

Ear Detection

Assignment #2

Image Based Biometrics 2020/21, Faculty of Computer and Information Science, University of Ljubljana

Mateo Kalem

Abstract—Namen moje domače naloge pri predmetu Slikovna Biometrija je bil razvoj detektorjev, ki bi uspešno detektirali ušesa bodisi preko slik bodisi preko spletne kamere. Ustvarjena sta bila v okolju Pycharm s pomočjo programskega jezika Python. Evalvacija detektorjev je bila narejena v orodju Matlab. Rezultati evalvacije so pokazali, da slikovni detektor deluje dovolj dobro, saj je dosegel povprečno 59.88 % prekrivanje, največje prekrivanje je bilo vidno pri sliki 0031.png, kjer je doseglo 77.87 %.

I. INTRODUCTION

Za drugo domačo nalogo pri predmetu Slikovna Biometrija je bilo potrebno ustvariti svoj lasten detektor ušes oz. uhljev. Zato sem se odločil, da bom ustvaril dva detektorja. Prvi bo detektiral ušesa s pomočjo spletne kamere, drugi bo detektiral ušesa na slikah iz neke podatkovne baze. Detektorja bo treba testirati in določiti, če dovolj dobro delujeta. Med izdelavo naloge sem si pomagal s spletnimi vodiči [1] (ang. tutorial), ki so mi pokazali osnove delovanja raznih detektorjev.

II. METHODOLOGY

Kot je bilo omenjeno zgoraj sem razvil dva detektorja, ki zaznava ušesa s pomočjo:

- Spletne kamere in
- slik znanih oseb, kjer se jim vidi eno ali oba ušesa.

Detektorja sem razvil v razvojnem okolju Pycharm s pomočjo programskega jezika Python. Uporabil sem knjižnico OpenCV, s katero sem opravil večino dela. Slike sem pridobil iz podatkovne zbirke AWE [2]. Prav tako sem imel na voljo dve Haar kaskadi za detekcijo levega in desnega ušesa. Detektorja sem razvil po načelu Viola-Jones algoritma, ta je "bruteforce" (izčrpen) algoritem za detekcijo določene modalnosti (ang. modality). Nato sem za evalvacijo v okolju Matlab naredil skripto, ki evalvira kvaliteto detektorja, ki preiskuje ušesa na slikah. Kvaliteto sem ocenil tako, da sem izračunal prekrivanje okvirjev (ang. bounding box overlap). Za izračun prekrivanja sem potreboval Matlov paket za računalniški vid. Vse skripte, ki sem jih razvil so vidne na mojemu GitHub repozitoriju [3].

III. RESULTS

Za detekcijo ušes je bilo potrebno nastaviti nekaj parametrov, za uspešno delovanje detektorja. Ta sta:

- `scaleFactor`: parameter, ki določa, za koliko se zmanjša velikost slike pri vsakem skaliranju slike.
- `minNeighbors`: parameter, ki določa, koliko sosedov bi moral imeti vsak kandidatni pravokotnik, da ga obdrži.

Drugi parametri, ki so še lahko določeni so `maxSize` in `minSize`, ki določata najmanjšo in največjo možno velikost objekta. Za detekcijo ušes s spletno kamero sem te parametre nastavljal na 1.2 za `scaleFactor` in 5 za `minNeighbors`. Rezultat te detekcije je viden na sliki 1.

Za detekcijo ušes prek slik sem imel drugačne parametre za vsako sliko. Te sem spreminjal zato, ker en set parametrov ne dela za vsako sliko. Te parametri prav tako vplivajo na kvaliteto detekcije, npr. če sem pri sliki 0008.png iz podatkovne zbirke

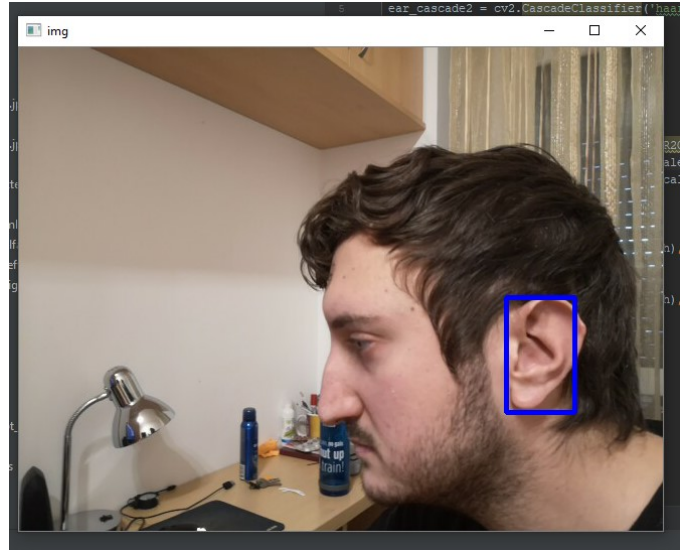


Figure 1. Detekcija ušes s pomočjo spletne kamere.

sem za parametra 1.05 in 2 dobil 64.52 % prekrivanje, če sem pa `scaleFactor` spremenil na 1.07 se je prekrivanje poslabšalo na približno 34.5 %. Za ostale slike sem dobil povprečno prekrivanje 59.88 %.

IV. CONCLUSION

Če povzamem, sem za detekcijo dobil precej dobre rezultate, saj je prekrivanje doseglo skoraj 60 %. Detektor s spletno kamero je prav tako dobro deloval, saj je zelo natančno detektiral ušesa, ko sem kamero premikal okoli svoje glave. Med testiranjem nisem opazil nobenih večjih napak pri detekciji. V bodoče bi lahko svoje programe razširil tako, da bi jih učil s pomočjo nevronske mreže, torej bi uporabil principe globokega učenja. Ker mi je zmanjkalo časa, mi pri tej nalogi tega žal ni uspelo implementirati, vendar, če bi se lotil tega, bi to implementiral v orodju Matlab.

REFERENCES

- [1] H. Valecha, "Ear biometrics — machine learning a little further...", <https://towardsdatascience.com/ear-biometrics-machine-learning-a-little-further-1839e5d3e322>, 2019.
- [2] as. dr. Žiga Emeršič, "Awe zbirka," <https://bit.ly/35eoe2h>, 2020.
- [3] M. Kalem, "Ibb ear detection," <https://github.com/Krlem/IBB-Ear-Detection>, 2020.