

Práctica 1

# **Detección automática de equipos de futbol en imagenes**

Visión por Computador

2019 – 2020 Q2

Ferran Velasco Olivera  
Joaquín Gómez Sánchez

## Índice

1.	Algoritmo para la detección del Barça en imágenes	2
2.	Descripción de los experimentos	4
2.1.	Para obtener el mejor threshold para el algoritmo	
2.2.	Para obtener la mejor parametrización del algoritmo	
3.	Resultados obtenidos	8
4.	Funciones empleadas	14
5.	Conclusiones	17

## Algoritmo para la detección del Barça en imágenes

El algoritmo implementado para la detección de imágenes del Barça mediante histogramas, que ha sido parametrizado a partir de los resultados de los experimentos descritos en el siguiente apartado, se encuentra en el fichero `Algorithm.m`.

El modelo de color que mejor resultados ofrece, y el que utiliza el algoritmo, es HSV. Todos los histogramas (calculados en `histogramHSV`), tanto de los modelos como de la imagen a decidir, sufren una reducción de dimensión eliminando la componente V. Además de una normalización, que permite una correcta comparación de estos.

Para realizar la comparación, el tamaño de los histogramas es de 128x128 bins. Este tamaño de bins permite tanto conservar suficientes detalles, como hacer una buena discriminación (barça o no barça) en las imágenes a decidir.

Para decidir si una imagen dada contiene barça o no, se comprueba si la distancia media ponderada con tres imágenes modelo es inferior a un threshold  $T$ . Queda entonces que una imagen se considerará que contiene barça si la siguiente expresión se evalúa a cierta:

$$\frac{\sum_i \alpha_i \text{dist}(\text{modelo}_i, \text{hist}_i)}{\sum_i \alpha_i} < T$$

donde, para el algoritmo, la función de distancia es Chi-Square (`compareChiDistance`), el threshold  $T$  es 0.5691 y las ponderaciones  $\alpha_i$  son 0.2, 0.4, 0.6 respectivamente para las siguientes imagen modelo:



Imagen Modelo 1  
(Recorte de 117x148px  
de la imagen 39)



Imagen Modelo 2  
(Recorte de 204x146px  
de la imagen 42)



Imagen Modelo 3  
(Recorte de 304x199px  
de la imagen 43)

Para las imágenes modelo se ha decidido utilizar los tres recortes anteriores de únicamente colores del Barça, puesto que son esos los que van a causar una diferenciación sustancial en las distancias de las imágenes que los presentan y las que no.

Se ha optado por incluir un recorte del dorsal, con poco peso en la ponderación, puesto que siempre es amarillo, y aunque no es un color representativo, es importante tenerlo en cuenta para las fotografías de jugadores de espalda.

Las otras dos imágenes, se ha seleccionado una de mayor peso con poca parte visible del cuerpo de los jugadores, y una con peso intermedio con mayor visible de los jugadores.

## Descripción de los experimentos

### Para obtener el mejor threshold para el algoritmo

En este apartado se explica el procedimiento seguido para calcular un threshold óptimo dado un grupo de imágenes del equipo que se quiere reconocer (Grupo 1) y otro grupo con imágenes de equipos diferentes al que queremos reconocer (Grupo 2). Para que la explicación sea más clara, se ha optado por escoger como ejemplo las imágenes del Madrid y del Barça.

En primera instancia, una vez hecho el preprocesamiento de las imágenes y el cálculo de sus histogramas, según la parametrización dada, se calculan las distancias de las imágenes de ambos grupos a las imágenes modelo. Para observar la distribución de estas, se hace un plot de las distancias de ambos grupos, diferenciándolas por el color, para intentar discernir si existe una diferencia entre ambos grupos.

Al realizar este plot se observa cómo la distribución de las distancias se ve potencialmente diferenciada entre grupos. Dada esta observación, se decide calcular el threshold óptimo cogiendo la distancia mayor del Grupo 1 y la menor del Grupo 2, las cuales representan los extremos de ambos grupos, y realizar la media de estas. El threshold se calcula como la media de estos dos valores, ya que el umbral de diferencia entre los dos grupos normalmente se encuentra entre estos.

En la Figura 1 está el resultado obtenido, donde se muestran los valores de las distancias al modelo calculadas para las imágenes del Madrid (en rojo) y del Barça (en azul) y el correspondiente threshold calculado representado por una línea verde.

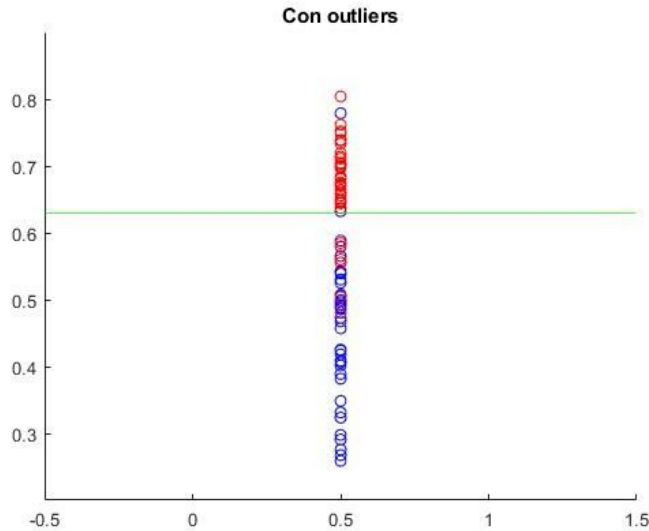


Figura 1: Primer plot de las distancias del Grupo 1 y Grupo 2 con el threshold calculado con todos los valores

No obstante, se puede observar un claro conflicto en un umbral sobre el medio, donde se encuentran puntos de grupos diferentes que se encuentran sobre la misma zona, además de claros outliers que se encuentran totalmente desplazados en relación a los de su mismo grupo.

Es por eso que, al extrapolar este cálculo de threshold a todas las imágenes de todos los equipos, se puede ver que los resultados en muchos casos no son del todo fiables. El motivo es que en muchas ocasiones, los extremos utilizados para calcular el threshold, no son miembros representativos del grupo al que pertenecen, ya que se trata de outliers.

Para tratar de solucionar este problema, se utiliza el mismo procedimiento pero en este caso sin tener en cuenta las distancias que resultan ser outliers. En concreto, “se rellena” el valor de todos aquellos valores que se encuentran fuera del intervalo entre el 10% y el 90% del total, con la llamada a la función `filloutliers(positiveResults,'linear', 'percentiles', [10 90]);` y con los valores resultantes, se vuelve a calcular el threshold cómo se ha especificado anteriormente.

En la figura 2, se puede observar que en este caso el threshold resultante es mucho más apropiado, y a su misma vez, al realizar este cálculo para el total de todas las imágenes, los modelos obtienen unos resultados claramente mucho más fiables.

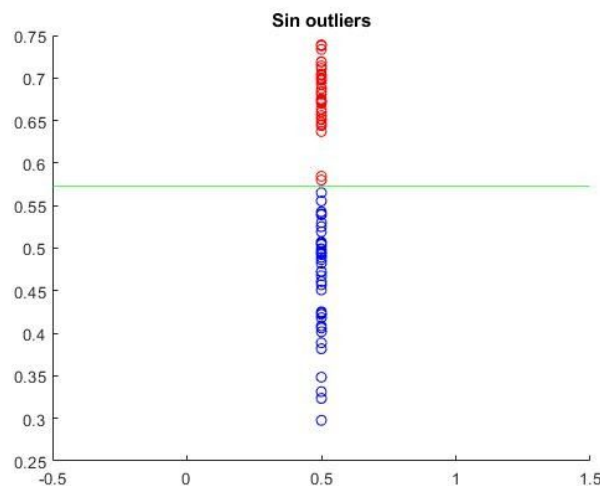


Figura 2: Plot de las distancias del Grupo 1 y Grupo 2 con el threshold calculado sin los valores Outliers

Todo este procedimiento aplicado a todo el conjunto de imágenes es el que se ha empleado para obtener el mejor threshold para la parametrización del modelo, se detalla a continuación.

### **Para obtener la mejor parametrización del algoritmo**

Durante la experimentación para obtener la mejor parametrización posible del algoritmo, se han probado diferentes modelos de color (RGB y HSV), diferentes métodos para calcular distancias entre histogramas (Chi-Square y distancia Euclidiana) y diferentes valores para los números de bins con los que realizar el histograma.

Para realizar los histogramas se ha decidido emplear posibles tamaños de 2x2, 4x4, 8x8,... hasta 128x128 bins; correspondiendo las dos dimensiones de los histogramas a las dos dimensiones utilizada en cada representación, R y B en RGB, y H y S en HSV.

Para probar la efectividad de la parametrización del algoritmo probada en el momento, se ha calculado el threshold como se ha indicado en el apartado anterior. Una vez obtenido el threshold, se ha comprobado la cantidad de imágenes que el modelo parametrizado acierta si son del Barça o no, con las distancias y el threshold calculados.

Se ha ejecutado la misma función para todas las posibles combinaciones de diferentes valores de bins, cualquier modelo de color y cálculo de la distancia. Para escoger cuál ofrece un mejor resultado, se ha obtenido la cantidad total de falsos positivos, verdaderos positivos, falsos negativos y verdaderos negativos; para calcular así la exactitud de cada conjunto de parámetros y saber cuál da un mejor resultado.

Utilizando la distancia Euclidiana se ha podido observar que se pierde exactitud cuanto más grande es el número de bins en el histograma. Con la distancia Chi-Square, en cambio, ocurre todo lo contrario, ya que la exactitud del modelo aumenta cuando también lo hace el número de bins. Esto ocurre tanto para el modelo de color RGB como HSV.

El mejor accuracy obtenida es de un 75,71% para el tamaño de histograma 128x128 (1 bin por cada 2 píxeles) con representación en HSV; y 75% para el tamaño de histograma 64x64 en el caso de RGB, ambos utilizando Chi-Square para calcular las distancias entre histogramas.

Podemos concluir entonces que la mejor para algoritmo es la expuesta en la sección anterior: modelo de color HSV, función de distancia Chi-Square, histogramas de 128x128, y el threshold calculado en este experimento para esta parametrización.



## Resultados obtenidos

En esta sección se muestran los resultados obtenidos por el algoritmo final, que emplea un tamaño de histograma de 128x128, con representación en HSV, y utilizando Chi-Square para medir las distancias. El threshold fijado en el algoritmo es de 0,5691, tal y como se ha descrito anteriormente.

		Predicción	
		Con barça	Sin barça
Real	Con barça	29	11
	Sin barça	57	183

Tabla 1 - Matriz de confusión del algoritmo con imágenes que contienen Barça

Cómo se puede observar en la Tabla 1, sobre un total de 280 imágenes, conteniendo Barça 20 imágenes, obtenemos que el ratio de positivos acertados es de un 72,5%, mientras que el ratio de negativos acertados es de un 76,25%. Se puede observar que este modelo se comporta mejor a la hora de reconocer imágenes que no son del Barça, en vez de reconocer las que sí lo son.

No obstante, si se observan imágenes del Barça que han fallado, se encuentran imágenes como las siguientes:



En estas, se puede ver que los colores “blaugrana” no se ven tan claros como en otras imágenes, y es por ello que es comprensible que sean justamente este tipo de imágenes las que fallan, dado el tipo de de comparación que estamos realizando.

Los **falsos negativos** del set de prueba adjunto a la práctica son las siguientes imágenes: 3, 7, 12, 14, 15, 22, 25, 29, 35, 37, 38.

A continuación se va a analizar cada equipo por separado, ya que como se ha observado en experimentos, unos equipos son reconocidos como “sin Barça” mucho más fácilmente que otros, debido a los colores que estos tienen.

## Madrid

En este caso, el resultado es realmente bueno, siendo el número de imágenes identificadas como Barça dos, lo que le da una precisión de 95%. Las dos imágenes identificadas como Barça:



Ambas imágenes tienen una distancia con el modelo de 0,54 y 0,52 respectivamente, es decir, están casi rozando el threshold, por lo que el resultado queda muy ajustado.

Aún así, cabe a tener en cuenta que ambas imágenes tienen presencia de colores azules y rojos (rojo en la primera, y azul en la segunda) que pueden tener una influencia considerable a la hora de calcular la distancia con los modelos.

Queda entonces que para el **Madrid** los **falsos positivos** son las imágenes: 18 y 25.

## Milan

Este equipo, al tener de colores el rojo y el negro, siendo el rojo uno oscuro muy parecido al granate, y el negro que según la iluminación toma un tono azulado, resulta un equipo más fácil de fallar para el modelo. Los resultados obtenidos son de 15 imágenes reconocidas como Barça, es decir, un 62,5% de precisión.

Entre las imágenes reconocidas como Barça encontramos:



Cómo se ha indicado, estas imágenes tienen gran parte de rojo, parecido al granate, lo que puede confundir al algoritmo, haciendo que resulte en que las imágenes contienen Barça.

Los 15 **falsos positivos** detectados corresponden a las imágenes del **Milan** son: 1, 2, 4, 6, 7, 10, 11, 18, 20, 26, 27, 32, 34, 36, 39.

## Chelsea

En el caso del Chelsea, el número de imágenes reconocidas como Barça es 7. Por lo tanto, la precisión en este caso es de un 82,5%.

Algunas imágenes reconocidas como Barça son:



Cómo ocurría con algunas imágenes del Milan, la mayoría de imágenes reconocidas como Barça tienen el azul característico del Chelsea y parte de rojo, lo que hace que sean más cercana al “blaugrana”, confundiendo al algoritmo.

Los falsos positivos obtenidos para el Chelsea son las imágenes: 6, 9, 15, 23, 29, 31, 32.

## Juventus

En el caso de la Juventus, el número de imágenes identificadas como Barça, son solamente 3, es decir, la precisión es de un 92,5%, realmente buena. Está dentro de lo que cabría esperar puesto que los colores de este equipo son el blanco y el negro.

Las 3 imágenes identificadas como Barça son las siguientes:



En este caso, vuelven a ser los colores rojos y azulados los que influyen en el hecho de que sean reconocidas como Barça. No obstante, la distancia que tienen al modelo es muy cercana al threshold, lo que indica que se encuentran en el rango más conflictivo.



Las 3 imágenes de la **Juventus** identificadas, y por lo tanto **falsos positivos**, son las siguientes: 4, 24, 29.

## Liverpool

En el caso de este equipo, el número de imágenes identificadas como Barça es 18. Claramente es un muy mal resultado, fallando casi el 50% de las veces, con una precisión solo del 55%. Esto denota que el color rojo es el que más pasa factura a la hora de calcular la distancia con el modelo.



Observando las imágenes identificadas como Barça se puede observar que vuelven a ser las grandes partes de rojo y azul las que generan el conflicto. No obstante, las distancias de la gran mayoría se encuentran igualmente muy cercanas al threshold.

Los 18 **falsos positivos** identificados en el **Liverpool** son las imágenes: 1, 5, 6, 8, 10, 14, 15, 17, 23, 25, 26, 27, 28, 29, 32, 33, 34, 37.

## PSV

Para el PSV se observan un total de 12 imágenes reconocidas como del Barça, dando así un 70% de precisión. Entre estas imágenes se encuentran:



Otra vez se observa cómo es el color rojo el que hace fallar al modelo, ya que las imágenes acaban obteniendo distancias muy parecidas a las del Barça.

Los **falsos positivos** del **PSV** son las siguientes imágenes: 2, 4, 8, 9, 14, 15, 21, 22, 35, 38, 39, 40.

### Otros equipos e imagenes del Barça

Por último, se ha probado el algoritmo en un set de imágenes de equipos variados, que se puede encontrar en el directorio `/soccer/others`.

Sobre el total de 40 imágenes se ha obtenido una respuesta satisfactoria en el 85% de ellas, dando **falsos positivos** en imágenes del Manchester United (imagen 21), Atlético de Madrid (imagen 18) y Selección Española (imagen 6).

Como se puede observar en las siguientes imágenes, estas presentan un altísimo componente rojo que avoca al algoritmo a confundirlas con imágenes que contienen Barça, hecho ya corroborado con el set de imágenes de la práctica.



Probando el algoritmo en otro set de imágenes del Barça de 20 elementos, que se encuentra en el directorio `/soccer/barcelonaothers`, se ha obtenido una tasa de acierto del 85%. Este valor es superior al obtenido con las imágenes del set de la práctica.

## Funciones empleadas

La **funciones** empleadas para el desarrollo de la presente práctica, y que han sido **desarrolladas únicamente por nosotros**, sin hacer uso de fuentes externas, son las siguientes.

Para el **preprocesamiento de imágenes, cálculo de distancias e histogramas**:

- `imagePreProcessingRGB`: Preprocesamiento de imágenes RGB, realizando una normalización del color.
- `imagePreProcessingHSV`: Preprocesamiento de imágenes HSV, para adaptarlas al cálculo de histogramas.
- `histogramRGB`: Cálculo del histograma asociado a la imagen de entrada, ignorando la componente G.
- `histogramHSV`: Cálculo del histograma asociado a la imagen de entrada, ignorando la componente V.
- `compareEuclideanDistance`: Cálculo de la distancia euclidiana entre dos histogramas, haciendo uso de la expresión:

$$d(h_1, h_2) = \sum_i (h_{1i} - h_{2i})^2$$

- `compareChiDistance`: Cálculo de la distancia Chi-Square entre dos histogramas, haciendo uso de la expresión:

$$d(h_1, h_2) = \frac{1}{2} \sum_i \frac{(h_{1i} - h_{2i})^2}{h_{1i} + h_{2i}}$$

Para la **carga de las imágenes en MATLAB**:

- `getTeamImages`: Obtención de todas las imágenes de un directorio dado o seleccionado en un explorador.
- `getAllImages`: Obtención de todas las imágenes de subdirectorios de un directorio raíz dado.

Para el **algoritmo** y la **experimentación**:

- `experiments`: Función para analizar los resultados de un grupo de imágenes dado un modelo y su `threshold`, la función de preprocesamiento, cálculo de histogramas y cálculo de distancia, y el número de bins.
- `globalExperiment`: Función para dadas las imágenes de un equipo que se quiere reconocer y imágenes que no son del equipo, un modelo, la función de preprocesamiento, cálculo de histogramas y cálculo de distancia, y el número de bins, se calcule el `threshold` óptimo y los resultados dado ese `threshold`.
- `calculateThresholdWithOutliers`: Cálculo del `threshold` óptimo dado un conjunto de distancias positivas y un conjunto de distancias negativas relativas a un modelo.
- `calculateThresholdWithoutOutliers`: Cálculo del `threshold` óptimo dado un conjunto de distancias positivas y un conjunto de distancias negativas relativas a un modelo, ignorando los outliers de estas distancias.
- `preprocessImages`: Obtención de un conjunto de imágenes preprocesadas, dado el conjunto de imágenes y la función de preprocesamiento.
- `getHistograms`: Obtención de todos los histogramas dadas las imágenes, una función de cálculo de histograma (RGB o HSV) y el número de bins.
- `calculateHistogramsDistance`: Obtención de todas las distancias a un modelo dado de un grupo de histogramas dados.
- `calculateDistance`: Cálculo de la distancia de un histograma a un modelo dado.
- `isTeam`: Función para saber si una imagen es de un equipo dado un modelo, la imagen, la función de cálculo del histograma y el `threshold` a considerar para saber si es del equipo del modelo.

El resto de funciones empleadas son **funciones MATLAB**, es decir, no se ha realizado ningún uso de software o funciones diferentes de las proporcionadas en el set estándar de toolboxes de MATLAB.



Para el **trabajo con estructuras de datos** se han empleado las funciones: `cell`, para la creación de cell arrays; `cell2mat`, que permite la conversión cell array a matriz; `cellfun`, que permite la aplicación de una función sobre todas las celdas de un cell array; `containers.Map`, para la creación de mapas/diccionarios; `zeros`, que permite crear matrices de ceros; `arrayfun`, que permite aplicar una función a todos los elementos de un array; y `size`, para obtener el tamaño de las estructuras.

Para la **gestión de cadenas de caracteres** se ha empleado: `strcmp`, que realiza comparaciones de strings; `strcat`, que devuelve los strings de entrada concatenados; y `num2str`, que convierte un formato numérico en cadena de caracteres.

Para el **trabajo con imágenes** se ha hecho uso de: `imread`, para leer imágenes; `rgb2hsv`, para realizar la conversión del modelo de color RGB a HSV.

**Otras funciones** empleadas son: `filloutliers`, para la detección y reemplazo de outliers; `sqrt`, para realizar raíces cuadradas; `uigetdir`, para abrir el explorador de ficheros; `dir`, para listar el contenido de un directorio; y `fullfile`, para construir el nombre de un fichero a partir de partes.

## Conclusiones

Tras observar los resultados obtenidos, desgranados equipo por equipo, se puede concluir que la diferenciación de equipos teniendo en cuenta los colores azul y granate puede no ser del todo precisa. O al menos no de forma tan precisa que siempre garantice ratios de accuracy superiores al 80-90%, empleando comparación de histogramas.

Muchas veces estos colores se pueden encontrar en equipos que no son el que se está buscando y esto hace que se genere una mala predicción.

En concreto, se concluye que este algoritmo desarrollado es mucho más sensible al color rojo o granate, que al azul, viendo el claro contraste entre el buen resultado del Chelsea y el no tan buen resultado del Liverpool.

Aun así cabe destacar las altas tasas de acierto obtenidas con el algoritmo, que aunque no sea especialmente bueno con las imágenes con un alto componente en rojo, si es lo es para el resto de posibles colores, incluyendo el azul.