

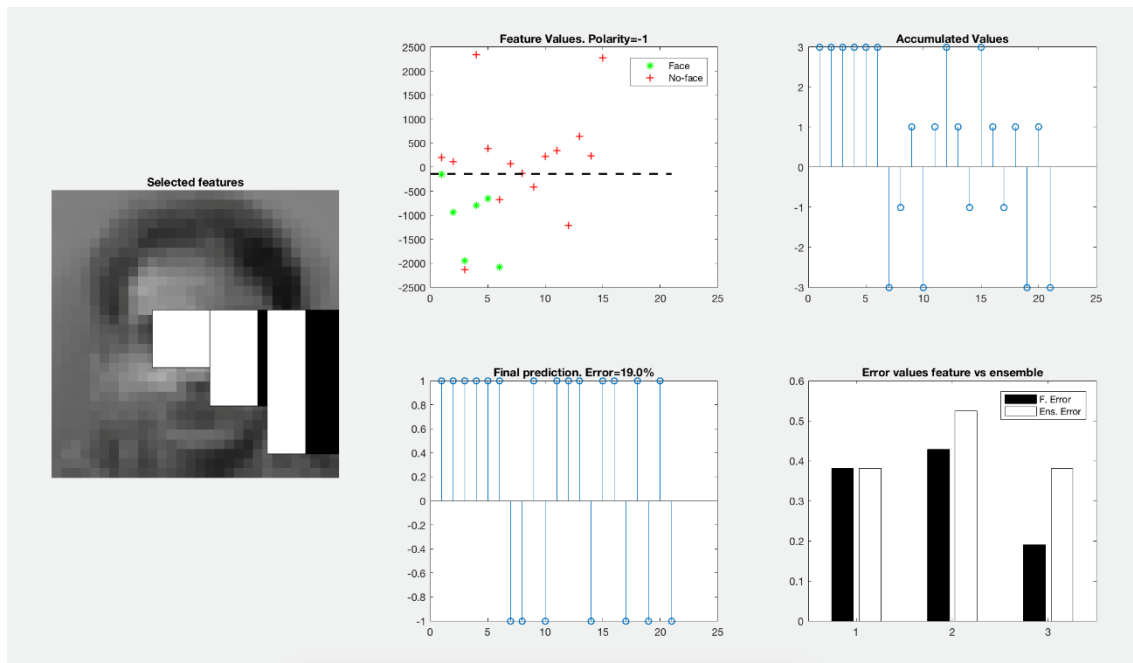
PRÁCTICA 5

DETECCION Y RECONOCIMIENTO DE CARAS

MARC FERRER MARGARIT/BOGDAN MARCUT

5.1. Características de Haar y clasificación

- a) Selecciona tres características y explica los resultados que aparecen en las distintas gráficas de la figura (adjunta la figura en el documento).



La figura de la izquierda representa la media de las imágenes que estamos usando para trabajar y las tres características de Haar que hemos seleccionado para el ejemplo. La figura “Feature Values” muestra el valor de las características calculado para cada imagen. La de arriba a la derecha representa para cada imagen la suma de votaciones de que es una cara, +1, y no es una cara, -1. Este valor es genera una vez cuando aplicamos las características.

Abajo a la izquierda observamos el error en el reconocimiento de cara. Y finalmente la de debajo de la derecha muestra el error de una cierta característica individual o considerando todas juntas.

- b) A partir de los resultados obtenidos para distintas características, indicar si hay algunos criterios que puedan ser utilizados para seleccionar una buena característica.

Un buen criterio para seleccionar una buena característica podría ser escoger características grandes y cercanas a los bordes, donde podemos observar un gran cambio de contraste en la imagen.

c) ¿Qué criterio se utiliza para seleccionar el valor de corte?

En este caso depende en la polaridad, por ejemplo, para una polaridad de +1 escogemos el mínimo valor de la región blanca de la característica y el máximo valor de la región negra y calculamos la media.

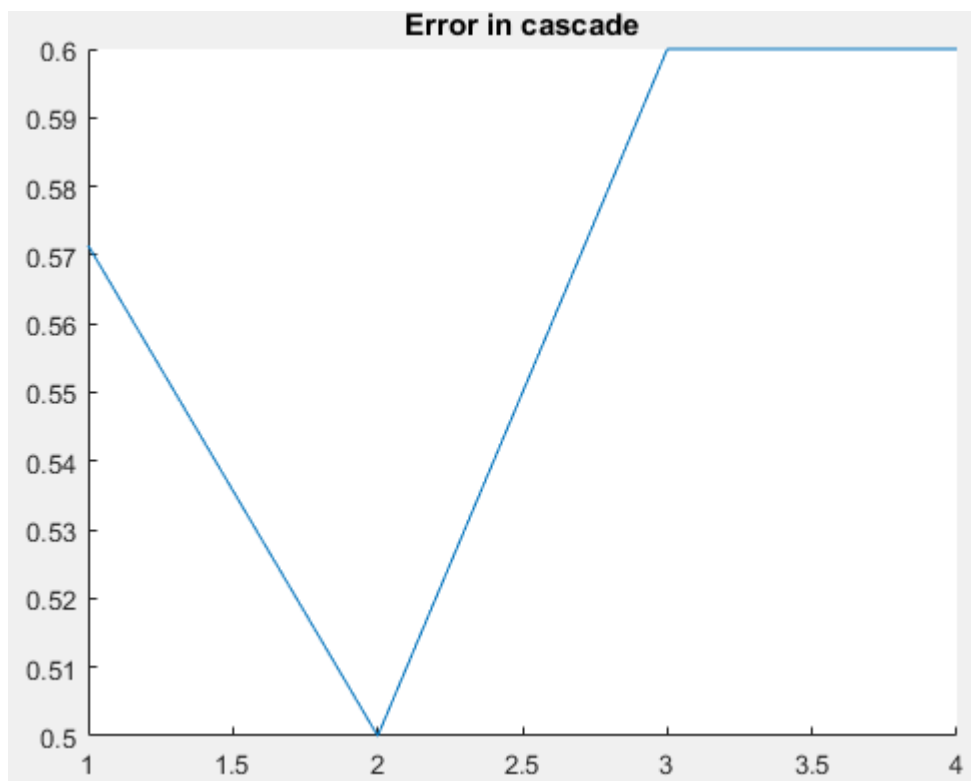
d) Combinar características no siempre mejora el resultado. ¿A qué se debe?

Se debe a que si encontramos una característica muy buena y le añadimos una no tan buena podemos provocar una introducción de ruido en el clasificador.

5.2. Cascadas de clasificación

a) Crea una función ej52 que, utilizando el método haarFeatureDemo del apartado anterior, entrene una cascada de tres niveles y dos detectores débiles por nivel.

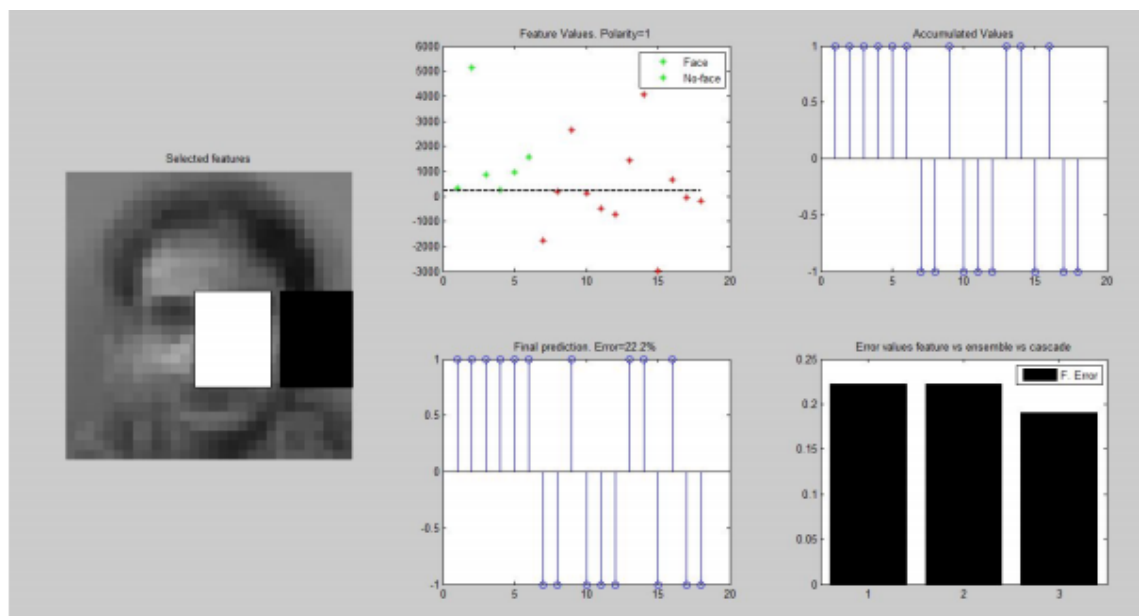
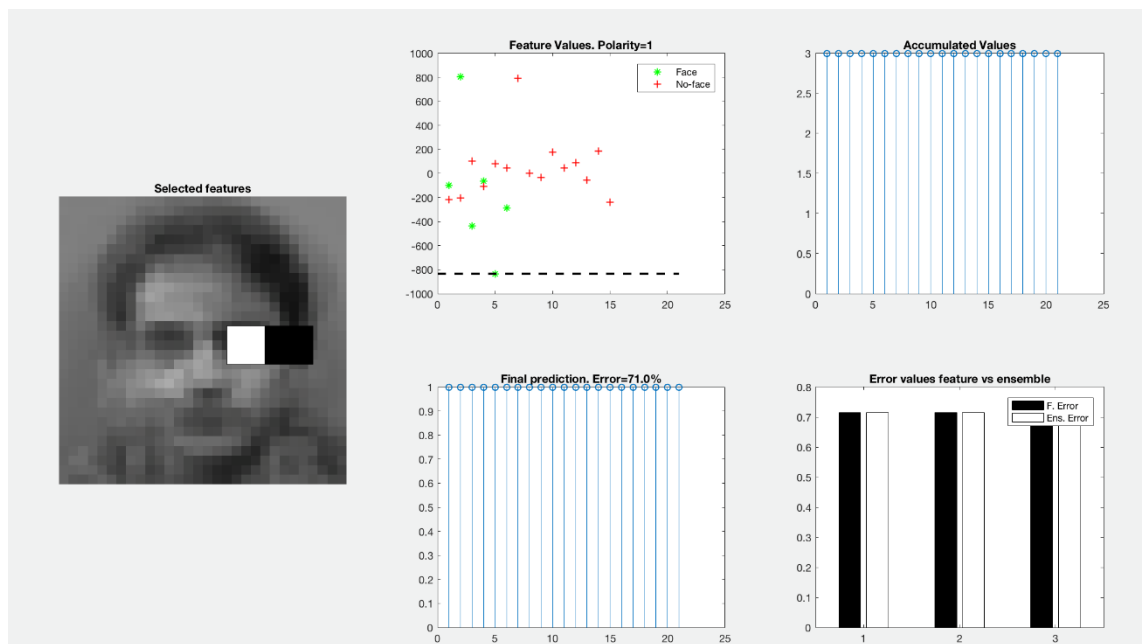
Hemos creado una función train que tiene como parámetros el número de niveles y el de características.



b) Repite el experimento anterior con dos configuraciones: a. Un nivel de cascada con 3 detectores débiles. b. Tres niveles de cascada con un detector débil en

cada nivel. Intenta seleccionar las mismas 3 características en los dos experimentos. ¿El valor del error es el mismo en ambos casos? Aunque las características sean las mismas, ¿Los detectores débiles son los mismos? Comenta los resultados obtenidos.

Podemos observar un mejor resultado con los 3 niveles de cascada. Por otra parte, en los niveles más altos verificamos los resultados de no-caras sólo por eso podemos suponer que en general los clasificadores débiles no son los mismos para la dimensión.



5.3. Detección de caras

- a) Dentro de esta carpeta veréis que hay otra carpeta HaarCascades, que contiene un fichero XML con la cascada de detección ya aprendida. Si abres el fichero verás que contiene los detectores débiles para distintos niveles, y que el tamaño de referencia es 20x20. Utiliza el método `getDataBase` para crear una imagen promedio de las caras a tamaño 20x20, tal como se hace en el método `haarFeatureDemo`, y dibuja sobre esta imagen la característica representada por estos parámetros utilizando el comando `rectangle` (tenéis un ejemplo de uso en el mismo fichero `haarFeatureDemo`). ¿Por qué un rectángulo se define con 5 parámetros en este caso? ¿Tiene sentido esta característica?

El parámetro 5 nos indica si el rectángulo será considerado como un inhibidor de zona si vale negativo si es positivo nos dará el activador de zona para esa característica. Tiene sentido porque nos da una buena calidad de información sobre el contraste de la zona de la frente y los ojos.

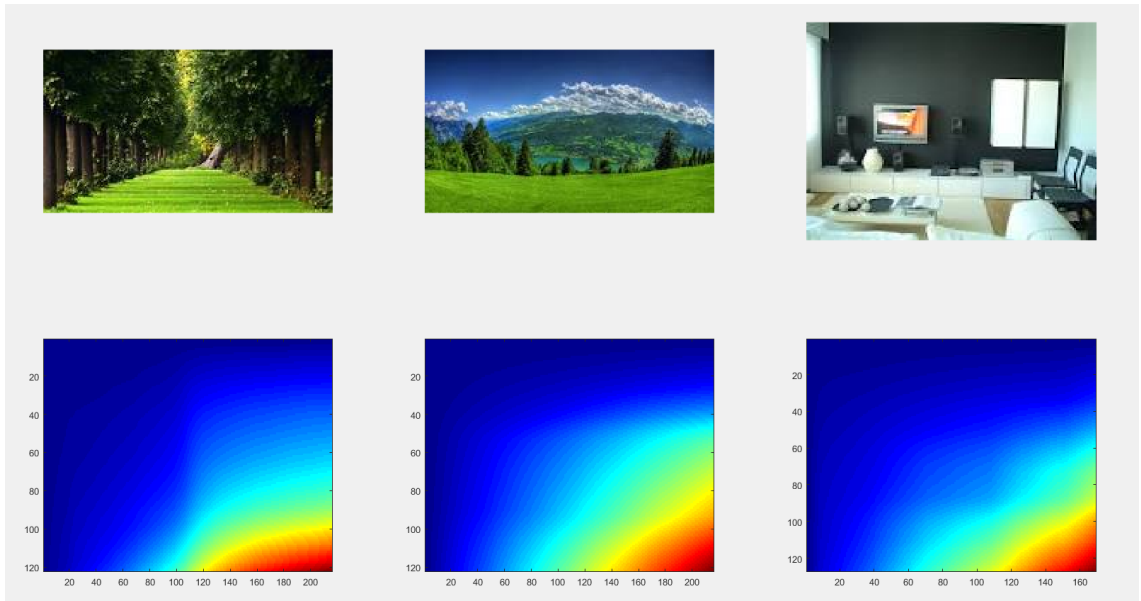


- b) Para optimizar el cómputo de las características de Haar se utiliza una representación de la imagen conocida como imagen integral. Utiliza el siguiente código para visualizar una imagen integral:

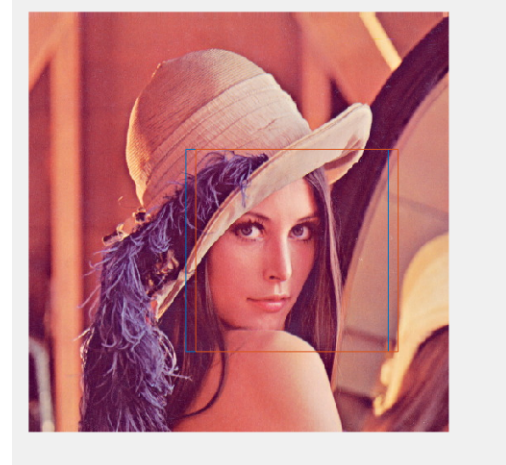
```
function showIntegralImage(image)  
    defaultoptions.Resize=false;  
    intImageStruct=GetIntegralImages(image,defaultoptions);  
    imagesc(intImageStruct.ii);colormap(jet);  
end
```

Muestra la imagen integral para dos o tres imágenes distintas (ver Figura 6) y explica qué tienen en común. Justifica tu respuesta. Calcula la característica Haar del punto a) a partir de la imagen integral.

Aquí podemos apreciar con los mapas de calor resultantes que el valor para la integral cerca del (0,0) es muy baja. Cuando el valor se aproxima al tamaño de la imagen, la integral tiene un valor más grande.



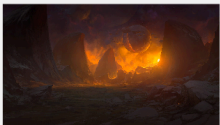
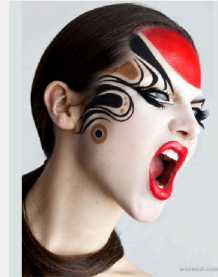
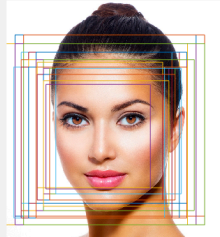
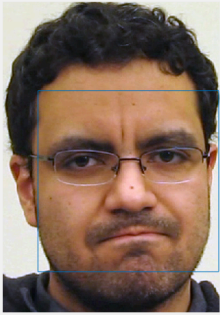
c) Utiliza los ejemplos 1 y 2 que hay en la documentación del método ObjectDetection en Matlab (incluido en la carpeta ViolaJones facilitada con el enunciado) para probar el detector con las imágenes 'testFaces1.jpg' y 'testFaces2.jpg'.



d) Baja de internet 5 imágenes con caras y 5 sin caras y aplica el método.

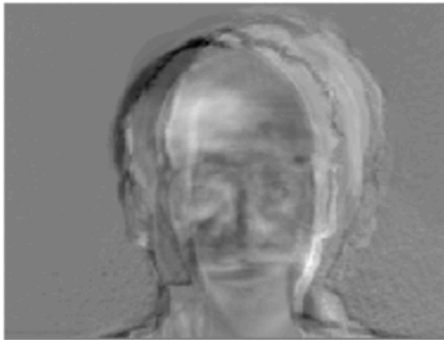
Podemos observar que detectamos 3 caras de las 5 que hay esto puede ser causado por la imagen o como está hecha la fotografía

También vemos como en las fotos sin caras en una nos la detecta, puede ser debido a que tiene forma circular y el programa lo detecta como una cara.

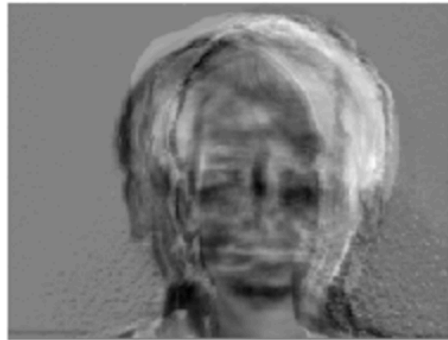


5.4. Reconocimiento de caras con el método Eigenfaces

Eigenface 1



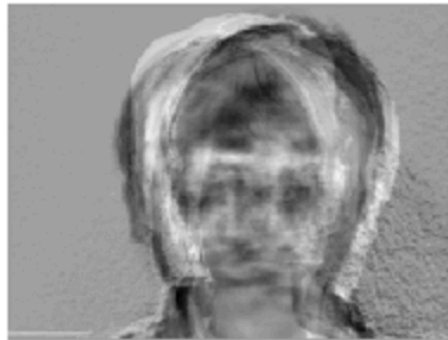
Eigenface 2



Eigenface 3



Eigenface 4



Test image



Similar face 1



Similar face 2



Similar face 3



1. ¿Qué sucede si aplicas el reconocimiento sobre el mismo conjunto training_data?

Da un error en el tamaño de la matriz, directamente no deja ejecutar el programa.

2. ¿Qué sucede si rotas las imágenes de las caras de test?

La imagen sale distorsionada, no se ve la imagen, solo se ven líneas y la imagen no se ve