



Universitat de Barcelona

Grado en Ingeniería Informática  
Proyecto Fin de Grado

# Multi-clasificación discriminativa por partes mediante Códigos Correctores de Errores

Autor: Juan Carlos Ortega Pérez

Directores: Sergio Escalera y Miguel Ángel Bautista

# Contenidos

- ▶ Introducción.
- ▶ Contexto y Motivación.
- ▶ Procesamiento de la BD HuPBA.
- ▶ Metodología.
- ▶ Funcionamiento del Proceso.
- ▶ Parámetros y Resultados.
- ▶ Conclusiones.

# Contexto y Motivación

# Contexto y Motivación

- ▶ ¿Qué queremos hacer?
  - Reconocer a personas y segmentar extremidades.
- ▶ ¿Por qué?
  - Campo aún por investigar.
  - En evolución, gracias a sistemas como Kinect.
- ▶ ¿Cómo?
  - Procesando la base de datos HuPBA.
  - Entrenando distintos tipos de Clasificadores.
  - Usando utilidades de segmentación.

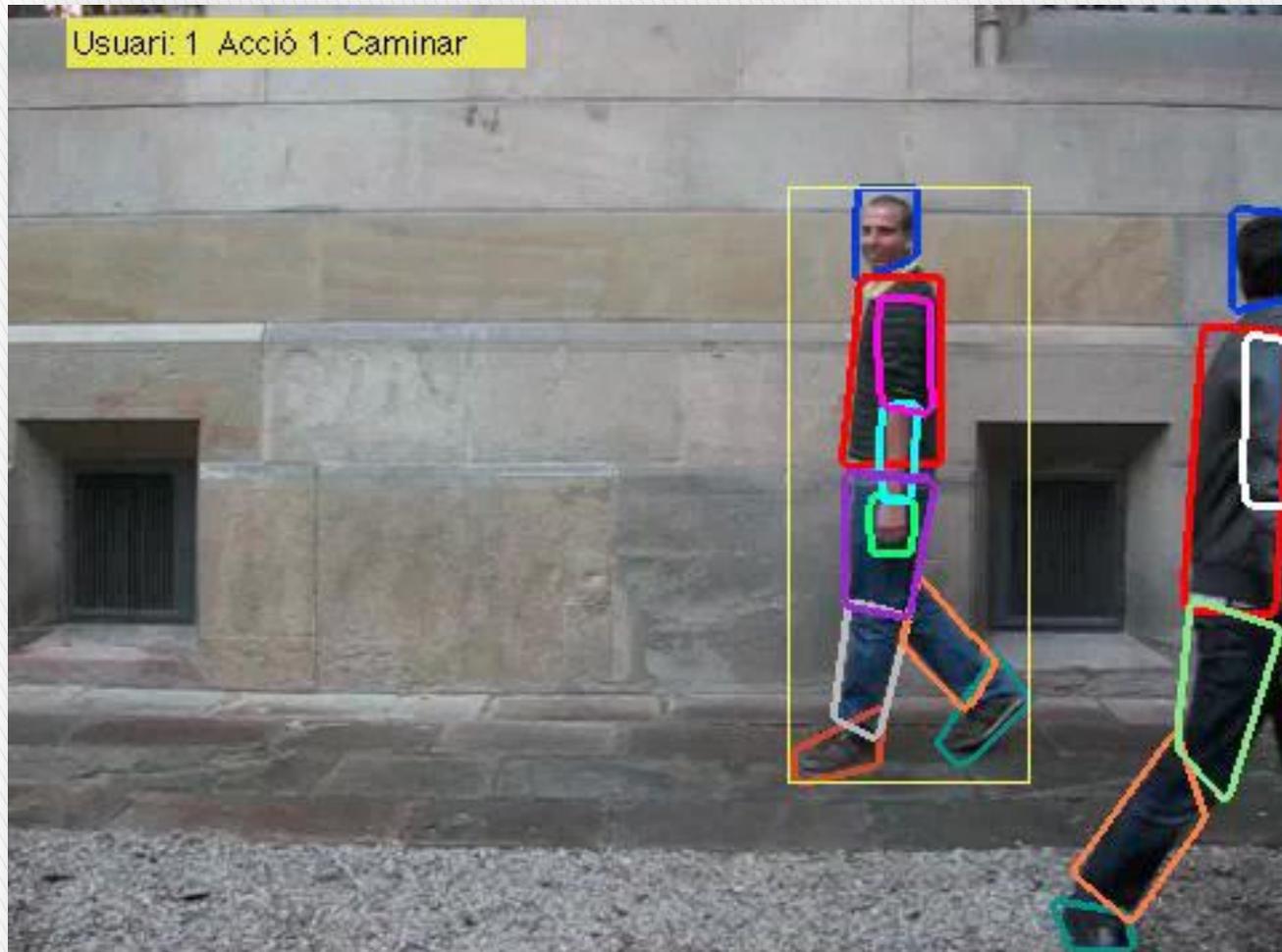
# Procesamiento de la BD HuPBA

# Procesamiento de la BD HuPBA

- ▶ ¿Qué es HuPBA?
  - Creada en la Universidad de Barcelona.
  - Compuesta por 9 actores en un mismo escenario.
  - Inclusión de distintas poses y gestos.
  - **288827** Imágenes distintas.
  
- ▶ Contenido:



# Procesamiento de BD HuPBA



# Procesamiento de BD HuPBA

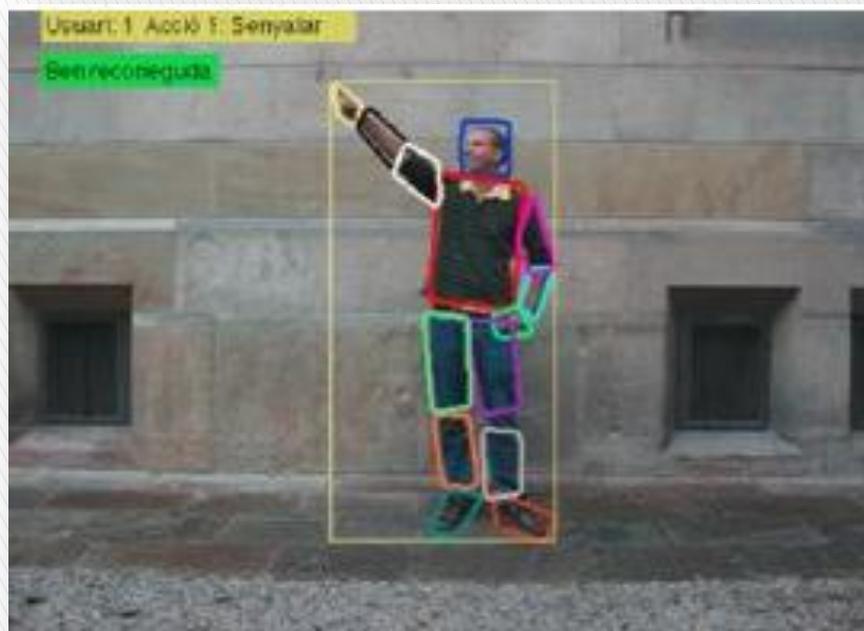
- ▶ ¿Problemas?
  - **Orientación** de las extremidades.
  - Extremidades demasiado **pequeñas**.
  - Conjunto de datos demasiado **voluminoso**.
- ▶ ¿Solución?
  - Uso de **orientación predominante**.
  - **Reescalar**, en caso necesario.
  - Obtener aquellas imágenes que realmente sean **distintas**.



# Procesamiento de BD HuPBA

## ▶ Conjunto completo:

- Cabeza
- Torso
- Brazos
- Antebrazos
- Muslos
- Piernas
- Manos
- Pies



## ▶ Conjunto utilizado:

- Cabeza
- Torso
- Brazos
- Antebrazos
- Muslos
- Piernas

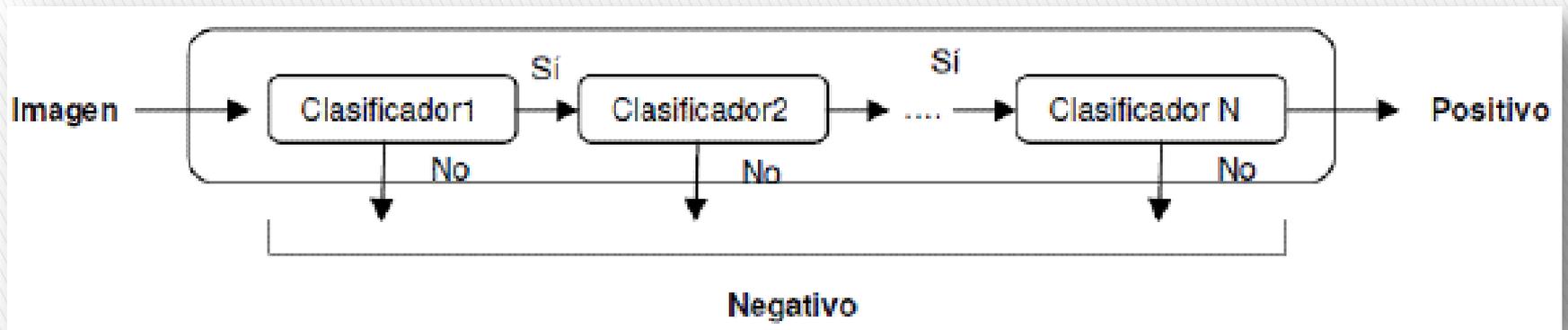
# Metodología

# Metodología

- ▶ Para llevar a cabo todo el proceso, necesitamos:
  - Clasificadores:
    - Cascadas **Adaboost** + **Haar Features**.
    - Clasificadores **SVM** + **HOG Descriptors**.
  - Sistemas de segmentación:
    - GrabCut.
    - GraphCut.

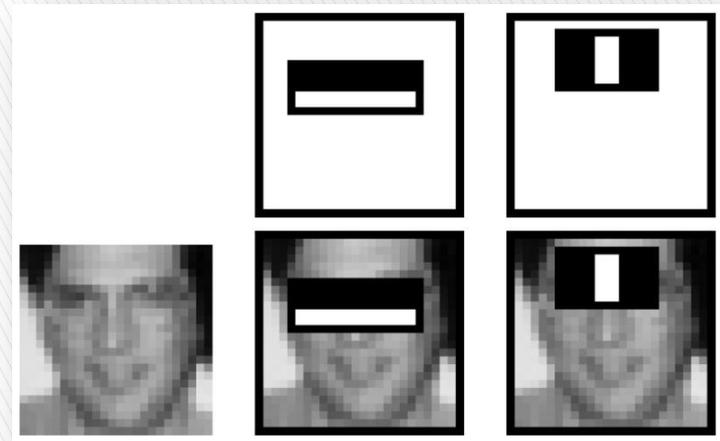
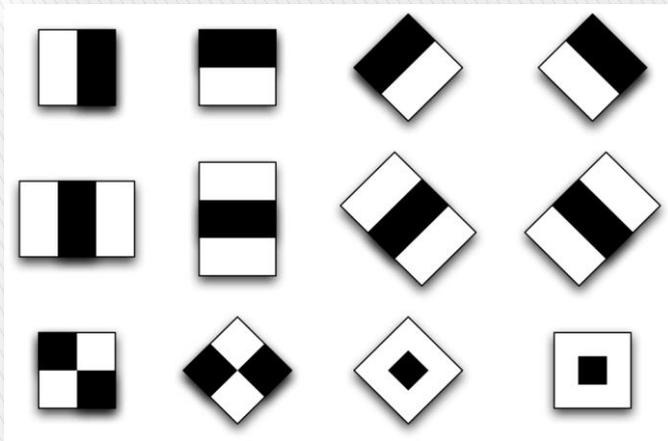
# Cascadas Adaboost + Haar Features

- ▶ Utilizadas para la detección de rostro.
- ▶ Desarrollado por Viola–Jones.
- ▶ Compuestas por  $n$  clasificadores débiles.
- ▶ Uso de **características Haar**.



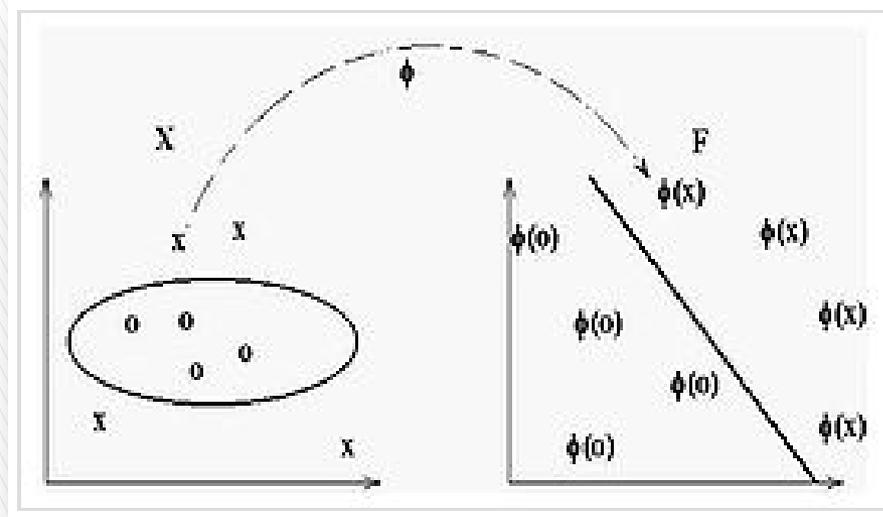
# Cascadas Adaboost + Haar Features

- ▶ Características Haar.



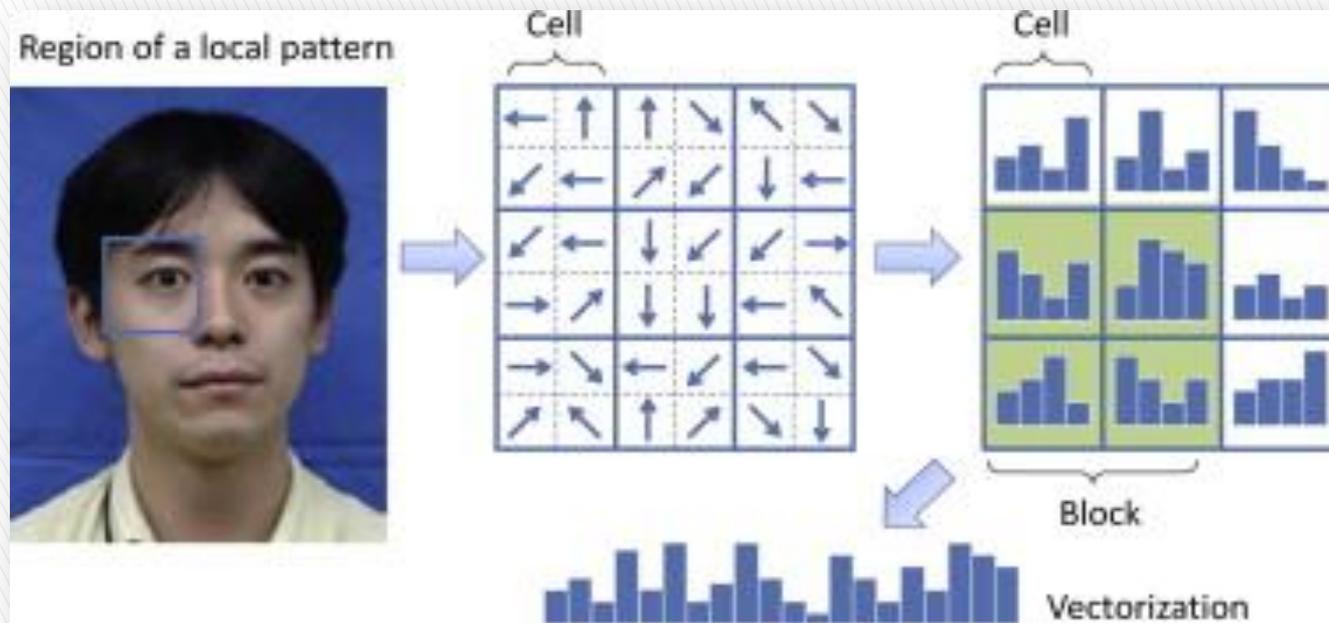
# Clasificadores SVM + HOG Descriptor

- ▶ Utilizadas ampliamente en Visión Artificial.
- ▶ Configuración de parámetros  $c$  y  $\gamma$ .
- ▶ Uso del descriptor HOG.



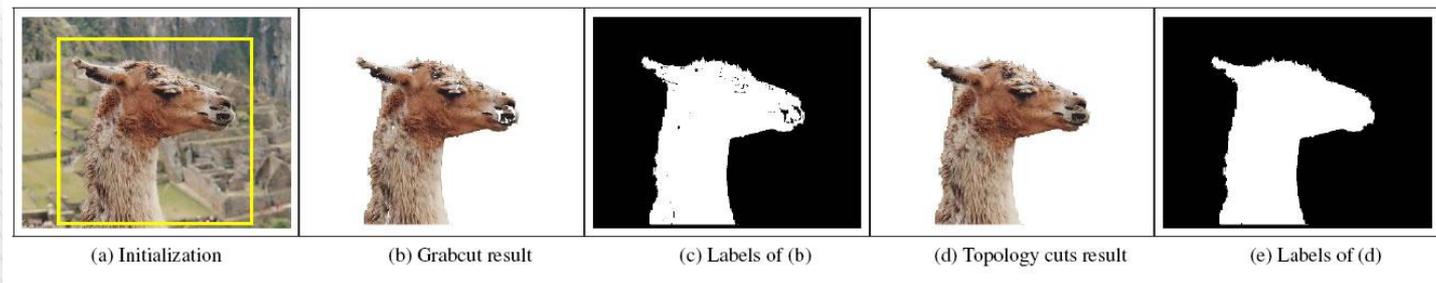
# Clasificadores SVM + HOG Descriptor

- ▶ Descriptor HOG.

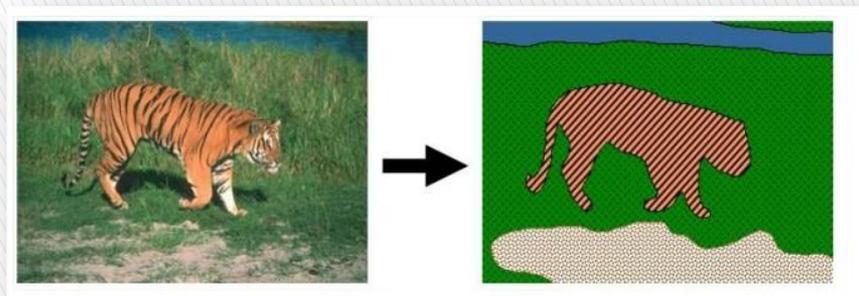


# GrabCut y GraphCut

- ▶ Sistemas de **segmentación**.
  - GrabCut: Segmentación **binaria**.

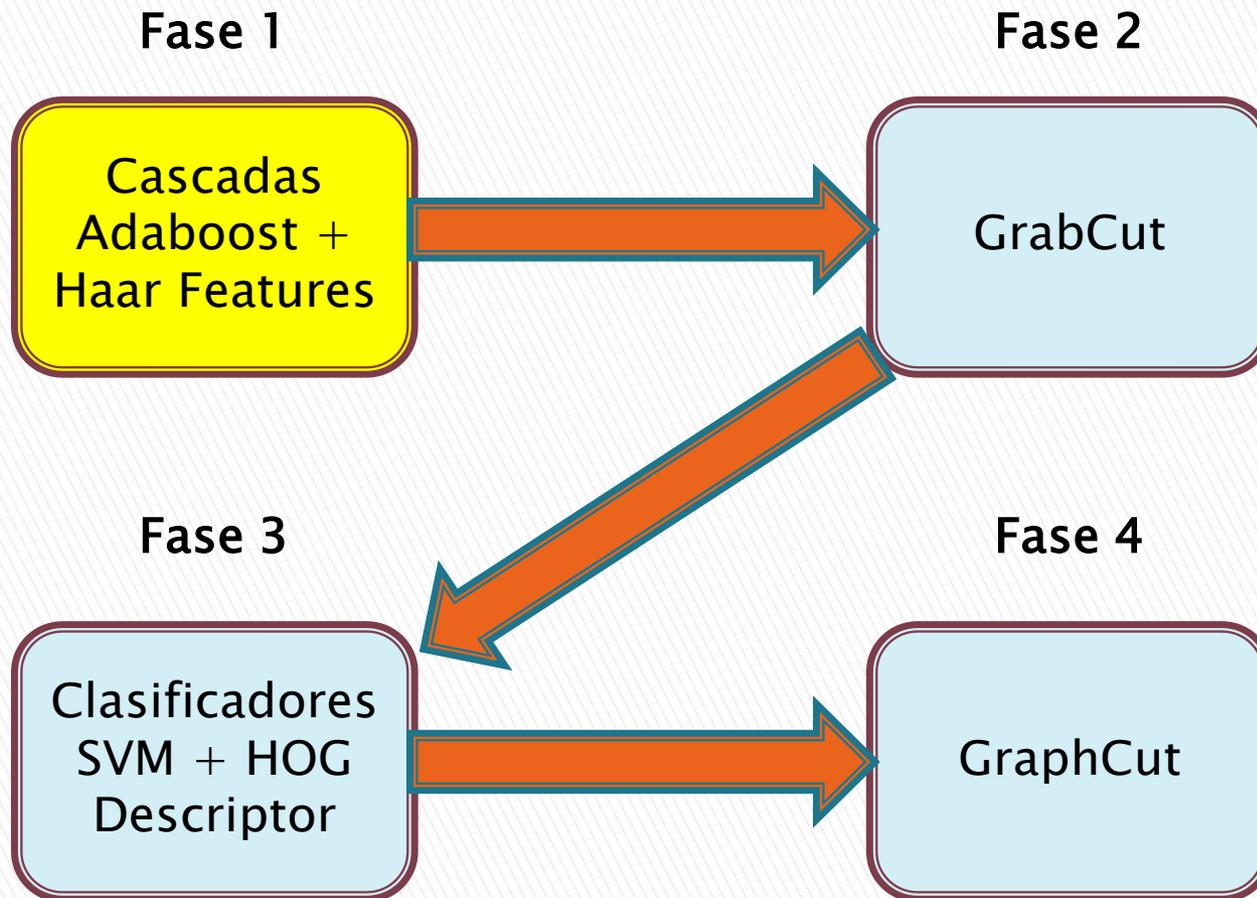


- GraphCut: Segmentación **multi-label**.



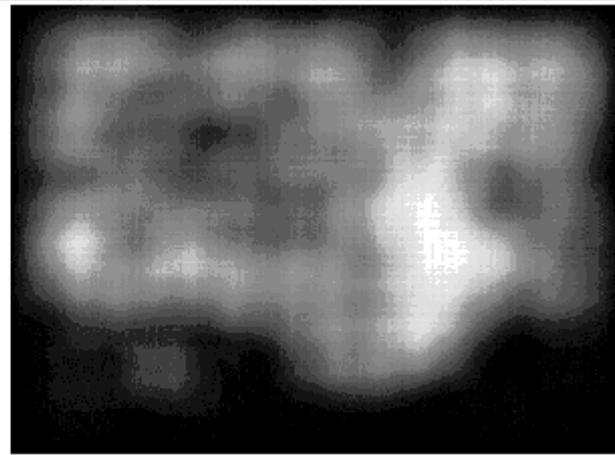
# Funcionamiento del proceso

# Funcionamiento del proceso

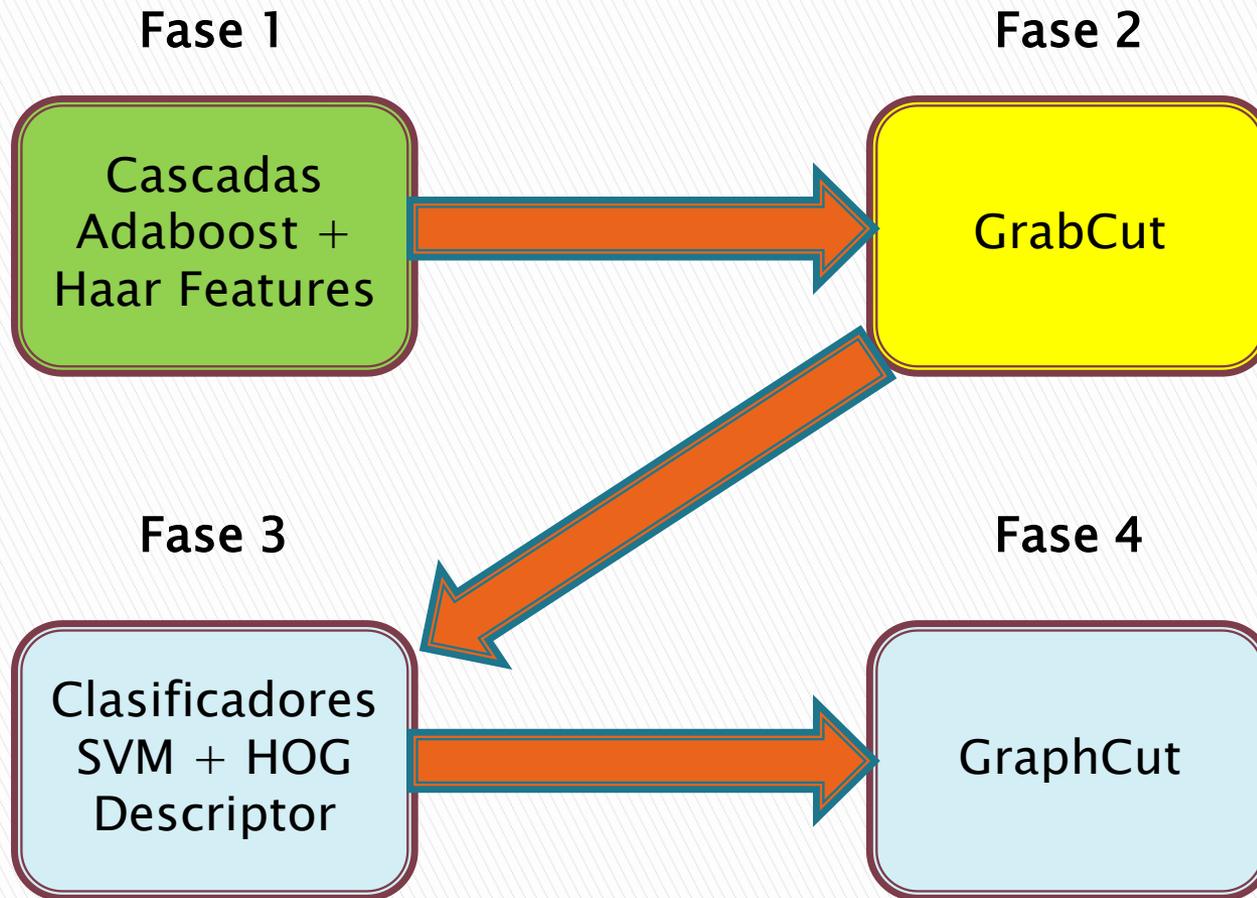


# Fase 1: Cascadas Adaboost y Haar

- ▶ **Objetivo:** Detectar a la persona en la imagen.
  - Realizar **sliding window** sobre la imagen.
  - Obtener un mapa de probabilidad de las extremidades.
  - Aplicar una etapa de **ECOC** para aumentar la robustez.



# Funcionamiento del proceso

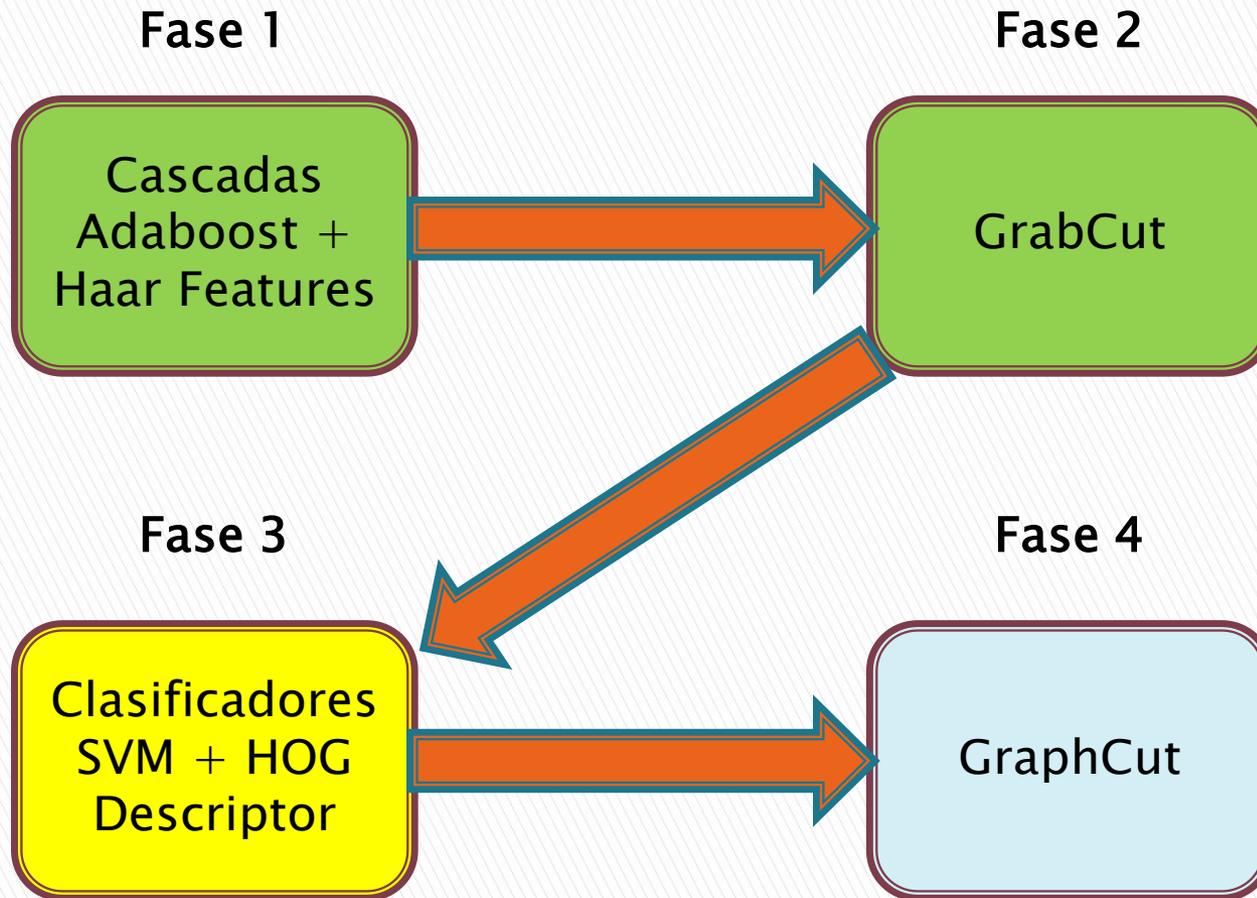


# Fase 2: Segmentación GrabCut

- ▶ **Objetivo:** Eliminar el fondo de la imagen.
  - Tomar como entrada la salida de las cascadas.
  - Configuración de los **parámetros óptimos**.
  - Obtener la imagen de segmentación sin fondo.

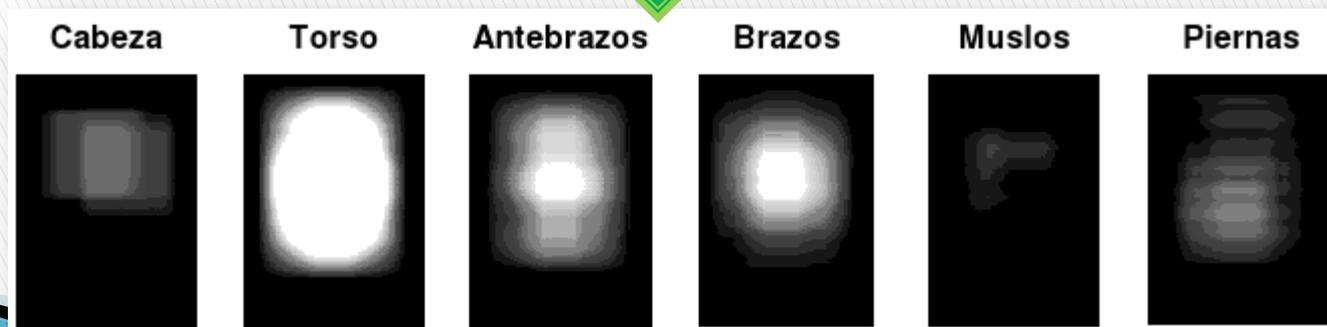


# Funcionamiento del proceso

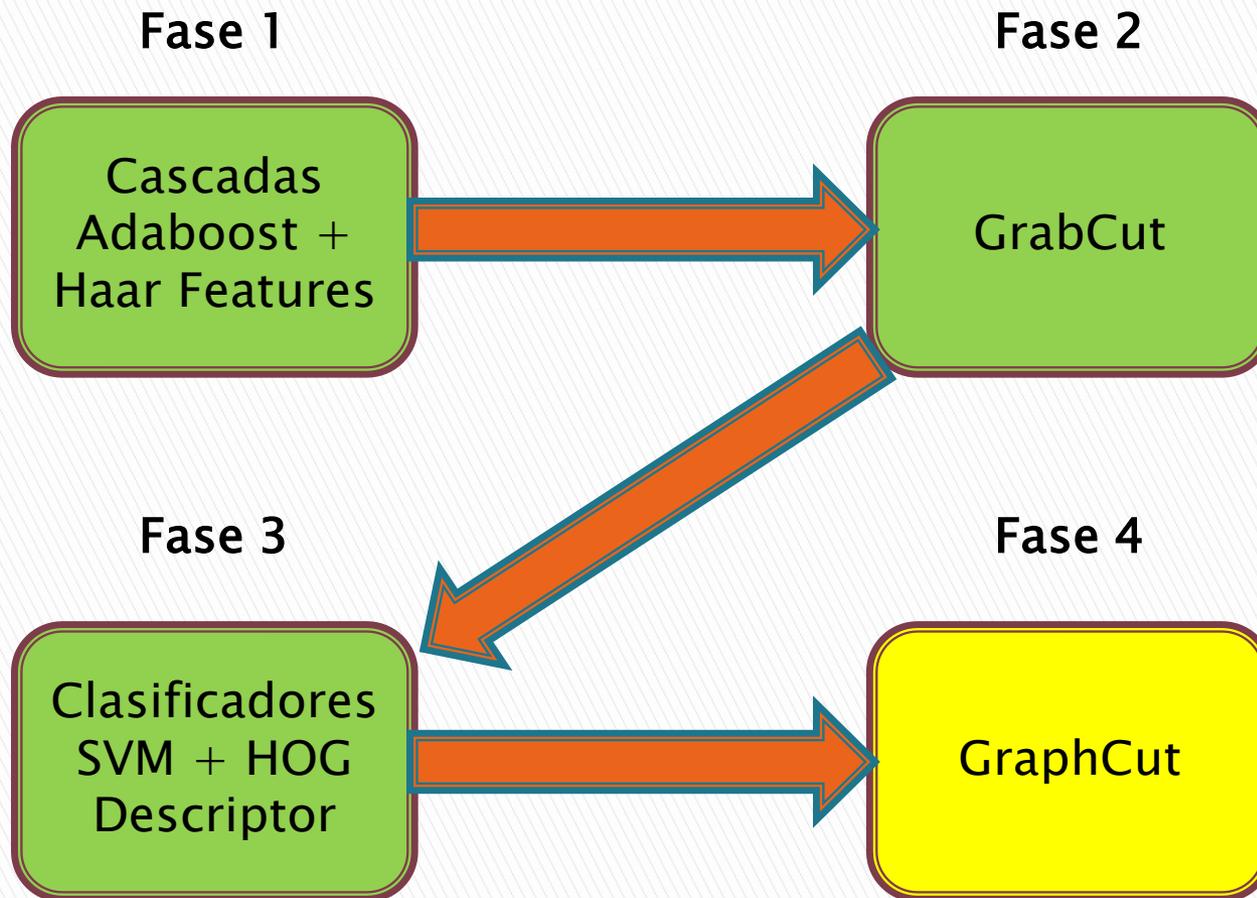


# Fase 3: SVM + HOG Descriptor

- ▶ **Objetivo:** Detectar las extremidades en la imagen.
  - Realizar **sliding window** sobre la imagen.
  - Obtener detecciones por extremidades individualmente.

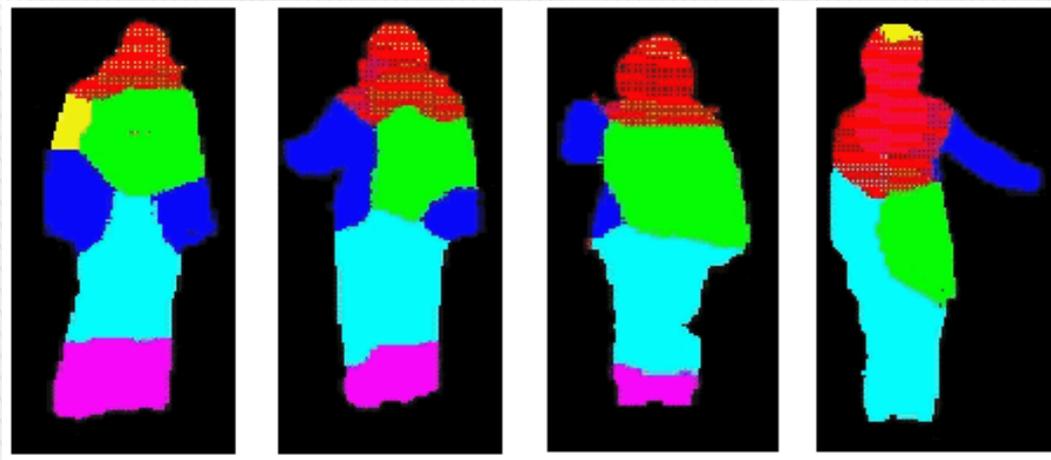


# Funcionamiento del proceso



# Fase 4: Segmentación GraphCut

- ▶ **Objetivo:** Segmentar las extremidades.
  - Tomar como entrada la salida de los SVM + ECOC.
  - Configuración de los **parámetros óptimos**.
  - Obtener la imagen de las **extremidades segmentadas**.



# Parámetros y Resultados

# Parámetros y Resultados

- ▶ Extracción de Extremidades.
- ▶ Configuración de las **Cascadas Adaboost**.
- ▶ Configuración del **Descriptor HOG**.
- ▶ Configuración de los **Clasificadores SVM**.

# Extracción de Extremidades

# Extracción de Extremidades

- ▶ **Orientación Predominante:** Conversión de Matlab a C/C++.
  - Normalizar las extremidades.
  - Código inicialmente en **Matlab**.
  - Serio problema de cara al procesamiento en **tiempo real**.

<u>Lenquaje</u>	Tiempo de Ejecución por Extremidad
Matlab	2.5 Segundos
C/C++	0.003 Segundos

# Configuración de las Cascadas Adaboost

# Configuración de las Cascadas Adaboost

- ▶ 3 pruebas (5000 VS 5000):
  - Extremidad VS Fondo (n=8).
  - Extremidad VS Fondo y resto de extremidades (n=8).
  - Extremidad VS Fondo y resto de extremidades (n=5).



VS



# Configuración de las Cascadas Adaboost

- ▶ Número de clasificadores = 6
- ▶ Número de etapas = 8
- ▶ Tamaño de la imagen = 32x32
- ▶ Min hit rate = 0.97
  
- ▶ Utilización de las utilidades:
  - `Opencv_createsamples` (conversión)
  - `Opencv_haartraining` (entrenamiento) -> XML
  - **OpenCV Face Detection API** (testeo)
    - `cvHaarClassifierCascade`
    - `cvHaarDetectObjects()`



# Configuración de las Cascadas Adaboost

- ▶ Tiempo de entrenamiento:

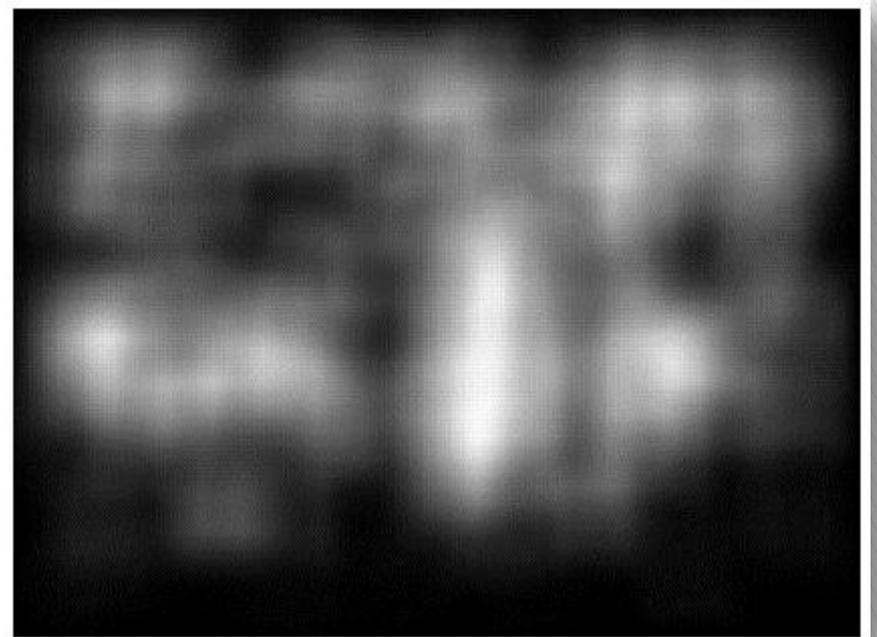
Cabeza	Torso	Antebrazo	Brazo	Pierna	Muslo
554min 38s	813min 29s	2802min 5s	2653min 4s	2252min 14s	2658min 31s

- ▶ Pesos Loss-Weight aplicados a ECOC:

Cabeza	Torso	Antebrazo	Brazo	Pierna	Muslo
0.53420	0.40452	0.24554	0.23288	0.27404	0.30747

$$\frac{\prod_{i=1}^N P_i + \prod_{i=1}^N (1 - P_i)}{2}$$

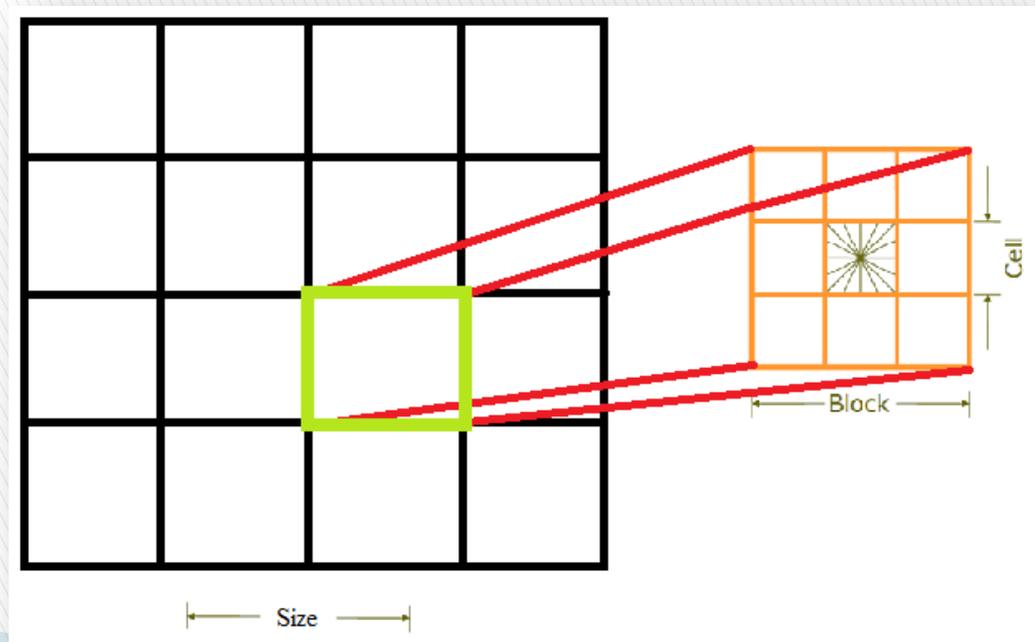
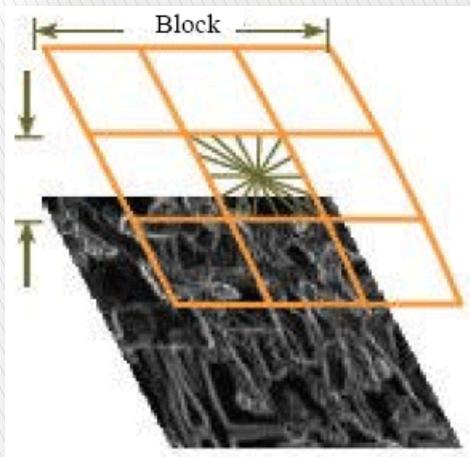
# Configuración de las Cascadas Adaboost



# Configuración del Descriptor HOG

# Configuración del Descriptor HOG

- ▶ Win Size =  $64 \times 64$
- ▶ Block Size =  $16 \times 16$
- ▶ Cell Size =  $8 \times 8$
- ▶ Longitud descriptor = 1568 elementos.



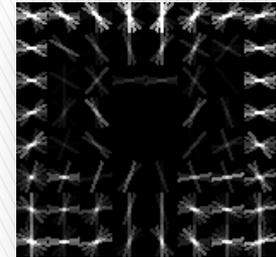
# Configuración del Descriptor HOG



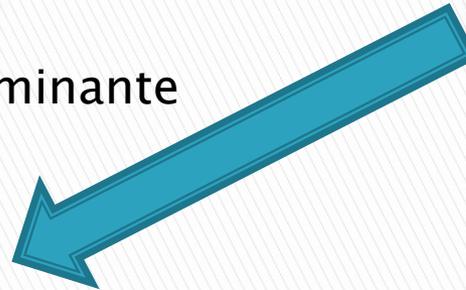
Original



Orientación Predominante



HOG



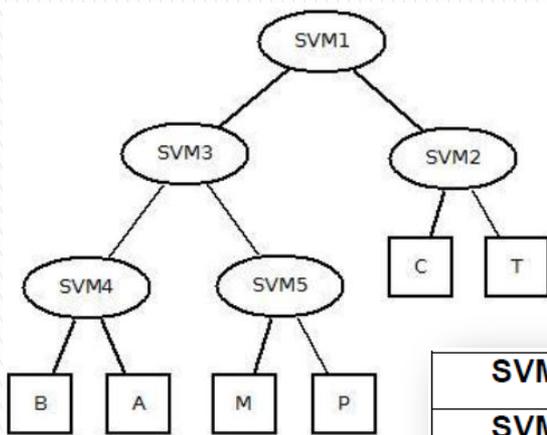
```
1 1:0.00699915 2:0.0113572 3:0.0124789 4:0.047543 5:0.047543 6:0.0109619 7:0.00304807 8:0.000828697 9:0.004070
1 1:0.021884 2:0.024313 3:0.0229773 4:0.029428 5:0.0335327 6:0.0335327 7:0.0284285 8:0.0173311 9:0.0177705 10:
1 1:0.00061855 2:0.00273009 3:0.00484905 4:0.0492934 5:0.0492934 6:0.0112297 7:0.00273988 8:0.000846404 9:0.00
1 1:0.00286839 2:0.0102004 3:0.0123459 4:0.0106457 5:0.0187393 6:0.0220552 7:0.00980319 8:0.00900703 9:0.00282
1 1:0.00422965 2:0.0081084 3:0.0084182 4:0.0433181 5:0.0433181 6:0.0189513 7:0.00460704 8:0.00202372 9:0.00721
1 1:0.00772063 2:0.00936664 3:0.0100734 4:0.0168224 5:0.00761974 6:0.00473248 7:0.00397521 8:0.00796033 9:0.01
1 1:0.00410367 2:0.0190545 3:0.0237341 4:0.0360496 5:0.0237466 6:0.00852922 7:0.00515365 8:0.00181645 9:0.0340
1 1:0.0132368 2:0.011144 3:0.0116069 4:0.0286116 5:0.0343334 6:0.0123561 7:0.00639371 8:0.00500002 9:0.0022769
1 1:0.00216756 2:0.00142074 3:0.0105471 4:0.0400286 5:0.0400286 6:0.0132688 7:0.0025817 8:0.000988751 9:0.0055
1 1:0.000365493 2:0.00241161 3:0.0167684 4:0.0475485 5:0.0475485 6:0.0112772 7:0.00308055 8:0.00105362 9:0.000
1 1:0.00203498 2:0.00851114 3:0.0265862 4:0.0358478 5:0.0358478 6:0.00908069 7:0.00482966 8:0.00418005 9:0.005
1 1:0.0136301 2:0.00864386 3:0.00655873 4:0.0207573 5:0.0207573 6:0.016330 7:0.0002782 8:0.011087 9:0.00850624
```

Descriptor

# Configuración de los Clasificadores SVM

# Configuración de los Clasificadores SVM

- ▶ Número de clasificadores = 5
- ▶ Ejemplos por clase: 1000
- ▶ Configuración ECOC Tree.



<b>SVM1</b>	Torso, Cabeza <b>VS</b> Antebrazo, Brazo, Muslo, Pierna
<b>SVM2</b>	Torso <b>VS</b> Cabeza
<b>SVM3</b>	Antebrazo, Brazo <b>VS</b> Muslo, Pierna
<b>SVM4</b>	Antebrazo <b>VS</b> Brazo
<b>SVM5</b>	Muslo <b>VS</b> Pierna

# Configuración de los Clasificadores SVM

## ▶ Matriz ECOC:

	SVM1	SVM2	SVM3	SVM4	SVM5
Torso	1	1	0	0	0
Cabeza	1	-1	0	0	0
Antebrazo	-1	0	1	1	0
Brazo	-1	0	1	-1	0
Muslo	-1	0	-1	0	1
Pierna	-1	0	-1	0	-1

## ▶ Modificación de la librería **LibSVM**:

- Adaptarla a predicción en tiempo real.
- Uso de múltiples modelos en una única instancia.

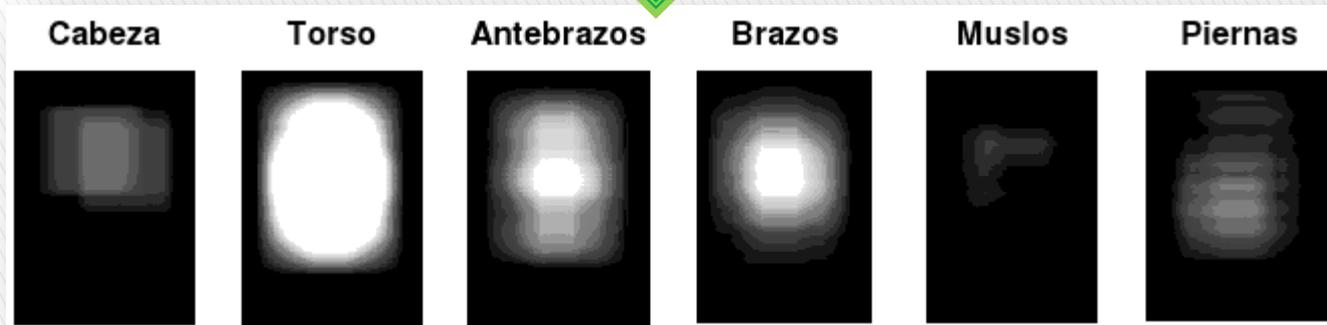
# Configuración de los Clasificadores SVM

- ▶ Utilidades:
  - Svm-train
  - Svm-predict (cambios para usar múltiples modelos)
- ▶ Parámetros óptimos de **c** y **gamma**:

Clasificador	Valor de C	Valor de Gamma	Probabilidad de Acierto
SVM1	10	0.95	97.7 %
SVM2	10	0.95	99.2 %
SVM3	10	0.95	93.5 %
SVM4	10	0.95	92.7 %
SVM5	1000	0.95	95.9 %

$$\text{PROB} = \frac{\text{N}^\circ \text{ ejemplos acertados}}{\text{N}^\circ \text{ ejemplos totales}}$$

# Configuración de los Clasificadores SVM



# Conclusiones

# Conclusiones

1. Procesamiento de la BD HuPBA.
  2. Utilización de múltiples clasificadores.
  3. Inclusión de técnicas de segmentación.
  4. Desarrollo del software necesario sobre OpenCV.
  5. Modificación de la utilidad LibSVM.
- ▶ Área que requiere mucha investigación.
  - ▶ Es un proceso complejo y costoso en tiempo.
  - ▶ Dificultad para aprender del mundo real.
    - En nuestro caso hemos utilizado 9 personas.
    - El mundo real está compuesto de muchísimas combinaciones.

# Conclusiones

- ▶ Mejoras:
  - Utilizar una configuración distinta de ECOC en SVM.
    - Ejemplo: Utilizar estrategia 1-vs-1.
  - Aplicar una fase de aprendizaje contextual después de la etapa SVM (Stacked Sequential Learning).
    - **Objetivo:** Aprender las predicciones erróneas.
  - Utilizar descriptores distintos como SURF, SIFT.
  - Obtener un conjunto de datos HuPBA mucho más grande, con más actores.

# PREGUNTAS