

UNIVERZA V LJUBLJANI

FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Matic Vrščaj

Avtomatsko potrjevanje profilne slike uporabnikov

DIPLOMSKO DELO

VISOKOŠOLSKI STROKOVNI ŠTUDIJSKI PROGRAM PRVE
STOPNJE RAČUNALNIŠTVO IN INFORMATIKA

Ljubljana, 2017

UNIVERZA V LJUBLJANI

FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Matic Vrščaj

Avtomatsko potrjevanje profilne slike uporabnikov

DIPLOMSKO DELO

VISOKOŠOLSKI STROKOVNI ŠTUDIJSKI PROGRAM PRVE
STOPNJE RAČUNALNIŠTVO IN INFORMATIKA

MENTOR: viš. pred. dr. Borut Batagelj
Ljubljana, 2017

To delo je ponujeno pod licenco *Creative Commons Priznanje avtorstva-Deljenje pod enakimi pogoji 2.5 Slovenija* (ali novejšo različico). To pomeni, da se tako besedilo, slike, grafi in druge sestavine dela kot tudi rezultati diplomskega dela lahko prosto distribuirajo, reproducirajo, uporabljajo, priobčujejo javnosti in predelujejo, pod pogojem, da se jasno in vidno navede avtorja in naslov tega dela in da se v primeru spremembe, preoblikovanja ali uporabe tega dela v svojem delu, lahko distribuira predelava le pod licenco, ki je enaka tej. Podrobnosti licence so dostopne na spletni strani creativecommons.si ali na Inštitutu za intelektualno lastnino, Streliška 1, 1000 Ljubljana.



Izvorna koda diplomskega dela, njeni rezultati in v ta namen razvita programska oprema je ponujena pod licenco *GNU Lesser General Public License*, različica 3 (ali novejša). To pomeni, da se lahko prosto distribuira in/ali predeluje pod njenimi pogoji. Podrobnosti licence so dostopne na spletni strani <http://www.gnu.org/licenses>.

Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Tematika naloge:

IZJAVA O AVTORSTVU DIPLOMSKEGA DELA

Spodaj podpisani Matic Vrščaj sem avtor diplomskega dela z naslovom:

Računalniško zaznavanje in potrjevanje profilnih slik

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal samostojno pod mentorstvom viš. pred. dr. Boruta Batagelja,
- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela,
- soglašam z javno objavo elektronske oblike diplomskega dela na svetovnem spletu preko univerzitetnega spletnega arhiva.

V Ljubljani, dne 13. februarja 2017

Podpis avtorja:

Kazalo

Poglavje 1	Uvod	8
Poglavje 2	Algoritmi	9
2.1	Uvod	9
2.2	Zaznavanje obrazov	9
2.2.1	Značilke Haar	9
2.2.2	Normalizirana razlika točk	12
2.3	Zaznavanje starosti in spola	14
2.3.1	Metoda lastnih obrazov (PCA)	14
2.3.2	Fisherjevi obrazi (LDA)	17
2.3.3	Lokalni binarni vzorci (LBP)	19
2.3.4	Metoda podpornih vektorjev (SVM)	21
2.3.5	Metoda najbližjih sosedov (k-NN)	24
Poglavje 3	Lastna implementacija	26
3.1.1	Uvod	26
3.1.2	Prevajanje knjižnice OpenCV	27
3.1.3	Priprava učne množice slik	27
3.1.4	Učenje algoritmov in prepoznava	30
3.1.5	Priprava obraza	33
3.1.6	Integracija z obstoječim sistemom	35
Poglavje 4	Sklepne ugotovitve	37
4.1.1	Predstavitve rezultatov	37
4.1.2	Možnosti za izboljšave	39

Seznam uporabljenih kratic

kratica	angleško	slovensko
LBP	Local binary patterns	Lokalni binarni vzorci
NPD	Normalized pixel difference	Normalizirana razlika pikslov
JNI	Java native interface	Izvirni Java vmesnik
PCA	Principle component analysis	Analiza glavnih komponent
LDA	Linear discriminant analysis	Linearna diskriminantna analiza
LBP	Local binary patterns	Lokalni binarni vzorci
SVM	Support vectors machine	Metoda podpornih vektorjev
k-NN	k-nearest neighbours	k najbližjih sosedov
CNN	Convolutional neural network	Konvolucijska nevronska mreža

Povzetek

Namen te diplomske naloge je bil raziskati področje računalniškega vida, ki bi se izkazal za koristnega pri avtomatskem potrjevanju profilnih fotografij informacijskega sistema. Za primer smo uporabili spletni portal, ki uporabniku omogoča nalaganje lastne profilne slike. V prvem delu smo razložili uporabne algoritme za zaznavanje obraza, starosti in spola. V drugem delu pa je bila osrednja tema implementacija lastne rešitve v Javi, ki algoritmično predlaga primernost slike na podlagi zaznanih značilk obraza – starosti in spola. Slike lahko označimo kot primerno, če je na njej natančno en obraz polnoletne osebe, je njena ločljivost dovolj velika in ni zamegljena. Z detekcijo zamegljenosti slike se v tem delu nismo obremenjevali, osredotočili pa smo se na zanavo obraza, oči, spola in starosti. Na koncu smo izračunali, s kolikšno verjetnostjo algoritem ugotovi primernost naključne uporabnikove slike, ter razpravljali, kako bi postopek avtomatskega potrjevanja lahko še izboljšali.

Ključne besede: računalniški vid, detekcija obraza, zaznava spola, zaznava starosti

Abstract

The purpose of this study was to explore areas of computer vision, which may prove useful in automatic profile picture validation in Information Systems. For practical example, we used a web portal that enables users to upload their own profile pictures. In the first part we explain useful algorithms for face detection, age and gender. The second part focuses on implementation of our own solution in Java, which uses explored algorithms to suggest the appropriateness of the image based on the detected facial features - age and gender. An image can be marked as appropriate, if it contains exactly one face of an adult and is not blurred. This work overlooks image quality but focuses on face detection. Finally, we calculate success rate, the likelihood that the algorithm can determine the suitability of random user image, and discuss how our approach could be improved.

Keywords: computer vision, face detection, gender detection, age detection

Poglavje 1 Uvod

V obdobju pisanja diplomske naloge sodelujemo pri razvoju spletnega spoznavnega portala Ona-on [1]. Na njem uporabniki izpolnijo svoj profil, naložijo profilno sliko ter spoznavajo druge uporabnike in uporabnice portala. Kakovost storitve lahko zagotovimo le tako, da vse naložene slike preveri administrator. Pomembno je, da vsebinsko neprimerne slike zavrremo, saj take slike niso redke. Zavrremo tudi slike, ki vsebujejo mladoletne osebe ali pa take, ki so splošno nekvalitetne (premajhna resolucija, preveč šuma, ipd.).

Storitev pridobiva na številu uporabnikov, administratorji imajo več in več dela s potrjevanjem slik, potreba po rešitvi, ki bi jim olajšala delo pa postaja nujna. Očitna rešitev bi bila zaposliti več administratorjev, vendar pa menimo, da ta rešitev ni dovolj razširljiva in morda celo predraga. Proces potrjevanja je potrebno avtomatizirati in s tem dolgoročno razbremeniti njihovo delo.

V prvem delu diplomske naloge se osredotočamo na uporabne algoritme, ki bi jih pri reševanju opisanega problema potencialno lahko uporabili. Skušamo jih podrobno razumeti in poenostavljeno razložiti. Nekatere od predstavljenih algoritmov smo uporabili pri lastni implementaciji rešitve. Sledi drugi del naloge, kjer lastno implementacijo podrobno opišemo. Ta zajema pripravo učne množice ogromne količine slik, pridobljenih s spletnega portala – uporabimo jih kot učno množico. Nadaljujemo z opisom priprave posamične slike, zaznavo obraza ter morebitnih pomembnih značilnk na njem in prepoznavo spola in starosti osebe na podlagi zaznanega obraza. Slike, pri katerih so algoritmi neuspešni dodelimo administratorju, ki jih ročno potrdi ali zavrne.

Na koncu naloge opišemo glavne ugotovitve, izračunamo uspešnost lastne implementacije ter razmišljamo, kako bi lahko rešitev še izboljšali.

Poglavje 2 Algoritmi

2.1 Uvod

Avtomatizirati skušamo potrjevanje profilnih slik, ki jih uporabniki nalagajo v sistem. Prepoznati želimo obraz, nato pa starost in spol osebe na sliki. Na podlagi teh prepoznav se odločimo ali je slika primerna. V tem poglavju so za primerjavo poleg uporabljenih algoritmov opisani tudi nekateri, ki jih nismo uporabili.

2.2 Zaznavanje obrazov

V nadaljevanju so opisani algoritmi za zaznavanje obrazov. Pri lastni implementaciji smo uporabili prvega, to je Viola-Jones.

2.2.1 Značilke Haar

Algoritem največkrat imenujemo kar Viola-Jones, po avtorjih Paul Viola in Michael J. Jones, ki sta ga prva predstavila leta 2001 [2]. Avtorja sta razvila svojevrsten algoritem za zaznavo objektov na sliki, predvsem pa ju je zanimalo zaznavanje obrazov. V primerjavi z istočasnimi se je izkazal za hitrega in natančnega, najbolj pa je učinkovit pri zaznavanju frontalnih obrazov.

Sliko razdelimo na manjše kvadratne odseke – okna, v njih pa iščemo značilke obraza. Okna izbiramo od leve proti desni in od zgoraj proti dol s korakom veliko manjšim od velikosti okna. Ko pridemo do desnega spodnjega roba, okno nekoliko zmanjšamo in ponovimo celoten postopek (Slika 1).



Slika 1: Okno iskanja se z majhnimi koraki pomika od leve proti desni in od zgoraj navzdol. Postopek iskanja se ponavlja, vsakokrat z manjšim oknom.

Avtorja sta predstavila tri ključne elemente algoritma:

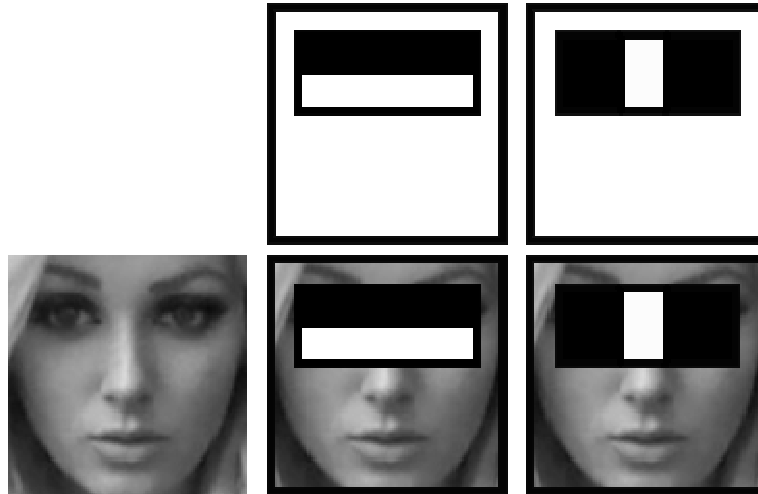
- Integralna slika,
- Klasifikator zgrajen iz majhnih pomembnih značilk,
- Kaskadno zaporedje klasifikatorjev

Integralna slika

Za vsako točko na sliki na začetku izvajanja algoritma izračunamo vsoto vrednosti vseh točk, ki ležijo levo in navzgor od trenutne. Izračun si za vsako točko shranimo. Na ta način lahko med izvajanjem detekcije učinkovito izračunamo vsoto vrednosti točk poljubnega kvadrata na sliki. Ključno je da do teh vsot lahko med iskanjem dostopamo v konstantem času, saj klasifikatorji naredijo mnogo primerjav vsot intenzitet točk med določenimi pari pravokotnikov v iskalnem oknu.

Učinkovit klasifikator, ki je zgrajen z izbiro majhnega števila pomembnih parov pravokotnikov t.i. šibkih značilk iz ogromnega nabora potencialnih s pomočjo AdaBoost [3]. Na sliki 2 lahko vidimo primerjavo območja oči z območjem pod njim. Zgornji mora biti temnejši od spodnjega, da ne ovržemo možnosti obstoja obraza v oknu. Na sliki sta prikazani dve najbolj učinkoviti značilki, število potencialnih pa je ogromno, veliko več kot točk na sliki. Za hitro zaznavo

moramo izbrati samo najbolj relevantne. Majhen nabor pravilno izbranih značilk nam prinese večino natančnosti.



Slika 2: Dve zelo učinkoviti šibki značilki kaskadnega detektorja

Tretji ključni element algoritma je **kaskadno zaporedje** čedalje bolj kompleksnih značilk. Najprej preverjamo ustrežanje tistih, ki najbolj pogosto ovržejo detekcijo obraza. Tako imamo največ možnosti, da v oknu, kjer obraza ni, to zaznamo zgodaj in ne izgubljam več procesorskega časa. Ko naletimo na okno, ki vsebuje obraz ali nekaj obrazu podobnega, pridejo na svoj račun kompleksnejši klasifikatorji, ki skrbijo za natančnost detekcije in ločijo obraz od npr. električne vtičnice. Med testiranjem je na eni od slik algoritem namesto obraza zaznal električno vtičnico na steni.

2.2.2 Normalizirana razlika točk

Shengcai Liao je v svojem algoritmu na nivoju točk vpeljal predstavitev normalizirane razlike točk (ang. Normalized Pixel Difference - NPD). NPD se izračuna kot razmerje razlik vrednosti dveh poljubnih točk proti njuni vsoti [4], kot kaže slika 3.



Slika 3: Primer izračuna normalizirane razlike dveh točk. $NPD = \frac{132 - 88}{132 + 88} = 0.2$

Ideja je zelo podobna, kot jo je imel psiholog E. H. Weber leta 1834. Weber je trdil, da je minimalen dražljaj, ki ga še zaznamo, v primerjavi z intenziteto ozadja konstanta. Torej zaznavamo razliko intenzitet, ne pa absolutno intenziteto dražljaja [5].

NPD ima kar nekaj zaželenih lastnosti: pričakovano se obnaša pri raztezanju slike, je omejen – obstaja končna največja in najmanjša vrednost, ima možnost rekonstrukcije originalne slike, poleg tega pa ga je enostavno izračunati.

Koncept zaporednih povezav je zelo preprost. Primerja svetlost dveh regij slike in kodira rezultat z 1, če je prva regija svetlejša ali z 0 v nasprotnem primeru. P. Sinha, profesor računalniškega vida na MIT, je pokazal, da zaporedne povezave lahko predstavljajo notranjo strukturo

predmetov, npr. človeški obraz in so neobčutljive na spremembe svetlobe in odporne na šum v sliki [6]. Namesto kodiranja zaporednih povezav slikovnih območij, lahko to naredimo na točkah, izračunanih s pomočjo NPD. Podmnožice NPD značilk avtomatično naučimo in združimo v regresijska drevesa. Tako potrebujemo en sam AdaBoost klasifikator za zaznavo poljubnega obraza z motnjami (motnje so lasje, roka ali kaj drugega kar lahko prekriva del obraza). Zaznava deluje za obraze različnih kotov, ne da bi potrebovali dodatno označevati in sortirati vhodne učne podatke.

V učno množico vstavimo slike z obrazi in slike, ki ne vsebujejo obrazov. Regresijska drevesa poiščejo šibke klasifikatorje za razlikovanje. Vsak tak šibek klasifikator doprinese nekaj malega k verjetnosti, ali je na sliki obraz ali ne. Šibki klasifikatorji skupaj tvorijo močan robusten klasifikator obrazov, odporen na nepredvidljiva ozadja in motnje.

2.3 Zaznavanje starosti in spola

V tem poglavju so opisani algoritmi, ki so primerni za klasifikacijo spola in starosti. Vsi se primarno uporabljajo za prepoznavo obrazov, na to temo je tudi največ raziskav. Namesto različnih oseb, pa bomo v učno množico vstavili npr. slike obeh spolov in algoritme naučili, kje so različne značilke med spoloma.

Zaznavo starosti bomo omejili na intervale, vsak od naštetih pa nam bo predstavljal po en razred:

- Od 18 do 35 let – starostna skupina 1
- Od 36 do 55 let – starostna skupina 2
- Od 56 let navzgor – starostna skupina 3

Starosti bi lahko razdelili na več manjših skupin, vendar pa je ugibanje starosti težka naloga že za človeške možgane, rezultati so bili toliko slabši, da smo se odločili omejiti ločevati le med tremi skupinami.

Za namene avtomatskega potrjevanja profilne slike uporabnikov smo sicer želeli ločevati le med dvema skupinama – mladoletna in polnoletna oseba. Avtomatizacija izločevanja slik, ki vsebujejo mladoletne osebe bi bil dober dosežek, vendar pa zaradi pomanjkanja slik mladoletnih oseb zaznavo mladoletnosti nismo uspeli avtomatizirati. Namesto tega smo se odločili raziskati področje zaznave starosti tako, da smo ločevali med tremi opisanimi starostnimi razredi.

2.3.1 Metoda lastnih obrazov (PCA)

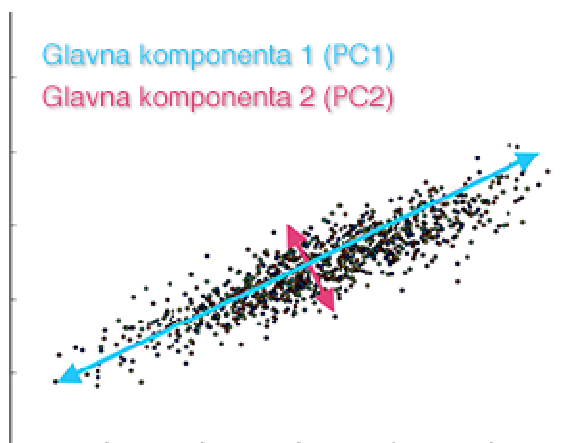
Izraz lastni obrazi je prevod iz angleščine (ang. Eigenfaces), tudi v angleščini pa je beseda eigen le tujka iz nemščine, ki pomeni lasten. Izraz prihaja iz matematike, lastni vektor (ang. Eigenvector). Gre za vektor, ki pri določeni operaciji vrne skalarni večkratnik samega sebe.

Idejo prepoznave človeških obrazov s pomočjo analize glavnih komponent (ang. Principal component analysis – PCA) sta razvila L. Sirovich in M. Kirby leta 1987 [7]. Nekaj let kasneje

pa sta svojo skoraj realno-časovno programsko rešitev za prepoznavo obrazov predstavila M. Turk in A. Pentland [8].

V matematiki lahko poljubni vektor \vec{v} izrazimo s pomočjo baznih vektorjev tako, da vsakega od baznih vektorjev pomnožimo z utežjo in zmnožke seštejemo. Metoda Eigenfaces dela enako s slikami. Iz učne množice slik obrazov izračunamo povprečno sliko, ki jo nato odštejemo vsaki učni sliki. Tako dobimo slike, ki predstavljajo odstopanja od povprečja. Iz njih in s pomočjo analize glavnih komponent izračunamo bazne slike, ki najbolj pokrivajo raznolikost učnih slik. S pomočjo tako izračunanih baznih slik pa lahko izrazimo vsako novo testno sliko obraza.

Analiza glavnih komponent (PCA) je statistični postopek pravokotnih transformacij, ki skuša zmanjšati dimenzionalnost problema tako, da izbira komponente, ki doprinesejo največ variance – najbolj odstopajo od povprečja. To pomeni, da z njegovo pomočjo izberemo take bazne slike, s katerimi bomo predstavili ostale slike s kar najmanj izgubami. Rezultat so t.i. glavne komponente (ang. Principal components), včasih jih imenujemo tudi lastni vektorji (ang. Eigenvectors). Prva glavna komponenta je vektor, ki predstavlja največjo možno varianco vhodnih vektorjev. Druga glavna komponenta predstavlja vektor z naslednjo največjo možno varianco, itd. Primer je predstavljen na sliki 4.



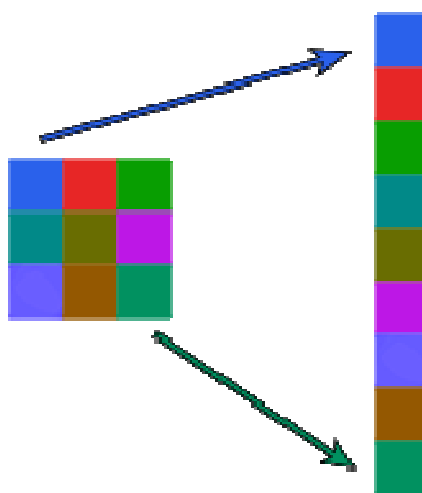
Slika 4: Primer prve in druge glavne komponente PCA. Črne točke predstavljajo vhodne podatke, puščici pa glavni komponenti.

S pomočjo PCA prepoznavo obraza začnemo tako, da vhodno sliko, ki smo ji že odšteli povprečno sliko, izrazimo z baznimi in dobimo vektor uteži. Izračunamo njegovo razdaljo z vsakim od vektorjev uteži, ki smo jih izračunali med učenjem za posamezne znane obraze. Izračunane razdalje sortiramo naraščajoče in tako dobimo rang ujemanj testnega obraza s posameznimi učnimi. Najkrajša razdalja pomeni najboljše ujemanje. Obraz, ki se najbolj ujema z našo testno sliko je torej najboljši približek naše testne slike, ki jo sedaj lahko klasificiramo z enakim razredom kot ga nosi najbolj ujemajoč znan obraz.

Včasih je lahko najboljše ujemanje še vedno slabo ujemanje, kar pomeni, da bi testno sliko klasificirali z veliko negotovostjo. To rešujemo tako, da uvedemo prag (ang. threshold), ki predstavlja omejitev največje razdalje. Kadar je izračunana razdalja večja od praga, to štejemo kot neujemanje testne slike z določeno učno. Kadar uporabljamo prag, moramo biti pozorni, saj lahko pride do primera, ko testnemu obrazu ne ustreza noben iz množice znanih obrazov.

Kako deluje PCA

Vhodne učne slike z obrazi pretvorimo v vektorje s toliko dimenzijami, kolikor točk vsebuje posamična slika, kot kaže Slika 5. Izračunamo povprečni vektor tako, da seštejemo istoležeče vrednosti vektorjev in rezultat delimo s številom seštevancev. Nato od vseh odštejemo izračunani povprečni vektor. Dobljeni vektorji predstavljajo odstopanja posamezne slike od povprečja.



Slika 5: Primer pretvorbe slike dimenzij 3x3 v vektor dimenzije 9

Predpostavimo, da imamo vhodne sivinske učne slike obrazov velikosti 100x100 točk. Vsak vektor, ki ga dobimo iz take slike ima P dimenzij, v našem primeru $P = 10^4$. Vektorje zložimo v matriko tako, da vsak zapolni en stolpec matrike. Imamo vhodno matriko podatkov za izračun glavnih komponent dimenzije $P \times N$, kjer je N število vhodnih slik.

Izračunamo kovariančno matriko C tako, da transponirano vhodno matriko množimo z originalno. Z njo bomo izračunali lastne vektorje in njihove lastne vrednosti. Kovariančna matrika je v našem primeru velikosti $P \times P$. Izračunamo matriki V in D tako da je $V^{-1}CV = D$.

Matrika V po stolpcih predstavlja lastne vektorje kovariančne matrike, matrika D pa je diagonalna matrika lastnih vrednosti. Obe dobljeni matriki sta prav tako velikosti $P \times P$. Imamo torej nabor lastnih vektorjev in za vsakega od njih lastno vrednost. Če se lastni vektor nahaja v stolpcu i matrike V, se njegova lastna vrednost nahaja na diagonali v stolpcu i ter vrstici i matrike D. Dobljene lastne vektorje uredimo po svojih lastnih vrednostih padajoče in dobili smo glavne komponente. Pozorni moramo biti, da ne pomešamo lastnih vrednosti. Lastni vektor z največjo lastno vrednostjo predstavlja prvo glavno komponento, naslednji predstavlja drugo glavno komponento, itd.

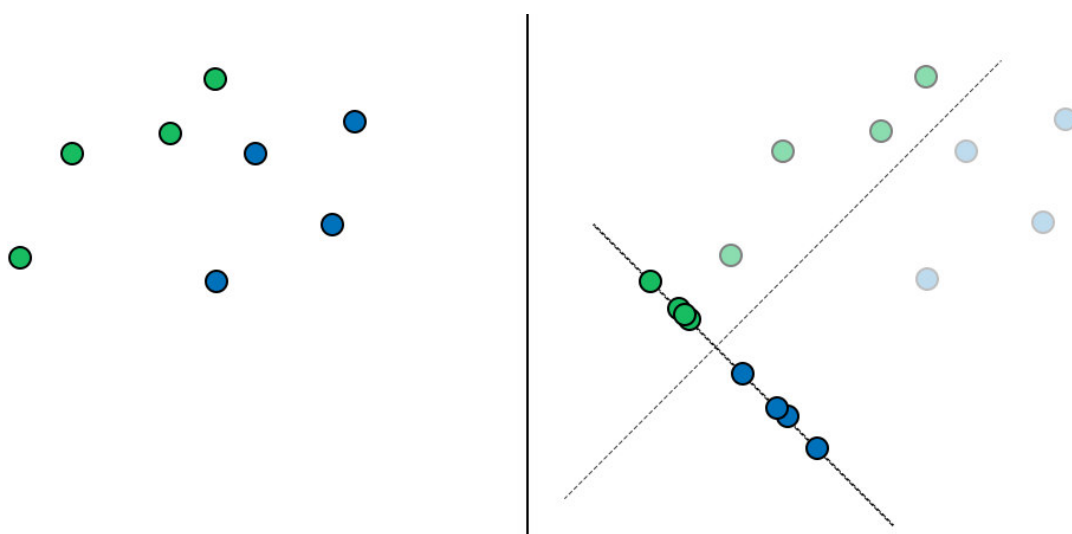
Izberemo poljubno število lastnih vektorjev oz. glavnih komponent z največjimi lastnimi vrednostmi. Večja kot je lastna vrednost vektorja, več variance predstavlja in bolj pomemben je iz stališča ohranjanja informacij. Izbrane lastne vektorje imenujemo bazni vektorji.

2.3.2 Fisherjevi obrazi (LDA)

R. A. Fisher je bil angleški statistik in biolog, ki je leta 1936 predstavil diskriminantno analizo (ang. Discriminant analysis) za napovedovanje ali so dane spremenljivke primerne za klasificiranje primerka v skupine [21]. Metoda fisherjevih obrazov (ang. Fisherfaces) temelji na linearni diskriminantni analizi (ang. Linear discriminant analysis - LDA), ki je izpeljana iz Fisherjeve diskriminantne analize.

Linearna diskriminantna analiza je tako kot PCA, metoda, ki se najpogosteje uporablja kot tehnika za zmanjševanje dimenzionalnosti vzorca pred začetkom klasifikacije. Cilj metode je

projecirati nabor podatkov na nižje-dimenzionalni prostor tako, da so ločnice med razredi čimbolj jasne. Z drugimi besedami, točke, ki sodijo v isti razred, želimo imeti čimbolj skupaj, vse točke različnih razredov pa v splošnem čimbolj narazen, tako kot kaže Slika 6. Ker zmanjšamo dimenzionalnost podatkov, se izognemo prezasičenosti podatkov, zmanjša pa se tudi računski strošek [9] [11].



Slika 6: Projeciranje točk dveh razredov. Iz dvodimenzionalnega prostora dobimo enodimenzionalnega.

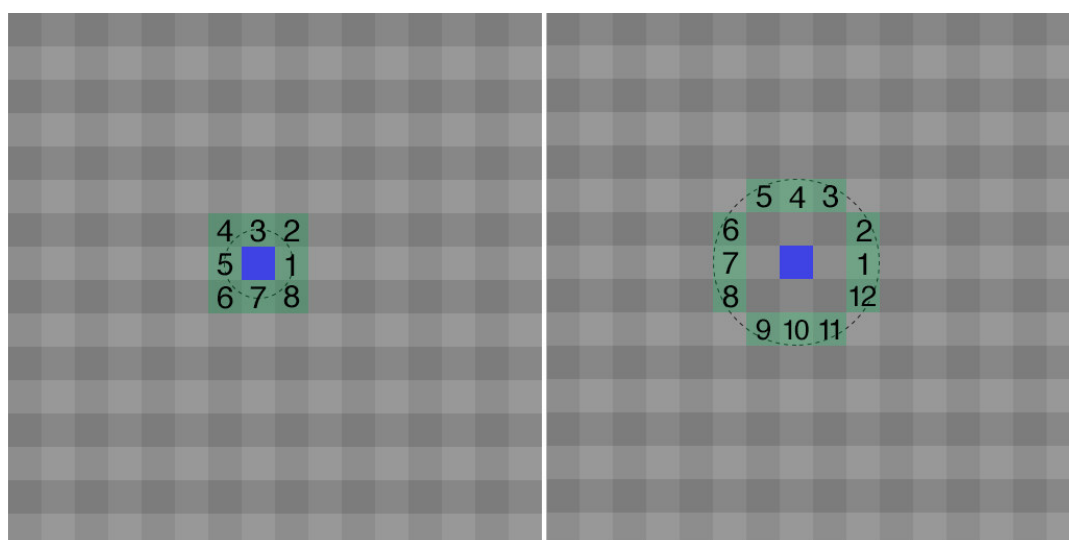
Ronald A. Fisher je oblikoval metodo linearne diskriminante leta 1936 za reševanje taksonomskega problema. Metoda je izjemno uporabna za klasificiranje podatkov. Prvotna metoda je opisana za ločevanje med dvema razredoma, a je bila kasneje posplošena in je kot taka primerna za reševanje problemov z več razredi [10].

2.3.3 Lokalni binarni vzorci (LBP)

Metoda lokalnih binarnih vzorcev (ang. Local binary patterns – LBP) je neparametrična metoda, ki učinkovito povzema lokalne strukture slike tako, da primerja vsako točko s soslednjimi [12] [13] [14].

Osnovni koncept

- Sliko razdelimo na manjše celice, npr 16x16 točk
- Za vsako točko v celici primerjamo P soslednjih, ki so od opazovane oddaljene za R točk.
- Če je intenziteta soslednje točke večja od opazovane, zapišemo bit 0, sicer 1. Tako dobimo P-bitno binarno število, ki ga shranimo za vsako točko v celici. Enako ponovimo za vse točke vseh celic. Na sliki 7 sta za lažjo predstavo prikazana dva primera.
- Nato izračunamo histogram ponavljanja vzorcev vsake od 2^P kombinacij za vsako celico.
- Histograme lahko normaliziramo.
- Histograme shranimo v vektor histogramov ki ga imenujemo prostorski histogram (ang. spatial histogram).



Slika 7: Primer izračuna LBP. Sivi kvadratici predstavljajo točke v celici, modra je opazovana, zelene pa so soslednje. (Levo: $R=1$ in $P=8$, desno: $R=2.5$ in $P=12$)

Tako imamo vektor vzorcev z dimenzijo 2^P . S pomočjo metode najbližjih sosedov, podpornih vektorjev ali katere druge metode strojnega učenja lahko sedaj klasificiramo slike. Obstaja veliko variacij in razširitev metode LBP [15] [16].

- OCLBP (ang. Over-complete LBP), kjer se celice prekrivajo med seboj,
- tLBP (ang. Transition LBP), kjer se vsaka sosednja točka primerja s prejšnjo, le prva se primerja s tisto na sredini,
- mLBP (ang. Modified LBP), kjer sosednje točke primerjamo s srednjo vrednostjo (ang. mean value) vseh točk v oknu.

Uniformni vzorci

Uporabna razširitev osnovnega LBP so tudi t.i. uniformni vzorci (ang. uniform patterns). Dejstvo, da se nekateri vzorci v slikah pojavljajo bolj pogosto kot drugi izkoristimo tako, da redke vzorce skupaj štejemo v en koš. S tem zmanjšamo velikosti vektorjev značilk in prostorsko ter časovno zahtevnost algoritma in morebitnih kasnejših algoritmov, ki primerjajo izračunane LBP histograme.

Pri izračunavanju histograma namesto štetja ponovitev vseh 2^P kombinacij štejemo v svoje stolpce le uniformne vzorce, vse neuniformne pa seštejemo v en stolpec. Vzorec je uniformni, kadar velja da vsebuje največ dva prehoda iz 0 v 1 ali iz 1 v 0. Za primer vzemimo vzorce za $P=8$. Vzorec z vrednostmi 00100000 je uniformen, saj vsebuje dva prehoda, medtem ko 00101011 ni uniformen, saj vsebuje 5 prehodov. Namesto 256 kombinacij jih imamo s pomočjo uniformnih vzorcev le 59.

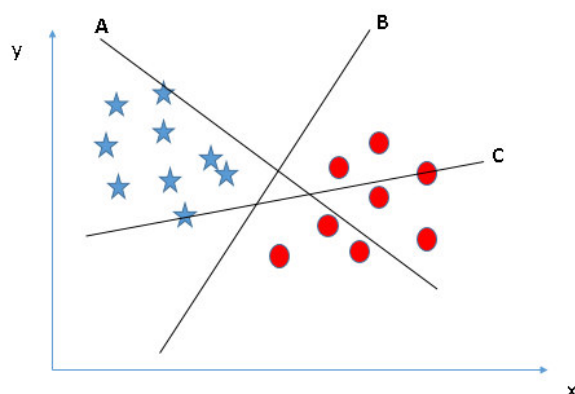
Uporabljena implementacija algoritma LBP iz knjižnice OpenCV ne uporabi uniformnih vzorcev.

2.3.4 Metoda podpornih vektorjev (SVM)

Metoda podpornih vektorjev (ang. Support vector machine – SVM) je matematičen postopek ločevanja primerov v razrede. Imamo npr. znane primere, za katere vemo v kateri razred sodijo. S pomočjo metode podpornih vektorjev lahko naučimo model, ki neznan primer klasificira v znan razred – npr. razred A ali razred B [17]. Za lažje razumevanje bomo opisovali ločevanje med dvema razredoma, vendar pa metoda odlično deluje tudi za več kot dva razreda.

Metoda SVM primere predstavi kot vektorje (x,y) v prostoru preslikane tako, da so primeri posamičnih kategorij razdeljeni z jasnimi, čim širšimi vmesnimi prostori. Nove primere nato razporedi v tem istem prostoru in napove v katero kategorijo spadajo, glede na to na kateri strani prostora se nahajajo.

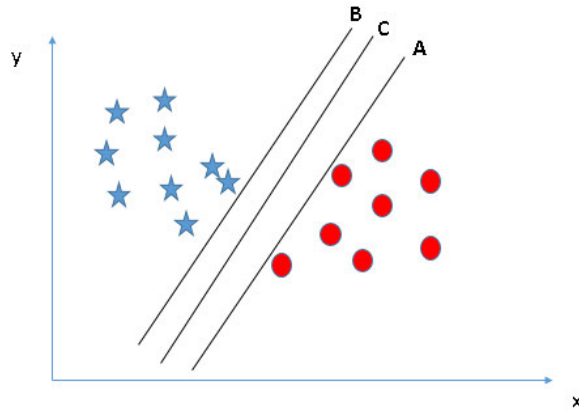
S pomočjo slik vzetih iz [22] bomo podrobneje opisali potek metode. Na sliki 8 vidimo primere dveh kategorij predstavljenih z zvezdicami in krogi. SVM skuša poiskati hiperravnino tako, da popolnoma loči vektorje obeh razredov med seboj, kadar je to mogoče. Na sliki 8 je izbrana hiperravnina B, ki izjemno lepo loči obe kategoriji.



Slika 8: SVM, iskanje hiperravnine, ki najbolj loči obe kategoriji.

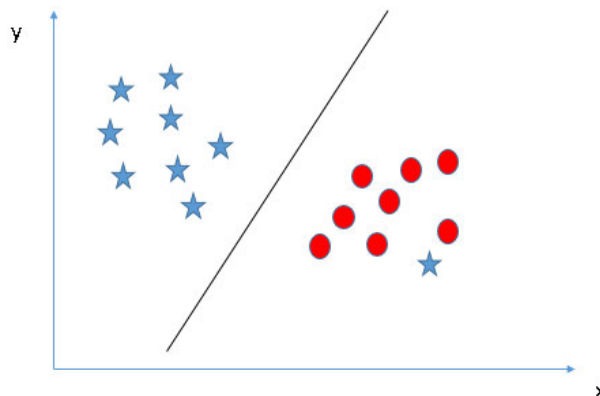
Število hiperravnin, ki popolnoma ločijo kategoriji je več. SVM izbere tisto, ki je najbolj oddaljena od primerov obeh množic. Na sliki 9 je to hiperravnina C. Izbor najbolj oddaljene

hiperravnine pomembno vpliva na robustnost klasifikacije. Morda očitno a pomembno dejstvo je, da je izbrana hiperravnina popolnoma določena z obrobni točkami obeh kategorij. Te imenujemo podporni vektorji, od tod tudi ime metode.



Slika 9: SVM izbere ločilno hiperravnino tako, da je čimbolj oddaljena od robnih točk obeh množic.

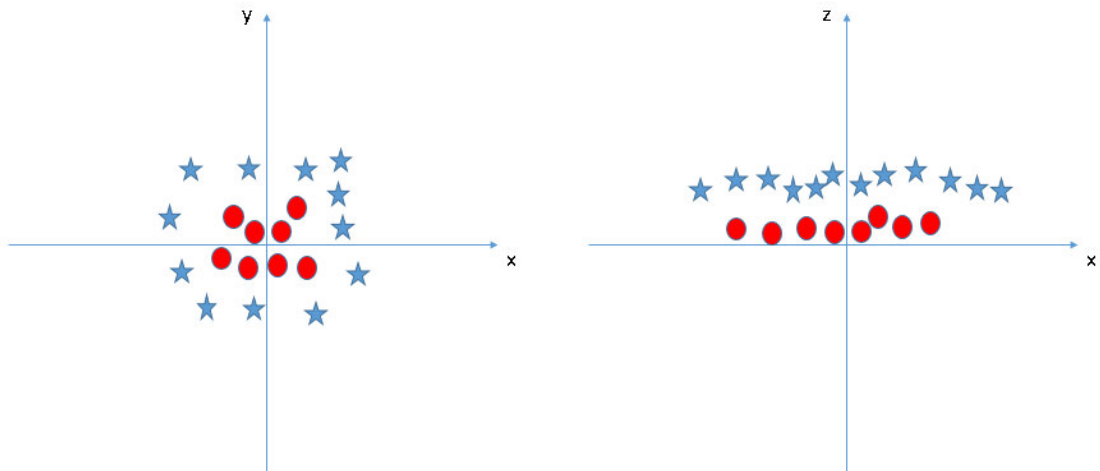
Včasih obeh množic linearno ni mogoče ločiti (Slika 10). SVM vpelje funkcijo izgube (ang. hinge loss function). Kadar točka leži na pravi strani ločnice je vrednost te funkcije 0, kadar pa točka leži na napačni strani je vrednost izgubne funkcije sorazmerna razdalji točke od ločnice. Imamo nov a rešljiv problem – skušamo minimizirati vrednosti izgubne funkcije. Na tak način lahko sprejmemo kompromis in dobimo dobro ločnico tudi kadar točke linearne niso popolnoma ločljive.



Slika 10: SVM primer, kjer linearna ločitev obeh množic ni mogoča, vendar pa z uporabo izgubne funkcije vseeno najdemo dobro ločnico.

Metoda podpornih vektorjev izkorišča še en prijem – zvijačo pozitivnih jeder (ang. Kernel trick). Ime izvira iz dokaza pozitivno končnih jeder (ang. Positive definite kernels) angleškega matematika J. Mercerja [23]. Beseda »jedro« v matematiki pojmuje utežno funkcijo za obteženo vsoto oz. integral. Zvijača jeder se s pomočjo skalarnega produkta vektorjev izogne izrecni preslikavi, ki je potrebna, da bi linearni učni algoritem naučili nelinearnega ločevanja.

V našem primeru gre za razširitev dimenzionalnosti, kadar imamo linearno neločljivi množici točk. Na sliki 11 imamo scenarij dveh množic, ki sta linearno popolnoma neločljivi. Dodamo novo dimenzijo z tako, da je $z = x^2 + y^2$. Ko izrišemo točke na osi x in z hitro vidimo, da je primere sedaj mogoče ločiti.

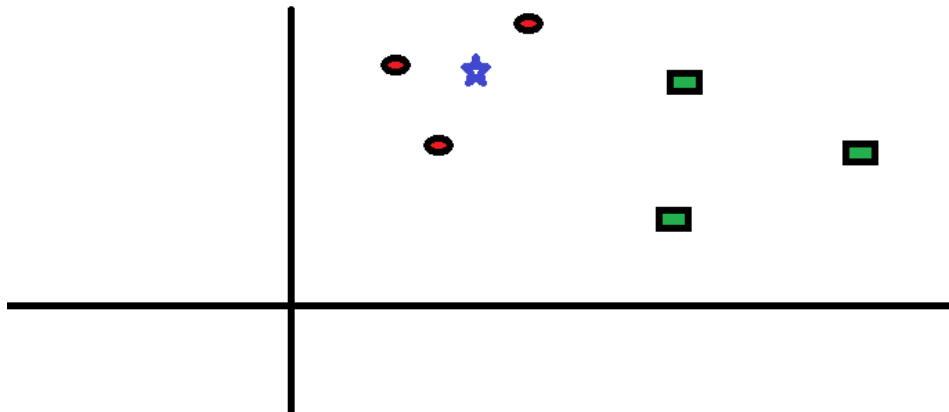


Slika 11: SVM ločevanje s pomočjo zvižaje pozitivnih jeder ($z = x^2 + y^2$).

2.3.5 Metoda najbližjih sosedov (*k*-NN)

Metoda najbližjih sosedov (ang. *k*-nearest neighbours – *k*-NN) v večdimenzionalnem prostoru poišče vhodnemu vektorju *k*-najbližjih sosedov in primerja njihove razrede [26].

Na sliki 12 vidimo učno množico rdečih krogcev in zelenih kvadratkov. Zvezdica predstavlja neznan primer, ki je bodisi rdeč krogec ali pa zelen kvadratik. Metoda poišče najbližjih nekaj sosedov, npr. 3. Njihove razrede lahko uteži sorazmerno z razdaljo tako, da so razredi bližjih bolj pomembni. Končni odgovor je povprečje *k*-najbližjih sosedov. V našem primeru metoda sklepa, da gre za rdeč krogec, saj so trije najbližji sosedje rdeči krogci.



Slika 12: Primer metode najbližjih sosedov

Metoda je priljubljena za klasificiranje neznanih primerov, saj je enostavna in hitra, postopek učenja pa ni potreben. Občutljiva je na pogostost pojavitev posameznega razreda. Torej v primeru, da bi imeli veliko zelenih kvadratkov in malo rdečih krogcev, bi imel poljuben neznan primerek veliko večjo verjetnost, da ga klasificiramo med zelene kvadratke.

Knjižnica OpenCV uporablja metodo najbližjega sosedu ($k=1$) pri vgrajenih algoritmih za prepoznavo, ki smo jih uporabili v lastni implementaciji. Več opišemo v poglavju 3.1.4.

Poglavje 3 Lastna implementacija

3.1.1 Uvod

Obstoječi spoznavni portal je napisan v Javi in želja, da bi zaznavanje in prepoznavanje lahko izvajali v Javi, je bila velika. Po raziskavi smo se odločili za uporabo knjižnice OpenCV, saj je preverjena, dobro podprta, implementira veliko uporabnih algoritmov, možno pa jo je klicati tudi iz Jave s pomočjo JNI (Java Native Interface). Prvi koraki so bili izredno zanimivi in navdušujoči. Ko pa smo želeli začeti s prepoznavo spola, smo ugotovili, da prevedena različica knjižnice, ki jo avtorji prilagajo na spletni strani in na repozitoriju, nima nekaterih funkcij namenjenih prepoznavanju oseb obrazov (Eigenfaces, Fisherfaces, ...). Te funkcije potrebujemo za klasifikacijo spola in starosti. Edina rešitev je bila, da knjižnico sami prevedemo. Uporabljali smo zadnjo stabilno različico: OpenCV 2.4.13. Več o prevajanju smo opisali v poglavju 3.1.2.

Algoritmov za zaznavo obraza je veliko. Nekateri so zapleteni v Javi pa bi jih zaradi pomanjkanja primerov morali implementirati sami. Uporabili smo kaskadni detektor Viola-Jones, nekoliko starejši, vendar še vedno popularen in hiter. Zanj smo se odločili, ker je za naše potrebe dovolj dober, poleg tega je že vgrajen v knjižnico OpenCV, avtorji knjižnice so priložili že naučen kaskadni klasifikator za zaznavo človeških obrazov in oči, kar nam dodatno olajša uporabo. V okviru diplomske naloge lastnega detektorja obrazov nismo učili.

V nadaljevanju se ukvarjamo s pripravo učne množice obrazov. Sam zaznan obraz ni dovolj. Potrebno ga je primerno izrezati, raztegniti in normalizirati rotacijo. Zadnje smo dosegli z zaznavo oči in na podlagi oči izračunali rotacijski kot. Bili smo mnenja, da je priprava učne množice najbolj pomemben korak za doseg dobrih rezultatov zaznave, zato smo učno množico slik veliko pregledovali tudi ročno. To pa ne pomeni, da zaznave nismo avtomatizirali. Učenje je le enkratni proces – kasnejše zaznavanje novih slik je lahko popolnoma avtomatizirano.

Sledi učenje že omenjenih algoritmov, ki so v osnovi sicer namenjeni prepoznavi oseb, mi pa smo jih uporabili za zaznavo najprej spola nato pa še starosti osebe.

Na koncu testiramo učinkovitost prepoznavne s testnimi slikami. Testne slike smo izvzeli iz množice učnih obrazov tako, da smo imeli za vsak spol in za skoraj vsako starost po eno naključno sliko. V starostnih skupinah od 55 let navzgor, smo imeli premalo kvalitetnih slik, zato je v tej množici nekoliko manj tudi testnih slik. Izvzete slike nismo uporabili pri učenju.

3.1.2 Prevajanje knjižnice OpenCV

Uporabiti želimo knjižnico OpenCV različico 2.4.13 [19]. Avtorji ponujajo že prevedeno knjižnico. Paket vsebuje datoteko ».dll« za operacijski sistem Windows, »so« za Linux, »dylib« za MacOS in datoteko »jar« za klice izvornih C++ funkcij iz Java.

Nekatere za nas zelo pomembne funkcije so avtorji knjižnice po pomoti pozabili preslikati v Java in edini način, da jih lahko uporabimo, je s popravkom izvorne kode in ponovnim prevajanjem za željene operacijske sisteme [20]. Na produkcijskih strežnikih spoznavnega portala je Linux, za razvoj uporabljamo Windows in MacOS, zato smo se odločili prevesti za vse tri operacijske sisteme.

V terminalu na Linux in MacOS sistemih zaženemo:

```
cmake -DBUILD_SHARED_LIBS=OFF ..  
make -j8
```

Na Windows sistemih je prevajanje najlažje izvesti tako, da naložimo Microsoft Visual Studio in program Cmake za sistem Windows. Cmake nam pripravi projektno datoteko, ki jo lahko enostavno odpremo in prevedemo z Visual Studiom.

Bolj podrobna navodila za prevajanje so na strani knjižnice [19].

3.1.3 Priprava učne množice slik

Uvod

Za učenje algoritmov smo vzeli profilne slike iz produkcije spoznavnega portala. Besedna zveza »profilna slika« se nanaša na predstavitevno sliko uporabnika, ne pa na sliko iz profila, kot bi jo lahko intuitivno narobe razumeli. Skupaj smo imeli približno 40.000 profilnih slik, od tega 15.000 ženskih in 25.000 moških. Velikosti slik so bile različne, vse pa so bile v formatu jpg.

Nekatere slike niso bile primerne zaradi slabe kvalitete, premajhne velikosti, brez frontalnih obrazov, ipd. V nadaljevanju je opisano, kako smo se slabih slik za namene učenja postopoma znebili.

Pridobitev in sortiranje vhodnih slik

Na portalu uporabniki vpišejo starost in spol. Pri prenosu slik s portala smo v ime datoteke zapisali spol in starost osebe. Tako smo razvrstili vse prenešene slike najprej v dve skupini – moške in ženske. Kasneje, ko smo slike nadaljno obdelali, pa smo jih razvrstili še po starostnih skupinah. Končna različica vsebuje 6 skupin: po 3 starostne skupine za vsak spol. Slike smo shranili v mape, po eno mapo za vsako skupino.

Priprava obraza

Programsko smo zaznali obraze vseh slik, izničili rotacijo ter prilagodili izrez obraza na enotno velikost in obliko. Temu pravimo priprava obraza, več o tem pa je opisano v poglavju 3.1.5. Ta korak smo ponovili tudi na testnih slikah in na vsaki nadaljni vhodni sliki, ki bi ji želeli določiti starost in spol. Na sliki 13 smo prikazali povprečne moške in ženske obraze treh starostnih skupin. Če bi povprečne slike izračunali pred postopkom priprave obraza, bi bile te popolnoma nejasne, saj različni deli obraza ne bi sovpadali. Slike, na katerih obraza nismo zaznali, smo izločili.



Slika 13: Povprečni obrazi posamezne skupine učne množice, zgoraj ženske, spodaj moški, starostne skupine: od 18 do 35 let, od 36 do 55 let, in več kot 55 let

Izločanje značilk

Na tej točki smo imeli pravilno obrnjene in večinoma lepe slike obrazov, razvrščene v 6 skupin, vendar pa smo v naši množici imeli pomešane tako frontalne kot tudi obraze iz profila. Ročno smo pregledali slike obrazov, ter tiste, ki niso bili popolnoma frontalni premaknili drugam. To so bili obrazi, na katerih nos ni usmerjen naravnost v kamero.

Pobrisali smo vse obraze, ki niso bili ustrezni. Kriteriji, po katerih smo se odločali za izbris, so slaba kvaliteta, nečloveški obraz (Slika 14), velika ovira čez obraz, sončna očala, ipd. Pri brisanju slik smo se trudili biti konsistentni, vendar pa je treba priznati, da je ta korak delno tudi subjektiven. Nekoliko manj smo bili tolerantni pri moških, saj smo imeli teh obrazov nekaj več kot ženskih, zato so bili moški obrazi izbrani nekoliko bolj konsistentno.



Slika 14: Zaznani obrazi, ki smo jih morali zaradi boljše rezultate odstraniti.

Nefrontalni obrazi

Z nefrontalnimi obrazi (t.i. obrazi iz profila) smo poizkušali različne prijeme, vendar pa večjega uspeha nismo dosegli. Skušali smo naučiti detektor spola na podlagi učne množice teh obrazov, vendar so bili rezultati zelo slabi, le nekoliko boljši od ugibanja (56%). Nekoliko boljše rezultate smo dosegli, ko smo vse nefrontalne obraze razvrstili v ločene množice, glede na kot nosu, vendar tega nismo znali avtomatizirati. Problem pri nefrontalnih obrazih je namreč to, da je lahko kot profila zelo različen. Frontalni obraz ima le eno rotacijo (pogled gor in dol), zato je 2d projekcija na sliki toliko bolj fiksna, profilni pa imajo dve različni rotaciji in tam kjer je na prvem obrazu npr. nos, je na drugem obrazu lice.

3.1.4 Učenje algoritmov in prepoznavna

Po obdelovanju učne množice slik je ostalo približno 1.200 ostrih in lepih frontalnih obrazov – 600 moških in 600 ženskih. Te smo uporabili za učenje algoritmov za prepoznavo spola in starosti.

Uporabili smo tri algoritme opisane v poglavju 2.3, torej Eigenfaces, Fisherfaces in LBP. Knjižnica OpenCV že izpostavi metode za delo s temi algoritmi.

Parametri učenja - Eigenfaces

Omenili smo da smo uporabili tri algoritme. Na sliki 15 smo prikazali izvleček programske kode, ki s pomočjo knjižnice OpenCV uporabi algoritem Eigenfaces. Algoritma Fisherfaces in LBP sledita enakemu vmesniku, torej so metode za delo enake, le pri inicializaciji različne implementacije sprejemajo različne parametre.

```
// Inicializacija Eigenfaces
1 FaceRecognizer recognizerEigenGender =
    Contrib.createEigenFaceRecognizer(0, Double.MAX_VALUE);

// Učenje algoritma Eigenfaces
2 recognizerEigenGender.train(imagesGender, labelsGenderMat);

// Klasifikacija novega obraza z algoritmom Eigenfaces
3 recognizerEigenGender.predict(image, outLabel, outConfidence);
```

Slika 15: Izvleček kode za inicializacijo, učenje in predikcijo.

Prva vrstica (takoj za prvim komentarjem) inicializira algoritem. Podamo parametra »num_components« in »threshold«. Prvi pove, koliko glavnih komponent želimo uporabiti. Preizkušali smo z različnimi števili glavnih komponent, vendar več kot zanemarljivih razlik v rezultatih ni bilo, zato smo uporabili vrednost 0, ki knjižnici pove, naj uporabi vse glavne komponente. Drugi parameter pove kolikšen je največji prag odstopanja z najbolj ujemajočo skupino, da sliko še klasificira. Kadar je drugi parameter določen s številko se pojavi scenarij, ko algoritem določene slike ne bo uspešno klasificiral, kadar je odstopanje dane slika z vsako od skupin večje od praga. Za naše potrebe smo želeli vedno dobiti najbolj ujemajočo skupino, tudi če je bilo odstopanje zelo veliko, zato smo uporabili največjo vrednost ki jo podatkovni tip Double lahko kodira.

Druga vrstica zažene proces učenja. V poglavju 2.3.1 smo opisali postopek prepoznavne z metodo lastnih obrazov. Knjižnica s podanim seznamom slik (imagesGender) in seznamom razredov (labelsGenderMat) zgradi t.i. lastne obraze z analizo glavnih komponent. Učne slike projicira v

PCA dimenzijski prostor. Ker smo nastavili parameter »num_components« na 0, algoritem uporabi vse glavne komponente.

Tretja vrstica sliko (image) klasificira v eno od skupin. V izhodni spremenljivki (outLabel in outConfidence) zapiše v katero skupino je algoritem sliko klasificiral in s kolikšno gotovostjo. Eigenfaces vhodno neznano sliko najprej projicira v zmanjšan dimenzijski prostor, nato pa poišče najbližjega soseda med lastnimi obrazi, ki se najboljše ujema. Najbližji sosed je tista slika, ki je vektorsko najbližje. OpenCV v spremenljivko outConfidence zapiše kar izračunano vektorsko razdaljo najbližjega soseda. Pri metodi Eigenfaces OpenCV izračuna Ekvlidovo razdaljo, torej koren vsote kvadratov razlik istoležečih komponent vektorja.

Parametri učenja – Fisherfaces

Implementacija algoritma fisherfaces v OpenCV po analizi glavnih komponent naredi še linearno diskriminantno analizo in dobljeni matriki zmnoži. Enako kot Eigenfaces, tudi Fisherfaces novo sliko projicira v enak podprostor kot vse slike učne množice, nato pa poišče najbližjega soseda. Več o algoritmu je opisano v poglavju 2.3.2. Enako kot pri prejšnjem smo uporabili vse glavne komponente in neomejen prag. Pri implementaciji Fisherfaces OpenCV prav tako računa Ekvlidove vektorske razdalje.

Parametri učenja – Lokalni binarni vzorci (LBP)

OpenCV v fazi učenja izračuna lokalne binarne vzorce za posamezne kose slik s podanimi parametri:

- Radij (ang. radius), $R = 1$,
- Število sosedov (ang. neighbors), $P = 8$,
- Mreža X (ang. grid_x) $grid_x = 8$,
- Mreža Y (ang. grid_y) $grid_y = 8$ in
- Prag (ang. threshold) $threshold =$ zelo velika vrednost (Double.MAX_VALUE)

Parametra radij in število sosedov se nanašata na izračun vrednosti vsake točke. Podrobneje sta opisana v poglavju 2.3.3. Mreža X in Y parametra se nanašata na razdelitev slike na manjše, na

katerih se izračunajo prostorski histogrami. Parameter prag že poznamo, gre za prag prepoznavne, ko se algoritem odloči, da vhodna neznana slika ni dovolj podobna nobeni iz učne množice, da bi jo lahko klasificiral. Želeli smo klasificirati vse slike, zato smo za prag uporabili zelo veliko številko. Za enačjem so zapisane vrednosti, ki smo jih uporabili za posamezen parameter.

V fazi prepoznavne za novo sliko enako izračuna lokalne binarne vzorce in prostorske histograme, nato pa primerja razlike histogramov slik iz učne množice, primerek z najbolj podobnimi prostorskimi histogrami pa je rezultat.

OpenCV histograme primerja po metodi Hi-kvadrat (ang. Chi-squared). Ta se izračuna kot kaže enačba 1, kjer je n število stolpcev v obeh primerjanih histogramih, x_i je vrednost i -tega stolpca v prvem histogramu, y_i pa vrednost i -tega stolpca v drugem histogramu.

$$\sum_{i=1}^n \frac{(x_i - y_i)^2}{(x_i + y_i)}$$

Enačba 1: Izračun razdalje po metodi Hi-kvadrat

3.1.5 Priprava obraza

Postopek priprave obraza smo izvedli na vsaki sliki učne množice, ter na vsaki neznani sliki, ki jo želimo klasificirati.

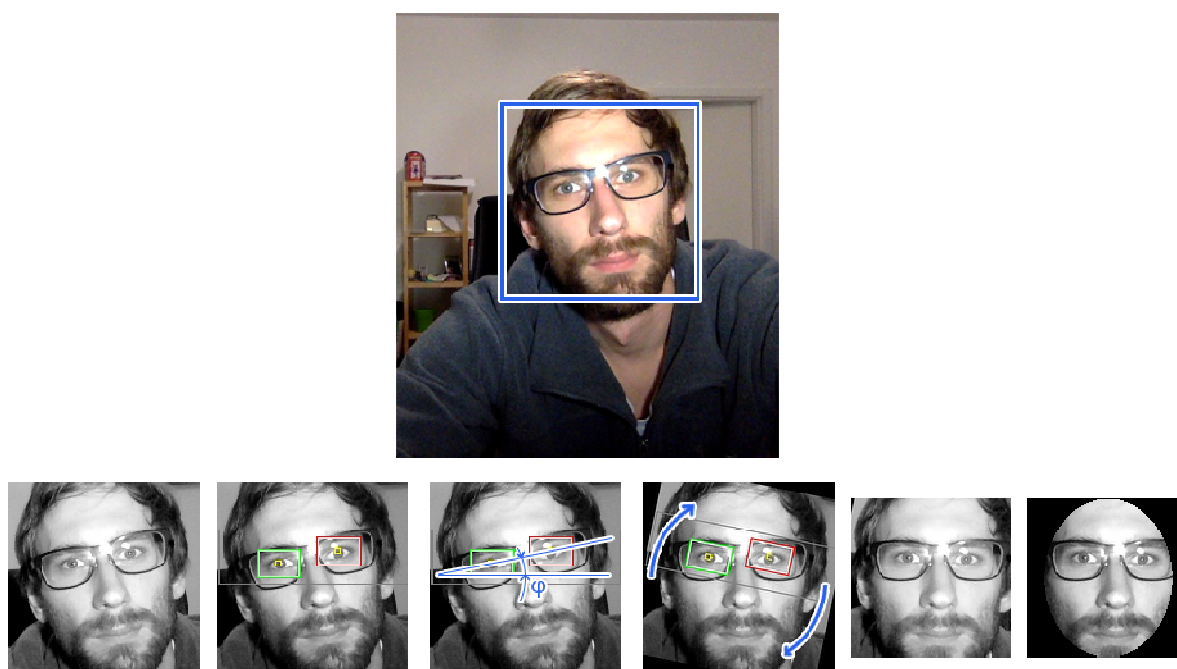
Zaznavanje obraza

Na vhodnih slikah smo skušali poiskati obraz s pomočjo OpenCV metode `detectMultiScale`, ki z algoritmom haar značilk (Viola-Jones) poišče obraze. Uporabili smo knjižnici priloženo datoteko »haarcascade_frontalface_alt.xml«, ki vsebuje naučene parametre za kaskadni klasifikator za zaznavanje človeških obrazov. Slik smo imeli veliko, zato smo si lahko privoščili zavreči vse

slike, na katerih nismo zaznali natančno enega obraza. Več o metodi detectMultiScale je opisano v [25]. Klic metode smo izvedli s privzetimi parametri.

Prilagajanje rotacije

Uporabljeni algoritmi za prepoznavanje spola in starosti so zelo občutljivi na vse tri prostorske rotacije obraza in tudi na ozadje. Iskali smo prijeme, da bi ozadje in rotacijo kar se da izničili. Rotacijo po z osi nam je uspelo izničiti programsko (Slika 16), rotacijo okrog y osi smo se znebili tako, da smo iz učne množice enostavno izbrisali vse slike, ki so jo vsebovale. Ročno smo izločili nefrontalne obraze – torej tiste, ki niso usmerjeni naravnost v kamero. Rotacijo okrog x osi smo dovolili, saj je bila majhna. Testni rezultati na nefrontalnih slikah so bili občutno slabši, zato smo se odločili nadaljevati samo s frontalnimi. Algoritmi delujejo zgolj na podlagi 2D slike. Različnih kombinacij rotacij nefrontalnih obrazov je preveč in posledično algoritmi postanejo precej manj natančni. Za namene te raziskave smo se zato odločili vse nefrontalne obraze izločiti.



Slika 16: Primer priprave obraza: zgoraj zaznan obraz, spodaj koraki obdelave.

Po zaznavi frontalnega obraza smo dobljeni kvadrat in s tem tudi izrez za 20% razširili, saj bomo v nadaljevanju sliko rotirali in nato obraz obrezali na enotno velikost 128x128 točk. S pomočjo knjižnice OpenCV smo zaznali vse oči, enako kot smo to storili za zaznavo obraza, le da smo uporabili drugi datoteki kaskadnih parametrov, tokrat:

- »haarcascade_mcs_lefteye.xml« in
- »haarcascade_mcs_righteye.xml«

Obe datoteki sta priloženi knjižnici OpenCV.

Očesa smo iskali v omejenem zgornjem območju slike obraza, saj smo se tako izognili napačnih zaznav oči. Okno iskanja oči smo poiskali s poskušanjem. Končno okno zaznave oči je zavzemalo pravokotnik, od zgornjega roba odmaknjen za četrtno celotne višine slike in visok $7/24$ višine celotne slike. Širine nismo omejevali, torej je bilo okno razširjeno po celi širini slike.

Zelo pogosto smo zaznali dve levi ali dve desni očesi ali pa celo dve levi in dve desni. Problem smo reševali tako, da smo vsa prekrivajoča se očesa združili, nato pa z enostavnim geometričnim algoritmom ugotavljali ali gre za levo ali za desno oko (Slika 16). Slike, na katerih na koncu nismo zaznali točno enega para oči – torej eno levo in eno desno – smo izločili. Natančno eno levo in eno desno oko smo zaznali na več kot 70% slikah, kar je bilo nad pričakovanji.

Sledil je izračun kota φ med premico, ki gre skozi središča obeh oči in abscisno osjo, s pomočjo matematične funkcije arcus tangens (Slika 16). Sliko smo nato zavrteli za izračunani kot tako, da sta bili očesi poravnani vodoravno. Obraz smo nato za 20% obrezali in skrčili na 128x128 točk.

Na koncu smo skušali zmanjšati vpliv ozadja. Na slikah obrazov smo pobarvali črno vse točke na sliki, ki ležijo izven obrazne elipse, kot kaže zadnji korak na sliki 16. Primerno elipso smo poiskali ročno s človeškim primerjanjem slik v učni množici, tako da smo obdržali celoten obraz in hkrati izbrisali čimveč ozadja.

3.1.6 Integracija z obstoječim sistemom

Z algoritmom bi radi olajšali delo administratorja. Ne moremo pričakovati, da ga bomo popolnoma nadomestili, saj je problem zaznavanje obraza in klasifikacija spola za poljubno sliko

zelo zahteven. Poleg tega je odločitev o ustreznosti fotografije nekoliko subjektivna. Zadovoljiti smo se pripravljani z dvotretjinsko uspešnostjo algoritma.

Življenjski cikel slik na portalu

- Uporabnik sliko naloži na spletni strani.
- Na strežniku poiščemo obraz in sliko obdelamo, kot smo opisali v poglavju 3.1.5.
- Poiščemo primeren izrez obraza, tako da razširimo kvadrat zaznanega obraza.
- Zaznamo spol in starost in jih primerjamo z vnešenimi podatki.
- Predlagamo programsko odločitev o ustreznosti slike.
- Administrator sliko potrdi, zavrne ali pa pred potrditvijo ročno označi nov izrez obraza.

Kadar je na sliki zaznan natanko en obraz in spol ustreza spolu uporabnika lahko tako sliko potrdimo. Kljub temu pa smo se odločili končno potrjevanje še vedno prepustiti človeku. Programsko le predlagamo ali je slika primerna ali ne. Veliko časa administratorju prihranimo s predlogom izreza obraza.

Poglavje 4 Sklepne ugotovitve

4.1.1 Predstavitev rezultatov

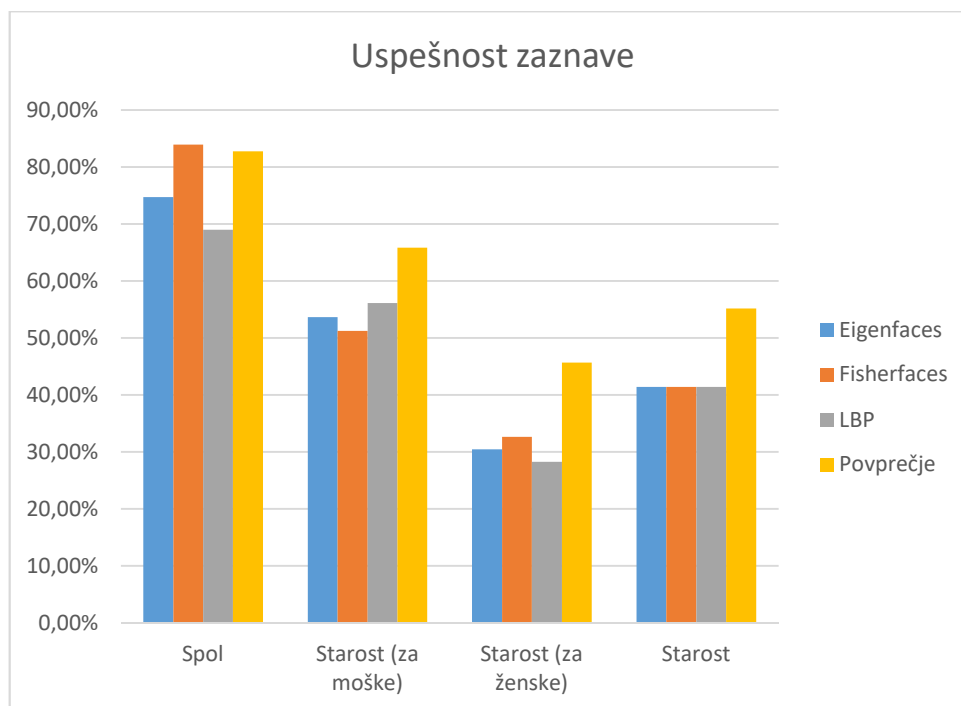
Testna množica je vsebovala 87 slik obrazov, 46 ženskih in 41 moških od tega pa po starosti:

- Starostna skupina 1 (od 18 do 35 let), ženskih: 28, moških: 23
- Starostna skupina 2 (od 36 do 55 let), ženskih: 15, moških: 15
- Starostna skupina 3 (več kot 55 let), ženskih: 3, moških: 3

Algoritmi so bili različno uspešni. Uporabili smo 3 algoritme, Eigenfaces (PCA), Fisherfaces (LDA) in LBP, vsi so predstavljeni v poglavju 2.3. Kot končni rezultat smo vzeli povprečje vseh treh algoritmov. Če sta torej za nek obraz dva algoritma izračunala ženski spol, en pa moškega, smo vzeli kot končno odločitev ženski spol.

	Eigenfaces	Fisherfaces	LBP	Povprečje
Uspešnost predikcije spola	74,71%	83,91%	68,97%	82,76%
Uspešnost predikcije starosti za moške	53,66%	51,22%	56,10%	65,86%
Uspešnost predikcije starosti za ženske	30,43%	32,61%	28,26%	45,65%
Uspešnost predikcije starosti skupaj	41,38%	41,38%	41,38%	55,17%

Tabela 1: Predstavitev relativne uspešnosti zaznavnih algoritmov.



Slika 17: Graf uspešnosti zaznave

Za popoln uspeh testnega primera upoštevamo tiste, kjer smo pravilno prepoznali tako spol kot tudi starostno skupino. Takih testnih primerov je bilo 48/87 ali 55,17%.

Komentar rezultatov

Z več kot 82 odstotno uspešnostjo prepoznavne spola smo bili zelo zadovoljni. Prepričani smo, da bi lahko izboljšali še marsikaj. Pri nekaterih virih smo naleteli na več kot 98 odstotno uspešnost s pomočjo Fisherfaces (LDA), kar se nam zdi osupljivo dobra uspešnost. Vendar pa menimo, da je uspešnost zaznave odvisna od konsistence učnih in testnih slik. Več vpliva kot imamo na ozadje, šum, kot pogleda, ipd., lažje izvedemo prepoznavo. V našem primeru pri vhodnih slikah nismo imeli kontrole nad zunanjimi dejavniki – uporabili smo obstoječe produkcijske slike uporabnikov spoznavnega portala, z njimi pa smo skušali narediti čim več.

Uspešnost prepoznavne starostnih skupin ni najboljša. Še slabša je bila, ko smo uporabili 5 starostnih skupin namesto samo 3. Prepoznavanje starosti je za marsikateri primer težka naloga

tudi za človeka. Zanimiva je opazna razlika v uspešnosti prepoznavanja starosti med ženskami in moškimi. Glede na vpogled v vhodne slike, smo mnenja, da je ta razlika nastala zaradi ličil, ki jih nosijo ženske. Meja med prvima dvema skupinama je 35 let. Ženski obrazi med 25 in 45 so lahko zelo podobni. Gube, hrapavost in še nekatere značilke starosti ženske na svojih profilnih slikah uspešno prikrijejo.

4.1.2 Možnosti za izboljšave

Če bi se lotili celotnega projekta z novo pridobljenim znanjem ponovno, bi za prepoznavanje uporabili konvolucijske nevronske mreže, morda v povezavi z lokalnimi binarnimi vzorci. Koncept nevronskih mrež je izredno zanimiv in zmogljiv. Časa za raziskovanje je vedno omejeno in glede na naše potrebe smo v splošnem zelo zadovoljni z izvedbo in rezultati. Zanimivo je bilo operirati s tolikšnim številom slik, ter opazovati delovanje različnih algoritmov.

Klasifikacija z nevronskimi mrežami

Konvolucijske nevronske mreže (ang. Convolutional neural networks – CNN) so vse bolj popularna in vroča tema na področju umetne inteligence in računalniškega vida. Ideja izvira iz človeških možganov, kjer se živčne celice (nevroni) povezujejo v kompleksne mreže. Posamezen nevron ima enostavno nalogo: glede na intenziteto signalov na vhodu in na določene uteži, oddaja določeno intenziteto na izhodu. Skupaj, kot mreža nevronov, kjer se izhodi enih povezujejo na vhode drugih nevronov so sposobni reševati kompleksne naloge. Izkaže se, da je tak pristop v mnogih primerih manj omejen in bolj učinkovit od konzervativnih metod [18].

Nevronske mreže lahko naučimo razlikovati značilke predmetov na sliki. Z njimi bi lahko prepoznavali spol in starost. Tema je nedvomno zelo zanimiva, rezultati pa so izvrstni pri reševanju različnih problemov računalniškega vida. A Karpathy je v svojem blogu predstavil rangiranje kvalitete selfijev z uporabo nevronskih mrež [24]. Mrežo je učil s slikami iz interneta. Učne slike je ocenil glede na število prejetih »všečkov« in komentarjev. Podoben princip bi lahko uporabili tudi mi, vendar je pomanjkljivost takšnega sistema v tem, da na koncu ne vemo točno zakaj je sistem posamezno sliko odobril. Razlaga odločitve je pri takšnem primeru težja.

Zaključek

Želeli smo avtomatizirati potrjevanje profilnih slik uporabnikov spletnega portala s pomočjo algoritmov računalniškega vida. Naloga ni bila lahka in potrebno je bilo pridobiti veliko novega znanja. Preučili smo delovanje algoritmov, uporabili pa knjižnico OpenCV, ki že implementira algoritme, ki smo jih potrebovali za izvedbo.

Želeli smo prepoznati starost, da bi ločili med polnoletnimi in mladoletnimi osebami in tako slike z mladoletnimi osebami avtomatsko zavrnili. Zaradi pomanjkanja slik mladoletnih obrazov smo to idejo opustili in v akademske namene raziskali kako prepoznati starost treh starstnih skupin in spol, kjer sta razreda samo dva.

Na voljo smo imeli podatke živega spletnega portala, kjer smo pridobili veliko količino produkcijskih slik uporabnikov. Slik smo imeli približno 40.000, od tega 25.000 moških in 15.000 ženskih. Imeli smo tudi osnovne podatke o osebi za vsako sliko (spol in starost). Slik uporabnikov zaradi varovanja osebnih podatkov nismo objavili, izredno pa smo bili veseli, da smo imeli priložnost pripraviti lastno učno množico slik.

Z zaznavo obrazov smo bili zadovoljni. V naprej naučen Viola-Jones za zaznavo človeških obrazov, ki so ga učili avtorji knjižnice, se je izkazal za dovolj dobrega. Vseh obrazov sicer ni zaznal, včasih je zaznal kaj, kar ni bilo obraz, npr. stensko električno vtičnico. Po razmisleku to sicer ni presenetljivo, saj je okrogla, ima dve temnejši luknji za oči ter pritrditveni vijak za nos. Kljub temu smo z njim uspešno zaznali veliko večino obrazov in oči brez večjega napora.

Algoritmi Eigenfaces, Fisherfaces in LBP so bili kljub nasprotnim pričakovanjem podobno učinkoviti. Pričakovali smo da se bo Eigenfaces odrezal precej slabše kot se v resnici je. Imeli smo veliko primerov, kjer se je Eigenfaces odločil pravilno, ostala algoritma pa ne. Vsesplošno se je najbolje odrezal Fisherfaces, tudi rezultati LBP pa niso bili veliko slabši. Odločitev, da uporabimo tri algoritme in upoštevamo povprečno odločitev vseh (odločitev dveh od treh), se je izkazala za dobro. Na sliki 17 graf lepo prikazuje, kako in koliko je povprečna odločitev boljša od vsakega od treh naštetih algoritmov posamično.

Zaznava spola z več kot 80% natančnostjo nas je izredno pozitivno presenetila. Obstajajo boljše rešitve, ki prepoznavajo spol z več kot 99% natančnostjo, vendar pa smo kot novinci na področju računalniškega vida bili zadovoljni z uspehom.

Nasprotno je bila prepoznavna spola z nekaj več kot 55% natančnostjo za 3 starostne razrede manjše razočaranje. Prepričani smo, da bi lahko rezultat še nekoliko izboljšali, vendar pa smo morali zaradi pomanjkanja časa nadaljevati z delom. Kljub relativno slabim rezultatom smo veseli izkušnje. Kot smo omenili v zadnjem poglavju: »Če bi se ponovno lotili prepoznavati starost in spol, bi podrobneje raziskali in uporabili nevronske mreže«. V tej nalogi smo uporabili algoritme, ki bi jih na področju računalniškega vida morda že lahko označili za nekoliko klasične, ne pa več toliko napredne, saj so nekateri stari že desetletja.

Na produkciji spletnega spoznavnega portala imamo implementacijo zaznave obraza in avtomatski izrez. To je prej moral ročno določati administrator, zdaj pa mu algoritmi precej olajšajo delo. Odločitev o primernosti obraza ostaja v človeških rokah, saj programsko inteligenco nismo uspeli dovolj približati za produkcijsko rabo. Implementirali pa smo programsko sugestijo, kadar je na sliki zaznan natanko en obraz, sliko administratorju programsko predlagamo kot primerno, na koncu pa to odločitev sprejme ali popravi človek.

Literatura

- [1] Spoznavni portal Ona-on. (januar 2017) [Online] <https://www.ona-on.com>
- [2] P. Viola in M. J. Jones, "Robust Real-Time Face Detection", v International Journal of Computer Vision 57, 137–154, 2004, dostopno: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.93.8268&rep=rep1&type=pdf>
- [3] Y. Freund in R. E. Schapire, "A Short Introduction to Boosting" (AdaBoost), 1995, v Journal of Japanese Society for Artificial Intelligence, 14(5): 771-780, september, 1999. (Prevod iz japonščine: Naoki Abe).
- [4] S. Liao, A. K. Jain in S. Z. Li, iz IEEE, "A Fast and Accurate Unconstrained Face Detector", dostopno: <https://arxiv.org/pdf/1408.1656v3.pdf>
- [5] E. H. Weber, "Weber's Law of Just noticeable difference", povzetek iz USD Internet Sensation & Perception Laboratory, dostopno: <http://apps.usd.edu/coglab/WebersLaw.html>
- [6] P. Sinha, "Qualitative Representations for Recognition", Massachusetts Institute of Technology, dostopno: http://web.mit.edu/sinhalab/Papers/qualitative_reps.pdf
- [7] L. Sirovich, M. Kirby, "Low-dimensional procedure for the characterization of human faces", Journal of the Optical Society of America A, Vol. 4, stran 519, marec 1987, dostopno: <http://www.face-rec.org/interesting-papers/General/ld.pdf>
- [8] Internetni vir, januar 2017, dostopno: <https://en.wikipedia.org/wiki/Eigenface>
- [9] M. Welling, "Fisher Linear Discriminant Analysis", oddelek za računalništvo, univerza v Torontu, dostopno: <https://www.ics.uci.edu/~welling/teaching/273ASpring09/Fisher-LDA.pdf>
- [10] Internetni vir, januar 2017, dostopno: https://en.wikipedia.org/wiki/Linear_discriminant_analysis
- [11] S. Raschka, "Linear Discriminant Analysis – Bit by Bit", avgust 2014, dostopno: http://sebastianraschka.com/Articles/2014_python_lda.html
- [12] D. Huang, C. Shan, M. Ardebilian, Y. Wang in L. Chen, "Local Binary Patterns and Its Application to Facial Image Analysis: A Survey", dostopno: <http://liris.cnrs.fr/Documents/Liris-5004.pdf>
- [13] M. Pietikäinen, J. Heikkilä, "Image and Video Description with Local Binary Pattern Variants", Machine Vision Group, Univerza v Oulu, Finska, junij 2011, dostopno: <http://www.ee.oulu.fi/research/imag/mvg/files/pdf/CVPR-tutorial-final.pdf>

- [14] Internetni vir, januar 2017, dostopno: http://www.scholarpedia.org/article/Local_Binary_Patterns
- [15] Internetni vir, januar 2017, dostopno: https://en.wikipedia.org/wiki/Local_binary_patterns
- [16] T. Jiří in M. Jiří, "Extended Set of Local Binary Patterns for Rapid Object Detection", Center za strojno zaznavanje, Fakulteta za elektrotehniko, Tehnična univerza v Pragi, Češka, dostopno: <ftp://147.32.84.2/pub/cv1/articles/matas/trefny-lbp-cvww10.pdf>
- [17] Internetni vir, januar 2017, dostopno: https://en.wikipedia.org/wiki/Support_vector_machine
- [18] S. Lawrence, C. L. Giles, A. C. Tsoi in A. D. Back, "Face Recognition: A Convolutional Neural Network Approach", v IEEE Transactions on Neural Networks, Special Issue on Neural Networks and Pattern Recognition, Volume 8, Number 1, strani od 98 do 113, 1997, dostopno: <http://handysolution.com/science/CNN.pdf>
- [19] Internetni vir, "Open source computer vision library", januar 2017, dostopno: <http://opencv.org/>
- [20] Internetni vir, januar 2017, dostopno: <https://github.com/opencv/opencv/issues/4538>
- [21] Internetni vir, januar 2017, dostopno: https://en.wikipedia.org/wiki/Ronald_Fisher
- [22] R. Sunil, »Understanding Support Vector Machine algorithm from examples«, oktober 2015, dostopno: <https://www.analyticsvidhya.com/blog/2015/10/understaing-support-vector-machine-example-code>
- [23] Internetni vir, januar 2017, dostopno: https://en.wikipedia.org/wiki/Positive-definite_kernel
- [24] A. Karpathy, »What a Deep Neural Network thinks about your #selfie«, oktober 2015, dostopno: <http://karpathy.github.io/2015/10/25/selfie/>
- [25] Internetni vir, januar 2017, dostopno: http://docs.opencv.org/2.4/modules/objdetect/doc/cascade_classification.html#cascade_classifier-detectmultiscale
- [26] Internetni vir, januar 2017, dostopno: <https://www.analyticsvidhya.com/blog/2014/10/introduction-k-neighbours-algorithm-clustering/>