Universitat Politècnica de Catalunya

Visió per Computador (VC)

Grau en Enginyeria Informàtica

Short Project: Detector d'ulls i mirades

Autors:
Jordi Armengol Estapé,
Aleix Vaz Vidal

Professor:
Manel Frigola Bourlon

16 de gener
Primer quadrimestre, curs 2018-2019

Contents

1 Introducció **1**

1.1 Descripció del treball 1

1.2 Resultats 1

1.3 Descripció de les dades 1

1.4 Notes respecte el codi 1

2 Generació de les dades, extracció de característiques i remostreig **2**

2.1 Generació de les dades (`genData.m`) 2

2.2 Justificació de les característiques utilitzades 3

2.3 Protocol de remostreig (`genSets.m`) 3

3 Aprenentatge **4**

3.1 Procediment general (`train.m`) 4

3.2 Sintonització de paràmetres 4

3.3 Resultats 4

3.4 Anàlisi dels resultats 11

4 Avaluació **11**

4.1 Resultats amb test (`evaluate.m`) 11

5 Aplicació **12**

5.1 Aplicació amb finestra lliscant (`app.m`) 12

5.2 Extra: aplicació interactiva (`app_interact.m`) 13

6 Implementacions pròpies **14**

6.1 Detalls d'implementació i codis propis 14

7 Conclusions **15**

1 Introducció

1.1 Descripció del treball

Aquest document consisteix en l'informe escrit del Short Project de l'assignatura de Visió per Computador. Partint d'una base de dades amb un conjunt d'imatges de cares (en nivell de gris de 384 x 286 pixels) extretes de la web de BioID, la tasca tractava d'implementar un sistema automàtic per detectar els ulls i la mirada a càmera mitjançant un predictor.

Les etapes i els objectius del projetce eren els següents:

- Construir un detector d'ulls en base a les imatges, utilitzant les característiques HOG i altres descriptors d'imatges de ndg.
- Realitzar un estudi estadístic de les prestacions del classificador (matriu de confusió, rellevància de les característiques, etc.) amb i sense les característiques HOG.
- Escollir un conjunt de característiques per classificar si un ull mira a la càmera i realitzar aquest classificador, fent una estadística dels resultats.
- Provar el rendiment de la predicció utlitzant una finestra de detecció lliscant en imatges reals.

1.2 Resultats

Pel que fa als resultats, anticipem que els resultats amb les imatges del mateix dataset (encara que no hagin estat vistes durant l'entrenament) són molt bones, però tenim dubtes de la capacitat de generalització del predictor amb imatges fora del dataset (tot i haver tingut un cert èxit en les proves que hem fet)

1.3 Descripció de les dades

La nostra base de dades, referida com a *BioID Face Database*, ha estat obtinguda de la pàgina web del software de reconeixement facial BioID. Consisteix en 1.521 imatges que mostren la vista frontal d'una cara de 23 persones de prova diferents. Per raons de comparació, el conjunt també conté posicions visuals fixades manualment. Les imatges s'anomenen "BioID_xxxx.pgm", on els caràcters xxxx se substitueixen per l'índex de la imatge actual. De la mateixa manera, els fitxers "BioID_xxxx.eye" contenen les posicions dels ulls les imatges corresponents. Així mateix, es disposa d'un fitxer Excel que indica, per cada imatge, si la persona està mirant o no.

1.4 Notes respecte el codi

Per a executar el codi, es recomana llegir les instruccions i notes del fitxer README. A diferència dels laboratoris, aquest cop hem considerat convenient no incloure fragments de codi en el document, perquè hi ha molt codi organitzat per arxius i directoris i creiem millor que el codi es llegeixi directament des de MATLAB.

2 Generació de les dades, extracció de característiques i remostreig

2.1 Generació de les dades (`genData.m`)

Els algorismes de classificació que utilitzarem (i de fet, la majoria d'algorismes) no estan pensats per rebre imatges directament com entrada. Per tant, haurem de realitzar *feature engineering* per decidir quines característiques extreure de les imatges, i un cop decidit, extreure-les efectivament.

Abans d'aquest procediment, però, haurem de dividir les imatges en les parts que tenen ulls i les que no. Amb la funció *scanFiles(PATH)*, obtenim, per cada imatge, els punts *lx,ly,rx,ry*, que representen la posició dels ulls, el nom de l'imatge *Inames* i si la persona està mirant a la càmera (*looking*).

Prendrem una imatge rectangular (més eixamplada horitzontalment que verticalment), la mida de la qual es decideix programàticament per cada cas a partir de la distància euclidiana entre els dos ulls, deixant suficient marge perquè també entrin les celles. En particular, el rectangle es defineix de la següent manera:

```
rect = [uint8(rx(i) - d*0.35), uint8(ry(i) -  d*0.3), uint8(d + d*0.75), uint8(d*0.5)];
```

on d és la distància euclidiana entre els dos ulls i rx i ry són les coordenades de l'ull dret.

Les constants que apareixen en la fórmula del rectangle les hem trobat per prova i error. Funcionen força bé per totes les imatges del dataset.

Donada una certa proporció de sub-imatges sense ulls (en el nostre cas, el 95%), agafem un cert nombre de sub-imatges de la imatge que no pertanyin al rectangle dels ulls (pel 95%, 19 sub-imatges).

Per tal que totes les sub-imatges tinguin la mateixa mida, s'escalen a un rectangle de 128×32, aplicant prèviament un filtre gaussià per tal de no pixelar excessivament.

Per a totes les imatges, extraiem les característiques (HOG, les proposades per l'enunciat, i LBP (Local Binary Patterns, serà explicat més endavant), les proposades per nosaltres) i les desem en una base de dades en disc (com un objecte de MATLAB, en particular). La base de dades resultant té 1680 columnes i 30420 (el nombre de columnes serà més gran a més gran sigui la mida del rectangle, en el nostre cas ja hem dit que és de 128×32). Òbviament l'última columna serà l'etiqueta, posada pel nostre codi segons si la sub-imatge provenia del rectangle dels ulls o no.

Pel que fa a la detecció de mirada, hem portat a terme el mateix procediment, però només amb les sub-imatges que continguin ulls, i extraient les etiquetes de l'Excel. Així doncs, aquesta segona base de dades comptarà amb 1521 files i el mateix nombre de columnes.

Observem que la base de dades deels ulls està molt desbalancejada, però a canvi té un gran nombre de files, mentre que de les mirades tenim moltes menys files però a canvi estan més balancejades entre les que estan mirant i les que no.

2.2 Justificació de les característiques utilitzades

Les característiques HOG, les proposades per l'enunciat, acostumen a donar bons resultats pel que fa a la detecció de cares. D'una sola imatgeja se'n generen moltíssimes.

Pel que fa a LBP, els nostres motius per escollir-los han estat:

- A diferència dels laboratoris (de fet, en un primer moment havíem pensat en utilitzar la nostra funció d'obtenir característiques del laboratori de reconeixement), en aquest cas necessitem característiques extretes a partir d'imatges en nivells de grisos, no binaritzades. No és viable binaritzar una cara, encara que es tracti morfològicament i s'utilitzin tècniques de binaritzat local. Així, totes les característiques geomètriques que havíem vist són descartades perquè l'entrada hauria de ser una imatge binaritzada. LBP treballa directament amb imatges en escales de grisos.
- Per com funciona, acostuma a funcionar bé per detectar textures.
- El resultat, un seguit de nombres (ja normalitzats), es pot donar d'entrada directament a un SVM.
- Hem fet recerca i hem vist que aquestes característiques donen bons resultats per la detecció de cares i similars (per exemple, millor que SWIFT).
- Per una sola sub-imatge, dóna força característiques, encara que molt menys que HOG. Que doni suficients característiques és un punt a favor. Que en doni moltes menys que HOG no queda clar d'entrada. Podria ser que donés menys informació, o podria ser que donés menys soroll. Ho veurem més endavant.
- Per altra banda, cal tenir en compte la viabilitat de la implementació. MATLAB ja proveeix una funció per extreure aquestes característiques, de la mateixa manera que per HOG. Hem preferit dedicar els nostres esforços a altres parts de la implementació.

Per veure quin subconjunt (o si les escollim totes) de les dades utilitzem, ens referim a la secció d'aprenentage, més endavant en aquest document.

2.3 Protocol de remostreig (`genSets.m`)

Com és gairebé estàndard en problemes d'aprenentatge computacional, remostrejarem les dades de la següent manera:

- $\frac{2}{3}$ de les dades per a aprenentatge.
- $\frac{1}{3}$ de les dades per a test.

A la seva vegada, la part d'aprenentatge, serà remostrejada iterativament mitjançant cross validation. D'aquesta manera, optimitzarem un paràmetre del predictor i decidirem quins són els millors models. La partició de test servirà per a donar una estimació el més honesta i realista possible de l'error de generalització del predictor final.

Cal destacar que el codi per dividir aleatòriament el dataset en dues particions ha estat implementat per nosaltres.

3 Aprenentatge

3.1 Procediment general (`train.m`)

Amb la partició d'aprenentatge (tant pels ulls com per les mirades), portarem a terme un K-fold cross validation fent cerca dels millors paràmetres i ponderant l'error. Així. seleccionarem el millor model per cada cas (detector d'ulls i de mirades). Provarem SVMs amb només HOG, només LBP o les dues combinades.

3.2 Sintonització de paràmetres

Optimitzarem el paràmetre `BoxConstraint` provant-ne diferents valors i quedant-nos amb el que doni millors resultats. Per a decidir quin és el millor, farem K-fold cross validation, és a dir, ho provarem repetint el procés, donada una certa partició, canviant quina part s'utilitza per entrenament i quina per avaluació.

Ens hagués agradat optimitzar més paràmetres (provar més kernels, a part del lineal; provar diferents valors per KernelScale...) però per costos computacionals no ha estat gaire viable en el nostre cas.

3.3 Resultats

A les següents pàgines observem els resultats obtinguts:

Iter	Active workers	Eval result	Objective	Objective runtime	BestSoFar (observed)	BestSoFar (estim.)	BoxConstraint
1	1	Best	0.00028177	20.909	0.00028177	0.00028177	19.776
2	1	Accept	0.00028177	20.731	0.00028177	0.00028177	0.56616
3	2	Best	0.00018785	17.69	0.00018785	0.00025045	0.1541
4	2	Accept	0.00028177	19.341	0.00018785	0.00025824	12.333
5	2	Accept	0.00051658	46.358	0.00018785	0.00018787	0.0042767
6	2	Accept	0.00018785	16.781	0.00018785	0.0001878	0.15274
7	2	Accept	0.00023481	16.307	0.00018785	0.00018775	0.086508
8	2	Accept	0.00028177	15.798	0.00018785	0.00018775	479.83
9	2	Accept	0.00023481	15.541	0.00018785	0.00018749	0.20236
10	2	Accept	0.00028177	15.292	0.00018785	0.00018749	119.05
11	2	Accept	0.00018785	16.191	0.00018785	0.00018759	0.12932
12	2	Accept	0.00028177	15.168	0.00018785	0.00018759	2.2493
13	2	Accept	0.00028177	16.112	0.00018785	0.0001876	999.19
14	2	Accept	0.00028177	16.41	0.00018785	0.0002022	51.558
15	2	Accept	0.00028177	16.375	0.00018785	0.00020245	5.0119
16	2	Accept	0.00018785	25.693	0.00018785	0.00018756	0.022705
17	2	Accept	0.00023481	23.731	0.00018785	0.00018757	0.032085
18	2	Accept	0.00018785	28.216	0.00018785	0.00018759	0.018016
19	2	Accept	0.00070442	94.273	0.00018785	0.00018776	0.0010011
20	2	Accept	0.00028177	17.869	0.00018785	0.00018777	1.1039

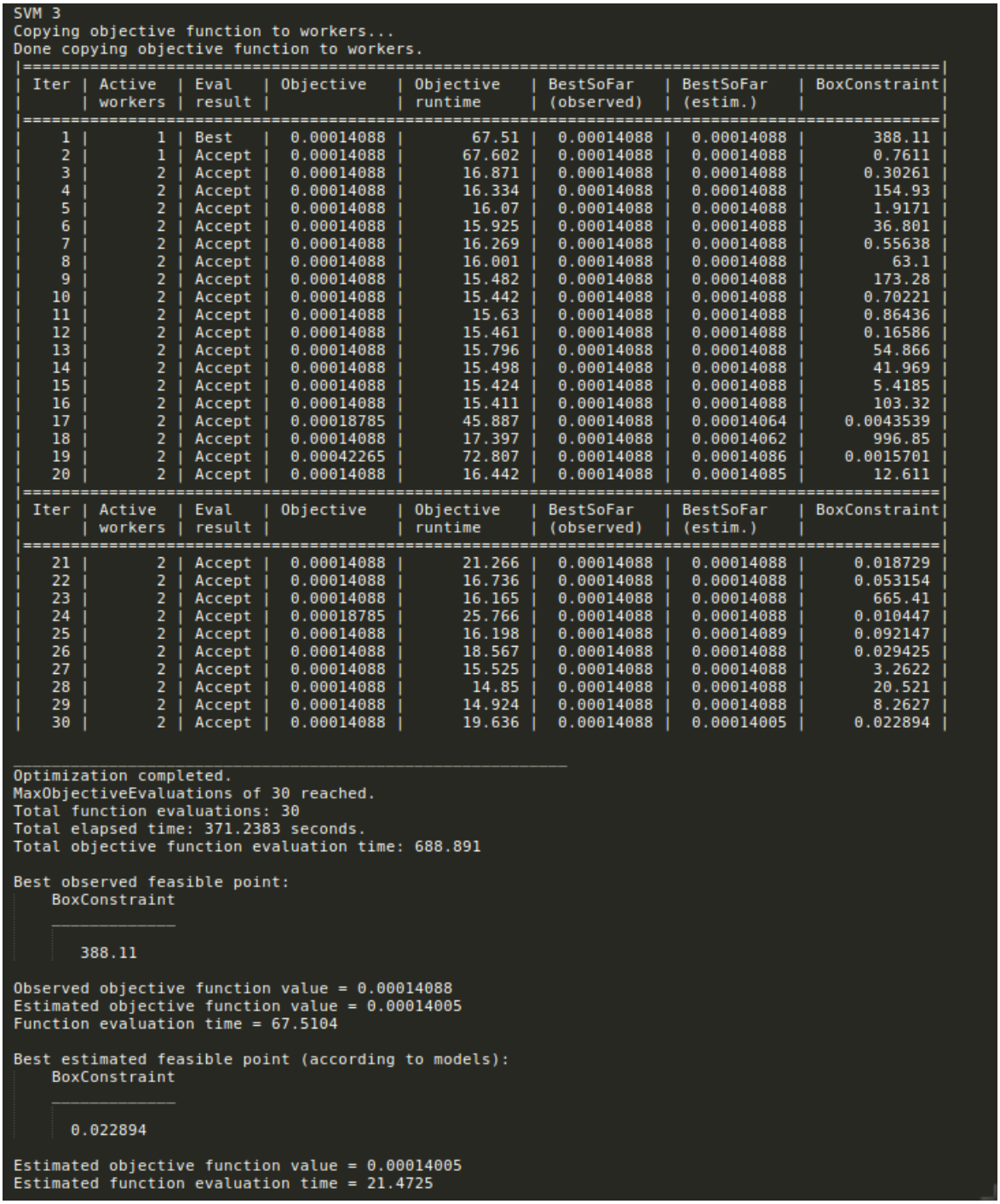
Iter	Active workers	Eval result	Objective	Objective runtime	BestSoFar (observed)	BestSoFar (estim.)	BoxConstraint
21	2	Accept	0.00018785	25.387	0.00018785	0.00018615	0.020138
22	2	Accept	0.00028177	15.457	0.00018785	0.00018604	237.89
23	2	Accept	0.00018785	15.256	0.00018785	0.00018595	0.14015
24	2	Accept	0.00018785	16.931	0.00018785	0.00018644	0.13902
25	2	Accept	0.00018785	23.592	0.00018785	0.00018653	0.020181
26	2	Accept	0.00018785	17.317	0.00018785	0.00018705	0.14144
27	2	Accept	0.00042265	32.343	0.00018785	0.00018731	0.008735
28	2	Accept	0.00028177	15.422	0.00018785	0.00018728	33.026
29	2	Accept	0.00018785	15.437	0.00018785	0.00018738	0.14431
30	2	Accept	0.00018785	23.271	0.00018785	0.00018739	0.019358

```
Optimization completed.
MaxObjectiveEvaluations of 30 reached.
Total function evaluations: 30
Total elapsed time: 365.4776 seconds.
Total objective function evaluation time: 675.2005

Best observed feasible point:
    BoxConstraint
    _____________

       0.1541

Observed objective function value = 0.00018785
Estimated objective function value = 0.00018739
Function evaluation time = 17.6903

Best estimated feasible point (according to models):
    BoxConstraint
    _____________

       0.14431

Estimated objective function value = 0.00018739
Estimated function evaluation time = 16.3785
```

Figure 1: Detector d'ulls: només HOG.

```
SVM 2
Copying objective function to workers...
Done copying objective function to workers.
|=====================================================================================================|
| Iter | Active   | Eval   | Objective   | Objective    | BestSoFar    | BestSoFar    | BoxConstraint|
|      | workers  | result |             | runtime      | (observed)   | (estim.)     |              |
|=====================================================================================================|
|    1 |       2  | Best   |  0.00014088 |       3.2687 |  0.00014088  |  0.00014088  |      0.14706 |
|    2 |       2  | Accept |  0.00018785 |       2.7156 |  0.00014088  |  0.00014629  |       11.964 |
|    3 |       2  | Accept |   0.0018785 |        9.768 |  0.00014088  |  0.00073566  |    0.0021345 |
|    4 |       2  | Accept |  0.00018785 |       2.6976 |  0.00014088  |  0.00014103  |       19.783 |
|    5 |       2  | Accept |  0.00018785 |        2.602 |  0.00014088  |  0.00014112  |        3.937 |
|    6 |       2  | Accept |  0.00018785 |       2.7343 |  0.00014088  |  0.00014114  |       999.95 |
|    7 |       2  | Accept |  0.00018785 |       2.7295 |  0.00014088  |  0.00014141  |        0.484 |
|    8 |       2  | Accept |  0.00018785 |       2.7801 |  0.00014088  |  0.00014153  |       209.38 |
|    9 |       2  | Accept |  0.00014088 |       2.8877 |  0.00014088  |  0.00014109  |       0.2046 |
|   10 |       2  | Accept |  0.00014088 |         2.92 |  0.00014088  |  0.00013972  |      0.17813 |
|   11 |       2  | Accept |  0.00014088 |       2.9832 |  0.00014088  |  0.00013996  |      0.17297 |
|   12 |       2  | Accept |  0.00014088 |       2.9862 |  0.00014088  |  0.00014015  |      0.17224 |
|   13 |       2  | Accept |  0.00018785 |       2.7608 |  0.00014088  |  0.00014019  |       73.846 |
|   14 |       2  | Best   |  9.3923e-05 |       4.2826 |  9.3923e-05  |  9.4231e-05  |     0.023996 |
|   15 |       2  | Accept |  0.00018785 |       2.7044 |  9.3923e-05  |  9.4266e-05  |       522.97 |
|   16 |       2  | Accept |  0.00014088 |       3.6385 |  9.3923e-05  |  9.4689e-05  |     0.041599 |
|   17 |       2  | Accept |  0.00018785 |       2.7828 |  9.3923e-05  |  9.4747e-05  |        1.477 |
|   18 |       2  | Accept |  9.3923e-05 |       4.5222 |  9.3923e-05  |  9.4319e-05  |     0.024119 |
|   19 |       2  | Accept |  9.3923e-05 |       4.8256 |  9.3923e-05  |  9.3847e-05  |     0.021956 |
|   20 |       2  | Accept |  9.3923e-05 |        4.805 |  9.3923e-05  |  9.3707e-05  |     0.024712 |
|=====================================================================================================|
| Iter | Active   | Eval   | Objective   | Objective    | BestSoFar    | BestSoFar    | BoxConstraint|
|      | workers  | result |             | runtime      | (observed)   | (estim.)     |              |
|=====================================================================================================|
|   21 |       2  | Accept |  9.3923e-05 |       4.7594 |  9.3923e-05  |  9.3719e-05  |     0.022608 |
|   22 |       2  | Accept |  0.00018785 |       2.6668 |  9.3923e-05  |  9.3781e-05  |       39.729 |
|   23 |       2  | Accept |  0.00028177 |       6.5717 |  9.3923e-05  |  9.0608e-05  |    0.0071954 |
|   24 |       2  | Accept |  0.00018785 |       2.6173 |  9.3923e-05  |   9.055e-05  |      0.87187 |
|   25 |       2  | Accept |  0.00014088 |       5.0315 |  9.3923e-05  |  9.3298e-05  |     0.015539 |
|   26 |       2  | Accept |  0.00014088 |       4.7279 |  9.3923e-05  |  9.1637e-05  |     0.018008 |
|   27 |       2  | Accept |  9.3923e-05 |       4.3401 |  9.3923e-05  |  9.2283e-05  |     0.025059 |
|   28 |       2  | Accept |  0.00014088 |       3.9999 |  9.3923e-05  |  0.00010383  |     0.027261 |
|   29 |       2  | Accept |  9.3923e-05 |       4.3483 |  9.3923e-05  |  0.00010215  |     0.026586 |
|   30 |       2  | Accept |  0.00018785 |       2.6269 |  9.3923e-05  |  0.00010246  |       6.6516 |

__________________________________________________________
Optimization completed.
MaxObjectiveEvaluations of 30 reached.
Total function evaluations: 30
Total elapsed time: 73.0483 seconds.
Total objective function evaluation time: 113.0846

Best observed feasible point:
    BoxConstraint
    _____________

      0.023996

Observed objective function value = 9.3923e-05
Estimated objective function value = 0.00010246
Function evaluation time = 4.2826

Best estimated feasible point (according to models):
    BoxConstraint
    _____________

      0.024119

Estimated objective function value = 0.00010246
Estimated function evaluation time = 4.4631
```

Figure 2: Detector d'ulls: només LBP.



```
SVM 3
Copying objective function to workers...
Done copying objective function to workers.
|=====================================================================================================|
| Iter | Active   | Eval   | Objective    | Objective    | BestSoFar    | BestSoFar    | BoxConstraint|
|      | workers  | result |              | runtime      | (observed)   | (estim.)     |              |
|=====================================================================================================|
|    1 |       1  | Best   |  0.00014088  |       67.51  |  0.00014088  |  0.00014088  |       388.11 |
|    2 |       1  | Accept |  0.00014088  |      67.602  |  0.00014088  |  0.00014088  |       0.7611 |
|    3 |       2  | Accept |  0.00014088  |      16.871  |  0.00014088  |  0.00014088  |      0.30261 |
|    4 |       2  | Accept |  0.00014088  |      16.334  |  0.00014088  |  0.00014088  |       154.93 |
|    5 |       2  | Accept |  0.00014088  |       16.07  |  0.00014088  |  0.00014088  |       1.9171 |
|    6 |       2  | Accept |  0.00014088  |      15.925  |  0.00014088  |  0.00014088  |       36.801 |
|    7 |       2  | Accept |  0.00014088  |      16.269  |  0.00014088  |  0.00014088  |      0.55638 |
|    8 |       2  | Accept |  0.00014088  |      16.001  |  0.00014088  |  0.00014088  |         63.1 |
|    9 |       2  | Accept |  0.00014088  |      15.482  |  0.00014088  |  0.00014088  |       173.28 |
|   10 |       2  | Accept |  0.00014088  |      15.442  |  0.00014088  |  0.00014088  |      0.70221 |
|   11 |       2  | Accept |  0.00014088  |       15.63  |  0.00014088  |  0.00014088  |      0.86436 |
|   12 |       2  | Accept |  0.00014088  |      15.461  |  0.00014088  |  0.00014088  |      0.16586 |
|   13 |       2  | Accept |  0.00014088  |      15.796  |  0.00014088  |  0.00014088  |       54.866 |
|   14 |       2  | Accept |  0.00014088  |      15.498  |  0.00014088  |  0.00014088  |       41.969 |
|   15 |       2  | Accept |  0.00014088  |      15.424  |  0.00014088  |  0.00014088  |       5.4185 |
|   16 |       2  | Accept |  0.00014088  |      15.411  |  0.00014088  |  0.00014088  |       103.32 |
|   17 |       2  | Accept |  0.00018785  |      45.887  |  0.00014088  |  0.00014064  |    0.0043539 |
|   18 |       2  | Accept |  0.00014088  |      17.397  |  0.00014088  |  0.00014062  |       996.85 |
|   19 |       2  | Accept |  0.00042265  |      72.807  |  0.00014088  |  0.00014086  |    0.0015701 |
|   20 |       2  | Accept |  0.00014088  |      16.442  |  0.00014088  |  0.00014085  |       12.611 |
|=====================================================================================================|
| Iter | Active   | Eval   | Objective    | Objective    | BestSoFar    | BestSoFar    | BoxConstraint|
|      | workers  | result |              | runtime      | (observed)   | (estim.)     |              |
|=====================================================================================================|
|   21 |       2  | Accept |  0.00014088  |      21.266  |  0.00014088  |  0.00014088  |      0.018729 |
|   22 |       2  | Accept |  0.00014088  |      16.736  |  0.00014088  |  0.00014088  |      0.053154 |
|   23 |       2  | Accept |  0.00014088  |      16.165  |  0.00014088  |  0.00014088  |       665.41 |
|   24 |       2  | Accept |  0.00018785  |      25.766  |  0.00014088  |  0.00014088  |      0.010447 |
|   25 |       2  | Accept |  0.00014088  |      16.198  |  0.00014088  |  0.00014089  |      0.092147 |
|   26 |       2  | Accept |  0.00014088  |      18.567  |  0.00014088  |  0.00014088  |      0.029425 |
|   27 |       2  | Accept |  0.00014088  |      15.525  |  0.00014088  |  0.00014088  |       3.2622 |
|   28 |       2  | Accept |  0.00014088  |       14.85  |  0.00014088  |  0.00014088  |       20.521 |
|   29 |       2  | Accept |  0.00014088  |      14.924  |  0.00014088  |  0.00014088  |       8.2627 |
|   30 |       2  | Accept |  0.00014088  |      19.636  |  0.00014088  |  0.00014005  |      0.022894 |

__________________________________________________________
Optimization completed.
MaxObjectiveEvaluations of 30 reached.
Total function evaluations: 30
Total elapsed time: 371.2383 seconds.
Total objective function evaluation time: 688.891

Best observed feasible point:
    BoxConstraint
    _____________

      388.11

Observed objective function value = 0.00014088
Estimated objective function value = 0.00014005
Function evaluation time = 67.5104

Best estimated feasible point (according to models):
    BoxConstraint
    _____________

     0.022894

Estimated objective function value = 0.00014005
Estimated function evaluation time = 21.4725
```

Figure 3: Detector d'ulls: HOG i LBP.

```
SVM 4
Copying objective function to workers...
Done copying objective function to workers.
|=====================================================================================================|
| Iter | Active   | Eval   | Objective   | Objective   | BestSoFar   | BestSoFar   | BoxConstraint|
|      | workers  | result |             | runtime     | (observed)  | (estim.)    |              |
|=====================================================================================================|
|    1 |       2  | Best   |     0.18515 |      3.3484 |     0.18515 |     0.18515 |       39.822 |
|    2 |       2  | Accept |     0.40038 |      4.5296 |     0.18515 |     0.19745 |    0.0021264 |
|    3 |       2  | Best   |     0.15508 |      2.7976 |     0.15508 |     0.16775 |      0.28158 |
|    4 |       2  | Accept |     0.18515 |      2.9021 |     0.15508 |     0.15509 |       110.81 |
|    5 |       2  | Best   |     0.15132 |      2.6861 |     0.15132 |     0.15139 |      0.20493 |
|    6 |       2  | Accept |     0.16353 |      2.7986 |     0.15132 |     0.15219 |       1.8549 |
|    7 |       2  | Accept |     0.18515 |      2.8579 |     0.15132 |     0.15212 |       997.82 |
|    8 |       2  | Best   |     0.15038 |      2.7353 |     0.15038 |      0.1517 |      0.46085 |
|    9 |       2  | Accept |     0.15038 |      2.7295 |     0.15038 |     0.15084 |      0.41357 |
|   10 |       2  | Accept |     0.15038 |      2.7592 |     0.15038 |     0.15076 |      0.41342 |
|   11 |       2  | Accept |     0.16353 |      3.6295 |     0.15038 |      0.1499 |     0.032398 |
|   12 |       2  | Accept |     0.15977 |      2.7872 |     0.15038 |     0.15054 |      0.79331 |
|   13 |       2  | Accept |     0.16071 |      3.0009 |     0.15038 |     0.15128 |     0.079762 |
|   14 |       2  | Accept |     0.18515 |      2.9498 |     0.15038 |     0.15149 |       7.1903 |
|   15 |       2  | Accept |     0.15226 |      2.6293 |     0.15038 |     0.15139 |      0.35043 |
|   16 |       2  | Accept |     0.18515 |      2.8743 |     0.15038 |     0.15149 |       383.6  |
|   17 |       2  | Accept |     0.21241 |      3.9435 |     0.15038 |     0.15148 |    0.0095307 |
|   18 |       2  | Accept |     0.18515 |      3.0379 |     0.15038 |     0.15152 |       16.746 |
|   19 |       2  | Accept |     0.15508 |      3.2544 |     0.15038 |     0.15149 |      0.13627 |
|   20 |       2  | Accept |     0.40038 |      4.4071 |     0.15038 |     0.15054 |    0.0010017 |
|=====================================================================================================|
| Iter | Active   | Eval   | Objective   | Objective   | BestSoFar   | BestSoFar   | BoxConstraint|
|      | workers  | result |             | runtime     | (observed)  | (estim.)    |              |
|=====================================================================================================|
|   21 |       2  | Accept |     0.17669 |      2.9409 |     0.15038 |     0.15053 |       3.3385 |
|   22 |       2  | Accept |     0.15226 |       2.767 |     0.15038 |     0.15108 |      0.46353 |
|   23 |       2  | Best   |     0.14944 |      2.7079 |     0.14944 |     0.15074 |       0.4535 |
|   24 |       2  | Accept |     0.18515 |      2.8182 |     0.14944 |     0.15083 |       208.57 |
|   25 |       2  | Accept |     0.18233 |      3.5253 |     0.14944 |     0.15069 |     0.017881 |
|   26 |       2  | Accept |     0.18515 |       2.881 |     0.14944 |     0.15071 |       658.96 |
|   27 |       2  | Accept |     0.18515 |      2.7919 |     0.14944 |     0.15073 |       65.662 |
|   28 |       2  | Accept |     0.16259 |      2.6574 |     0.14944 |     0.15072 |       1.2168 |
|   29 |       2  | Accept |     0.16635 |      3.1623 |     0.14944 |     0.15083 |     0.050548 |
|   30 |       2  | Accept |     0.31955 |      4.3843 |     0.14944 |     0.15064 |    0.0049327 |

_____________________________________________________________
Optimization completed.
MaxObjectiveEvaluations of 30 reached.
Total function evaluations: 30
Total elapsed time: 63.8085 seconds.
Total objective function evaluation time: 93.2945

Best observed feasible point:
    BoxConstraint
    _____________

       0.4535

Observed objective function value = 0.14944
Estimated objective function value = 0.15064
Function evaluation time = 2.7079

Best estimated feasible point (according to models):
    BoxConstraint
    _____________

       0.4535

Estimated objective function value = 0.15064
Estimated function evaluation time = 2.734
```

Figure 4: Detector de mirades: només HOG.

```
SVM 5
Copying objective function to workers...
Done copying objective function to workers.
|=====================================================================================================|
| Iter | Active   | Eval   | Objective   | Objective   | BestSoFar   | BestSoFar   | BoxConstraint|
|      | workers  | result |             | runtime     | (observed)  | (estim.)    |              |
|=====================================================================================================|
|    1 |       1  | Best   |     0.28947 |      0.2329 |     0.28947 |     0.31393 |       12.158 |
|    2 |       1  | Accept |     0.40038 |      0.2171 |     0.28947 |     0.31393 |      0.45838 |
|    3 |       2  | Best   |     0.28665 |     0.26099 |     0.28665 |     0.28667 |       8.4066 |
|    4 |       2  | Accept |     0.40038 |     0.18857 |     0.28665 |     0.28668 |      0.26533 |
|    5 |       2  | Best   |      0.2735 |     0.58009 |      0.2735 |     0.27351 |       111.78 |
|    6 |       2  | Accept |     0.40038 |     0.17051 |      0.2735 |     0.27455 |    0.0010013 |
|    7 |       2  | Accept |     0.28289 |     0.27786 |      0.2735 |     0.27426 |       35.517 |
|    8 |       2  | Accept |      0.2782 |     0.99732 |      0.2735 |     0.27435 |       213.15 |
|    9 |       2  | Best   |      0.2735 |      0.5199 |      0.2735 |     0.27349 |       90.896 |
|   10 |       2  | Accept |     0.27726 |      4.0239 |      0.2735 |     0.27345 |       806.13 |
|   11 |       2  | Best   |     0.27256 |     0.46867 |     0.27256 |     0.27298 |       80.621 |
|   12 |       2  | Accept |     0.40038 |     0.18555 |     0.27256 |     0.27298 |     0.014942 |
|   13 |       2  | Accept |     0.27256 |     0.50413 |     0.27256 |     0.27276 |       77.611 |
|   14 |       2  | Accept |     0.27256 |     0.46873 |     0.27256 |     0.27272 |       83.382 |
|   15 |       2  | Accept |     0.27256 |     0.48543 |     0.27256 |     0.27266 |       75.587 |
|   16 |       2  | Accept |     0.40038 |     0.25002 |     0.27256 |     0.27266 |    0.0036433 |
|   17 |       2  | Accept |     0.27256 |     0.48799 |     0.27256 |     0.27263 |       75.533 |
|   18 |       2  | Accept |     0.40038 |     0.24679 |     0.27256 |     0.27263 |     0.058543 |
|   19 |       2  | Accept |     0.35808 |     0.19537 |     0.27256 |     0.27264 |       2.2667 |
|   20 |       2  | Accept |     0.29981 |      0.2014 |     0.27256 |     0.27264 |       4.5345 |
|=====================================================================================================|
| Iter | Active   | Eval   | Objective   | Objective   | BestSoFar   | BestSoFar   | BoxConstraint|
|      | workers  | result |             | runtime     | (observed)  | (estim.)    |              |
|=====================================================================================================|
|   21 |       2  | Accept |     0.27444 |     0.48006 |     0.27256 |     0.27274 |       70.021 |
|   22 |       2  | Best   |     0.27162 |     0.46506 |     0.27162 |      0.2725 |       86.527 |
|   23 |       2  | Accept |     0.40038 |     0.17005 |     0.27162 |     0.27251 |    0.0018257 |
|   24 |       2  | Accept |      0.2782 |      2.3704 |     0.27162 |     0.27251 |       457.19 |
|   25 |       2  | Accept |     0.40038 |     0.24092 |     0.27162 |     0.27252 |      0.11846 |
|   26 |       2  | Accept |     0.28571 |     0.31628 |     0.27162 |     0.27248 |       21.207 |
|   27 |       2  | Accept |     0.39756 |     0.19593 |     0.27162 |     0.27249 |       1.0522 |
|   28 |       2  | Accept |     0.40038 |     0.17658 |     0.27162 |     0.27249 |    0.0076597 |
|   29 |       2  | Accept |     0.40038 |     0.18907 |     0.27162 |      0.2725 |     0.029621 |
|   30 |       2  | Accept |     0.29041 |     0.19811 |     0.27162 |     0.27252 |       6.1982 |

__________________________________________________________
Optimization completed.
MaxObjectiveEvaluations of 30 reached.
Total function evaluations: 30
Total elapsed time: 23.5341 seconds.
Total objective function evaluation time: 15.7657

Best observed feasible point:
    BoxConstraint
    _____________

      86.527

Observed objective function value = 0.27162
Estimated objective function value = 0.27252
Function evaluation time = 0.46506

Best estimated feasible point (according to models):
    BoxConstraint
    _____________

      83.382

Estimated objective function value = 0.27252
Estimated function evaluation time = 0.49589
```

Figure 5: Detector de mirades: només LBP.

```
SVM 5
Copying objective function to workers...
Done copying objective function to workers.
|=====================================================================================================|
| Iter | Active   | Eval   | Objective   | Objective   | BestSoFar   | BestSoFar   | BoxConstraint|
|      | workers  | result |             | runtime     | (observed)  | (estim.)    |              |
|=====================================================================================================|
|    1 |       1  | Best   |     0.28947 |      0.2329 |     0.28947 |     0.31393 |       12.158 |
|    2 |       1  | Accept |     0.40038 |      0.2171 |     0.28947 |     0.31393 |      0.45838 |
|    3 |       2  | Best   |     0.28665 |     0.26099 |     0.28665 |     0.28667 |       8.4066 |
|    4 |       2  | Accept |     0.40038 |     0.18857 |     0.28665 |     0.28668 |      0.26533 |
|    5 |       2  | Best   |      0.2735 |     0.58009 |      0.2735 |     0.27351 |       111.78 |
|    6 |       2  | Accept |     0.40038 |     0.17051 |      0.2735 |     0.27455 |    0.0010013 |
|    7 |       2  | Accept |     0.28289 |     0.27786 |      0.2735 |     0.27426 |       35.517 |
|    8 |       2  | Accept |      0.2782 |     0.99732 |      0.2735 |     0.27435 |       213.15 |
|    9 |       2  | Best   |      0.2735 |      0.5199 |      0.2735 |     0.27349 |       90.896 |
|   10 |       2  | Accept |     0.27726 |      4.0239 |      0.2735 |     0.27345 |       806.13 |
|   11 |       2  | Best   |     0.27256 |     0.46867 |     0.27256 |     0.27298 |       80.621 |
|   12 |       2  | Accept |     0.40038 |     0.18555 |     0.27256 |     0.27298 |     0.014942 |
|   13 |       2  | Accept |     0.27256 |     0.50413 |     0.27256 |     0.27276 |       77.611 |
|   14 |       2  | Accept |     0.27256 |     0.46873 |     0.27256 |     0.27272 |       83.382 |
|   15 |       2  | Accept |     0.27256 |     0.48543 |     0.27256 |     0.27266 |       75.587 |
|   16 |       2  | Accept |     0.40038 |     0.25002 |     0.27256 |     0.27266 |    0.0036433 |
|   17 |       2  | Accept |     0.27256 |     0.48799 |     0.27256 |     0.27263 |       75.533 |
|   18 |       2  | Accept |     0.40038 |     0.24679 |     0.27256 |     0.27263 |     0.058543 |
|   19 |       2  | Accept |     0.35808 |     0.19537 |     0.27256 |     0.27264 |       2.2667 |
|   20 |       2  | Accept |     0.29981 |      0.2014 |     0.27256 |     0.27264 |       4.5345 |
|=====================================================================================================|
| Iter | Active   | Eval   | Objective   | Objective   | BestSoFar   | BestSoFar   | BoxConstraint|
|      | workers  | result |             | runtime     | (observed)  | (estim.)    |              |
|=====================================================================================================|
|   21 |       2  | Accept |     0.27444 |     0.48006 |     0.27256 |     0.27274 |       70.021 |
|   22 |       2  | Best   |     0.27162 |     0.46506 |     0.27162 |      0.2725 |       86.527 |
|   23 |       2  | Accept |     0.40038 |     0.17005 |     0.27162 |     0.27251 |    0.0018257 |
|   24 |       2  | Accept |      0.2782 |      2.3704 |     0.27162 |     0.27251 |       457.19 |
|   25 |       2  | Accept |     0.40038 |     0.24092 |     0.27162 |     0.27252 |      0.11846 |
|   26 |       2  | Accept |     0.28571 |     0.31628 |     0.27162 |     0.27248 |       21.207 |
|   27 |       2  | Accept |     0.39756 |     0.19593 |     0.27162 |     0.27249 |       1.0522 |
|   28 |       2  | Accept |     0.40038 |     0.17658 |     0.27162 |     0.27249 |    0.0076597 |
|   29 |       2  | Accept |     0.40038 |     0.18907 |     0.27162 |      0.2725 |     0.029621 |
|   30 |       2  | Accept |     0.29041 |     0.19811 |     0.27162 |     0.27252 |       6.1982 |

__________________________________________________________
Optimization completed.
MaxObjectiveEvaluations of 30 reached.
Total function evaluations: 30
Total elapsed time: 23.5341 seconds.
Total objective function evaluation time: 15.7657

Best observed feasible point:
    BoxConstraint
    _____________

      86.527

Observed objective function value = 0.27162
Estimated objective function value = 0.27252
Function evaluation time = 0.46506

Best estimated feasible point (according to models):
    BoxConstraint
    _____________

      83.382

Estimated objective function value = 0.27252
Estimated function evaluation time = 0.49589
```

Figure 6: Detector de mirades: HOG i LBP.

3.4 Anàlisi dels resultats

Com podem veure, els resultats de la cross validation pel detector d'ulls són molt bons per les tres possibilitats de característiques utilitzades. En els tres casos l'error tendeix a 0, i són pràcticament idèntics. Estadísticament, les diferències no són significatives. Arbitràriament, escollirem el predictor amb només HOG, perquè entenem que és l'opció més habitual. En aquest cas, el millor valor trobat per `BoxConstraint` és 0.1541.

Pel que fa a la cross validation pel detector de mirades, els resultats són pitjors (cosa comprensible al tenir menys dades i tractar-se d'un problema segurament més difícil) en general, però en aquest cas les diferències sí que són significatives. Utilitzant només HOG o HOG i LBP conjuntament, l'error és d'aproximadament el 15%. En canvi, utilitzant només LBP, l'error és d'aproximadament del 27%. Per tant, estadísticament és evident que LBP soles segur que no són les millors característiques. Per escollir entre només HOG i HOG i LBP, en aquest cas, també de manera arbitrària, optarem per HOG i LBP, perquè fent recerca hem trobat que és una combinació habitual per detectar ulls. En aquest cas, el millor valor trobat per `BoxConstraint` és 0.37611.

4 Avaluació

4.1 Resultats amb test (`evaluate.m`)

Per fer una estimació honesta de l'error de generalització del model final, el provarem en la partició de test.

Cal destacar que el codi per generar la matriu de confusió, l'accuracy i la F1 score (útil en aquest cas al tractar-se d'un dataset desbalancejat) ha estat implementat per nosaltres (exceptuant la pròpia matrriu, que la genera MATLAB).

Pel que fa al detector d'ulls:

```
confMat1 =

        8688           0
           0         438


accuracy1 = 1


FScoreMinority1 = 1
```

Encerta de ple, cosa que concorda amb l'error de validació del cross validation. Cal dir, però, que la poblacioó d'imatges amb ulls deu seguir una distribució molt més complexa que la presentada per aquesta base de dades. Per això cal anar en compte. La F score de la classe en minoria és rellevant perquè el dataset està desbalancejat.

Per alta banda, pel que fa al detector de mirades:

```
confMat6 =

   143    35
    42   237


accuracy6 = 0.8315


FScoreMinority6 = 0.8603
```

Els resultats ja observem que són més mediocres, però continuen sent bons. De nou, depenem de com de representativa sigui la base de dades.

5 Aplicació

5.1 Aplicació amb finestra lliscant (`app.m`)

Hem programat *a mà* una finestra lliscant que va aplicant el detector d'ulls i, en cas que en trobi, el de mirades. Es pot decidir la mida de la finestra. Tanmateix, el predictor és bastant sensible a aquest paràmetre.

Per altra banda, també es pot decidir com de ràpid ha d'avançar la finestra (per defecte, només es mou 1 píxel a cada iteració). Així mateix, es pot escollir si l'algorisme ha d'aturar-se al trobar el primer element que li sembli un ull o ha d'examinar tota la imatge i retornar el rectangle amb major probabilitat d'ulls (donat que n'hi hagi), mitjançant el flag `earlyStop`, per defecte definit com a false.

Aquest algorisme no funciona sempre amb imatges fora del dataset, perquè és molt sensible a la finestra i les probabilitats de ser o no ser ulls són semblants per tot el voltant de la zona dels ulls. Això fa, per exemple, que no sempre retorni un rectangle centrat, i també hi ha el problema que el segon predictor, el de la mirada, s'aplica amb aquest requadre.

Tanmateix, aquí incorporem un example de resposta correcta de l'algorisme amb una imatge real fora del dataset, encara que ja veiem que el rectangle no està centrat com esperaríem:

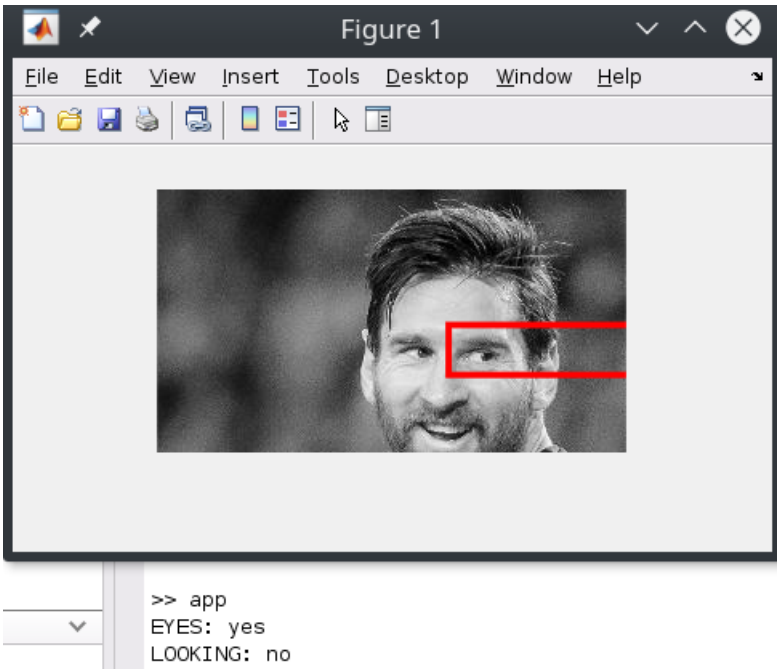


Figure 7: Aplicació amb finestra lliscant..

5.2 Extra: aplicació interactiva (`app_interact.m`)

Pensem que és millor utilitzar l'altra aplicació que hem utilitzat, explicada a continuació. Per a provar de manera més interactiva el predictor i veure amb més detall com de bé (o malament) funciona, hem desenvolupat un script en què l'usuari selecciona un rectangle de la imatge i aleshores s'aplica el predictor. Per a provar-ho de manera justa, el rectangle seleccionat hauria de tenir la mida i forma que s'espera que tingui el rectangle contenidor dels ulls.

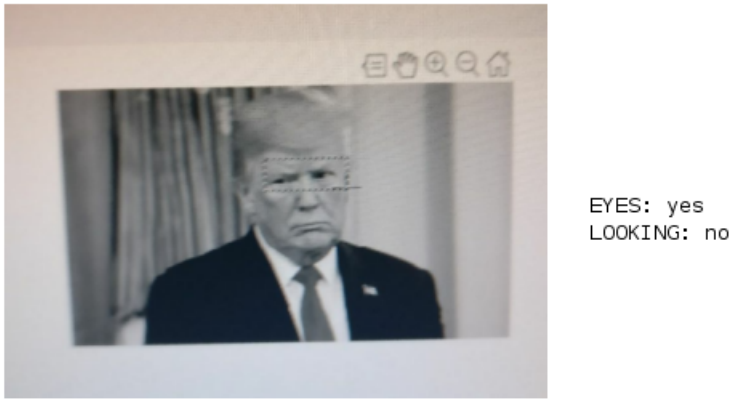


Figure 8: Aplicació interactiva.

6 Implementacions pròpies

6.1 Detalls d'implementació i codis propis

Cal dir que tot el codi ha estat desenvolupat per nosaltres d'una manera o altra (òbviament consultant exemples o documentació), amb l'excepció de la funció per a escanejar arxius, la base de la qual ha estat proporcionada pel professor. Tanmateix, l'hem hagut de modificar considerablement, per a llegir també els noms de cada imatge i l'Excel amb les etiquetes de les mirades.

Dit això, hi ha codis en què hem recorregut molt més a funcions predefinides de MATLAB que en d'altres. Ho especifiquem a continuació. Per a extreure les característiques i per a la cross validation i l'optimització de paràmetres, hem utilitzat bàsicament funcions predefinides de MATLAB. En canvi, el codi per generar les dades, el codi per remostrejar les dades en diferents particions aleatòriament, el codi per generar les estadístiques (precisió i F1 score), el codi per aplicar la finestra lliscant i el codi en què l'usuari selecciona un rectangle està desenvolupat *a mà* per nosaltres, com a mínim en gran part.

Per tant, considerem que no ens hem limitat a cridar funcions predefinides, sinó que una part notable del codi ha estat directament implementat per nosaltres. A més, creiem que ho hem fet amb un cert sentit. Les funcions predefinides que hem cridat funcionaven bé i les teníem molt a l'abast. En canvi, en el cas de codis que hem implementat, no hem trobat bones funcions predefinides que s'adaptessin a les nostres característiques.

Així mateix, fem notar que cada secció del codi es pot executar independentment, gràcies a que hem desat en disc els models ja entrenats. Hem indicat una seed per tal que els resultats siguin reproduïbles.

7 Conclusions

Finalment, hem arribat a les següents conclusions:

- Per a poder utilitzar predictors clàssics amb imatges, cal extreure'n característiques d'una manera o altra, i aquest procés és altament determinant de cara al rendiment del predictor.
- El problema de detectar ulls és més fàcil que el de detectar mirades, atenent els resultats.
- Que un predictor funcioni bé amb una base de dades, encara que s'hagi validat i testejat sense que hagi vist les dades durant l'entrenament, no és garantia que funcioni perfectament a la realitat, perquè les imatges del dataset poden tenir uns patrons comuns que no compleixin totes les imatges.
- En el nostre cas, en test aconseguim 100% d'encert pels ulls, i un 85% aproximadament per les mirades. Però no és cap garantia que a la realitat tingui el mateix èxit. En el cas de la detecció d'ulls, els 3 conjunts de característiques provats han funcionat igualment bé, però en el cas de les mirades, utilitzar només LBP era clarament inferior.
- Com a principals limitacions d'aquest treball, per falta de dades, temps i, sobretot, recursos computacionals, no hem pogut fer tot l'entrenament que s'hagués pogut portar a terme. Per exemple, optimitzar més paràmetres dels SVMs, o provar altres algorismes de classificació, o ampliar el dataset. Ho suggerim com a treball futur.
- Considerem que hem fet un bon treball i hem aconseguit bons resultats.