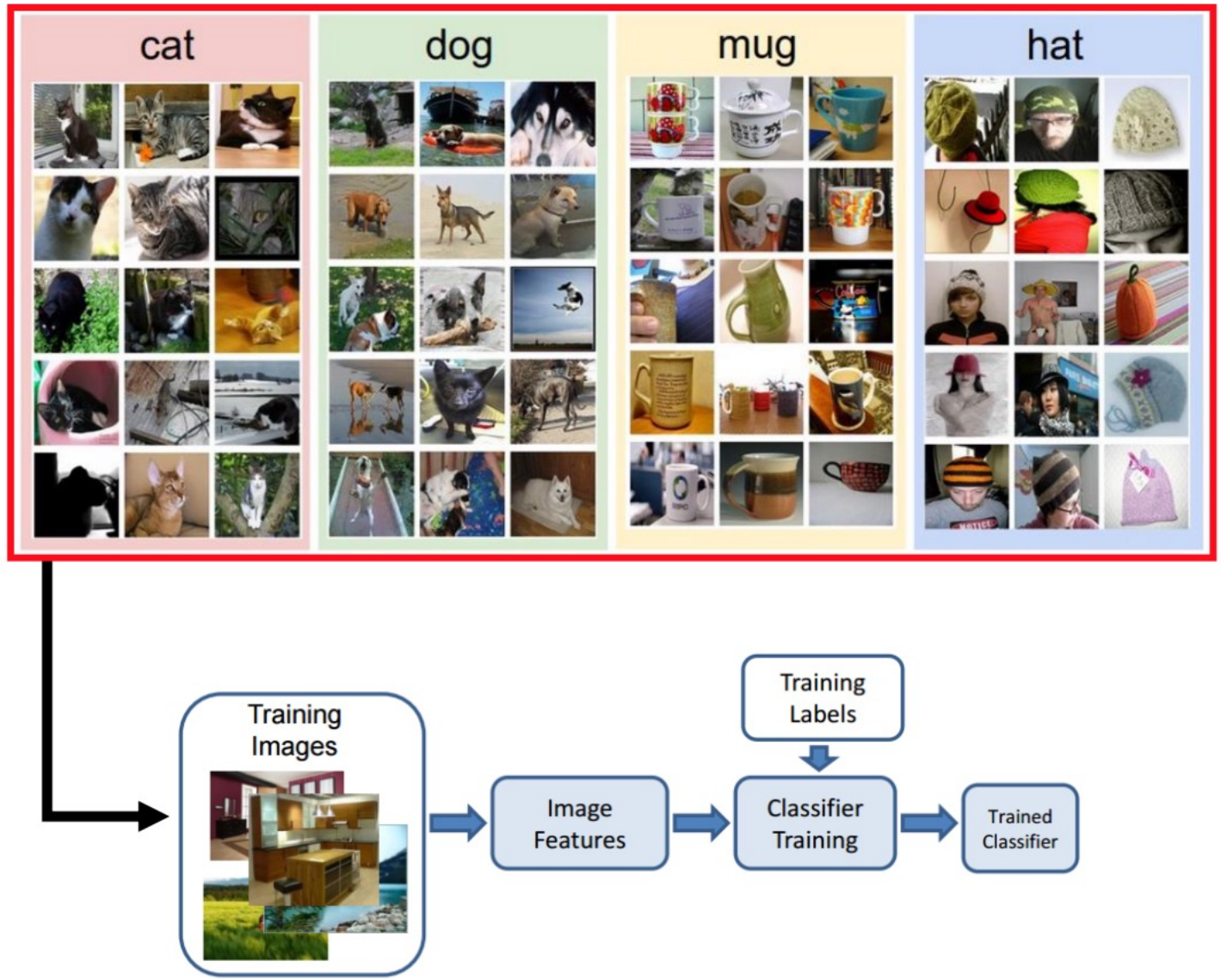
Deep Learning for Computer Vision

- **Task:**
  - Object detection
  - Object tracking
  - Semantic segmentation
  - Recognition
  - …



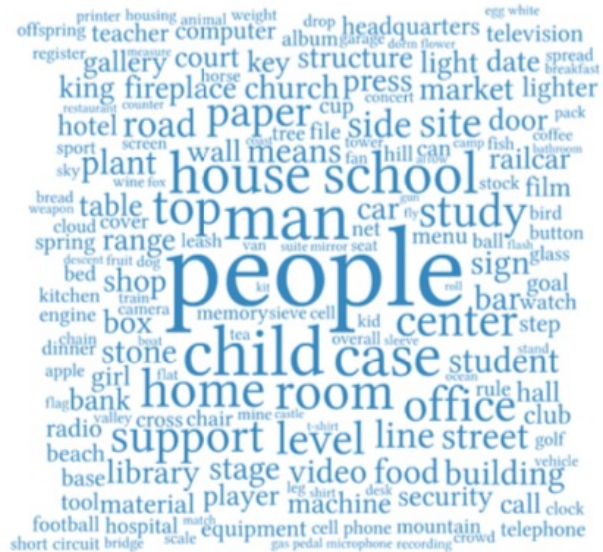
Scene Understating

Image Classification Pipeline

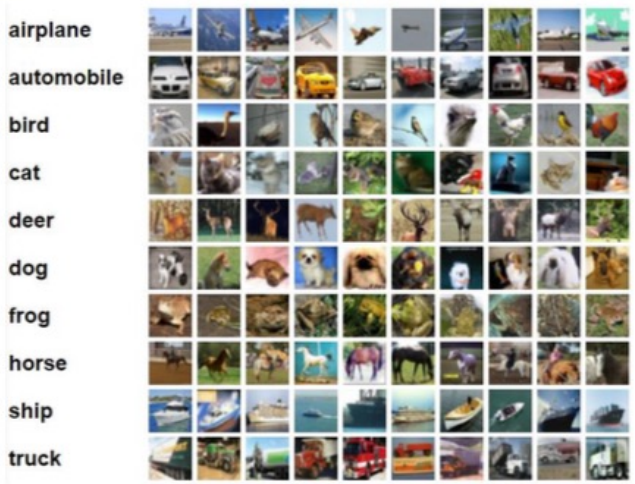


Famous Computer Vision Datasets

MNIST: handwritten digits

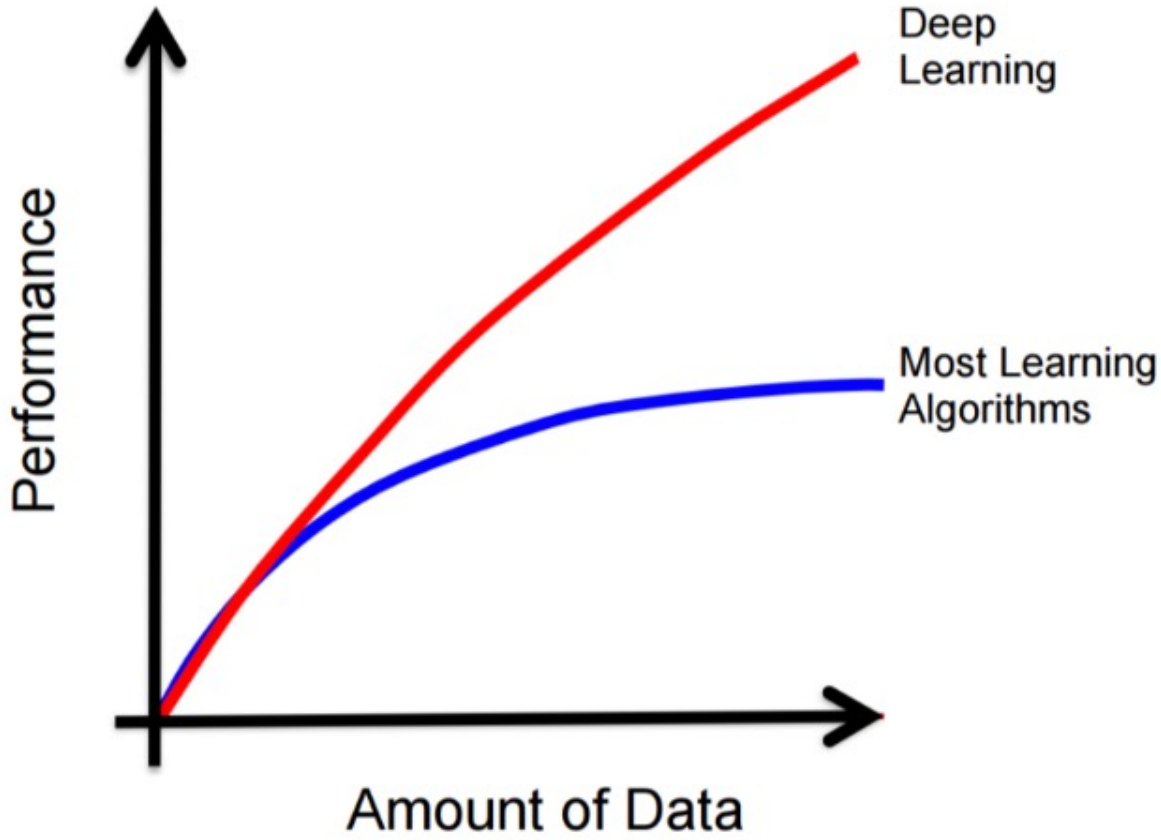


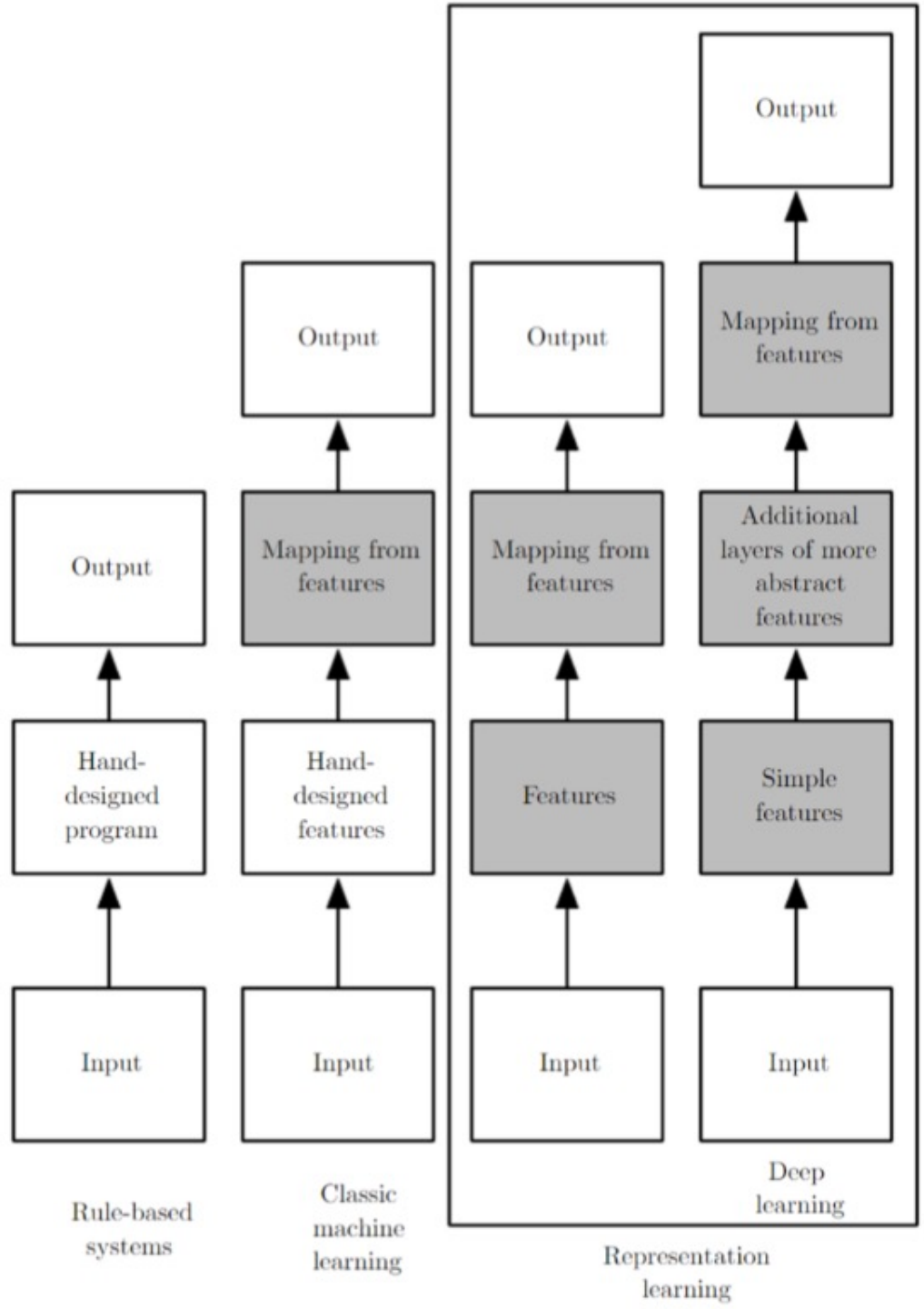
ImageNet: WordNet hierarchy



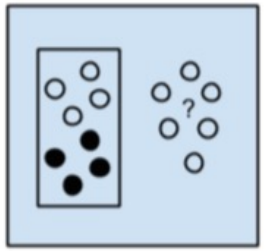
CIFAR-10(0): tiny images

Places: natural scenes

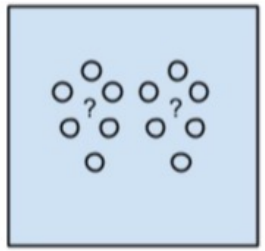

Deep
Learning
Most Learning
Algorithms
Performance
Amount of Data


Output
Output
Output
Output
Mapping from features
Mapping from features
Mapping from features
Additional layers of more abstract features
Hand-designed program
Hand-designed features
Features
Simple features
Input
Input
Input
Input
Rule-based systems
Classic machine learning
Representation learning
Deep learning

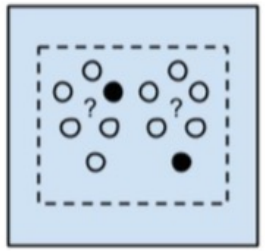
Computer Vision & Machine Learning



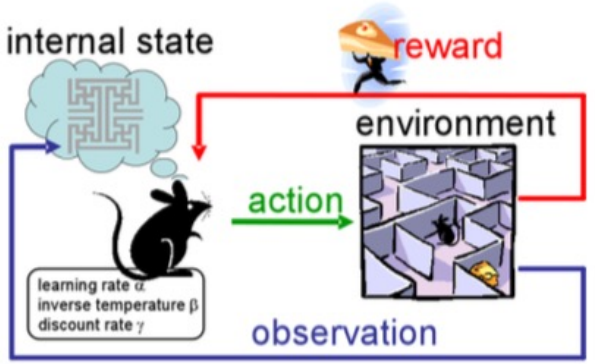
Supervised Learning



Unsupervised Learning

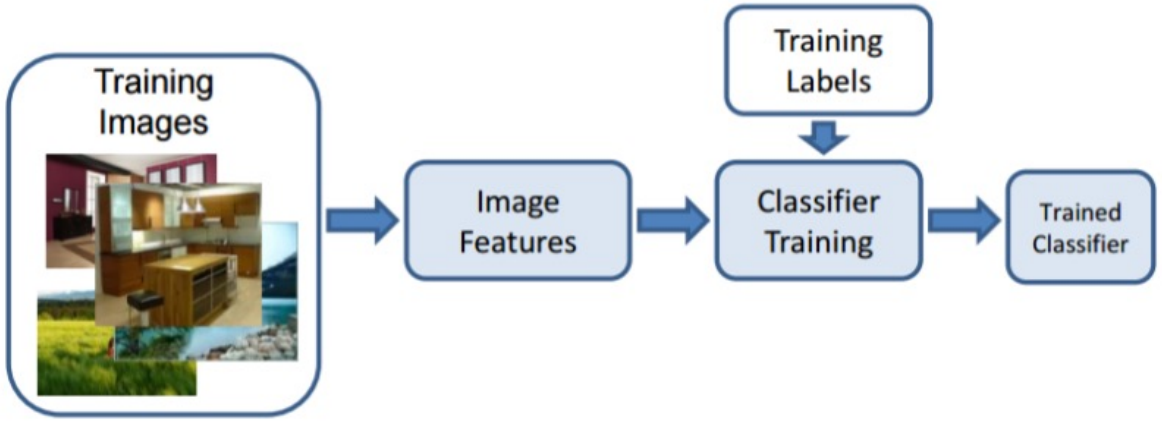


Semi-Supervised Learning



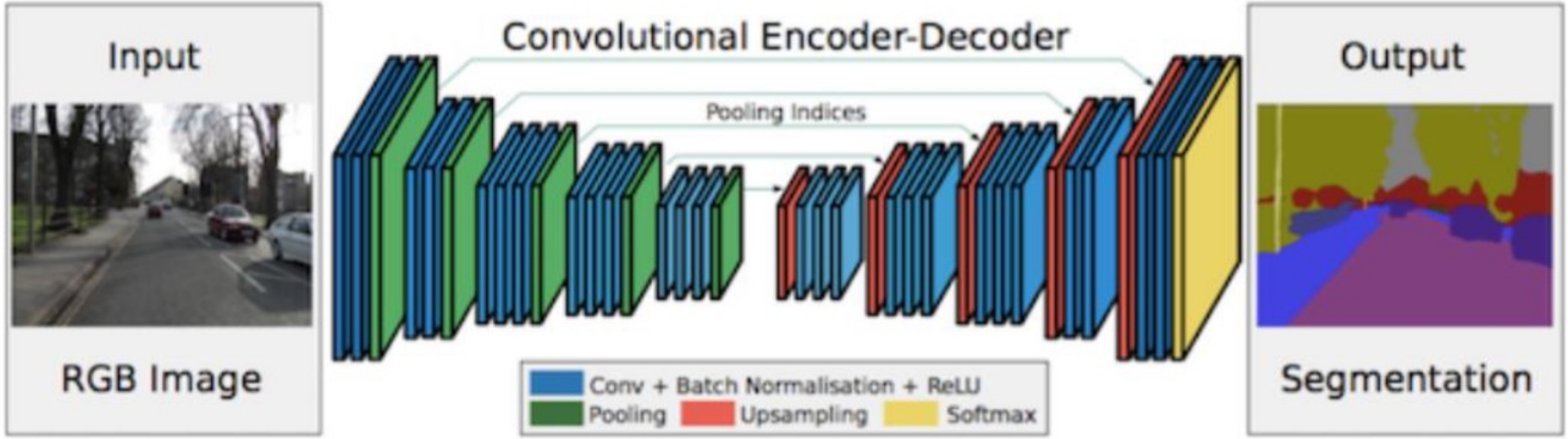
Reinforcement Learning

Computer Vision



Deep learning for Segmentation

Deep learning for Segmentation

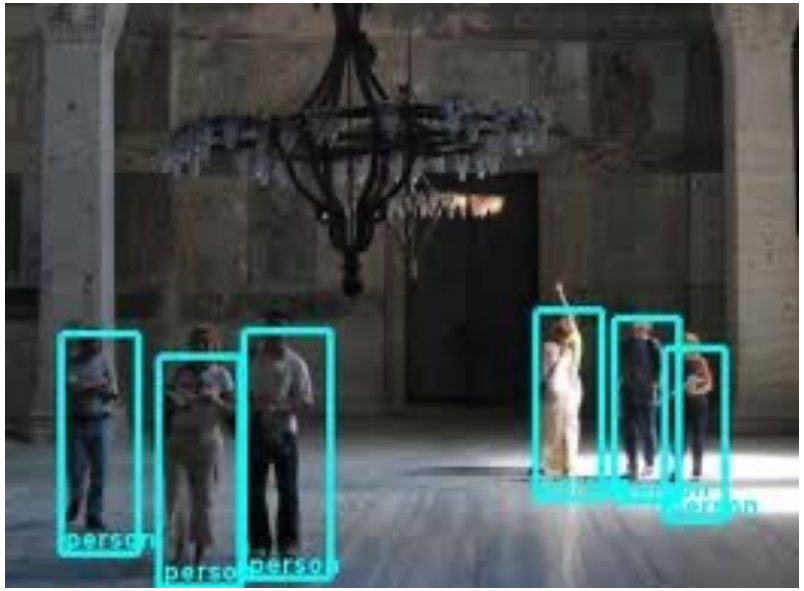


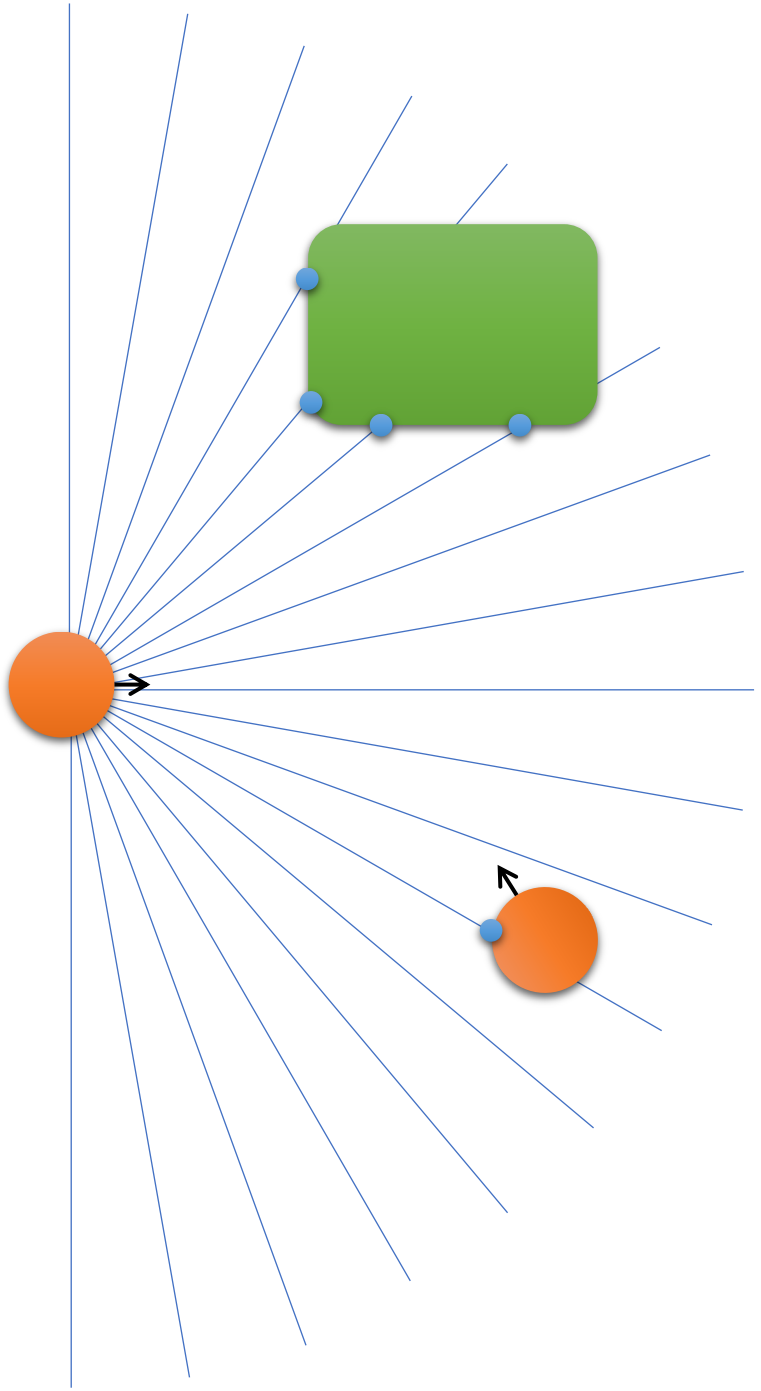
Example: Object Detection

How can we detect a given object (Yellow Duck)?

Detección

- Encontrar los objetos presentes en una medición
- Los objetos aparecen como una región de la medición

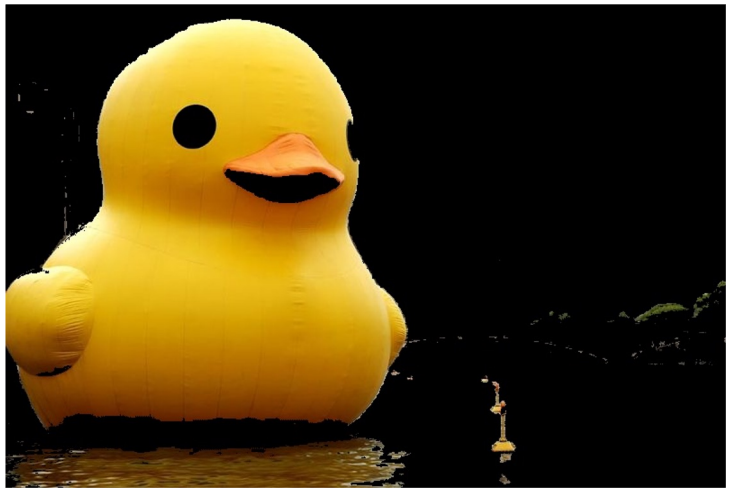




Approach 1: (Classic)
Extraer Features -> Luego Clasificar

- Feature: Vector que representa a un punto o una región de la medición
- Normalmente contiene cierta información de vecindad

Feature: yellow?
Every non yellow pixel is not a duck

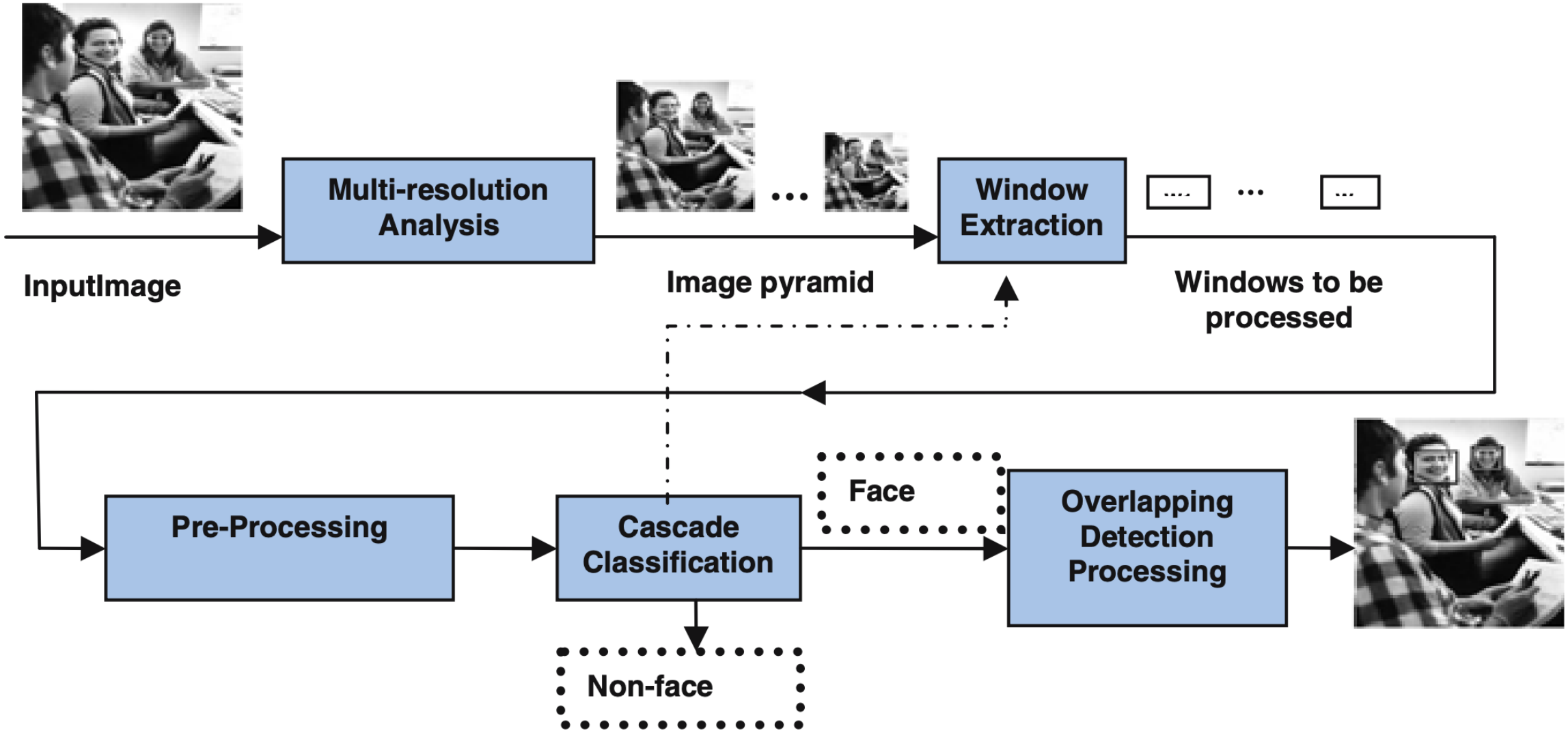


Where are the ducks?

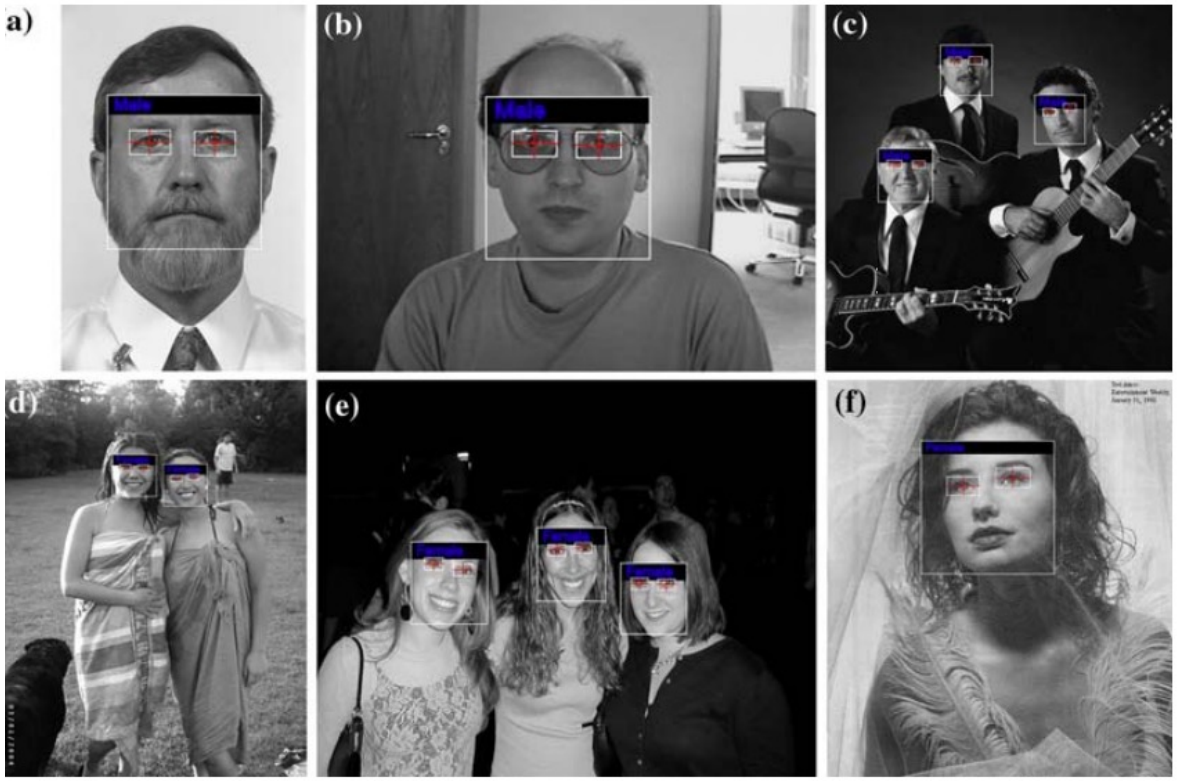
Approach 1: (Classic)
Extraer Features -> Luego Clasificar

- Feature: Vector que representa a un punto o una región de la medición
- Normalmente contiene cierta información de vecindad
- Ej:
  - Blobs (e.g. connected yellow pixels)
  - Descriptores (SIFT, SURF, DAISY, etc.)
  - Clases de color, Textura
  - Descriptores de ventanas
  - Geométricos

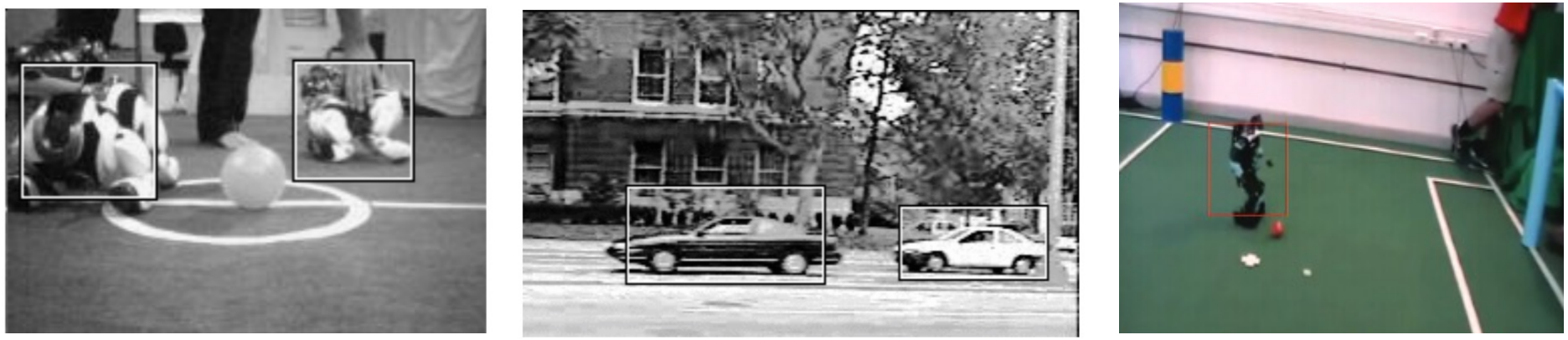
Approach 2:
Sliding Windows + Classifier



Frontal Faces & Eye Detection, Gender Recognition

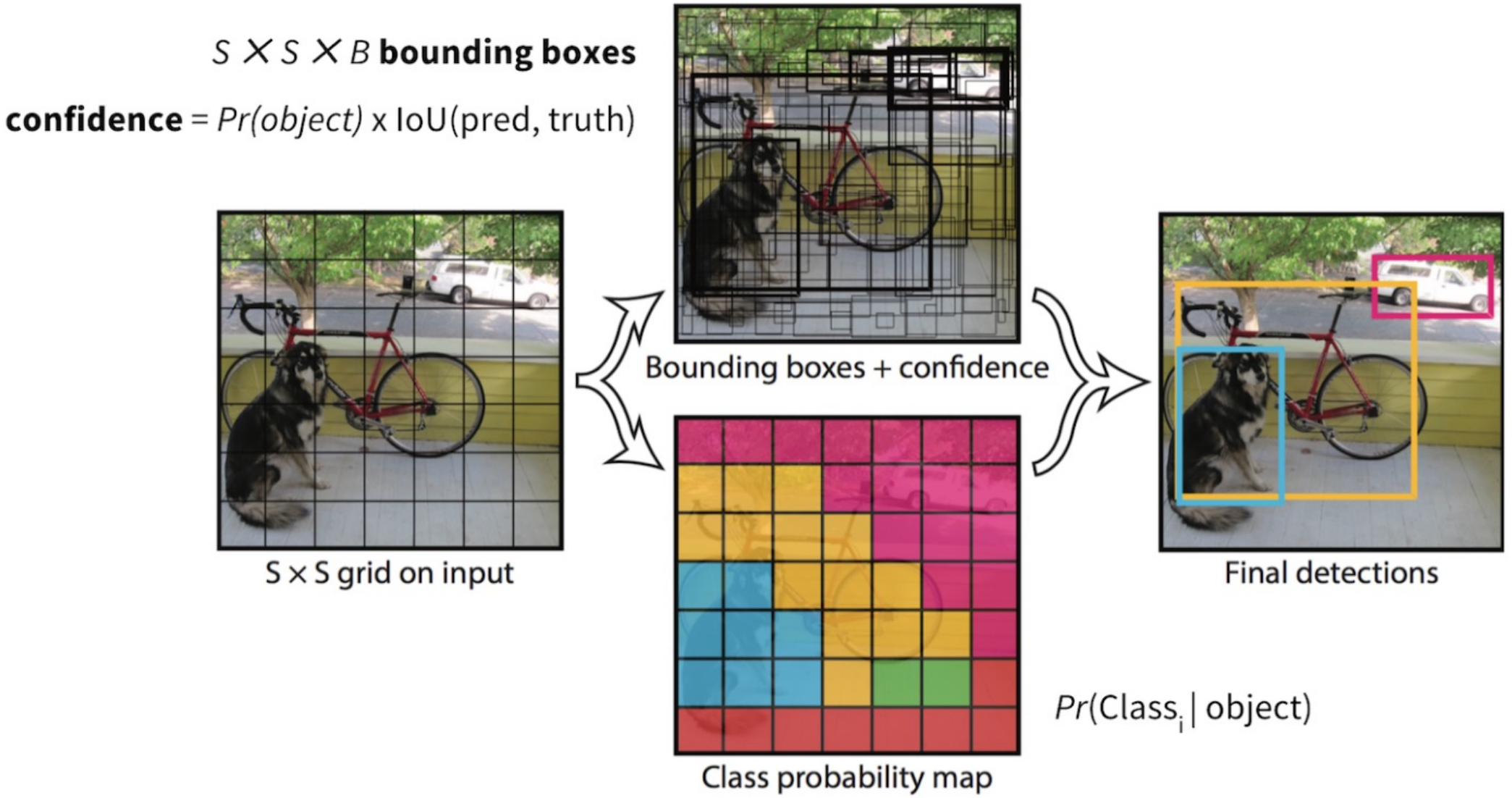


Detection of other object classes: cars, robots, hand, etc.

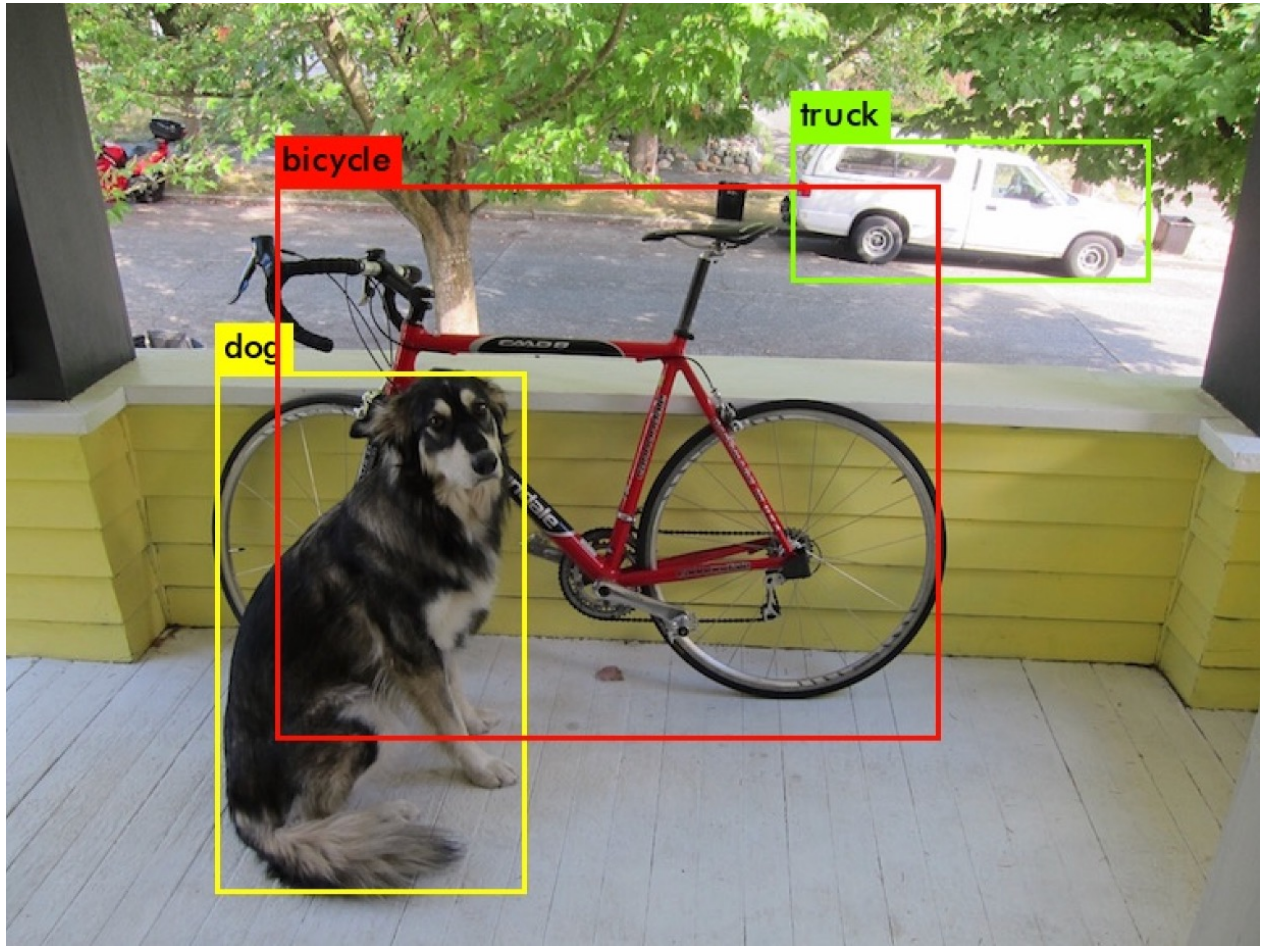


[Machine Vision and applications, 2008]

Approach 3:
Yolo Object detector. Deep learning method that predict bounding boxes (sliding window is not needed)



Yolo v3: https://youtu.be/MPU2HistivI

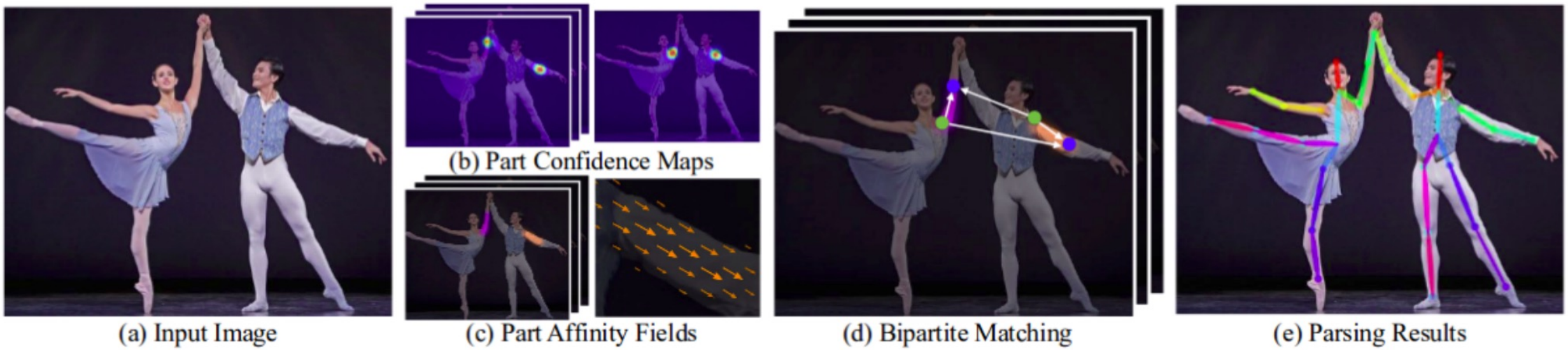


https://pjreddie.com/darknet/yolo/

Approach 4: DeepLearning + Association/Grouping Open Pose (person detection)

Top: Typical Top-Down approach. Bottom: Typical Bottom-Up approach. (Image Source)

Approach 4:
Open Pose (person detection)



Steps involved in human pose estimation using OpenPose. (Source)

Approach 4:
Open Pose (person detection)