

VISIÓ PER COMPUTADOR

Projecte Final

Reconeixement d'equips de futbol en imatges



Daniel Escribano Pérez

Martí Serra Aguilera

Sufang Zhou

Index

| | |
|------------------------------------------------------------------------------------|-----------|
| Introducció | 4 |
| Etapa 1 | 5 |
| Explicació de l'algorisme utilitzat | 5 |
| Segmentació de la imatge | 5 |
| Característiques dels histogrames | 5 |
| Quantitat d'histogrames | 5 |
| Tractament dels histogrames | 5 |
| Descripció dels experiments realitzats | 7 |
| Dimensions dels histogrames | 7 |
| Decisió de imatges model | 7 |
| Mètriques de comparació | 8 |
| Factors de selectivitat | 9 |
| Resultats obtinguts | 10 |
| Resultats amb histogrames en RGB | 10 |
| Resultats amb histogrames en HSV | 10 |
| Funcions implementades | 12 |
| Annex 1 | 13 |
| ReconoceBarca.m | 13 |
| esDelBarca.m | 15 |
| rbghist.m | 17 |
| hsvhist.m | 18 |
| distanciaEntreHist.m | 19 |
| Etapa 2 | 20 |
| Modificacions | 20 |
| Canvi de tractament del negre en RGB | 20 |
| Suprimir les comparacions dels subimatges de les vores de les imatges a reconèixer | 20 |
| Treure llindar per reconèixer | 20 |
| Canviar distancia euclídea per chi quadrat | 21 |
| Descripció dels nous experiments realitzats | 22 |
| Nova divisió de les subimatges | 22 |
| Nous criteris per detectar les samarretes | 22 |
| Canvi de mètrica de distancia | 22 |
| Resultats obtinguts | 24 |
| Annex 2 | 27 |
| modelsRMadrid.m | 27 |
| modelsPSV.m | 28 |
| modelsMilan.m | 29 |

| | |
|--------------------------|----|
| modelsLiverpool.m | 30 |
| modelsJuventus.m | 31 |
| modelsChelsea.m | 32 |
| modelsBarca.m | 33 |
| main.m | 34 |
| distanciaEntreHists.m | 35 |
| hsvhist.m | 36 |
| inicialitzaReals.m | 37 |
| inicialitzaTaulaModels.m | 37 |
| puntuacioEquip.m | 39 |
| reconeixEquip.m | 39 |
| rgbhist.m | 41 |

Introducció

En aquest informe explicarem els diferents estats pels quals ha passat aquesta primera etapa del nostre projecte, comentarem les principals complicacions i presentarem els diferents resultats obtinguts durant aquesta primera etapa.

Principalment hem centrat el nostre estudi als histogrames de color i les propietats que aquests ens presenten. Per tal d'aprofitar aquests hem desenvolupat un algorisme que ens permet utilitzar els histogrames de color com a descriptor principal dels nostres estudis.

A continuació presentarem un resum del funcionament del nostre algorisme, posteriorment, durant l'apartat de descripció dels experiments, explicarem les diferents fases que hem tingut durant el desenvolupament i, per concloure, un estudi dels resultats extrets al llarg d'aquesta primera fase.

Etapa 1

Explicació de l'algorisme utilitzat

Segmentació de la imatge

Per treballar amb la imatge a reconèixer si hi ha algun jugador amb la samarreta del Barça, dividim la imatge amb 25 subimatges del mateix tamany (o molt similar). Per decidir, aquest tamany vam fer diferents proves amb diferents mides; vam provar amb 100 i 50 subimatges, però eren massa petites.

Característiques dels histogrames

En la creació dels histogrames, el nostre algorisme funciona creant un vector amb tots els components RGB. Seguidament normalitzem els components RG (eliminen el color Blau perquè és redundant). Es crea l'histograma amb la mida de núm.de bins seleccionat que en el nostre cas són 128 i finalment el normalitzem pel nombre de píxels que hi ha per poder fer la comparativa.

La mida de l'histograma és de 128 núm. de bins. Vam arribar a aquesta conclusió després d'haver provat amb 32 núm. de bins i 64 núm. de bins, però la informació que ens donava era molt poca per poder prendre una decisió, així que vam anar pujant fins a arribar a 128 que ens donen una informació suficient per poder, més endavant, fer la comparativa.

Quantitat d'histogrames

Respecte a l'histograma model, hem escollit quatre histogrames representatius, el primer i el segon histograma és d'imatges que es veu la camiseta de l'equip per davant, el tercer pels costats i el quart pel darrere. Hem agafat aquestes tres perspectives perquè considerem que són totes els possibles variants que pot haver-hi.

Per tal de reconèixer les imatges, al principi hem optat per implementar una finestra lliscant per poder reconèixer els equips, però ens vam adonar que seria molt costós pel que respecte al temps d'execució. Al final, per tal de fer més eficient el programa, hem desenvolupat un codi que adapta la idea de la finestra lliscant, que consisteix a dividir la imatge en 25 fragments, 25 perquè que considerem que la mida del fragment és adequat. Un cop s'obté el tros de la imatge, es compara amb tots els histogrames model.

Tractament dels histogrames

Per tal de poder obtenir resultats robustos als canvis d'il·luminació que poguéssim trobar a les nostres imatges, necessitem normalitzar els colors dels histogrames a comparar.

Per normalitzar els histogrames, hem aplicat la normalització de naril a partir d'una imatge RGB, ja que amb altres mètodes és molt més complicat realitzar la normalització. Aquesta

normalització consisteix en dividir cada component de color per la suma de les tres components. A més a més de aplicar aquest mètode de normalització, necessitem tractar els colors molt foscos per tal de que no generin problemes a l'hora de normalitzar amb el mètode esmentat anteriorment. Per tal de tractar aquests colors foscos el que farem és, abans de normalitzar, comprovarem els tres components de color d'un píxel i, si tots tres es troben per sota d'un llindar de fosc, llavors aquest píxel serà considerat massa fosc i es tindrà en compte al histograma.

Pel que fa al model HSV ens trobem amb una problemàtica important en relació al tractament del color blanc i el color negre. El model HSV ens permet obtenir un resultat robust als canvis d'il·luminació per la pròpia natura del model. En poder extreure el component V podem aconseguir extreure la lluminositat del color. Això, però, ens comporta que perdem la possibilitat d'identificar el color negre en la imatge, ja que necessitem el Value per poder identificar el negre. Si construïssim els histogrames sense el value es podria donar la situació que un color negre tingués un valor de Hue igual a 0 i confonguéssim aquest color negre amb el color vermell, donant lloc a resultats erronis.

Per aquest motiu hem de triar entre dues opcions: o bé construïm els histogrames HSV amb els tres components i podem identificar el color negre, o bé eliminem el component V i aconseguim un resultat robust als canvis d'il·luminació però perdem la possibilitat d'identificar el negre

Nosaltres hem aconseguit trobar una solució que ens permet tractar el color negre: Hem decidit construir histogrames bidimensionals (amb la H i la S), però comprovarem el valor del Value i si aquest es troba per sota d'un llindar de fosc no tindrem en compte el píxel per a l'histograma.

Per tal de comparar els histogrames hem hagut d'escollir una mètrica de distància entre aquests histogrames. Davant d'aquesta situació ens hem plantejat diverses opcions: la distància Chi Quadrat, la distància Euclídea, una suma de valors absoluts de les diferències, etc. Totes elles oferien diferents característiques, però al final ens vam decidir per la distància Euclídea.

Hem pres aquesta decisió, ja que era una mètrica senzilla d'implementar i en realitzar proves ens ha donat bons resultats. Podríem haver optat per mètriques més potents, com podria ser la distància Earth Movers, però aquestes impliquen una implementació més complexa i, a més, són més costoses computacionalment que la nostra. Donat que el nostre projecte no era crític per cap sistema d'importància i que els nostres recursos com a estudiants són bastant limitats, hem optat per una alternativa més senzilla com és la distància Euclídea.

Descripció dels experiments realitzats

Dimensions dels histogrames

Inicialment havíem decidit treballar amb histogrames de color de 3 dimensions. Aquesta decisió ens va limitar molt en el nombre de bins que podem escollir per construir els histogrames, ja que un histograma tridimensional de més de 64 bins ocupa més espai que la subimatge que tractarem en cada moment.

Però després de realitzar la normalització del color d'aquests histogrames vam poder eliminar un dels components en el model RGB (ja que un dels tres es redundant). Això va suposar un decrement important del espai que ocupa cada histograma i per tant vam poder treballar amb més bins i amb més informació.

Amb el model HSV no ens vam trobar amb aquesta problemàtica ja que desde l'inici ja treballavem amb histogrames de dues dimensions.

Decisió de imatges model

En començar a realitzar proves amb el nostre algorisme vam decidir escollir unes imatges representatives de les perspectives que ens podem trobar en les diferents imatges. Aquestes imatges que vam escollir enfocaven la samarreta pel davant, pel lateral i pel darrere.

Les imatges escollides per establir els histogrames models, al principi van ser relativament aleatòries, ja que vam escollir les imatges dels diferents costats sense tenir cap criteri definit més que l'esmentat anteriorment. Són les imatges de la figura 1.



Figura 1: Imatge escollides per establir els histogrames al inici

Després de realitzar proves, ens vam adonar que les imatges escollides no eren les més adequades, especialment les del lateral i darrere, perquè tenen molt de vermell, en tenir molt d'un color, fa que altres equips tinguessin un nivell de semblança molt alt al Barça, especialment el Liverpool i el Milan. Per aquest motiu, van agafar altres imatges en que es mostra un fragment de la samarreta amb els colors representatius de l'equip. En la figura 2 es pot veure les noves imatges escollides per establir els histogrames model.



Figura 2: Imatge escollides per establir els histogrames al final

Mètriques de comparació

Pel que respecta a la comparació dels histogrames al principi vam decidir establir que, un cop calculem la distància entre els dos histogrames (model i imatge a reconèixer), realitzarem una suma de tots els valors del vector que emmagatzema els resultats. Llavors establiríem un llindar que ens indiqués que, si la suma del vector de distàncies era inferior a aquest, llavors podem considerar que el histograma era prou semblant al histograma model i, en conseqüència, podem considerar que era del Barça.

Després de realitzar diferents proves varem veure que els llindars varien en funció del histograma model, ja que, per exemple, el histograma model de la imatge en que agafem la part de darrere de la samarreta també conté una part de groc (pel número de la samarreta i el nom) i això fa que siguin histogrames diferents entre sí. A més a més, també necessitavem establir un nombre fix de bins per histograma per tal de fixar els llindars, ja que un major nombre de bins també implica un increment en les distàncies obtingudes. Després de realitzar diferents observacions i de fixar les imatges que faríem servir com a model varem fixar uns valors pels llindars.

Un cop vam decidir repetir l'experiment amb el model HSV, necessitavem tornar a modificar els llindars pel canvi de model de color.

Factors de selectivitat

Un cop establerts aquests líndars, es van realitzar proves amb tots els equips, i els resultats obtinguts no eren gaire apropiats. Molts equips contenien un gran nombre de falsos positius així que necessitavem fer més selectius els paràmetres que estableixen si un equip era del Barça o no.

Per tal de ser més selectius ens vam plantejar diferents opcions, però al final vam optar per les següents.

- Establir un mínim de coincidències entre els histogrames. Per tal de no prendre qualsevol coincidència com un reconeixement del Barça, vam establir que el histograma de la subimatge hagués de coincidir amb, com a mínim, 3 dels histogrames model.
- Establir un mínim de reconeixements. Lligat amb l'opció anterior, vam decidir que en una mateixa imatge necessitavem reconèixer al Barça un mínim de cops per tal de poder establir que, verdaderament, la imatge contenia la samarreta del Barça. En el nostre cas vam decidir que s'hauria de reconèixer un mínim de dos cops, ja que un nombre major portava a uns resultats de reconeixement molt baixos.
- Incrementar el nombre de histogrames model. Un cop implementada la primera opció, vam pensar en que podríem incrementar el nombre mínim de coincidències si augmentàvem el nombre de histogrames model i, en conseqüència faríem més selectiu el nostre descriptor.

Després de implementar això vam aconseguir uns millors resultats, en comparació als obtinguts anteriorment.

Resultats obtinguts

Resultats amb histogrames en RGB

| Equip | Nombre d'imatges | Nombre d'imatges que no reconeix que són del Barça | Nombre d'imatges que reconeix que són del Barça | Falsos negatius |
|-------------|------------------|----------------------------------------------------|-------------------------------------------------|-----------------|
| PSV | 36 | 35 | 1 | 0 |
| Real Madrid | 36 | 35 | 1 | 0 |
| Liverpool | 36 | 36 | 0 | 0 |
| Juventus | 36 | 36 | 0 | 0 |
| Chelsea | 36 | 32 | 4 | 0 |
| ACMilan | 36 | 31 | 5 | 0 |
| Barcelona | 36 | 17 | 19 | 17 |

Podem observar en el resultats que descarta prou bé els equips que no pertanyen al Barcelona, però que en el reconeixement de les imatges que tenen samarretes del Barça, només en reconeix un 50% de la mostra .

Resultats amb histogrames en HSV

| Equip | Nombre d'imatges | Nombre d'imatges que no reconeix que són del Barça | Nombre d'imatges que reconeix que són del Barça | Falsos negatius |
|-------------|------------------|----------------------------------------------------|-------------------------------------------------|-----------------|
| PSV | 36 | 12 | 24 | 0 |
| Real Madrid | 36 | 26 | 10 | 0 |
| Liverpool | 36 | 21 | 15 | 0 |
| Juventus | 36 | 34 | 2 | 0 |
| Chelsea | 36 | 27 | 9 | 0 |
| ACMilan | 36 | 18 | 18 | 0 |
| Barcelona | 36 | 11 | 25 | 11 |

La representació HSV crea els colors mitjançant la tonalitat (quin color és), la saturació (si està molt aigualit) i el valor (ens indica si és fosc o no). Per tant, al crear l'histograma hem de tenir en compte amb que si eliminem la component S no podem identificar el blanc i si eliminem la component V no podem identificar el negre. D'aquesta manera vam decidir treballar fer un tractament especial per aquells pixels sense croma i ignorar-los a l'hora d'incrementar les posicions. Com podem veure en la taula el resultat no s'ha pogut aconseguir de manera adequada la creació de l'histograma fent els equips ratlles vermelles i blanques o ratlles vermelles i negres, els confongui amb l'equip de ratlles vermelles i blaves.

Funcions implementades

L'únic tros del codi que hem utilitzat d'altres fonts és el tros que construeix els histogrames de color, el tros de la comparació de la funció `rbghist.m`.

Hem decidit fer servir aquesta funció trobada per internet perquè ens va semblar una forma adequada i senzilla d'entendre, en comparació a altres alternatives plantejades. Tot i haver agafat aquesta funció, nosaltres la vàrem modificar per poder implementar els canvis necessaris per al nostre projecte, com per exemple la normalització del color.

Annex 1

ReconoceBarca.m

```
clear
%Imagen que coge la parte de delante de la camiseta
ImatgeFront1 = imread(strcat('barcelona/', '22.jpg'));
%Suavizamos la imagen antes de obtener el histograma
ImatgeFront1 = imgaussfilt3(ImatgeFront1);
crop = imcrop(ImatgeFront1);
HistogramaFront1 = rgbhist(crop,128,1);

%Imagen que coge la parte de atras de la camiseta
ImatgeBack = imread(strcat('barcelona/', '11.jpg'));
ImatgeBack = imgaussfilt3(ImatgeBack);
crop = imcrop(ImatgeBack);
HistogramaBack = rgbhist(crop,128,1);

%Imagen que coge la parte del costado de la camiseta
ImatgeSide = imread(strcat('barcelona/', '21.jpg'));
ImatgeSide = imgaussfilt3(ImatgeSide);
crop = imcrop(ImatgeSide);
HistogramaSide = rgbhist(crop,128,1);

%Imagen que coge la parte de delante de la camiseta
ImatgeFront2 = imread(strcat('barcelona/', '29.jpg'));
%Suavizamos la imagen antes de obtener el histograma
ImatgeFront2 = imgaussfilt3(ImatgeFront2);
crop = imcrop(ImatgeFront2);
HistogramaFront2 = rgbhist(crop,128,1);

%Contador de cuantas veces aparece el barca en los directorios
cuantosHay = 0;

% listingPSV = dir('psv');
% for ima = 3:size(listingPSV)
%     imagenActual = imread(strcat('psv/', listingPSV(ima).name));
%     loEs = esDelBarca(imagenActual, HistogramaFront1,
HistogramaBack, HistogramaSide, HistogramaFront2);
%     if loEs == 1
%         cuantosHay = cuantosHay + 1;
%     end
% end
%
% listingMADRID = dir('madrid');
% for ima = 3:size(listingMADRID)
```

```

%     imagenActual =
imread(strcat('madrid/',listingMADRID(ima).name));
%     loEs = esDelBarca(imagenActual, HistogramaFront1,
HistogramaBack, HistogramaSide, HistogramaFront2);
%     if loEs == 1
%         cuantosHay = cuantosHay + 1;
%     end
% end
%
% listingLIVERPOOL = dir('liverpool');
% for ima = 3:size(listingLIVERPOOL)
%     imagenActual =
imread(strcat('liverpool/',listingLIVERPOOL(ima).name));
%     loEs = esDelBarca(imagenActual, HistogramaFront1,
HistogramaBack, HistogramaSide, HistogramaFront2);
%     if loEs == 1
%         cuantosHay = cuantosHay + 1;
%     end
% end
%
% listingJUVENTUS = dir('juventus');
% for ima = 3:size(listingJUVENTUS)
%     imagenActual =
imread(strcat('juventus/',listingJUVENTUS(ima).name));
%     loEs = esDelBarca(imagenActual, HistogramaFront1,
HistogramaBack, HistogramaSide, HistogramaFront2);
%     if loEs == 1
%         cuantosHay = cuantosHay + 1;
%     end
% end
%
% listingCHELSEA = dir('chelsea');
% for ima = 3:size(listingCHELSEA)
%     imagenActual =
imread(strcat('chelsea/',listingCHELSEA(ima).name));
%     loEs = esDelBarca(imagenActual, HistogramaFront1,
HistogramaBack, HistogramaSide, HistogramaFront2);
%     if loEs == 1
%         cuantosHay = cuantosHay + 1;
%     end
% end
%
% listingACMILAN = dir('acmilan');
% for ima = 3:size(listingACMILAN)
%     imagenActual =
imread(strcat('acmilan/',listingACMILAN(ima).name));
%     loEs = esDelBarca(imagenActual, HistogramaFront1,
HistogramaBack, HistogramaSide, HistogramaFront2);
%     if loEs == 1
%         cuantosHay = cuantosHay + 1;
%     end
% end

```

```

% end
%
% listingBARCELONA = dir('barcelona');
% for ima = 3:size(listingBARCELONA)
%     imagenActual =
imread(strcat('barcelona/',listingBARCELONA(ima).name));
%     loEs = esDelBarca(imagenActual, HistogramaFront1,
HistogramaBack, HistogramaSide, HistogramaFront2);
%     if loEs == 1
%         cuantosHay = cuantosHay + 1;
%     end
% end

```

esDelBarca.m

```

function [result] = esDelBarca(Imagen,HistModel1,HistModel2,HistModel3,
HistModel4)

```

```

result = 0; % Variable que ens indica si el la imatge satisfà els
criteris per considerar que té un jugador del Barça
esDelBarca = 0; %variable que ens diu quants cops hem identificat a un
jugador del Barça

```

```

tamanhovertical = size(Imagen,1)/5;
tamanohorizontal = size(Imagen,2)/5;

```

```

for i = 1:tamanhovertical:(size(Imagen,1) - tamanhovertical)
    iend = i + tamanhovertical;
    for j = 1:tamanohorizontal:(size(Imagen,2) - tamanohorizontal)
        auxResultat = 0; %Variable que ens indica amb quants
histogrames tenim una distancia acceptable

```

```

        jend = j + tamanohorizontal;
        SubImagen = imcrop(Imagen, [j i jend-j iend-i]);
        imshow(SubImagen);

```

```

        HistoImagen = rgbhist(SubImagen,128,1);
        HistSuavitzat = imgaussfilt3(HistoImagen);

```

```

        %Comparar histogramas
        distanciaM1 = distanciaEntreHists(HistModel1,HistSuavitzat);
        sumaDistM1 = sum(distanciaM1);

```

```

    distanciaM2 = distanciaEntreHists(HistModel2,HistSuavitzat);
    sumaDistM2 = sum(distanciaM2);

    distanciaM3 = distanciaEntreHists(HistModel3,HistSuavitzat);
    sumaDistM3 = sum(distanciaM3);

    distanciaM4 = distanciaEntreHists(HistModel4,HistSuavitzat);
    sumaDistM4 = sum(distanciaM4);

    if sumaDistM1 <= 0.012
        auxResultat = auxResultat + 1;
    end

    if sumaDistM2 <= 0.008
        auxResultat = auxResultat + 1;
    end

    if sumaDistM3 <= 0.003
        auxResultat = auxResultat + 1;
    end

    if sumaDistM4 <= 0.0046
        auxResultat = auxResultat + 1;
    end
    % Si coincide con como minimo dos histogramas, lo consideramos
un
    % acierto
    if auxResultat > 2
        esDelBarca = esDelBarca + 1;
    end

    end
end

if esDelBarca >= 2
    result = 1;
end

end

```


rgbhist.m

```
function H = rgbhist(I,nBins,Nind)
% RGBHIST: color Histogram of an RGB image.
%
% nBins    : number of bins per EACH color => histogram is 'nBins^3'
long.
% Nind     : Normalization index
%
%          0 -> Un-Normalized historam
%          1 -> 11 normalized
%          2 -> 12 normalized
%
% H        : The vectorized histogram.
%
% Author   : Mopuri K Reddy, SERC, IISc, Bengalur, INDIA.
% Date     : 25/10/2013.
if (size(I, 3) ~= 3)
    error('rgbhist:numberOfSamples', 'Input image must be RGB.')
end
if(nargin<3)
    Nind=0;
    % Default is un-normalized histogram
end

H=zeros([nBins nBins]);
for i=1:size(I,1)
    for j=1:size(I,2)
        p=double(reshape(I(i,j,:),[1 3])); %Ajuntem els valors RGB del
        píxel i,j en una matriu 1x3

        %Fem que el color sigui invariant a la il·luminació
        p(1) = p(1)/(p(1) + p(2) + p(3));
        p(2) = p(2)/(p(1) + p(2) + p(3));
        %p(3) = p(3)/(p(1) + p(2) + p(3));
        %
        p=uint8(floor(p*(nBins)))+1; %Assignem els valors al bin
        corresponent del histograma (per això fem 256/nbins)
        if p(1) == nBins + 1
            p(1) = nBins;
        end
        if p(2) == nBins + 1
            p(2) = nBins;
        end
        %
        if p(3) == nBins + 1
        %
            p(3) = nBins;
        %
        end
        H(p(1),p(2))=H(p(1),p(2))+1; %Guardem els valors a les
        corresponents columnes del resultat
    end
end
```

```

H=H(:);

% Un-Normalized histogram
if(Nind==1)
    H=H./sum(H);
    % l1 normalization
else if(Nind==2)
    H=normc(H);
    % l2 normalization
end
end
% We can use 'reshape' to get back to 3D histogram

```

hsvhist.m

```

function [H] = hsvhist(I,nBins,Nind)
%UNTITLED Summary of this function goes here
% Detailed explanation goes here
I = rgb2hsv(I);

H=zeros([nBins nBins]);
for i=1:size(I,1)
    for j=1:size(I,2)
        p=double(reshape(I(i,j,:),[1 2])); %Ajuntem els valors RGB del
        píxel i,j en una matriu 1x3
        p=uint8(floor(p*(nBins)))+1; %Assignem els valors al bin
        corresponent del histograma (per això fem 256/nbins)
        if p(1) == nBins + 1
            p(1) = nBins;
        end
        if p(2) == nBins + 1
            p(2) = nBins;
        end
        % if p(3) == nBins + 1
        % p(3) = nBins;
        % end
        H(p(1),p(2))=H(p(1),p(2))+1; %Guardem els valors a les
        corresponents columnes del resultat
    end
end
H=H(:);

% Un-Normalized histogram
if(Nind==1)
    H=H./sum(H);
    % l1 normalization
else if(Nind==2)
    H=normc(H);

```

```
        % l2 normalization
    end
end
```

distanciaEntreHist.m

```
function [resultat] = distanciaEntreHists(HistogramaBase,HistAComparar)
%De moment fem servir la distancia Euclidea
resultat = zeros(size(HistogramaBase,1),1);
for i = 1:size(HistogramaBase,1)
    resultat(i) = (HistogramaBase(i,1) - HistAComparar(i,1))^2;
end
```

Etapa 2

Modificacions

Per poder reconèixer els equips, hem fet una sèrie de modificacions en el codi:

Canvi de tractament del negre en RGB

Per no tenir problemes amb els colors foscos quan apliquem el mètode de normalització, necessitem tractar els colors molt foscos especialment per tal de que no generin problemes a l'hora de normalitzar amb el mètode esmentat anteriorment. Per tal de tractar aquests colors foscos el que farem és, abans de normalitzar, comprovarem els tres components de color d'un píxel i, si tots tres es troben per sota d'un llindar de foscor, llavors aquest píxel serà considerat massa fosc i no es tindrà en compte al histograma.

Suprimir les comparacions dels subimatges de les vores de les imatges a reconèixer

Pel tal de disminuir el temps d'execució, hem canviat el numero de fragments de la imatge que es farà la comparació amb els histogrames models. Abans hi havien 25 fragments amb les que es feia comparació. Amb el canvi, es suprimirà la comparació de les vores, és a dir, comencem a comptar desde la posició 2 i acabem en el penúltim fragment. Amb això podem disminuir el número de comparacions, i d'aquesta manera, disminuïrem el temps d'execució. A més a més, aquest canvi millorarà la detecció d'equips, ja que el que hem observat de moltes imatges, és que a les vores de les imatges no hi ha molta informació rellevant.

Treure llindar per reconeixer

En el reconeixement de l'equip Barça, només era necessari saber si l'equip de la imatge era de Barça o no, per tant, amb llindars és suficient per fer la detecció. Però en l'etapa 2, en tenir diferents equips, el mètode d'utilitzar llindar no serà viable, per aquesta raó hem suprimit el llindar. En l'etapa 2, per poder reconèixer l'equip de la imatge, el que fem és comparar els fragments de la imatge amb tots els histogrames models de tots els equips, veure amb qui té la distància mínima i tenir un comptador per poder saber l'equip que té el màxim nombre de fragments amb distància mínima amb l'histograma model de l'equip. Si l'equip té el màxim número, llavors considerem que la imatge és d'aquell equip.

Canviar distancia euclídea per chi quadrat

Al principi fèiem la comparació dels histogrames amb la distància Euclídea, però donava resultats poc agradables, per aquest motiu hem provat de fer la comparació amb Chi-quadrat i ens ha donat millors resultats.

Descripció dels nous experiments realitzats

Nova divisió de les subimatges

En començar aquesta nova etapa del projecte ens vam plantejar com millorar la divisió de les imatges a reconèixer, ja que ara hauríem de fer un major nombre de comparacions i trobàvem que seria molt ineficient mantenir el sistema que havíem fet servir en la primera etapa. Per aquest motiu vam trobar que seria millor no tenir en compte les subimatges de la imatge a reconèixer que es troben a la "vora", és a dir, ara només tindrem en compte les subimatges que es troben al centre de la imatge.

Creiem que això és molt millor que el que tenim fins ara, ja que només agafarem les subimatges que poden contenir les samarretes dels equips i a més reduïrem el nombre de comparacions. A més a més, fent servir aquest mètode evitem agafar parts del fons de la imatge que poden causar resultats erronis a l'hora de reconèixer la samarreta de l'equip.

Nous criteris per detectar les samarretes

Un cop ja havíem fet els canvis necessaris a les altres parts de l'algorisme ens vam posar a modificar els criteris de reconeixement de les samarretes.

Com ara necessitàvem reconèixer de quin equip era la imatge a tractar, en comptes de decidir si era o no del Barça, vam trobar necessari canviar el criteri de detecció. Pel que feia al reconeixement del Barça, vàrem establir uns llindars de semblança entre els histogrames models i l'histograma actual, fent servir la distància entre els histogrames com la puntuació que hauria de passar el llindar.

Aquest mètode era molt útil per filtrar imatges que tinguessin molt poca semblança amb un equip en concret, però ja que ara necessitem detectar de quin dels 7 equips és una imatge en concret, aquest criteri de detecció ens sembla inútil. No ens serviria de res establir un llindar, ja que si més d'un equip passes aquest, hauríem de triar entre aquests per escollir quin equip s'ha reconegut. Per això vam decidir eliminar els llindars i, entre les diferents comparacions amb els histogrames models de cada equip, triar el resultat més petit.

Per obtenir la puntuació de semblança de cada equip vam decidir tenir en compte totes les comparacions amb els histogrames models, fent la mitjana de tots els resultats i, un cop obtinguts els resultats de cada equip, triar la puntuació mínima, ja que seria la que més s'assembla a l'histograma actual.

També vam ponderar l'opció de triar el resultat mínim de les comparacions amb els histogrames models de cada equip, però la vam descartar, ja que ens semblava erroni no tenir en compte totes les comparacions de cada equip, ja que es podria donar un cas extrem amb alguns dels histogrames models que provoques un reconeixement erroni.

Canvi de mètrica de distància

En realitzar proves amb el nou algorisme de reconeixement, vàrem observar que els resultats obtinguts no eren gaire satisfactoris, així que vam pensar en aspectes que podíem modificar per tal de millorar-los.

En primer lloc vàrem pensar a augmentar el mínim de reconeixements necessaris per establir que la detecció era correcta, però això ens va portar a resultats on quasi cap imatge era reconeguda com un dels equips del nostre projecte.

A continuació ens vam plantejar augmentar el nombre d'histogrames models amb els quals fer la comparació, però això tampoc ens va proporcionar una millora substancial en els resultats.

Ja que canviar els criteris de detecció no ens va proporcionar cap millora, vàrem decidir canviar la mètrica de distància entre els histogrames. En comptes de fer servir la distància Euclidea, vàrem usar la distància chi quadrat. En realitzar aquest canvi vàrem obtenir una millora dels resultats prou significativa.

Resultats obtinguts

A l'hora d'obtenir el valor amb el qual s'haurà de fer la decisió a quin equip pertany la imatge, fem el càlcul del valor amb la mitjana de les distàncies de cada model. Fem d'aquesta manera el càlcul per no tractar amb un valor particular de l'estudi, és a dir, si haguéssim usat el mínim absolut podria ser que aquell valor no representes els altres obtinguts.

En les taules inferiors es representen la matriu de confusió usant les distàncies euclidianes i les distàncies chi-quadrat en els models HSV i RGB. La matriu està formada de 8 files i 8 columnes, cada columna/fila representa:

1. Els resultats que no es coneix el seu equip
2. Els resultats referents al Barcelona
3. Els resultats referents al Chelsea
4. Els resultats referents al Liverpool
5. Els resultats referents al Juventus
6. Els resultats referents al Milan
7. Els resultats referents al PSV
8. Els resultats referents al Madrid

Com podem veure en els resultats, la distància chi-quadrat obté millors resultats en general, ja que la diagonal principal té més valors d'encert. Tot i així podem apreciar que els resultats de la chi-quadrat no són perfectes, ja que confon molts valors com si no tinguessin equip.

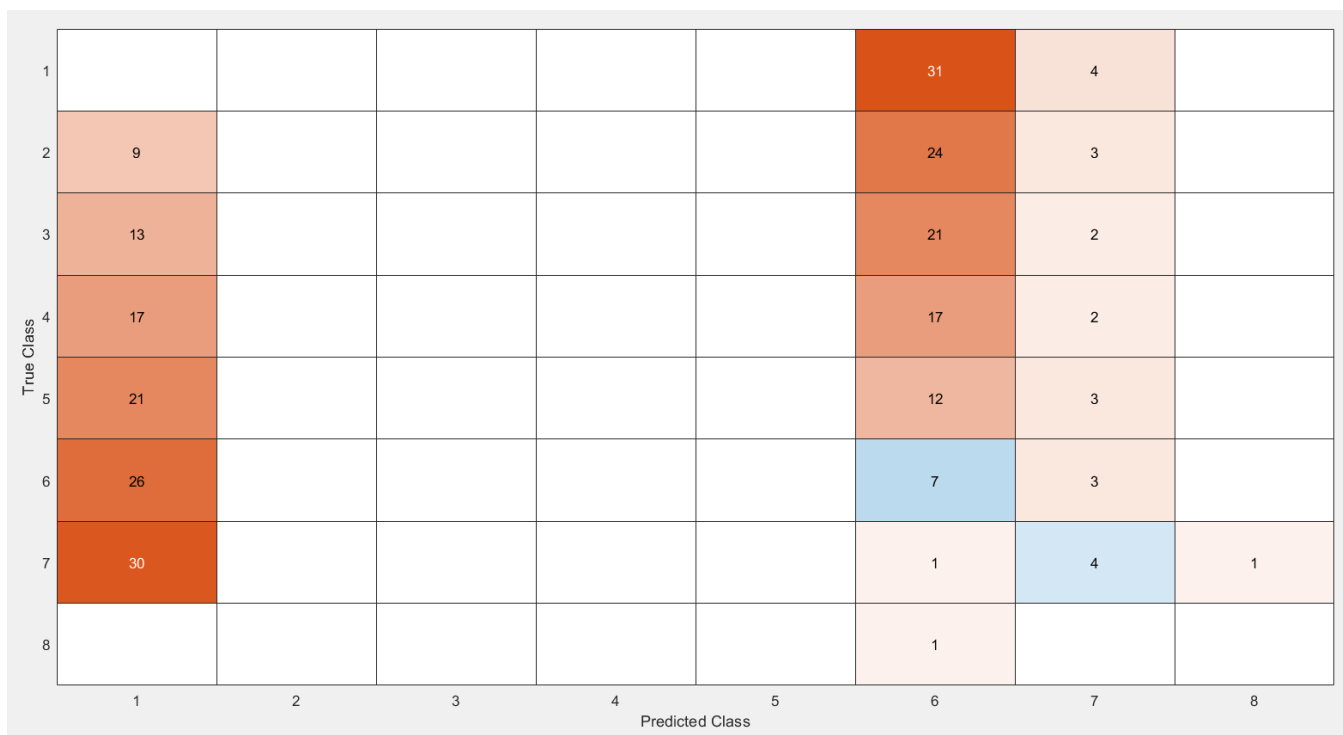


Figura 3: Resultat distància Euclidiana RGB

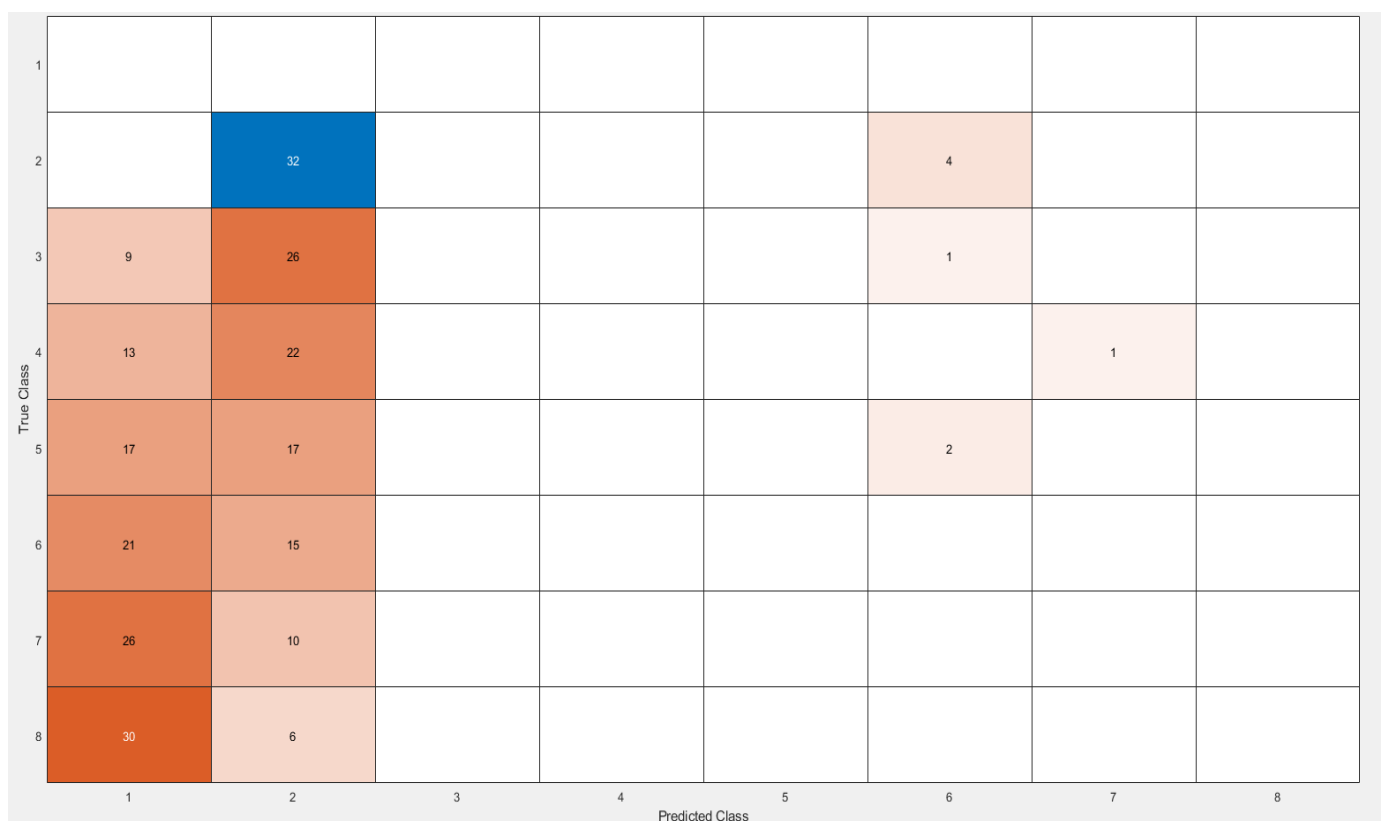


Figura 4: Resultats distància Euclidiana HSV

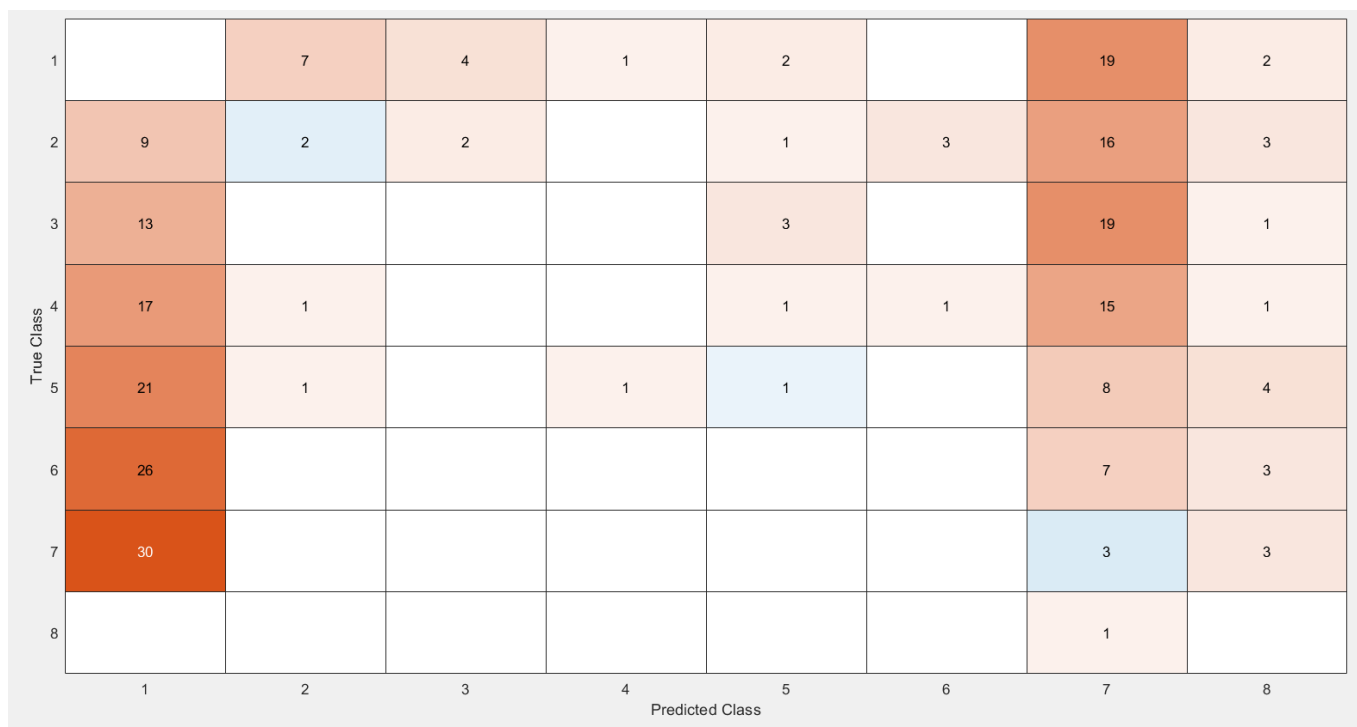


Figura 5: Resultat distància chi-quadrat RGB

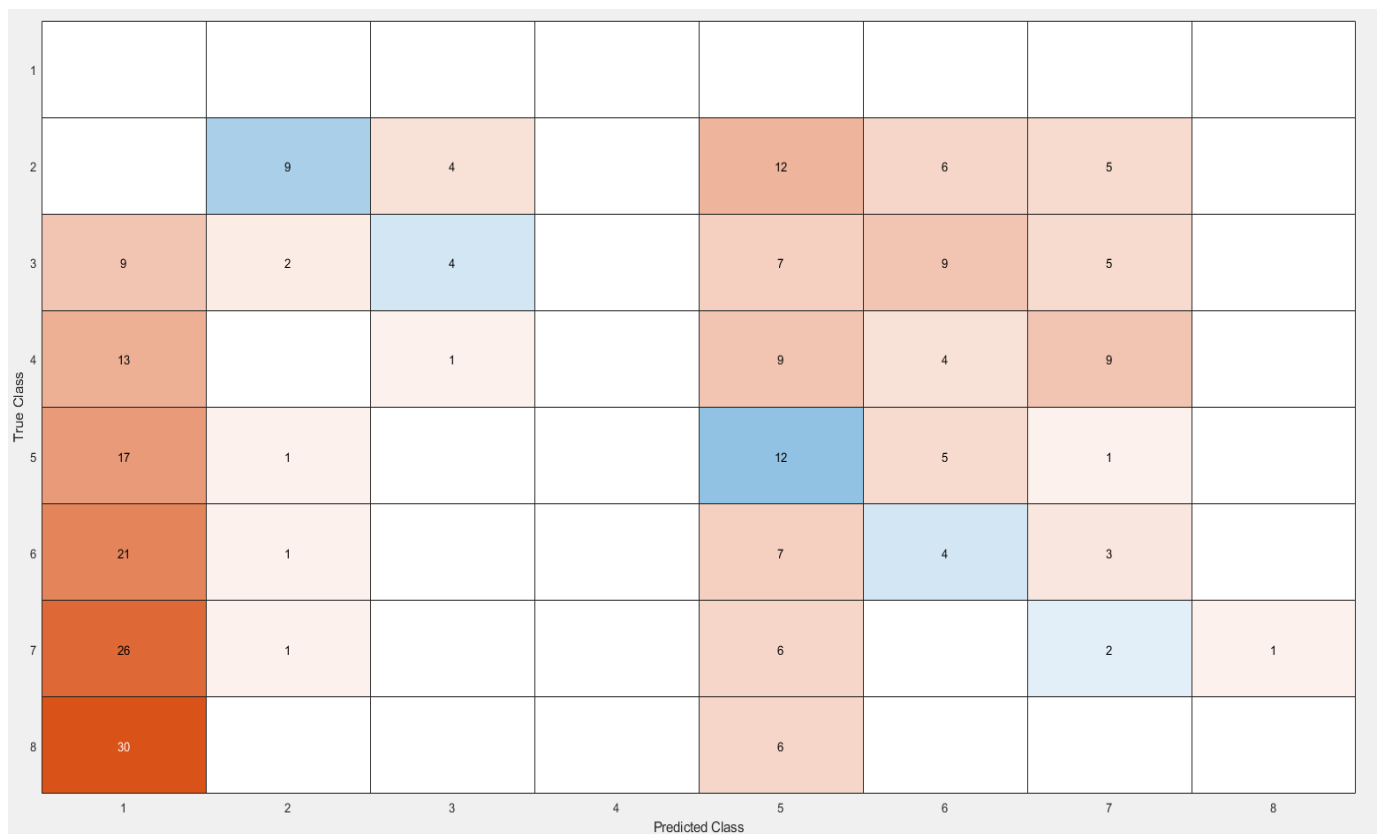


Figura 6: Resultats distància chi-quadrat HSV

Com podem observar en les matrius de confusió, els resultats no han estat gaire bons. La millor versió podem veure que és la de chi-quadrat amb el model HSV, ja que és la que millor reconeix en general els equips que hi ha.

En aquest projecte no hem pogut provar altres models de color o mètriques de distància diferents, però aquestes podrien suposar potencials millores en els resultats obtinguts.

Annex 2

Per tal de poder executar la funció main.m, és necessari afegir al path la carpeta FuncionsModels, la qual conté les funcions necessàries per crear els histogrames models de cada equip.

modelsRMadrid.m

```
function [hist1,hist2, hist3,hist4] = modelsRMadrid()

%Imagen que coge la parte de delante de la camiseta
ImatgeFront1 = imread(strcat('madrid/','03.jpg'));
%Suavizamos la imagen antes de obtener el histograma
ImatgeFront1 = imgaussfilt3(ImatgeFront1);
rect1 = [69 57 35 47];
crop = imcrop(ImatgeFront1, rect1);
%RGB
%hist1 = rgbhist(crop,128,1);
%HSV
hist1 = hsvhist(crop,128,1);

%Imagen que coge la parte de atras de la camiseta
ImatgeBack = imread(strcat('madrid/','15.jpg'));
ImatgeBack = imgaussfilt3(ImatgeBack);
rect2= [153 129 87 68];
crop = imcrop(ImatgeBack,rect2);
%RGB
%hist2 = rgbhist(crop,128,1);
%HSV
hist2 = hsvhist(crop,128,1);

%Imagen que coge la parte del costado de la camiseta
ImatgeSide = imread(strcat('madrid/','25.jpg'));
ImatgeSide = imgaussfilt3(ImatgeSide);
rect3 = [143 148 52 71];
crop = imcrop(ImatgeSide, rect3);
%RGB
%hist3 = rgbhist(crop,128,1);
%HSV
hist3 = hsvhist(crop,128,1);

%Imagen que coge la parte de delante de la camiseta
ImatgeFront2 = imread(strcat('madrid/','29.jpg'));
ImatgeFront2 = imgaussfilt3(ImatgeFront2);
rect4 = [18 83 51 51];
crop = imcrop(ImatgeFront2,rect4);
%RGB
```

```

%hist4 = rgbhist(crop,128,1);
%HSV
hist4 = hsvhist(crop,128,1);

end

```

modelsPSV.m

```

function [hist1,hist2, hist3,hist4] = modelsPSV()

%Imagen que coge la parte de delante de la camiseta
ImatgeFront1 = imread(strcat('psv/','02.jpg'));
%Suavizamos la imagen antes de obtener el histograma
ImatgeFront1 = imgaussfilt3(ImatgeFront1);
rect1 = [186,1.305000000000001e+02,90,119];
crop = imcrop(ImatgeFront1, rect1);
%RGB
%hist1 = rgbhist(crop,128,1);
%HSV
hist1 = hsvhist(crop,128,1);

%Imagen que coge la parte de atras de la camiseta
ImatgeBack = imread(strcat('psv/','07.jpg'));
ImatgeBack = imgaussfilt3(ImatgeBack);
rect2= [233,54,45,65];
crop = imcrop(ImatgeBack,rect2);
%RGB
%hist2 = rgbhist(crop,128,1);
%HSV
hist2 = hsvhist(crop,128,1);

%Imagen que coge la parte del costado de la camiseta
ImatgeSide = imread(strcat('psv/','30.jpg'));
ImatgeSide = imgaussfilt3(ImatgeSide);
rect3 = [108,75,33,57];
crop = imcrop(ImatgeSide, rect3);
%RGB
%hist3 = rgbhist(crop,128,1);
%HSV
hist3 = hsvhist(crop,128,1);

%Imagen que coge la parte de delante de la camiseta
ImatgeFront2 = imread(strcat('psv/','33.jpg'));
ImatgeFront2 = imgaussfilt3(ImatgeFront2);
rect4 = [93.99999999999990,101,53.000000000000014,61];
crop = imcrop(ImatgeFront2,rect4);
%RGB

```

```

%hist4 = rgbhist(crop,128,1);
%HSV
hist4 = hsvhist(crop,128,1);
end

```

modelsMilan.m

```

function [hist1,hist2, hist3,hist4] = modelsMilan()

%Imagen que coge la parte de delante de la camiseta
ImatgeFront1 = imread(strcat('acmilan/','30.jpg'));
%Suavizamos la imagen antes de obtener el histograma
ImatgeFront1 = imgaussfilt3(ImatgeFront1);
rect1 = [52 117 63 78];
crop = imcrop(ImatgeFront1, rect1);
%RGB
%hist1 = rgbhist(crop,128,1);
%HSV
hist1 = hsvhist(crop,128,1);

%Imagen que coge la parte de atras de la camiseta
ImatgeBack = imread(strcat('acmilan/','29.jpg'));
ImatgeBack = imgaussfilt3(ImatgeBack);
rect2= [103 88 97 118];
crop = imcrop(ImatgeBack,rect2);
%RGB
%hist2 = rgbhist(crop,128,1);
%HSV
hist2 = hsvhist(crop,128,1);

%Imagen que coge la parte del costado de la camiseta
ImatgeSide = imread(strcat('acmilan/','21.jpg'));
ImatgeSide = imgaussfilt3(ImatgeSide);
rect3 = [134 105 48 76];
crop = imcrop(ImatgeSide, rect3);
%RGB
%hist3 = rgbhist(crop,128,1);
%HSV
hist3 = hsvhist(crop,128,1);

%Imagen que coge la parte de delante de la camiseta
ImatgeFront2 = imread(strcat('acmilan/','07.jpg'));
ImatgeFront2 = imgaussfilt3(ImatgeFront2);
rect4 = [67 107 122 88];
crop = imcrop(ImatgeFront2,rect4);
%RGB
%hist4 = rgbhist(crop,128,1);

```

```
%HSV
hist4 = hsvhist(crop,128,1);

end
```

modelsLiverpool.m

```
function [hist1,hist2, hist3,hist4] = modelsLiverpool()

%Imagen que coge la parte de delante de la camiseta
ImatgeFront1 = imread(strcat('liverpool/','16.jpg'));
%Suavizamos la imagen antes de obtener el histograma
ImatgeFront1 = imgaussfilt3(ImatgeFront1);
rect1 = [282 75 35 64];
crop = imcrop(ImatgeFront1, rect1);
%RGB
%hist1 = rgbhist(crop,128,1);
%HSV
hist1 = hsvhist(crop,128,1);;

%Imagen que coge la parte de atras de la camiseta
ImatgeBack = imread(strcat('liverpool/','11.jpg'));
ImatgeBack = imgaussfilt3(ImatgeBack);
rect2= [274 84 54 60];
crop = imcrop(ImatgeBack,rect2);
%RGB
%hist2 = rgbhist(crop,128,1);
%HSV
hist2 = hsvhist(crop,128,1);

%Imagen que coge la parte del costado de la camiseta
ImatgeSide = imread(strcat('liverpool/','06.jpg'));
ImatgeSide = imgaussfilt3(ImatgeSide);
rect3 = [112 125 67 123];
crop = imcrop(ImatgeSide, rect3);
%RGB
%hist3 = rgbhist(crop,128,1);
%HSV
hist3 = hsvhist(crop,128,1);

%Imagen que coge la parte de delante de la camiseta
ImatgeFront2 = imread(strcat('liverpool/','10.jpg'));
ImatgeFront2 = imgaussfilt3(ImatgeFront2);
rect4 = [310 153 51 61];
crop = imcrop(ImatgeFront2,rect4);
%RGB
```

```

%hist4 = rgbhist(crop,128,1);
%HSV
hist4 = hsvhist(crop,128,1);

end

```

modelsJuventus.m

```

function [hist1,hist2, hist3,hist4] = modelsJuventus()

%Imagen que coge la parte de delante de la camiseta
ImatgeFront1 = imread(strcat('juventus/','19.jpg'));
%Suavizamos la imagen antes de obtener el histograma
ImatgeFront1 = imgaussfilt3(ImatgeFront1);
rect1 = [48,114,49,53];
crop = imcrop(ImatgeFront1, rect1);
%RGB
%hist1 = rgbhist(crop,128,1);
%HSV
hist1 = hsvhist(crop,128,1);

%Imagen que coge la parte de atras de la camiseta
ImatgeBack = imread(strcat('juventus/','16.jpg'));
ImatgeBack = imgaussfilt3(ImatgeBack);
rect2=
[106,1.3499999999999999e+02,1.1100000000000000e+02,1.4599999999999999
e+02];
crop = imcrop(ImatgeBack,rect2);
%RGB
%hist2 = rgbhist(crop,128,1);
%HSV
hist2 = hsvhist(crop,128,1);

%Imagen que coge la parte del costado de la camiseta
ImatgeSide = imread(strcat('juventus/','15.jpg'));
ImatgeSide = imgaussfilt3(ImatgeSide);
rect3 = [24,1.1200000000000000e+02,104,99.0000000000000030];
crop = imcrop(ImatgeSide, rect3);
%RGB
%hist3 = rgbhist(crop,128,1);
%HSV
hist3 = hsvhist(crop,128,1);

%Imagen que coge la parte de delante de la camiseta
ImatgeFront2 = imread(strcat('juventus/','10.jpg'));
ImatgeFront2 = imgaussfilt3(ImatgeFront2);

```

```

rect4 = [49.999999999999986,247,47,52];
crop = imcrop(ImatgeFront2,rect4);
%RGB
%hist4 = rgbhist(crop,128,1);
%HSV
hist4 = hsvhist(crop,128,1);

end

```

modelsChelsea.m

```

function [hist1,hist2,hist3,hist4] = modelsChelsea()

ImatgeFront1 = imread(strcat('chelsea/','11.jpg'));
%Suavizamos la imagen antes de obtener el histograma
ImatgeFront1 = imgaussfilt3(ImatgeFront1);
rect1 = [51.5 103.5 83 105];
crop = imcrop(ImatgeFront1,rect1);
%RGB
%hist1 = rgbhist(crop,128,1);
%HSV
hist1 = hsvhist(crop,128,1);

%Imagen que coge la parte de atras de la camiseta
ImatgeBack = imread(strcat('chelsea/','32.jpg'));
ImatgeBack = imgaussfilt3(ImatgeBack);
rect2= [341.5 193.5 67 87];
crop = imcrop(ImatgeBack, rect2);
%RGB
%hist2 = rgbhist(crop,128,1);
%HSV
hist2 = hsvhist(crop,128,1);

%Imagen que coge la parte del costado de la camiseta
ImatgeSide = imread(strcat('chelsea/','25.jpg'));
ImatgeSide = imgaussfilt3(ImatgeSide);
rect3 = [123.5 90.5 48 84];
crop = imcrop(ImatgeSide,rect3);
%RGB
%hist3 = rgbhist(crop,128,1);
%HSV
hist3 = hsvhist(crop,128,1);

%Imagen que coge la parte de delante de la camiseta
ImatgeFront2 = imread(strcat('chelsea/','29.jpg'));

```



```

ImatgeFront2 = imgaussfilt3(ImatgeFront2);
rect4 = [106.5 111.5 44 55];
crop = imcrop(ImatgeFront2, rect4);
%RGB
%hist4 = rgbhist(crop,128,1);
%HSV
hist4 = hsvhist(crop,128,1);

end

```

modelsBarca.m

```

function [hist1,hist2, hist3,hist4] = modelsBarca()

%Imagen que coge la parte de delante de la camiseta
ImatgeFront1 = imread(strcat('barcelona/', '22.jpg'));
%Suavizamos la imagen antes de obtener el histograma
ImatgeFront1 = imgaussfilt3(ImatgeFront1);
rect1 = [67 158 89 117];
crop = imcrop(ImatgeFront1, rect1);
%RGB
%hist1 = rgbhist(crop,128,1);
%HSV
hist1 = hsvhist(crop,128,1);

%Imagen que coge la parte de atras de la camiseta
ImatgeBack = imread(strcat('barcelona/', '11.jpg'));
ImatgeBack = imgaussfilt3(ImatgeBack);
rect2= [19 99 67 113];
crop = imcrop(ImatgeBack,rect2);
%RGB
%hist2 = rgbhist(crop,128,1);
%HSV
hist2 = hsvhist(crop,128,1);

%Imagen que coge la parte del costado de la camiseta
ImatgeSide = imread(strcat('barcelona/', '21.jpg'));
ImatgeSide = imgaussfilt3(ImatgeSide);
rect3 = [200.5 110.5 56 130];
crop = imcrop(ImatgeSide, rect3);
%RGB
%hist3 = rgbhist(crop,128,1);
%HSV
hist3 = hsvhist(crop,128,1);

%Imagen que coge la parte de delante de la camiseta
ImatgeFront2 = imread(strcat('barcelona/', '29.jpg'));

```

```

ImatgeFront2 = imgaussfilt3(ImatgeFront2);
rect4 = [90.5 187.5 145 194];
crop = imcrop(ImatgeFront2,rect4);
%RGB
%hist4 = rgbhist(crop,128,1);
%HSV
hist4 = hsvhist(crop,128,1);
end

```

main.m

```

clear

taulaDeModels = inicialitzaTaulaModels();

resultatsReals = inicialitzaReals();
resultats = zeros(252,1);

%1=Barca
%2=Chelsea
%3=Liverpool
%4=Juventus
%5=Milan
%6=PSV
%7=Madrid
%
listingBARCELONA = dir('barcelona');
for ima = 3:size(listingBARCELONA)
    imagenActual =
imread(strcat('barcelona/',listingBARCELONA(ima).name));
    equip = reconeixEquip(imagenActual,taulaDeModels);
    resultats(ima - 2,1) = equip;
end

listingCHELSEA = dir('chelsea');
for ima = 3:size(listingCHELSEA)
    imagenActual =
imread(strcat('chelsea/',listingCHELSEA(ima).name));
    equip = reconeixEquip(imagenActual,taulaDeModels);
    resultats((ima-2)*2,1) = equip;
end

listingLIVERPOOL = dir('liverpool');
for ima = 3:size(listingLIVERPOOL)
    imagenActual =
imread(strcat('liverpool/',listingLIVERPOOL(ima).name));

```

```

        equip = reconeixEquip(imagenActual,taulaDeModels);
        resultats((ima-2)*3,1) = equip;
    end

    listingJUVENTUS = dir('juventus');
    for ima = 3:size(listingJUVENTUS)
        imagenActual =
    imread(strcat('juventus/',listingJUVENTUS(ima).name));
        equip = reconeixEquip(imagenActual,taulaDeModels);
        resultats((ima-2)*4 ,1) = equip;
    end

    listingACMILAN = dir('acmilan');
    for ima = 3:size(listingACMILAN)
        imagenActual =
    imread(strcat('acmilan/',listingACMILAN(ima).name));
        equip = reconeixEquip(imagenActual,taulaDeModels);
        resultats((ima-2)*5,1) = equip;
    end

    listingPSV = dir('psv');
    for ima = 3:size(listingPSV)
        imagenActual = imread(strcat('psv/',listingPSV(ima).name));
        equip = reconeixEquip(imagenActual,taulaDeModels);
        resultats((ima-2)*6,1) = equip;
    end
    %
    listingMADRID = dir('madrid');
    for ima = 3:size(listingMADRID)
        imagenActual =
    imread(strcat('madrid/',listingMADRID(ima).name));
        equip = reconeixEquip(imagenActual,taulaDeModels);
        resultats((ima-2)*7,1) = equip;
    end

    C = confusionmat(resultatsReals,resultats);
    confusionchart(C);

```

distanciaEntreHists.m

```

function [resultat] =
distanciaEntreHists(HistogramaBase,HistAComparar)
%De moment fem servir la distancia Euclidea
resultat = zeros(size(HistogramaBase,1),1);
for i = 1:size(HistogramaBase,1)

```

```

asota = (HistogramaBase(i,1)+ HistAComparar(i,1) );
if asota == 0
    resultat(i) = 0;
else
    %chi-quadrat
    resultat(i) = ( (HistogramaBase(i,1) - HistAComparar(i,1) ) ^2
/(HistogramaBase(i,1)+ HistAComparar(i,1) ) );
    %Euclidia
    %resultat(i) = ( (HistogramaBase(i,1) - HistAComparar(i,1)
)^2 );
end
end
end

```

hsvhist.m

```

function [H] = hsvhist(I,nBins,Nind)
%UNTITLED Summary of this function goes here
% Detailed explanation goes here
I = rgb2hsv(I);

H=zeros([nBins nBins]);
for i=1:size(I,1)
    for j=1:size(I,2)
        p=double(reshape(I(i,j,:),[1 3])); %Ajuntem els valors RGB
del píxel i,j en una matriu 1x3
        if p(3) >= 0.1 && p(2) >= 0.1
            p=uint8(floor(p*(nBins)))+1; %Assignem els valors al
bin corresponent del histograma (per això fem 256/nbins)
            if p(1) == nBins + 1
                p(1) = nBins;
            end
            if p(2) == nBins + 1
                p(2) = nBins;
            end
            % if p(3) == nBins + 1
            % p(3) = nBins;
            % end
            H(p(1),p(2))=H(p(1),p(2))+1; %Guardem els valors a les
corresponents columnes del resultat
        end
    end
end
end
H=H(:);

% Un-Normalized histogram
if(Nind==1)

```

```

        H=H./sum(H);
        % l1 normalization
    else if(Nind==2)
        H=normc(H);
        % l2 normalization
    end
end
end

```

inicialitzaReals.m

```

function [result] = inicialitzaReals()
result = zeros(252,1);

for i = 1:252
    result(i,1) = idivide(int64(i),int64(36),'ceil');
end

```

inicialitzaTaulaModels.m

```

function [taulaModels] = inicialitzaTaulaModels()

%1=Barca
%2=Chelsea
%3=Liverpool
%4=Juventus
%5=Milan
%6=PSV
%7=Madrid

taulaModels = cell(7,4);

[res1B,res2B,res3B,res4B] = modelsBarca();

taulaModels{1,1} = res1B;
taulaModels{1,2} = res2B;
taulaModels{1,3} = res3B;
taulaModels{1,4} = res4B;

[res1C,res2C,res3C,res4C] = modelsChelsea();
taulaModels{2,1} = res1C;
taulaModels{2,2} = res2C;
taulaModels{2,3} = res3C;
taulaModels{2,4} = res4C;

[res1L,res2L,res3L,res4L] = modelsLiverpool();

```

```

taulaModels{3,1} = res1L;
taulaModels{3,2} = res2L;
taulaModels{3,3} = res3L;
taulaModels{3,4} = res4L;

[res1J,res2J,res3J,res4J] = modelsJuventus();

taulaModels{4,1} = res1J;
taulaModels{4,2} = res2J;
taulaModels{4,3} = res3J;
taulaModels{4,4} = res4J;

[res1M,res2M,res3M,res4M] = modelsMilan();

taulaModels{5,1} = res1M;
taulaModels{5,2} = res2M;
taulaModels{5,3} = res3M;
taulaModels{5,4} = res4M;

[res1P,res2P,res3P,res4P] = modelsPSV();
taulaModels{6,1} = res1P;
taulaModels{6,2} = res2P;
taulaModels{6,3} = res3P;
taulaModels{6,4} = res4P;

[res1R,res2R,res3R,res4R] = modelsRMadrid();

taulaModels{7,1} = res1R;
taulaModels{7,2} = res2R;
taulaModels{7,3} = res3R;
taulaModels{7,4} = res4R;

end

```

puntuacioEquip.m

```
function [result] = puntuacioEquip(hist1,hist2,hist3,hist4,
histActual)

    distanciaM1 = distanciaEntreHists(hist1,histActual);
    sumaDistM1 = sum(distanciaM1);

    distanciaM2 = distanciaEntreHists(hist2,histActual);
    sumaDistM2 = sum(distanciaM2);

    distanciaM3 = distanciaEntreHists(hist3,histActual);
    sumaDistM3 = sum(distanciaM3);

    distanciaM4 = distanciaEntreHists(hist4,histActual);
    sumaDistM4 = sum(distanciaM4);

    aux = [sumaDistM1 sumaDistM2 sumaDistM3 sumaDistM4];

    result = mean(aux);
end
```

reconeixEquip.m

```
function [result] = reconeixEquip(Imagen,taulaDeModels)

result = 0; % Variable que ens indica si el la imatge satisfà els
criteris per considerar que té un jugador del Barça

contatgesEquips = zeros(1,7);
tamanovertical = size(Imagen,1)/5;
tamanohorizontal = size(Imagen,2)/5;

for i = 1*tamanovertical:tamanovertical:(size(Imagen,1) -
2*tamanovertical)
    iend = i + tamanovertical;
    for j = 1*tamanohorizontal:tamanohorizontal:(size(Imagen,2) -
2*tamanohorizontal)
        jend = j + tamanohorizontal;
        SubImagen = imcrop(Imagen, [j i jend-j iend-i]);
        imshow(SubImagen);

        %HistoImagen = rgbhist(SubImagen,128,1);
        HistoImagen = rgbhist(SubImagen,128,1);
    end
end
```

```

    HistSuavitzat = imgaussfilt3(HistoImagen);

    %Comparar histogramas
    puntuacioBarca = puntuacioEquip(taulaDeModels{1,1},
    taulaDeModels{1,2}, taulaDeModels{1,3}, taulaDeModels{1,4},
    HistSuavitzat);

    puntuacioChelsea = puntuacioEquip(taulaDeModels{2,1},
    taulaDeModels{2,2}, taulaDeModels{2,3}, taulaDeModels{2,4},
    HistSuavitzat);

    puntuacioLiverpool = puntuacioEquip(taulaDeModels{3,1},
    taulaDeModels{3,2}, taulaDeModels{3,3}, taulaDeModels{3,4},
    HistSuavitzat);

    puntuacioJuventus = puntuacioEquip(taulaDeModels{4,1},
    taulaDeModels{4,2}, taulaDeModels{4,3}, taulaDeModels{4,4},
    HistSuavitzat);

    puntuacioMilan = puntuacioEquip(taulaDeModels{5,1},
    taulaDeModels{5,2}, taulaDeModels{5,3}, taulaDeModels{5,4},
    HistSuavitzat);

    puntuacioPSV = puntuacioEquip(taulaDeModels{6,1},
    taulaDeModels{6,2}, taulaDeModels{6,3}, taulaDeModels{6,4},
    HistSuavitzat);

    puntuacioMadrid = puntuacioEquip(taulaDeModels{7,1},
    taulaDeModels{7,2}, taulaDeModels{7,3}, taulaDeModels{7,4},
    HistSuavitzat);

    %Ordenats en funció de la taula de histogrames model
    aux = [puntuacioBarca puntuacioChelsea puntuacioLiverpool
    puntuacioJuventus puntuacioMilan puntuacioPSV puntuacioMadrid];

    [result,posicio] = min(aux);
    contatgesEquips(1,posicio) = contatgesEquips(1,posicio) + 1
;

    end
end

[nombreAparicions,index] = max(contatgesEquips);

if nombreAparicions >= 2
    result = index;
end

```


rgbhist.m

```
function H = rgbhist(I,nBins,Nind)
% RGBHIST: color Histogram of an RGB image.
%
% nBins    : number of bins per EACH color => histogram is
'nbins^3' long.
% Nind     : Normalization index
%
%          0 -> Un-Normalized histogram
%          1 -> l1 normalized
%          2 -> l2 normalized
%
% H        : The vectorized histogram.
%
% Author   : Mopuri K Reddy, SERC, IISc, Bengalur, INDIA.
% Date     : 25/10/2013.
if (size(I, 3) ~= 3)
    error('rgbhist:numberOfSamples', 'Input image must be RGB.')
end
if(nargin<3)
    Nind=0;
    % Default is un-normalized histogram
end

H=zeros([nBins nBins]);
for i=1:size(I,1)
    for j=1:size(I,2)
        p=double(reshape(I(i,j,:),[1 3])); %Ajuntem els valors RGB
del píxel i,j en una matriu 1x3
        if ~(p(1) <= 65 && p(2) <= 65 && p(3) <= 65)
            %Fem que el color sigui invariant a la il·luminació
            p(1) = p(1)/(p(1) + p(2) + p(3));
            p(2) = p(2)/(p(1) + p(2) + p(3));
            %p(3) = p(3)/(p(1) + p(2) + p(3));
            %
            p=uint8(floor(p*(nBins)))+1; %Assignem els valors al bin
corresponent del histograma (per això fem 256/nbins)
            if p(1) == nBins + 1
                p(1) = nBins;
            end
            if p(2) == nBins + 1
                p(2) = nBins;
            end
            if p(3) == nBins + 1
```

```

%           p(3) = nBins;
%           end
%           H(p(1),p(2))=H(p(1),p(2))+1; %Guardem els valors a les
corresponents columnes del resultat
%           end
%           end
end
H=H(:);

% Un-Normalized histogram
if(Nind==1)
    H=H./sum(H);
    % l1 normalization
else if(Nind==2)
    H=normc(H);
    % l2 normalization
end
end
% We can use 'reshape' to get back to 3D histogram

```