2022-02-11 Olimpiādes 3. Uzdevums

Datu kopa:
http://share.yellowrobot.xyz/1644411423-olimpiade-2022/fer_features_dataset_incomplete.csv

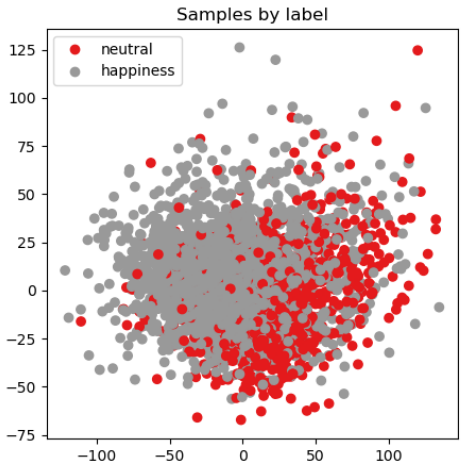
Python sagatave:
http://share.yellowrobot.xyz/1644411423-olimpiade-2022/3_template.py.zip

Nepieciešams izveidot programmu, kura kā ievades datus saņem CSV failu ar datiem un kā izvades datus saglabā šajā pašā failā trūkstošos marķējumus (labels). Pārbaudes fails nesaturēs nevienu marķējumu (label). Modelis drīkst izmantot apmācību failu un citus papildus failus. Uzdevumu iesniegt kā ZIP failu, kurš satur pirmkodu un instrukcijas kā to izmantot.

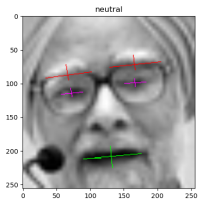
Datu kopa satur emociju marķējums (label) un vektoru kopu, kura raksturo, uzacis, acis, lūpas un zobus (ja tādi ir). Daļai datu nav pieejams emociju marķējums, bet tos var izmantot, lai uzlabotu modeļa precizitāti. Tāpat daļai datu nav pieejami zobu vektori, jo attēlā var nebūt redzami zobi. Orģinālais attēlu izmērs, no kura iegūti šie vektori ir 256x256 pikseļi.

Par uzdevumu piešķirtie punkti tiks aprēķināti, balstoties uz to cik daudz marķējumi tiks pareizi piešķirti pārbaudes datu kopai.

Piemērs klasterizācijai starp neitrālām un pozitīvām emocijām, izmantojot lūpu vektorus.



Piemēri īpašību vektoriem, kuri uzzīmēti pa virsu orģinālajam attēlam.

Neitrāls (neutral)	Prieks (happiness)	Bēdas (sadness)	Dusmas (anger)
			

Dati ir pieejami CSV formātā, bet zemāk parādīts to saturs kā Dict:

```
record = {
    'label': labels[Y],

    'eye_a_h_x1': X[0][0][0],
    'eye_a_h_y1': X[0][0][1],
    'eye_a_h_x2': X[0][1][0],
    'eye_a_h_y2': X[0][1][1],

    'eye_a_v_x1': X[1][0][0],
    'eye_a_v_y1': X[1][0][1],
    'eye_a_v_x2': X[1][1][0],
    'eye_a_v_y2': X[1][1][1],

    'eye_b_h_x1': X[2][0][0],
    'eye_b_h_y1': X[2][0][1],
    'eye_b_h_x2': X[2][1][0],
    'eye_b_h_y2': X[2][1][1],

    'eye_b_v_x1': X[3][0][0],
    'eye_b_v_y1': X[3][0][1],
    'eye_b_v_x2': X[3][1][0],
    'eye_b_v_y2': X[3][1][1],

    'brow_a_h_x1': X[4][0][0],
    'brow_a_h_y1': X[4][0][1],
    'brow_a_h_x2': X[4][1][0],
    'brow_a_h_y2': X[4][1][1],

    'brow_a_v_x1': X[5][0][0],
    'brow_a_v_y1': X[5][0][1],
    'brow_a_v_x2': X[5][1][0],
```

```
    'brow_a_v_y2': X[5][1][1],

    'brow_b_h_x1': X[6][0][0],
    'brow_b_h_y1': X[6][0][1],
    'brow_b_h_x2': X[6][1][0],
    'brow_b_h_y2': X[6][1][1],

    'brow_b_v_x1': X[7][0][0],
    'brow_b_v_y1': X[7][0][1],
    'brow_b_v_x2': X[7][1][0],
    'brow_b_v_y2': X[7][1][1],

    'lips_a_h_x1': X[8][0][0],
    'lips_a_h_y1': X[8][0][1],
    'lips_a_h_x2': X[8][1][0],
    'lips_a_h_y2': X[8][1][1],

    'lips_a_v_x1': X[9][0][0],
    'lips_a_v_y1': X[9][0][1],
    'lips_a_v_x2': X[9][1][0],
    'lips_a_v_y2': X[9][1][1],

    'lips_b_h_x1': X[10][0][0],
    'lips_b_h_y1': X[10][0][1],
    'lips_b_h_x2': X[10][1][0],
    'lips_b_h_y2': X[10][1][1],

    'lips_b_v_x1': X[11][0][0],
    'lips_b_v_y1': X[11][0][1],
    'lips_b_v_x2': X[11][1][0],
    'lips_b_v_y2': X[11][1][1],

    'teeth_h_x1': X[12][0][0] if len(X) > 12 else '',
    'teeth_h_y1': X[12][0][1] if len(X) > 12 else '',
    'teeth_h_x2': X[12][1][0] if len(X) > 12 else '',
    'teeth_h_y2': X[12][1][1] if len(X) > 12 else '',

    'teeth_v_x1': X[13][0][0] if len(X) > 12 else '',
    'teeth_v_y1': X[13][0][1] if len(X) > 12 else '',
    'teeth_v_x2': X[13][1][0] if len(X) > 12 else '',
    'teeth_v_y2': X[13][1][1] if len(X) > 12 else '',
}
```