

## Documentatie Tema 2 CAVA

### Task 1.

Fisierul Parameters.py contine clasa Parameters, care are diversi parametrii ai programului (spre ex: dimensiunea unui exemplu pozitiv, dimensiunea unei celule hog etc.)

Fisierul FacialDetector.py contine functii care ajuta la detectarea faciale a caracterelor din Familia Flinstone.

Functia `get_positive_descriptors` calculeaza descriptorii pozitivi pentru exemplele pozitive. Va returna un numpy array de dimensiune  $N * D$ , unde  $N$  = nr. Exemple pozitive si  $D$  = dimensiune descriptor.

Functia `get_negative_descriptors` calculeaza descriptorii negativi. Va returna tot un numpy array de dimensiuni  $N * D$ , unde  $N$  = nr. exemple negative si  $D$  = dimensiune descriptor.

Functia `train_classifier` antreneaza 5 clasificatori si il alege pe cel cu cea mai buna acuratete. Afiseaza si un plot pe ecran cu distributia scorurilor clasificatorului pe exemplele de antrenare.

Functia `intersection_over_union` calculeaza Intersectia peste Uniune (IoU) pentru doua cutii de delimitare.

Functia `non_maximal_suppression` va returna `image_detections` si `image_scores` care sunt maximale.

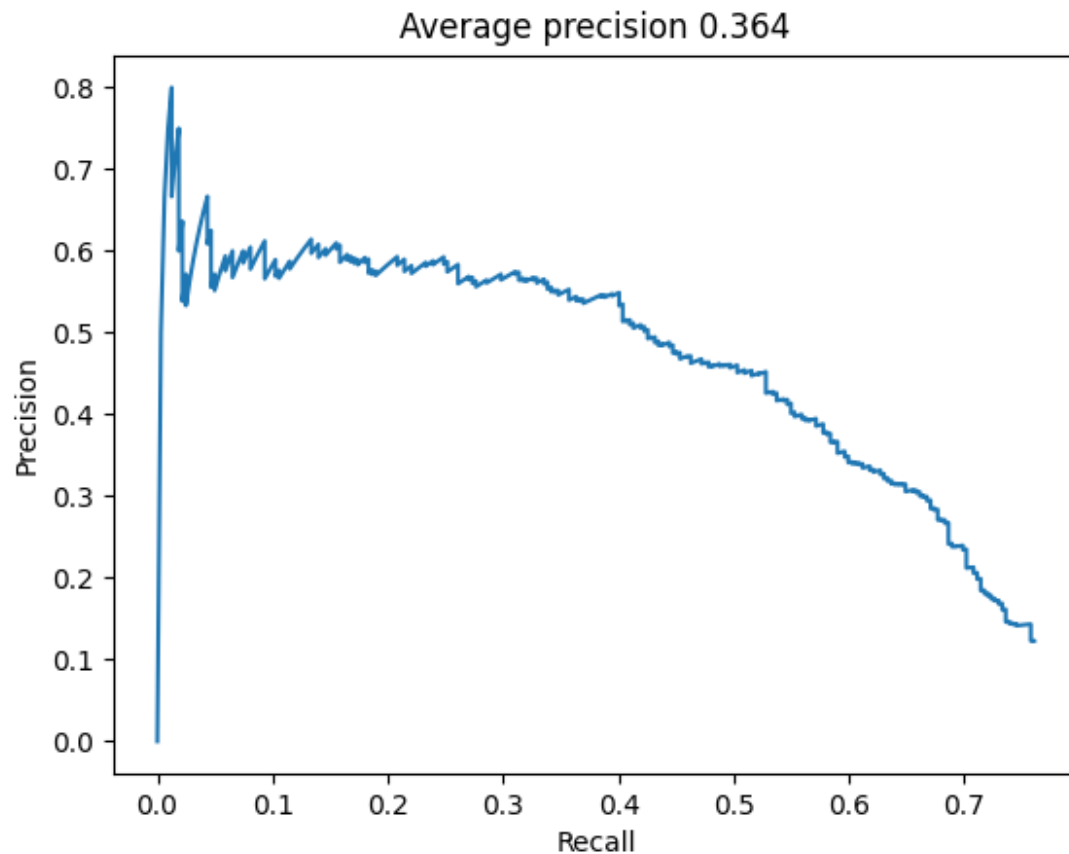
Detectiile cu scor mare suprima detectiile ce se suprapun cu acestea dar au scor mai mic. Detectiile se pot suprapune partial, dar centrul unei detectii nu poate fi in interiorul celeilalte detectii.

Parametrii: - `image_detections`: numpy array de dimensiune  $N \times 4$ , unde  $N$  este numarul de detectii; - param `image_scores`: numpy array de dimensiune  $N$ ; - param `image_size`: tuplu, dimensiunea imaginii.

Functia `run` returneaza toate detectiile (= ferestre) pentru toate imaginile din `self.params.dir_test_examples`. Directorul cu numele `self.params.dir_test_examples` contine imagini ce pot sau nu contine fete. Aceasta functie ar trebui sa detecteze fete atat pe setul de date de antrenare, dar si pentru alte imagini. Functia '`non_maximal_suppression`' suprimeaza detectii care se suprapun (protocolul de evaluare considera o detectie duplicata ca fiind falsa). Suprimarea non-maximelor se realizeaza pe pentru fiecare imagine. Functia returneaza: - `detections`: numpy array de dimensiune  $N * 4$ , unde  $N$  este numarul de detectii pentru toate imaginile. `detections[i, :] = [x_min, y_min, x_max, y_max]`; - `scores`: numpy array de dimensiune  $N$ , scorurile pentru toate

detectiile pentru toate imaginile; - file\_names: numpy array de dimensiune N, pentru fiecare detectie trebuie sa salvam numele imaginii. (doar numele, nu toata calea).

Functia eval\_detections evalueaza detectiile si afiseaza un grafic (plot) cu precizia medie.



Fisierul RunProject.py apeleaza diverse functii si attribute din celelalte clase importate si efectueaza detectarea faciale. Algoritmul urmeaza 4 pasi: Pasii 1+2+3. Incarcam exemplele pozitive (cropate) si exemple negative generate (verificam daca sunt deja existente); Pasul 4. Invatam clasificatorul liniar. In final saleaza 3 fisiere (detections\_all\_faces.npy, scores\_all\_faces.npy, file\_names\_all\_faces.npy) cu output-urile intr-un folder.