



FIB

Facultat d'Informàtica
de Barcelona

Departament d'Enginyeria de Sistemes,
Automàtica i Informàtica Industrial

UNIVERSITAT POLITÈCNICA DE CATALUNYA

VISIÓ PER COMPUTADOR

SHORT PROJECT

Facultat d'Informàtica de Barcelona

**Miquel Gotanegra
Josep Navarro**

Barcelona, Juny de 2022

1. Objectius del projecte

L'objectiu del projecte és implementar un sistema automàtic per detectar la posició dels ulls implementant un descriptor d'imatges propi. La idea principal del projecte és trobar un bon descriptor d'imatges que funcioni correctament per a un classificador donat, enfront de l'altre estratègia que seria trobar un bon classificador per un descriptor d'imatges donat.

Les imatges amb les que es treballaran seran principalment amb un enquadrament del bust d'un usuari amb rotacions del cap poc apreciables en relació a la càmera.

2. Treball realitzat

Per tal d'assolir els objectius del projecte l'hem tractat assolint apartat per apartat el que ens demanaven fins finalment satisfer els propòsits.

Primerament, per recopilar la base de dades hem fet ús del codi en Python que tenim adjuntat al document del ShortProject, l'hem executat per les dues pàgines que ens aconsellen utilitzar en el document (*thispersondoesnotexist.com* i <https://picsum.photos/>) i les hem emmagatzemat en dues carpetes per distingir de les imatges que són cares i no cares.

Tot seguit hem procedit a carregar les imatges de les bases de dades. Per a les fotos cares, hem agafat un dels ulls i una zona en resolució 4:3 al voltant, sabent que a les fotos de mostra els ulls estan sempre a la mateixa posició. Per a la resta hem agafat una zona aleatòria de la foto de la mateixa mida que les anteriors. Finalment convertim les imatges de RGB a 256 nivells de gris.

A l'hora de fer el descriptor ens hem basat en pràctiques anteriors i treballs fets durant les classes de laboratori per veure quines característiques podríem utilitzar i quines ens poden resultar més útils. Després de redimensionar les imatges, hem acabat decidint de tenir en compte aspectes com la mitjana dels nivells de gris, la variància, la desviació típica i la magnitud del gradient.

En conseqüència, hem creat una taula amb les diferents variables dels descriptors i una última columna on tindrem si la respectiva imatge es tracta d'una imatge que conté ulls o no, l'etiqueta.

Un cop arribat aquest punt, per tal d'obtenir una major *accuracy*, el millor és segmentar la imatge, així que ens disposarem a provar el programa en quatre casos diferents; imatge sense subdivisions, imatge amb 3 subdivisions, imatge amb 6 subdivisions i finalment, imatge amb 12 subdivisions. Per escollir el nombre de subdivisions hem tingut en compte òbviament la mida de la imatge.

Per tal d'emmagatzemar la classificació de les imatges, podem veure que hem utilitzat un tipus d'array categòric. Aquest array el que ens permet és emmagatzemar categories, en el nostre cas, ens agradaria tenir en compte les dues categories Ulls / NoUlls.

Així doncs, un cop ja tenim totes les taules de descriptors de les imatges procedim a l'entrenament del kernel lineal per tal de veure en cada cas, quina és l'*accuracy* obtinguda per poder extreure conclusions.

Es pot observar que utilitzem un descriptor diferent per cada cas, és a dir tenim un descriptor per a la imatge amb 3 subdivisions, 6, 12 i sense subdivisions. L'única modificació que es fa en cada cas és la fragmentació en les diferents parts, on per cada part obtenim les característiques desitjades. Un cop obtingudes aquestes característiques procedim a fer el càlcul del mèrit emprant la fórmula que tenim al word del short project.

Finalment per l'edició del vídeo, per a cada frame passem el detector de cares i, per a cada cara que trobem, retallem la sub imatge i fem i passem el trainedClassifier12 per les zones on creiem que hi haurà d'haver els ulls. Si en detectem en una zona, la marquem amb un rectangle vermell.

Conclusions amb rati d'encert global, falsos positius i falsos negatius:

Durant la realització de la pràctica hem anat obtenint diferents conclusions.

Primerament ens hem adonat que per a les imatges amb cares en compte de fixar-nos o obtenir característiques sobre els ulls, estàvem obtenint les característiques de les cares. Per solucionar aquest problema, simplement hem modificat a l'hora de carregar de la base de dades les imatges ens quedem només amb la part dels ulls que és de la que ens interessa realment extreure les característiques.

Un cop hem modificat això, l'accuracy ha augmentat considerablement.

Per norma general, hem observat que quants més descriptors posem, millor serà l'accuracy, però ens hem adonat que per a les imatges amb moltes subdivisions, el fet de tenir en compte el gradient afectava negativament a la precisió. Això creiem que degut a que quantes més subdivisions té la imatge, menor la resolució de cada una, i per tant més difícil de veure els màxims/mínims locals ja que els valors que els envolten estan molt més limitats.

També hem fet comprovacions a l'hora de realitzar les diferents subdivisions, hem anat provant si era millor efectuar les subdivisions en horitzontal o en vertical, és a dir si al fer les 12 subdivisions era millor fer 4x3 o 3x4.

Un cop realitzades aquestes observacions procedim a veure els resultats de la accuracy segons quin cas estem efectuant.

Accuracy i mèrit

A les següents imatges podem veure la precisió de cada un dels myDescriptors, agafant un subconjunt dels descriptors. Per a tes les figures els descriptors agafats són :

{2: Mitjana;

3: Mitjana i Variança;

4: Mitjana, Variança i Desviació Estandard;

5: Mitjana, Variança, Desviació Estandard i ValorGradient}

Pel mèrit, l'hem calculat a partir del model amb el major percentatge d'encert.

Sense Subdivisions

☆ 2 SVM	Accuracy (Validation): 57.3%
Last change: Added feature 'Mitjana' 1/4 features	
☆ 3 SVM	Accuracy (Validation): 58.7%
Last change: Removed feature 'GradientMag' 2/4 feat	
☆ 4 SVM	Accuracy (Validation): 68.6%
Last change: Removed feature 'GradientMag' 3/4 feat	
☆ 5 SVM	Accuracy (Validation): 75.4%
Last change: Linear SVM 4/4 features	

Podem veure què quantes més característiques proporcionen al classificar, millor accuracy obtenim.

Accuracy = 0,754

Mèrit = 1,249

3 Subdivisions

Per a 3 subdivisions obtenim millors resultats per a tots els models respecte a fer-ho sense subdivisions. Observem que a quantes més característiques, millor accuracy.

Accuracy = 0,956
Mèrit = 1,377

☆ 2 SVM	Accuracy (Validation): 91.0%
Last change: Added feature 'Mitjana 3' 3/12 features	
☆ 3 SVM	Accuracy (Validation): 94.3%
Last change: Removed feature 'GradientMag 3' 6/12 f	
☆ 4 SVM	Accuracy (Validation): 95.3%
Last change: Removed feature 'GradientMag 3' 9/12 f	
☆ 5 SVM	Accuracy (Validation): 95.6%
Last change: Linear SVM 12/12 features	

6 Subdivisions

☆ 2 SVM	Accuracy (Validation): 92.0%
Last change: Added feature 'Mitjana 6' 6/24 features	
☆ 3 SVM	Accuracy (Validation): 95.6%
Last change: Removed feature 'GradientMag 6' 12/24	
☆ 4 SVM	Accuracy (Validation): 96.7%
Last change: Removed feature 'GradientMag 6' 18/24	
☆ 5 SVM	Accuracy (Validation): 96.4%
Last change: Linear SVM 24/24 features	

Per al model amb 6 blocs observem que és millor que el de tres, pero que no es compleix que al augmentar el nombre de features augmenti també la precisió. En aquest cas disminueix al afegir el valor del gradient. Això creiem que és degut a que al ser subimatges més petites, el gradient es una mètrica menys fiable ja que els màxims/mínims locals no seran tan obvis.

Accuracy = 0,967
Mèrit = 1,379

12 Subdivision

Pel cas de 12blocs veiem un resultats molt similars al anterior i per tant no seguirem subdividint la imatge, ja que això augmentaria el temps d'entrenament.

Tot i això, en alguns casos el model amb 12 subdivisions ens ha donat millors resultats en quant a la accuracy, per tant ha sigut el que hem agafat per realitzar el video

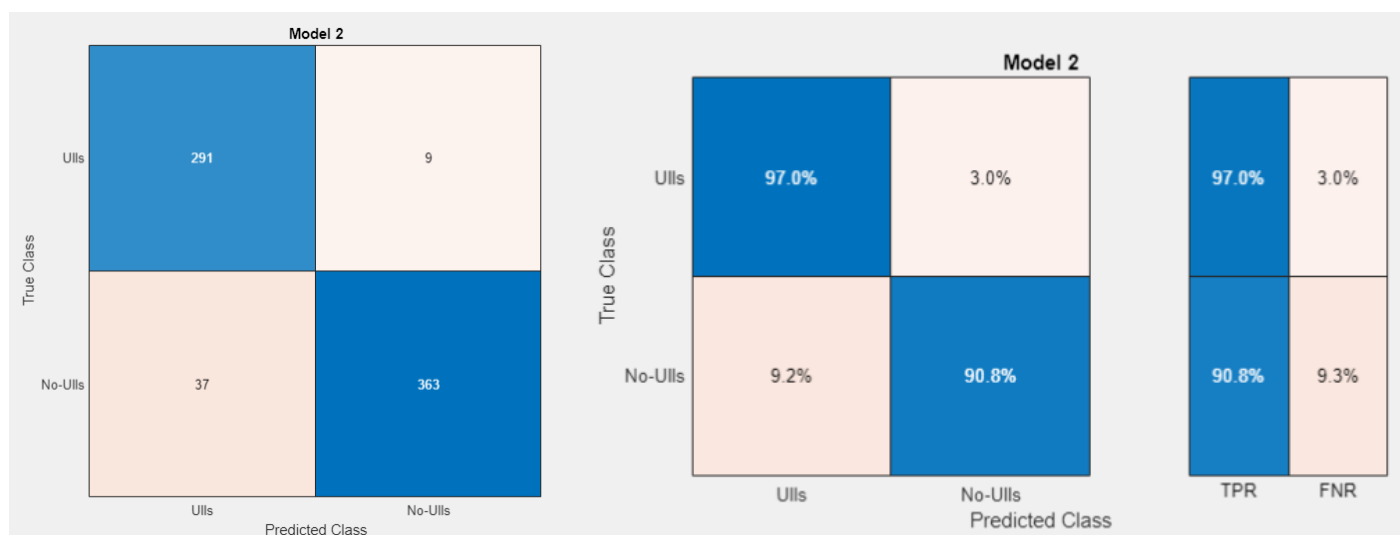
Accuracy = 0,967
Mèrit = 1,368

Sort by: Model Number	
☆ 2 SVM	Accuracy (Validation): 93.4%
Last change: Added feature 'Mitjana 4' 12/48 features	
☆ 3 SVM	Accuracy (Validation): 95.6%
Last change: Removed feature 'GradientMag 12' 24/48 featur	
☆ 4 SVM	Accuracy (Validation): 96.7%
Last change: Removed feature 'GradientMag 12' 36/48 featur	
☆ 5 SVM	Accuracy (Validation): 96.4%
Last change: Linear SVM 48/48 features	

Veiem que els models amb subdivisions tenen un mèrit similar, però el que guanya és el de 6 ja que ofereix el millor balanç entre nombre de descriptors i percentatge d'encerts.

Pel video hem fet servir el model amb 12 subdivisions, per tant analitzarem la seva matriu de confusió, tot i que la resta segueixen una distribució similar. Per a que els resultats estiguin més marcats, hem agafat el pitjor model (el que només té en compte la mitjana).

Podem veure a les figures de sota que les fotos que són ulls tenen un molt alt percentatge de que es classifiquin com a ulls, però el nombre de falsos positius del No-ulls és bastant més elevat que el dels ulls, i per tant molts més no-ulls es marcaran com a ulls que ulls marcats com a no-ull.



3. Enumeració de les funcions utilitzades i treball que realitzen:

-myDescriptorFull: Funció la qual ens retorna un array amb les característiques que volem per a una imatge sense subdivisions. Aquesta calcula la mitjana de nivells de gris, la variància, la desviació estàndard i la suma de la magnitud dels gradients per cada imatge.

-myDescriptor3: Funció la qual ens retorna un array amb les característiques que volem per a una imatge amb 3 subdivisions. Aquesta calcula la mitjana de nivells de gris, la variància, la desviació estàndard i la suma de la magnitud dels gradients per cada subdivisió.

-myDescriptor6: Funció la qual ens retorna un array amb les característiques que volem per a una imatge amb 6 subdivisions. Aquesta calcula la mitjana de nivells de gris, la variància, la desviació estàndard i la suma de la magnitud dels gradients per cada subdivisió.

-myDescriptor12: Funció la qual ens retorna un array amb les característiques que volem per a una imatge amb 12 subdivisions. Aquesta calcula la mitjana de nivells de gris, la variància, la desviació estàndard i la suma de la magnitud dels gradients per cada subdivisió.

També trobem les funcions trainClassifier les quals són generades automàticament pel matlab les quals realitzen la implementació dels diferents models.

4. Annex del codi amb totes les seves funcions

```
myDescriptor6.m  myDescriptor3.m  myDescriptorFull.m  myDescriptor12.m  +
1  function [S] = myDescriptorFull(I)
2  I = imresize(I,[48 32]);
3  %% tractem la imatge sencera
4  %%Possibles característiques que podem tenir en compte per tal d'obtenir els
5  %%resultats que volem:
6  %%Mitjana de nivells de grisos
7  mitj = mean(I(:));
8  %%Variancia de la imatge
9  V = double(I);
10 vari = mean(var(V));
11 %%desviació standard
12 std_dev = mean(std(V));
13
14 %%Value and direction of the gradient
15 [Gmag, Gdir] = imgradient(I,'prewitt');
16 sumGradMag = sum(sum(Gmag));
17 S = [mitj, vari,std_dev,sumGradMag];
18 end

myDescriptor6.m  myDescriptor3.m  myDescriptorFull.m  myDescriptor12.m  +
2
3  %% tractem la dividida en 3 blocs
4  %%
5  %% x
6  %% x
7  %% x
8  I = imresize(I,[48 32]);
9  [f c] = size(I);
10 stepy = f/3;
11 S = [];
12
13 for j=1:3
14
15     X = I( ((j-1)*stepy)+1:(j*stepy),:);
16     mitj = mean(X(:));
17     %%Variancia de la imatge
18     V = double(X);
19     vari = mean(var(V));
20
21     %%desviació standard
22     std_dev = mean(std(V));
23
24     %%Value and direction of the gradient
25     [Gmag, Gdir] = imgradient(I,'prewitt');
26     sumGradMag = sum(sum(Gmag));
27     |
28     S = [S,mitj, vari,std_dev,sumGradMag];
29
30 end
31
32 end
```

```

myDescriptor6.m x myDescriptor3.m x myDescriptorFull.m x myDescriptor12.m x +
2
3 %% tractem la dividida en 6 blocs
4 %%
5 %% x x
6 %% x x
7 %% x x
8 %%
9 I = imresize(I,[48 32]);
10 [f c] = size(I);
11 stepx = c/2;
12 stepy = f/3;
13 S = [];
14 for i = 1:2
15     for j=1:3
16
17         X = I( ((j-1)*stepy)+1:(j*stepy),((i-1)*stepx)+1:(i*stepx));
18         mitj = mean(X(:));
19         %Variancia de la imatge
20         V = double(X);
21         vari = mean(var(V));
22
23         %desviacio standard
24         std_dev = mean(std(V));
25
26         %%Value and direction of the gradient
27         [Gmag, Gdir] = imgradient(I,'prewitt');
28         sumGradMag = sum(sum(Gmag));
29
30         S = [S,mitj, vari,std_dev,sumGradMag];
31
32     end

```

```

myDescriptor6.m x myDescriptor3.m x myDescriptorFull.m x myDescriptor12.m x +
3 %% tractem la dividida en 12 blocs
4 %%
5 %% x x x x
6 %% x x x x
7 %% x x x x
8 %%
9 I = imresize(I,[48 32]);
10 [f c] = size(I);
11 stepx = c/4;
12 stepy = f/3;
13 S = [];
14 for i = 1:4
15     for j=1:3
16
17         X = I( ((j-1)*stepy)+1:(j*stepy),((i-1)*stepx)+1:(i*stepx));
18         mitj = mean(X(:));
19         %Variancia de la imatge
20         V = double(X);
21         vari = mean(var(V));
22
23         %desviacio standard
24         std_dev = mean(std(V));
25
26
27         %%Value and direction of the gradient
28         [Gmag, Gdir] = imgradient(I,'prewitt');
29         sumGradMag = sum(sum(Gmag));
30
31         S = [S,mitj, vari,std_dev,sumGradMag];
32
33     end

```

Short Project

Detector d'ulls

Miquel Gotanegra

Josep Navarro

```
% Primerament procedirem a carregar les imatges de les cares i no cares de la base
% de dades que hem obtingut mitjançant el codi en Python que tenim adjuntat
% al document explicatiu del ShortProject.

% Fem un clear per esborrar tot el que tinguem al workspace
clear

% Llegim les imatges de les cares i procedim a fer un redimensionament a més
% de convertir la imatge a 256 nivells de gris
list = dir('Cares\*.jpg');
FilesUlls = length(list);
% Creem un conjunt de les imatges de la carpeta
Ulls = cell(1,length(list));

%% seleccionem només els ulls, ja que si no fariem un detector de cares, en
%% relació 4:3
mida = 50;
dimx = mida*4;
dimy = mida*3;
eyeCenter = [381 476];
for i=1:FilesUlls
    image_file=imread(list(i).name);
    I = rgb2gray(image_file);
    I = I(eyeCenter(2)-dimy/2:eyeCenter(2)+dimy/2,eyeCenter(1)-dimy/2:eyeCenter(1)+dimx/2);
    Ulls{i}=I;
end
list = dir('No_ulls\*.jpg');
FilesNoUlls = length(list);
No_ulls = cell(1,length(list));
%% guardem un fragment aleatori de la mateixa mida que el ull
for j=1:FilesNoUlls
    image_file=imread(list(j).name);
    I = rgb2gray(image_file);
    win = randomWindow2d(size(I),[dimy,dimx]);
    I = imcrop(I,win);
    No_ulls{j}=I;
end
```



```
%%Podem dividir la imatge en blocs seguint la idea del LBP i agafem les característiques de ca  
%% la imatge esta en 3:2, per tant probarem amb 3,6 i 12 blocs
```

```
%% per al myDescriptorFull, sense dividir la imatge
```

```
DescriptorTypes = {'Mitjana','Variança','Desviacio','GradientMag','Deteccio'};
```

```
%% per al myDescriptor3
```

```
DescriptorType3 = {'Mitjana 1','Variança 1','Desviacio 1','GradientMag 1',...  
                  'Mitjana 2','Variança 2','Desviacio 2','GradientMag 2',...  
                  'Mitjana 3','Variança 3','Desviacio 3','GradientMag 3',...  
                  'Deteccio'};
```

```
%% per al myDescriptor6
```

```
DescriptorType6 = cat(2,DescriptorType3(1:end-1),{  
                  'Mitjana 4','Variança 4','Desviacio 4','GradientMag 4',...  
                  'Mitjana 5','Variança 5','Desviacio 5','GradientMag 5',...  
                  'Mitjana 6','Variança 6','Desviacio 6','GradientMag 6',...  
                  'Deteccio'});
```

```
DescriptorType12 = cat(2,DescriptorType6(1:end-1),{  
                  'Mitjana 7','Variança 7','Desviacio 7','GradientMag 7',...  
                  'Mitjana 8','Variança 8','Desviacio 8','GradientMag 8',...  
                  'Mitjana 9','Variança 9','Desviacio 9','GradientMag 9',...  
                  'Mitjana 10','Variança 10','Desviacio 10','GradientMag 10',...  
                  'Mitjana 11','Variança 11','Desviacio 11','GradientMag 11',...  
                  'Mitjana 12','Variança 12','Desviacio 12','GradientMag 12',...  
                  'Deteccio'});
```

```
%%S'ha de fer fora del bucle porque si inicialitzem una taula buida tots
```

```
%%els elements estan com a doubles
```

```
DescriptorFull = array2table(myDescriptorFull(Ulls{1}));
```

```
%posem categorical per a que nomes hi hagi dues opcions a l'hora de fer la predicció,  
%ulls o no-ulls; si posesim un string normal n'hi hauria infinites
```

```
DescriptorFull.Deteccio = categorical({'Ulls'});
```

```
DescriptorFull.Properties.VariableNames = DescriptorTypes;
```

```
for i=2:FilesUlls
```

```
    fila = array2table(myDescriptorFull(Ulls{i}));
```

```
    fila.Deteccio = categorical({'Ulls'});
```

```
    fila.Properties.VariableNames = DescriptorTypes;
```

```
    DescriptorFull = [DescriptorFull;fila];
```

```
end
```

```
for i = 1:FilesNoUlls
```

```
    fila = array2table(myDescriptorFull(No_ulls{i}));
```

```
    fila.Deteccio = categorical({'No-Ulls'});
```

```

    fila.Properties.VariableNames = DescriptorTypes;
    DescriptorFull = [DescriptorFull;fila];
end
DescriptorFull;
writetable(DescriptorFull,'DescriptorFull.csv','Delimiter',' ','QuoteStrings',true)
[trainedClassifierFull, validationAccuracyFull] = trainClassifierFull(DescriptorFull);
validationAccuracyFull

```

```
validationAccuracyFull = 0.7343
```

```
MeritFull = sqrt( (1-size(DescriptorFull,2)/(48*32))^2 + validationAccuracyFull^2)
```

```
MeritFull = 1.2380
```

```

Descriptor3 = array2table(myDescriptor3(Ulls{1}));
Descriptor3.Deteccio = categorical({'Ulls'});
Descriptor3.Properties.VariableNames = DescriptorType3;

for i=2:FilesUlls
    fila = array2table(myDescriptor3(Ulls{i}));
    fila.Deteccio = categorical({'Ulls'});
    fila.Properties.VariableNames = DescriptorType3;
    Descriptor3 = [Descriptor3;fila];

end
for i = 1:FilesNoUlls
    fila = array2table(myDescriptor3(No_ulls{i}));
    fila.Deteccio = categorical({'No-Ulls'});
    fila.Properties.VariableNames = DescriptorType3;
    Descriptor3 = [Descriptor3;fila];
end
Descriptor3;
writetable(Descriptor3,'Descriptor3.csv','Delimiter',' ','QuoteStrings',true)
[trainedClassifier3, validationAccuracy3] = trainClassifier3(Descriptor3);
validationAccuracy3

```

```
validationAccuracy3 = 0.9600
```

```
Merit3 = sqrt( (1-size(Descriptor3,2)/(48*32))^2 + validationAccuracy3^2)
```

```
Merit3 = 1.3801
```

```

Descriptor6 = array2table(myDescriptor6(Ulls{1}));
Descriptor6.Deteccio = categorical({'Ulls'});
Descriptor6.Properties.VariableNames = DescriptorType6;

for i=2:FilesUlls
    fila = array2table(myDescriptor6(Ulls{i}));
    fila.Deteccio = categorical({'Ulls'});
    fila.Properties.VariableNames = DescriptorType6;

```

```

Descriptor6 = [Descriptor6;fila];

end
for i = 1:FilesNoUlls
    fila = array2table(myDescriptor6(No_ulls{i}));
    fila.Deteccio = categorical({'No-Ulls'});
    fila.Properties.VariableNames = DescriptorType6;
    Descriptor6 = [Descriptor6;fila];
end
Descriptor6;
writetable(Descriptor6, 'Descriptor6.csv', 'Delimiter', ',', 'QuoteStrings', true)
[trainedClassifier6, validationAccuracy6] = trainClassifier6(Descriptor6);
validationAccuracy6

```

```
validationAccuracy6 = 0.9714
```

```
Merit6 = sqrt( (1-size(Descriptor6,2)/(48*32))^2 + validationAccuracy6^2)
```

```
Merit6 = 1.3825
```

```

Descriptor12 = array2table(myDescriptor12(Ulls{1}));
Descriptor12.Deteccio = categorical({'Ulls'});
Descriptor12.Properties.VariableNames = DescriptorType12;

for i=2:FilesUlls
    fila = array2table(myDescriptor12(Ulls{i}));
    fila.Deteccio = categorical({'Ulls'});
    fila.Properties.VariableNames = DescriptorType12;
    Descriptor12 = [Descriptor12;fila];

end
for i = 1:FilesNoUlls
    fila = array2table(myDescriptor12(No_ulls{i}));
    fila.Deteccio = categorical({'No-Ulls'});
    fila.Properties.VariableNames = DescriptorType12;
    Descriptor12 = [Descriptor12;fila];
end
Descriptor12;
writetable(Descriptor12, 'Descriptor12.csv', 'Delimiter', ',', 'QuoteStrings', true)
%%ens quedem amb 12 blocs i tots els descriptors, que es com ens dona millors resultats
[trainedClassifier12, validationAccuracy12] = trainClassifier12(Descriptor12);
validationAccuracy12

```

```
validationAccuracy12 = 0.9757
```

```
Merit12 = sqrt( (1-size(Descriptor12,2)/(48*32))^2 + validationAccuracy12^2)
```

```
Merit12 = 1.3745
```

```
C =imread('mj.png');
```

```
I = rgb2gray(C);  
[x y] = size(I)
```

```
x = 362  
y = 364
```

```
%% fem les proves amb un frame concret del video per calcular mes o menys la posicio dels ulls  
imshow(I);
```



```
stepy = round(y/6)
```

```
stepy = 61
```

```
stepx = round(x/8)
```

```
stepx = 45
```

```
UllDret = I(2*stepy:stepy*3,stepx:stepx*4);
```



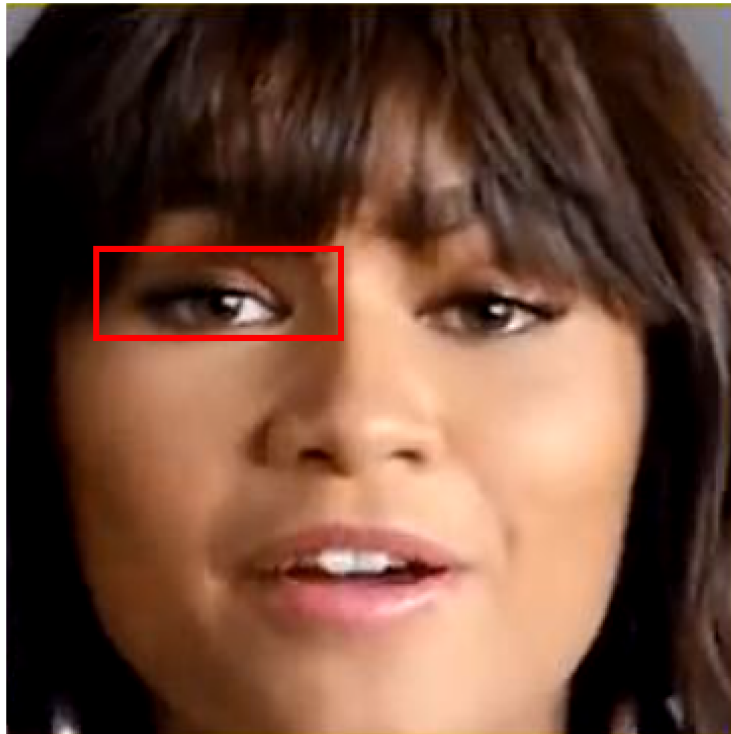
```
U11Esquerre = I(2*stepy:stepy*3,stepx*4:stepx*7);  
imshow(U11Dret)  
imshow(U11Esquerre)
```



```

fila = array2table(myDescriptor12(U11Dret));
fila.Properties.VariableNames = DescriptorType12(1:end-1);
if trainedClassifier12.predictFcn(fila) == categorical({'U11s'})
    K = insertShape(C, 'Rectangle', [stepx, 2*stepy, stepy*2, stepx], 'Color', 'red', 'LineWidth', 3);
end
imshow(K)

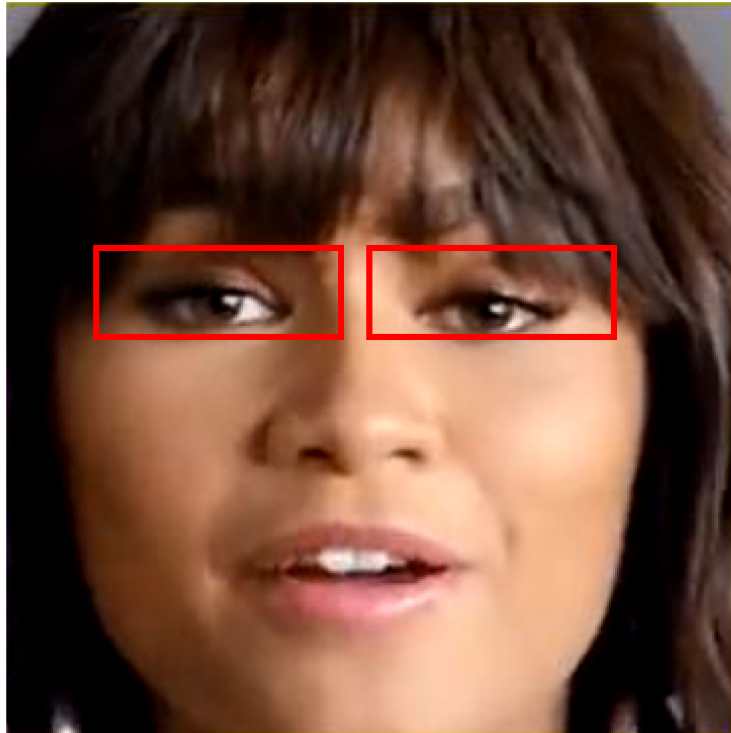
```



```

fila = array2table(myDescriptor12(U11Esquerre));
fila.Properties.VariableNames = DescriptorType12(1:end-1);
if trainedClassifier12.predictFcn(fila) == categorical({'U11s'})
    K = insertShape(K, 'Rectangle', [4*stepx, 2*stepy, stepy*2, stepx], 'Color', 'red', 'LineWidth', 3);
end
imshow(K)

```



```
% Create a cascade detector object.

Detector = vision.CascadeObjectDetector('FrontalFaceLBP');
delete 'edit.avi'
edit = VideoWriter('edit.avi');
open(edit)
% Read a video frame and run the face detector.

videoReader = VideoReader('Zendaya.mp4');

while hasFrame(videoReader)

    % get the next frame

    videoFrame = readFrame(videoReader);

    bbox = step(Detector, videoFrame);
    % Draw the returned bounding box around the detected face.
    videoFrame = insertShape(videoFrame, 'Rectangle', bbox);

    for i = 1:size(bbox,1)
        caja = bbox(i,:);
        y = caja(1);
```

```

x = caja(2);
stepy = caja(3);
stepx = caja(4);
C = videoFrame(x:x+stepx,y:y+stepy,:);
I = rgb2gray(C);

[f c] = size(I);
stepy = round(c/6);
stepx = round(f/8);

%%pel cas concret de la Zendaya, les deteccions son a k=2, pero
%%ensem que buscant els ulls en un espai vertical mes ampli ens
%%ajuda a abarcar mes tipus de cares diferents (mirem a la part superior de la imatge
for k=1:3
    UllDret = I(k*stepy:stepy*(k+1),stepx:stepx*4);
    UllEsquerre = I(k*stepy:stepy*(k+1),stepx*4:stepx*7);
    fila = array2table(myDescriptor12(UllDret));
    fila.Properties.VariableNames = DescriptorType12(1:end-1);
    if trainedClassifier12.predictFcn(fila) == categorical({'Ulls'})
        videoFrame = insertShape(videoFrame,'Rectangle',[caja(1) + stepx, caja(2) + 2*stepx,
    end

    fila = array2table(myDescriptor12(UllEsquerre));
    fila.Properties.VariableNames = DescriptorType12(1:end-1);
    if trainedClassifier12.predictFcn(fila) == categorical({'Ulls'})
        videoFrame = insertShape(videoFrame,'Rectangle',[caja(1) + 4*stepx,caja(2)+2*stepx,
    end
end

end
writeVideo(edit,videoFrame)
end
close(edit)

```