



Seminar iz biometričnih sistemov

Druga Domača naloga – Poročilo

Iva Eftimska

Ljubljana, 2022

1. Uvod

- Cilj domače naloge je bil spoznati problem verifikacije, pri čemer je potrebno za dan par slik ugotoviti ali gre za isto osebo ali ne. Uporabili smo dva različna načina: prvič z LPB deskriptorjem in mero podobnosti, drugič z HOG deskriptorjem in SVM.

2. Potek dela

- Program je napisan v Pythonu, izris ROC krivulj pa s pomočjo Matlaba. Pri prvem načinu je bilo potrebno inštalirati knjižnica OpenBR, potem uporabiti ukaz `br - algorithm FaceRecognition -compare slika_1.jpg slika_2.jpg`, ker za primerjavo parov slik izberemo tistih ki so navedeni v `pairsDevTest.txt`, in s pomočjo python skripte `DN2.py`, usvtarimo pipe ki omogoča komunikacijo med programa in terminalu, ustrezno izberemo parov slik od tekstovnem fajlu in na koncu se izvede ta ukaz v terminalu. Program izvede algoritem 4SF, vrne pa mero podobnosti med parov slik. Te vrednosti shranimo v `mera_podobnosti.txt` in v `oznake_ujemanje.txt` shranimo 1 pri parov slik ker gre za ista oseba, v tekstovnem fajlu so to prva polovica vrstice fajla in 0 ko ne gre za ista oseba. Potem ta dva tekstovna fajla uporabimo za risanje ROC krivulja v Matlabu. Pri drugem načinu uporabimo knjižnico `scikit-image` ker ima že implementiran hog deskriptor in pa knjižnico `scikit-learn` ker je implentiran algoritem SVM.

3. 4SF

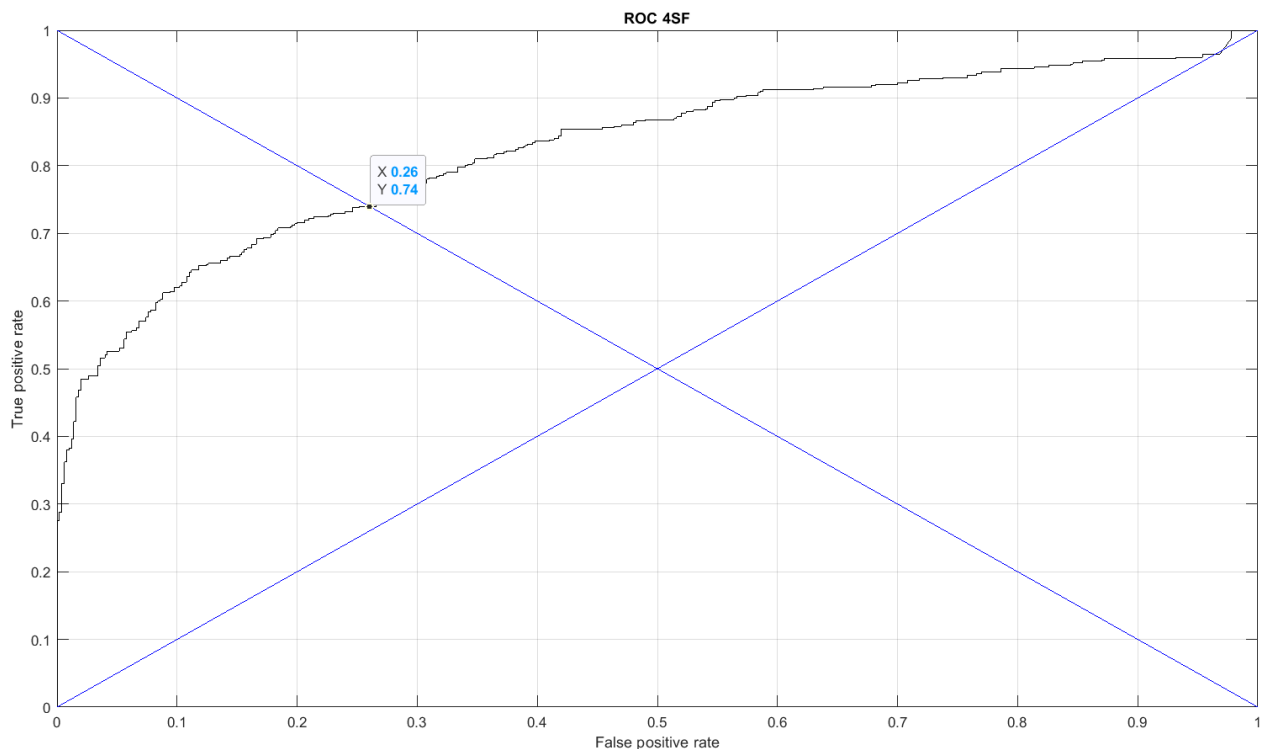
- Uporabimo slike iz zbirke LFW(Labeled Faces in The Wild)

Vprašanja

Vprašanje 1:

Kakšna je napaka enakega razvrščanja (angl. equal error rate - EER) za algoritem 4SF na tej bazi in pri izbranem testu? EER je operacijska točka na ROC krivulji, kjer imata napaki FAR in FRR enako vrednost.

- Linija ker ima FAR in FRR ista vrednost se seka z ROC krivuljo v točko (0.26; 0.74). Torej ta točka je EER.



Slika 1: ROC krivulja za algoritem 4SF

krivulje je 0.8245(1 je idealno).

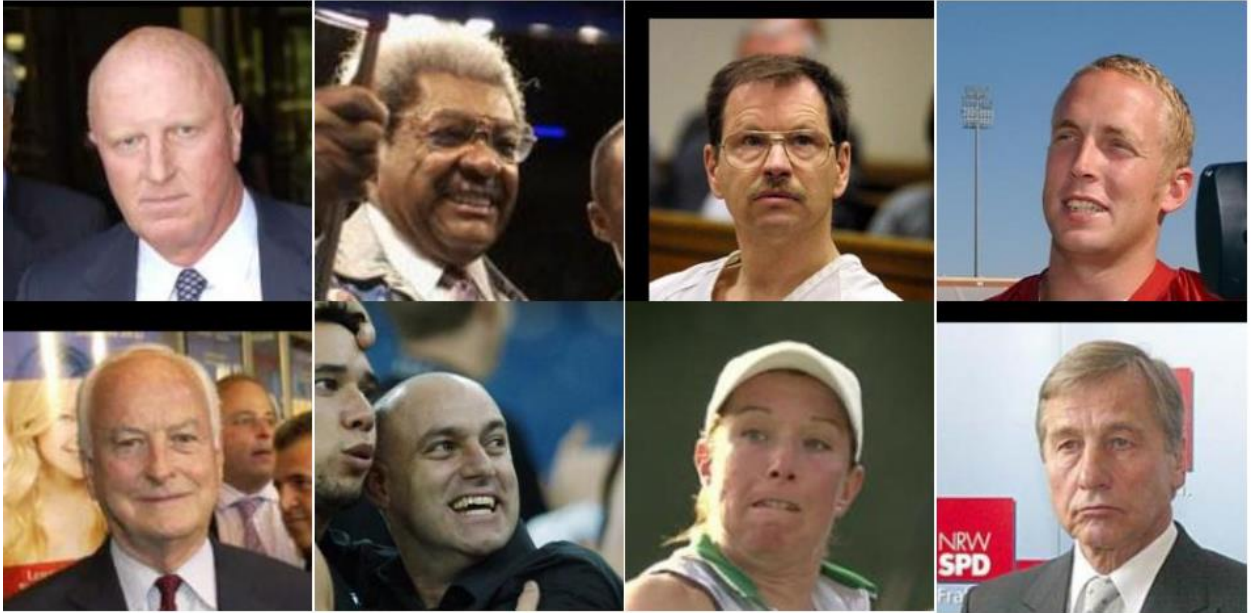
Vprašanje 2:

Kakšne napake dela algoritem 4FS pri parih, ki se v resnici ujemajo in kakšen pri parih, ki se naj ne bi ujemali? Podajte nekaj vizualnih primerov. Na podlagi informacij, ki ste jih pridobili iz članka, ki opisuje algoritem 4SF, ocenite (oz. podajte vaše mnenje) kaj je razlog tem napakam.

-



Slika 2: Parov slik ki jih ne pozna da gre za ista oseba(kolone so pare)



Slika 3: Parov slik ki jih pozna da gre za ista oseba, dejansko pa ni ista oseba(kolone so pare)

Iz slika 2 lahko opazimo da ko je obraz za nek kot zarotiran, program br ga ne pozna da gre za ista oseba, sam algoritem ima takšne težave. Iz slike 3 pa se vidi da jih opazi kot ista oseba dejansko pa ne gre za ista oseba, na podlagi tega da imajo podobne grimase, ker je spet napaka v algoritmu.

4. SVM

- Tukaj je bila uporabljena podatkovna zbirka LFWa(Labeled Faces in The Wild alligned), ker so slike poravnane tako da glavne dele obraza(npr. oči) se pojavljajo na isto mesto na vseh slik. Potrebno je bilo slike izrezati tako da so vidljivi samo dele tam kjer je obraz, to sem naredila z nekaj poiskusov, sem izbrala ročno najboljše koordinate kako sliko odrezati. HOG deskriptor, pomaga tako da ker dobimo histogram of oriented gradients, z vizualizacijo hog deskriptorje dobimo sliko ki nam prikaže obliko/izgled obraza, ker hog deskriptorje

pokažejo kje na sliki imamo večja/manjša magnituda/amplituda gradienta in smer gradienta, ker približno obliko obraza je podobna, hog deskriptorje lahko damo v nek algoritem za razvrščanje/klasifikacije.

Deskriptorje za parov slik ki so ista/različna oseba sem združila v en ist vektor in potem sem tisti vektor uporabila pri učenju SVM algoritma. Za učenje sem uporabila slike ki so podane v pairsDevTrain.txt(torej tudi te deskriptorje), tako da so bili označeni za kateri par slik gre/ne gre za isto osebo, testiranje je pa uporabljeno na parov slik iz pairsDevTest.txt.

Vprašanja

Vprašanje 3:

Predstavite postopek izračuna HOG deskriptorjev, ki ste ga uporabili (vrednost parametrov, uporaba prekrivanja, normiranje, način izreza obraznega področja, ...) ter postopek učenja SVM razvrščevalnika (kaj so bili vhodni podatki za učenje, kakšno vrsto normiranja ste uporabili, ...).

- Vrednosti parametrov pri hog deskriptorju sem določila hevristično, večinoma sem pustila default vrednosti, orientations sem povečala iz 9 na 12, tako da histogram posameznih celic ima 12 binov, večje št. binov omogoča prikaz bolj fine orientacije(recimo če je kak del obraza, npr. usta mal izkrivljena ali pokrita, z več binov lahko nekak približno dobimo točno obliko uste). Gradientni kot je v mojem primeru v rangi $[0, 180]$, torej imamo samo pozitivne kote. Negativne kote pa omogočajo vidljiv prikaz smeri iz svetlejšem k temnejšem področje in obratno.

Pixels_per_cell je (8,8) večje vrednosti omogočajo zaznavo bolj large-scale detaljev, zgubijo se pa detailje small-scale, zato sem pustila default vrednost.

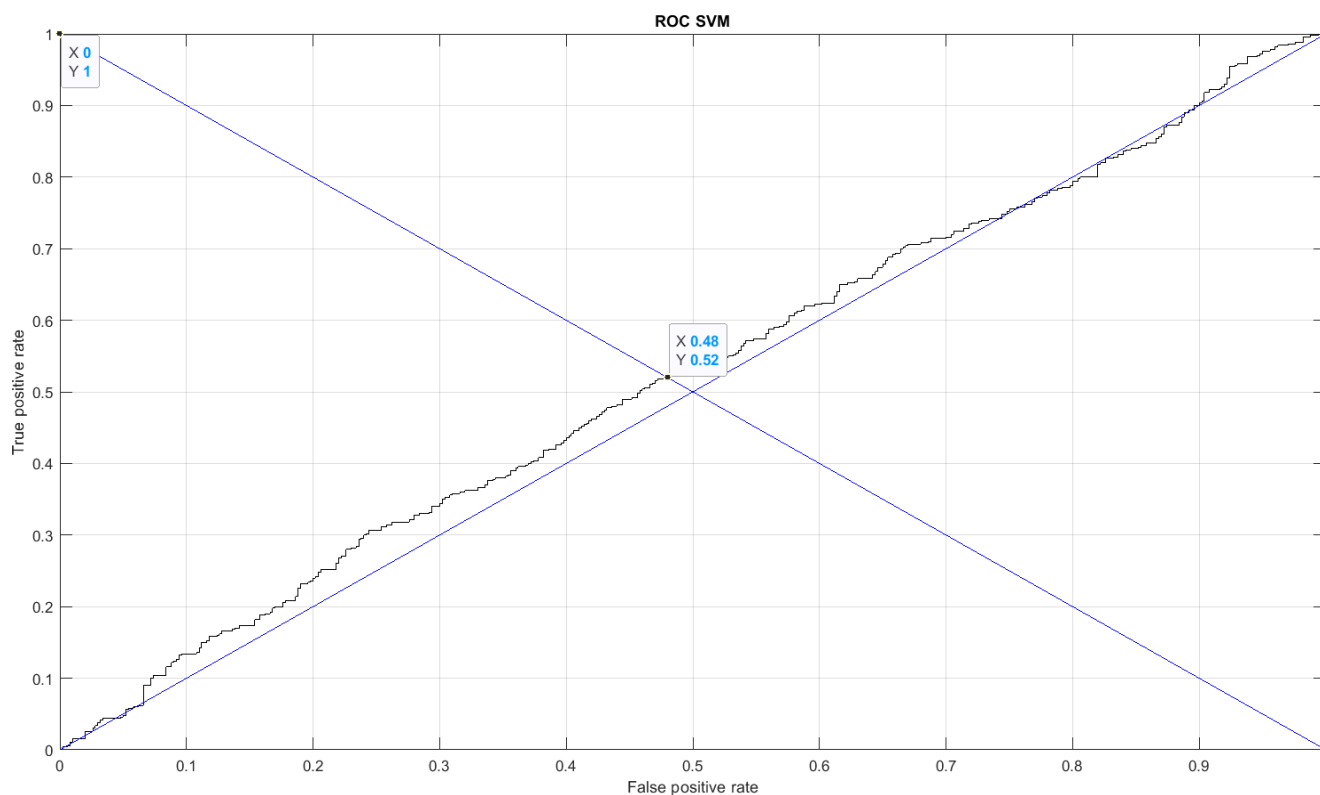
Cells_per_block je zelo pomemben parameter, ker s pomočjo tega grupiramo celice v bloke, v tem primeru blok velikosti (2,2), tako da potem naredimo normalizacijo gradientov, ker se znebimo efekta spremembe histograme zaradi različne svetlosti/kontrasta, omogočimo tudi prekrivanje $\frac{1}{2}$ celic enega bloka z drugega bloka, to omogoča da se pol v združen vektor deskriptorjev, celice pojavljajo večkrat samo da so normalizirani z različna vrednost, to daje rezultat da dobimo lokalno invariantost na spremembo svetlosti/kontrasta. Večje vrednosti blok velikosti 'zadušijo' lokalno invariantnost na svetlost/kontrast. Za normiranje bloka sem uporabila L2-Hyst, ker sem to našla v eni literaturi da daje dobre rezultate pri detekcije obrazov. Velikosti združenega deskriptorja je odvisna od orientations, pixels_per_cells, cells_per_block, recimo ko je pixels_per_cells večji, dobimo tudi manjši združen deskriptor, ker imamo manj celice slike, ker piksli na celico povečani. Vrednosti ki jih je vrnil HOG so bili med 0 in 1 tako da sem potem lahko uporabila združen deskriptor(dobljen z slike iz pairsDevTrain.txt) za učenje SVM algoritma. Pri SVM algoritmu sem uporabila linearno jedro in pri testiranju sem dobila točnost 52%. Vrsto jedra definira vrsto hyperplane, kakšna bo oblika hyperplane pri razvrščanju deskriptorje parov slik ali gre/ne gre za ista oseba. Velike vrednosti parametra C, zmanjšajo napake pri razvrščanju, razdalja od ločilne meje do obeh razredov je lahko manjša, samo pa dobimo več točno klasificiranih podatkov. Manjše vrednost C-ja pa naredi ravno obratno. Obstaja možnost da nastavljamo parametre funkcije svm čim bolj glede na podano podatkovno zbirko z GridSearchCV, tako da potem dobimo boljša točnost, tega

pa nisem implementirala zaradi računske potratnosti. Za risanje ROC krivulje, sem kot mere ujemanje uporabila, razdaljo do ločilne meje.

Vprašanje 4:

Kakšna je napaka enakega razvrščanja (angl. equal error rate - EER) za preizkušen algoritem na tej bazi in pri izbranem testu?

- EER je (0.48; 0.52), AUC pa 0.5227.



Slika 4: ROC krivulja za SVM

Vprašanje 5:

Kakšne napake dela algoritem pri parih, ki se v resnici ujemajo in kakšen pri parih, ki se naj ne bi ujemali? Podajte nekaj vizualnih primerov. Na podlagi znanja o HOG deskriptorjih in SVM razvrščevalnikih podajte vaše mnenje kaj je razlog tem napakam.



Slika 5: Parov slik ki jih ne pozna da gre za ista oseba(kolone so pare)



Slika 6: Parov slik ki jih pozna da gre za ista oseba, dejansko pa ni ista oseba(kolone so pare)

- Pri sliki 5 vidimo da ne zazna da gre za isto osebo ko je obraz mal zarotiran, ker HOG deskriptorje niso invariantni na rotacijo, če je mal zarotiran obraz, pol dobimo drugačne histograme, posledično drugačne vrednosti deskriptorjev in pol na podlagi te vrednosti učimo SVM algoritem, ker se vrednosti precej drugačni od deskriptorjev ko obraz ni zarotiran, pol se algoritem nauči da to ni ista oseba. Tudi algoritem SVM ne dela dobro ko imamo veliko podatkovno zbirko, takšna je naša zbirka. Možen razlog zaradi katerega pride do napačno ujemanje oseb, je ker HOG deskriptorje lahko vrnejo vrednosti ki se za posameznih parov slik mal prekrivajo in potem SVM težko razloči ali gre/ne gre za ista oseba.

Vprašanje 6:

Kateri od obeh preizkušenih algoritmov se je izkazal kot bolj učinkovit na tej zbirki?

- Na tej zbirki je bolj učinkovit algoritem 4SF, ker je mogoče razlog to da je knjižnica OpenBR bolj naučena/namenjena klasifikaciji ali gre za isto osebo/ali ne, algoritem SVM pa za druge probleme klasifikacije(bolj splošne) ne dejansko ravno za obraze. Pa tudi to da SVM dela boljše pri manjša velikost zbirke.

Vprašanje 7:

Če bi bili sposobni razviti boljši razpoznavalnik obrazov, na katere težave razpoznavanja se bi osredotočili?

- Bolj bi se osredočila da naprej se uporabi bolj sofisticirano metodo za detekcijo obrazov, da bo invariantna na poza, grimase, svetlost/kontrast, prekrite dele obraza, tako da bo vsak obraz mogel zaznati, pa potem še izbrati primerno metodo za izločanju deskriptorjev ki bodo orientacijski, geometrijski, na skalo invariantni. In na koncu izbrati še dober klasifikacijski algoritem ki bo mogel vse dobro razvrščati glede na dano podatkovno zbirko.