

Detekcija starosti i pola uz pomoć dubokog učenja

Djordje Karišić 657/2019 i Nikola Ivanović 637/2019

Jul 2022.

1 Uvod

Detekcija starosti i pola predstavlja jedno od ključnih polja i zadataka biometrije. Proces rešavanja ovog zadatka može dovesti do određenih problema:

1. Raznolikost ljudskih lica predlaže korišćenje velikog skupa podataka za treniranje
2. Problemi sa slikom-kvalitet kamere, osvetljenje, pozadina...

Kako bi se izbeglo treniranje modela nad velikim skupom slika, koje može potrajati dosta dugo u ovom projektu, korišćen je CAFFE framework, koji doprinosi projektu tako što obezbeuje neuronsku mrežu koja je prethodno istrenirana na računarima bržim od klasičnih.

Moguće je na ovaj problem gledati kao na problem klasifikacije (godine, pol) umesto na problem regresije. Ovim pristupom smanjen je uticaj anomalija navedenih u stavki 2.

Rešenje ovog zadatka predstavlja projektovanje softvera koji može jasno i precizno svrstati osobu u neku od predefinisanih klasa starosti ili pola uz pomoć slike njenog lica.

2 Implementacija

2.1 Ulazni podaci

Kako bi sistemu bilo omogućeno dobro okruženje za donošenje razumne pretpostavke, potrebno je transformisati ulazni podatak, tj. sliku tako da odgovara CAFFE modelu.

```

9      img=cv2.imread(img)
10     img = cv2.resize(img, (720, 640))
11     frame = img.copy()
12

```

Figure 1: Učitavanje slike

```

14     age_weights = "data/age_deploy.prototxt"
15     age_config = "data/age_net.caffemodel"
16     age_Net = cv2.dnn.readNet(age_config, age_weights)
17     gender_weights="data/gender_deploy.prototxt"
18     gender_config="data/gender_net.caffemodel"
19     gender_Net=cv2.dnn.readNet(gender_config,gender_weights)
20
21     ageList = ['(0-2)', '(4-6)', '(8-12)', '(15-20)',
22              |(25-32)', '(38-43)', '(48-53)', '(60-100)']
23
24     genderList=["Muski","Zenski"]
25     model_mean = (78.4263377603, 87.7689143744, 114.895847746)
26

```

Figure 2: Podešavanje parametara

Uz pomoć biblioteke openCV se čita ulazna slika, i nakon toga transformiše u odgovarajući oblik.

Nakon toga kreiraju se promenljive za čuvanje putanja datoteka modela, klase godina i polova, a zatim na liniji 25 se definišu promenljive modela. Promenljive modela bi trebalo imati definisane vrednosti kao na liniji 25.

2.2 Prepoznavanje lica

Kako bi se detektovalo lice na slici, koristi se funkcija *get_frontal_face_detector* iz biblioteke dlib. Ova funkcija ne prima argumente, a vraća prethodno istrenirani HOG+Linear SVM detektor lica.

Nakon toga, vrši se konverzija slike u Greyscale, i poziva se face detector sa argumentom koji predstavlja Greyscale verziju slike osobe. Ukoliko na slici ne postoji nijedno lice, korisniku će biti prikazana originalna slika sa tekstom da nije pronađeno nijedno lice.

```

34     # detekcija lica
35     face_detector = dlib.get_frontal_face_detector()
36     # grayscale
37     img_gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
38
39
40     faces = face_detector(img_gray)

```

Figure 3: Detekcija lica

```

43     if not faces:
44         mssg = 'Nema prepoznatog lica'
45         cv2.putText(img, f'{mssg}', (40, 40),
46                     cv2.FONT_HERSHEY_SIMPLEX, 2, (200), 2)
47         cv2.imshow('Greška', img)
48         cv2.waitKey(0)
49

```

Figure 4: Ukoliko na slici nema lica

2.3 Detekcija pola i starosti

U slučaju da je na slici pronadjeno barem jedno lice, prelazi se na drugi deo programa. Za svako lice na slici iscrtavamo žutu "kutiju" koja će obuhvatati to lice unutar sebe.

Za svako lice, tj. kutiju na slici, se pravi "blob" verzija, kako bi ista prosledila kao parametar neuronskoj mreži.

Rezultat `age-genderNet.forward` vraća listu brojeva koji govore o tome koliko je mreža sigurna da lice pripada toj klasi godina ili pola. Poziv funkcije `argmax` će vratiti indeks najvećeg broja iz liste, koji je ujedno i indeks klase kojoj lice najverovatnije pripada.

Uz pomoć `putText` funkcije moguće je dodati tekst na sliku, iznad kutija, koji govori o tome kojoj klasi godina i pola pripada lice sa slike. Slika, (Numpy ndarray) sa dodatim tekstom i kutijama se vraća kao izlaz funkcije.

```

51     else:
52
53         for face in faces:
54             x = face.left() # koordinata lica
55             y = face.top()
56             x2 = face.right()
57             y2 = face.bottom()
58
59             # kutija
60             box = [x, y, x2, y2]
61             Boxes.append(box)
62             cv2.rectangle(frame, (x, y), (x2, y2),
63                           (0, 255, 255), 2)
64
65         for box in Boxes:
66             face = frame[box[1]:box[3], box[0]:box[2]]
67
68             # ocitavanje lica
69             blob = cv2.dnn.blobFromImage(
70                 face, 1.0, (227, 227), model_mean, swapRB=False)
71
72
73             age_Net.setInput(blob)
74             age_preds = age_Net.forward()
75             age = ageList[age_preds[0].argmax()]
76             print(age_preds[0]*100)
77             gender_Net.setInput(blob)
78             gender_preds=gender_Net.forward()
79             val=gender_preds[0].argmax()
80             gender=genderList[val]
81             gender_confidence_score=gender_preds[0][val]
82
83             cv2.putText(frame, f'{msgg}{age}', (box[0],
84                                                  box[1] - 10),
85                          cv2.FONT_HERSHEY_SIMPLEX, 0.8,
86                          (0, 255, 255), 2, cv2.LINE_AA)
87             cv2.putText(frame, f'Pol:{gender}', (box[2]-130,
88                                                  box[3] + 20),
89                          cv2.FONT_HERSHEY_SIMPLEX, 0.8,
90                          (0, 255, 255), 2, cv2.LINE_AA)
91
92     return frame

```

Figure 5: Detekcija pola i starosti

3 Prikaz izlaznih podataka

Preko korisničkog interfejsa je potrebno izlaznu sliku prikazati korisniku. Ukoliko je izlaz funkcije 0, korisniku će biti prikazano da na slici nema lica, u suprotnom važi sledeće:

Izlazni ndarray se pretvara u nizove plave, zelene i crvene boje, uz pomoć kojih se kreira nova slika u funkciji `Image.fromarray`. Moguće je koristiti samo `Image.fromarray`, bez izvlačenja boja kako bi se iz ndarray-ja dobila slika.

Tkinter Label-u se u konstruktoru kao argument dodaje slika koju je potrebno prikazati, on se "pakuje", i prikazuje korisniku.

```
74     img=agedetector.doStuff(slika)
75     if len(img)==1 and img[0]==0:
76         prikaz=tk.Label(self.itemframe,text="Nema lica na slici")
77         prikaz.grid(row=2,column=3)
78     else:
79         #cv2.imshow("DA",img)
80         #img=cv2.imread(img)
81         img = cv2.resize(img, (360, 320))
82         blue,green,red = cv2.split(img)
83         img = cv2.merge((red,green,blue))
84         im = Image.fromarray(img)
85         imgtk = ImageTk.PhotoImage(image=im)
86
87         prikaz=tk.Label(self.itemframe,image=imgtk)
88         prikaz.image=imgtk
89         prikaz.grid(row=2,column=3)
90
```

Figure 6: Izlazna slika