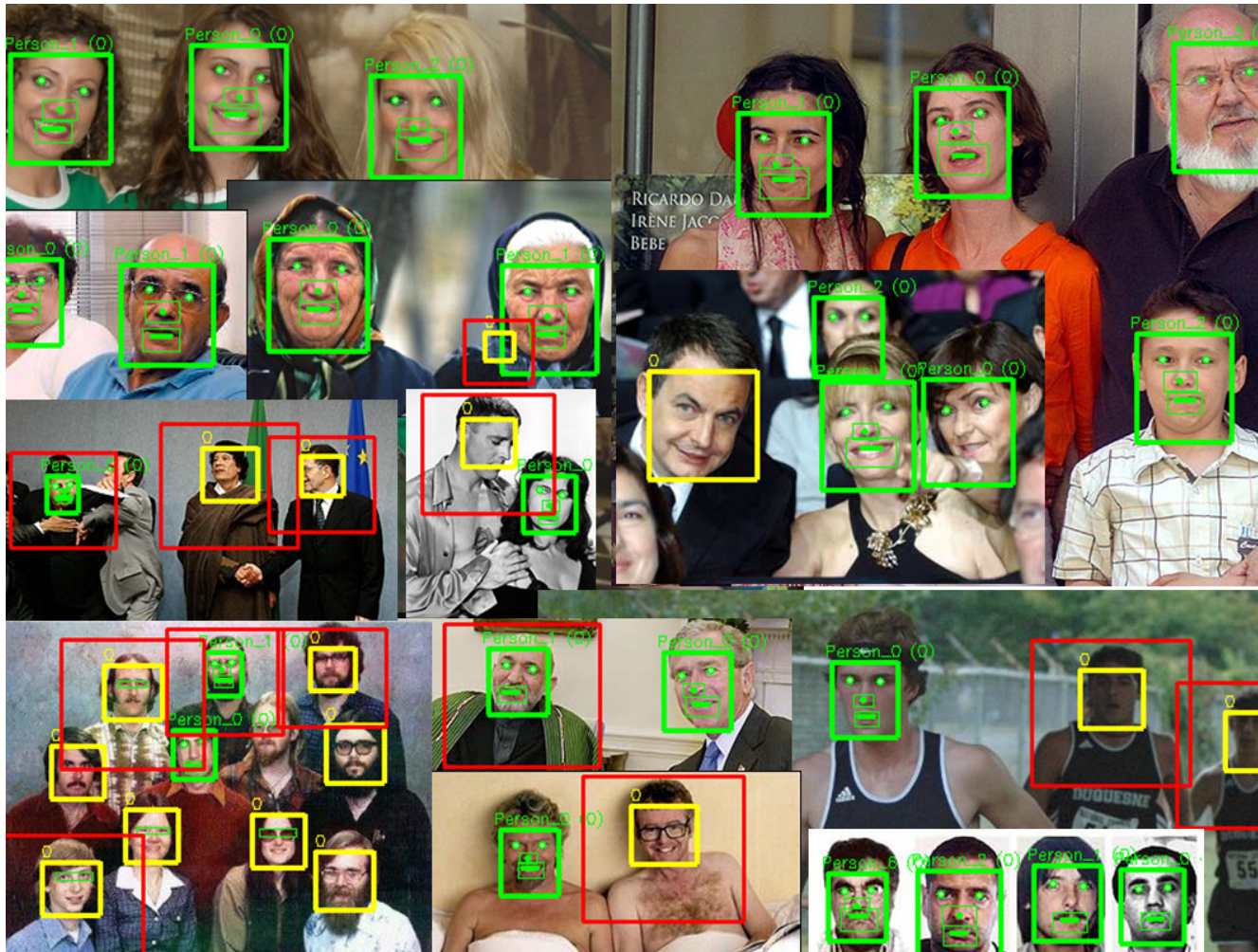


Aula de Inteligencia Artificial

Detección



World's Largest Selfie ©

Powered by Lumia 730



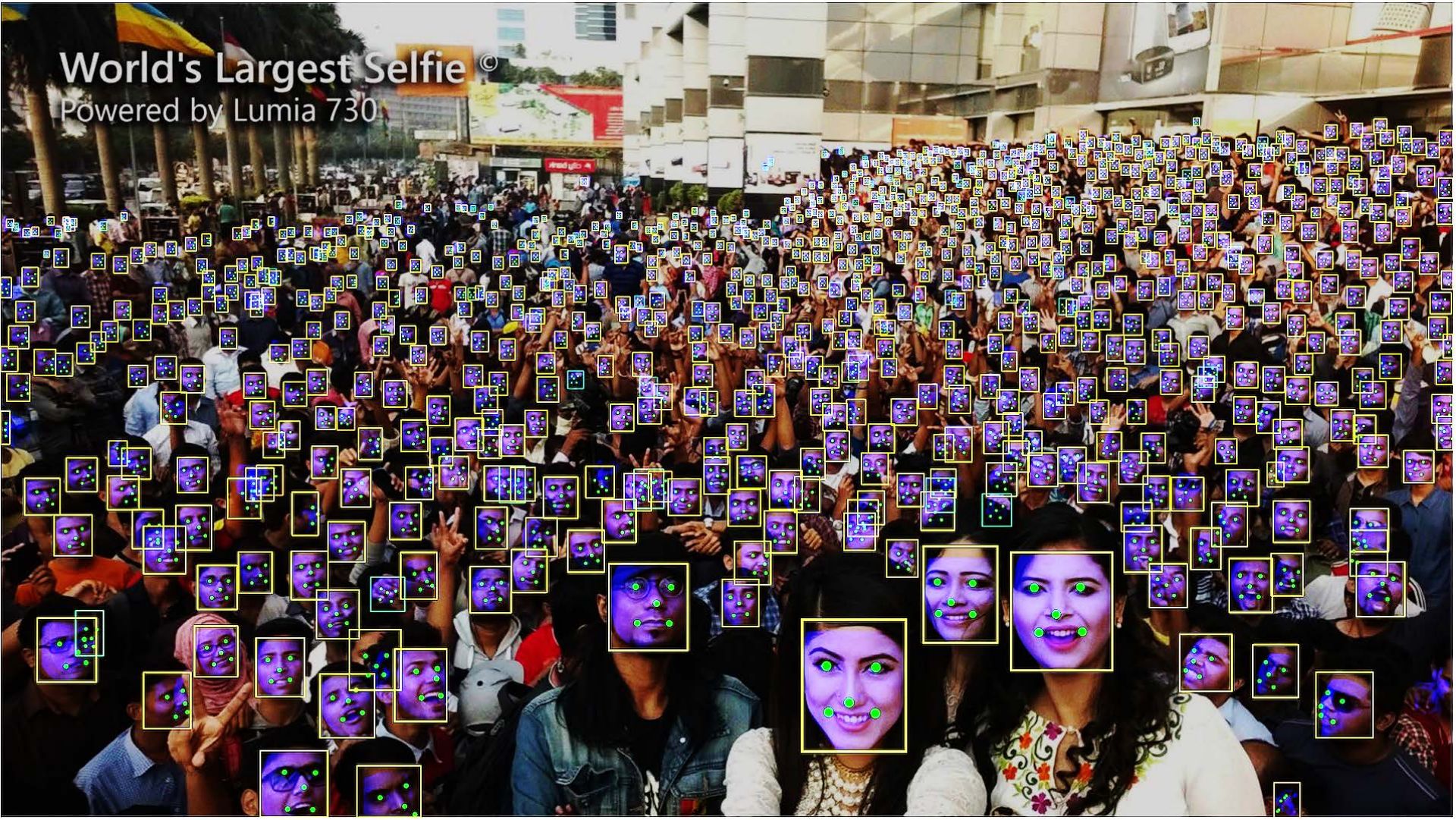
World's Largest Selfie ©

Powered by Lumia 730



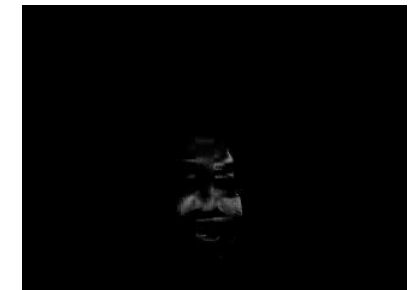
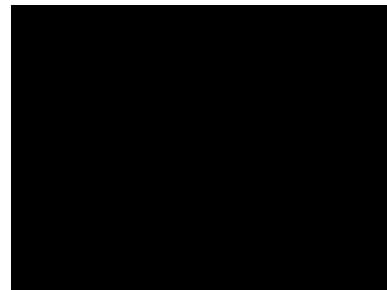
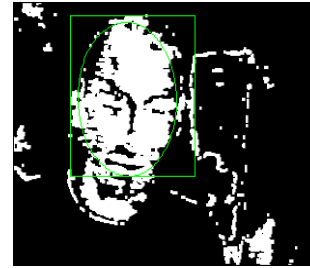
World's Largest Selfie ©

Powered by Lumia 730



Detección de caras

Color y movimiento, restricciones



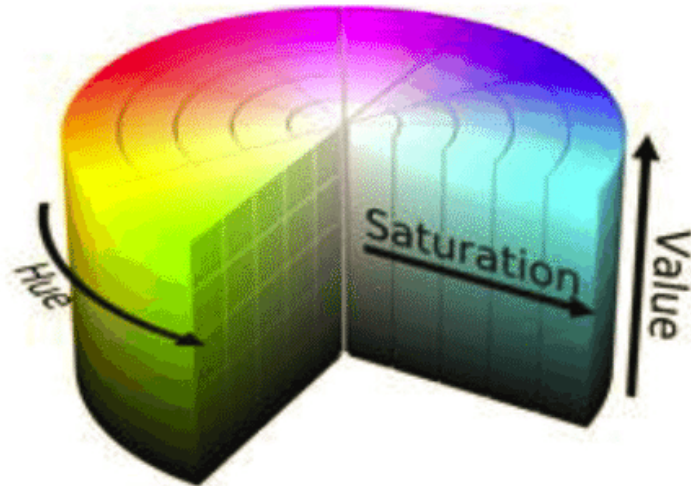
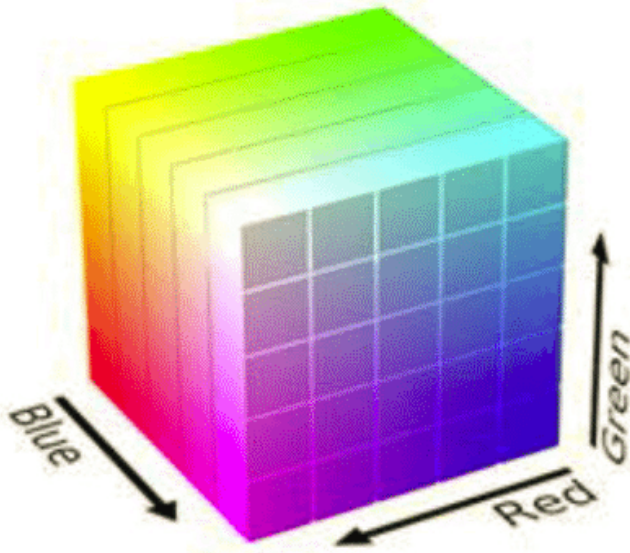
Esquema simple

Jones and Rehg [1]

Kruppa [2]

Detección de caras

Espacios de color

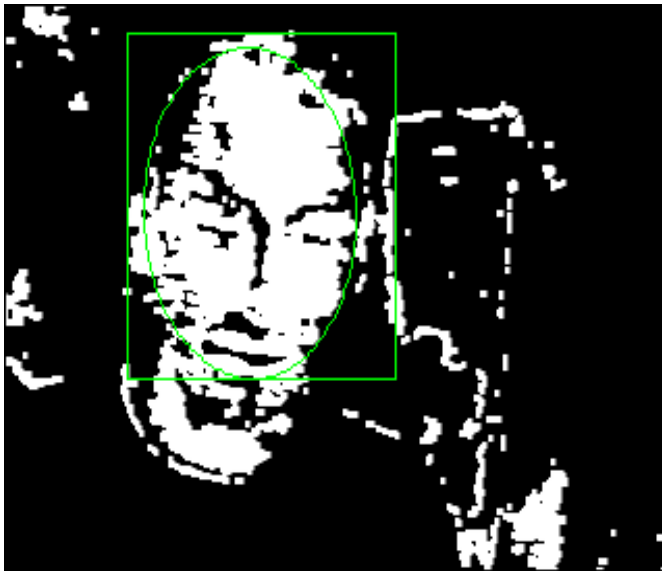


Tarea

Abrir proyecto AulalA_Detectores

Ejecutar *DetectaColor.py*

Determina los valores habituales en los espacios de color RGB y HSV para tu piel



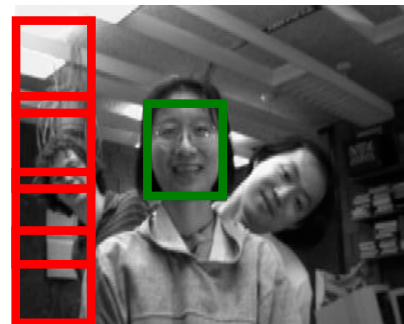
Detección de caras

Detección no basada en heurísticas

Ventana deslizante

Coste temporal del clasificador

Mayor velocidad

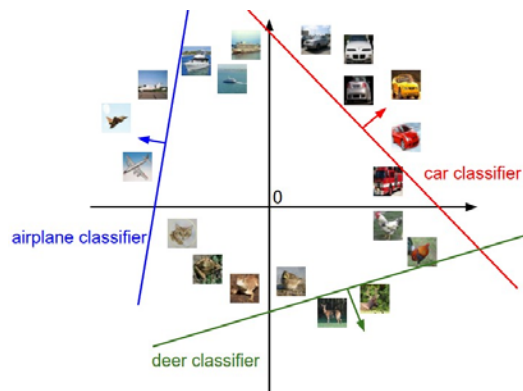


Detección de caras

Clasifica cada ventana

Clasificador

¿Qué caracteriza las X?

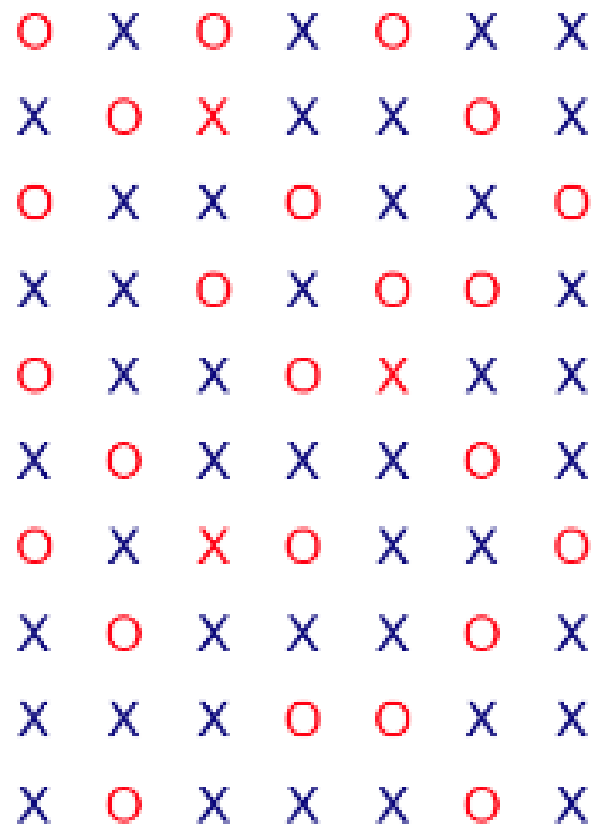
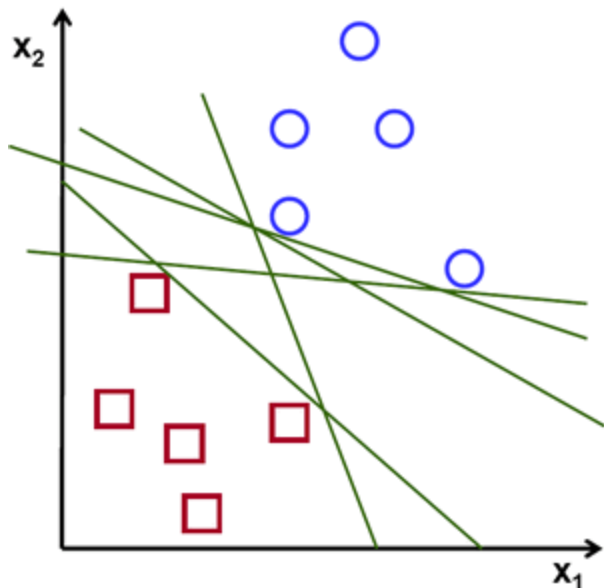


Detección de caras

Clasifica cada ventana

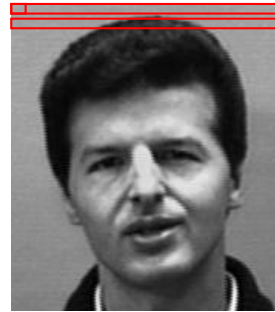
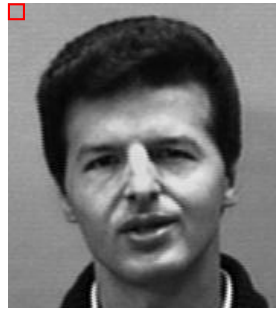
Clasificador

¿Qué caracteriza las **X**?

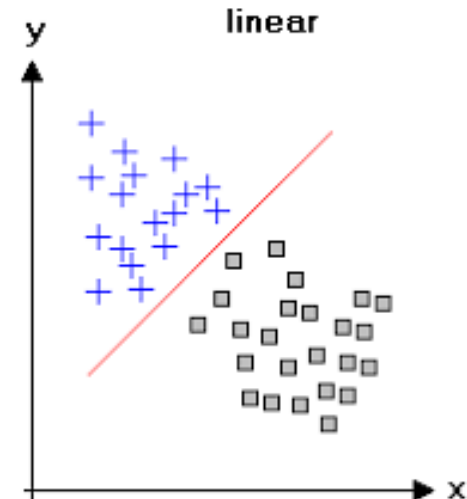


Detección de caras

Medidas de una imagen



$$x \in \mathbb{R}^n, \quad y \in \{\pm 1\}$$

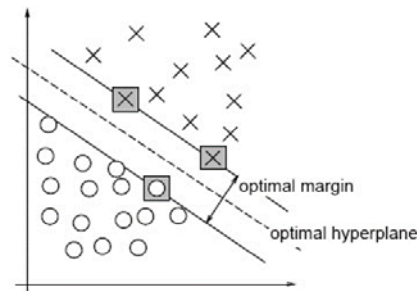
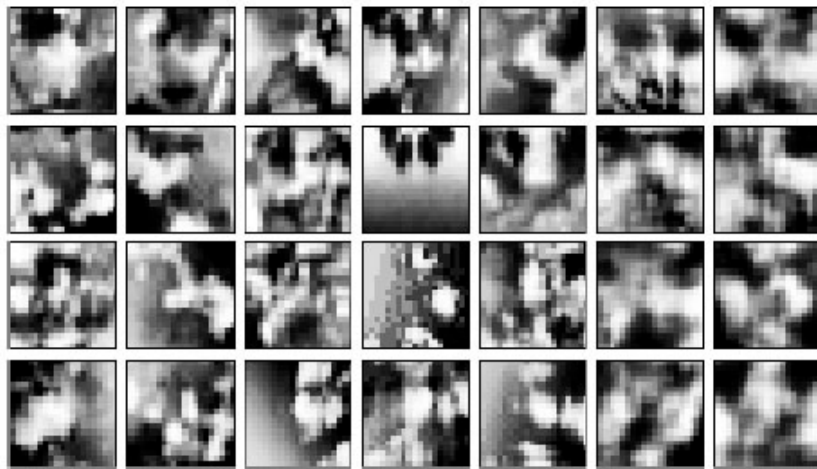


Detección de caras

Rowley y Kanade

Clasificador supervisado

Experiencia previa



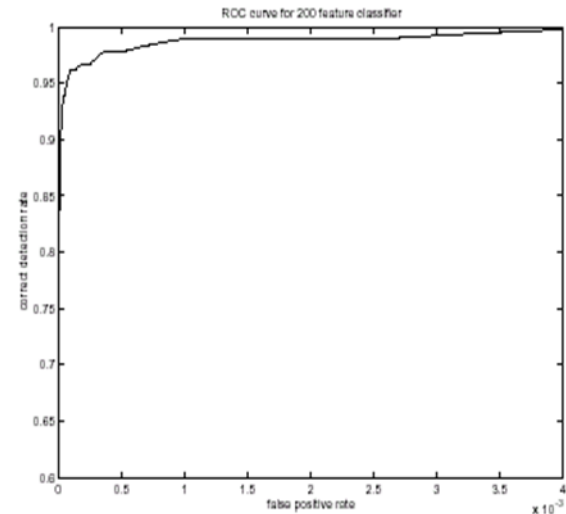
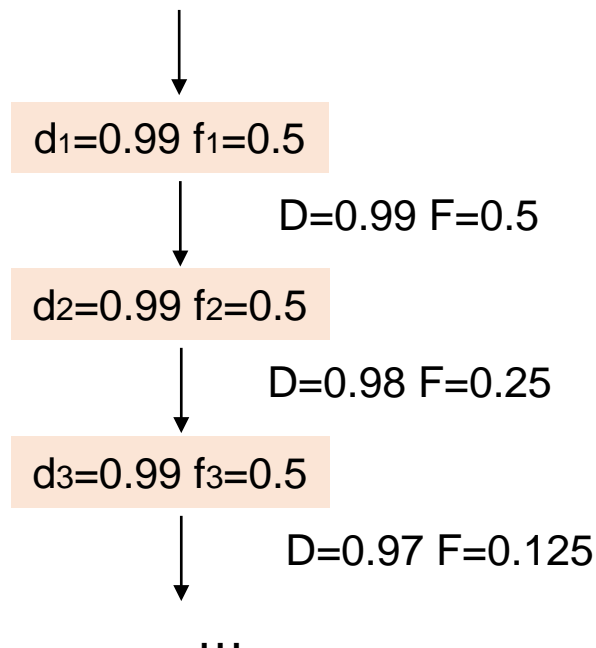
Detección de caras

Viola y Jones

Clasifica cada ventana

Clasificador en cascada, desecha zonas poco prometedores

Cascada clasificadores débiles

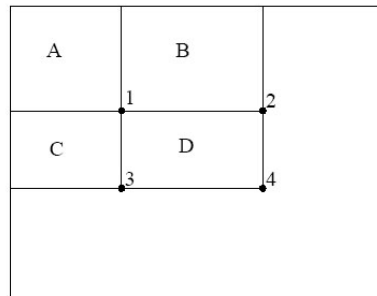
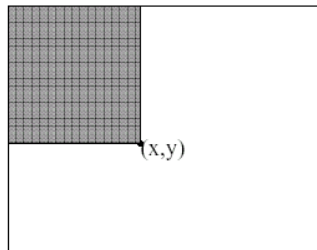


Detección de caras

Viola y Jones

Características de cómputo rápido

Imagen integral

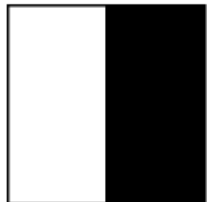


31	2	4	33	5	36
12	26	9	10	29	25
13	17	21	22	20	18
24	23	15	16	14	19
30	8	28	27	11	7
1	35	34	3	32	6

31	33	37	70	75	111
43	71	84	127	161	222
56	101	135	200	254	333
80	148	197	278	346	444
110	186	263	371	450	555
111	222	333	444	555	666

$$15 + 16 + 14 + 28 + 27 + 11 = 111$$

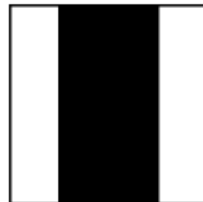
$$101 + 450 - 254 - 186 = 111$$



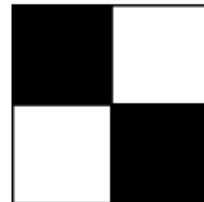
1



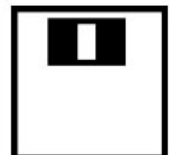
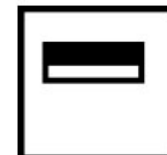
2



3



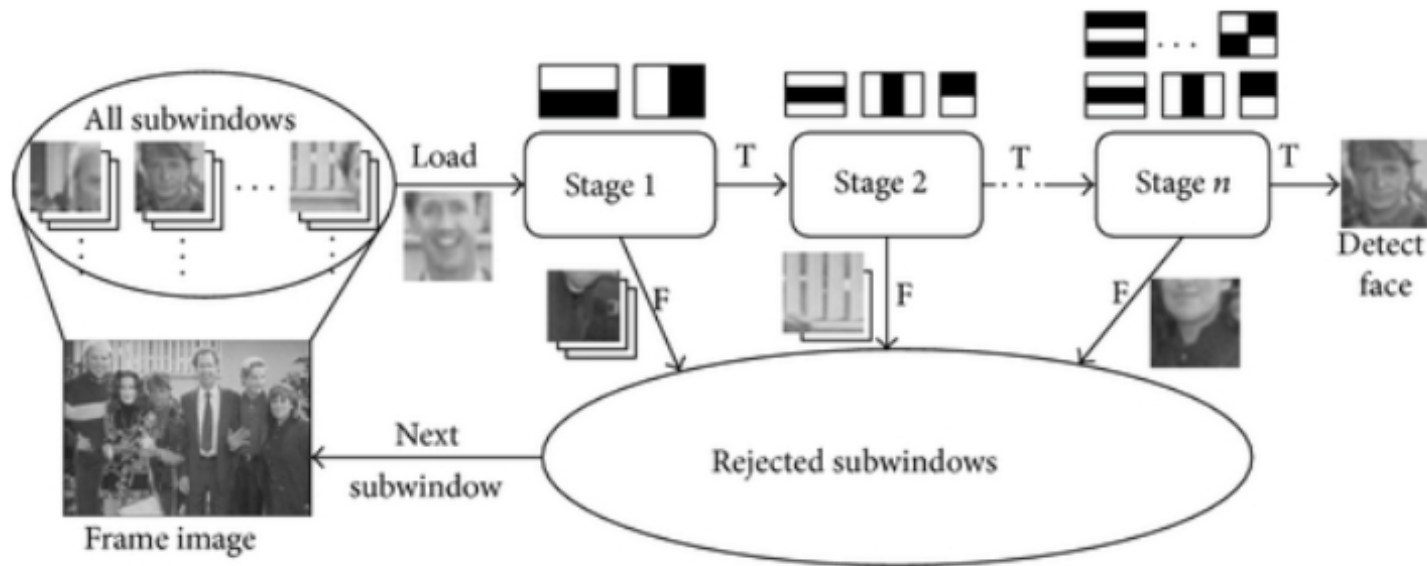
4



Detección de caras

Viola y Jones

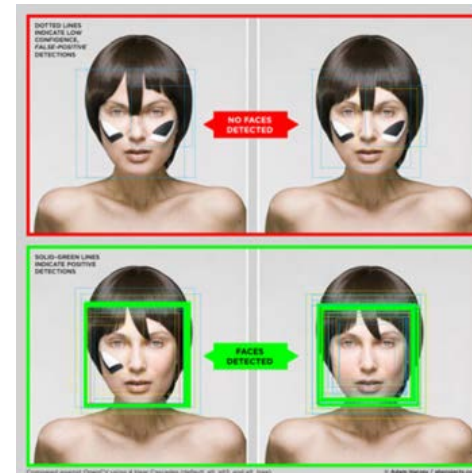
Esquema general



Fuente: [Cascade structure for Haar classifiers.](#)

Detección de caras

Viola y Jones



Detección de caras

Código python

```
import cv2

# Carga del clasificador para detección
cascada = cv2.CascadeClassifier('./haarcascade_frontalface_alt.xml')
# Carga la imagen
imagen = cv2.imread("worlds-largest-selfie.jpg")

# Conversión a grises
gris = cv2.cvtColor(imagen, cv2.COLOR_BGR2GRAY)

# Detecta objetos
caras = cascada.detectMultiScale(gris)

# Para cada cara detectada
for (x, y, w, h) in caras:
    # Dibuja contenedor
    imagen = cv2.rectangle(imagen, (x, y), (x + w, y + h), (255, 0, 0), 2)

cv2.imshow("Imagen", imagen)
```


Detección de caras

Código ejemplo Viola Jones: AulalA_Detectores

- DetectaVJenimagen
- DetectaVJenimagenysalva
- DetectaVJencam
- DetectaVJcarasyojos
- DetectaFacemarks (no funciona con versión actual opencv)

Repositorios clasificadores

- [opencv](#)
- [opencvcontrib](#)

Tarea

Detecta caras con sonrisa y dibuja un sol en su caso



Detección de caras

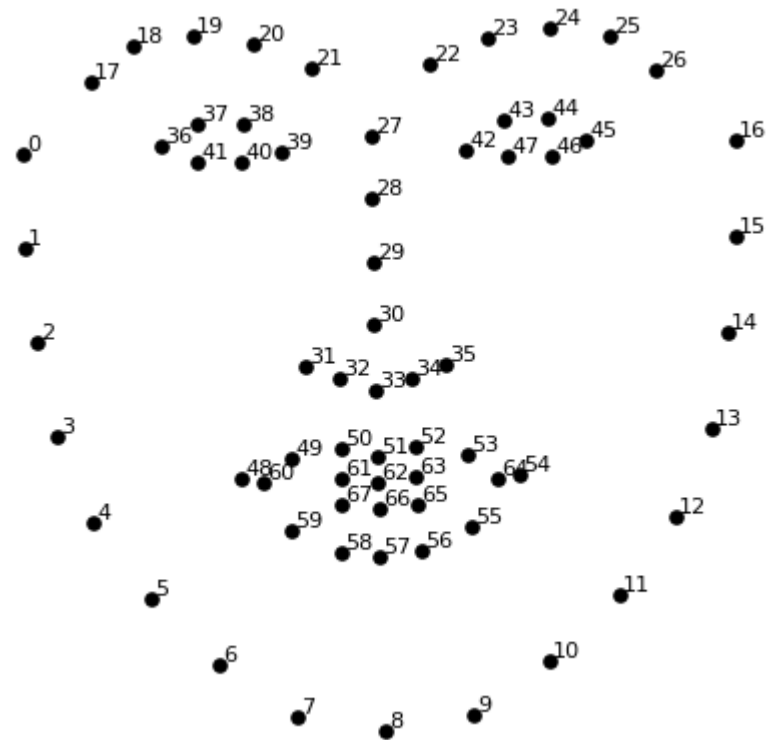
Tarea: Crea tu propio filtro

Detector dlib, HOG+SVM (requiere instalar dlib)

- DetectaDlibcaras
- DetectaDlibcarasnarizpayaso



Más que la cara, Zach
Lieberman, 2017

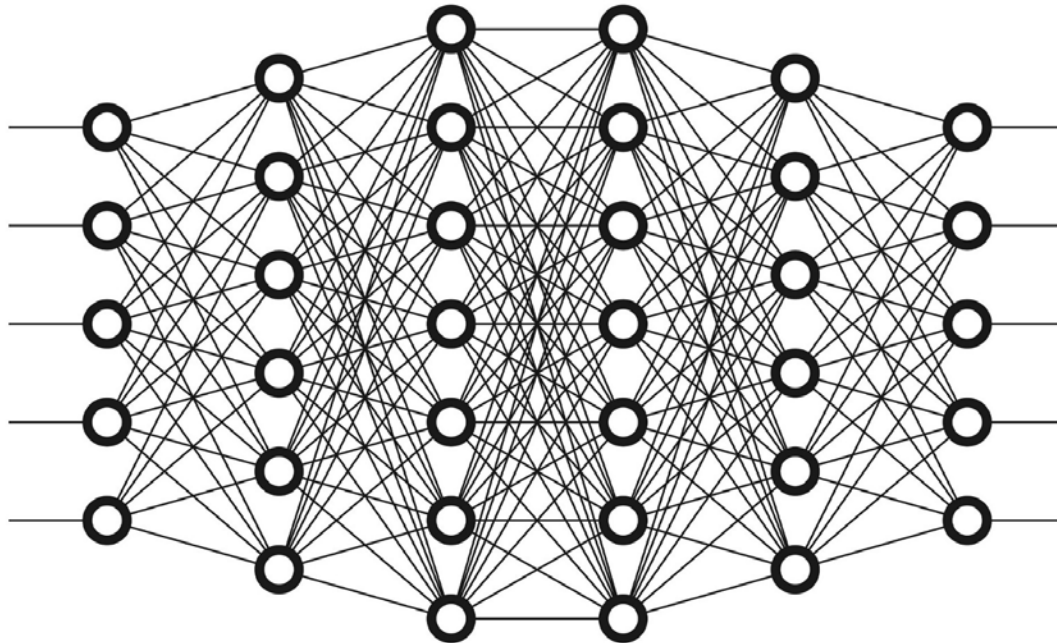


Detección de caras

Redes profundas

GPUs

Paralelización masiva



Tarea

Detección con comportamiento diferenciado por sexo/edad

Proyecto AulalA_Detectores

- DetectaDNNcaras
- DetectaDNNedadysexo
- DetectaVJedadysexo

Modelos clasificadores sexo y edad

- Sexo: https://www.dropbox.com/s/iyv483wz7ztr9gh/gender_net.caffemodel?dl=0
- Edad: https://www.dropbox.com/s/xfb20y596869vbb/age_net.caffemodel?dl=0

Referencias

- P. Viola and M. J. Jones. Rapid Object Detection using a Boosted Cascade of Simple Features. In Computer Vision and Pattern Recognition, 2001
- Rainer Lienhart and Jochen Maydt. An extended set of Haar-like features for rapid object detection. In IEEE International Conference on Image Processing, 2002
- Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu and Alexander C. Berg. SSD: Single Shot MultiBox Detector. In European Conference on Computer Vision, 2016