

10. Detectia de persoane

10.1. Scop

Scopul acestei lucrari este de a implementa o metoda de detectie de persoane sau parti corporale. Metoda se bazeaza pe detectia de trasaturi de tip Haar calculate pe sub-regiuni din imagine folosind imaginea integrala (vezi cursurile 7-8), si identificarea prezentei persoanelor sau a unor parti ale corpului uman cu un clasificator de tip cascada.

Modelele clasificatorilor cascada bazati pe trasaturi Haar pentru detectia de persoane sunt disponibile in locatia: `%OPENCV_DIR%\data\haarcascades\` si sunt urmatoarii:

```
haarcascade_fullbody.xml
haarcascade_lowerbody.xml
haarcascade_upperbody.xml
haarcascade_mcs_upperbody.xml
```

10.2. Mersul lucrării

1. Se va crea o functie de procesare folosind sablonul de la metoda de detectie a fetelor pe imagini statice (L8). Se va modifica aceasta functie pentru a incarca modelele de clasificatori pentru persoane (*fullbody*, *lowerbody*, *upperbody*). Se va apela functia de detectie a obiectelor `detectMultiScale`, succesiv, pentru fiecare dintre cele 3 modele.
2. Se va parcurge lista de obiecte obtinuta pentru fiecare tip de obiect, si se vor afisa dreptunghiurile care incadreaza fiecare parte corporala in culori distincte: cyan - *fullbody*, magenta - *upperbody*, yellow - *lowerbody*.

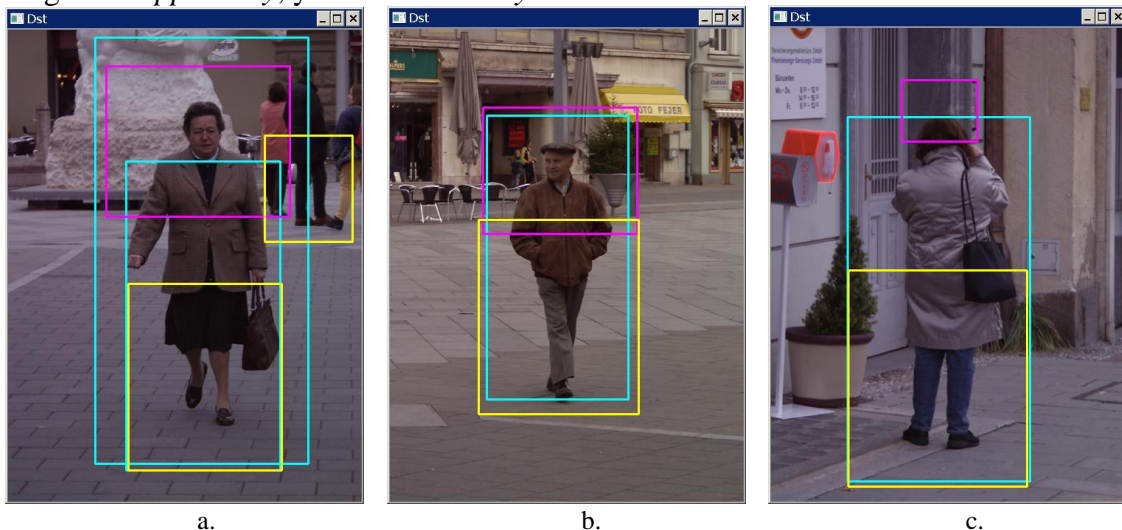


Fig. 10.1. Rezultate ale detectiilor (detectia pentru fiecare model in parte *fullbody*, *lowerbody*, *upperbody* se va afisa/incadra cu cate un dreptunghi de culoare diferita)

Observatie: ultimii 2 parametri ai functiei `detectMultiScale` specifica dimensiunea minima initiala (width, height) a zonei rectangulare in care se cauta partile corpului. Pentru fiecare tip de obiect se vor respecta proportiile corporale:

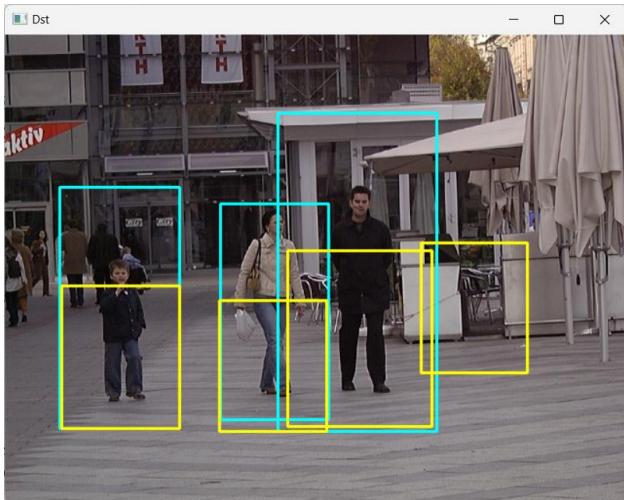
(*fullbody*, *lowerbody*, *upperbody*) \rightarrow width $\sim 0.4 \dots 0.5 * \text{minBodyHeight}$;

(*lowerbody*, *upperbody*) \rightarrow height $\sim 0.5 * \text{minBodyHeight}$;

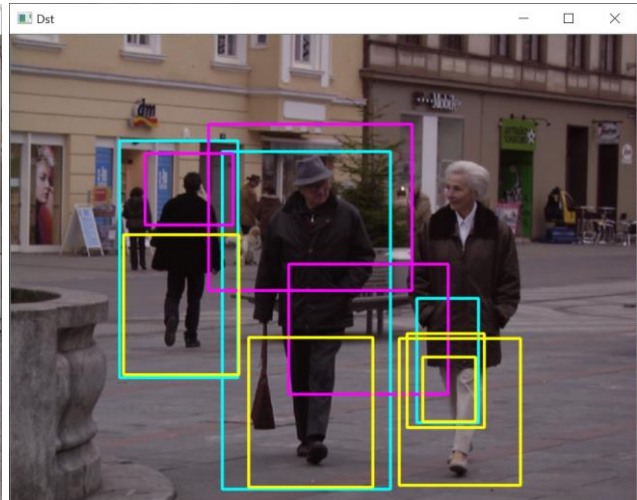
fullbody \rightarrow height = minBodyHeight (aprox 100 ... 150 pt. imaginile de test din arhiva *Persons.zip*)

Rezultate detectiilor, cu paarametrii de apel de mai jos se pot observa in fig. 10.2:

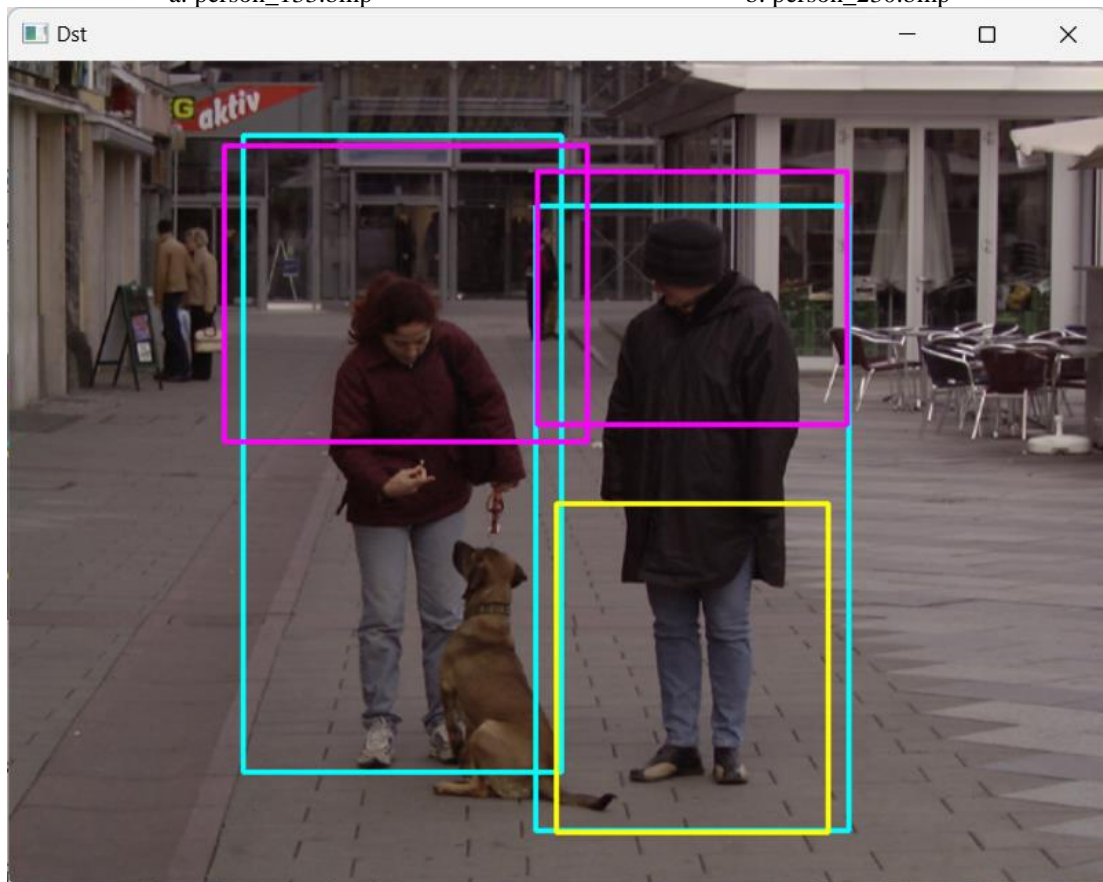
```
int minBodyHeight = 150;
/-- Detect bodies
fullbody_cascade.detectMultiScale(frame_gray, fbodies, 1.1, 2, 0, Size(minBodyHeight *
0.5f, minBodyHeight));
upperbody_cascade.detectMultiScale(frame_gray, ubodies, 1.1, 2, 0, Size(minBodyHeight *
0.5f, minBodyHeight * 0.5));
lowerbody_cascade.detectMultiScale(frame_gray, lbodies, 1.1, 2, 0, Size(minBodyHeight *
0.5f, minBodyHeight * 0.5));
```



a. person_133.bmp



b. person_230.bmp



c. person_238.bmp

Fig. 10.2. Rezultatele detectiilor pe imagini cu persoane multiple.

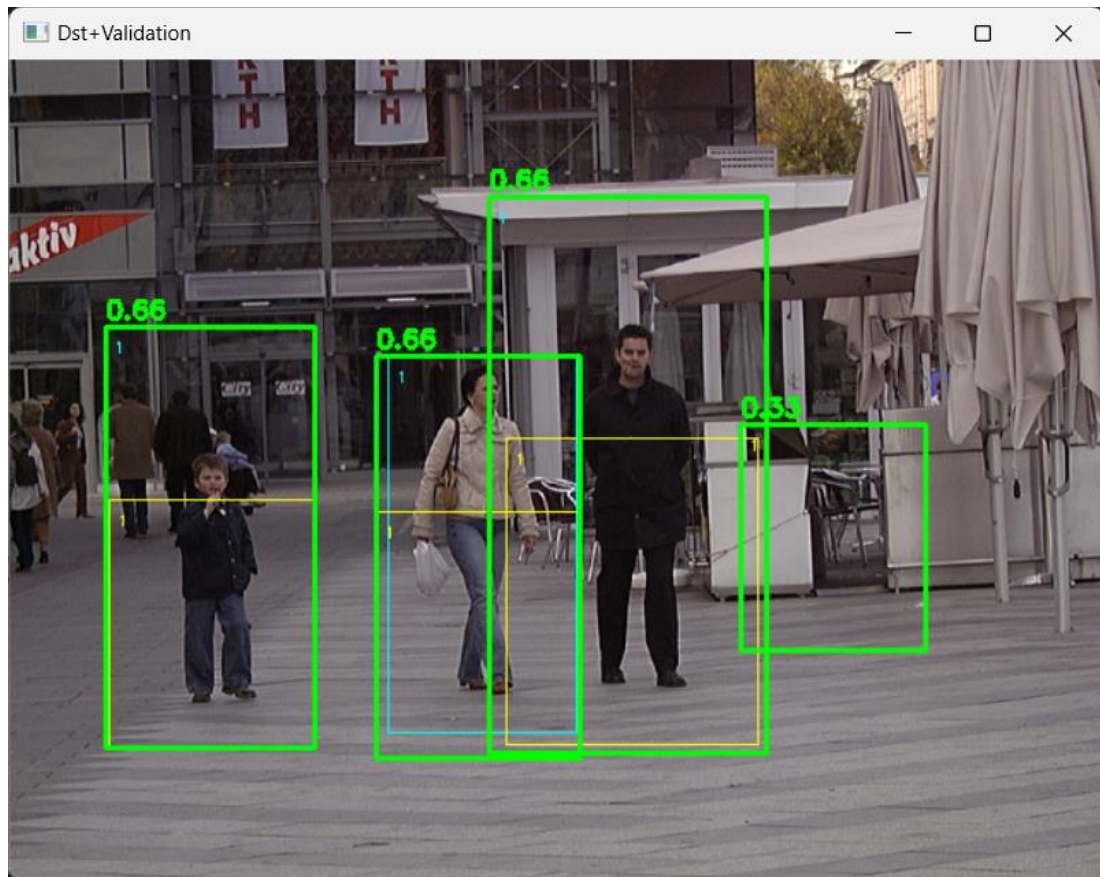
3. Se va face o validare suplimentara pe baza stabilirii unui set de reguli legate de proportiile obiectelor detectate si de pozitia lor relativa, pentru a grupa detectiile aferente aceleasi persoane. Pentru fiecare obiect/persoana detectata se va afisa un scor de incredere.

Exemplu:

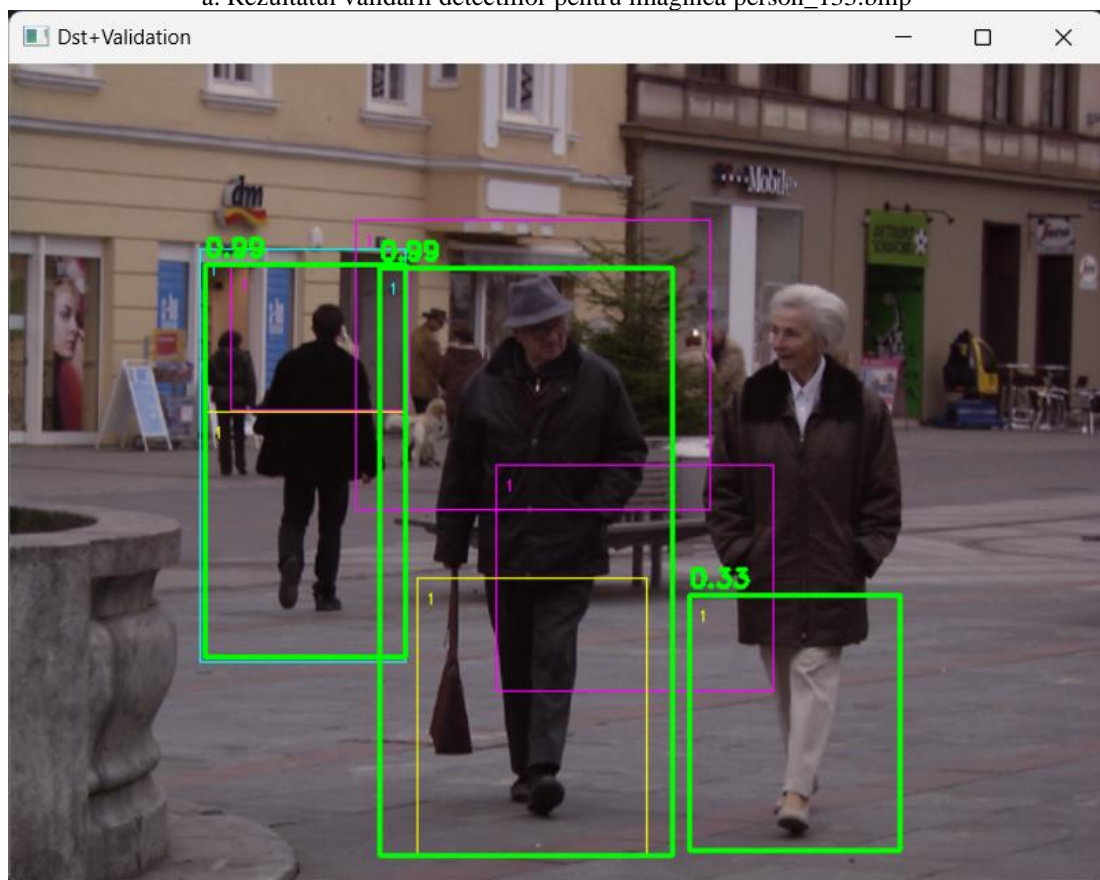
- Daca se detecteaza 3 obiecte (*fullbody*, *lowerbody*, *upperbody*) cu centrele aproximativ pe aceeasi verticala (diferenta pe orizontala intre coordonatele x ale centrele acestor dreptunghiuri sa fie mai mica decat $0.5 * \text{minBodyHeight}$) **SI** centrele lor au valori crescatoare ale coordonatelo y, in ordinea *upperbody*, *fullbody*, *lowerbody*, **SI** ariile lor respecta urmatoarea relatie: $0.7 < \text{aria}(\text{fullbody}) / \text{aria}(\text{lowerbody} \cup \text{upperbody}) < 1.3$ **atunci** scorul este **0.99** si retineti/memorati un rectange $\text{fullbody} \cap (\text{lowerbody} \cup \text{upperbody})$ (vezi exemplele din figura 10.3 si exemplul de cod aferent implementarii acestui criteriu in anexa 5)
- Daca se detecteaza 2 obiecte (*lowerbody*, *upperbody*) cu centrele aproximativ pe aceeasi verticala **SI** *upperbody* este detectat deasupra lui *lowerbody* si distanta pe vertical intre centrele lor este mai mica decat $2.5 * \text{minBodyHeight}$ **atunci** scorul este **0.66** si retineti un dreptunghi ($\text{lowerbody} \cup \text{upperbody}$).
- Daca se detecteaza 2 obiecte (*upperbody*, *fullbody*) cu centrele aproximativ pe aceeasi verticala **SI** centrul lui *upperbody* este detectat deasupra centrului lui *fullbody* **SI** $\text{aria}(\text{upperbody} \cap \text{fullbody}) / \text{aria}(\text{upperbody}) > 0.5$ **atunci** scorul este **0.66** si retineti un dreptunghi ($\text{upperbody} \cup \text{fullbody}$).
- Daca se detecteaza 2 obiecte (*lowerbody*, *fullbody*) cu centrele aproximativ pe aceeasi verticala **SI** centrul lui *lowerbody* este detectat sub centrul lui *fullbody* **SI** $\text{aria}(\text{lowerbody} \cap \text{fullbody}) / \text{aria}(\text{lowerbody}) > 0.5$, **atunci** scorul este **0.66** si retineti un dreptunghi ($\text{lowerbody} \cup \text{fullbody}$).
- Obiectele neprocesate se raporteaza ca si persoane distincte cu scorul de **0.33**, daca nu se suprapun cu obiecte validate la criteriile a .. d.

Sugestii pentru implementare:

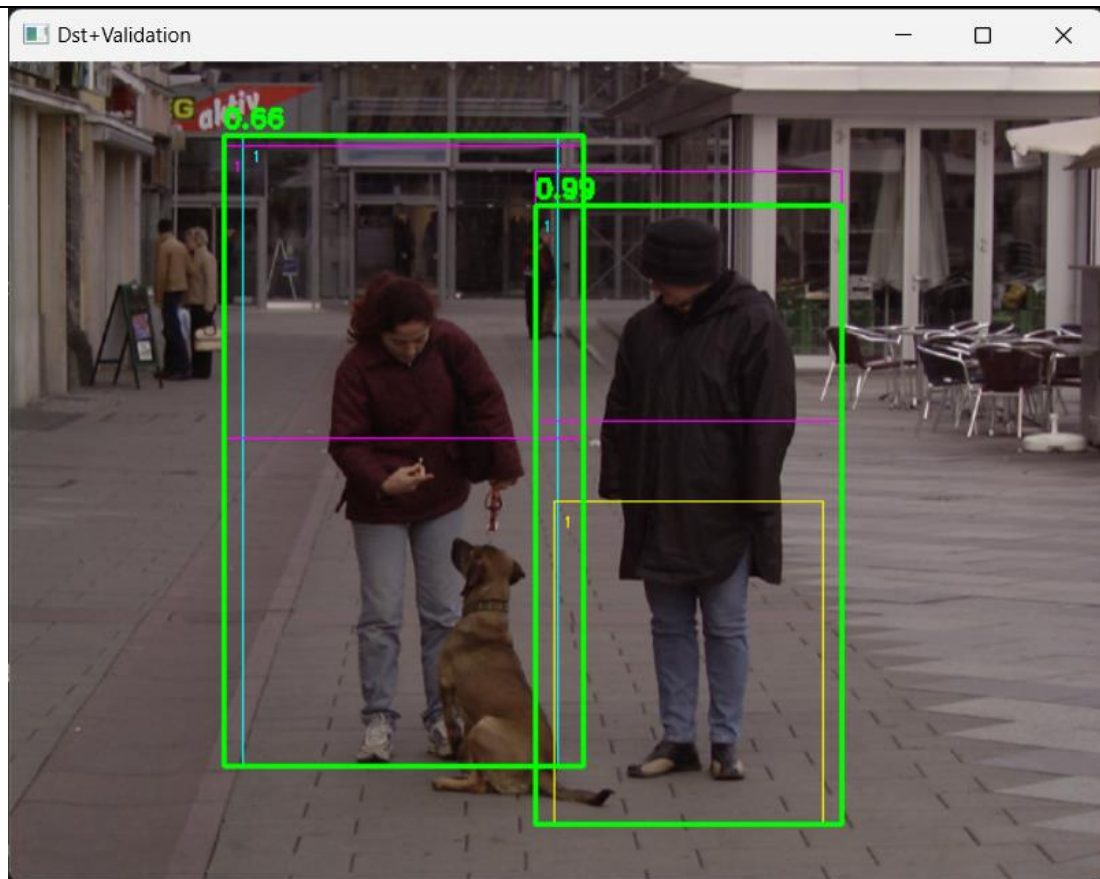
- Creati-va vectori pentru a stoca dreptunghiul care circumscrie o persoana validata si scorul de confidenta asociat/calculat (exemplu: *persons* si *personsCF*). Odata ce ati validat o persoana conform criteriilor a ... e inserati dreptunghiul si scorul in vectorii creati.
- Utilizati niste vectori (*fproc*, *lproc*, *uproc*), avand aceeasi lungime ca si vectorii *fbodies*, *ubodies*, *lbodies*, si initialzati cu 0, pentru a marca partile corporala (detectiile) pe care le-ati procesat. Odata ce le-ati procesat, schimati valoarea din vectori respectivi din 0 in 1 pentru a evita sa reprocesati detectia respectiva.
- Pentru criteriul e:
 - parcurgeti lista/vectorul cu persoane valdate (*persons*) si pentru fiecare element *i* din listele/vectorii *fbodies*, *ubodies*, *lbodies* (care este marcat ca neprocesat in vectorii *fproc*, *lproc*, *uproc*), verificati daca se suprapune peste vreun element din lista de persoane validate (*persons*), pe baza unui test de arie de tipul: $\text{aria}(\text{Xbodies}[i]) \cap \text{persons}[ii]) / \text{aria}(\text{Xbodies}[i]) > 0.5$. In caz afirmativ marcati detectiile respective ca si procesate (fara sa le mai inserati il vectorul *persons*).
 - Parcurgeti din nou, pe rand elementele din vectorii *fbodies*, *ubodies*, *lbodies*, si daca sunt marcate ca neprocesate in *fproc*, *lproc*, *uproc*, le schimbati starea din 0 (neprocesat) in 1 (procesat) si le inserati in lista *persons* cu un CF = 0.33.
- Pentru a calcula centrul unui obiect de tip *Rect* si aria lui, aveteti functiile predefinite in modulul *Functions*: *RectCenter* si *RectArea* (vedeti anexele 3 si 4).
- Pentru a realiza intersectia respectiv reuniunea dintre doua obiecte de tip *Rect*, aveti operatorii *&* respectiv *|* din *OpenCV* (vezi anexa 1 si 2).



a. Rezultatul validării detectiilor pentru imaginea person_133.bmp



b. Rezultatul validării detectiilor pentru imaginea person_230.bmp



c. Rezultatul validarii detectiilor pentru imaginea person_238.bmp
 Fig. 10.3. Rezultatele detectiilor validate pe imagini cu persoane multiple.

10.3. Activitati practice

Implementati cerintele 1-3 intr-o singura functie de procesare.

Rezultatul final al validarii puteti sa il desenati peste imaginea color in care afisati rezultatele detectiilor sub forma unui chenar verde si a scorului de confidenta asociat (utilizati functia putText).

Exemplele (imaginele de test) sunt selectate din setul de date: http://www.emt.tugraz.at/~pinz/data/GRAZ_01/ si se gasesc la locatia: <http://users.utcluj.ro/~tmarita/HCI/Media/Images/Persons.zip> sau pe Teams. Pentru a rezolva/testa punctul 3 din mersul lucrarii este recomandat sa lucrati atat pe imagini cu o persoana (011, 090, 096) cat si pe imagini cu persoane multiple (138, 230, 238)

10.4. ANEXE

1. Calculul intersectiei dintre 2 dreptunghiuri

OpenCV (Rect): http://docs.opencv.org/2.4/modules/core/doc/basic_structures.html#rect
`rect = rect1 & rect2 //rectangle intersection`

2. Calculul reuniunii dintre 2 dreptunghiuri (dreptunghiul de arie minima care contine cele doua dreptunghiuri componente)

OpenCV (Rect): http://docs.opencv.org/2.4/modules/core/doc/basic_structures.html#rect
`rect = rect1 | rect2 //rectangle union`

3. Calcul centru dreptunghi – functie definita in modulul *Functions*

```
Point RectCenter(Rect R)
{
```

```

    Point P;
    P.x = R.x + R.width / 2;
    P.y = R.y + R.height / 2;
    return P;
}

```