Dpnt. de Ciència de la Computació i Intel·ligència Artificial
Dpto. de Ciencia de la Computación e Inteligencia Artificial

Percepción automática

Reconocimiento de objetos

Universitat d'Alacant
Universidad de Alicante

Dpnt. de Ciència de la Computació i Intel·ligència Artificial
Dpto. de Ciencia de la Computación e Inteligencia Artificial

Índice

- Reconocimiento de objetos
- Reconocimiento mediante características
- Reconocimiento de caras

Universitat d'Alacant
Universidad de Alicante

Dpnt. de Ciència de la Computació i Intel·ligència Artificial
Dpto. de Ciencia de la Computación e Inteligencia Artificial

Introducción

- El reconocimiento de objetos consiste en, dado algún conocimiento (forma, apariencia, etc.) sobre uno o varios objetos y una imagen, encontrar qué objetos están en la imagen y dónde
- El reconocimiento es un proceso difícil debido a:
  - Presencia de otros objetos no modelados
  - Cambio de iluminación
  - Cambio de punto de vista del objeto
  - Oclusión
  - Escala

Universitat d'Alacant
Universidad de Alicante

Dpnt. de Ciència de la Computació i Intel·ligència Artificial
Dpto. de Ciencia de la Computación e Inteligencia Artificial

Ejemplos: reconocimiento facial

- Ampliamente extendido: cámaras, Picasa
- Se suele reconocer una posible posición en la imagen para una cara (mediante color de la piel, identificación de los ojos, etc.) y luego se reconoce la persona (con técnicas de aprendizaje)

Universitat d'Alacant
Universidad de Alicante

Dpnt. de Ciència de la Computació i Intel·ligència Artificial
Dpto. de Ciencia de la Computación e Inteligencia Artificial

Reconocimiento con características

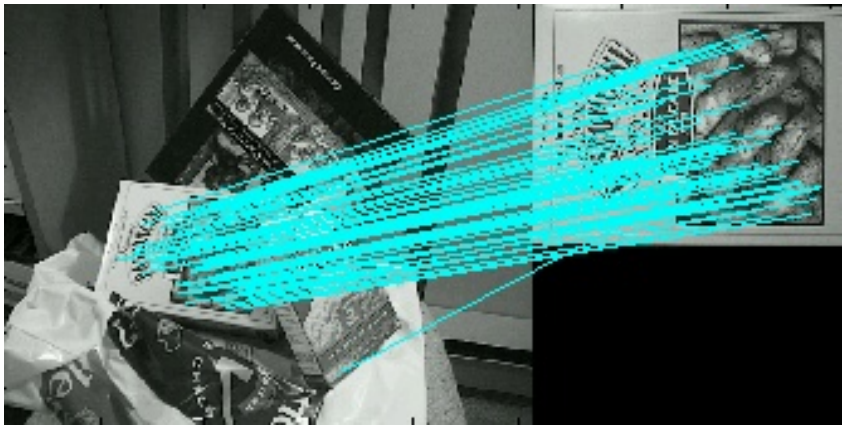
- Imaginemos que tenemos una imagen de un objeto a reconocer (modelo)
- Extraemos, por ejemplo, las características SIFT de dicha imagen. El objeto ahora es representado por sus características SIFT
- Ahora tenemos una nueva imagen (escena) donde queremos “buscar” ese objeto
- Extraemos los SIFT de esta nueva imagen
- Encontramos las correspondencias entre las características del modelo y de la imagen

Universitat d’Alacant
Universidad de Alicante

Dpnt. de Ciència de la Computació i Intel·ligència Artificial
Dpto. de Ciencia de la Computación e Inteligencia Artificial

Reconoc. Características: emparejamiento

- Esto se puede hacer calculando la distancia euclídea del descriptor
- Para cada característica del modelo:
  - Encontramos la característica de la escena cuya distancia euclídea esté por debajo de un cierto umbral
  - Ahora tenemos una correspondencia entre los descriptores del modelo y de la escena



Universitat d'Alacant
Universidad de Alicante

Dpnt. de Ciència de la Computació i Intel·ligència Artificial
Dpto. de Ciencia de la Computación e Inteligencia Artificial

Reconocimiento características: transformación

- Ahora debemos encontrar la transformación entre el modelo y la escena
- Para simplificar, vamos a ver cómo se puede obtener la transformación 2D-2D afín:

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} m_1 & m_2 \\ m_3 & m_4 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix}$$

donde las variables *m* son los parámetros de rotación y escala y los *t* son los de traslación

Universitat d'Alacant
Universidad de Alicante

Dpnt. de Ciència de la Computació i Intel·ligència Artificial
Dpto. de Ciencia de la Computación e Inteligencia Artificial
Universitat d'Alacant
Universidad de Alicante

Reconocimiento características: transformación

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} m_1 & m_2 \\ m_3 & m_4 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix}$$

- Tenemos un sistema de varias ecuaciones con varias incógnitas, reescribimos la ecuación de arriba

$$\begin{bmatrix} x & y & 0 & 0 & 1 & 0 \\ 0 & 0 & x & y & 0 & 1 \\ & & \cdots & & & \\ & & \cdots & & & \end{bmatrix} \begin{bmatrix} m_1 \\ m_2 \\ m_3 \\ m_4 \\ t_x \\ t_y \end{bmatrix} = \begin{bmatrix} u \\ v \\ \vdots \end{bmatrix}$$

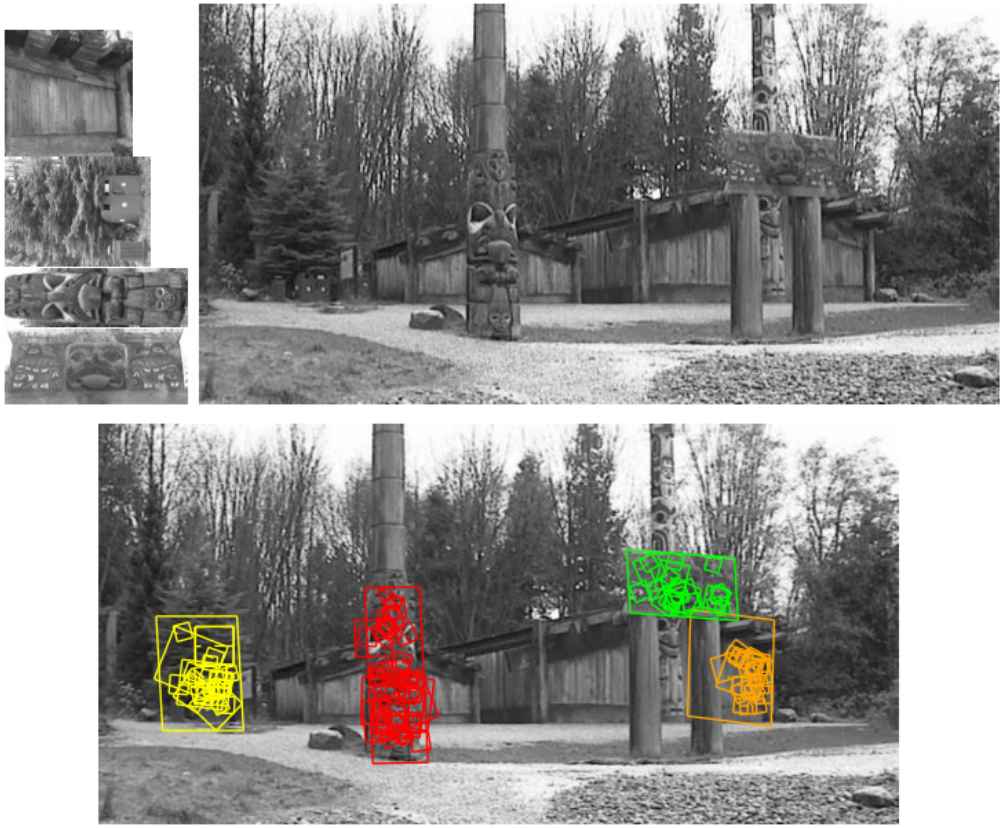
- Cada emparejamiento introduce dos nuevas filas a la primera y última matriz. Nombramos las matrices como:

$$\mathbf{Ax} = \mathbf{b}$$

Reconocimiento características: transformación

- Resolvemos el sistema anterior mediante mínimos cuadrados
- Consiste en encontrar la matriz *x* que minimiza el error cuadrático medio entre todos los emparejamientos
- Se resuelve este sistema:

$$\mathbf{x} = [\mathbf{A}^{\mathbf{T}}\mathbf{A}]^{-1}\mathbf{A}^{\mathbf{T}}\mathbf{b},$$



Dpnt. de Ciència de la Computació i Intel·ligència Artificial
Dpto. de Ciencia de la Computación e Inteligencia Artificial

Reconocimiento de caras

- Algoritmo de Viola&Jones: Robust Real-time Object Detection. International Journal of Computer Vision. 2001
- El objetivo es detectar caras, no reconocerlas
- Otro objetivo es que sea muy rápido: este método tarda pocos milisegundos en procesar una imagen
- El método tiene pocos falsos positivos y un alto porcentaje de detección correcta
- Sólo sirve para caras frontales o con poco giro
- Le afecta el cambio de luminosidad

Universitat d'Alacant
Universidad de Alicante

Dpnt. de Ciència de la Computació i Intel·ligència Artificial
Dpto. de Ciencia de la Computación e Inteligencia Artificial

Estructura general del método

- Extracción de características
  - Uso de imagen integral!!
  - Extracción de muchísimas características
- Selección de características
  - Algoritmo AdaBoost de entrenamiento
- Cascada de clasificadores
  - Conseguir más velocidad

Universitat d'Alacant
Universidad de Alicante

Dpnt. de Ciència de la Computació i Intel·ligència Artificial
Dpto. de Ciencia de la Computación e Inteligencia Artificial

Extracción de características

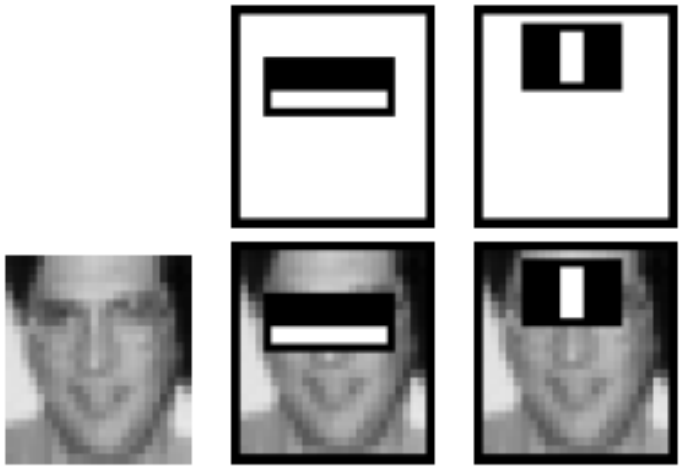
- Las caras comparten algunas características comunes:
  - La región del puente de la nariz es más clara que la de los ojos
  - La región de los pómulos es también más clara que los ojos
- Características a buscar:
  - Localizar nariz-ojos
  - Valores: claro-oscuro

Universitat d'Alacant
Universidad de Alicante

Dpnt. de Ciència de la Computació i Intel·ligència Artificial
Dpto. de Ciencia de la Computación e Inteligencia Artificial
Universitat d'Alacant
Universidad de Alicante

Características en rectángulo

- Valor de la característica
  - ∑(píxeles en área negra) - ∑(píxeles en área blanca)
- Tres tipos: dos, tres, cuatro-rectángulos



- Cada característica está relacionada con una localización especial en la sub-ventana
- Cada característica puede tener cualquier tamaño y orientación

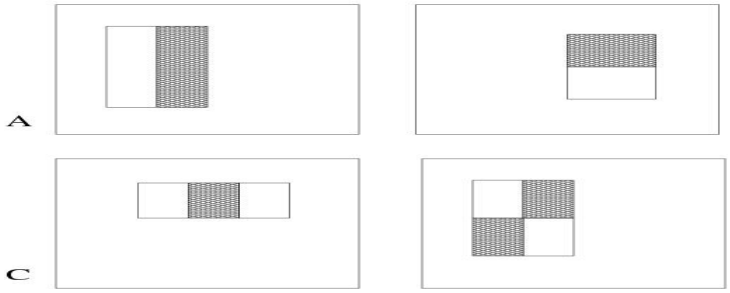
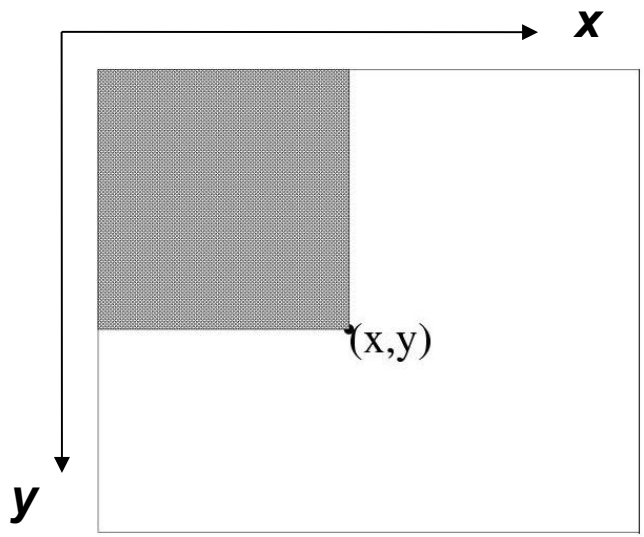


Imagen Integral

- Dada una resolución de ventana de 24x24 existen 180000 posibles características!
- Por ello, es necesario un cálculo rápido
- Introducen el uso de la imagen integral: cada punto en la imagen integral es la suma de todos los píxeles de la imagen original a la izquierda y arriba de ese punto

$$imagen\ original = I(x, y)$$
$$imagen\ integral = II(x, y) = \sum_{x' \le x,\, y' \le y} I(x', y')$$



Dpnt. de Ciència de la Computació i Intel·ligència Artificial
Dpto. de Ciencia de la Computación e Inteligencia Artificial

Cálculo de la imagen Integral

- Se puede calcular en un solo paso:

$$s(x,y) = s(x,y-1) + i(x,y)$$
$$ii(x,y) = ii(x-1,y) + s(x,y)$$

donde *s(x,y)* es la suma acumulada de la columna. Hay que tener en cuenta las primeras fila y columna

Imagen

0	1	1	1
1	2	2	3
1	2	1	1
1	3	1	0

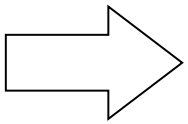


Imagen integral

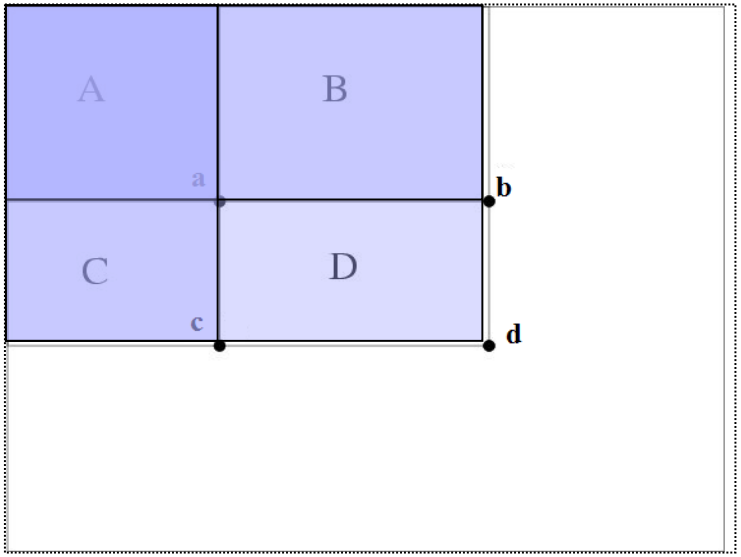
0	1	2	3
1	4	7	11
2	7	11	16
3	11	16	21

Universitat d'Alacant
Universidad de Alicante

Dpnt. de Ciència de la Computació i Intel·ligència Artificial
Dpto. de Ciencia de la Computación e Inteligencia Artificial

¿Qué ventaja tiene la imagen integral?

- Como lo que queremos calcular es la suma de los píxeles dentro de un rectángulo, la imagen integral permite hacer este cálculo con tres operaciones básicas
- Para calcular características con dos, tres o cuatro rectángulos es necesario realizar 6, 8 y 9 referencias a la imagen, respectivamente (ejemplo con 2)



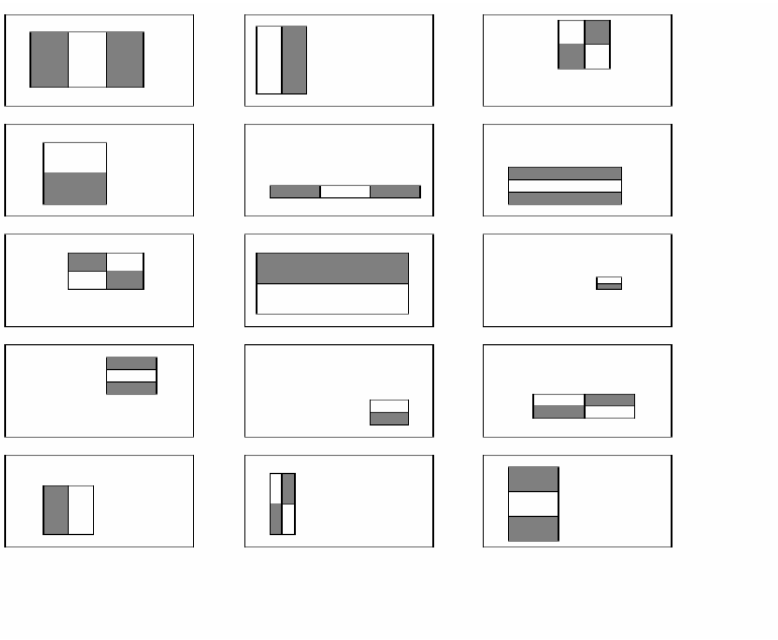
ii(a) = A
ii(b) = A+B
ii(c) = A+C
ii(d) = A+B+C+D
D = ii(d)+ii(a)-ii(b)-ii(c)

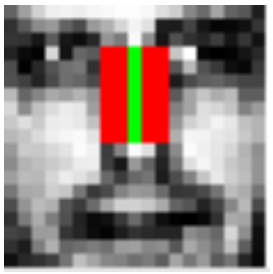
Universitat d'Alacant
Universidad de Alicante

Dpnt. de Ciència de la Computació i Intel·ligència Artificial
Dpto. de Ciencia de la Computación e Inteligencia Artificial
Universitat d'Alacant
Universidad de Alicante

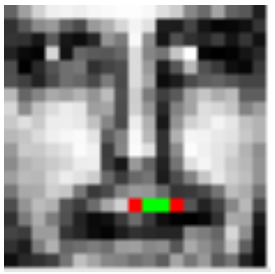
Selección de características

- Demasiadas características: en una ventana 24x24 hay 180000 posibles combinaciones de tamaño y orientación
- Hay que seleccionar aquellas que sean relevantes para detectar una cara
- Para ello, usaremos el algoritmo AdaBoost





Carac. relevante



Carac. irrelevante

Dpnt. de Ciència de la Computació i Intel·ligència Artificial
Dpto. de Ciencia de la Computación e Inteligencia Artificial

Algoritmo AdaBoost

- Construye un clasificador "fuerte" a partir de una combinación lineal de clasificadores "débiles"

$$F(x) = \alpha_1 f_1(x) + \alpha_2 f_2(x) + \alpha_3 f_3(x) + \ldots$$

Clasificador débil

Clasificador fuerte

Imagen

Pesos

- En el caso de la selección de características, cada característica es usada como clasificador débil
- AdaBoost es usado para seleccionar las características y para construir el clasificador fuerte

Universitat d'Alacant
Universidad de Alicante

Dpnt. de Ciència de la Computació i Intel·ligència Artificial
Dpto. de Ciencia de la Computación e Inteligencia Artificial

Algoritmo AdaBoost

- Un clasificador *h(x)* está compuesto por una característica *f*, un umbral θ y un signo *p* que permite cambiar el signo de la igualdad

$$h_j(x) = \begin{cases} 1 & \text{if } p_j f_j(x) < p_j \theta_j \\ 0 & \text{otherwise} \end{cases}$$

- Dado un conjunto de entrenamiento (x_1, y_1), …, (x_n,y_n), donde *x* es una imagen (de tamaño 24x24) e *y* toma valores 1 ó 0 indicando si es una cara o no.
- Cada imagen tiene asociada un peso *w*

Universitat d'Alacant
Universidad de Alicante

Dpnt. de Ciència de la Computació i Intel·ligència Artificial
Dpto. de Ciencia de la Computación e Inteligencia Artificial

Idea del método

- Los pesos de las imágenes son iguales
- Hacer T veces
  - Escoger el clasificador más eficiente (con menor error). Este clasificador formará parte del clasificador final.
  - Actualizamos los pesos para dar más peso a los ejemplos (imágenes) que fueron mal clasificadas en este paso (hacemos que el siguiente clasificador se “focalice” en esos errores).
- El clasificador final estará formado por una combinación lineal de los clasificadores encontrados en las iteraciones anteriores, ponderados por un peso directamente proporcional a lo bien que han clasificado

Universitat d’Alacant
Universidad de Alicante

Dpnt. de Ciència de la Computació i Intel·ligència Artificial
Dpto. de Ciencia de la Computación e Inteligencia Artificial

Seudo-algoritmo

- Inicializar los pesos
  - Para aquellas imágenes con *y=1*, *w=1/2m*, con *y=0, w=1/2n*, donde *m* es el número de imágenes con caras y *n* sin caras *l=m+n*
- Repetir desde t=0 hasta T:
  - Normalizar los pesos $w_i = \frac{w_i}{\sum_{j=1}^{l} w_j}$
  - Calcular el error al usar cada uno de los clasificadores restantes
  - Elegir el clasificador con menor error $\varepsilon_j = \sum_i w_i |h_j(x_i) - y_i|$
  - Actualizar los pesos $w_{t+1,i} = w_{t,i} \beta_t^{1-e_i}$ donde $e_i=0$ si el ejemplo *i* se clasificó correctamente y $e_i=1$ en otro caso y $\beta_t = \frac{\varepsilon_t}{1-\varepsilon_t}$
- El clasificador final es:

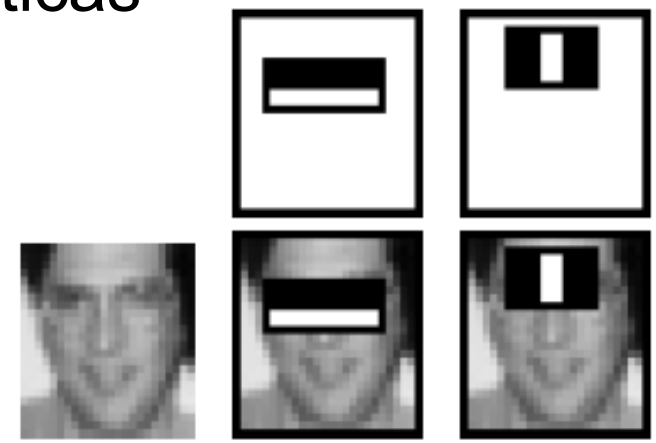
$$h(x) = \begin{cases} 1 & \sum_{t=1}^{T} \alpha_t h_t(x) \geq \frac{1}{2} \sum_{t=1}^{T} \alpha_t \\ 0 & en\ otro\ caso \end{cases}$$

donde $\alpha_t = \log(\frac{1}{\beta_t})$

Universitat d'Alacant
Universidad de Alicante

AdaBoost

- Los autores de este método seleccionaron 200características
- Los experimentos obtuvieron un 95% de aciertos y un falso positivo en más de 14000 imágenes
- Obtuvieron un tiempo de 0.7 segundos para procesar una imagen
- Otros métodos podían obtener ratios similares de aciertos y menos falsos positivos, aunque no eran tan rápidos
- Las dos primeras características fueron:



Dpnt. de Ciència de la Computació i Intel·ligència Artificial
Dpto. de Ciencia de la Computación e Inteligencia Artificial

Un paso más allá

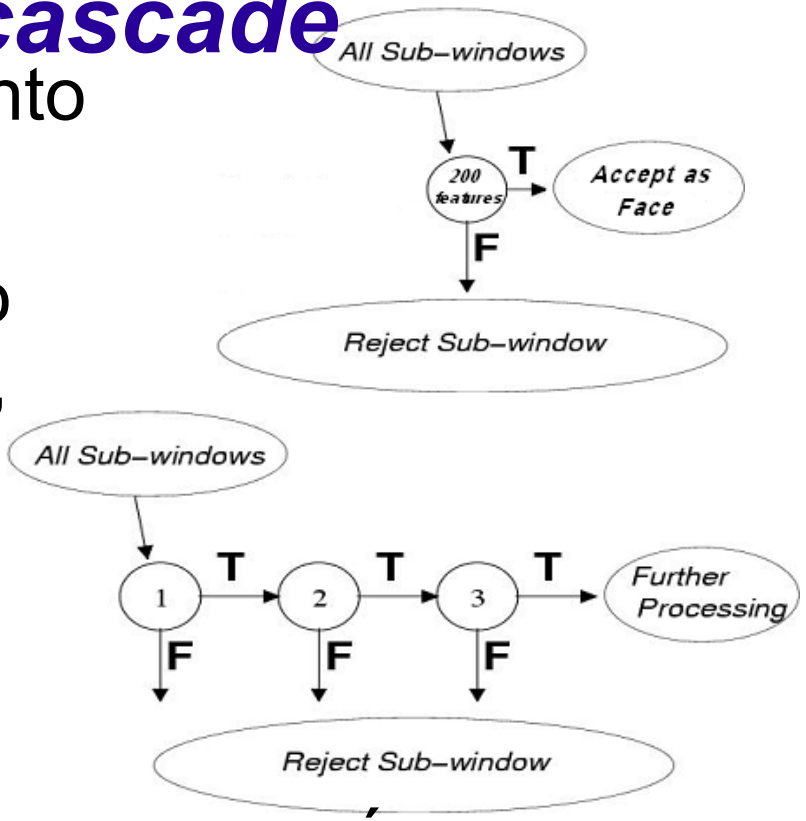
- Los autores buscaban más eficiencia
- En una simple imagen, potencialmente sólo el 0,01% de las ventanas analizadas son caras, dedicando mucho tiempo a procesar posibles positivos que no lo son
- La idea es buscar algún método que permita eliminar rápidamente lo que no es una cara, sin perder el porcentaje de aciertos
- Un clasificador con dos características tiene un acierto del 100% con un 50% de falsos positivos, pero es muy rápido (10-20 sumas)

Universitat d'Alacant
Universidad de Alicante

Dpnt. de Ciència de la Computació i Intel·ligència Artificial
Dpto. de Ciencia de la Computación e Inteligencia Artificial

The attentional cascade

- El esquema usado crea un conjunto de clasificadores por niveles
- En el primer nivel, se entrena uno con pocas características, rápido, que permita eliminar la mayoría de no caras
- Los que sobrevivan pasan al siguiente nivel, que tendrá otro clasificador con más características, pero más especializado
- De media, se necesita menos procesamiento para eliminar no caras



Universitat d'Alacant
Universidad de Alicante

Dpnt. de Ciència de la Computació i Intel·ligència Artificial
Dpto. de Ciencia de la Computación e Inteligencia Artificial

The Attentional Cascade

- Para definir este mecanismo, hay que determinar:
  - El número de niveles
  - Número de características en cada nivel
  - El umbral de cada clasificador
- El problema es muy complicado de resolver
- Los autores proponen una heurística:
  - El usuario fija tanto el nivel de falsos positivos aceptables como el ratio de detección.
  - Cada nivel reduce el ratio de falsos positivos y decrece el ratio de detección.
  - Se asigna un valor de reducción mínimo para falsos positivos y otro para la detección
  - Cada nivel se entrena con esos valores, añadiendo características hasta que se alcanza el valor deseado
  - Se añaden niveles hasta alcanzar el valor tanto de falsos positivos como de ratio de detección

$$h(x) = \begin{cases} 1 & \sum_{t=1}^{T} \alpha_t h_t(x) \geq \left(\frac{1}{2}\sum_{t=1}^{T} \alpha_t\right) \\ 0 & en\ otro\ caso \end{cases}$$

Universitat d'Alacant
Universidad de Alicante

Dpnt. de Ciència de la Computació i Intel·ligència Artificial
Dpto. de Ciencia de la Computación e Inteligencia Artificial

Resultados

- Probado con el conjunto de prueba MIT+MCU
- Una imagen de 384x288 en un PC (del año 2001) tardó 0.067 segundos

	False detections						
Detector	**10**	**31**	**50**	**65**	**78**	**95**	**167**
Viola-Jones	76,1%	88,4%	91,4%	92,0%	92,1%	92,9%	93,9%
Rowley-Baluja-Kanade	83,2%	86,0%	-	-	89,2%	89,2%	90,1%
Schneiderman-Kanade	-	-	-	94,4%	-	-	-
Roth-Yang-Ajuha	-	-	-	-	-	-	-

Universitat d'Alacant
Universidad de Alicante

Dpnt. de Ciència de la Computació i Intel·ligència Artificial
Dpto. de Ciencia de la Computación e Inteligencia Artificial

Referencias

- Método de Viola&Jones para reconocimiento de caras: http://research.microsoft.com/~viola/Pubs/Detect/violaJones_IJCV.pdf
- http://www.pigeon.psy.tufts.edu/avc/kirkpatrick/default.htm

Universitat d'Alacant
Universidad de Alicante