

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Domen Lušina

**Prepoznavna obrazov z opisnikom
Alfa-Gama**

DIPLOMSKO DELO

UNIVERZITETNI ŠTUDIJSKI PROGRAM
PRVE STOPNJE
RAČUNALNIŠTVO IN INFORMATIKA

Ljubljana, 2017

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Domen Lušina

**Prepoznavna obrazov z opisnikom
Alfa-Gama**

DIPLOMSKO DELO

UNIVERZITETNI ŠTUDIJSKI PROGRAM
PRVE STOPNJE
RAČUNALNIŠTVO IN INFORMATIKA

MENTOR: izr. prof. dr. Matej Kristan

SOMENTOR: izr. prof. dr. Jasna Maver

Ljubljana, 2017

Rezultati diplomskega dela so intelektualna lastnina avtorja in Fakultete za računalništvo in informatiko Univerze v Ljubljani. Za objavo in koriščenje rezultatov diplomskega dela je potrebno pisno privoljenje avtorja, Fakultete za računalništvo in informatiko ter mentorja.

Besedilo je oblikovano z urejevalnikom besedil L^AT_EX.

Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Tematika naloge:

Prepoznavna obrazov na slikah zahteva učinkovito izločanje značilnic, hitro izgradnjo opisnikov in hitro primerjavo le teh z opisniki v obsežnih slikovnih bazah. V nalogi raziščite možnost uporabe opisnika Alfa-Gama, ki uporablja model neodvisnih učinkov, za nalogo prepoznavanja obrazov. Opišite popularne opisnike, ki so v rabi za razpoznavanje obrazov in jih kvantitativno primerjajte z opisnikom Alfa-Gama.

Zahvaljujem se moji družini za podporo ob izdelavi diplomskega dela in prijateljem za spodbudne besede. Zahvalil bi se tudi mentorju, prof. Mateju Kristanu, in še posebej somentorici, prof. Jasni Maver, za vodenje in pomoč ob izdelavi diplomskega dela.

Kazalo

Povzetek

Abstract

1	Uvod	1
1.1	Motivacija	1
1.2	Cilj diplomskega dela	2
1.3	Pregled področja	2
1.4	Struktura dela	3
2	Programska oprema in tehnologije	5
2.1	MATLAB	5
2.2	Knjižnica OpenCV	6
2.3	Paket mexopencv	6
2.4	Knjižnica dLib	6
3	Predprocesiranje slik	9
3.1	Zaznava obraza	9
3.2	Zaznava verige značilnk na obrazu	13
3.3	Poravnava obraza	14
3.4	Obrezovanje obraza	16
4	Kategorizacija obrazov	17
4.1	Metoda vreče besed (BOW)	17

5	Alfa-Gama opisnik	23
5.1	Predstavitev lokalne regije z DEM-om	23
5.2	Odvečnost vektorja učinkov β	25
5.3	Robustni deskriptor AG	26
6	Podatkovne baze	29
6.1	Zbirka slik obrazov Univerze v Essexu	29
6.2	Podatkovna baza UFI	30
7	Rezultati	33
7.1	Izbira tipa obrezave obraza	33
7.2	Rezultati na vseh bazah	35
7.3	Rezultati ob prisotnosti šuma ali zameglitve	36
7.4	Interpretacija rezultatov	37
8	Sklepne ugotovitve	39
	Literatura	41

Seznam uporabljenih kratic

kratica	angleško	slovensko
1D	one-dimensional	enodimenzijski
3D	three-dimensional	tridimenzijski
AG	Alpha-Gamma	Alfa-Gama
API	application programming interface	vmesnik za namensko programiranje
BOW	bag of words	vreča besed
BRIEF	binary robust independent elementary features	binarne robustne neodvisne elemente
BRISK	binary robust invariant scalable keypoints	binarne robustne invariantne skalabilne značilke
DEM	dependent effects model	model odvisnih učinkov
HOG	histogram of oriented gradients	histogram usmerjenih gradientov
FAST	features from accelerated segment test	značilke iz pospešenega segmentacijskega testa
FREAK	fast retina keypoint	hitre retina značilke
ICA	independent component analysis	analiza neodvisnih komponent
LATCH	the learned arrangements of three patch codes	naučena koda postavitve treh zaplat
LDA	linear discriminant analysis	linearna diskriminantna analiza
ORB	oriented FAST and rotated BRIEF	usmerjeni FAST in rotirani BRIEF
PCA	principal component analysis	metoda glavnih komponent
SIFT	scale invariate feature transformation	skalirno invariantna transformacija
SURF	speeded up robust features	pohitrene robustne značilke
SVM	support vector machine	metoda podpornih vektorjev

Povzetek

Naslov: Prepoznavna obrazov z opisnikom Alfa-Gama

Avtor: Domen Lušina

Povzetek: Za prepoznavo obrazov obstajajo številne metode, nekatere med njimi za prepoznavo uporabljajo lokalne opisnike. V tej diplomskem delu si bomo za namen prepoznave obrazov ogledali delovanje popolnoma novega lokalnega opisnika, imenovanega Alfa-Gama (krajše AG), ki je bil razvit v Laboratoriju za vizualne in spoznavne sisteme na Fakulteti za računalništvo in informatiko v Ljubljani. Pripravili smo sistem za prepoznavo obrazov na podlagi algoritma vreče besed in klasifikatorja SVM. Predlagan sistem smo razvili za številne popularne opisnike in opisnik AG. Sistem smo testirali na javno dostopnih podatkovnih bazah. V večini primerov je opisnik AG izboljšal rezultate prepoznavne in se je izkazal za dobro alternativo k ostalim testiranim opisnikom.

Ključne besede: Alfa-Gama opisnik, računalniški vid, prepoznavna obraza, detekcija obraza, lokalni opisniki.

Abstract

Title: Face recognition using Alpha-Gamma descriptor

Author: Domen Lušina

Abstract: For facial recognition there are numerous methods available, some of them use local descriptors. In this thesis, for the purpose of face recognition, we will take a closer look at novel local descriptor Alpha-Gamma (shorter AG) developed at the Visual Cognitive Systems Laboratory at the Faculty Of Computer and Information Science, University of Ljubljana. We developed a system for face recognition using bag of words algorithm and SVM classifier. The proposed system was developed for several popular local descriptors and AG. We used publicly available databases for testing. In most cases AG improved facial recognition rate and was shown as a good alternative to other tested local descriptors.

Keywords: Alpha-Gamma descriptor, computer vision, facial recognition, face detection, local descriptors.

Poglavje 1

Uvod

1.1 Motivacija

Prepoznavanje obrazov je ena izmed osnovnih človeških zmožnosti, ki smo jo v večini osvojili v zgodnjem obdobju življenja. Težko si je predstavljati vsakdanje življenje brez zmožnosti, da bi na ulici prepoznali znano osebo, prijatelja ali sorodnika. Čeprav se nam prepoznavanje obrazov zdi samoumevna funkcija, je to zelo kompleksna naloga, ki smo jo sposobni upravljati zelo hitro in učinkovito. Kompleksnost je posebej opazna, ko želimo zmožnost prepoznavanja obrazov prenesti na računalniški sistem.

Z nalogo prepoznavanja obrazov se v računalniškem svetu ukvarja računalniški vid – interdisciplinarno področje, ki se ukvarja z uporabo računalnikov za razumevanje digitalnih slik oziroma videov. Z inženirskega vidika si računalniški vid želi naloge, ki jih je človeški vizualni sistem zmožen izvesti, prenesti na računalniške sisteme, jih avtomatizirati in tudi izboljšati [3, 15, 33].

Prvi začetki razvoja sistemov za prepoznavanje obrazov segajo v šestdeseta, ko se je začel razvoj prvih polavtomatskih sistemov, ki so zahtevali ročno vnašanje značilnic na obrazu (oči, ušesa, nos in usta) [25]. Skozi leta je prišlo do veliko izboljšav [13, 32, 34, 29], vendar je prepoznavanje obrazov še vedno zelo aktualna tema raziskovanja.

Eden izmed razlogov, zakaj je prepoznavanje obrazov zanimivo, je velik

potencial uporabe v vladne in komercialne namene na različnih področjih. Med drugim se uporablja za kontrolo dostopa, varnost, nadzor in zdravstvene namene. V zadnjih letih se je pojavila tudi bolj komercialna uporaba prepoznavne obrazov. Pogosto so to socialna omrežja, kot sta Facebook [11] in Google+ [14], ki so v večini dostopna na pametnih telefonih.

1.2 Cilj diplomskega dela

Za prepoznavanje obrazov se uporabljajo različne metode. Eden izmed učinkovitih pristopov za prepoznavo obrazov je uporaba lokalnih opisnikov. Med prvimi sta se za prepoznavo obrazov uporabljala opisnika SIFT [23] in SURF [5].

Skozi leta pa je sledil razvoj novih lokalnih opisnikov. Med drugim je bil v Laboratoriju za vizualne in spoznavne sisteme na Fakulteti za računalništvo in informatiko v Ljubljani razvit opisnik AG. Gre za opisnik, ki uporablja model odvisnih učinkov.

Cilj diplomskega dela je preizkusiti lokalni opisnik AG za nalogo prepoznavne obrazov. Pripravili bomo sistem za prepoznavo obrazov, sestavljen iz modula za detekcijo obrazov in modula za prepoznavo obrazov. Prepoznavna obrazov bo realizirana z različnimi opisniki, kar nam bo omogočilo primerjavo rezultatov prepoznavne in vrednotenje primernosti različnih opisnikov za to nalogo.

1.3 Pregled področja

Prepoznavna obrazov je zelo obsežno področje, ki ga lahko grobo razdelimo na dva dela, ki uporabljata lokalni oz. globalni pristop. Dolgo časa so prevladovali globalni pristopi. Globalni pristop so med drugim metoda glavnih komponent (PCA) [17], linearna diskriminantna analiza (LDA) [10] in analiza neodvisnih komponent (ICA) [4]. Uporaba teh metod ima svoje slabosti, med drugim je znano, da je metoda PCA občutljiva na spremembe v osvetlitvi. Na splošno so te metode, ker opisujejo obraz kot celoto, zelo občutljive

na spremembe videza obraza, kot je npr. sprememba v izrazih. Na drugi strani imamo lokalni pristop, kjer obraz opišemo s specifično karakteristiko na določenih točkah, imenovanih značilne točke (*angl. landmarks*). Okoli teh značilnih točk lahko informacijo o sosednjih slikovnih elementih zapišemo kot lokalni opisnik. Lokalni pristopi so bolj robustni na spremembe v izrazih obraz. Z uporabo ustreznega opisnika lahko minimiziramo vpliv osvetlitve. Med prvimi je bil opisnik SIFT [23], skozi leta pa so bili razviti številni novi opisniki [19, 6, 1, 2, 20, 31]. Prepoznava obrazov pogosto vključuje uporabo metod s področja umetne inteligence. Med drugim so se za primerne izkazale metode: klasifikator SVM, naključni gozdovi, umetne nevronske mreže itd. Prej omenjene lokalne opisnike lahko uporabimo v kombinaciji z metodami za klasifikacijo. Uporaba metod umetne inteligence je postala skoraj neizogibna, saj so doprinosi zelo veliki.

1.4 Struktura dela

Najprej si bomo ogledali programsko opremo in tehnologije, uporabljene v diplomskem delu. Sledi poglavje Predprocesiranje slik, kjer opišemo postopek detekcije in izrezave obraza iz slike. V poglavju Kategorizacije obrazov si bomo pogledali delovanje metode vreče besed (BOW), ki smo jo uporabili za prepoznavo obrazov in naša pristopa pri pridobivanju značilnk. Sledi poglavje, ki opisuje delovanje Alfa-Gama opisnika. V poglavju Podatkovne baze si bomo pogledali uporabljene podatkovne baze obrazov iz Univerze v Essexu in nenadzorovane podatkovne baze obrazov Češke novinarske agencije (ČTK). Sledi prikaz rezultatov in njihova interpretacija.

Poglavje 2

Programska oprema in tehnologije

2.1 MATLAB

MATLAB¹ je programsko okolje in programerski jezik četrte generacije, ustvarjen s strani podjetja MathWorks. MATLAB omogoča računanje z matrikami, prikazovanje funkcij in podatkov, implementiranje algoritmov, ustvarjanje uporabniških vmesnikov. Podprto je tudi povezovanje z ostalimi programerskimi jeziki, kot so C, C++, C#, Java, Fortran in Python [36].

Za MATLAB sem se odločil zaradi orodji, ki jih ponuja na področju računalniškega vida (*Computer Vision System Toolbox*) in umetne inteligence (*Statistics and Machine Learning Toolbox*). Ta orodja so se izkazala za zelo uporabne za naš problem, ki posega na področja umetne inteligence in računalniškega vida. Prednost MATLAB-a so tudi številna orodja za prikazovanje podatkov in izrisovanje grafov. Drugi programskimi jeziki, ki bi bili tudi primerni za implementacijo diplomske naloge, sta Python in Octave. Ena izmed prednost pred Octavom je hitrejše izvajanje kode, saj je MATLAB boljše optimiziran. Python je predstavljal zanimivo alternativo, vendar sem se na koncu odločil za MATLAB, ker je bil algoritem za izgradnjo opisnika

¹Spletna stran MATLAB-a: <https://www.mathworks.com/products/MATLAB.html/>.

AG že spisan v MATLAB-u.

2.2 Knjižnica OpenCV

OpenCV² (*angl. Open Source Computer Vision Library*) je odprtokodna knjižnica za računalniški vid, ki je napisana in optimizirana v C/C++. Prednost knjižnice je uporaba večjedrnega procesiranja, omogočenega z OpenCL. Knjižnica je podprta tudi na več operacijskih sistemih, kot so Windows, Linux in Mac OS. Ker je knjižnica izdana pod licenco BSD, je uporaba dovoljena za akademske, kot tudi komercialne namene [22].

Projekt OpenCV se je uradno začel leta 1999 s strani podjetja Intel. Zajemal je več področij, kot so sledenje žarkom (*angl. ray tracing*) v realnem času, 3D prikaz zidov itd. Prva verzija alfa je na voljo od leta 2000 [37].

2.3 Paket mexopencv

Ker MATLAB za interpretacijo C/C++ jezika potrebuje *mex* funkcije, smo za uporabo knjižnice OpenCV uporabili paket mexopencv³.

Paket vsebuje na stotine vmesnikov za OpenCV API. Paket vsebuje tudi C++ razred, ki pretvori MATLAB podatkovne tipe v OpenCV podatkovne tipe. Paket je primeren za hitro prototipiranje OpenCV aplikacij v MATLAB-u, uporabo OpenCV kot zunanjo orodjarno v MATLAB-U in razvoj svojih *mex* funkcij.

2.4 Knjižnica dLib

Dlib⁴ je splošno namenska knjižnica, napisana v programerskem jeziku C++. Knjižnica je odprtokodna in izdana pod Boost programersko licenco. Od

²Knjižnica OpenCV je dosegljiva na strani: <http://opencv.org/>.

³Paket mexopencv je dosegljiv na strani: <http://kyamagu.github.io/mexopencv/>.

⁴Spletna stran knjižnice dLib: <http://dlib.net/>.

začetka razvoja leta 2002 so se v knjižnico dodala številna nova orodja. Vsebuje orodja za uporabniške vmesnike, strukture podatkov, linearno algebro, strojno učenje, obdelovanje slik [38] itd.

Dlib omogoča tudi gradnjo vmesnikov za MATLAB. V našem primeru smo konkretno zgradili vmesnike za zaznavo obrazov in zaznavo značilnk na obrazu, opisanih v naslednjem poglavju.

Poglavje 3

Predprocesiranje slik

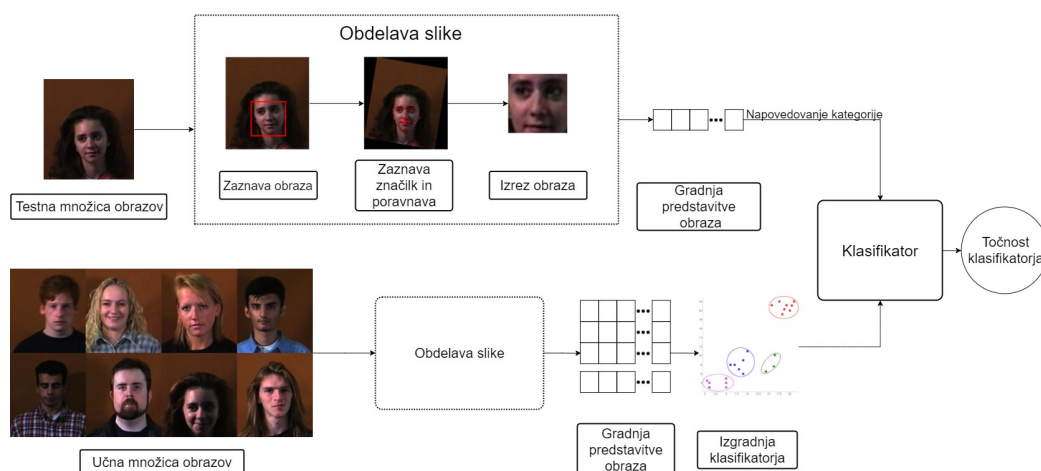
Sistem za prepoznavanje obrazov, ki smo ga implementirali v tej diplomski nalogi je sestavljen iz več delov. Slika 3.1 prikazuje osnovne module iz katerih je sistem sestavljen. V tem poglavju se bomo osredotočili na predprocesiranje slik, ki vključuje zaznavo obraza, izločanje verige obraznih značilk, poravnavo in obrezovanje obraza ter normalizacijo velikosti obraznega dela.

Predprocesirane slike smo shranili in jih uporabljali v nadaljnjih modulih. Za implementacijo prvega dela smo uporabili orodje MATLAB in knjižnico dLib.

3.1 Zaznava obraza

Zaznava obrazov velja za poseben primer zaznave objektov. Primer zaznave obraza prikazuje slika 3.2. Med številnimi algoritmi za zaznavo objektov smo izbrali algoritem za detekcijo obrazov v knjižnici dLib¹. Algoritem uporablja HOG piramido slik, skozi katero pošlje linearni klasifikator. Vsaka raven v piramidi predstavlja HOG značilke, izračunane pri različni resoluciji slike. Bolj podroben opis je na voljo v članku [8]. Algoritem v knjižnici dLib se rahlo razlikuje od prej omenjenega članka [8] in uporablja različico HOG

¹Zaznava slik frontalnih obrazov knjižnice dLib:
http://dlib.net/imaging.html#get_frontal_face_detector/.

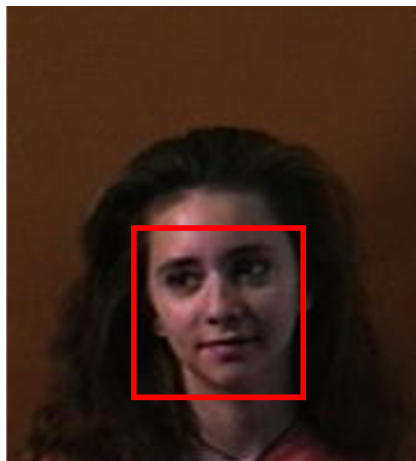


Slika 3.1: Sistem za prepoznavo obrazov.

značilnik iz članka [12]. Razlog je boljša zmogljivost teh HOG značilnik [9].

Glavna ideja algoritma je, da se oblika objekta lahko dobro predstavi z lokalno razporeditvijo intenzitete gradienta ali robov. Pri tem ni potrebno natančno poznavanje položaja gradientov ali robov. Preden začnemo z učenjem detektorja obrazov, najprej povečamo slike za faktor dva, tako lažje zaznamo majhne obraze. Ker so človeški obrazi levo-desno simetrični, z zrcaljenjem slik povečamo učno množico slik. Sledi gradnja piramide slik HOG.

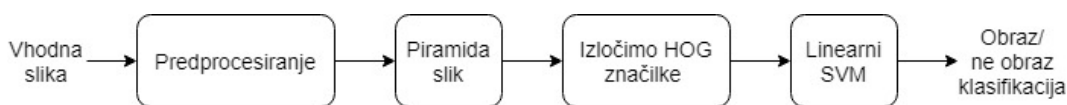
Sliko na vsaki ravni razdelimo v majhne prostorske regije oziroma celice (slika 3.5). Za vsako celico zgradimo lokalni 1D histogram smeri gradientov ali usmeritve roba. Za vsak slikovni element znotraj celice izračunamo magnitudo in smer gradienta. Histogram vsebuje 9 enako širokih razrednih intervalov, ki pokrijejo celotni razpon orientacij od 0° do 360° . V histogramu seštevamo magnitudo gradienta, ki pripadajo slikovnim elementom v celici v ustrezni orientaciji. Posamične histograme zložimo v skupni histogram in tako ustvarimo opisnik. Za manjšo varianco osvetlitve, senc itd. je zelo uporabna normalizacija kontrasta. Normalizacijo kontrasta lahko storimo s kopičenjem meritev lokalnega histograma „energije“ čez velike prostorske regije oziroma bloke in uporabimo rezultate za normalizacijo vseh celic znotraj bloka. Normalizirane bloke bomo poimenovali histogrami usmerjenih



Slika 3.2: Zaznan obraz je označen z rdečim okvirjem.



Slika 3.3: Obraz, predstavljen s HOG značilkami.



Slika 3.4: Pregled verige za pridobivanje značilnk in zaznavo objektov.



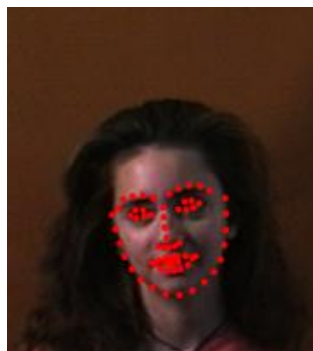
Slika 3.5: Na sliki zeleni kvadratici označujejo celice in moder kvadrat prikazuje blok. Slika je pridobljena iz [28].

gradientov (HOG). Primer predstavitve obraza z značilkami HOG je viden na sliki 3.3. Za zaznavanje objektov uporabimo zaznavalno okno, ki je sestavljeno iz mreže prekrivajočih opisnikov HOG in z uporabo skupnega vektorja značilnk in konvencionalnega klasifikatorja SVM zaključimo verigo za zaznavanje (slika 3.4).

3.1.1 Izboljšava zaznave obraza

Algoritem zaznave obrazov je v večini primerov dobro zaznal prednje obraze (*angl. frontal face detection*), vendar so se včasih pojavile težave. Detekcijo smo izboljšali tako, da smo dano sliko I rotirali za kot $\alpha \in [-30^\circ, 30^\circ]$. Za to smo uporabili funkcijo MATLAB $imrotate(I, \alpha)$.

Za manjše obraze, katere drseče okno ni zaznalo, smo sliko povečali in detekcijo ponovili. Za povečavo smo uporabili funkcijo MATLAB $resize$, ki določi vrednost izhodnim pikslom na podlagi bikubične interpolacije sose-



Slika 3.6: Primer zaznave 68 značilnk na obrazu s knjižnico dLib.

dnjih pikslov.

3.2 Zaznava verige značilnk na obrazu

V tem poglavju si bomo pogledali zaznavo verige značilnk (*angl. landmarks*) na obrazu, ki smo jo uporabili za izločanje in poravnavo obraza. Za to nalogo bomo uporabili algoritem ansambla regresijskih dreves (*angl. face alignment with an ensemble of regression trees*), ki za zaznavo značilnk uporablja gradientno ojačavo (*angl. gradient boosting*). Uporabili smo algoritem, implementiran v knjižnici dLib z modelom za zaznavo 68 obraznih značilnk (slika 3.6). Algoritem sledi modulu za zaznavo obrazov (slika 3.1). Bolj podrobno opis delovanja in rezultatov algoritma si bralec lahko prebere v članku [16], od koder je bila vsebina tudi povzeta.

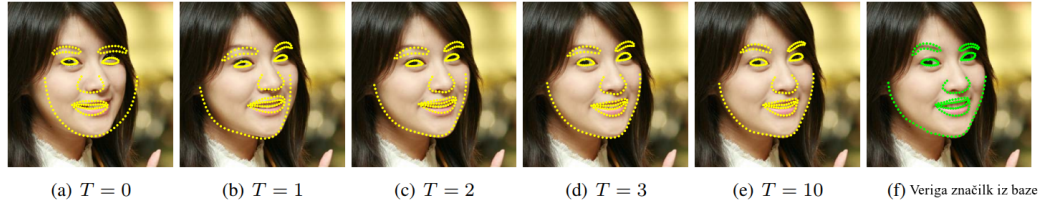
Prednost tega algoritma je, da zazna verigo značilnk na obrazu v milisekundi z veliko natančnostjo. Tudi če obraz delno zakrijemo, algoritem uspešno zazna položaj značilnk. Za uspešno delovanje algoritma sta zaslužna dva dela.

Značilke so lahko zelo različne zaradi deformacije oblike (različni koti opazovanja, različni izrazi na obrazu) in neugodnih dejavnikov, kot je npr. sprememba osvetlitve. To naredi nalogo težko. Potrebujemo zanesljive značilke, da natančno napovemo obliko obraznih delov, na drugi strani pa potrebu-

jemo dobre približke oblike, da pridobimo zanesljive značilke. Težave se lotimo z iterativnim pristopom (kaskade). Namesto, da regresiramo parametre oblike na osnovi značilk v koordinatah slike, rajše sliko transformiramo v normalizirani koordinatni sistem, osnovan na trenutni napovedi oblike in zatem pridobimo značilke, da napovemo posodobljen vektor parametrov oblike. Postopek ponavljamo, dokler ne konvergira. Drugi del se ukvarja z obdelovanjem težavnosti napovedovanja. Algoritem za iskanje značilk mora oceniti obliko, visokodimenzijski vektor, ki se najbolj prilagaja s podatki slike in našim modelom oblike. Težava je nekonveksna z veliko lokalnimi optimumi. Predvideva se, da mora ocenjena oblika biti v linearnem podprostoru, katerega se lahko odkrije z odkritjem glavnih komponent učne množice oblik. Ta predpostavka zmanjša število možnih oblik in pomaga, da se izognemo lokalnemu optimumu. Regresijska funkcija vsebuje oba elementa. Za vsak regresor se naučimo z gradientno ojačavo s kvadratno napako funkcijo izgube (*angl. loss function*). Isto funkcijo izgube želimo optimizirati v testnem času. Raztresena množica pikslov, uporabljena kot vhod regresorja, je izbrana skozi kombinacijo algoritma ojačitve gradientov in apriorne verjetnosti na razdalji med pari vhodnih pikslov. Apriorna razporeditev omogoča ojačitvenemu algoritmu, da učinkovito raziskuje veliko množico relevantnih značilk. Rezultat je kaskada regresorjev, ki lahko lokalizirajo obrazne značilke, ko so inicializirani s povprečno obliko obraza (slika 3.7).

3.3 Poravnava obraza

Obraz smo poravnali glede na položaj oči. Središči obeh oces smo izračunali s pomočjo lokacij obraznih značilk, pridobljenih v prejšnjem koraku. Naj $\mathbf{p}_{i,s}$ označuje koordinati $(x_{i,s}, y_{i,s})$ i -te značilne obrazne točke, v zaporedju značilnih točk, ki pripadajo očesu. Naj velja $i \in \{1, \dots, N\}$, kjer je N število značilk razporejenih okoli očesa in naj s označuje, ali gre za levo ali desno oko ($s \in \{l, r\}$). Središče očesa C_s zapišemo kot:



Slika 3.7: Ocenjene značilke obraza skozi iteracijo. Prva slika na levi prikazuje značilke (veriga rumenih točk), ki pripadajo povprečni obliki obraza, postavljena na sredini zaznanega obraza z Viola-Jones [35] detektorjem obrazov. Vsaka nadaljnja slika je nova ocena oblike obraza v naslednjem koraku iteracije do zadnje slike, ki predstavlja končno obliko obraza oz. verigo značilk, shranjeno v bazi.

$$C_s = (x_{c, s}, y_{c, s}) = \frac{1}{N} \sum_{i=1}^N \mathbf{p}_{i, s}. \quad (3.1)$$

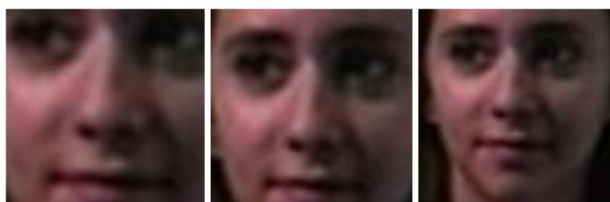
Kot za rotacijo slike α pridobimo iz položaja središča levega in desnega očesa

$$\alpha = \tan^{-1} \left(\frac{y_{c, l} - y_{c, r}}{x_{c, l} - x_{c, r}} \right). \quad (3.2)$$

Za rotacijo slike definiramo rotacijsko matriko R

$$R = \begin{bmatrix} \cos(\alpha) & -\sin(\alpha) & 0 \\ \sin(\alpha) & \cos(\alpha) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.3)$$

Sliko I zarotiramo s funkcijo MATLAB `imwrap(I, R)`. Rotirana slika se zajame z novim okvirjem in nezasedeni piksli se pobarvajo črno. Hkrati ob rotaciji pa moramo koordinate obraznih značilk transformirati v koordinate rotirane slike. Poznamo novo lokacijo zgornjega levega vogala (x_0, y_0) slike v novem koordinatnem sistemu. Nove koordinate obraznih značilk (x', y')



Slika 3.8: Različni tipi obrezave: tip 1, tip 2 in tip 3.

lahko dobimo s transformacijo:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & x_0 \\ 0 & 1 & y_0 \\ 0 & 0 & 1 \end{bmatrix} \left(R \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \right). \quad (3.4)$$

3.4 Obrezovanje obraza

Ko imamo poravnani obraz in poznamo tudi lokacije obraznih značilk, želimo odstraniti rob slike, ki ne vključuje obraza. Ustvarili smo tri različne tipe obrezovanja (slika 3.8):

- (a) tip 1 – zajeto območje zajema oči, usta in nos,
- (b) tip 2 – zajeto območje zajema oči, usta, nos in obrvi,
- (c) tip 3 – zajeto območje zajema celoten obraz.

Glede na dele obraza, ki zajema neki tip obrezovanja, smo določili okvir in obrezali del slike izven okvirja. Okvir smo določili glede na obrazne značilke tako, da smo za ustrezni del obraza izračunali minimalni in maksimalni x in y koordinati, ki smo jih uporabili za izračun štirih vogalnih točk. Velikost slike smo s funkcijo MATLAB *imresize* postavili na 128 pikslov v višino in širino.

Poglavje 4

Kategorizacija obrazov

Ko smo minimizirali vpliv ozadja na sliki, se lahko lotimo klasifikacije. Za klasifikacijo potrebujemo učno množico slik, na podlagi katerih zgradimo klasifikator. Za gradnjo klasifikatorja je na voljo več različnih metod, v našem primeru smo se odločili za metodo *vreča besed* (BOW - *angl. bag of words*), implementirano v MATLAB-u. Obrazne značilnice predstavimo z različnimi opisniki, ki predstavljajo „besede“ v zgoraj navedeni metodi. Po uspešni izgradnji klasifikatorja s pomočjo učne množice, ocenimo uspešnost naučenega klasifikatorja s pomočjo testne množice. Najprej si podrobno pogledjmo metodo BOW, uporabljeno v MATLAB-u in naše načine pridobivanja značilnic in njihovih opisnikov. Podrobno izvajanje metode vreče besed je opisana v članku [7], od koder je vsebina tudi povzeta.

4.1 Metoda vreče besed (BOW)

Metoda BOW, ki se je sprva uporabila v procesiranju naravnega jezika, se lahko uporabi tudi na področju računalniškega vida za klasifikacijo slik tako, da predstavimo lokalne dele slike s pomočjo opisnikov. Glavni koraki metode so:

- Določanje značilnih točk v sliki in gradnja opisnikov. Opisnik opisuje bližnjo okolico značilne točke in je predstavljen kot vektor.

- Razvrščanje opisnikov v roje z algoritmom za kvantizacijo vektorja.
- Gradnja histograma značilnic (*angl. bag of words*), ki prešteje število lokalnih regij, dodeljenih vsakemu roju.
- Uporaba večrazrednega klasifikatorja, kjer histogram značilnic, predstavljen v vektorski obliki, uporabimo za dodeljevanje h kategoriji oz. osebi.

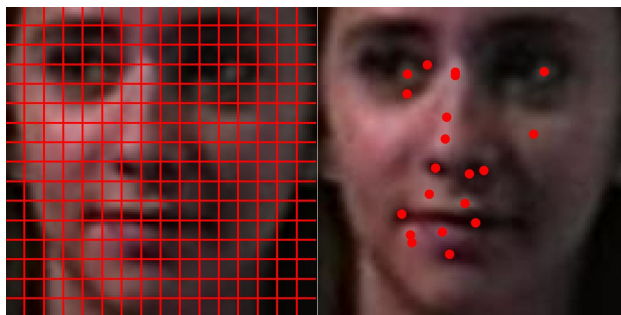
Idealno so ti koraki zasnovani tako, da maksimizirajo klasifikacijsko natančnost, medtem ko minimizirajo računsko zahtevnost. Torej mora za pridobljene opisnike v prvem koraku veljati, da so robustni na spremembe, ki niso povezane s kategorizacijo (npr. razlike zaradi osvetlitve), hkrati pa morajo vsebovati dovolj informacij, ki omogoča, ločevanje med kategorijami. V drugem koraku, pa moramo poskrbeti, da je besednjak dovolj velik, da ločuje pomembne spremembe v delih slike in hkrati ne preveč velik, da ne ločuje med nepomembnimi podrobnostmi, kot je npr. šum.

4.1.1 Pridobivanje značilk

Pridobivanje vektorjev opisnikov deluje tako, da najprej na sliki določimo točke oz. značilke, okoli katerih želimo opisati lokalno regijo. Za to smo uporabili dva pristopa.

Prvi pristop deluje tako, da na sliko postavimo „mrežo“ (slika 4.1 levo). Lokacije, kjer se črte mreže križajo, predstavljajo značilke, za katere izračunamo opisnike. Uporabili smo naslednje lokalne opisnike:

- BRIEF [6],
- BRISK [19],
- FREAK [1],
- KAZE [2],
- LATCH [20],



Slika 4.1: Izbira značilnk. Levi del slike prikazuje izbiro na podlagi mreže, desni del slike prikazuje izbiro z detektorjem ključnih točk. Uporabili smo detektor SURF.

- ORB [31],
- SIFT [23] in
- AG (poglavje 5).

Za vsako značilko na sliki imamo zdaj opisnik v vektorski obliki, ki predstavlja okolico.

Večina algoritmov lokalnih opisnikov vsebuje tudi zaznavo ključnih točk. Zato je drug pristop takšen, da na sliki zaznamo ključne točke (slika 4.1 desno) in zgradimo opisnike za te točke. Za ta korak smo uporabili kombinacije algoritmov za zaznavo ključnih točk in algoritmov za gradnjo opisnikov, prikazanih v tabeli 4.1.

Vsi naštetni lokalni opisniki in detektorji ključnih točk, z izjemo opisnika AG so implementirani v knjižnici OpenCV, katero implementacijo smo tudi uporabili.

4.1.2 Gradnja besednjaka

Pri gradnji klasifikatorja moramo paziti, kako izgradimo vektor značilnk. Če bi vsak opisnik iz prejšnjega koraka predstavljal besedo v slovarju, bi bilo učenje časovno zelo zahtevno. Če bi vzeli drug ekstrem in bi veliko opisnikov dodelili enemu roju oz. besedi, bi izgubili na natančnosti klasifikatorja. Za to

Zaznavalec značilnk	Opisnik
SIFT	SIFT
SURF [5]	SURF
KAZE	KAZE
SIFT	AG
SURF	AG
ORB	AG
KAZE	AG

Tabela 4.1: Kombinacije zaznavalca značilnk in opisnika, uporabljenega v algoritmu BOW.

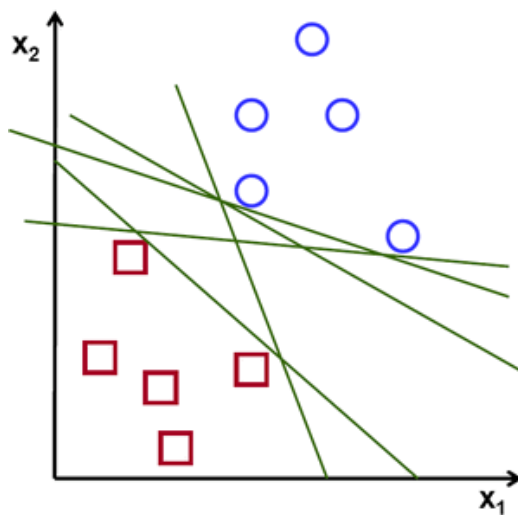
nalogo uporabimo algoritem k -means [26]. Za vsak obraz zgradimo frekvenčni histogram za besede določene z algoritmom k -means. Histogram predstavlja vektor s katerim predstavimo obraz.

4.1.3 Kategorizacija s klasifikatorjem SVM

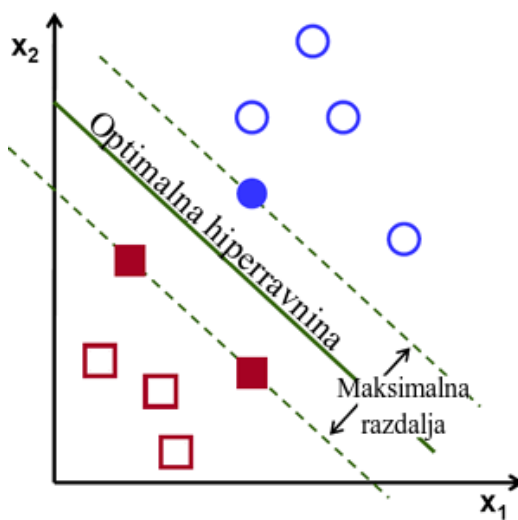
Ko pridobimo histograme, se lotimo težave kategorizacije. Kategorizacija vključuje učenje kategorizatorja in njegovo testiranje. Za to nalogo uporabimo klasifikator SVM.

Klasifikator SVM najde hiperravnino, ki razdeli prostor podatkov v dva razreda. Ker obstaja več hiperravnin (slika 4.2), ki ločujejo dva razreda, za ločevanje uporabimo optimalno hiperravnino. Optimalna hiperravnina (slika 4.3) je določena tako, da maksimizira razdaljo (*angl. margin*) najbližjih učnih podatkov do hiperravnine.

SVM se v osnovi uporablja za klasifikacijo dveh razredov. Za razširitev na večrazredno klasifikacijo moramo opraviti primerjavo vseh možnih razredov N med seboj, torej končno število primerjav znaša $N \cdot (N - 1)/2$.



Slika 4.2: Hiperravnine, ki ločujejo dva razreda, prikazana kot krogi in kvadrati. Slika, pridobljena s spletne strani [27].



Slika 4.3: Optimalna hiperravnina, ki maksimizira razdaljo med razredoma. Slika, pridobljena s spletne strani [27].

Poglavje 5

Alfa-Gama opisnik

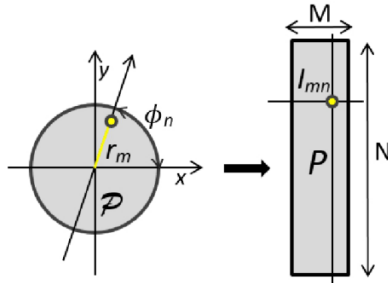
V tem poglavju si bomo podrobno pogledali delovanje povsem novega opisnika AG, ki so ga razvili v Laboratoriju za vizualne in spoznavne sisteme na Fakulteti za računalništvo in informatiko v Ljubljani. Opisnik je osnovan na modelu odvisnih učinkov (*angl. Dependent Effects Model* ali krajše DEM) [24].

5.1 Predstavitev lokalne regije z DEM-om

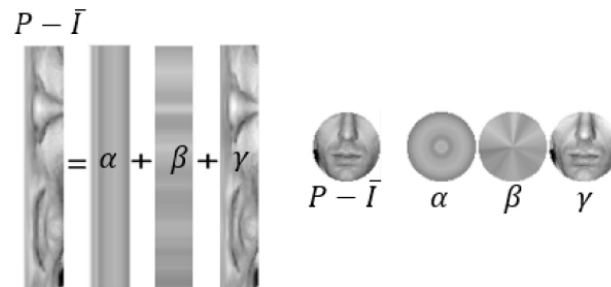
Naj bo \mathcal{P} okrogla lokalna regija, ki jo vzorčimo. Vzorce jemljemo iz krožnic M z radijem $r_m = m$; $m = 0 \dots M-1$ in za N kotov, velikosti $\phi_n = \frac{2\pi n}{N}$; $n = 0, \dots, N-1$. Na ta način prevedemo \mathcal{P} v pravokotnik P . Prevedbo ilustrira slika 5.1. Vrednost intenzitete na lokaciji (r_m, ϕ_n) pripada m -temu stolpcu in n -ti vrstici P -ja. To vrednost označimo kot $I_{mn} = I(m, n)$.

Naj C_m in R_n označujeta seštevek intenzitet iz m -tega stolpca in n -te vrstice pravokotnika P . Velja $C_m = \sum_{n=0}^{N-1} I_{mn}$ in $R_n = \sum_{m=0}^{M-1} I_{mn}$. Pravokotnik P lahko predstavimo s tremi tipi učinkov: učinki stolpcev α_m , učinki vrstic β_n , in učinki preostanka γ_{mn} , za $m = 0 \dots M-1$ in $n = 0 \dots N-1$. Vsak I_{mn} predstavimo, kot linearno kombinacijo treh zgoraj navedenih učinkov (slika 5.2)

$$I_{mn} = \bar{I} + \alpha_m + \beta_n + \gamma_{mn}. \quad (5.1)$$



Slika 5.1: Transformacija okrogle lokalne regije \mathcal{P} v pravokotnik P s polar-
nim vzorčenjem. Slika povzeta iz članka [30].



Slika 5.2: Dekompozicija pravokotnik $P - \bar{I}$ v α, β in γ učinke. Opazimo,
da so vse vrstice v α pravokotniku enake. V β pravokotniku so med seboj
enaki stolpci. Elementi vrstic α sovpadajo z α učinki in elementi stolpcev
 β sovpadajo z β učinki. γ učinki sovpadajo s pravokotnikom γ . Slika, pov-
zeta iz članka [30].

\bar{I} označuje povprečno vrednost izračunano na intenzitetah pravokotnika P .

$$\bar{I} = \frac{1}{MN} \cdot \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} I_{mn}, \quad (5.2)$$

α_m in β_n označujeta razliko povprečne vrednosti stolpca m in vrstice n od povprečne vrednosti \bar{I} :

$$\alpha_m = \frac{1}{N} \cdot \mathcal{C}_m - \bar{I}, \quad (5.3)$$

$$\beta_n = \frac{1}{M} \cdot \mathcal{R}_n - \bar{I}. \quad (5.4)$$

Učinke preostanka γ_{mn} lahko izrazimo kot:

$$\gamma_{mn} = I_{mn} - \bar{I} - \alpha_m - \beta_n. \quad (5.5)$$

5.2 Odvečnost vektorja učinkov β

Središčna točka okrogle regije \mathcal{P} je transformirana v vse piksele prvega stolpca pravokotnika P :

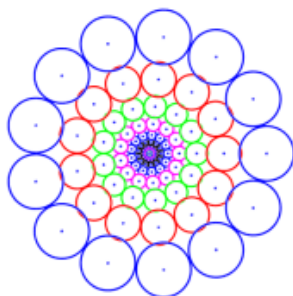
$$I_0 = I_{00} = I_{01} = I_{02} = \dots = I_{0(N-1)}.$$

Za prvi stolpec tako velja

$$\mathcal{C}_0 = \sum_{n=0}^{N-1} I_{0n} = N \cdot I_0. \quad (5.6)$$

Če vstavimo enačbo (5.6) v (5.3), dobimo:

$$\alpha_0 = \frac{1}{N} \cdot \mathcal{C}_0 - \bar{I} = \frac{1}{N} \cdot N \cdot I_0 - \bar{I} = I_0 - \bar{I}. \quad (5.7)$$



Slika 5.3: Vzorec vzorčenja opisnika AG. Modre točke predstavljajo mesta kjer vzorčimo z Gaussovimi filtrom. Polmeri barvnih krogov ustrezajo standardnemu odklonu Gaussovega filtra. Vzorec je sestavljen iz 9 krožnic. Na vsaki krožnici zajamemo 13 vrednosti. Slika, povzeta iz članka [30].

Vstavimo enačbo (5.7) v (5.5):

$$\begin{aligned}\gamma_{0n} &= I_{0n} - (\bar{I} + \alpha_0 + \beta_n) \\ \gamma_{0n} &= I_0 - (\bar{I} + I_0 - \bar{I} + \beta_n) = -\beta_n.\end{aligned}\tag{5.8}$$

Ker je vektor β učinkov že prisoten v prvem stolpcu γ učinkov, je dovolj, če v opisniku hranimo samo α in γ učinke. Od tod sledi tudi ime opisnika AG kot okrajšava za alfa-gama.

5.3 Robustni deskriptor AG

Slika objekta se spremeni s spremembo točke opazovanja, ali ko spremenimo osvetlitev objekta. Da bi bil opisnik odporen na tovrstne spremembe smo uporabili naslednje tehnike.

5.3.1 Vzorec vzorčenja

Vzorčimo na krožnicah z eksponentno naraščajočim radijem. Zajete vrednosti so Gaussovega povprečja, izračunana na okoliških pikslih mesta vzorčenja.

Standardni odklon filtrov narašča eksponentno s številko kroga. Z naslednjimi enačbami določimo radij m -te krožnice r_m , na kateri vzorčimo in standardni odklon σ_m Gaussovih filtrov:

$$\Delta\sigma_m = \Delta\sigma_0 \cdot \text{step}^m, \quad (5.9)$$

$$\sigma_m = \sqrt{\Delta\sigma_m^2 + \sigma_0^2}, \quad (5.10)$$

$$r_m = r_{m-1} + \Delta\sigma_m, \quad (5.11)$$

za $m = 1, \dots, M - 1$. Vrednosti za $r_0, \text{step}, \Delta\sigma_0$ in σ_0 so konstantni parametri opisnika.

5.3.2 Normalizacija vrednosti učinkov

Z normalizacijo želimo narediti opisnik robusten na spremembo kontrasta. Vrednosti učinkov normaliziramo s standardnimi odkloni, izračunanimi za vektor α učinkov in pravokotnik γ učinkov. Velja:

$$\hat{\alpha} = \left(\frac{\alpha_0}{\sigma_\alpha}, \frac{\alpha_1}{\sigma_\alpha}, \dots, \frac{\alpha_{M-1}}{\sigma_\alpha} \right) \quad (5.12)$$

in

$$\hat{\gamma} = \begin{pmatrix} \frac{\gamma_{00}}{\sigma_{\gamma 0}} & \frac{\gamma_{10}}{\sigma_{\gamma 1}} & \dots & \frac{\gamma_{(M-1),0}}{\sigma_{\gamma(M-1)}} \\ \frac{\gamma_{01}}{\sigma_{\gamma 0}} & \frac{\gamma_{11}}{\sigma_{\gamma 1}} & \dots & \frac{\gamma_{(M-1),1}}{\sigma_{\gamma(M-1)}} \\ \vdots & \ddots & \vdots & \\ \frac{\gamma_{0,(N-1)}}{\sigma_{\gamma 0}} & \frac{\gamma_{1,(N-1)}}{\sigma_{\gamma 1}} & \dots & \frac{\gamma_{(M-1),(N-1)}}{\sigma_{\gamma(M-1)}} \end{pmatrix}, \quad (5.13)$$

kjer so σ_α in $\sigma_{\gamma m}; m = 0 \dots M - 1$ ocenjeni standardni odkloni, izračunani z naslednjima enačbama

$$\sigma_\alpha = \sqrt{\frac{\sum_{m=0}^{M-1} \alpha_m^2}{M-1}}, \quad \sigma_{\gamma m} = \sqrt{\frac{\sum_{n=0}^{N-1} \gamma_{mn}^2}{N-1}}. \quad (5.14)$$

Poglavje 6

Podatkovne baze

Predlagano metodo smo uporabili na štirih podatkovnih bazah, ki se po težavnosti razlikujejo.

6.1 Zbirka slik obrazov Univerze v Essexu

Zbirka¹ je sestavljena iz treh podatkovnih baz: „faces94“, „faces95“ in „faces96“. Težavnost narašča po vrsti tako, da je „faces96“ najtežja baza. Baza „faces96“ je težja zaradi sprememb ozadja in skale.

Podatkovna baza „faces94“ je sestavljena iz slik 153 posameznikov, med katerimi je 20 žensk in 133 moških. V bazi je skupno prisotnih 3059 slik. Ozadje je zeleno platno. Ne vsebuje sprememb v osvetlitvi, kot tudi ne vsebuje različne velikosti obraza. Obrazi so lahko rahlo nagnjeni, prisotna pa je tudi sprememba v izrazih na obrazu.

Podatkovna baza „faces95“ vsebuje 72 posameznikov na 1440 slikah. Ozadje je rdeča zavesa, ki vsebuje sence, ustvarjene s strani posameznika. S premikanjem posameznika naprej se zgodijo velike spremembe v osvetlitvi zaradi umetnega osvetljevanja. Prisotne so rahle spremembe v velikosti obraza, nagibu glave in izrazih na obrazu. Na sliki je obraz lahko zamaknjen.

Podatkovna baza „faces96“ vsebuje 152 posameznikov na 3016 slikah. Iz

¹Zbirka Univerze v Essexu je dosegljiva na: <http://cswww.essex.ac.uk/mv/allfaces/>.



Slika 6.1: Primeri slik iz zbirke slik obrazov Univerze v Essexu. Prva vrstica predstavlja slike iz „faces94“, druga iz „faces95“ in tretja vrstica iz podatkovne baze „faces96“.

podatkovne baze so bile izločene 3 slike, na katerih nismo zaznali obraza, ker naš detektor obrazov in značilnik zaznava prednji del obraza in ne stranskega profila obraza, ki je prisoten na teh slikah. Ozadje je kompleksno, saj vsebuje bleščeče posterje. Prisotne so spremembe v velikosti, izrazih in nagibu obraza. Sprememba osvetlitve se spreminja, ko se posameznik pomakne naprej.

6.2 Podatkovna baza UFI

Podatkovna baza UFI [18] (*angl. Unconstrained Facial Images*)². Slike so pridobljene iz Češke novinarske agencije (ČTK). Sestavljena je iz dveh različnih delov: en del vsebuje obrezane slike, drugi velike slike brez obrezave. V našem primeru bomo testiranje izvedli na obrezanih slikah.

Obrezane slike so sestavljene iz slik 605 posameznikov, kjer ima vsak posameznik v povprečju 7,1 sliko za učenje in eno za testiranje. Slike so

²Podatkovna baza UFI je dosegljiva na: <http://www.face-rec.org/databases/>.



Slika 6.2: Primer obrezanih slik dveh posameznikov iz podatkovne baze UFI.

črno-bele in so velikosti 128 pikslov v širino in višino. Slike vsebujejo poleg obraza, pridobljenega z Viola-Jones algoritmom [35], tudi malo ozadja. Nekaj slik iz obrezane podatkovne baze prikazuje slika 6.2.

Poglavje 7

Rezultati

Ker imamo več možnih nastavitev parametrov in omejeno število virov, bomo poizkusili odkriti najboljšo kombinacijo parametrov za najboljše rezultate. Začeli bomo z izbiro tipa obrezave.

7.1 Izbira tipa obrezave obraza

Za odločitev o izbiri najbolj primerne tipa obrezave obraza smo uporabili podatkovno bazo „faces94“. Za ocenitev primernosti tipa obrezave obraza smo uporabili povprečno klasifikacijsko točnost. Klasifikacijska točnost je delež pravilno napovedanih oseb, ki jim obrazi pripadajo, glede na vse testirane obraze. Podatkovno bazo smo naključno razdelili na učno in testno množico. Učno množico predstavlja 80 % vseh slik, testno množico pa preostalih 20 % slik. Parameter K , ki predstavlja število rojev za algoritem *k-means*, smo nastavili na 500 in izvedli 10 ponovitev. Ker določamo najboljši tip obrezave obraza, smo testiranje izvedli za vse tipe obrezav.

Za pridobivanje opisnikov značilk, določenih na podlagi mreže z velikostjo celic 8×8 pikslov, smo za podatkovno bazo „faces94“ dobili rezultate, prikazane v tabeli 7.1. Kot je razvidno iz tabele, obrezava tipa 2 dosega najvišjo klasifikacijsko točnost, zato bomo v nadaljnje uporabljali obrezavo tipa 2.

	BRIEF	BRISK	FREAK	KAZE	LATCH	ORB	SIFT	AG	povp.
Tip 1	98,12 %	96,63 %	92,70 %	97,75 %	97,81 %	96,85 %	96,32 %	98,79 %	96,87 %
Tip 2	98,86 %	97,86 %	95,59 %	98,91 %	98,55 %	98,71 %	97,84 %	99,15 %	98,18 %
Tip 3	97,71 %	98,10 %	94,25 %	99,08 %	98,09 %	97,40 %	98,24 %	99,22 %	97,76 %

Tabela 7.1: Klasifikacijska točnost na bazi „faces94“ z izbiro značilnk na podlagi mreže ob uporabi različnih tipov obrezave. Testiranje je bilo izvedeno za opisnike BRIEF, BRISK, FREAK, KAZE, LATCH, ORB, SIFT in AG. Zadnji stolpec prikazuje povprečno klasifikacijsko, točnost izračunano na vseh testiranih opisnikih.

Za pridobivanja opisnikov značilnk, določenih z detektorjem ključnih točk, smo za podatkovno bazo „facs94“ dobili rezultate, prikazane v tabeli 7.2. Za najboljšo se je izkazala obrezava tipa 3.

Zaznavalec značilnk	SIFT	SURF	KAZE	SIFT	SURF	ORB	KAZE	
Opisnik	SIFT	SURF	KAZE	AG	AG	AG	AG	povp.
Tip 1	98,82 %	98,71 %	99,18 %	98,94 %	99,07 %	93,99 %	99,23 %	98,28 %
Tip 2	99,10 %	99,12 %	99,26 %	99,08 %	99,18 %	98,68 %	99,30 %	99,10 %
Tip 3	99,22 %	99,02 %	99,30 %	99,30 %	99,26 %	99,22 %	99,31 %	99,23 %

Tabela 7.2: Klasifikacijska točnost, pridobljena na bazi „faces94“ z opisniki, izračunanimi na podlagi detektorja značilnk ob uporabi različnih tipov obrezave. Zadnji stolpec prikazuje povprečno klasifikacijsko točnost za vse testirane opisnike.

Ti rezultati ne potrjujejo niti ne zavračajo, da sta izbrana tipa obrezave najboljša. Želeli smo izbrati zgolj tip obrezave, ki se nasplošno izkaže za dober. Izkazalo se je, da gre za majhne razlike v rezultatih, tip obrezave nima velik vpliv na končni rezultat.

7.2 Rezultati na vseh bazah

Sprva si pogledjmo rezultate s pridobivanjem opisnikov značilnk na podlagi mreže. Parametri so nastavljeni enako kot v prejšnjem podpoglavju. Uporabili smo obrezavo tipa 2. Samo v primeru podatkovne baze UFI smo pri učenju in testiranju uporabljali učno in testno množico, kot je bila vnaprej določena z bazo. Za ostale baze velja, da učna in testna množica predstavlja 80 % oz. 20 % vseh slik. Rezultati so prikazani v tabeli 7.3.

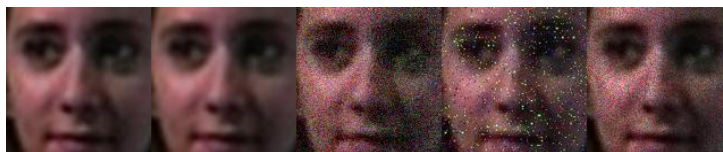
	BRIEF	BRISK	FREAK	KAZE	LATCH	ORB	SIFT	AG
faces94	98,86 %	97,86 %	95,59 %	98,91 %	98,55 %	98,71 %	97,84 %	99,15 %
faces95	98,02 %	94,93 %	90,52 %	97,12 %	97,15 %	97,65 %	95,97 %	97,81 %
faces96	96,37 %	92,63 %	86,79 %	96,14 %	95,83 %	95,41 %	93,84 %	97,43 %
UFI	19,80 %	14,26 %	7,12 %	13,75 %	12,07 %	18,89 %	8,66 %	17,77 %

Tabela 7.3: Klasifikacijska točnost, izračunana na bazah „faces94“, „faces95“, „faces96“ in UFI. Ključne točke so določene z mrežo.

Tabela 7.4 prikazuje rezultate, pridobljene z drugo metodo, kjer ključne točke pridobimo s pomočjo detektorjev. Uporabili smo obrezavo tipa 3.

Zaznavalec značilnk	SIFT	SURF	KAZE	SIFT	SURF	ORB	KAZE
Opisnik	SIFT	SURF	KAZE	AG	AG	AG	AG
faces94	99,22 %	99,02 %	99,30 %	99,30 %	99,26 %	99,22 %	99,31 %
faces95	98,65 %	98,47 %	99,03 %	98,92 %	99,03 %	96,42 %	99,48 %
faces96	95,10 %	95,89 %	94,32 %	94,85 %	96,87 %	58,66 %	97,22 %
UFI	18,10 %	19,88 %	25,74 %	23,02 %	29,36 %	21,74 %	32,23 %

Tabela 7.4: Klasifikacijska točnost, izračunana na bazah „faces94“, „faces95“, „faces96“ in UFI. Za izbiro ključnih točk smo uporabili različne detektorje.



Slika 7.1: Iz leve proti desni vidimo originalno sliko, spremembo z Gassovo zameglitvijo, Gassovim šumom, šumom sol in poper in točkasti šum.

7.3 Rezultati ob prisotnosti šuma ali zameglitve

Preizkusiti smo želeli tudi robustnost opisnikov ob prisotnosti šumov in zameglitve (slika 7.1). Ker nam zaznava ključnih točk z detektorjem lahko vrne prazno množico točk, smo se odločili za pridobivanje točk na podlagi mreže. Za testiranje smo uporabili bazo „faces96“, s parametrom K , nastavljenim na 500, velikostjo kvadratka v mreži 8×8 pikslov in obrezavo tipa 2. Opravili smo 5 ponovitev in množico slik obrazov razdelili na testno in učno množico, kjer prva predstavlja 20 % vseh slik in slednja 80 % vseh slik. Uporabili smo Gaussovo zameglitev, s standardnim odklonom nastavljenim na 1. Uporabili smo tudi več različnih šumov. Med drugim Gaussov šum s povprečjem 0 in varianco 0,01. Omeniti je treba, da je slika predstavljena z vrednostmi na intervalu $[0,1]$ in ne na intervalu $[0,255]$, saj bi bile spremembe drugače praktično neopazne. Uporabili smo tudi točkasti šum (*angl. speckle noise*), to je multiplikativni šum do slike I , ki je določen z uporabo enačbe $J = I + n * I$, kjer je n enakomerno porazdeljen naključni hrup s povprečjem 0 in varianco v . Vrednost v je bila nastavljena na 0,04. Zadnji šum, ki smo ga uporabili, je šum, imenovan sol in poper. Šum sol in poper spremeni slikovne elemente v črno oziroma belo barvo. Delež teh elementov je v našem primeru znašal 5 %. Rezultati so prikazani v tabeli 7.5.

	BRIEF	BRISK	FREAK	KAZE	LATCH	ORB	SIFT	AG
Gaussova zameglitev	95,63 %	90,59 %	82,38 %	94,20 %	94,93 %	94,90 %	92,55 %	96,75 %
Gaussov šum	93,97 %	59,83 %	55,00 %	84,03 %	87,65 %	91,89 %	80,40 %	70,37 %
Točkasti šum	95,53 %	82,98 %	72,35 %	91,43 %	94,27 %	94,17 %	89,90 %	91,09 %
Šum sol in poper	92,81 %	48,34 %	44,70 %	78,51 %	62,25 %	87,12 %	72,78 %	57,25 %

Tabela 7.5: Klasifikacijska točnost, izračunana na bazi „faces96“ ob prisotnosti šuma oz. zameglitve. Za izbiro ključnih točk smo uporabili mrežo.

7.4 Interpretacija rezultatov

Metodi pridobivanja značilk s pomočjo mreže in detektorja ključnih točk sta se izkazali za zelo uspešni na podatkovnih bazah Univerze v Essexu. V primeru nenadzorovanega okolja so bili rezultati občutno slabši, kar nakazuje, da bi bilo treba izboljšati trenutno metodo ali uporabiti drugo metodo, ki bi bolje delovala v nenadzorovanem okolju. Na drugi strani, pa se je podatkovna baza UFI izkazala za dobro pri izpostavitvi sprememb opisnikov. Vidimo lahko, da je opisnik AG med tremi najboljšimi v primeru pridobivanja značilk s pomočjo mreže. Pridobivanje značilk na podlagi točk, pridobljenih z detektorjem KAZE in opisnikom AG je na podatkovni bazi UFI prikazala najboljši rezultat z 32,23 %. Na splošno lahko rečemo, da se je opisnik AG izkazal za najboljšega oziroma enega izmed najboljših opisnikov. V drugi fazi testiranja, kjer smo uporabili različne šume in Gaussovo zameglitev, se je najbolj izkazal opisnik BRIEF. Opisnik AG se je dobro izkazal ob prisotnosti Gaussove zameglitve in točkastega šuma. Največji padec v točnosti opisnika AG je bil ob uporabi šuma sol in poper. Ena izmed možnih izboljšav v prihodnje bi bila uporaba mediana filtra za odstranitev tega šuma.

Rezultate bi lahko izboljšali, če bi povečali parameter K in nastavili bolj

gosto mrežo. Razlog, da tega nismo storili, je omejitev v številu virov. Kljub temu smo poizkusili celoten postopek pohitriti s paralelnim izvajanjem na štirih jedrih procesorja in opravljanje nekaterih operacij na grafični kartici.

Poglavje 8

Sklepne ugotovitve

V diplomskem delu smo opisali delovanje detektorja obrazov in obraznih značilk, implementiranih v knjižnici dLib. Izboljšali smo detekcijo obraza in obraz na podlagi pridobljenih obraznih značilk tudi poravnali. Uporabili smo tri tipe obrezave obraza na podlagi zaznanih obraznih značilk.

Opisali smo, kako smo algoritem BOW in klasifikator SVM uporabili za prepoznavo obrazov. Implementirali smo dve različni metodi za pridobivanje značilk, ki jih uporabimo kot vhod za algoritem BOW. Prva metoda pridobi točke na sliki na podlagi mreže in njihovo okolico zapiše kot vektor. Druga metoda pa točke pridobi na podlagi detektorja ključnih točk in prav tako opiše njihovo okolico.

Predstavili smo tudi delovanje popolnoma novega opisnika AG, ki je bil uporabljen v tej nalogi. Poleg opisnika AG smo za sistem prepoznave obrazov uporabili številne priljubljene lokalne opisnike. Testiranje smo izvedli tudi na slikah z različnimi šumi in Gaussovo zameglitvijo. Za namen testiranja smo uporabili javno dostopne baze iz Univerze v Essexu in UFI. Rezultati so bili na podatkovnih bazah Univerze v Essexu izjemni, na bazi UFI pa občutno slabši.

Ena izmed možnihboljšav bi bila uporaba BBOW (*block-based bag of words*) iz članka [21]. Hkrati se izbira uporabe MATLAB-a ni izkazala za najboljšo, saj nismo mogli dostopati do vseh orodij, ki jih omogoča OpenCV.

Predvsem bi bilo dobrodošlo, da bi lahko uporabljali funkcije, ki se izvajajo na grafični kartici in tako celoten postopek pohitrili.

Literatura

- [1] Alexandre Alahi, Raphael Ortiz, and Pierre Vandergheynst. Freak: Fast retina keypoint. In *Computer vision and pattern recognition (CVPR), 2012 IEEE conference on*, pages 510–517. Ieee, 2012.
- [2] Pablo Fernández Alcantarilla, Adrien Bartoli, and Andrew J. Davison. Kaze features. In *European Conference on Computer Vision*, pages 214–227. Springer, 2012.
- [3] Dana H. Ballard and Christopher M. Brown. *Computer vision*. Prentice-Hall, 1982.
- [4] Marian Stewart Bartlett, Javier R. Movellan, and Terrence J. Sejnowski. Face recognition by independent component analysis. *IEEE Transactions on neural networks*, 13(6):1450–1464, 2002.
- [5] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Speeded-up robust features (surf). *Computer vision and image understanding*, 110(3):346–359, 2008.
- [6] Michael Calonder, Vincent Lepetit, Christoph Strecha, and Pascal Fua. Brief: Binary robust independent elementary features. *Computer Vision–ECCV 2010*, pages 778–792, 2010.
- [7] Gabriella Csurka, Christopher R. Dance, Lixin Fan, Jutta Willamowski, and Cédric Bray. Visual categorization with bags of keypoints. In *In Workshop on Statistical Learning in Computer Vision, ECCV*, pages 1–22, 2004.

-
- [8] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886–893. IEEE, 2005.
- [9] dlib C Library Image Processing. Dosegljivo: http://dlib.net/imaging.html#scan_fhog_pyramid. Pridobljeno 8. 8. 2017.
- [10] Kamran Etemad and Rama Chellappa. Discriminant analysis for recognition of human face images. *JOSA A*, 14(8):1724–1733, 1997.
- [11] Facebook. Dosegljivo: <https://www.facebook.com>. Pridobljeno 10. 7. 2017.
- [12] Pedro F. Felzenszwalb, Ross B. Girshick, David McAllester, and Deva Ramanan. Object detection with discriminatively trained part-based models. *IEEE transactions on pattern analysis and machine intelligence*, 32(9):1627–1645, 2010.
- [13] A Jay Goldstein, Leon D Harmon, and Ann B Lesk. Identification of human faces. *Proceedings of the IEEE*, 59(5):748–760, 1971.
- [14] Google+. Dosegljivo: <https://plus.google.com>. Pridobljeno 10. 7. 2017.
- [15] T. Huang. *Computer vision: Evolution and promise*. 1996.
- [16] Vahid Kazemi and Josephine Sullivan. One millisecond face alignment with an ensemble of regression trees. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1867–1874, 2014.
- [17] Michael Kirby and Lawrence Sirovich. Application of the karhunen-loeve procedure for the characterization of human faces. *IEEE Transactions on Pattern analysis and Machine intelligence*, 12(1):103–108, 1990.

-
- [18] L. Lenc and P. Král. Unconstrained Facial Images: Database for face recognition under real-world conditions. In *14th Mexican International Conference on Artificial Intelligence (MICAI 2015)*, Cuernavaca, Mexico, 25–31 October 2015. Springer.
- [19] Stefan Leutenegger, Margarita Chli, and Roland Y Siegwart. Brisk: Binary robust invariant scalable keypoints. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 2548–2555. IEEE, 2011.
- [20] Gil Levi and Tal Hassner. Latch: learned arrangements of three patch codes. In *Applications of Computer Vision (WACV), 2016 IEEE Winter Conference on*, pages 1–9. IEEE, 2016.
- [21] Zisheng Li, Jun-ichi Imai, and Masahide Kaneko. Robust face recognition using block-based bag of words. In *Pattern Recognition (ICPR), 2010 20th International Conference on*, pages 1285–1288. IEEE, 2010.
- [22] OpenCV library. Dosegljivo: <http://opencv.org/>. Pridobljeno 17. 8. 2017.
- [23] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.
- [24] Jasna Maver. Self-similarity and points of interest. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(7):1211–1226, 2010.
- [25] Patricia Melin and Witold Pedrycz. *Soft Computing for Recognition Based on Biometrics*, volume 312. Springer, 2010.
- [26] D. G. Stork O. Duda, P. E. Hart. *Pattern classification*. John Wiley & Sons, 2000.
- [27] OpenCV. Introduction to support vector machines. Dosegljivo: http://docs.opencv.org/2.4/doc/tutorials/ml/introduction_to_svm/introduction_to_svm.html. Pridobljeno 10. 8. 2017.

-
- [28] Learn OpenCV. Dosegljivo: <http://www.learnopencv.com/histogram-of-oriented-gradients/>. Pridobljeno 13. 8. 2017.
- [29] Dimitri Pissarenko. Eigenface-based facial recognition. *December 1st*, 2002.
- [30] Mandeljc R., Skočaj D., and Maver J. Afs: local descriptors based on dependent effects model.
- [31] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary Bradski. Orb: An efficient alternative to sift or surf. In *Computer Vision (ICCV), 2011 IEEE international conference on*, pages 2564–2571. IEEE, 2011.
- [32] Lawrence Sirovich and Michael Kirby. Low-dimensional procedure for the characterization of human faces. *Josa a*, 4(3):519–524, 1987.
- [33] Milan Sonka, Vaclav Hlavac, and Roger Boyle. *Image processing, analysis, and machine vision*. Cengage Learning, 2014.
- [34] Matthew A. Turk and Alex P. Pentland. Face recognition using eigenfaces. In *Computer Vision and Pattern Recognition, 1991. Proceedings CVPR'91., IEEE Computer Society Conference on*, pages 586–591. IEEE, 1991.
- [35] Paul Viola and Michael J. Jones. Robust real-time face detection. *International journal of computer vision*, 57(2):137–154, 2004.
- [36] Wikipedia. Matlab. Dosegljivo: <https://en.wikipedia.org/wiki/MATLAB>. Pridobljeno 17. 8. 2017.
- [37] Wikipedia. Opencv. Dosegljivo: <https://en.wikipedia.org/wiki/OpenCV>. Pridobljeno 17. 8. 2017.
- [38] Wikipedia. Opencv. Dosegljivo: <https://en.wikipedia.org/wiki/Dlib>. Pridobljeno 17. 8. 2017.