

Zbornik strokovne konference

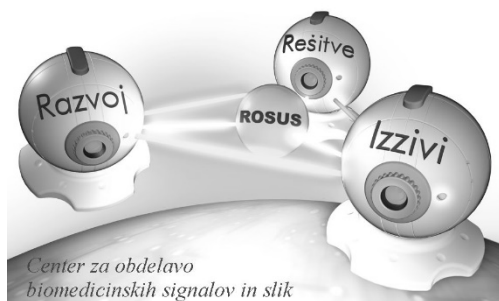
ROSUS 2019

Računalniška obdelava slik in
njena uporaba v Sloveniji 2019

Urednik:
Božidar Potočnik



Maribor, 21. marec 2019



ROSUS 2019 - Računalniška obdelava slik in njena uporaba v Sloveniji 2019

Zbornik 14. strokovne konference

Urednik
dr. Božidar Potočnik

Maribor, marec 2019

Naslov	ROSUS 2019 - Računalniška obdelava slik in njena uporaba v Sloveniji 2019
Podnaslov	Zbornik 14. strokovne konference
Title	ROSUS 2019 - Computer image processing and its application in Slovenia 2019
Subtitle	Proceedings of the 14 th Professional Conference
Urednik <i>Editor</i>	izr. prof. dr. Božidar Potočnik (Univerza v Mariboru, Fakulteta za elektrotehniko, računalništvo in informatiko)
Recenzija <i>Review</i>	red. prof. dr. Aleš Holobar (Univerza v Mariboru, Fakulteta za elektrotehniko, računalništvo in informatiko) izr. prof. dr. Simon Dobrišek (Univerza v Ljubljani, Fakulteta za elektrotehniko)
Tehnični urednik <i>Technical editor</i>	Jan Perša, mag. inž. prom. (Univerzitetna založba Univerze v Mariboru)
Oblikovanje ovitka <i>Cover designer</i>	Gašper Sedej, mag. inž. (Univerza v Mariboru, Fakulteta za elektrotehniko, računalništvo in informatiko)
Konferenca <i>Conference</i>	Strokovna konferenca ROSUS 2019 - Računalniška obdelava slik in njena uporaba v Sloveniji 2019
Datum konference <i>Date and location</i>	21. marec 2019, Maribor, Slovenija
Grafične priloge <i>Graphics material</i>	Avtorji

Izdajatelj / Co-published by

Univerza v Mariboru, Fakulteta za elektrotehniko, računalništvo in informatiko
Koroška cesta 46, 2000 Maribor, Slovenija
<http://feri.um.si>, feri@um.si

Založnik / Published by

Univerzitetna založba Univerze v Mariboru
Slomškov trg 15, 2000 Maribor, Slovenija
<http://press.um.si>, zalozba@um.si

Izdaja
Edition Prva izdaja

Vrsta publikacije
Publication type E-knjiga

Dostopno na
Available at <http://press.um.si/index.php/ump/catalog/book/393>
<https://rosus.feri.um.si/rosus2019>

Izdano
Published Maribor, marec 2019

© Univerza v Mariboru, Univerzitetna založba

Vse pravice pridržane. Brez pisnega dovoljenja založnika je prepovedano reproduciranje, distribuiranje, predelava ali druga uporaba tega dela ali njegovih delov v kakršnemkoli obsegu ali postopku, vključno s fotokopiranjem, tiskanjem ali shranjevanjem v elektronski obliki.

CIP - Kataložni zapis o publikaciji
Univerzitetna knjižnica Maribor

004.932 (497.4) (082) (0.034.2)

STROKOVNA konferenca ROSUS (14 ; 2019 ; Maribor)
ROSUS 2019 [Elektronski vir] : računalniška obdelava slik
in njena uporaba v Sloveniji 2019 : zbornik 14. strokovne
konference / urednik Božidar Potočnik. - 1. izd. - El. zbornik. -
Maribor : Univerzitetna založba Univerze, 2019

Način dostopa (URL):

<http://press.um.si/index.php/ump/catalog/book/393>

ISBN 978-961-286-244-2 (pdf)

doi: 10.18690/978-961-286-244-2

1. Potočnik, Božidar

COBISS.SI-ID [96256769](#)

ISBN 978-961-286-244-2 (PDF)
978-961-286-245-9 (Broš.)

DOI <https://doi.org/10.18690/978-961-286-244-2>

Cena
Price Brezplačen izvod

Odgovorna oseba založnika red. prof. dr. Zdravko Kačič, rektor Univerze v Mariboru

Prispevki predstavljajo stališča avtorjev, ki niso nujno usklajena s stališči organizatorja, programskega odbora in urednika zbornika, zato ne sprejemajo nobene formalne odgovornosti zaradi morebitnih avtorjevih napak, netočnosti in neustrezne rabe virov. Avtorji prispevkov odgovarjajo za pravilno rabo jezika.

ROSUS 2019 - Računalniška obdelava slik in njena uporaba v Sloveniji 2019

BOŽIDAR POTOČNIK

Povzetek ROSUS 2019 – Računalniška obdelava slik in njena uporaba v Sloveniji 2019 je strokovna računalniška konferenca, ki jo od leta 2006 naprej vsako leto organizira Inštitut za računalništvo iz Fakultete za elektrotehniko, računalništvo in informatiko, Univerze v Mariboru. Konferenca povezuje strokovnjake in raziskovalce s področij digitalne obdelave slik in strojnega vida z uporabniki tega znanja, pri čemer uporabniki prihajajo iz raznovrstnih industrijskih okolij, biomedicine, športa, zabavnštva in sorodnih področij. Zbornik konference ROSUS 2019 združuje 14 strokovnih prispevkov petdesetih avtorjev, od tega vabljeno predavanje ter 4 demonstracijske prispevke. Prispevki podajajo najnovejše dosežke slovenskih strokovnjakov s področij digitalne obdelave slik in strojnega vida, osvetljujejo pa tudi trende in novosti na omenjenih strokovnih področjih. Velik poudarek prispevkov je na promoviranju ekonomske koristnosti aplikacij računalniške obdelave slik in vida v slovenskem prostoru. Takšne računalniške aplikacije zaradi visoke natančnosti, robustnosti in izjemnih hitrosti pri obdelovanju informacij nudijo namreč nove priložnosti za uveljavitev na trgu visokih tehnologij.

Ključne besede: • računalniška obdelava slik • strojni vid • biomedicina • industrijske aplikacije • prenos znanja •

NASLOV UREDNIKA: dr. Božidar Potočnik, izredni profesor, Univerza v Mariboru, Fakulteta za elektrotehniko, računalništvo in informatiko, Koroška cesta 46, 2000 Maribor, Slovenija, e-pošta: bozidar.potocnik@um.si.

DOI <https://doi.org/10.18690/978-961-286-244-2>
Dostopno na: <http://press.um.si>.

ISBN 978-961-286-244-2

ROSUS 2019 - Computer image processing and its application in Slovenia 2019

BOŽIDAR POTOČNIK

Abstract ROSUS 2019 – Computer image processing and its application in Slovenia 2019 is a professional conference that, since 2006, has been organised every year by the Institute of Computer Science of the Faculty of Electrical Engineering and Computer Science, University of Maribor. The conference connects researchers in the fields of Image Processing and Machine Vision with users of this knowledge, whereby users are coming from diverse industrial environments, such as Biomedicine, Sport, Entertainment, and related fields. The proceedings of ROSUS 2019 combines 14 scientific articles by 50 authors, including invited lecture and four demonstration articles. Contributions represent the latest achievements of Slovenian experts in the fields of Image Processing and Vision, and also highlight trends and novelties in these areas. Great emphasis is on promotion of the economic usefulness of Image Processing and Vision applications in the Slovenian region. Namely, such software, due to high precision, robustness, and exceptional speed in information processing, provides new opportunities for penetration on the high technologies market.

Keywords: • Computer Image Processing • Machine Vision • Biomedicine • industrial applications • knowledge transfer •

CORRESPONDENCE ADDRESS: Božidar Potočnik, PhD, Associate Professor, University of Maribor, Faculty of Electrical Engineering and Computer Science, Koroška cesta 46, 2000 Maribor, Slovenia, e-mail: bozidar.potocnik@um.si.

DOI <https://doi.org/10.18690/978-961-286-244-2>
Available at: <http://press.um.si>

ISBN 978-961-286-244-2

Kazalo

VABLJENA PREDAVANJA	1
Uporaba metod računalniškega vida pri delu sodnega izvedenca Borut Batagelj	3
POPOLDANSKA SEKCIJA	9
Globoko učenje: Praktični nasveti strokovnjakov	
Aplikativna uporaba segmentacije osnovane na globokem učenju Danijel Skočaj, Domen Tabernik, Borja Bovcon, Vitjan Zavrtanik in Matej Kristan	11
Praktična uporaba metod umetne inteligence v industrijskem okolju Jure Skvarč in Samo Šela	19
Globoko učenje: Od samovozečih avtomobilov do anotacije rentgenskih slik Gašper Sedej, Martin Šavc in Božidar Potočnik	21
State of the art hyperspectral data analysis and prospects with respect to AI Matthias Kerschhaggl	31
DOPOLDANSKA SEKCIJA	33
Industrijske aplikacije Medicinske in biomedicinske aplikacije Drugo Študentske aplikacije	
Staranje obrazov s pomočjo globokih generativnih nevronskih mrež Nejc Vesel, Peter Peer, Vitomir Štruc in Blaž Meden	35

Detekcija razpok v steklu - klasična metoda proti globoki nevronski mreži	47
Marija Ivanovska, Klemen Grm, Vitomir Štruc in Janez Perš	
Poravnava 3D modelov objektov na realne 3D oblake točk	59
Martin Pernuš, Simon Dobrišek in Janez Perš	
Učenje igranja računalniških iger z globoko obdelavo slik in drevesnim preiskovanjem Monte Carlo	69
Matej Vitek in Peter Peer	
Contactless fingerprint identification using mobile phone camera	81
Luka Loboda, Matej Vitek, Blaž Meden, Peter Peer & Žiga Emeršič	
Optično branje podatkov iz računa	91
Aleš Horvat in Borut Batagelj	
Subdivided Ear Recognition	99
David Romero, Matej Vitek, Blaž Meden, Peter Peer & Žiga Emeršič	
Comparing performance of biometric models trained on different groups with Bayesian statistics	109
Alen Ajanović, Peter Peer & Žiga Emeršič	
Spletni sistem in javna podatkovna baza USOVA3D za nepristrano validiranje postopkov detektiranja jajčnih mešičkov na 3D ultrazvočnih posnetkih jajčnikov	119
Jurij Munda, Božidar Potočnik, Milan Reljič, Ksenija Rakić, Jure Knez, Veljko Vlasisavljević, Gašper Sedej, Boris Cigale, Aleš Holobar in Damjan Zazula	

Spoštovani!

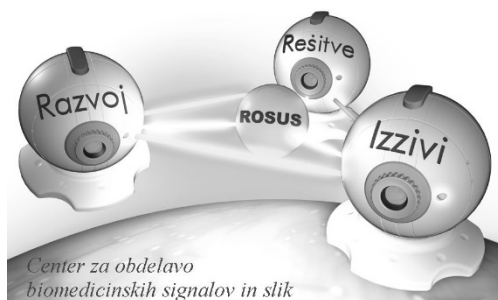
Po trinajstih konferencah ROSUS 2006–2018 želimo tudi s konferenco ROSUS 2019 nadaljevati s promoviranjem pomembnosti ekonomske koristi računalniške obdelave slik na področjih industrije, biomedicine in drugih poslovnih procesov. Vezi, ki smo jih na prejšnjih konferencah stkali med raziskovalci, razvijalci, ponudniki rešitev ter uporabniki računalniške obdelave slik v slovenskem prostoru, želimo še dodatno okrepiti, ob tem pa nuditi tudi možnosti sklepanja novih sodelovanj in svetovanja pri razreševanju konkretnih poslovnih oziroma raziskovalnih problemov.

Tudi letos namenjamo glavni poudarek aplikacijam s področja računalniške obdelave slik, ki so že integrirane oziroma pripravljene za integracijo v poslovne procese. Novost letošnje konference je popoldanska sekcija z naslovom »Globoko učenje: Praktični nasveti strokovnjakov«, ki smo jo organizirali v sodelovanju s podjetjem Kolektor. Demonstrirali bomo, da avtomatska obdelava v industriji lahko zaradi svoje natančnosti in hitrosti prinaša velike ekonomske koristi, hkrati pa nakazali, da aplikacije računalniške obdelave slik nudijo nove priložnosti za uveljavitev na trgu visokih tehnologij. Seveda ne smemo pozabiti na možnost diskusije ter predstavitev konkretnih problemov in potreb, ki se porajajo pri uporabnikih, s katerimi bomo računalniško obdelavo slik in njeno koristnost še bolj približali avditoriju.

Naj sklenemo uvodne misli še s prisrčno zahvalo Javnemu skladu Republike Slovenije za podjetništvo, ki je v okviru konference ROSUS 2019 predstavil

zanimive finančne instrumente za spodbujanje prenosa tehnoloških rešitev v podjetniško sfero. Izpostaviti želimo še medijska pokrovitelja revijo IRT3000 in Računalniške novice, ki sta intenzivno promovirala konferenco ROSUS 2019 ter pomen strojnega vida v slovenskem prostoru.

Božidar Potočnik
predsednik konference
ROSUS 2019



Programski odbor ROSUS 2019:

dr. Aleš Holobar
dr. Damjan Zazula
dr. Boris Cigale
dr. Marjan Mernik
dr. Peter Peer
dr. Tomaž Tomažič
dr. Matjaž Colnarič
dr. Slobodan Ribarič
dr. Vojko Flis
dr. Saša Divjak

dr. Stanislav
Kovačič
dr. Zdravko Kačič
dr. Aleš Leonardis
dr. Simon Dobrišek
dr. Mitja Lenič
dr. Franc Solina
dr. Andrej Šoštarič
dr. Franjo Pernuš
dr. Jure Skvarč
dr. Božidar
Potočnik

Organizacijski odbor 2019:

Gašper Sedej
Jurij Munda
dr. Uroš Mlakar
dr. Matjaž Divjak
Martin Šavc
Jernej Kranjec
dr. Danilo Koržec
dr. Božidar Potočnik

POKROVITELJI



ROSUS 2019
<http://rosus.feri.um.si>

VABLJENA PREDAVANJA

Uporaba metod računalniškega vida pri delu sodnega izvedenca

BORUT BATAGELJ

Povzetek Ko sodišče presodi, da mu je pri njihovi presoji potrebna pomoč strokovnjaka zaprosi sodnega izvedenca za izdelavo mnenja glede strokovnih vprašanj. Predavatelj bo predstavil težave in izzive s katerimi se kot sodni izvedenec srečuje pri svojem delu na področju biometričnih prepoznav obrazov in avtentičnosti slik. Preden lahko na sliki iz nadzorne kamere uporabimo sistem za samodejno prepoznavo obrazov, jo moramo z različnimi metodami za obdelavo slik izboljšati in popraviti. Z metodo računalniškega vida iz slik rekonstruiramo 3D informacije. Včasih je treba potrditi tudi pristnost videomateriala, za kar lahko uporabimo različne tehnike na podlagi poznavanja formata zapisa in delovanja algoritmov za obdelavo slike in videa. Če identifikacija na osnovi obraza ni mogoča ali je neuspešna, lahko uporabimo tudi druge biometrične karakteristike osebe – mehke biometrike. Sodni izvedenec za slikovno biometrijo mora razumeti in znati uporabiti različne metode za obdelavo slik in različna orodja računalniškega vida. Sposoben mora biti tudi razložiti svoje ugotovitve in oceniti verjetnost končnega rezultata.

Ključne besede: • forenzični ekspert • analiza fotografij • avtentičnost posnetkov • identifikacija oseb • nadzorni sistem •

NASLOV AVTORJA: dr. Borut Batagelj, višji predavatelj, Univerza v Ljubljani, Fakulteta za računalništvo in informatiko, Večna pot 113, 1000 Ljubljana, Slovenija, e-pošta: borut.batagelj@fri.uni-lj.si.

DOI <https://doi.org/10.18690/978-961-286-244-2.1>
Dostopno na: <http://press.um.si>.

ISBN 978-961-286-244-2

1 UVOD

Obdani smo z vizualnimi podobami, ampak za razliko od nekoč, ko smo novico še dodatno podkrepili s sliko ali posnetkom dogodka pa je danes ravno nasprotno. Zaupanje v pristnost video in slikovnega materiala se je močno zmanjšalo. Živimo namreč v svetu lažnih novic. Lažnih novic so poslužujejo različni mediji od rumenega tiska do znanstvenih revij. Nameni so različni, od politične propagande do ponarejanja dokazov na sodišču [1]. Če še nismo zasledili ponarejene slike smo zagotovo že kdaj prejeli kakšno lažno elektronsko sporočilo.

Ponarejanje je dandanes zelo enostavno. Razširjenost naprav za zajem slik in videa, dostopnost do programske opreme za manipulacijo, vse to nam omogoča, da lahko ponaredimo različen multimedijijski material [2, 3]. Naloga digitalnih forenzikov je nazaj povrniti zaupanje.

2 MANIPULACIJA SLIK

Metode za odkrivanje ponaredkov delimo v dve skupini: aktivne metode in pasivne metode. Za razliko od aktivnih metod, ki od nas zahtevajo posebno strojno ali programsko opremo za dodajanje ustreznih oznak v sliko, ali naj bo to vodni žig ali digitalni podpis, pa pasivne metode predpostavljajo, da se je, kljub temu, da ni vidnih sprememb na sliki spremenila statistika slike. Spremembo statistike lahko odkrivamo s pomočjo različnih metod. Pasivne metode lahko delujejo na nivoju slikovnega elementa, na nivoju formata, izkoriščajo lastnosti kamere, fizikalne lastnosti ali geometrijske lastnosti [4].

Pri metodah na osnovi slikovnega elementa je glavni element preiskave slikovni element. Z različnimi tehnikami lahko odkrijemo ali je bil kakšen del slike kloniran ali ponovno vzorčen. Fourierjeve statistike višjega reda se spremenijo, če kombiniramo več slik v novo sliko. Statistike se spremenijo tudi v primeru povečave ali filtriranja slike. S pomočjo statistike lahko odkrijemo tudi ali gre za pravo fotografijo ali računalniško generirano. V sliki lahko tudi odkrijemo skrita sporočila, tako imenovano steganografijo. V veliko pomoč pri odkrivanju sprememb na sliki nam je tudi izgubni format jpeg. S pomočjo kvantizacijske tabele lahko ugotovimo napravo s katero je bila slika zajeta. Na osnovi formata lahko tudi ugotovimo, če slika vsebuje različno kompresijo. Ker algoritem za

stiskanje večinoma deluje na blokkih velikosti 8x8 lahko na robovih blokov opazimo, če je bila slika manipulirana. Vsaka naprava pa s pomočjo svojih algoritmov za stiskanje in lastnosti leč, ki jih uporablja pusti na sliki določeno sled, ki nam pomaga odkriti izvor naprave ali potrditi ali je bila slika res zajeta z določeno napravo ali mogoče kasneje kaj spremenjena. Tudi fizikalne lastnosti odboja svetlobe in senčenja nam pomagajo odkriti spremembe kompozicije na sliki ali dele, ki so bili izrezani iz drugih slik posnetimi pod drugačnimi pogoji.

Ko imamo dovolj dobre posnetke pa lahko s pomočjo fotogrametrije izmerimo določene razdalje med predmeti ali značilkami na sliki in s tem potrdimo avtentičnost ali identiteto [3]. Različne transformacije ali slikovni popravki (posvetlitev, izravnava histograma, povečava kontrasta, odstranitev šuma, odstranitev zamegljenosti, stabilizacija videa, popravek premika, korekcija leč) nam pomagajo posnetek izboljšati, da postanejo posamezne značilke vidnejše (slika 1).



Slika 1: Majica obtoženega, ki je bila zasežena takoj po dejanju (levo), in slika majica osebe iz posnetka nazdorne kamere, ki je bila ustrezno izboljšana (izostritev, posvetlitev, izravnava histograma).

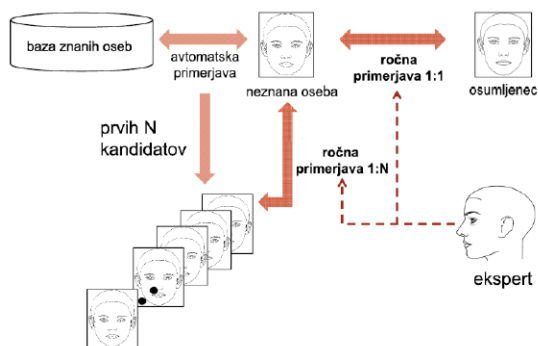
Zelo pomembno je, da vsak korak ki ga naredimo dokumentiramo in da vedno delamo izboljšave na kopiji čim bolj originalnega posnetka ali slike.

2 IDENTIFIKACIJA OSEB

Drugi del nalog s katerimi se srečuje forenzični sodni izvedenec za slikovni material pa je identifikacija osebe na posnetku. Če je posnetek dovolj kakovosten se osredotočimo na obrazne značilnosti. Pri identifikaciji si lahko pomagamo tudi z najnovejšimi programi za identifikacijo [6], vendar moramo na koncu sami presoditi in utemeljiti katere značilnice obraza se ujemajo, oziroma se ne ujemajo, da osebo potrdimo ali izločimo (slika 2). Pri primerjavi

obraznih značilnosti moramo vzeti v obzir kvaliteto posnetkov, časovno komponento med zajetimi posnetki in pregledati morebitna kožna znamenja ali odrgnine.

Računalniški sistemi za identifikacijo so nam v veliko pomoč tudi pri iskanju potencialnih osumljencev. Poleg opisnih značilnosti lahko iščemo tudi po skici osebe oziroma fotorobotu ali sliki iz nadzorne kamere. Med izbranimi osebami nato lažje poiščemo najbolj verjetno osebo (slika 2).



Slika 2: Standardni postopek prepoznavne neznane osebe, kjer je ključno da sodeluje ekspert za prepoznavo.

Velikokrat pa obraz osebe na posnetku ni viden ali je preslabe kvalitete (osvetlitev, različen kot, oddaljenost). V tem primeru se moramo posluževati tudi drugih značilnosti osebe, kot je hoja, postava, višina, oblika glave, oblika uhljev (slika 3) [7].



Slika 3: Primera, kjer obrazne značilnosti niso vidne ali so zakrite: ocena velikosti osebe (levo), in viden uhlj (desno).

3 ZAKLJUČEK

Zelo pomembno je, da sodni izvedenec pri svojem delu zna pravilno uporabiti najnovejšo tehnologijo računalniškega vida za odkrivanje značilnosti slike in da zna pravilno predstaviti rezultate, ki nam jih vrnejo različne metode za preverjanje avtentičnosti. Posnetek, ki je slabe kvalitete ali bil večkrat shranjen nas lahko zavede, da je bil spremenjen, čeprav temu ni vedno tako. Tudi obrazi pod različnimi koti in osvetlitvami ter slabe kvalitete zaradi stiskanja lahko izgledajo popolnoma drugačni. Znati moramo presoditi ali so vidne značilnosti dovolj kvalitetne in dovolj redke, da lahko z določeno gotovostjo trdimo, da gre za isto osebo/predmet ali da lahko osebo/predmet izločimo.

Literatura

1. *Photo tempering through history*; Pridobljeno na <http://pth.izitru.com/>.
2. Jin, Zeyu, et al. (2017), VoCo: text-based insertion and replacement in audio narration, *ACM Transactions on Graphics (TOG)*, vol. 36(4), str. 96.
3. Thies, Justus, et al. (2016), Face2face: Real-time face capture and reenactment of rgb videos, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.
4. Farid, H. (2009). A survey of image forgery detection. *IEEE Signal, Processing Magazine*, vol. 26(2), str. 16–25.
5. Criminisi, A., Reid, I. in Zisserman, A. (2000). Single view metrology, *International Journal of Computer Vision*, vol. 40(2), str. 123–148.
6. Phillips, P. Jonathon, et al. (2018), Face recognition accuracy of forensic examiners, superrecognizers, and face recognition algorithms, *Proceedings of the National Academy of Sciences*.
7. Žiga Emeršič, Blaž Meden, Vitomir Štruc, Peter Peer (2018), Evaluation and Analysis of Ear Recognition Models: Performance, Complexity and Resource Requirements, *Neural Computing & Applications*.

POPOLDANSKA SEKCIJA

Globoko učenje: Praktični nasveti strokovnjakov

Aplikativna uporaba segmentacije osnovane na globokem učenju

DANIJEL SKOČAJ, DOMEN TABERNIK, BORJA BOVCON, VITJAN
ZAVRTANIK IN MATEJ KRISTAN

Povzetek V zadnjih letih smo priča zelo velikemu napredku na področju računalniškega vida, ki predvsem na krilih globokega učenja dosegajo odlične rezultate in v obliki raznih aplikacij in robotskih sistemov vstopa na najrazličnejša področja našega življenja. V tem članku bomo osvetlili te zadnje trende in predstavili zmogljivosti globokega učenja na primerih praktičnih aplikacij. Predvsem se bomo osredotočili na probleme, ki so rešljivi z uporabo semantične segmentacije oz. njenih nadgradenj. Predstavili bomo primere detekcije anomalij na industrijskih (pol)izdelkih, detekcije ovir na vodni površini pred avtonomnim plovilom ter štetja predmetov oz. polipov, ki svojo funkcionalnost gradijo na segmentacijskih konvolucijskih globokih nevronske mrežah. Metodologija, osnovana na globokem učenju, omogoča zamenjavo paradigme razvoja aplikacij računalniškega vida; namesto klasičnega inženirskega pristopa s snovanjem namenskih značilnic in algoritmov se vse bolj uveljavlja pristop, ki temelji na podatkih in strojnem učenju. Tak pristop vse bolj prodira tudi na področje strojnega vida.

Ključne besede: • računalniški vid • strojni vid • segmentacija • globoko učenje • detekcija vizualnih napak •

NASLOVI AVTORJEV: dr. Danijel Skočaj, izredni profesor, Univerza v Ljubljani, Fakulteta za računalništvo in informatiko, Večna pot 113, 1000 Ljubljana, Slovenija, e-pošta: danijel.skocaj@fri.uni-lj.si. Domen Tabernik, raziskovalec, Univerza v Ljubljani, Fakulteta za računalništvo in informatiko, Večna pot 113, 1000 Ljubljana, Slovenija, e-pošta: domen.tabernik@fri.uni-lj.si. Borja Bovcon, raziskovalec, Univerza v Ljubljani, Fakulteta za računalništvo in informatiko, Večna pot 113, 1000 Ljubljana, Slovenija, e-pošta: borja.bovcon@fri.uni-lj.si. Vitjan Zavrtnik, Univerza v Ljubljani, Fakulteta za računalništvo in informatiko, Večna pot 113, 1000 Ljubljana, Slovenija, e-pošta: vitjan.zavrtanik@fri.uni-lj.si. dr. Matej Kristan, izredni profesor, Univerza v Ljubljani, Fakulteta za računalništvo in informatiko, Večna pot 113, 1000 Ljubljana, Slovenija, e-pošta: matej.kristan@fri.uni-lj.si.

1 UVOD

Področje računalniškega vida je v zadnjih nekaj letih zelo napredovalo. Napredek gre pripisati predvsem preboju na področju globokega učenja in uporabi tega načina modeliranja znanja v računalniškem vidu. Povečalo se je število učnih slik, ki so na voljo za učenje takšnih algoritmov, ravno tako se je povečala zmogljivost grafičnih procesnih enot, ki omogočajo učinkovito učenje globokih nevronske mreže. Seveda pa se vse bolj izboljšujejo tudi algoritmi za globoko učenje, predvsem pa se pojavljajo nove izvedenke za reševanje najrazličnejših problemov, od detekcije in klasifikacije predmetov do segmentacije, izboljševanja in barvanja slik ter video posnetkov in sorodnih aplikacij.

Nekateri problemi so prezahtevni za eksplicitno načrtovanje značilnic in algoritmov, kot je to običaj v klasičnem pristopu, so pa rešljivi s pomočjo učenja, oz. implicitnega podajanja znanja preko učnih slik. Tak način reševanja problemov je pogosto tudi bolj enostaven, saj ne zahteva specifičnega dodatnega inženirskega dela, marveč le zadostno število (označenih) učnih slik. Problem se torej definira z množico učnih slik; učni algoritem potem sam poskrbi za izločanje relevantnih značilnic in uporabi le-teh za izvajanje zadane naloge. Reševanje problema se tako prevede na snovanje ustreznega učnega algoritma in zagotavljanje primernih učnih slik. Učni algoritmi so veliko bolj ponovno uporabljivi kot rešitve v klasičnih pristopih strojnega vida. Glavni koncepti so precej splošni in se s prilagoditvami uporabljajo za reševanje najrazličnejših problemov računalniškega vida.

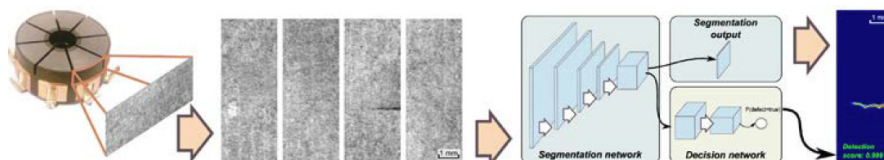
V tem članku se bomo osredotočili na specifičen tip algoritmov, ki temeljijo na semantični segmentaciji in polno-konvolucijskem pristopu h konvolucijskim nevronske mrežam. Pokazali bomo kako lahko podoben princip uporabimo za reševanje treh različnih problemov. Najprej bomo prikazali uporabo tega principa za segmentacijo in detekcijo napak na površinah industrijskih (pol)izdelkov z namenom vizualnega nadzora kvalitete. Nato bomo podoben princip preverili na primeru segmentacije vodne površine in detekcije ovir pred avtonomnim plovilom. Na koncu bomo ta princip preverili še na povsem drugi nalogi in sicer za štetje predmetov, oz. konkretnje štetje polipov na podvodnih slikah.

2 DETEKCIJA POVRŠINSKIH NAPAK

Problem detekcije površinskih napak smo se odločili nasloviti kot problem binarne klasifikacije slik, ki smo ga reševali z metodami globokega učenja. V ta namen smo razvili novo arhitekturo globoke mreže, ki bazira na dvostopenjski arhitekturi, kot je prikazano na Sliki 1.

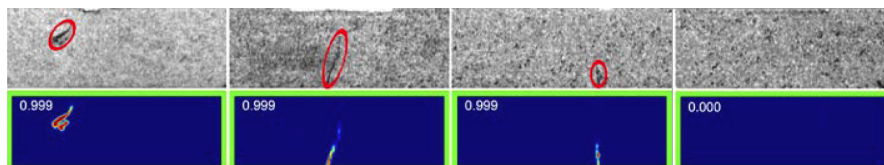
Na prvi stopnji predlagane arhitekture se izvede groba lokalizacija površinske napake oz. segmentacija napake. Segmentacijska mreža [1] se z vzratnim razširjanjem napake in stohastičnim gradientnim spustom nauči segmentirati sliko. Kot vhod v učni algoritem podamo poleg slike tudi segmentacijsko masko z označenimi napakami. Cilj učenja segmentacijske mreže je minimizirati razliko med rezultati segmentacije in podanimi segmentacijskimi maskami; pri tem za kriterijsko funkcijo uporabljamo navzkrižno entropijo, pri čemer se vsak slikovni element smatra kot učni primer. Na ta način smo praktično povečali število učnih primerov in smo se izognili problemu prevelikega prilagajanja učni množici, ko je na voljo majhno število učnih primerov.

Druga stopnja predlagane arhitekture nato poskrbi za binarno klasifikacijo slike oz. za določitev ali slika vsebuje napako ali ne. Le-to smo implementirali z dodatno mrežo, ki za vhodne značilnice vzame rezultate segmentacijske mreže. Uporabili smo tako segmentacijski izhod kot tudi značilnice segmentacijske mreže s predzadnjega nivoja



Slika 1: Postopek detekcije površinskih napak: industrijski polizdelek, slike zgornjega dela površine kot vhod v nevronske mreže, segmentacijski in klasifikacijski del nevronske mreže, končni rezultat: segmentacija z lokalizirano napako in rezultat klasifikacije.

Nekaj tipičnih rezultatov, dobljenih na primeru detekcije napak na površini električnega komutatorja, je predstavljenih na Sliki 2. Na njej so prikazane slike z obkroženimi napakami ter rezultati segmentacije in klasifikacije. Vidimo, da je nevronska mreža zelo uspešno segmentirala napako ter tudi z veliko gotovostjo klasificirala defektne primere kot slike z napako (vrednost klasifikacije blizu 1), zadnji primer pa kot sliko brez napake (vrednost klasifikacije v razred z napako je 0). Uspešnost predlaganega pristopa so potrdili tudi kvantitativni rezultati. Predlagana mreža je na množici 400 primerov dosegla povprečno natančnost 99,9%; ni ji uspelo detektirati zgolj enega primera z napako, pri čemer ni niti enega primera brez napake zmotno klasificirala kot defektnega. Metodo smo na tej domeni primerjali tako s sorodnimi sodobnimi metodami za semantično segmentacijo [2,3] kot tudi s komercialnim orodjem za detekcijo površinskih napak; naša metoda je izmed vseh evalviranih pristopov dosegla najboljše rezultate.

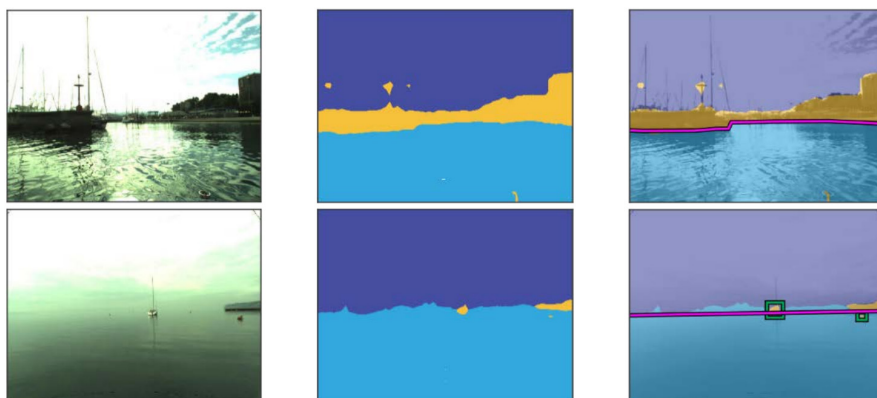


Slika 2: Nekaj primerov klasifikacije: zgoraj vhodne slike z obkroženimi napakami, spodaj rezultat segmentacije in klasifikacije.

3 DETEKCIJA OVIR

Druga preizkušena aplikativna domena se uvršča na področje pomorske in terenske robotike na katerem je prišlo v zadnjem desetletju do vzpostavitve novega razreda plovil brez posadke (USV). Takšna plovila so običajno velikosti približno 2 metra, njihova glavna prednost pa je prenosljivost in zmožnost delovanja v plitvih vodah. Področje uporabe sega od nadzora priobalnega pasu do nadzora težko dosegljivih področij, ki predstavljajo nevarnost človeku (npr. jezovi). Te naloge zahtevajo visoko stopnjo avtonomije, ki je v prvi vrsti odvisna od pravočasnega odkrivanja ter izogibanja bližnjim oviram in plavajočim odpadkom.

Cenovno dostopni in informacijsko bogati senzorji, kot so kamere, opremljene z metodami računalniškega vida, postajajo vodilni mehanizem za razpoznavanje okoliša. S pomočjo postopka semantične segmentacije lahko vsak slikovni element pripišemo eni izmed treh komponent - nebo, voda ali kopno/ovira. Slikovni elementi vode, ki tvorijo povezano regijo, predstavlja plovno površino. Elementi komponente z oznako »ovira«, ki se pojavijo znotraj ovojnice regije vode, pa predstavljajo potencialne ovire. Razdaljo do posamezne ovire lahko izračunamo z uporabo stereo sistema kamer in triangulacije. Izvedli smo obsežno evalvacijo segmentacijskih metod, ki smo jih aplicirali na domeni detekcije ovir na vodni površini [4]. Na Sliki 3 so prikazani rezultati semantične segmentacije metode DeepLab-v2 [2], skupaj z odkritimi ovirami po naknadni obdelavi rezultatov segmentacije.



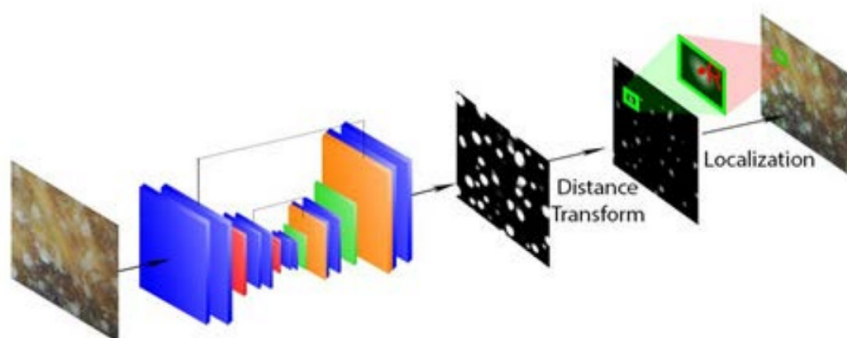
Slika 3: Vhodna slika (levo), rezultati segmentacije (sredina), ter odkrite ovire (desno). Komponente neba, ovir in vode so označene s temno modro, rumeno in svetlo modro barvo, medtem, ko je zlati standard roba morja označen z vijolično barvo ter pravilno odkrite ovire z zeleno barvo.

4 ŠTETJE POLIPOV

Zadnji problem, ki ga predstavljamo, je detekcija polipov meduz na podvodnih slikah. To je težek problem zaradi velike količine objektov, ki jih je potrebno detektirati, problem pa dodatno otežujejo še negativne vizualne lastnosti polipov in ozadja, saj se polipi med seboj močno razlikujejo, ozadje ni

uniformno in vsebuje dodatne objekte, poleg tega pa se polipi pogosto medsebojno prekrivajo.

Za naslavljanje problema detekcije polipov meduz na slikah ostrig smo uporabili detekcijo z interpretacijo segmentacijske maske. Predlagali smo dvostopenjsko metodo, orisano na Sliki 4, ki polipe najprej segmentira od ozadja z uporabo konvolucijske nevronske mreže, nato pa z metodo interpretacije generirane segmentacijske maske polipe natančno lokalizira. Izhodno segmentacijsko masko metoda najprej binarizira z uporabo fiksne mejne vrednosti, nato pa polipe lokalizira na lokacijah lokalnih maksimumov maske oddaljenosti, na kateri je vrednost posameznega slikovnega elementa enaka razdalji tega elementa do najbližjega slikovnega elementa, ki v binarni maski pripada ozadju. Interpretacijska metoda izkorišča cirkularno simetrijo polipov za njihovo natančno lokalizacijo saj je lokacija lokalnega maksimuma maske oddaljenosti enaka lokaciji centra polipa le, če drži predpostavka o cirkularni simetričnosti polipov. Postopek generiranja detekcij z interpretacijo maske je učinkovit in invarianten na velikost detektiranih polipov, poleg tega pa je robusten na njihovo delno prekrivanje. Za učenje segmentacijske mreže potrebujemo učne segmentacijske maske, vendar pa bi bila ročno označevanje segmentacijskih mask na slikovni element natančno zaradi svoje zamudnosti v primeru polipov težko izvedljivo, saj bi moral označevalec na vsaki sliki natančno označiti približno 1000 polipov. Zaradi višje hitrosti ročnega označevanja pravokotnih regij, v primerjavi z ročnim označevanjem na slikovni element natančno, smo mrežo učili na segmentacijskih maskah generiranih iz pravokotnih označb polipov. Učno segmentacijsko masko smo generirali tako, da smo uporabili predpostavko, da so polipi približno cirkularno simetrični. Preverili smo tudi vpliv zahtevnosti segmentacijske mreže na uspešnosti detekcije in ugotovili, da lahko na danem problemu dosežemo dobre rezultate tudi z uporabo mreže, ki uporablja manjše število učljivih filtrov.



Slika 4: Metoda je sestavljena iz segmentacijske konvolucijske nevronske mreže in metode za lokalizacijo polipov z interpretacijo izhodne segmentacijske maske.

Metoda dosega odlične rezultate na testni množici in presega rezultate drugih modernih metod za detekcijo polipov oz. objektov [5,6]. Ročno označevanje polipov od označevalcev zahteva dolgotrajno zbranost in natančnost. Zaradi dolgotrajnosti postopka se pri ročnem označevanju pojavljajo napake, saj označevalec pogosto ne opazi in posledično ne označi dela polipov. Označevalec v povprečju doseže 90% priklic in skoraj 100% natančnost [5]. Naša metoda je po konsistentnosti detekcij podobna človeku, saj v povprečju doseže 94% priklic in 95% natančnost.

5 ZAKLJUČEK

V tem članku smo predstavili uporabnost globokega učenja, ki temelji na semantični segmentaciji, za reševanje različnih problemov računalniškega vida. Orisali smo tri metodološko sorodne rešitve, ki smo jih razvili za tri različne praktične probleme: detekcijo vizualnih napak na površinah v proizvodnji, detekcijo ovir na vodni gladini pred avtonomnim plovilom, ter štetje polipov. Rezultati potrjujejo uspešnost predlaganih rešitev. Rezultati tudi nakazujejo, da se lahko velika množica problemov rešuje na sorodne načine, pri čemer veliko vlogo pri uspešnosti delovanja igra kvaliteta učnih primerov.

Trend uporabe diskriminativnega nadzorovanega globokega učenja je na področju računalniškega vida popolnoma prevladal. V svoji osnovi omogoča zamenjavo klasične paradigme razvoja aplikacij računalniškega vida, ki temelji

na ročnem snovanju namenskih značilnic in algoritmov, z bolj splošnim in poenotenim pristopom, ki sloni na učnih slikah in globokem učenju. Tak pristop vse bolj prodira tudi na področje strojnega vida in v prihodnjih letih se bo uporaba tega principa v industrijskih aplikacijah samo še povečevala. Lahko pa pričakujemo porast zanimanja za delno-nadzorovane in nenadzorovane pristope h globokemu učenju, ki bodo poenostavili proces priprave učnih podatkov in bodo omogočili uporabo teh pristopov tudi v primerih, ko je zbiranje in označevanje specifičnih učnih primerov zelo težavno.

Litertura

1. D. Tabernik, D. Skočaj (2017), Detekcija napak na površinah z uporabo anotiranih slik in globokim učenjem, *Zbornik šestindvajsete mednarodne Elektrotehniške in računalniške konference ERK 2017*, Portorož, Slovenija, 25. - 26. september 2017.
2. L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, in A. L. Yuille (2018), DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs, *IEEE TPAMI*, 40(4):834-848.
3. O. Ronneberger, P. Fischer, and T. Brox (2015), UNet: Convolutional Networks for Biomedical Image Segmentation. In *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pages 234–241.
4. B. Bovcon, M. Kristan (2019), Benchmarking Semantic Segmentation Methods for Obstacle Detection on a Marine Environment, *Computer Vision Winter Workshop CVWW 2019*, Stift Vorau, Avstrija, 6. – 8. 2. 2019.
5. M. Vodopivec, R. Mandeljc, T. Makovec, A. Malej, M. Kristan (2018), Towards automated scyphistoma census in underwater imagery: useful research and monitoring tool, *Journal of Sea Research*, 142, pp. 147-156.
6. T.-Y. Lin, P. Goyal, R. Girshick, K. He, P. Dollár (2017), Focal loss for dense object detection, *International Conference on Computer Vision ICCV 2017*, pp 2999-3007.

Praktična uporaba metod umetne inteligence v industrijskem okolju

JURE SKVARČ IN SAMO ŠELA

Povzetek Umetna inteligenca (UI) je področje informatike z interdisciplinarnim značajem. Cilj UI je razvoj naprav, ki se vedejo, kot da bi razpolagale z inteligenco. S postopki strojnega in globokega učenja računalnike naučimo, da so sposobni učinkovito opravljati naloge v ozki domeni. S tehnologijo vizualne inteligence lahko izjemno nadgradimo zmogljivosti sistemov strojnega vida v industriji. Lahko rešujemo do sedaj nerešene probleme in odpiramo nove priložnosti za uporabo. V prispevku bodo predstavljeni konkretni primeri iz prakse, ki bodo pokazali, da UI spreminja paradigmo reševanja izzivov na področju strojnega vida v industrijskem okolju.

Ključne besede: • umetna inteligenca • industrija • globoko učenje • strojni vid • reševanje izzivov •

NASLOVA AVTORJEV: dr. Jure Skvarč, Kolektor Vision, Cesta v Gorice 34, 1000 Ljubljana, Slovenija, e-pošta: jure.skvarc@kolektor.com. Samo Šela, Kolektor Vision, Cesta v Gorice 34, 1000 Ljubljana, Slovenija, e-pošta: samo.sela@kolektor.com.

DOI <https://doi.org/10.18690/978-961-286-244-2.3>
Dostopno na: <http://press.um.si>.

ISBN 978-961-286-244-2

Globoko učenje: od samovozečih avtomobilov do anotacije rentgenskih slik

GAŠPER SEDEJ, MARTIN ŠAVC IN BOŽIDAR POTOČNIK

Povzetek Članek opisuje postopek predelave nevronske mreže namenjene avtonomni vožnji v nevronske mreže, ki je sposobna anotirati karakteristične točke na biomedicinskih slikah. Kot osnovo smo uporabili arhitekturo za segmentacijo slik imenovano Segnet. Mrežo smo priredili za branje rentgenskih slik in detekcije karakterističnih točk ter jo poimenovali KeypointNet. Preizkus smo opravili na bazi 124 kefalometričnih rentgenskih slik. Na slikah smo označili 10 kefalometričnih točk. Mrežo smo učili na učni množici velikosti 100 slik in testirali na testni množici velikosti 24 slik. Rezultati meritev na testni množici so pokazali povprečno napako 2,7 piksla pri velikosti slike 480×360 .

Ključne besede: • globoko učenje • konvolucijske nevronske mreže • kefalometrija • razvrščanje vzorcev • oslonilne točke •

NASLOVI AVTORJEV: Gašper Sedej, tehniški sodelavec, Univerza v Mariboru, Fakulteta za elektrotehniko, računalništvo in informatiko, Koroška cesta 46, 2000 Maribor, Slovenija, e-pošta: gasper.sedej@um.si. Martin Šavc, asistent, Univerza v Mariboru, Fakulteta za elektrotehniko, računalništvo in informatiko, Koroška cesta 46, 2000 Maribor, Slovenija, e-pošta: martin.savc@um.si. dr. Božidar Potočnik, izredni profesor, Univerza v Mariboru, Fakulteta za elektrotehniko, računalništvo in informatiko, Koroška cesta 46, 2000 Maribor, Slovenija, e-pošta: bozidar.potocnik@um.si.

1 UVOD

Globoko učenje in konvolucijske nevronske mreže dosegajo v zadnjem času izjemen razcvet. Napredek in podpora ustrezne strojne opreme v obliki grafičnih procesnih enot je omogočila razvoj praktičnih aplikacij, predvsem na področju samovozečih avtomobilov. Prav to področje se zelo naslanja na računalniški vid, podprt z globokim učenjem. Ena izmed ključnih nalog na tem področju je segmentacija slik, tj. razčlenitev slike v ustrezne regije oz. razrede. V tem članku si bomo ogledali postopek, kako tehnologijo na osnovi konvolucijskih nevronske mreže, prvenstveno namenjeno samovozečim avtomobilom, prirediti za biomedicinske raziskave in aplikacije. Slednje bomo demonstrirali na konkretnem problemu zaznave karakterističnih kefalometričnih točk na rentgenski sliki profila obraza.

2 KEFALOMETRIJA

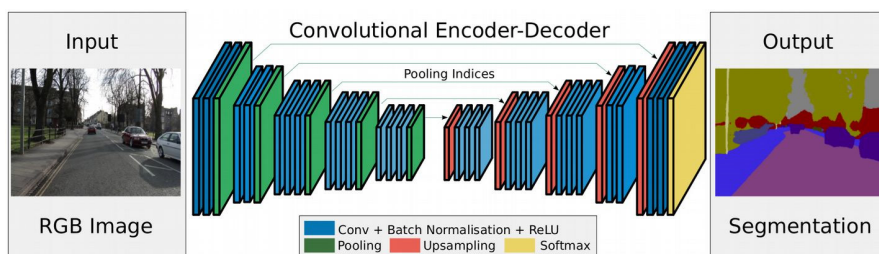
Kefalometrija (*angl. cephalometry*) je veda, ki se ukvarja z merjenjem človeške glave, običajno na radiogramskih (rentgenskih) slikah. Najpogosteje se jo uporablja pri dentalni medicini in ustni kirurgiji za, na primer, merjenje razdalj med zobmi in dlesnimi ter za načrtovanje posegov. Kefalometrične točke (*angl. cephalometric landmarks*) so točke, ki se uporabljajo pri kefalometrični analizi in se jih določi na podlagi rentgenskih slik. Točke se nahajajo na različnih delih kosti in mehkega tkiva, najpogosteje tam, kjer se stikajo črte, robovi ali kotički ploskev [1].

Naravna pozicija glave (*angl. natural head position*) [2] je pozicija glave, ki omogoča ponovljivost pri kefalometričnem zajemu slike. Pacient ima glavo postavljeno tako, da je ploskev obraza poravnana navpično, oči pa gledajo naprej. Ta standard, ki se uporablja še danes, omogoča primerjavo različnih analiz [1].

Lateralni kefalogram je rentgenska slika glave "s strani". Radiogram je postavljen na eni strani glave, film pa na drugi strani. Druga pozicija je posteroanteriorni kefalogram. Film se nahaja za glavo, radiogram pa pred obrazom. V naši raziskavi uporabljamo slike lateralnega kefalograma.

3 ARHITEKTURA SEGNET

Našo rešitev za detektiranje karakterističnih kefalometričnih točk na stranskih rentgenskih slikah glave pacienta smo zasnovali na arhitekturi SegNet, ki jo v tem poglavju pregledno predstavimo. SegNet [3] je arhitektura globoke konvolucijske nevronske mreže za ogrodje Caffe, namenjena segmentaciji slike na osnovi posameznih pikselov (*angl. pixel-wise segmentation*). Je globoka, popolnoma konvolucijska (*angl. fully convolutional*) nevronska mreža, razdeljena v dva dela: kodirnik (*angl. encoder*) in dekodirnik (*angl. decoder*).



Slika 1: Shema arhitekture SegNet (vir: [3])

Na sliki 1 je predstavljena arhitektura »segnet-basic«, ki ima štiri nivoje globine. Leva polovica, kodirnik, izdvaja nizko ločljivostni zemljevid značilnic (*angl. low resolution feature map*). Ta del je sestavljen iz slojev konvolucije, ReLU in združevanja. Desni del predstavlja dekodirnik, ki preslika nizko ločljivostne značilnice v značilnice izvirne ločljivosti. Te značilnice predstavljajo labele (razrede) segmentov za posamezni piksel. V tem delu so uporabljeni naslednji sloji: nadvzorčenje (*angl. upsample*), konvolucijski sloj in sloj Softmax.

Avtorji članka [3] so mrežo preizkusili na bazi [4], ki vsebuje slike zajete s kamero, nameščeno na armaturni plošči avtomobila. Uporabili so lastne anotacije segmentov. Na osnovi rezultatov razberemo, da je arhitektura SegNet primernejša od alternativnih pristopov, zajetih v primerjavo (npr. metoda DeepLab-LargeFOV in metoda Super Passing). Avtorji zaključijo prispevek z ugotovitvijo, da je globoka mreža SegNet učinkovit detektor, s čimer je ta mreža uporabna tudi pri implementaciji avtonomne vožnje avtomobila.

4 PREDLAGANA ARHITEKTURA KEYPOINTNET

Da bi zgoraj opisane dobre prakse, prvenstveno namenjene avtonomni vožnji, prenesli v biomedicinsko okolje za zaznavanje kefalometričnih točk na rentgenskih slikah, smo morali nekoliko poseči v arhitekturo nevronske mreže SegNet ter jo modificirati. V tem poglavju strnjeno opisujemo naše glavne modifikacije. Novo nastalo arhitekturo poimenujemo KeypointNet.

Ohranili smo veliki sredinski del mreže SegNet, tj. sloje kodirnika in dekodirnika, zamenjali pa smo vhodne in izhodne sloje. V nadaljevanju si bomo pogledali te modificirane sloje.

Bistvena razlika v primerjavi z arhitekturo SegNet je v samem vhodu. Namesto ene slike z labelami, kjer številke pomenijo razred/regijo kateri pripada piksel, v naši arhitekturi uporabljamo več slik. Za vsako karakteristično točko namreč uporabimo svojo t. i. labelno sliko, ki vsebuje Gaussovo jedro, poravnano z želeno lokacijo točke.

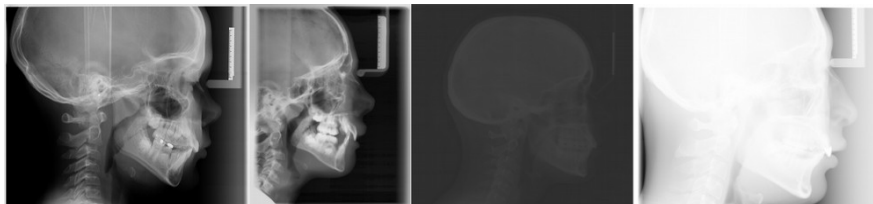
Naslednja pomembna modifikacija je sprememba aktivacijske funkcije v izhodnem sloju. Aktivacijsko funkcijo softmax smo zamenjali s sigmoidno. Eden izmed razlogov je bil tudi ta, ker za to funkcijo obstaja še t. i. izgubna varianta sloja (SoftmaxWithLoss). Ta izračuna izgubo med labelo in njeno napovedjo. Cilj predikcije oz. "faze naprej" v naši arhitekturi KeypointNet je dobiti neke vrste verjetnostno sliko za vsako kefalometrično točko (v matematičnem smislu ne gre za čisto verjetnostno funkcijo, saj seštevek na posamezni lokaciji piksla ni točno 1). Dejanski položaj kefalometrične točke določimo kot položaj maksimuma v tej t. i. verjetnostni sliki. Vsaka karakteristična točka ima torej svoj izhod, ki je podoben vhodni labeli (je enake velikosti kot vhodna slika). Izhod za kefalometrično točko na koncu predstavimo kot sivinsko sliko.

Vse spremembe arhitekture nevronske mreže smo zapisali v tekstovno datoteko .prototxt, ki je ena izmed datotek, potrebnih za učenje nevronske mreže v okviru ogrodja Caffe.

5 EKSPERIMENTALNO OKOLJE

Za razvoj naše rešitve in eksperimentiranje smo imeli na razpolago 124 unikatnih stranskih rentgenskih slik glav pacientov (t. i. lateralnih kefalogramov). Slike so bile zajete z različnimi rentgenskimi napravami in so se tudi bistveno razlikovale med seboj (velikost, grafični format, osvetlitev, šum ipd.). Ob osnovni ugotovitvi, da so bile slike zajete z različnimi napravami, smo opazili tudi, da so bile nekatere slike zajete z isto napravo, a z različnimi nastavitvami. Opažamo še, da se na nekaterih slikah ne vidijo temni detajli, nekje svetli detajli, nekatere slike pa ne zajemajo celotne glave (npr. manjka del zatilja). Povzamemo lahko, da smo imeli veliko variabilnost v podatkih.

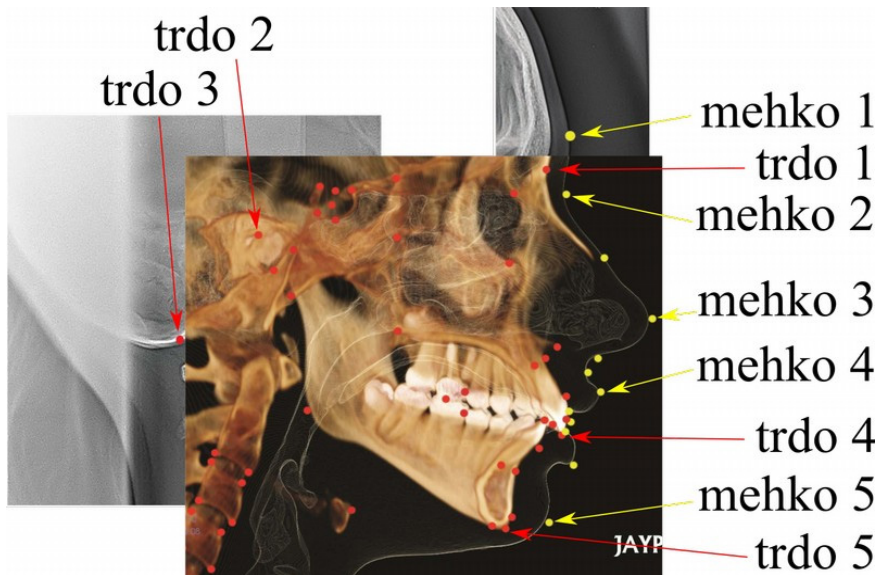
V raziskavi smo se odločili za uporabo vseh podatkov, tudi tistih nekoliko slabših. Na ta način smo želeli uporabiti čim več podatkov za učenje ter s tem povečati odpornost mreže na slabše vhode v nevronske mreže. Slika 2 prikazuje štiri primere različnih vhodnih slik.



Slika 2: Različne vhodne slike. Od leve proti desni: dobra slika, slika z manjkajočim zatiljem, pretemna slika in presvetljena slika.

Najprej smo vse slike iz različnih slikovnih formatov pretvorili v enoten grafični format PNG. Slike niso imele označenih kefalometričnih točk, zato smo jih morali označiti. Ta raziskava je bila pilotski industrijski projekt, zato smo kefalometrične točke označevali kar sami. V ta namen smo izdelali aplikacijo za anotacijo točk. Pri izbiri in označevanju radiogramov smo si pomagali z atlasom kefalometričnih točk [5]. Po lastni presoji smo izbrali 10 kefalometričnih točk (5 točk na trdem tkivu in 5 točk na mehkem tkivu) in jih označili na vseh slikah (slika 3). Vse slike in njihove anotacije smo v zadnjemu koraku predobdelave ustrezno skalirali na velikost 480×360 pikselov. Anotacija točke ima obliko slike labele (slika labele ima enako velikost kot vhodna slika).

Vsako kefalometrično točko smo torej predstavili kot posamezno sliko labele, kjer je na lokaciji kefalometrične točke vrednost 1, ostale točke v sliki labele pa imajo vrednost 0. Zaradi možnih napak pri označevanju, sami namreč nismo eksperti, smo označbam dodali še t. i. negotovost, ki smo jo modelirali v obliki Gaussovega jedra (velikost maske 31 pikslov in vrednost sigme 4).



Slika 3: Prikaz izbranih kefalometričnih točk na kompozitni sliki

Množico vseh podatkov smo razdelili v večjo učno (100 podatkov) in manjšo testno množico (24 podatkov). Naj poudarimo, da en podatek vsebuje originalno sivinsko rentgensko sliko in 10 slik labele z označbami (vsaka kefalometrična točka ima svojo sliko labele). Razdelitev je potekala naključno. Ker obstoječi vhodni sloj v nevronske mreže ne podpira našega tipa podatkov (1 vhodna slika in 10 labelnih slik za posamezen podatek), smo množico pretvorili v podatkovno bazo LMDB (*angl. Lightning Memory-Mapped Database*). Po tej pretvorbi ima en vhodni podatek dimenzijo $11 \times 360 \times 480$ (ena sivinska slika in 10 labelnih slik).

6 REZULTATI

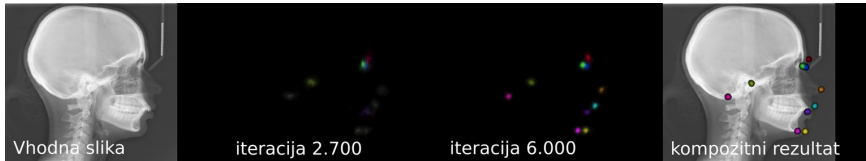
Za učenje nevronske mreže KeypointNet smo uporabili ogrodje Caffe. Postopek učenja smo izvajali na grafični postaji Inštituta za računalništvo (UM FER), ki vsebuje 4 grafične kartice NVIDIA GeForce Titan X Pascal z 12 GB GDDR5X pomnilnika.

Na grafično postajo smo najprej namestili ogrodje Caffe, zatem pa smo naložili še naše podatke v obliki baze LMDB, arhitekturno datoteko KeypointNet (.prototxt) ter nastavitveno datoteko »solver«, ki vsebuje vrednosti krmilnih hiperparametrov za zagon učenja nevronske mreže. V naši raziskavi smo z iskanjem po mreži (*angl. grid search*) poiskali ustrezen nabor hiperparametrov. Nastavitev hiperparametrov je bila naslednja: gama ima vrednost 0,8, moment ima vrednost 0,5, končno število učnih iteracij pa je 100.000. Ostale parametre smo pustili na priporočenih vrednostih, in sicer smo uporabili učni algoritem stohastičnega gradientnega sestopa (*angl. Stochastic Gradient Descent, SGD*), pravilo učenja "step" in hitrost učenja 0,1.

Posamezno učenje je trajalo 3 ure. Zaradi stohastičnosti učenja smo izvedli 5 ponovitev učenja. Rezultat učenja je datoteka .caffemodel, ki vsebuje nabor naučenih uteži nevronske mreže. Uteži je okoli 1.400.000, kar privede do velikosti izhodne datoteke okoli 5,6 MB.

Nevronsko mrežo smo učili na osnovi primerjave dveh labelnih slik (tj. temeljne resnice in napovedi mreže), zato vrednost izgube (*angl. loss*) ni merodajna o dejanski napaki položaja kefalometrične točke. Ustrezen kriterij za ocenjevanje napake položaja smo zato definirali kot evklidsko razdaljo med položajem pravilne in položajem napovedane točke. Izhod napovedi za posamezno vhodno rentgensko sliko je 10 slik label (ena slika za eno kefalometrično točko), ki so sivinske slike. Sivina piksla na posamezni labeli predstavlja "verjetnost" pravilnega položaja kefalometrične točke. Končna lokacija napovedi oz. kefalometrične točke se izbere kot položaj maksimuma v labelni sliki.

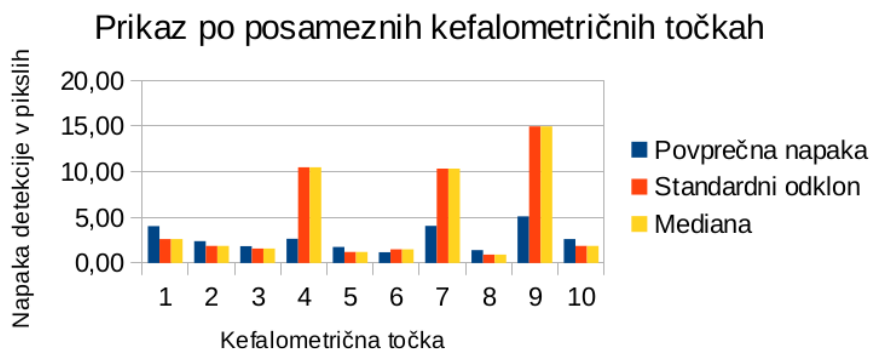
Na spodnji sliki (slika 4) vidimo vhodno sliko in dva vmesna rezultata pri različnih učnih iteracijah. Z barvo so označene različne kefalometrične točke, njihova intenziteta pa predstavlja "verjetnost". Na desni strani vidimo vhodno sliko zlito s končnim rezultatom.



Slika 4: Prikaz vmesnih rezultatov učenja nevronske mreže KeypointNet pri posamezni iteraciji.

V nadaljevanju podajamo še kvantitativne rezultate naše raziskave. Imamo torej 24 testnih slik z 10 kefalometričnimi točkami. Zaradi stohastičnosti smo učenje ponovili 5-krat. Slike se med seboj lahko bistveno razlikujejo (slika 2), hkrati pa se tudi zahtevnost detekcije različnih kefalometričnih točk razlikuje.

Poleg povprečenja napake preko vseh slik, točk in ponovitev, si bomo pogledali še povprečno napako in standardni odklon za posamezno točko (na vseh slikah in ponovitvah) in za posamezno sliko (na vseh točkah in ponovitvah). Dodali smo še metriko mediane. Na prvem grafu (glej sliko 5) so za 10 kefalometričnih točk prikazane povprečna evklidska napaka detektiranega položaja, standardni odklon in mediana te napake.



Slika 5: Povprečna evklidska napaka, standardni odklon in mediana napake detektiranega položaja posamezne kefalometrične točke (kefalometrične točke od 1 do 10) v pikslih.

Na sliki 6 vidimo podoben prikaz, vendar po različnih testnih slikah.



Slika 6: Povprečna evklidska napaka, standardni odklon in mediana napake detektiranega položaja kefalometričnih točk v pikslih za posamezno testno sliko (testna slika od 1 do 24).

Iz obeh grafov razberemo, da detekcija na večini testnih slik in večino kefalometričnih točk vrača zelo dobre rezultate. To so na primer kefalometrične točke 3, 4, 5 in 8 ter slike 6, 10, 15, 22 in 24. Imamo pa tudi nekaj osamelcev, za katere sklepamo, da bistveno negativno vplivajo na končni rezultat oz. statistike. To so kefalometrične točke 4, 7 in 9 ter testne slike 2, 9 in 11.

Na koncu še omenimo, da se na področju ortodontstva pogosto uporablja tudi metrika dveh milimetrov. Ta metrika pove odstotek detektiranih kefalometričnih točk, ki so od pravega položaja točke oddaljene do dveh milimetrov. Te metrike žal nismo mogli uporabiti v tej raziskavi, saj nismo imeli podatkov o prostorski ločljivosti naših kefalogramov, pa tudi referenčnih označb nam niso posredovali eksperti. V pripravi so novi podatki, s katerimi bomo preverili uspešnost našega postopka še s to metriko.

7 SKLEP

V tem članku smo se ukvarjali z detekcijo kefalometričnih točk na rentgenski sliki glave s pomočjo globokega učenja. Rezultati meritev so pokazali, da je povprečna evklidska napaka med označeno kefalometrično točko in njeno napovedjo s predlagano arhitekturo KeypointNet na testnih podatkih 2,7 piksla. Meritve so pokazale tudi, da je detekcija na določenih testnih slikah in kefalometričnih točkah še uspešnejša. Na osnovi tega ugotavljamo, da je naš algoritem lahko primeren pripomoček medicinskemu osebju pri izvajanju kefalometričnih analiz. Vsekakor bo pa moral strokovnjak še vedno preveriti detekcije in to predvsem zaradi morebitnih osamelcev.

Literatura

1. G. Sedej (2019), *Detekcija karakterističnih točk na rentgenskih posnetkih glave s pomočjo tehnik globokega učenja*, magistrsko delo, Maribor: Fakulteta za elektrotehniko računalništvo in informatiko, Univerza v Mariboru.
2. C. F. A. Moorrees, M. R. Kean (2003), Natural head position, a basic consideration in the interpretation of cephalometric radiographs. *American Journal of Physical Anthropology*, št. 16, str. 213–234.
3. V. Badrinarayanan, A. Handa, R. Cipolla (2017), Segnet: A deep convolutional encoder-decoder architecture for robust semantic pixel-wise labelling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, zv. 39, št. 39, str. 2481 – 2495.
4. G. J. Brostow, J. Fauqueur, R. Cipolla (2009), Semantic object classes in video: A high-definition ground truth database, *Pattern Recognition Letters*, vol 30, št. 2, str. 88-97.
5. B. Phulari (2013), *An atlas on cephalometric landmarks*. New Delhi: JP Medical Ltd.

State of the art hyperspectral data analysis and prospects with respect to AI

MATTHIAS KERSCHHAGGL

Abstract Multidimensional data such as stemming from hyperspectral imagery (HSI) with its high degree of collinearity need sophisticated multivariate methods to extract relevant features in various machine vision applications. Most of the current methods emphasize the spectroscopic nature of the data and focus less on other features such as object shape and texture also present in HSI-cubes. Corresponding features are often treated separately and sequentially rather than analysing the data following a more holistic approach. This limitation can be overcome by introducing e.g. deep learning methods (such as stacked autoencoders) to these combined spatio-spectral datasets. This talk will give a brief introduction into state of the art hyperspectral analysis algorithms as well as an outlook into the benefits of modern AI methods related to this high potential machine vision technology.

Keywords: • Multidimensional data • Hyperspectral imagery • Machine vision • Deep learning • Artificial intelligence •

CORRESPONDENCE ADDRESS: dr. Matthias Kerschhaggl, EVL DI Kerschhaggl GmbH, Josef Krainer Str. 35, 8074 Raaba-Graz, Austria, e-mail: matthias.kerschhaggl@evk.biz.

DOI <https://doi.org/10.18690/978-961-286-244-2.5>
Available at: <http://press.um.si>.

ISBN 978-961-286-244-2

DOPOLDANSKA SEKCIJA

Industrijske aplikacije
Medicinske in biomedicinske aplikacije
Drugo
Študentske aplikacije

Staranje obrazov s pomočjo globokih generativnih nevronskih mrež

NEJC VESEL, PETER PEER, VITOMIR ŠTRUC IN BLAŽ MEDEN

Povzetek Staranje obrazov je področje, ki se ukvarja z modeliranjem staranja osebe iz ene same referenčne slike. Želimo ustvariti generativni model, ki nam s pomočjo nevronskih mrež ustvari slike referenčne osebe pri različnih starostnih skupinah. Pri našem pristopu smo želeli cilj doseči z uporabo različnih generativnih arhitektur. Implementirali smo različne generativne modele, podrobneje pa predstavimo arhitekturo nasprotniškega avtoenkoderja naučenega v namene staranja obrazov. Dobljene rezultate ovrednotimo tako kvalitativno kot kvantitativno.

Ključne besede: • staranje obrazov • variacijski avtoenkoder • generativne mreže • nevronske mreže • generativne nasprotniške mreže • nasprotniški avtoenkoder •

NASLOVI AVTORJEV: Nejc Vesel, Univerza v Ljubljani, Fakulteta za računalništvo in informatiko, Večna Pot 113, 1000 Ljubljana, Slovenija, e-pošta: nejc.vesel@gmail.com. dr. Peter Peer, izredni profesor, Univerza v Ljubljani, Fakulteta za računalništvo in informatiko, Večna Pot 113, 1000 Ljubljana, Slovenija, e-pošta: peter.peer@fri.uni-lj.si. dr. Vitomir Štruc, izredni profesor, Univerza v Ljubljani, Fakulteta za elektrotehniko Tržaška cesta 25, 1000 Ljubljana, Slovenija, e-mail: vitomir.struc@fe.uni-lj.si. Blaž Meden, asistent, Univerza v Ljubljani, Fakulteta za računalništvo in informatiko, Večna Pot 113, 1000 Ljubljana, Slovenija, e-pošta: e-mail: blaz.meden@fri.uni-lj.si.

1 UVOD

Staranje obrazov je podpodročje splošnega generiranja obrazov z določenim dodatnim pogojem, da je med generiranimi slikami potrebno obdržati identiteto. Področje je staro že kar nekaj časa, vendar je z razmahom globokega učenja in nevronske mreže dobilo nov zagon. Metode globokega učenja nam omogočajo razvoj algoritmov staranja, kjer je dovolj imeti podatkovno bazo oseb različnih starosti. Uporabnost metode je večplastna. Ena od glavnih motivacij je izboljšanje sistemov za razpoznavo obrazov. Človeški obraz se namreč z leti včasih močno spremeni. Če nam uspe modelirati učinke staranja in pomlajevanja, lahko uporabimo to znanje, da naše metode prepoznavanja obrazov naredimo robustnejše na spremembe v starosti osebe. To lahko uporabljamo v forenziki, kjer lahko lažje določimo izgled zločincev, ki so iskani že več let.

Ena od dodatnih motivacij je tudi področje iskanja pogrešanih oseb. Podobno kot pri lovljenju ubežnikov, želimo iz stare obrazne slike predvideti trenutni izgled osebe, tako da upoštevamo spremembe obraznih potez zaradi staranja. Te slike se lahko nato javno objavi, kot referenco za trenutni izgled osebe, kar omogoča lažje iskanje.

2 SORODNA DELA

2.1 Avtoenkoder

Ena od glavnih arhitektur na področju generativnih modelov je avtoenkoder. Avtoenkoderji so sestavljeni iz dveh glavnih delov enkoderja E in dekoderja D. Cilj enkoderja je stisniti vhodne podatke v latentno (skrito) reprezentacijo manjše dimenzije, cilj dekoderja pa je iz latentne reprezentacije rekonstruirati vhodne podatke. V splošnem se bo v procesu kodiranja in odkodiranja vedno zgodila izguba informacij, saj je latentni prostor manjše dimenzije kot vhodni podatek. Na primer, sliko velikosti 28x28 točk stisnemo v vektor velikosti 10x1. V praksi je to koristno za namene razšumljanja slik in zapolnjevanja nepopolnih delov slik [1]. V teh primerih mrežo naučimo, da iz nepopolnih (manjkajočih, šumnatih) slik generira čiste slike.

2.2 Generativne nasprotniške mreže

Generativne nasprotniške mreže so bile prvič predstavljene v članku [2]. Glavna ideja algoritma je, da pomerimo generativni model (generator) proti nasprotniku (discriminator), ki določa ali je generiran rezultat podoben tistemu, ki ga želimo modelirati. Predstavljamo si lahko bitko med ponarejevalcem denarja in strokovnjakom, ki določa ali je kos denarja pristen. Želimo, da oba akterja skozi iterativni nasprotniški proces izboljšujeta drug drugega. Želimo, da generator in diskriminator ostajata v ravnovesju. Če eden od njiju postane preveč učinkovit, težko pride do nadaljnjega izboljšanja v kvaliteti rezultatov, kar je tudi ena glavnih slabosti tega pristopa.

2.3 Variacijski avtoenkoderji

Eden od glavnih pristopov pri generativnem strojnem učenju je uporaba variacijskih avtoenkoderjev [3]. Ti so posplošitev avtoenkoderja, kjer enkoder pogojimo tako, da ustvarjeni latentni vektorji okvirno sledijo normalni porazdelitvi. Generiranje novih slik je torej samo vzorčenje iz normalne porazdelitve z določeno srednjo vrednostjo in standardno deviacijo, ki jo dobimo iz mreže. Vedno je potreben kompromis med rekonstrukcijsko napako in prileganjem normalni porazdelitvi.

2.4 Staranje z uporabo pogojnih generativnih nasprotniških mrež

Eden od pristopov, opisan v [4], simulira staranje s pomočjo pogojne generativne nasprotniške mreže. Glavna ideja razdeli postopek na tri dele:

- pogojno generativno nasprotniško mrežo naučimo generirati slike;
- glede na podano sliko x starosti y_0 , poišči latentni vektor z^* , pri katerem generator generira sliko, ki je najbolj podobna podani. Torej želimo minimizirati razliko med x in $\hat{x} = G(z^*, y_0)$;
- staranje dosežemo tako, da generatorju ob optimalnem vektorju z^* namesto originalne starosti podamo ciljno starost y_{cilj} , torej $x_{cilj} = G(z^*, y_{cilj})$.

V tem primeru generator na vhod sprejme vektor šuma z dimenzije 100×1 . Nato ga s pomočjo polnopovezanega sloja in preoblikovanja razširi na dimenzijo $1024 \times 4 \times 4$, katero nato s pomočjo zaporedja obratno koračnih kovolucij spremenimo v dimenzijo slike, ki je $64 \times 64 \times 3$.

Oblika diskriminatorja je simetrična tej, z razliko, da namesto obratno koračnih, uporabljamo koračne kovolucije z velikostjo koraka 2. Arhitekturi se ločita tudi v tem, da je zadnji sloj v diskriminatorju polnopovezan z velikostjo 1, saj je cilj vračanje enega bita informacije.

3 MODEL STARANJA

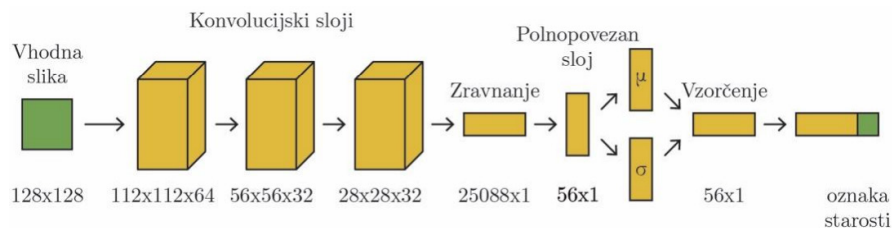
V tem razdelku predstavimo nasprotniški avtoenkoder, ki smog a razvili v okviru našega razikovlanega dela za namene simulacije staranja slik obrazov.

3.1 Opis modela

Nasprotniški avtoenkoder, ki smog a razvili, je osnovan na konvolucijski arhitekturi. Na vhod smo podajali barvne slike velikosti 224×224 slikovnih točk ter oznake o starosti oseb. V primeru nasprotniškega avtoenkoderja je naš model sestavljen iz treh glavnih delov, tj.:

- enkoder E, ki je zadolžen, da se vhodni podatki zakodirajo v latentno reprezentacijo,
- generator (dekoder) G, ki je zadolžen za generiranje slike iz latentne reprezentacije, ter
- diskriminator D, ki je zadolžen za ohranjanje enotske naravne porazdelitve latentnega vektorja.

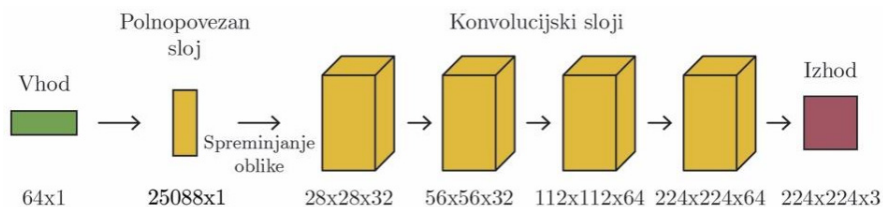
Kot lahko vidimo na sliki 1, je enkoder sestavljen iz treh zaporednih konvolucijskih slojev, nato pa reprezentacijo sploščimo in iz nje naredimo dva polnopovezana sloja srednje vrednosti ter variance. Uporabljamo torej normalno porazdeljeni enkoder.



Slika 1: Struktura enkoderja v nasprotniškem avtoenkoderju

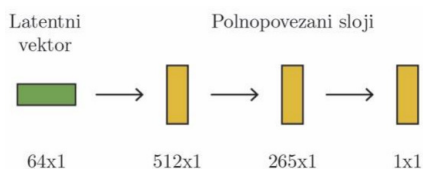
V vsakem od konvolucijskih slojev uporabljamo velikost jedra (3,3), ter aktivacijo z usmerjeno linearno enoto, in za manjšanje višine in širine uporabljamo koračenje s korakom velikosti 2.

Strukturo dekoderja lahko vidimo na sliki 2. Prvi sloj je polnopravezan sloj velikosti 25088. Nato sledijo štiri konvolucijski sloji, med katerimi v namene povečanja dimenzije uporabljamo sloj nadvzorčenja (angl. upsampling layer). Za polnopravezani sloj ter za prve tri konvolucijske uporabljamo aktivacijo z usmerjeno linearno enoto, za zadnjega pa uporabljamo sigmoid aktivacijo. Vsa jedra konvolucij so velikosti 3x3.



Slika 2: Struktura dekoderja v nasprotniškem avtoenkoderju.

Diskriminator ima zelo enostavno strukturo, ki jo lahko vidimo na sliki 3.



Slika 3: Struktura diskriminatorja v nasprotniškem avtoenkoderju.

Na vhod dobi latentni vektor, ki ga generira enkoder, nato sledita dva polnopovezana sloja velikosti 512 in 256 nevronov, v katerih uporabljamo aktivacijo s puščajčo usmerjeno linearno enoto. Zadnji sloj je polnopovezan velikosti 1, pri katerem uporabljamo sigmoid aktivacijo.

Učenje poteka v dveh korakih. Najprej učimo diskriminator. Želimo, da nam pove ali je latentni vektor normalno porazdeljen, zato mu med učenjem podajamo normalno porazdeljene vektorje, ki jih mora klasificirati kot pravilne (1) ter ostale, za katere želimo, da jih klasificira kot nepravilne (0). Kot kriterijsko funkcijo v modelu diskriminatorja uporabljamo binarno navzkrižno entropijo.

Po učenju diskriminatorja učimo še skupni model, ki ga dobimo tako, da sestavimo skupaj vse tri dele. Imamo torej enkoder, ki iz vhodne slike in oznake o starosti ustvari latentno reprezentacijo. To podamo dekoderju, ki nam ustvari rekonstruirano sliko. Latentni vektor podamo diskriminatorju in dobimo veljavnost. Naš model na vhod dobi vhodno sliko in oznako starosti. Na izhod pa vrača rekonstruirano sliko ter veljavnost. Kot kriterijsko funkcijo za rekonstrukcijo uporabljamo MSE:

$$MSE(Y, \hat{Y}) = \frac{1}{n} \sum_{i=1}^n (Y_i - \hat{Y}_i)^2 \quad (1)$$

3.2 Dodatna optimizacija

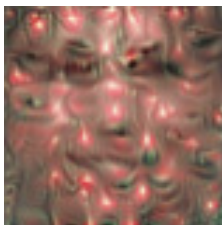
Implementirali smo dodatno optimizacijo identitete, ki je bila omenjena v [4]. Glavna ideja metode je, da latentno reprezentacijo še bolj optimiziramo iz stališča ohranjanja identitete tako, da uporabimo eno od optimizacijskih metod, ki nam ob začetnem približku konvergira v boljše rešitev. V članku so idejo uporabljali na primeru generativnih nasprotniških mrež, vendar je težava, da pri generativnih nasprotniških mrežah ne dobimo preslikave iz vhodne slike v latentni vektor, ampak samo obratno. Zato so morali avtorji v članku naučiti dodatno nevronska mrežo, da se nauči to preslikavo.

Ker v našem primeru uporabljamo avtoenkoder, dodatna nevronska mreža ni potrebna, saj je ena od glavnih lastnosti avtoenkoderjev ravno to, da nam določijo preslikavo med vhodnimi podatki in latentnim prostorom. Določiti

moramo metriko, ki je bolj primerna za namene ohranjanja identitete kot klasična primerjava, kjer gledamo razliko na nivoju posameznih slikovnih točk. Odločili smo se, da bomo uporabili VGG-Face arhitekturo, ki je bila prednaučena za klasifikacijo identitet na obraznih slikah [5]. Ko vhodno sliko spustimo skozi to mrežo, lahko iz zadnjega polnopravezanega sloja preberemo vektor značilk, ki nam predstavlja identiteto osebe. V našem primeru za vsako vhodno sliko dobimo vektor značilk velikosti 2048.

Kot identitetno razliko med dvema osebama lahko uporabimo kosinusno razdaljo. Celoten optimizacijski postopek lahko zapišemo v naslednjih korakih:

- vhodno sliko vstavimo v enkoder nasprotniškega avtoenkoderja da dobimo latentno reprezentacijo z_0 ;
- v izbrano optimizacijsko metodo vstavimo z_0 kot začetni približek;
- optimizacijska metoda nam izračuna optimizirani latentni vektor. Vzame trenutno latentno reprezentacijo, s pomočjo dekoderja generira sliko in jo primerja z referenčno sliko. Nato popravi približek in postopek se ponavlja do konvergence;
- ko metoda dokonča konvergacijo, dobljeni vektor vstavimo v dekoder ter spreminjamo oznako starosti, da dobimo slike različnih starosti.



Slika 4: Artefakti na sliki ob prisotnosti kritika identitete pri učenju.

4 PODATKOVNA ZBIRKA TER EKSPERIMENTALNI PROTOKOL

Pri učenju našega modela smo za učno množico uporabljali bazo slik UTKFace [9], ki vsebuje približno 20.000 slik oseb različnih etničnih izvorov, spola ter starosti. Slike v bazi so obrezane ter poravnane ter vsebujejo osebe pri različnih osvetlitvah, obraznih mimikah ter stopnjah pokritosti. Baza je neuravnotežena proti starostnim skupinam med 20 in 40 let. Prav tako je manjše število slik ljudi starejših od 70 let, kar pa je za naše potrebe sprejemljivo, saj nam staranje do ekstremnih starosti ni poglobitnega pomena. Pri ovrednotenju smo uporabljali podatkovno bazo FG-Net, ki vsebuje slike oseb pri različnih starostih. Za namene naše evalvacije smo za vsako osebo izbrali dve fotografiji, ki sta predstavljali osebo z več kot 10 let časovnega razmika.

Za ovrednotenje modela smo sestavili tri različne anketne vprašalnike. Pri prvem vprašalniku smo osebam kazali par slik, kjer je prva bila referenčna slika, za drugo pa je veljalo, da je z verjetnostjo 0,5 naključna generirana slika iste starostne skupine in z verjetnostjo 0,5 referenčna oseba postarana na to starost.

Pri drugem testu smo imeli samo prave pare, torej referenčno sliko in generirano sliko, ki je bila narejena tako, da smo neko drugo (mlajšo) podobo referenčne osebe postarali s pomočjo naše mreže.

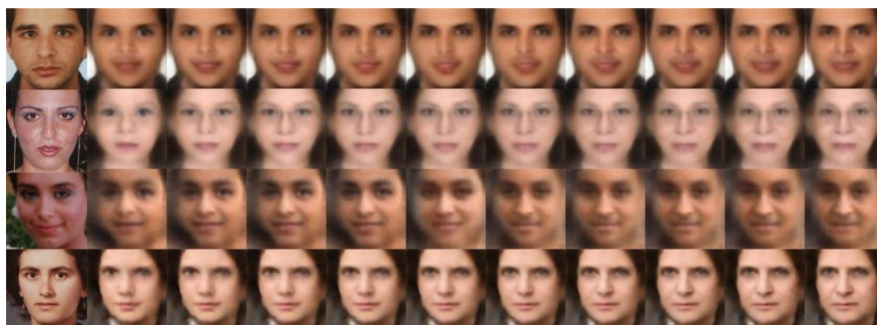
Pri vsakem paru so lahko ocenjevalci izbrali eno od treh možnosti:

- Na slikah nista isti osebi.
- Nisem prepričan.
- Na slikah sta isti osebi.

Pri tretjem testu pa smo kazali generirane slike in ocenjevalci so se morali odločiti, v katero starostno skupino spada oseba na sliki. Pri vseh treh testih je bilo potrebno oceniti 50 parov. Vzorec ljudi, ki so rešili vse tri teste je 10.

5 REZULTATI

Najboljše rezultate smo dobili s pomočjo nasprotniškega avtoenkoderja. Najprej si pogledjmo nekaj dobljenih vizualnih rezultatov na sliki 5, kjer se identiteta osebe precej ohrani, poleg tega pa se oseba tudi vidno postara.

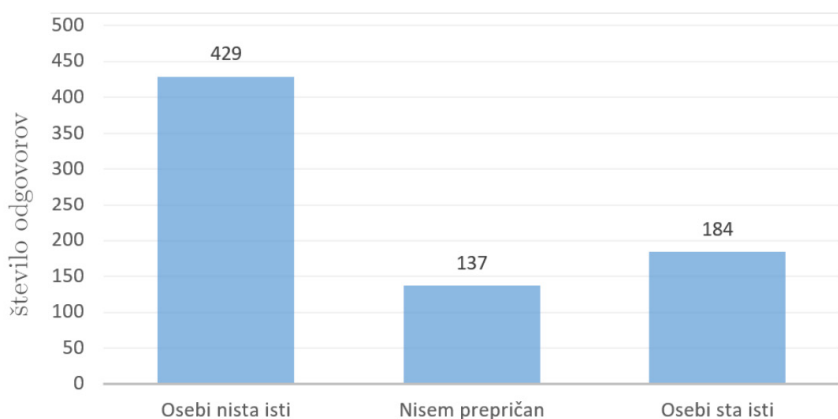


Slika 5: Generirani obrazi s pomočjo nasprotniškega avtoenkoderja.

Največja težava nastane pri vhodnih slikah, kjer nastopajo osebe z brki, brado ali očali. V teh primerih naš model težko zazna obraz in generirane slike so popačene. To se zgodi predvsem zato, ker v učni množici ni dovolj fotografij oseb z brado in ostalimi dodatki, npr. očali. Kljub temu da so vizualni rezultati na prvi pogled delovali zelo solidno, pa je evalvacija rezultatov pokazala velike pomanjkljivosti v naših rezultatih.

5.1 Test z mešanimi pari

Največjo težavo predstavlja velika mera napačno negativno klasificiranih vzorcev. To nam pove, da nam identite ni uspelo pravilno ohraniti. Če pogledamo porazdelitev izbranih možnosti na grafu na sliki 6 vidimo, da imamo preveč odgovorov, ki navedejo, da osebi nista enaki.



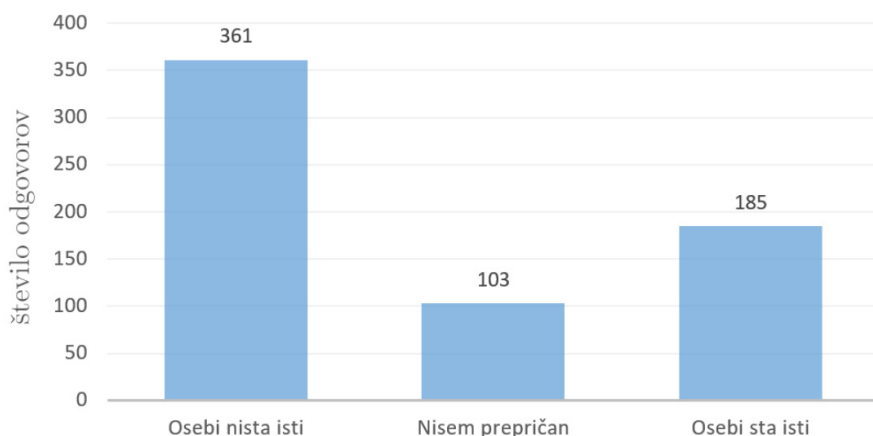
Slika 6: Porazdelitev odgovorov pri testu z mešanimi pari.

Pri popolni metodi bi si želeli enakovredno število odgovorov za Osebi nista isti in za Osebi sta isti ter minimalno število odgovorov za Nisem prepričan. Izračunali smo tudi metriko natančnosti, za katero smo dobili vrednost 0,54 ter priklica, kjer smo dobili 0,33.

5.2 Test z zgolj dejanskimi pari

Zanimiva je primerjava z drugim testom, kjer so nastopali samo dejanski pari med referenčno sliko ter generirano sliko. Dobili smo porazdelitev, ki je vidna na sliki 7. Vidimo, da je v tem primeru manjši odstotek odgovorov Osebi nista isti, kar pomeni, da naša metoda vsaj delno deluje in ohrani identiteto. Pri drugem testu dobimo torej, da je 55,7 % odgovorov *Osebi nista isti*, 28,5 % *Osebi sta isti* ter 15,8 % *Nisem prepričan*.

Če to primerjamo z vrednostmi v enem od referenčnih člankov s podobnim protokolom [10] vidimo, da so naši rezultati slabši, saj so avtorji v tem članku dosegli 48 % odgovorov Osebi sta isti. Velja omeniti, da se je naš protokol nekoliko razlikoval. V članku so avtorji vsaki osebi pokazali tri slike: generirano sliko, referenčno sliko, ki je bila vhod za generirano sliko ter sliko osebe pri cilji starosti. Menimo, da prikaz vhodne slike negativno vpliva na točnost evalvacije rezultatov, saj anketirancem omogoča primerjati vhodno sliko z referenčno starostno sliko, kar deluje sugestivno.

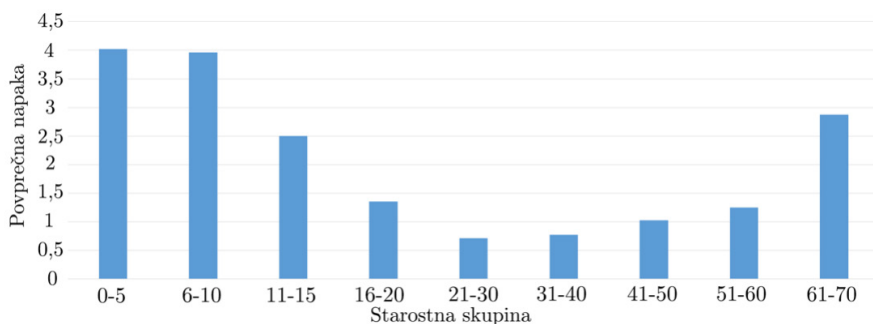


Slika 7 Porazdelitev odgovorov pri testu s samo dejanskimi pari.

5.3 Test ocenjevanja starosti

Poglejmo si še rezultate testa ocenjevanja starosti. Za vsako od slik so morali anketiranci izbrati, kateri starostni skupini mislijo, da pripada. Tudi tukaj smo prišli do presenetljivih rezultatov, saj so mladostnike starosti od (0-20) ljudje večkrat klasificirali v starosti med 20 in 40 let.

Imamo graf povprečne absolutne napake med dejansko in predvideno starostjo osebe na sliki 8, ki je ločena glede na dejansko starost generirane osebe. Vidimo, da je napaka zelo velika v zgodnjih letih ter pri generiranju obrazov starosti večje od 60 let.



Slika 8 Porazdelitev povprečne absolutne napake pri starostnih skupinah.

6 ZAKLJUČEK

V referatu smo predstavili rezultate dela na problemu staranja slik obrazov. Pokazali smo, da lahko s pomočjo nasprotniškega avtoenkoderja razmeroma učinkovito simuliramo proces staranja za slike obrazov, pri čemer pa se pri tem postopku do neke mere izgubi informacija s identiteto osebe na sliki. V okviru nadaljnjih raziskav se bomo zato posvetili problemu ohranjanja identitete v postaranih slikah.

Literatura

1. J. Xie, L. Xu, E. Chen, Image denoising and inpainting with deep neural networks, in: *Advances in neural information processing systems*, 2012, pp. 341–349.
2. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial nets, in: Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, K. Q. Weinberger (Eds.), *Advances in neural information processing systems 27*, Curran Associates, Inc., 2014, pp. 2672–2680.
3. D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.
4. G. Antipov, M. Baccouche, J.-L. Dugelay, Face aging with conditional generative adversarial networks, in: *The IEEE International conference on Image processing (ICIP)*, 2017, pp. 2089–2093.
5. K. Grm, V. Štruc, A. Artiges, M. Caron, and H. K. Ekenel, "Strengths and weaknesses of deep learning models for face recognition against image degradations," *IET Biometrics*, vol. 7, no. 1, pp. 81-89, 2017
6. C. Zhu, R. H. Byrd, P. Lu, and J. Nocedal, "Algorithm 778: L-BFGS-B: Fortran subroutines for large-scale bound-constrained optimization," *ACM Transactions on Mathematical Software (TOMS)*, vol. 23, no. 4, pp. 550-560, 1997.
7. M. J. Powell, "A direct search optimization method that models the objective and constraint functions by linear interpolation," in *Advances in optimization and numerical analysis*: Springer, 1994, pp. 51-67.
8. J. A. Nelder and R. Mead, "A simplex method for function minimization," *The computer journal*, vol. 7, no. 4, pp. 308-313, 1965.
9. *UTKFace*. Pridobljeno na: <http://aicip.eecs.utk.edu/wiki/UTKFace>
10. Z. Zhang, Y. Song, and H. Qi, "Age progression/regression by conditional adversarial autoencoder," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 5810-5818.

Detekcija razpok v steklu – klasična metoda proti globoki nevronske mreži

MARIJA IVANOVSKA, KLEMEN GRM, VITOMIR ŠTRUC IN JANEZ PERŠ

Povzetek Predstavljena je problematika detekcije razpok v steklu z uporabo cenene strojne opreme (osvetlitev v vidnem spektru in RGB kamera). Kot primer so uporabljene steklene vial, kjer smo na množici vial umetno povzročili razpoke. Slike vial so bile zajete v nadzorovanem okolju, kjer smo lahko vialo nadzorovano rotirali okrog glavne osi, ter spreminjali smer osvetlitve. Zasnovali smo klasično metodo strojnega vida, ki temelji na strokovnem znanju avtorjev, za primerjavo pa smo uporabili še prilagojeno globoko nevronske mrežo VGG-16. Obe metodi dosežeta 100% delovanje, iz ablacijske študije pa je očitno da je najbolj pomemben faktor pri obeh metodah ravno velika, redundantna količina vhodne slikovne informacije.

Ključne besede: • strojni vid • razpoke v steklu • Gaborjevi filtri • nevronske mreže • loboko učenje • konvolucijske nevronske mreže •

NASLOVI AVTORJEV: Marija Ivanovska, Univerza v Ljubljani, Fakulteta za elektrotehniko, Tržaška cesta 25, 1000 Ljubljana, Slovenija, e-pošta: marijasivanovska@gmail.com. Klemen Grm, mladi raziskovalec, Univerza v Ljubljani, Fakulteta za elektrotehniko, Tržaška cesta 25, 1000 Ljubljana, Slovenija, e-pošta: klemen.grm@fe.uni-lj.si. dr. Vitomir Štruc, izredni profesor, Univerza v Ljubljani, Fakulteta za elektrotehniko Tržaška cesta 25, 1000 Ljubljana, Slovenija, e-mail: vitomir.struc@fe.uni-lj.si. dr. Janez Perš, docent, Univerza v Ljubljani, Fakulteta za elektrotehniko Tržaška cesta 25, 1000 Ljubljana, Slovenija, e-mail: janez.pers@fe.uni-lj.si.

1 UVOD

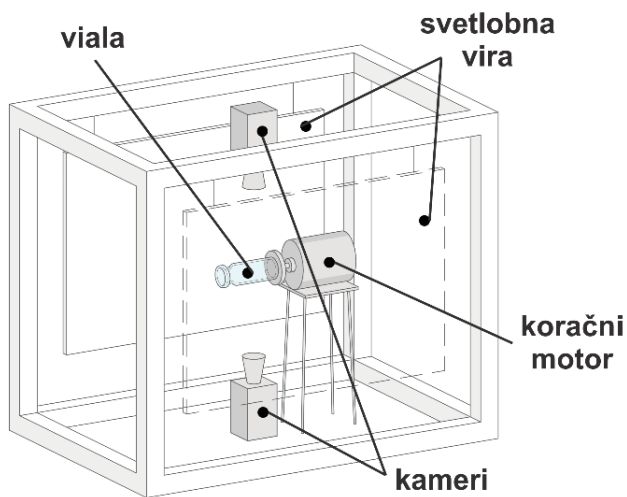
V tem članku se bomo osredotočili na dva različna pristopa rešavanja industrijskega problema iz področja farmacije. Predstavili bomo iskanje poškodb na steklu vial najprej s pomočjo klasičnih metod strojnega vida, nato pa še z uporabo metod globokega učenja. Za razvoj algoritmov je bila uporabljena zbirka vzorcev, ki smo jo ustvarili sami. V poglavju 2 je predstavljen način nastajanja zbirke in analiza pridobljenih slik. V poglavju 3 predstavimo predložene metode za zaznavo poškodovanih vial. Rezultati razvitih algoritmov so prikazani v poglavju 4, v poglavju 5 pa je predstavljena ablacijska študija, ki je bila izvedena z zmanjševanjem količine vhodnih podatkov. V poglavju 6 podamo zaključke pri primerjavi obeh pristopov.

2 ZBIRKA VZORCEV

2.1 Zajem slik vzorcev

Osnova za razvoj algoritmov strojnega vida za iskanje napak na steklu vial je zbirka slik poškodovanih in nepoškodovanih vial. Za ustvarjanje takšne zbirke smo najprej podrobno proučili lastnosti stekla iz katerega so viale narejene [1], ter tipične napake, ki nastanejo na njihovem steklu med proizvodnjo zdravil [2,3]. S pridobljenimi znanji smo nato umetno povzročili različne poškodbe, ki so predstavljale razpoke v steklu ali odkrušeni delčki stekla. Celotna množica vial je zajemala 32 poškodovanih in 32 nepoškodovanih vial, ki so bile uporabljene za pridobivanje slik.

Za fotografiranje vial smo zgradili avtomatiziran sistem za zajem slik, ke je vključeval: 2 LED panela kot svetlobna vira, koračni motor za obračanje viale okrog njene osi, 2 kameri za zajem slik in krmilna vezja za krmiljenje osvetlitve, motorja in kamer. Skica sistema je prikazana na Sliki 1.



Slika 1: Prototip sistema uporabljen za zajem slik vial.

Med zajemom slik je bila uporabljena tehnika temnega polja. Viala, ki je pritrjena na platformo motorja se obrača od začetne (0°) do končne pozicije (180°) po korakih za 15° . Po vsakem koraku obračanja vsaka kamera zajame 32 slik pri 32 različnih osvetlitvah. Ta sekvenca se izvaja, dokler ne zajamemo vseh 832 slik, 416 z vsako kamero, pri 32 različnih osvetlitvah, v 13 različnih legah viala. Z različnimi orientacijami poskrbimo za zajem slik celotnega plašča in celotnega obsega grla viala. Z različnimi osvetlitvami, ki jih izvedemo s pomočjo prikaza različnih vzorcev na programirljivem LED panelu pa posvetimo na vialo iz različnih zornih kotov, s čimer zagotovimo, da bo prišlo do loma svetlobnih žarkov, vkolikor je steklo poškodovano. Kot rezultat smo dobili slike, katerih svetla območja različnih velikostih in nepravilne oblike ponazarjajo dejanske poškodbe stekla. V glavnem so razpoke vidne kot tanke svetle linije, ki se razvejijo v več smeri, njihova debelina pa ne presega 1mm. Vsem slikam pa so skupna svetla območja pravilne oblike in enake velikosti, ki so vidna na istih mestih vseh vial in so posledica bleščanja. Izločanje vpliva bleščanja na slikah je bil eden izmed največjih izzivov med razvojem algoritmov za iskanje razpok.

2.2 Predstavitev zbirke slik

Za namen učenja in evaluacije naučenih modelov nujno potrebujemo tako imenovani zlati standard (ang. ground truth), ki ga sestavljajo oznake vzorcev. V našem primeru smo zbirko slik označevali na dva načina. Najprej smo vsakemu vzorcu pripisali pripadnost določenemu razredu v odvisnosti od prisotnosti poškodb na njem. Nato smo poškodbe poškodovanih vzorcev označili z mnogokotnikom, ki obriše poškodbo tako, da se ta nahaja v notranjem delu mnogokotnika. Da bi se izognili ročnega označevanja mnogokotnikov na vseh slikah, smo to naredili samo na eni sliki posamezne viale (pri eni orientaciji). Mnogokotnike, ki obrisujejo poškodbe določene viale pri ostalih orientacijah smo pridobili z matematično transformacijo točk v prostoru, saj smo poznali točno orientacijo vial, ki jo je obračal koračni motor.

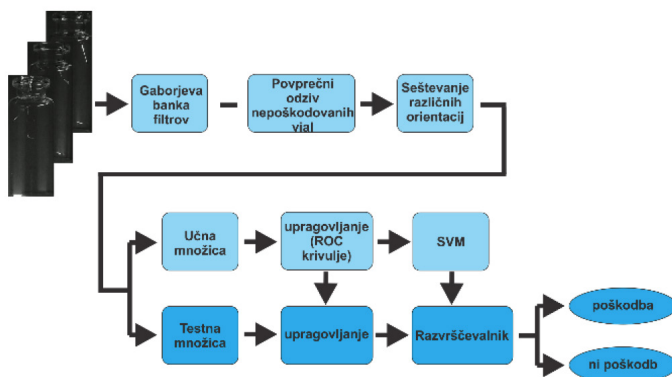
Med 64 vialami je 17 vial s poškodbami na trupu, 12 s poškodbami na grlu in 3 s poškodbami na obeh mestih. Vsi vzorci so bili razdeljeni v dve neprekrivajoči množici - učno in testno množico. Pri tem smo imeli ločeni učni oz. testni množice za poškodbe na trupu in poškodbe na grlu vial. Učna množica za iskanje poškodb na grlu vial je vsebovala 8 poškodovanih ter 8 nepoškodovanih vzorcev, medtem ko je učna množica za iskanje poškodb na trupu vial vsebovala 10 poškodovanih in 10 nepoškodovanih vzorcev. Vsi ostali vzorci so bili del testnih množic.

3 METODE

Za iskanje poškodb na steklu vial smo razvili dva algoritma. Prvi algoritam temelji na klasičnih metodah strojnega vida, drugi pa uporablja metode globokega učenja. Vsak algoritam je bil uporabljen dvakrat – prvič za zaznavanje poškodb na grlu vial, drugič za zaznavanje poškodb na trupu vial.

3.1 Klasične metode

Pri uporabi klasičnih metod smo predlagali zaporedje obdelovanja vzorcev, ki je grafično prikazano na Sliki 2.



Slika 2: Predlagan algoritem za iskanje poškodb v steklu vial z uporabo klasičnih metod strojnega vida

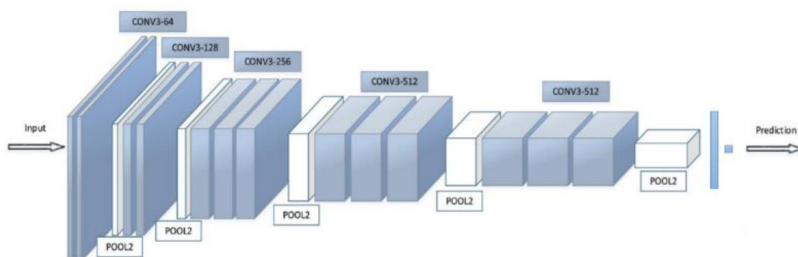
Kot je razvidno iz Slike 2, posamezen vzorec najprej obdelamo z banko Gaborjevih filtrov [4,5,6], ki jo sestavljajo filtri z valovnimi dolžinami 3, 6 ali 9 slikovnih točk in 12 različnimi orientacijami na intervalu $[0^\circ, 165^\circ]$. Iz odziva je bil nato odštet povprečni odziv vseh nepoškodovanih vial, s čimer smo učinkovito izločili vpliv bleščanja na končni rezultat - ostal nam je samo odziv banke filtrov na dejanske poškodbe. Nato smo združili informacije, ki smo jih imeli na osnovi različnih slik iste vial pri isti osvetlitvi, vendar pri različnih orientacijah. Ko je v steklu prisotna razpoka ali poškodba, le-ta pri rotaciji rotira skupaj z vialo in tako spreminja svojo lokacijo na sliki. Ker se odziv poškodb na banko filtrov prepozna po višjih vrednostih slikovnih točk, ki predstavljajo poškodbo, je rezultat združevanja odzivov pri različnih orientacijah slika, kjer vrednost vsakega slikovnega elementa predstavlja maksimum med vsemi vrednostmi istoležečih slikovnih elementov posameznih slik ($P(i,j) = \max(P_1(i_1,j_1), P_2(i_2,j_2), \dots, P_{32}(i_{32},j_{32}))$). Da dobimo segmentirano sliko z dvema področji (področje nepoškodovanega stekla in področje poškodb) smo v nadaljevanju takšno sliko upravgovili s pomočjo ustreznega eksperimentalno določenega praga upravgovljanja. Pomagali smo si z ROC krivuljami [9], ki so na osnovi učnih vzorcev prikazovale točnost razvrščanja vzorcev v primeru različnih pragov. Za vsako osvetlitev od 32 je bil določen optimalen prag upravgovljanja.

Za dokončno razvrstitev vseh vial v skupini poškodovanih ali nepoškodovanih, smo združili informacije o isti viali pri različnih osvetlitvah. Z uporabo metode podpornih vektorjev [7] smo zgradili razvrščevalnik po metodi podpornih vektorjev (angl. Support Vector Machine, SVM) z jedrom RBF. Po upravljanju vsake od 32 slik ene viale smo izračunali število slikovnih elementov, ki predstavljajo področja potencialnih poškodb ter vse te vsote sestavili v 32 dimenzijski vektor. Takšne vektorje iz učne množice smo nato izkoristili za učenje modela SVM. S predstavljenim postopkom smo tako pri iskanju poškodb na grlu vial kot tudi pri iskanju poškodb na trupu vial dosegli 100% razpoznavnost poškodovanih in nepoškodovanih vial.

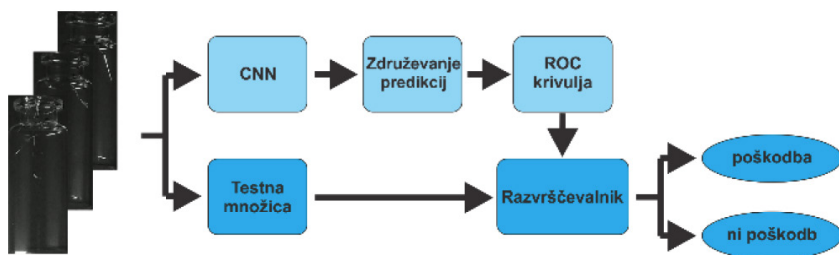
3.2 Metode globokega učenja

Pri razvoju algoritma z globokim učenjem smo kot osnovni model za luščenje značilk uporabili konvolucijsko mrežo VGG16 [8], pri kateri smo polno povezane plasti na vrhu mreže zamenjali z združevalnimi plastmi globalnega povprečenja velikosti 512, katerim sledita dve polno povezani plasti. Prva polno povezana plast ima velikost 128 in aktivacijsko funkcijo ReLU. Druga polno povezana plast, ki je obenem tudi izhodna plast nevronske mreže, ima velikost 1 in sigmoidno aktivacijsko funkcijo. Prilagojena nevronska mreža je prikazana na Sliki 3.

Ustvarjeno arhitekturo nevronske mreže smo uporabili dvakrat, prvič za učenje razvrščevalnika poškodb na grlu vial, drugič pa za učenje razvrščevalnika poškodb na njihovem trupu. Predlagan pristop za razvoj razvrščevalnika vial s pomočjo globoke konvolucijske mreže je prikazan na Sliki 4.



Slika 3: Prilagojena globoka nevronska mreža (osnova je globoka konvolucijska nevronska mreža VGG16) uporabljena za iskanje razpok v steklu vial



Slika 4: Predlagan algoritem za iskanje poškodb v steklu vial z uporabo metod globokega učenja

Vhod v mrežo so posamezne slike vial, pri eni osvetlitvi in eni orientaciji. Glede na to, da je vsaka viala predstavljena s 416 vzorci, lahko tako kot pri konvencionalnih metodah, tudi v tem primeru združujemo ocene večih vzorcev, s čimer izboljšamo rezultate razvrščanja. Zato, da bi omogočili pravično primerjavo obeh metod, konvencionalne in metode globokega učenja, smo v končni fazi združili predikcije vseh 416 vzorcev posamezne viale tako da smo poiskali povprečje predikcij, pri čemer je vsaka predikcija prispevala enak delež k skupni oceni. Na takšen način smo tudi s tem pristopom dobili 100% razpoznavnost poškodovanih in nepoškodovanih vial.

5 ABLACIJSKA ŠTUDIJA

Zbrikanje slik, ki je bila uporabljena pri razvoju algoritmov za razpoznavanje razpok, vsebuje 416 slik ene viale. Zajem tako velikega števila slik posamezne viale je bil za raziskovalne namene neizogiben, saj nismo vedeli kako bodo posamezne osvetlitve in število rotacij vial vplivale na zaznavanje poškodb. Poleg tega smo se srečali še s problemom zadostno velike učne množice za učenje nevronske mreže. Posledično smo se tako pri konvencionalnih metodah, kot tudi pri nevronske mreže za končno odločitev o prisotnosti poškodbe na posamezni viali odločali na osnovi vseh njenih 416 vzorcev.

Ker se zavedamo, da je čas v industriji ključnega pomena in, da se od algoritmov pričakuje tako natančnost, kot tudi maksimalna možna hitrost procesiranja vzorcev, smo izvedli ablacijsko študijo pri kateri smo sistematično izločevali značilke (zmanjševali smo število osvetlitev in orientacij ene viale) z

namenom ugotavljana kako to vpliva na končni rezultat razvrščanja. Najprej smo začeli z izločevanjem vzorcev zajetih pri določenih osvetlitvah tako, da smo na začetku izločili vsako drugo osvetlitev, nato smo izločili vsako tretjo osvetlitev itd. Po končanih iteracijah smo primerjali rezultate razpoznavanja razpok pri 32, 16, 10, 8, 7, 6, 5, 4, 3, 2 in 1 osvetlitvi. V tej fazi smo primerjali rezultate pri združevanju informacij vseh 13 orientacij vzorcev. V drugi fazi ablacijske študije pa smo ob istočasnem zmanjševanju števila osvetlitev, kot smo ga opisali, začeli zmanjševati tudi število orientacij vial. Na koncu smo primerjali rezultate razpoznavanja pri 13, 7, 5, 4 in 3 orientacijah. Rezultati ablacijske študije so prikazani na slikah 5, 6, 7 in 8.

Odstotek pravilno zaznanih poškodb na grlu vial v primeru uporabe konvencionalnih metod						Odstotek pravilno zaznane odstotnosti poškodb na grlu vial v primeru uporabe konvencionalnih metod					
	3 orientacije	4 orientacije	5 orientacij	7 orientacij	13 orientacij		3 orientacije	4 orientacije	5 orientacij	7 orientacij	13 orientacij
1 osvetlitev	85,71	85,71	100,00	100,00	100,00	1 osvetlitev	79,17	79,17	95,83	95,83	100,00
2 osvetlitvi	85,71	85,71	100,00	100,00	100,00	2 osvetlitvi	79,17	91,67	95,83	95,83	100,00
3 osvetlitve	100,00	100,00	100,00	100,00	100,00	3 osvetlitve	91,67	91,67	95,83	100,00	100,00
4 osvetlitve	100,00	100,00	100,00	100,00	100,00	4 osvetlitve	91,67	100,00	100,00	100,00	100,00
5 osvetlitev	100,00	100,00	100,00	100,00	100,00	5 osvetlitev	100,00	100,00	100,00	100,00	100,00
6 osvetlitev	100,00	100,00	100,00	100,00	100,00	6 osvetlitev	100,00	100,00	100,00	100,00	100,00
7 osvetlitev	100,00	100,00	100,00	100,00	100,00	7 osvetlitev	100,00	100,00	100,00	100,00	100,00
8 osvetlitev	100,00	100,00	100,00	100,00	100,00	8 osvetlitev	100,00	100,00	100,00	100,00	100,00
10 osvetlitev	100,00	100,00	100,00	100,00	100,00	10 osvetlitev	100,00	100,00	100,00	100,00	100,00
16 osvetlitev	100,00	100,00	100,00	100,00	100,00	16 osvetlitev	100,00	100,00	100,00	100,00	100,00
32 osvetlitev	100,00	100,00	100,00	100,00	100,00	32 osvetlitev	100,00	100,00	100,00	100,00	100,00

Slika 5: Rezultati ablacijske študije iskanja poškodb na grlu vial s pomočjo konvencionalnih metod

Odstotek pravilno zaznanih poškodb na trupu vial v primeru uporabe konvencionalnih metod						Odstotek pravilno zaznane odstotnosti poškodb na trupu vial v primeru uporabe konvencionalnih metod					
	3 orientacije	4 orientacije	5 orientacij	7 orientacij	13 orientacij		3 orientacije	4 orientacije	5 orientacij	7 orientacij	13 orientacij
1 osvetlitev	90,00	90,00	90,00	100,00	100,00	1 osvetlitev	80,00	80,00	90,00	90,00	100,00
2 osvetlitvi	90,00	90,00	100,00	100,00	100,00	2 osvetlitvi	90,00	90,00	90,00	90,00	100,00
3 osvetlitve	100,00	100,00	100,00	100,00	100,00	3 osvetlitve	90,00	90,00	100,00	100,00	100,00
4 osvetlitve	100,00	100,00	100,00	100,00	100,00	4 osvetlitve	100,00	100,00	100,00	100,00	100,00
5 osvetlitev	100,00	100,00	100,00	100,00	100,00	5 osvetlitev	100,00	100,00	100,00	100,00	100,00
6 osvetlitev	100,00	100,00	100,00	100,00	100,00	6 osvetlitev	100,00	100,00	100,00	100,00	100,00
7 osvetlitev	100,00	100,00	100,00	100,00	100,00	7 osvetlitev	100,00	100,00	100,00	100,00	100,00
8 osvetlitev	100,00	100,00	100,00	100,00	100,00	8 osvetlitev	100,00	100,00	100,00	100,00	100,00
10 osvetlitev	100,00	100,00	100,00	100,00	100,00	10 osvetlitev	100,00	100,00	100,00	100,00	100,00
16 osvetlitev	100,00	100,00	100,00	100,00	100,00	16 osvetlitev	100,00	100,00	100,00	100,00	100,00
32 osvetlitev	100,00	100,00	100,00	100,00	100,00	32 osvetlitev	100,00	100,00	100,00	100,00	100,00

Slika 6: Rezultati ablacijske študije iskanja poškodb na trupu vial s pomočjo konvencionalnih metod

Odstotek pravilno zaznanih poškodb na grlu vial v primeru uporabe globokih konvolucijskih mrež						Odstotek pravilno zaznane odstotnosti poškodb na grlu vial v primeru uporabe globokih konvolucijskih mrež					
	3 orientacije	4 orientacije	5 orientacij	7 orientacij	13 orientacij		3 orientacije	4 orientacije	5 orientacij	7 orientacij	13 orientacij
1 osvetlitev	100,00	100,00	100,00	100,00	100,00	1 osvetlitev	100,00	100,00	100,00	100,00	100,00
2 osvetlitvi	100,00	100,00	100,00	100,00	100,00	2 osvetlitvi	100,00	100,00	100,00	100,00	100,00
3 osvetlitve	100,00	100,00	100,00	100,00	100,00	3 osvetlitve	100,00	100,00	100,00	100,00	100,00
4 osvetlitve	100,00	100,00	100,00	100,00	100,00	4 osvetlitve	100,00	100,00	100,00	100,00	100,00
5 osvetlitev	100,00	100,00	100,00	100,00	100,00	5 osvetlitev	100,00	100,00	100,00	100,00	100,00
6 osvetlitev	100,00	100,00	100,00	100,00	100,00	6 osvetlitev	100,00	100,00	100,00	100,00	100,00
7 osvetlitev	100,00	100,00	100,00	100,00	100,00	7 osvetlitev	100,00	100,00	100,00	100,00	100,00
8 osvetlitev	100,00	100,00	100,00	100,00	100,00	8 osvetlitev	100,00	100,00	100,00	100,00	100,00
10 osvetlitev	100,00	100,00	100,00	100,00	100,00	10 osvetlitev	100,00	100,00	100,00	100,00	100,00
16 osvetlitev	100,00	100,00	100,00	100,00	100,00	16 osvetlitev	100,00	100,00	100,00	100,00	100,00
32 osvetlitev	100,00	100,00	100,00	100,00	100,00	32 osvetlitev	100,00	100,00	100,00	100,00	100,00

Slika 7: Rezultati ablacijske študije iskanja poškodb na grlu vial s pomočjo globokih konvolucijskih mrež

Odstotek pravilno zaznanih poškodb na trupu vial v primeru uporabe globokih konvolucijskih mrež						Odstotek pravilno zaznane odstotnosti poškodb na trupu vial v primeru uporabe globokih konvolucijskih mrež					
	3 orientacije	4 orientacije	5 orientacij	7 orientacij	13 orientacij		3 orientacije	4 orientacije	5 orientacij	7 orientacij	13 orientacij
1 osvetlitev	85,71	85,71	100,00	100,00	100,00	1 osvetlitev	100,00	100,00	100,00	100,00	100,00
2 osvetlitvi	85,71	85,71	100,00	100,00	100,00	2 osvetlitvi	100,00	100,00	100,00	100,00	100,00
3 osvetlitve	100,00	100,00	100,00	100,00	100,00	3 osvetlitve	100,00	100,00	100,00	100,00	100,00
4 osvetlitve	100,00	100,00	100,00	100,00	100,00	4 osvetlitve	100,00	100,00	100,00	100,00	100,00
5 osvetlitev	100,00	100,00	100,00	100,00	100,00	5 osvetlitev	100,00	100,00	100,00	100,00	100,00
6 osvetlitev	100,00	100,00	100,00	100,00	100,00	6 osvetlitev	100,00	100,00	100,00	100,00	100,00
7 osvetlitev	100,00	100,00	100,00	100,00	100,00	7 osvetlitev	100,00	100,00	100,00	100,00	100,00
8 osvetlitev	100,00	100,00	100,00	100,00	100,00	8 osvetlitev	100,00	100,00	100,00	100,00	100,00
10 osvetlitev	100,00	100,00	100,00	100,00	100,00	10 osvetlitev	100,00	100,00	100,00	100,00	100,00
16 osvetlitev	100,00	100,00	100,00	100,00	100,00	16 osvetlitev	100,00	100,00	100,00	100,00	100,00
32 osvetlitev	100,00	100,00	100,00	100,00	100,00	32 osvetlitev	100,00	100,00	100,00	100,00	100,00

Slika 8: Rezultati ablacijske študije iskanja poškodb na trupu vial s pomočjo globokih konvolucijskih mrež

Ablacijska študija je pokazala, da so rezultati razvrščanja bistveno bolj občutljivi na osvetlitev kot na število orientacij vial. Posledično zajem zbirke lahko optimiziramo z zajemom minimalnega potrebnega števila slik ene vial. Pri ustvarjanju zbirke vzorcev lahko obdržimo le toliko orientacij, kolikor jih je potrebno, da s kamero zajamemo celoten plašč vial brez prekrivanja. Fizično rotiranje vial lahko nadomestimo tudi z večjim številom kamer, ki slike zajemajo iz različnih kotov. Tudi število osvetlitev se lahko močno zmanjša, vendar mora biti ta optimizacija premišljena. Po drugi strani pa je sprememba osvetlitve pri uporabi programirljivih LED panelov zelo hitra ter ne zahteva nobene mehanske operacije. Zato je načeloma možno z dobro sinhronizacijo kamere in krmiljenja osvetlitve zajeti različno osvetljene slike v zelo kratkem času.

6 ZAKLJUČEK

Iz priloženih rezultatov lahko zaključimo, da sta obe metodi precej učinkoviti pri iskanju poškodb na steklu, saj sta dosegli 100% razpoznavnost poškodovanih in nepoškodovanih vial. Z ablacijsko študijo pa smo pokazali, da se kljub zmanjševanju podatkov o eni viali rezultati ne spreminjajo do nekega kritičnega števila podatkov, kjer točnost razvrščanja začne upadati. Ta upada hitreje pri konvencionalnih metodah kot pri uporabi nevronske mreže.

Razliko med rezultati konvencionalnih metod in rezultati konvolucijske nevronske mreže v primeru manjšega števila podatkov si lahko razlagamo z analizo filtrov obeh metod. Predlagana konvencionalna metoda ima preproste filtre, ki se izrazito odzivajo na določene oblike linij z določenimi smermi, vendar ti niso sposobni ločevati kompleksnejših tekstur. V primeru globokih mrež imamo bistveno večje število filtrov, ki pa se spreminjajo skozi plasti mreže. V začetnih plasteh uporabljene konvolucijske mreže lahko opazujemo enostavnejše filtre, ki se odzivajo na različno usmerjene linije na slikah, ločujejo pa tudi barve slikovnih točk (v našem primeru odtenke sive). V višjih plasteh se značilke kombinirajo, zato so filtri bolj kompleksni in se v najvišjih plasteh naučijo odzivati na texture, ki so specifične za objekte v učni zbirki.

Literatura

1. D. A. Dean, E. R. Evans, I. H. Hall (2005), *Pharmaceutical packaging technology*, CRC Press.
2. L. Eberle, A. Svensson, A. Graser, J. Luemkemann, H. Sugiyama, R. Schmidt, K. Hungerbuehler (2017), Innovative approach for identifying root causes of glass defects in sterile drug product manufacturing, *European Journal of Pharmaceutical Sciences*, vol. 104, str. 162-170.
3. N. D. Surdyka, C. G. Pantano, S. H. Kim (2014), Environmental effects on initiation and propagation of surface defects on silicate glasses: scratch and fracture toughness study, *Applied Physics A*, vol. 116, str. 519-528.
4. D. Gabor (1946), Theory of communication. part 1: The analysis of information, *Journal of the Institution of Electrical Engineers-Part III: Radio and Communication Engineering*, vol. 93, no. 26, str. 429-441.
5. J. G. Daugman (1985), Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters, *JOSA A*, vol. 2, str. 1160-1169.
6. G. Strang, T. Nguyen (1996), *Wavelets and filter banks*. SIAM

7. C. Cortes, V. Vapnik (1995), Support-vector networks, Machine learning, vol. 20, str. 273-297.
8. K. Simonyan, A. Zisserman (2014), Very deep convolutional networks for largescale image recognition, arXiv preprint arXiv:1409.1556
9. T. Fawcett (2006), An introduction to roc analysis, Pattern recognition letters, vol. 27, str. 861-874.

Poravnava 3D modelov objektov na realne 3D oblake točk

MARTIN PERNUŠ, SIMON DOBRIŠEK IN JANEZ PERŠ

Povzetek Predstavljena je problematika poravnavanja trodimenzionalnih modelov objektov na trodimenzionalne oblake točk, ki so bili zajeti z realnimi (neidealnimi) 3D senzorji. Ker gre za šestdimenzionalen optimizacijski problem, je zelo verjetno, da se bo poravnava ustavila v lokalnem minimumu in bo končni rezultat neuporaben. Predlagano delo temelji na metodi Drosta in soavtorjev, predstavljeni na CVPR 2010, ki jo najdemo v knjižnici OpenCV ter v komercialni programski opremi za strojni vid. Po naših izkušnjah je samo na podlagi dostopne dokumentacije nemogoče doseči dobro delovanje na realističnih podatkih, zato v tem članku na kratko predstavimo metodo, predvsem pa damo praktične napotke, kako pripraviti podatke ter nastaviti parametre, da bo metoda dobro delovala in jo bo mogoče uporabiti tudi v industrijski robotiki.

Ključne besede: • oblak točk • 3D točke • lokalizacija objektov
• nevronske mreže • globalno prilaganje • fino prilaganje •

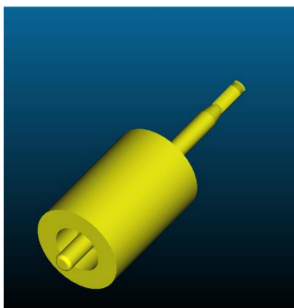
NASLOVI AVTORJEV: Martin Pernuš, mag. inž. el., asistent, Univerza v Ljubljani, Fakulteta za elektrotehniko Tržaška cesta 25, 1000 Ljubljana, Slovenija, e-mail: martin.pernus@fe.uni-lj.si. dr. Simon Dobrišek, izredni profesor, Univerza v Ljubljani, Fakulteta za elektrotehniko Tržaška cesta 25, 1000 Ljubljana, Slovenija, e-mail: simon.dobrissek@fe.uni-lj.si. dr. Janez Perš, docent, Univerza v Ljubljani, Fakulteta za elektrotehniko Tržaška cesta 25, 1000 Ljubljana, Slovenija, e-mail: janez.pers@fe.uni-lj.si.

1 UVOD

V industriji se pojavlja vse več senzorjev, ki so sposobni zajema 3D oblaka točk nekega področja. Ob nakupu takega sensorja je ponavadi dana še možnost nakupa pripadajočega programskega okolja, ki vam na 3D oblaku točk poišče želeni objekt. Za iskanje objekta ponavadi okolje potrebuje le kak CAD model izdelka. Poleg komercialnih rešitev obstajajo tudi odprtokodne. Ob uporabi ali komercialnih ali odprtokodnih rešitev pa se zaradi slabe dokumentacije pogosto dogodi, da jih moramo obravnavati kot črne škatle, zaradi česar so rezultati pogosto slabi oziroma pridobitev dobrih rezultatov traja predolgo za praktične namene.

V tem članku na kratko opišemo delovanje algoritma za iskanje objekta v oblaku točk z uporabo algoritma, ki temelji na delu Drosta in soavtorjev [1]. Drostov algoritem je sposoben globalne grobe lokalizacije objektov, kateremu ponavadi dodamo še fino prilagajanje s katero od standardnih metod. Njihov algoritem je, po naših izkušnjah, deloval tudi najboljše v primerjavi z nekaj konkurenčnimi algoritmi. Na tem mestu velja omeniti, da je algoritem patentiran [2], sicer pa implementiran tudi v komercialni programski opremi HALCON [3]. Za raziskovalno uporabo vseeno lahko uporabimo odprtokodno različico algoritma, ki je dostopna preko knjižnice OpenCV [4], ki zadošča za raziskovalne namene. Oba postopka sta zapakirana v programsko opremo, ki direktno ne dopušča *predhodne obdelave* oblakov točk, *čeprav je ravno ta poglavitnega pomena za kvaliteto in hitro lokalizacijo*.

V nadaljevanju bomo opisali teoretičen opis postopka in napotke za uporabo algoritma, praktično prikazane na primeru objekta iz proizvodnje podjetja Kolektor Vision [5]. Objekt, ki ga bomo obravnavali, je prikazan na sliki 1. Proizvod se proizvaja v velikih količinah, pri čemer obstaja želja po popolni avtomatizaciji kontrole kvalitete.



Slika 1: Obravnavani objekt, ki ga želimo poiskati na zajetih oblakih točk.

2 DELOVANJE ALGORITMA

Algoritem je sestavljen iz dveh faz, ki jih bomo poimenovali učna faza in sprotna faza. Učna faza temelji na primerni predstavitvi oblaka točk našega modela, sprotna faza pa temelji na lokalizaciji tega modela v zajeti sceni. Sprotno fazo lahko razdelimo na globalno prileganje, ki najde približno lokacijo iskanega modela, in fino prileganje, ki približno lokacijo dodatno izboljša. Pri vsakem sklopu najprej na kratko opišemo teoretično ozadje, nato pa še podamo praktične napotke.

2.1 Učna faza

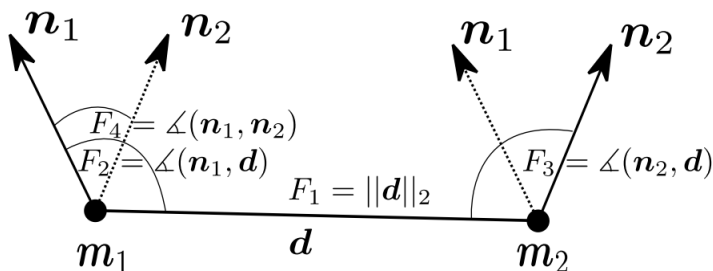
2.1.1 Teorija

Za učno fazo potrebujemo model v obliki oblaka točk s pripadajočimi vektorji normal za vsako točko. Označimo točko modela $\mathbf{m} = (x, y, z)$, $\mathbf{m} \in \mathbf{M}$, kjer je \mathbf{M} množica vseh točk. Vsaka točka ima pripadajoč vektor normale \mathbf{n} . V eni iteraciji izberemo nek par \mathbf{m}_1 in \mathbf{m}_2 , glede na katerega izračunamo štiridimenzionalen vektor značilik, ki ga sestavljajo naslednje značilke:

- Razdalja med točkama $F_1 = \|\mathbf{d}\|_2 = \|\mathbf{m}_2 - \mathbf{m}_1\|_2$,
- Kot med normalo 1. točke in vektorjem razdalje med točkama $F_2 = \angle(\mathbf{n}_1, \mathbf{d})$,
- Kot med normalo 2. točke in vektorjem razdalje med točkama $F_3 = \angle(\mathbf{n}_2, \mathbf{d})$,
- Kot med normalama točk $F_4 = \angle(\mathbf{n}_1, \mathbf{n}_2)$.

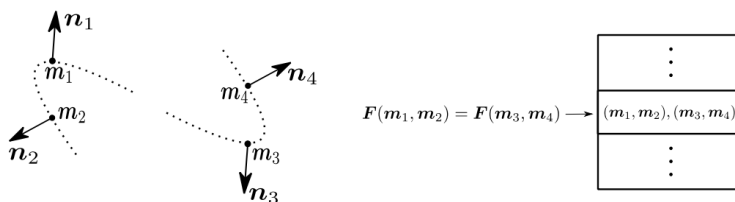
Pridobljene značilke so vizualno prikazane na sliki 2. Vektor značilk je tako predstavljen kot

$$F(m_1, m_2) = (F_1, F_2, F_3, F_4) = (||d||_2, \angle(n_1, d), \angle(n_2, d), \angle(n_1, n_2)). \quad (1)$$



Slika 2: Vizualizacija pomena posameznih značilk vektorja

Razdalja med točkama se diskretizira glede na parameter d_{dist} , koti pa glede na d_{angle} . S tem dobimo vektor značilk, ki ima le končno število možnih vrednosti. Na tem vektorju izračunamo funkcijo zgoščevanja (angl. *hash function*) in v pripadajoč ključ v zgoščevalni tabeli (angl. *hash table*) shranimo trenutni par točk. Postopek iteracije se ponovi za vsak možen par točk. Po končanem postopku imamo zgoščevalno tabelo s shranjenimi pari točk. Ti pari točk se bodo uporabili v sprotni fazi za iskanje korespondenc med pari točk na zajeti sceni in shranjenimi pari točk modela, pridobljenimi v učni fazi. Primer shranjevanja dveh parov točk, ki sta si med seboj podobni po značilkah, v isti ključ zgoščevalne tabele prikazuje slika 3.



Slika 3: Na levi sta prikazana dva para točk na modelu, (m_1, m_2) in (m_3, m_4) , ki imata podobno razdaljo in medsebojno orientacijo normal. Njun diskreditiran ključ $F(,)$ je torej enak, po izračunu zgoščevalne funkcije na ključu pa oba para shranimo v isti prostor zgoščevalne tabele (desno).

2.1.2 Praksa

Najprej se posvetimo najpomembnejšemu delu, to je pridobitev ustreznega učnega oblaka točk, ki vsebuje naš objekt. V industrijski rabi so predmeti, ki jih proizvajamo ponavadi v obliki različnih CAD modelov. Te lahko spremenimo v oblak točk z orodjem, kot je npr. odprtokodni CloudCompare [7]. CloudCompare ponuja grafični vmesnik, s katerim lahko obdelujemo oblake točk, na njih izvajamo izrezovanje, filtriranje in mnogo drugih operacij.

Bistveno je, da model sestavljajo čim bolj reprezentativen oblak točk, torej tak, kot se bo pojavil tudi na sceni, zajeti s 3D senzorjem. Zato točke v središču modela, torej take, ki s 3D senzorjem niso vidne, preslikamo na površino modela ali po potrebi celo odrežemo (odstranimo). Poskrbimo tudi, da se znebimo osamelcev. Na površinskih točkah nato izračunamo normalo za vsako točko, pazimo pa tudi, da so normale na sosednjih točkah orientirane v isto smer in ni preskokov za 180° . Z redčenjem oblaka še poskrbimo, da učenje ne bo predolgo trajalo. Ta postopek izvedemo samo enkrat, zato tu avtomatizacija ni bistvenega pomena.

Parameter d_{dist} nastavimo relativno glede na obseg modela. Po navadi je vrednost $d_{dist} = 0,03$ zadostna, Pretirano manjšanje parametra za večjo natančnost globalnega prileganja ni smiselno, saj kasneje izvajamo še fino prileganje. Parameter diskretizacije kota nastavljam preko $d_{angle} = \frac{2\pi}{n_{angle}}$, kjer je n_{angle} privzet na 30. Večanje tega parametra in posledično manjšanje d_{angle} zaradi finega prileganja ponovno nima veliko smisla, ob šumno zajeti sceni pa lahko n_{angle} zmanjšamo na npr. 25.

V učni fazi algoritem zahteva še stopnjo redčenja vhodnega modela. Priporočljivo je, da potrebno stopnjo redčenja vključimo že v pripravo modela in te opcije ne uporabljamo. Rado se namreč zgodi, da smo model ponevedoma tako razdredčili, da ga niti sami ne bi več prepoznali.

Naš obravnavani model smo s programom CloudCompare spremenili v oblak točk. Nadaljnje se izkaže, da je na zajeti sceni vedno vidna le polovica modela. Spodnjo polovico modela smo programsko odstranili, preostali del pa s preslikanjem na površino preoblikovali tako, da je vseboval le površinske točke.

Izračun vektorjev normal in pravilno orientacijo je ponovno omogočal CloudCompare. Nastali model je prikazan na sliki 4.

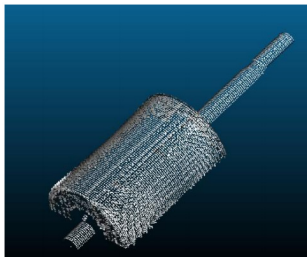
2.2 Sprotna faza

2.2.1 Teorija

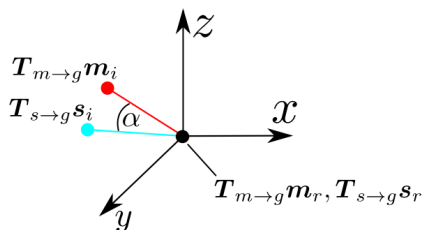
Za sprotno fazo potrebujemo že naučen model v obliki zgoščevalne tabele, ki vsebuje pare točk. Obravnavamo 3D prizor, kjer se nahaja en ali več iskanih objektov. Za ta namen podobno kot pri učni fazi obravnavamo vse pare točk in računanje štiridimenzionalnih vektorjev značilnk. Obravnavajmo sedaj trenutno referenčno točko \mathbf{s}_r . Predvidevamo, da ta točka leži tudi na modelu $\mathbf{m} \in \mathcal{M}$, ki smo ga uporabili v učni fazi. a njeno trenutno parno točko \mathbf{s}_i izračunamo vektor značil $\mathbf{F}(\mathbf{s}_r, \mathbf{s}_i)$. V zgoščevalni tabeli pridobimo sorodne točke modela \mathbf{m}_r in \mathbf{m}_i . Če točki \mathbf{s}_r in \mathbf{m}_r transformiramo v izhodišče koordinatnega sistema tako, da sta njuni normali poravnani v x osi, potem pa moramo poznati le še kot α okoli x osi, za kolikor moramo rotirani točko \mathbf{s}_i , da poravna z \mathbf{m}_i . S tem dobimo vse potrebne informacije za izračun transformacijske matrike. Transformacijo iz shranjene točke \mathbf{m}_i v trenutno obravnavano točko \mathbf{s}_i lahko torej zapišemo kot

$$\mathbf{s}_i = \mathbf{T}\mathbf{m}_i = \mathbf{T}_{\{s \rightarrow g\}}^{-1} \mathbf{R}_x(\alpha) \mathbf{T}_{\{m \rightarrow g\}} \mathbf{m}_i, \quad (2)$$

Kjer $\mathbf{T}_{\{s \rightarrow g\}}$ označuje transformacijsko matriko, ki točko \mathbf{s}_r reslika v izhodišče koordinatnega sistema, njeno normalo \mathbf{n}_r pa zarotira v smer osi x . $\mathbf{T}_{\{m \rightarrow g\}}$ naredi podobno za točko \mathbf{m}_r . Postopek je grafično prikazan na sliki 5.



Slika 4: Obdelan model, primeren za učno fazo.



Slika 5: Prikaz izračuna transformacije med paroma točk (s_r, s_i) . Točki s_r in m preslikamo v koordinatno izhodišče tako, da se njuni normali ujemata z x osjo. Nato izračunamo še potreben kot α za ujevanje točk s_i in m_i . S tem dobimo vse potrebne informacije za izračun transformacijske matrike T med dvema paroma točk.

S tem, ko iteriramo čez vse točke s_j pridobivamo torej točke m_i in kote α . Kote α podobno, kot v učni fazi, diskretiziramo s parametrom n_{angle} . Točka m_i in diskretiziran kot α štejejo kot glas v akumulatorju velikost $|M| \times n_{angle}$, kjer je $|M|$ število vseh točk, kot je prikazano na sliki 6. Postopek je soroden polnjenju akumulatorja pri Houghovi transformaciji. Po procesiranju vseh točk dobimo napolnjen akumulator za trenutno referenčno točko s_r . Transformacijsko matriko izračunamo le za najbolj napolnjen predel akumulatorja. Po obravnavi vseh referenčnih točk podobne transformacijske matrike združimo, njihove glasove pa seštejemo glede na podobnost translacije in rotacije med matrikami. Transformacijske matrike na koncu še uredimo po pripadajočemu številu glasov.

	α_1	α_i	$\alpha_{n_{angle}}$
m_1			
m_i		+1	
$m_{ M }$			

Slika 6: Glede na par točk (s_r, s_i) v zgoščevalni tabeli poiščemo podobne pare točk (m_r, m_i) . Po opisanem postopku izračunamo še kot α in pripadajočemu predelu akumulatorja prištejemo glas Na sliki prikazan akumulator, ki mu glede na trenutni par točk (s_r, s_i) poiščemo soroden par v zgoščevalni matriki (m_r, m_i) , nato pa izračunamo še kot α_i . Predel akumulatorja, ki ustreza m_r, α_i povečamo za ena.

Model transformiramo glede na nekaj najboljših transformacijskih matrik. S tem dobimo grobo oceno lokacije, ki je odvisna od parametrov d_{dist} , n_{angel} in kriterijev za združevanje transformacijskih matrik. To lokacijo popravimo s finim prileganjem z uporabo klasičnega algoritma iterativne najbližje točke (angl. *iterative closest point*).

2.2.2 Praksa

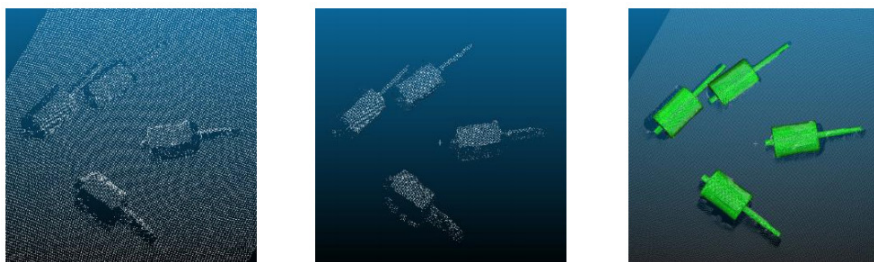
Znova se posvetimo najpomembnejšemu delu – pridobitev kvalitetnega oblaka točk scene. V nasprotju s pripravo modela, se tu ne moremo poslužiti grafičnega vmesnika, kot je CloudCompare, saj za industrijsko uporabo želimo avtomatiziran postopek. Tu smo sami uporabili odprtokodno knjižnico Point Cloud Library (PCL) [6] za obdelavo oblakov točk.

Globalno prileganje najbolj zanesljivo deluje s čim več reprezentativnimi točkami. Vendar pa pri prevelikem številu točk kaj kmalu naletimo na omejitve zaradi računske kompleksnosti. Oblak točk lahko močno zmanjšamo z uporabo predhodnega znanja. Z uporabo knjižnice PCL lahko glede na predhodno znanje odrežemo željeno območje, osamelcev se znebimo s filtriranjem točk glede na število in oddaljenost sosednjih točk, neuporabnih ravnin pa se znebimo z algoritmom RANSAC (pri čemer iz prizora odrežemo točke, ki ležijo v določenem tolerančnem pasu od ravnine). Dodatno nam knjižnica PCL omogoča tudi računanje vektorjev normal na zajetem oblaku točk. Pri računanju le-teh pazimo na njihovo orientacijo, gledati morajo iz smeri objekta ven!

Globalno prileganje je računsko bolj zahteven postopek kot fino prileganje. Zato ponavadi za grobo prileganje uporabimo manj točk, kot pri finem prileganju, saj potrebujemo le približno lokacijo modela. Pri finem prileganju nam več točk koristi, saj je zanesljivost končne dobljene pozicije s tem natančnejša. Funkcije za prileganje tako ponavadi zahtevajo stopnjo redčenja zajete scene, ki se uporabi za fino prileganje, nato pa še dodatno redčenje, ki pa se uporabi za globalno prileganje. Kot pri pripravi modela, se tudi tu spleča redčenje izvesti v koraku predobdelave za kar lahko prav tako uporabimo knjižnico PCL.

Po pridobitvi transformacijskih matrik se moramo še odločiti, katera je »najboljša« in katere so sploh tiste transformacijske matrike, ki dejansko ustrezajo objektu na sceni. Pri našem testiranju je najboljše rezultate dajala kriterijska funkcija, ki pove, kakšen delež točk transformiranega modela se pokriva s točkami prizora, znotraj izbrane tolerance.

Primer prizora, zajetega s 3D senzorjem s strani podjetja Kolektor Vision je predstavljen na sliki 7 levo. Točke, ki pripadajo ravnini, identificiramo s postopkom RANSAC in jih odstranimo, kar opazimo na sredinski sliki. Na novo pridobljenem oblaku točk izvedemo postopek globalnega in finega prilaganja. Rezultati prilaganja za najboljše tri rešitve glede na zgoraj opisano kriterijsko funkcijo so prikazani na desni sliki (obarvana zeleno), skupaj s prvotnim oblakom točk (bele barve).



Slika 7: Oblak točk, zajet s 3D senzorjem (levo), odstranjevanje ravnine (sredina) in lokalizacija objekta (desno).

3 PRIMERJAVA IZVEDBE ALGORITMA V KOMERCIALNI IN ODPRTOKODNI RAZLIČICI

Omenimo naj še nekaj razlik med različico algoritma iz OpenCV in komercialno implementirano različico v orodju HALCON. Slednja ima pred odprtokodno različico v OpenCV nekaj prednosti:

- omogoča shranjevanje naučenega modela, medtem ko OpenCV tega nima implementiranega,
- v verziji OpenCV včasih pride do napačnih rezultatov, saj je v kodi hrošč, kjer uporabljeni kvaternioni niso normirani,

- kriterijska funkcija deleža bližnjih točk modela v OpenCV ni implementirana in jo moramo sprogramirati sami,
- Ob zajetem prizoru zna HALCON izračunati normale, medtem ko različica algoritma v OpenCV pričakuje že izračunane normale.

Omeniti velja še, da obe verziji omogočata vzporedno izvajanje algoritma, tako da ju lahko pohitrimo z nakupom boljše strojne opreme.

4 ZAKLJUČEK

V članku smo predstavili napotke za uporabo Drostovega algoritma za lokalizacijo objektov v oblakih točk. Kratkim teoretičnim razlagam ozadja delovanja so sledili praktični nasveti nastavljanja parametrov in potrebnega predprocesiranja za čim boljše delovanje. Razlago smo podprli s primerom na podatkih, pridobljeni s strani podjetja Kolektor Vision.

Literatura

1. B. Drost, et al. (2010), Model globally, match locally: Efficient and robust 3D object recognition, *Computer Vision and Pattern Recognition*, str. 998-1005.
2. B. Drost, et al. (2014) »Recognition and pose determination of 3D objects in 3D scenes.«, *U.S. Patent No. 8,830,229*, Washington, DC: U.S. Patent and Trademark Office
3. HALCON – *The power of machine vision*. Pridobljeno na: <https://www.mvtec.com/products/halcon/>
4. OpenCV – *surface_matching. Surface Matching*. Pridobljeno na: https://docs.opencv.org/3.0-beta/modules/surface_matching/doc/surface_matching.html
5. Kolektor. Pridobljeno na: <http://www.kolektorvision.com/>
6. PCL. *Documentation*. Pridobljeno na: <http://www.pointclouds.org/documentation/>
7. *CloudCompare. 3D point cloud and mesh processing software Open Source Project*. Pridobljeno na: <https://www.danielgm.net/cc/>

Učenje igranja računalniških iger z globoko obdelavo slik in drevesnim preiskovanjem Monte Carlo

MATEJ VITEK IN PETER PEER

Povzetek Področje globokega učenja je v zadnjem desetletju doživelo precejšen razcvet. Uporablja se za reševanje premnogih problemov, v zadnjih petih letih pa precej tudi za igranje iger. Dva pomembna dosežka sta bila globoke Q-mreže (DQN) in AlphaZero. DQN se je naučila igrati klasične igre za Atari 2600 (Pong, Space Invaders, itd.), AlphaZero pa se je s samo-igranjem naučil igrati šah, šogi in Go. Mi smo na temelju AlphaZero poskusili zgraditi agenta FighterZero, ki bi se prav tako s samo-igranjem naučil igrati pretepaške računalniške igre. Rezultati so bili manj uspešni, kot smo pričakovali, saj se je časovna zahtevnost izkazala za nepremagljivo oviro.

Ključne besede: • umetna inteligenca • igre • samo-igranje • globoko učenje • drevesno preiskovanje Monte Carlo •

NASLOVA AVTORJEV: Matej Vitek, Univerza v Ljubljani, Fakulteta za računalništvo in informatiko, Večna Pot 113, 1000 Ljubljana, Slovenija, e-pošta: matej.vitek@fri.uni-lj.si. dr. Peter Peer, izredni profesor, Univerza v Ljubljani, Fakulteta za računalništvo in informatiko, Večna Pot 113, 1000 Ljubljana, Slovenija, e-pošta: peter.peer@fri.uni-lj.si.

1 UVOD

Umetna inteligenca je področje, ki se je začelo razvijati v petdesetih letih prejšnjega stoletja, uporablja pa se v zelo raznolike namene. Tako pristope umetne inteligence najdemo v medicini, v gospodarstvu, v robotiki, v matematiki, v biometriji, . . .

V zadnjem desetletju pa se je precej razširilo predvsem področje globokega učenja. K temu so največ pripomogli precejšnji napredki v hitrosti in sposobnosti strojne opreme, ki je na voljo. Algoritmi globokega učenja so sicer, vsaj v osnovni obliki, obstajali že pred desetletji, vendar pa s takratno strojno opremo niso bili zmožni doseči dobrih rezultatov na področjih, kjer se uporabljajo danes. Globoko učenje v zadnjih letih na mnogih problemih dosega boljše dosežke kot katerikoli drugi pristopi. Uporablja se za prepoznavo govora [1; 2; 3], računalniški vid [4], procesiranje naravnega jezika [5; 6; 7], v priporočilnih algoritmih [8], v bioinformatiki [9], . . .

V tem delu razvijemo agenta, ki se sam nauči igrati pretepaško igro. Naš agent temelji na enem od izrednih dosežkov umetne inteligence v zadnjem desetletju: DeepMindovem agentu AlphaZero, ki se je sam naučil igrati šah, šogi in Go ter v vseh treh igrah premagal dosedanje najboljše računalniške agente razvite za vsako od teh treh iger posebej. Tako je AlphaZero pri vseh treh igrah dosegel nadčloveški nivo igranja.

2 PREGLED PODROČJA

Uporaba umetne inteligence za igranje računalniških iger je v zadnjih dveh desetletjih eno širših področij umetne inteligence tako v industriji kot tudi v akademskih raziskavah. V zadnjih petih letih pa se je (predvsem na akademski strani) začelo v ta namen uporabljati tudi globoko spodbujevalno učenje, ki je v umetni inteligenci relativno mlado področje.

Prva uspešna implementacija globokega spodbujevalnega učenja za igranje iger je prišla leta 2013, ko je Googlova ekipa DeepMind globoko Q-mrežo (DQN) naučila igrati klasične igre za Atari 2600 [10; 11].

DQN sicer temelji na še enem pomembnem dosežku. Že leta 1995 je G. Tesauro [12] uspešno uporabil kombinacijo spodbujevalnega učenja in nevronske mreže za implementacijo agenta za igranje igre backgammon. Ta dosežek pa je za nas pomemben tudi, ker je uporabljal pristop samo-igranja, ki ga za učenje uporablja (med drugimi) tudi naš algoritem.

Leta 2014 je nato ekipa z michiganske univerze nadgradila DQN s tremi pristopi drevesnega preiskovanja Monte Carlo (MCTS), ki so pomagali pri učenju mreže [13]. MCTS se v njihovem pristopu uporablja le za učenje, dejansko igranje pa potem izvaja mreža sama. Ta pristop je opazno izboljšal rezultate DQN.

Januarja 2016 je ekipa DeepMind presenetila svet s svojim agentom AlphaGo [14], ki je v igri Go premagal evropskega prvaka Fana Huija z rezultatom 5-0 in kasneje še zmagovalca 18 mednarodnih naslovov Leeja Sedola z rezultatom 4-1 ter še nekoliko kasneje takrat najmočnejše človeške igralce z rezultatom 60-0 [15]. Podobno kot v [13] je tudi v AlphaGo globoko spodbujevalno učenje združeno z MCTS. Vendar pa se v AlphaGo MCTS uporablja tudi pri samem igranju, ne le pri učenju mreže. Za učenje mreže so uporabili nadzorovano učenje na primerih potez strokovnjakov iz preteklih iger in spodbujevalno učenje s samo-igranjem.

Leta 2017 je DeepMind nato AlphaGo nadgradil v AlphaGo Zero [15], ki je vse prejšnje verzije agenta AlphaGo premagal prepričljivo. Glavna razlika med agentoma je, da se je AlphaGo Zero igro Go naučil igrati povsem brez kakršnegakoli predznanja (razen pravil igre). Učenje je torej v celoti realizirano s spodbujevalnim učenjem s samo-igranjem brez uporabe preteklih iger človeških strokovnjakov. Nekaj implementacijskih razlik je tudi v MCTS, ki je v AlphaGo Zero nekoliko poenostavljen.

Proti koncu leta 2017 pa je DeepMind svoj algoritem posplošil še na igri šah in šogi in predstavil splošnega agenta AlphaZero [16; 17]. Ta je v vsaki od teh treh iger premagal do tedaj najboljše računalniške igralce. V AlphaZero je dodana možnost izenačenih izidov (in bolj splošno kakršnihkoli izidov, ki omogočajo točkovanje). Poleg tega se AlphaZero ne zanaša več na invariantnost igre na

rotacije in zrcaljenje. V igri Go taka invariantnost velja, v šahu in šogiju (in bolj splošno v igrah) pa ne.

Poleg dosežkov DeepMinda pa se je tudi v splošnem v zadnjih treh letih precej razcvetela uporaba globokega učenja za igranje iger. Kar nekaj del na to temo uporablja podobne pristope, kot jih je uporabila ekipa DeepMind, nekateri pa poskušajo tudi z drugačnimi pristopi globokega učenja. Tako najdemo npr. uporabo DQN za igranje igre Pac-Man [18], iger za Atari 2600 [19], igre FlappyBird [20] in Tetrisa [21]. Poskus nadgradnje AlphaGo najdemo v [22] in v [23]. HearthBot [24] uporablja prilagodljive nevronske mreže za igranje spletnih iger s kartami Hearthstone. Prav tako v literaturi najdemo uporabo globokega učenja za igranje 3D dirkalnih iger [25]. Kakršnokoli 3D okolje predstavlja nekoliko težji problem, saj dobimo iz okolja več dimenzij podatkov. Za igranje 3D prvoosebne streljaške igre zato zaenkrat obstaja le nekaj delnih agentov, ki se posvetijo posameznim problemom, ki jih agent v takem okolju sreča, ne pa celotni igri [26; 27]. Za še nekoliko težje področje pa se izkažejo realno-časovne strateške igre. Za realno-časovno igro Starcraft najdemo dve deli, ki naredita nekaj korakov proti razvoju celotnega agenta z globokim učenjem [28; 29].

Najbližja našemu delu pa je uporaba DQN za igranje pretepaških iger [30]. Vendar niti to delo niti kakšno od zgoraj naštetih (z izjemo nadgradenj AlphaGo [23; 22]) ne uporablja pristopa samo-igranja, ki ga je ekipa DeepMind uporabila v svojih agentih Alpha [14; 15; 16]. Uporaba le-tega je torej na področju pretepaških iger novost, ki je, kolikor nam je znano, v literaturi ne najdemo.

3 UPORABLJENO OGRODJE IN METODE

3.1 Ogrodje FightingICE

FightingICE [31] je pretepaška igra, ki so jo razvili v Intelligent Computer Entertainment (ICE) Lab., Ritsumeikan University. Razvita je bila v namen uporabe v raziskavah in na tekmovanju Fighting Game AI Competition (FTGAIC), ki ga prav tako organizira ta laboratorij.

Igralec igra enega od treh likov, ki so v igri na voljo: Zen, Garnet in Lud. Na sliki 1 vidimo, kako igra izgleda, več informacij o poteku igre, likih in ogrođu samem pa je na voljo na spletni strani tekmovanja [31]. Za potrebe našega agenta smo se osredotočili le na en lik (Zen), delo pa bi se sicer trivialno posplošilo tudi na preostala dva lika.



Slika 1: Igra FightingICE.

3.2 Simulator stanja in predprocesiranje slik

Ogrodje sicer omogoča pridobivanje slikovne informacije o trenutnem stanju, vendar pa enake funkcionalnosti nima implementirane za simulirana prihodnja stanja. Simulator pa nujno potrebujemo v drevesnem preiskovanju Monte Carlo. Zato smo implementirali približek slikovnega simulatorja, ki je iz slike trenutnega stanja, podatkov o trenutnem stanju in podatkov o simuliranem stanju (pridobljenih iz vgrajenega simulatorja) naredil približek slike v simuliranem stanju. To je izvedel tako, da je popravil pravokotnike, ki predstavljajo HP in energiji likov, nato pa lika preslikal na pravi novi poziciji (in ju pobrisal s stare).

Delovanje našega približka simulatorja vidimo na sliki 2. Simulator deluje na slikovni informaciji, ki jo dobimo iz ogrođa in jo pomanjšamo na velikost 96×64 ter pretvorimo v črno-belo. Poleg tega uporabimo sivo ozadje in za nasprotnikov lik uporabimo barvni negativ (da lahko razlikujemo lika med sabo). Kot vidimo na sliki, simulator ni popoln, saj ne zna pravilno popraviti

položaja rok in nog. Za implementacijo tega bi potrebovali možnost simulacije dejanske slikovne izvedbe akcij, tega pa ogrodje ne omogoča.

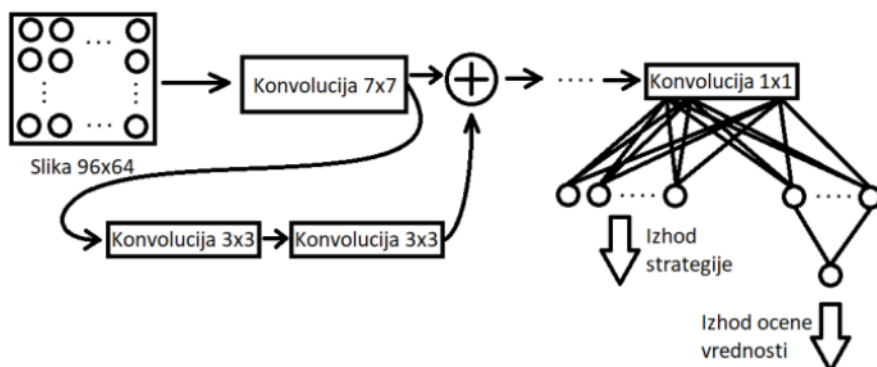


Slika 2: Delovanje našega simulatorja. Levo vidimo sliko trenutnega stanja (kot jo prejme simulator), desno pa simulirano sliko prihodnjega stanja.

3.3 Nevronske mreže

Nevronske mreže so eno najbolj razširjenih področij strojnega učenja. Za obdelavo slik se v zadnjih letih največ uporabljajo konvolucijske mreže (CNN). Za bolj podroben pregled globokih mrež in globokega učenja nasploh priporočamo [32]. Nevronske mreže se večinoma uporabljajo za nadzorovano učenje, kjer se učijo iz že znanih, označenih primerov. Mi pa svojo nevronske mrežo učimo s spodbujevalnim učenjem.

Naša nevronska mreža dobi na vhod sliko trenutnega stanja v velikosti 96×64 . Nato sledi začetna konvolucijska plast s konvolucijskim jedrom velikosti 7×7 in 4 plasti preostankov. Plasti preostankov so povzete iz arhitekture ResNet [33] in so sestavljene iz 2 konvolucijskih plasti z jedrom 3×3 . Mreža ima dva izhoda: evaluacijo trenutnega stanja (ali stanje vodi v zmago ali poraz) in strategijo (funkcijo izbire akcije lika). Kot izhod strategije imamo še konvolucijsko plast s konvolucijskim jedrom velikosti 1×1 in izhodno plast velikosti 41 (toliko je vseh možnih akcij), kot izhod evaluacije stanja pa imamo prav tako konvolucijsko plast z jedrom 1×1 , nato še eno polno povezano plast velikosti 20 in na koncu izhodno plast velikosti 1. Po posameznih plasteh se v mreži vedno izvede še normalizacija in aktivacija nevronov, za preprečevanje prekomernega prileganja pa uporabljamo tudi odpadne plasti [34]. Celotna arhitektura mreže je prikazana na sliki 3.



Slika 3: Arhitektura nevronske mreže. Po konvolucijski plasti 7×7 vidimo razgrnjeno plast preostankov, ki se potem ponovi še nekajkrat. Zaradi preglednosti so izpuščene normalizacijske, aktivacijske in odpadne plasti.

3.4 Drevesno preiskovanje Monte Carlo

Drevesno preiskovanje Monte Carlo (MCTS) je hevrističen preiskovalni algoritem. V osnovni obliki se uporablja za igranje iger za 2 igralca, kjer igralca igrata eden proti drugemu. Bolj podroben opis različnih izvedb MCTS je na voljo v [35]. V našem algoritmu MCTS uporabljamo za izboljšanje strategije.

3.5 Končni algoritem FighterZero

Naš algoritem FighterZero temelji na algoritmu AlphaZero [16], ki ga je za šah, šogi in Go implementirala Googlova ekipa DeepMind. Uči se s samo-igranjem (igranjem sam proti sebi). Koda našega agenta in popravljenega ogrodja FightingICE je na voljo na spletnem repozitoriju [36].

Mreža z izhodom strategije napoveduje trenutno strategijo. To strategijo nato izboljšamo z MCTS. Izboljšano strategijo uporabimo za nemudno izbiro poteze, si jo pa tudi zapomnimo (skupaj s trenutnim stanjem) za nadaljnje učenje mreže. Na tak način igro igramo bolje, kot bi jo igrala mreža sama (saj si pomagamo z MCTS), obenem pa se izhod strategije uči napovedovanja vedno boljše strategije. Na koncu vsake igre pa poleg shranjenih podatkov o stanjih (in napovedih strategije) shranimo še nagrade glede na izid igre (en

igrallec je zmagal, drugi izgubil) in s temi podatki učimo na mreži še izhod evaluacije stanja. Ta postopek nato ponavljamo – mi smo za učenje odigrali 1000 iger.

3.6 Metodologija testiranja

Za testiranje smo naša agenta pognali proti trem nasprotnikom, ki so na voljo na spletni strani ogrodja FightingICE [31]. V naraščajočem vrstnem redu glede na težavnost so to bili:

1. KickAI je najpreprostejši agent, ki izvaja akcijo brcanja ne glede na situacijo.
2. MctsAi je agent, ki, prav tako kot naša dva, temelji na drevesnem preiskovanju Monte Carlo, vendar pa ne uporablja strojnega učenja. Ta agent je uradni primerjalni agent tekmovanja in se tudi uporablja kot nasprotnik v njihovem hitrostnem tekmovanju (angl. speedrunning competition).
3. GigaThunder je zmagovalec tekmovanja FTGAIC iz leta 2017. Je precej bolj kompleksen in za vsakega od likov uporablja drugačna pravila obnašanja. Prav tako si pomaga z drevesnim preiskovanjem Monte Carlo, ima pa še mnogo drugih pravil oblike če-potem, po katerih se ravna. Ne uporablja pa nobenih bolj naprednih metod strojnega učenja.

Vsaka tekma je bila sestavljena iz 9 iger po 3 runde. Vsaka runda je pri tem trajala 60 sekund, oziroma dokler eden od likov ni izgubil vseh točk zdravja (HP). Oba lika sta vsako rundo začela s 400 HP. Za izgubljeno igro agent ni prejel nobene točke, za zmago je prejel točko, za morebiten neodločen izid pa 0,5 točke.

4 REZULTATI

V tabeli 1 so izidi, ki sta jih naša dva agenta dosegla pri igranju proti agentom iz ogrodja in s tekmovanja. Izid oblike x-y pomeni, da je naš agent dosegel x točk, nasprotnik pa y.

Tabela 1: Rezultati našega agenta proti testnim agentom.

<i>KickAI</i>	<i>MctsAi</i>	<i>GigaThunder</i>
8,5 - 0,5	0 - 9	0 - 9

Rezultati so bili dokaj neuspešni – agent je zmagal le proti najosnovnejšemu nasprotniku. Za glavno težavo se je izkazal kratek časovni interval, ki ga je agent imel na voljo za posamezno izbiro akcije (16ms za vsako potezo – za primerjavo, AlphaZero je imel 60 sekund na potezo). Poleg tega se je precej časa izgubilo tudi pri uporabi ovojnice Py4J, ki jo za Pythonov vmesnik uporablja ogrodje FightingICE. Kljub številnim poskusom optimizacije algoritma (s katerimi smo ga pospešili za red 10×) drevesno preiskovanje ni izvedlo dovolj simulacij na vsakem koraku izbire akcije.

5 ZAKLJUČEK

V tem delu smo pregledali pristope, ki so se do sedaj uporabljali za globoko učenje v igrah. Nato smo predstavili ogrodje FightingICE in dodatke, ki smo jih v ogrodje implementirali. Predstavili smo nevronske mreže in drevesno preiskovanje Monte Carlo in opisali, kako se združita v celoto.

Implementirali smo samo-igralnega učljivega agenta za pretepaško igro FightingICE. Pri tem smo uporabili pristop, ki je temeljil na algoritmu AlphaZero, ki ga je za igranje iger na igralni deski razvila Googlova ekipa DeepMind. Naš agent je sicer deloval in se je učil, vendar pa se je kratka časovna doba odločanja (16 ms) izkazala za problematično, še posebej ob težavah, ki so nastale pri uporabi ovojnice Py4J. Tako je agent izvajal premalo simulacij MCTS, kar je onemogočilo tako dobro učenje kot tudi dobro igranje igre.

Implementacija torej lahko služi kot osnova za samo-igralnega učljivega agenta za pretepaške igre, vendar pa v trenutnem stanju v tem ogrodju ni primerna za tekmovanje. Nadgradnja ogrodja z uporabo boljše pythonske ovojnice ali pa prevedba agenta v programski jezik Java (in uporaba temu primernih knjižnic

za globoko učenje) pa bi ga lahko pripravila do precej boljšega delovanja in morda celo do zmage proti najboljšim agentom tekmovanja.

Literatura

1. Graves, Alex, et al. Biologically plausible speech recognition with LSTM neural nets. 2004. International Workshop on Biologically Inspired Approaches to Advanced Information Technology. pp. 127-136.
2. Yu, Dong and Deng, Li. Automatic Speech Recognition: A Deep Learning Approach. s. l. : Springer-Verlag London, 2015.
3. Hannun, Awni, et al. Deep Speech: Scaling up end-to-end speech recognition. 2014, arXiv preprint arXiv:1412.5567.
4. Cireşan, Dan, et al. Multi-column deep neural network for traffic sign classification. s. l. : Elsevier, 2012, Neural networks, Vol. 32, pp. 333-338.
5. Gers, Felix A. and Schmidhuber, Jürgen. LSTM recurrent networks learn simple context-free and context-sensitive languages. s.l. : IEEE, 2001, IEEE Transactions on Neural Networks, Vol. 12, pp. 1333-1340.
6. Bengio, Yoshua, et al. A neural probabilistic language model. 2003, Journal of machine learning research, Vol. 3, pp. 1137-1155.
7. Wu, Yonghui, et al. Google's neural machine translation system: Bridging the gap between human and machine translation. 2016, arXiv preprint arXiv:1609.08144.
8. Covington, Paul, Adams, Jay and Sargin, Emre. Deep Neural Networks for YouTube Recommendations. 2016. Proceedings of the 10th ACM Conference on Recommender Systems. pp. 191-198.
9. Chicco, Davide, Sadowski, Peter and Baldi, Pierre. Deep autoencoder neural networks for gene ontology annotation predictions. 2014. Proceedings of the 5th ACM Conference on Bioinformatics, Computational Biology, and Health Informatics. pp. 533-540.
10. Mnih, Volodymyr, et al. Playing Atari with Deep Reinforcement Learning. 12 2013, ArXiv e-prints.
11. Mnih, Volodymyr, et al. Human-level control through deep reinforcement learning. s. l. : Nature Publishing Group, a division of Macmillan Publishers Limited. All Rights Reserved. SN, 25 2 2015, Nature, Vol. 518, p. 529 EP.
12. Tesauero, Gerald. Temporal Difference Learning and TD-Gammon. New York, NY, USA : ACM, 3 1995, Communications of the ACM, Vol. 38, pp. 58-68. ISSN: 0001-0782.
13. Guo, Xiaoxiao, et al. Deep Learning for Real-Time Atari Game Play Using Offline Monte-Carlo Tree Search Planning. [ed.] Zoubin Ghahramani, et al. Advances in Neural Information Processing Systems 27. s.l. : Curran Associates, Inc., 2014, pp. 3338-3346.
14. Silver, David, et al. Mastering the game of Go with deep neural networks and tree search. s.l. : Nature Publishing Group, 27 1 2016, Nature, Vol. 529, pp. 484-489.
15. Silver, David, et al. Mastering the game of Go without human knowledge. s.l. : Nature Publishing Group, 10 2017, Nature, Vol. 550, pp. 354-359. ISSN: 0028-0836.

16. Silver, David, et al. Mastering Chess and Shogi by Self-Play with a General Reinforcement Learning Algorithm. 12 2017, ArXiv e-prints.
17. Silver, Albert. The future is here -- AlphaZero learns chess.
<https://en.chessbase.com/post/the-future-is-here-alphazero-learns-chess>
18. Ranjan, Kushal, Christensen, Amelia and Ramos, Bernardo. Recurrent Deep Q-Learning for PAC-MAN. Stanford University, 450 Serra Mall, Stanford, CA 94305, USA. 2016.
19. Fu, Justin and Hsu, Irving. Model-Based Reinforcement Learning for Playing Atari Games. Stanford University, 450 Serra Mall, Stanford, CA 94305, USA. 2016.
20. Appiah, Naveen and Vare, Sagar. Playing FlappyBird with Deep Reinforcement Learning. Stanford University, 450 Serra Mall, Stanford, CA 94305, USA. 2016.
21. Stevens, Matt and Pradhan, Sabeek. Playing Tetris with Deep Reinforcement Learning. Stanford University, 450 Serra Mall, Stanford, CA 94305, USA. 2016.
22. Zhang, Rongxiao. Convolutional and Recurrent Neural Network for Gomoku . Stanford University, 450 Serra Mall, Stanford, CA 94305, USA. 2016.
23. Lu, Mengxuan and Li, Xuejun. Deep reinforcement learning policy in Hex game system. 2018. 2018 Chinese Control And Decision Conference (CCDC). pp. 6623-6626. ISSN: 1948-9447.
24. Silva, Alysson Ribeiro da and Wanderley Goes, Luis Fabricio. HearthBot: An Autonomous Agent based on Fuzzy ART Adaptive Neural Networks for the Digital Collectible Card Game Hearthstone. 2017, IEEE Transactions on Computational Intelligence and AI in Games, Vol. PP, pp. 1-1. ISSN: 1943-068X.
25. Perot, Etienne, et al. End-to-End Driving in a Realistic Racing Game with Deep Reinforcement Learning. 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW). pp. 474-475.
26. Serafim, Paulo Bruno S., et al. Towards Playing a 3D First-Person Shooter Game Using a Classification Deep Neural Network Architecture. 2017. 2017 19th Symposium on Virtual and Augmented Reality (SVR). pp. 120-126.
27. Wang, Yuanda, He, Haibo and Sun, Changyin. Learning to Navigate through Complex Dynamic Environment with Modular Deep Reinforcement Learning. 2018, IEEE Transactions on Games, p. 1. ISSN: 2475-1502.
28. Justesen, Niels and Risi, Sebastian. Learning macromanagement in starcraft from replays using deep learning. 2017 IEEE Conference on Computational Intelligence and Games (CIG). pp. 162-169.
29. Stanescu, Marius, et al. Evaluating real-time strategy game states using convolutional neural networks. 2016 IEEE Conference on Computational Intelligence and Games (CIG). pp. 1-7.
30. Yoon, Seonghun and Kim, Kyung-Joong. Deep Q networks for visual fighting game AI. 2017 IEEE Conference on Computational Intelligence and Games (CIG). pp. 306-308.
31. Intelligent Computer Entertainment Lab. (ICE Lab.), Ritsumeikan University. FTGAIC: FighTing Game AI Competition.
<http://www.ice.ci.ritsumei.ac.jp/~ftgaic/>
32. Schmidhuber, Jürgen. Deep learning in neural networks: An overview. 2015, Neural Networks, Vol. 61, pp. 85-117. ISSN: 0893-6080.
33. He, Kaiming, et al. Deep Residual Learning for Image Recognition. s.l. : IEEE, 2016. IEEE Conference on Computer Vision and Pattern Recognition.

34. Srivastava, Nitish, et al. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. 2014, Journal of Machine Learning Research, Vol. 15, pp. 1929-1958.
35. Browne, Cameron B., et al. Survey of Monte Carlo Tree Search Methods. 3 2012, IEEE Transactions on Computational Intelligence and AI in Games, Vol. 4, pp. 1-43. ISSN: 1943-068X.
36. Vitek, Matej. FighterZero: A Deep Learning AI for the ICE Fighting Game AI Competition Framework. <https://github.com/MatejVitek/FighterZero>

Contactless fingerprint identification using mobile phone camera

LUKA LOBODA, MATEJ VITEK, BLAŽ MEDEN, PETER PEER &
ŽIGA EMERŠIČ

Abstract Fingerprint systems typically use images acquired with contact based sensors. This poses a problem as additional hardware is needed and full cooperation of the user is required. To address these issues we test the possibility of using mobile phone's camera as a contactless fingerprint sensor. We created our own database of contactless fingerprints acquired with regular mobile phone camera. We used this database to test performance of fingerprint matching algorithm designed to compare regular fingerprint images. With image enhancement we managed to achieve a promising equal error rate of 13.81%. To showcase the usability of the proposed concept of contactless mobile authentication we also developed a mobile application. Authentication with mobile application achieves sensitivity of 80% with 100% of imposters being rejected.

Keywords: • Contactless Identification • Fingerprint • Mobile phone • Authentication • SourceAFIS library •

CORRESPONDENCE ADDRESS: Luka Loboda, University of Ljubljana, Faculty of Computer and Information Science, Večna pot 113, 1000 Ljubljana, Slovenia, e-mail: ll6075@student.uni-lj.si. Matej Vitek, PhD, Associate Professor, University of Ljubljana, Faculty of Computer and Information Science, Večna pot 113, 1000 Ljubljana, Slovenia, e-mail: matej.vitek@fri.uni-lj.si. Blaž Meden, Assistant, University of Ljubljana, Faculty of Computer and Information Science, Večna pot 113, 1000 Ljubljana, Slovenia, e-mail: blaz.meden@fri.uni-lj.si. Peter Peer, PhD, Associate Professor, University of Ljubljana, Faculty of Computer and Information Science, Večna pot 113, 1000 Ljubljana, Slovenia, e-mail: peter.peer@fri.uni-lj.si. Žiga Emeršič, PhD Student, University of Ljubljana, Faculty of Computer and Information Science, Večna pot 113, 1000 Ljubljana, Slovenia, e-mail: ziga.emersic@fri.uni-lj.si.

1 INTRODUCTION

Determining identity of individual has become very important in our society. Using biometric traits to recognize people is becoming more and more important. Fingerprints are a great candidate as a biometric modality and can be represented as entire images, finger ridges or features derived from ridges (minutiae) [1]. They are believed to be unique and they do not change over time significantly [2]. However, fingerprint patterns are usually captured by using a dedicated capacitive or optical sensors [3].

In this paper we test possibility of fingerprint identification based on fingerprints acquired by photographing fingers with regular smart phone camera. First we test the possibility of identification based on contactless images of fingerprints. For matching we used an algorithm designed to match regular contact based fingerprints. We created our own database of fingerprint images acquired by smart phone camera and tested performance of identification on it. Then we designed simple mobile application that enables user to enrol in the system and then perform identification by taking pictures of his fingerprints with the smart phone's camera.

Fingerprint identification systems typically use contact based imaging sensor that operates in a controlled environment. Those systems usually have problem with partial or degraded images due to improper placement of finger, dirt or deformation of finger. One approach to solving these problems is using contactless identification. However, human fingers are curved 3D surfaces and therefore the images can vary with angle and illumination. Presentation of finger also depends on distance from camera and quality of focus [4].

2 FINGERPRINT MATCHING

We used SourceAFIS library for comparing fingerprints. SourceAFIS is an open source library implemented in Java and .NET. The algorithm is based on minutiae, which are basically ridge endings and bifurcations with associated direction angle. Algorithm stores minutia points in template, which we later use to compare fingerprints. Algorithm also uses edges which are lines connecting two minutia points (Fig. 2).

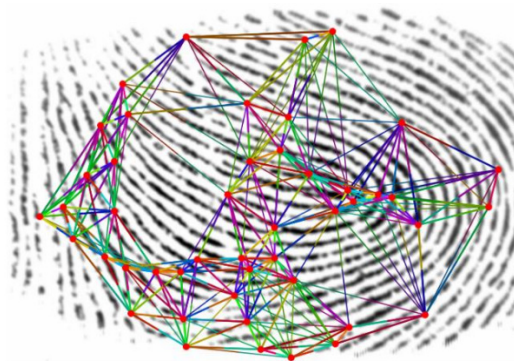


Figure 1: Visualization of edges. Red points are minutiae. Color of edge is determined by length and angles, similar edges have similar colors [6].

Each edge has three properties: length and two angles inherited from its minutia. These three properties are useful as they do not change when edge is moved or rotated. Algorithm then tries to find at least one edge shared by two fingerprints and this edge is the root pair. From the root pair algorithm then follows edges outside and builds pairing consisting of paired minutia and edges. At the end algorithm counts matched features and scores them on how closely they match and this gives us the final score of matching between two fingerprints [6]. For Android mobile application we used Android port of SourceAFIS library [7].

3 FINGERPRINT DATABASE

For evaluation of contactless fingerprint identification, we used our own database of fingerprints captured with camera on smart phone. We used Huawei P10 Lite mobile phone with a 12MP camera with F/2.2 aperture; phase detection autofocus and a single LED flash [5]. Our database of fingerprints includes fingerprints of 8 test subjects. We took pictures of 5 different fingers for each test subject and for every finger we took 6 individual pictures of fingerprints (Fig. 1). In total, our database consists of 240 ($8 \times 5 \times 6$) contactless fingerprints. All images for one finger were taken within one hour in good lightening from the minimum distance that allowed good focus on fingerprint area and with flash on. We decided to use flash as it reveals

fingerprint better as ridges reflect light very well. Information about finger's identity and sequence number (1 to 6) is stored in the name of the image.



Figure 2: Example of one finger sample in our database. Each finger is represented with 6 images of fingerprints.

4 IDENTIFICATION EVALUATION

To test possibility of using contactless fingerprints acquired with smart phone's camera we performed evaluation of identification on our database of contactless fingerprint images. First, we preprocess the database (Fig. 3) by manually rotating images of finger so they are all oriented the same (finger pointing to the top of the image) and cropping images so they only contain fingertips with fingerprints and the background is minimized. Then we transformed all images to grayscale and used Contrast Limited Adaptive Histogram Equalization (CLAHE) [8] to enhance the contrast on images that may be blurred because of poor focus or movement while capturing them.

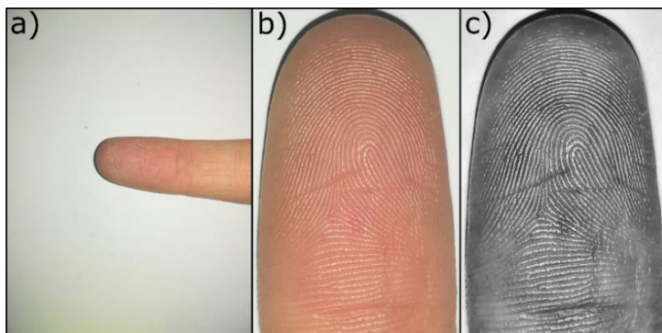


Figure 3: Preprocessing pipeline. We start with original image taken with smart phone's camera (a), which is then cropped and rotated (b), converted to grayscale and finally contrast is enhanced (c).

We designed the experiment to evaluate performance of identification using SourceAFIS [6] library on our database of contactless fingerprints. We decided to use equal error rate (EER) as measure of performance. EER is the point where false match rate (FMR) and false non-match rate (FNMR) are equal. The lower the EER is the better the system is. The experiment consists of two stages. In the first stage, we prepare templates for all fingerprints in database using SourceAFIS [6] library (Fig. 4).

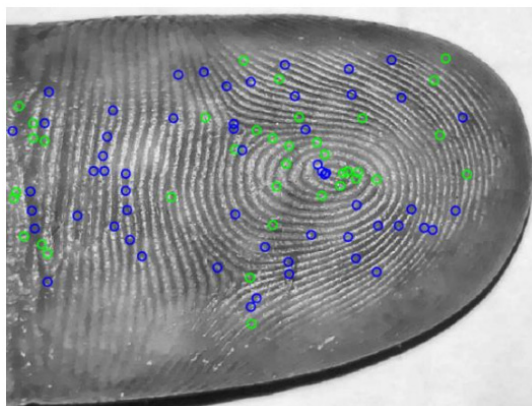


Figure 4: Example of minutiae points detected on an image from our contactless fingerprint database with SourceAFIS [6] library. Green points are endings and blue bifurcations.

We mark each template with identification number of the finger and number of sample, so we can prevent comparing the template with itself and we can easily determine if identification was successful. In second stage we run test sequence of comparing each sample with the rest of samples. This way we get 28.680 comparisons of fingerprints and the scores of those comparisons returned by SourceAFIS [6] algorithm. Then we compare the scores to different values of threshold and if the score is larger than given threshold, we mark this comparison as match. Because we know if the two templates actually belong to the same finger, we can calculate FMR and FNMR for every value of threshold. To determine the value of threshold where EER point is, we search for value of threshold where the difference between FMR and FNMR is the smallest or equal to 0.

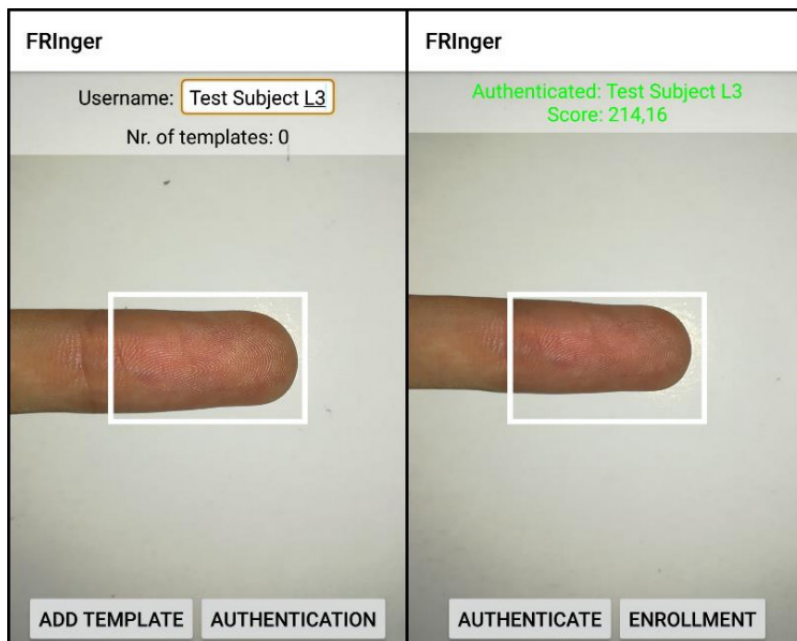


Figure 5. Two modes of mobile application. On the left is enrollment mode and on the right identification mode. White rectangle on the center of screen shows which area will be used to generate template.

5 MOBILE APPLICATION

The main idea of this assignment was to test if mobile phone can be used as an identification system based on contactless fingerprint images taken with unmodified mobile phone's camera. To test this we developed a simple Android application (Fig. 5) that uses Android port of SourceAFIS [7] library for fingerprint comparison and mobile phone's camera to acquire fingerprints.

The application consists of two modes of operation. In the first mode user can add new templates to the database. To add a template user specifies template name and then captures his fingerprint using mobile phone's camera. Extraction of fingerprint from the captured image is implemented by extracting central area of the image, defined by rectangle display on preview screen, so the user can position his finger properly and view quality of focus and lightening. This cropped image of fingerprint is then used to generate template with SourceAFIS [7] library. We did not use any enhancement of acquired

images. Templates are stored on the mobile phone in JSON files and there is no need for storing fingerprints as raw images. Information about identity of the template is stored in the name of template. When templates are added to the database user can try to identify in second mode, where the process of acquiring fingerprint is the same, but now the template extracted from the image is compared to the templates in database. The best matching score is compared to predefined threshold and if the score is larger the identity of matched fingerprint is displayed, otherwise the error is displayed.

6 RESULTS

The results of performance evaluation of SourceAFIS [2] fingerprint matching algorithm on our database of contactless fingerprints are quite promising. Initial result of matching contactless fingerprints using images that are only cropped and rotated was EER of 27.5%. We managed to improve this result by using histogram equalization and achieved EER of 13.81%. Interestingly performance of only grayscale images was worse than using just cropped and rotated images with EER of 42.31%. What is also promising is the fact that FNMR is raising slowly as we are increasing threshold, while FMR drops very fast to 0, as shown in Figure 6.

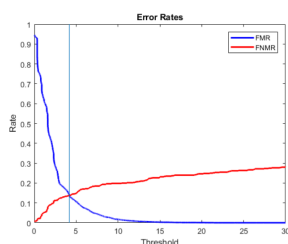


Figure 6. Graph showing changes of FMR and FNMR with different thresholds. Vertical blue line shows where the EER point is.

	Genuine	Imposter
Accepted	40	0
Rejected	10	50

Figure 7. Confusion matrix for authentication test with mobile application.

This is important when considering applications of this system as we can ensure imposters cannot enter our system while still keeping rate of rejected genuine users low. The second test we performed was used to determine if our mobile application could be used as an authentication device. For test we added 5 templates of the same finger to the database representing genuine user. Then

we performed 100 authentication attempts with our mobile app. 50 attempts were with the genuine finger used to generate templates and 50 attempts with imposter fingers. We achieved sensitivity of 80% and specificity of 100% (Fig. 7). Although the number of performed attempts of authentication and range of imposter fingers are not very large we can still say that the application performed well. Because the threshold was set to 35 which is in the area where FMR was equal to 0, when we tested performance of SourceAFIS [2] on our database, our application successfully rejected all imposters. This high threshold meant that some genuine attempts will be rejected but only 20%, which is acceptable as another attempt can be performed in seconds and this one will probably succeed.

7 CONCLUSION

In this work we showed that using regular camera on mobile phone for fingerprint identification is viable. We created our own contactless fingerprint database consisting of 240 fingerprints and used it to test algorithm designed to compare regular contact-based fingerprints. Our test show that EER of 13.81% can be obtained using cropped images and enhanced contrast. To prove possibility of contactless fingerprint identification with mobile phone we developed our Android application. Application was tested with 50 imposter authentications and all of them were rejected, with false rejection rate of 20%. We also noticed that match scores with genuine fingerprints were much higher than the threshold, so the threshold could be even raised to prevent any false acceptance. However, the performance of the mobile application depends on the quality of templates generated when enrolling user. One possible improvement of the application is to check the quality of template when enrolling and reject bad templates. Current implementation of the application also uses auto-focus which does not work well sometimes, especially when the background is complex. With all these improvements, we argue that mobile phones could soon be used as mobile contactless fingerprint authentication devices.

References

1. A. K. Jain, R. Bolle, and S. Pankanti, *Biometrics: personal identification in networked society*. Springer Science & Business Media, 2006.
2. P. Komarinski, *Automated fingerprint identification systems (AFIS)*. Elsevier, 2005.
3. D. Peralta, I. Triguero, R. Sanchez-Reillo, F. Herrera, and J. Benitez, "Fast fingerprint identification for large databases," vol. 47, no. 2, pp. 588–602, 2014.
4. A. Kumar and Y. Zhou, "Contactless fingerprint identification using level zero features," in *CVPR 2011 Workshops*, June 2011, pp. 114–119.
5. *Huawei P10 Lite review*. Pridobljeno na https://www.gsmarena.com/huawei_p10_lite-review-1617.php
6. R. Važan, "SourceAFIS," <https://sourceafis.machinezoo.com/algorithm>, accessed: 6. 1. 2019.
7. *Source ASFIS - How it works*. Pridobljeno na <https://sourceafis.machinezoo.com/algorithm>
8. A. M. Reza, "Realization of the contrast limited adaptive histogram equalization (CLAHE) for real-time image enhancement," *Journal of VLSI signal processing systems for signal, image and video technology*, vol. 38, no. 1, pp. 35-44, 2004.

Optično branje podatkov iz računa

ALEŠ HORVAT IN BORUT BATAGELJ

Povzetek V članku je predstavljen postopek, ki smo ga razvili v sklopu diplomske naloge na fakulteti za računalništvo in informatiko, kot rešitev za problem avtomatiziranega vnosa stroškov v mobilno aplikacijo za osebno finančno evidenco. Aplikacija zajema storitev, ki uporabniku omogoča zajem slike računa nad katero izvede optično prepoznavo znakov, ter prebran tekst posreduje algoritmom za iskanje podatkov na računu. Končne rezultate lahko uporabnik v primeru napake popravi in s tem tudi zmanjša možnost ponovitve enake napake pri naslednjih branjih. Podatki, ki jih aplikacija išče so: končni znesek, datum, izdajatelj računa, naslov na katerem je bil račun izdan in kategorija. Za optično prepoznavo znakov je bila uporabljena knjižnica Google Mobile Vision. Na koncu članka so predstavljeni rezultati testiranja učinkovitosti skeniranja računov.

Ključne besede: • Optična prepoznavna znakov • mobilna aplikacija • branje računa • računalniški vid • avtomatizirano prepoznavanje •

NASLOVA AVTORJEV: Aleš Horvat, Univerza v Ljubljani, Fakulteta za računalništvo in informatiko, Večna pot 113, 1000 Ljubljana, Slovenija, e-pošta: horvat.ales93@gmail.com. dr. Borut Batagelj, višji predavatelj, Univerza v Ljubljani, Fakulteta za računalništvo in informatiko, Večna pot 113, 1000 Ljubljana, Slovenija, e-pošta: borut.batagelj@fri.uni-lj.si.

1 UVOD

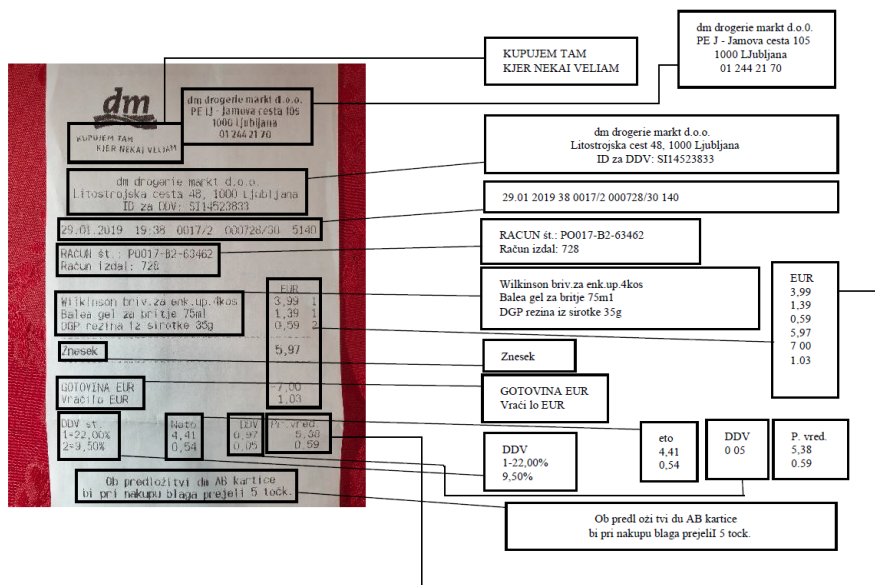
Nepreglednost nad osebnimi finančnimi sredstvi je problem s katerim se sooča veliko število ljudi. Tudi sam se včasih sprašujem kam so šla moja sredstva v preteklem mesecu, na katerih področjih bi lahko prihranil nekaj denarja in kakšen je moj finančni napredek glede na pretekla obdobja. Aplikacij, ki odgovarjajo na takšna vprašanja je na trgu zelo veliko, vendar velika večina le teh nima storitve, ki bi uporabnika nekoliko razbremenile pri vnosu podatkov v aplikacijo [1, 2, 3]. Zato smo se odločili, da v sklopu diplomske naloge razvijemo mobilno aplikacijo, katere glavna prednost bi bila avtomatiziran vnos računov v podatkovno bazo.

2 POSTOPEK ISKANJA PODATKOV NA RAČUNU

Uporabnik s pomočjo aplikacije zajame sliko računa iz katere je potrebno izluščiti podatke, ki jih želimo evidentirati. V prvem koraku je torej potrebna optična prepoznavna znakov. Odločili smo se za uporabo knjižnice Google Mobile Vision, ki poleg OCR omogoča še prepoznavanje obrazov, bar/QR kod, ter objektov na slikah in videih [4].

2.1 Delovanje Google Mobile Vision knjižnice

Kot rezultat obdelave slike nam knjižnica vrne polje objektov tipa TextBlock, v katerem posamezen element predstavlja odstavek oziroma smiselno združeno komponento prebranega teksta. Ta komponenta je razdeljena na pod-komponente enakega tipa, ki predstavljajo vrstice. Vrstice so nato razdeljene na besede, ki predstavljajo najnižji nivo prebranih komponent. Vsak objekt v hierarhiji zajema tudi podatke o poziciji na zajeti sliki, in sicer odmik od levega, desnega, zgornjega in spodnjega roba. Zaznane komponente so razvidne na sliki 1.



Slika 1: Rezultat knjižnice za optično prepoznavo znakov

Iz računa želimo razbrati končni znesek, datum izdaje računa, naziv podjetja, ki je izdalo račun, morebitno lokacijo (kje je bil račun izdan), ter v katero kategorijo lahko umestimo račun. Za vsakega od podatkov smo kreirali razred z implementacijo metode, ki prejme objekt tipa `TextBlock`. Vsak razred s pomočjo te metode v prejetih podatkih išče enega od naštetih atributov računa. Tekstovne bloke najprej sortiramo, tako da si sledijo od zgoraj navzdol in od desne proti levi, ter se skozi njih sprehodimo v zanki, v kateri potem vsaki instanci razreda, ki je zadolžena za iskanje specifičnega podatka v tekstu podamo tekstovni blok.

2.2 Algoritmi za iskanje podatkov na računu

Prebran tekst iz računa velikokrat vsebuje napake, saj so si določeni znaki med seboj preveč podobni. Ključno je, da te napake ne vplivajo na pravilnost algoritma za iskanje podatka med prebranim tekstom. Iz OCR-ja pogosto dobimo Z (črko) namesto 7 (število), O (črko) namesto 0 (število), presledek med dvema znakoma, čeprav ga dejansko ni, in tako naprej. Takšne napake poskušamo zaznati in odpraviti predno se lotimo iskanja podatka, in sicer tako, da v primeru zaznane napake znak zamenjamo z znakom iz podmnožice na

katero je omejen iskan podatek. Tabelo za mapiranje znakov smo sestavili ročno, glede na statistiko napačno prebranih znakov.

2.2.1 Znesek

Prvi korak v iskanju zneska je, da izločimo vrstice, ki ne vsebujejo nobenega kandidata za znesek oziroma nobenega decimalnega števila. To lahko storimo s pomočjo regularnih izrazov. V drugem koraku iz preostalih vrstic izločimo besede za katere vemo, da ne ustrezajo formatu iskanega podatka. Tako med kandidati za končni znesek ostanejo samo zneski posameznih izdelkov na računu, končni znesek, znesek plačila, znesek vračila in tako naprej. Glede na to, da se končni znesek običajno nahaja pod zneski posameznih izdelkov, ki so poravnani na desni rob računa, si lahko pri iskanju končnega zneska pomagamo s koordinatami, ki nam jih vrne knjižnica OCR. Zneske, ki imajo približno enak odmik od desnega roba spravimo v skupen seznam. Tako dobimo sezname zneskov, ki v bistvu predstavljajo stolpce. Potencialne cene izdelkov v stolpcu seštejemo in seštevek primerjamo s potencialnim končnim zneskom. Algoritem je sledeč: Če je element na n -tem mestu končni znesek, potem velja da je seštevek elementov na mestih $n-1$, $n-2$, $n-x$ enak elementu na n -tem mestu. V primeru, ko v več kot enem stolpcu pridemo do takšnega rezultata, vzamemo največjega od potencialnih končnih zneskov.

2.2.2 Datum

Formati v katerih se lahko pojavi datum so prikazani v tabeli 1.

Tabela 1: Primeri različnih formatov datuma, ki se lahko pojavi na računu

DD.MM.YYYY	DD-MM-YYYY	DD/MM/YYYY
YYYY.MM.DD	YYYY-MM-DD	YYYY/MM/DD
DD.MM.YY	DD-MM-YY	DD/MM/YY
YY.MM.DD	YY-MM-DD	YY/MM/DD

Formati datumov imajo različna ločila med komponentami. Za lažje procesiranje jih poenotimo tako, da zamenjamo vse pojavitve znakov minus (-) in poševnic (/) s piko (.). S tem smo število možnih formatov zmanjšali na 4. Z zanko se nato sprehodimo čez vse komponente na najnižjem nivoju (besede)

in preverimo če se trenutna beseda ujema s katerim od možnih formatov. V primeru, da se kandidat ujema s formatom, ki predpisuje okrajšano letnico, ocenimo kater del predstavlja letnico, ter jo zamenjamo. Med najdenimi datumi v tekstu vedno izberemo datum, ki je najbližji trenutnemu datumu.

2.2.3 Izdajatelj računa

Izdajatelj računa se običajno nahaja nekje pri vrhu računa. Vzamemo prvi tekstovni blok na računu in se z zanko sprehodimo čez besede v bloku. Če se beseda ujema s katerim koli podjetjem v bazi podatkov, zaključimo procesiranje in vrnemo najdeno podjetje. V nasprotnem primeru predvidevamo, da je naziv prva vrstica v tekstovnem bloku.

2.2.4 Lokacija izdaje računa

V prvem koraku iskanja lokacije zmanjšamo nabor kandidatov, tako da izločimo vse vrstice, ki ne vsebujejo potencialnega imena ulice in hišne številke (zaporedje več abecednih znakov, presledek in pozitivno število z morebitno pripono, ki označuje deljeni naslov). Zapise filtriramo z regularnim izrazom. Drugi korak je validacija kandidatov, ki ustrezajo našemu filtru. Naslov namreč ni zapisan v nekakšnem unikatnem formatu, saj je to le neko zaporedje besed in števil, ki nam brez predhodnega znanja ne pove veliko. Na izgled enak zapis (na primer: Miza 108, Natakara 1) lahko aplikaciji izgleda kot legitimen naslov, vendar to ni. Za validacijo smo uporabili Googlovo spletno storitev, Geocoding API [5]. Dostopna je preko HTTP protokola, z metodo GET. Za vsak potencialen naslov pošljemo zahtevek do storitve, ki nam vrne seznam formatiranih naslovov, ki ustrezajo našemu iskanju. Če je seznam prazen, pomeni, da tekst, ki smo ga poslali ni veljaven naslov. Če na računu najdemo več veljavnih naslovov izberemo prvega, ki še ni bil popravljen s strani uporabnika, po zaključenem skeniranju računa.

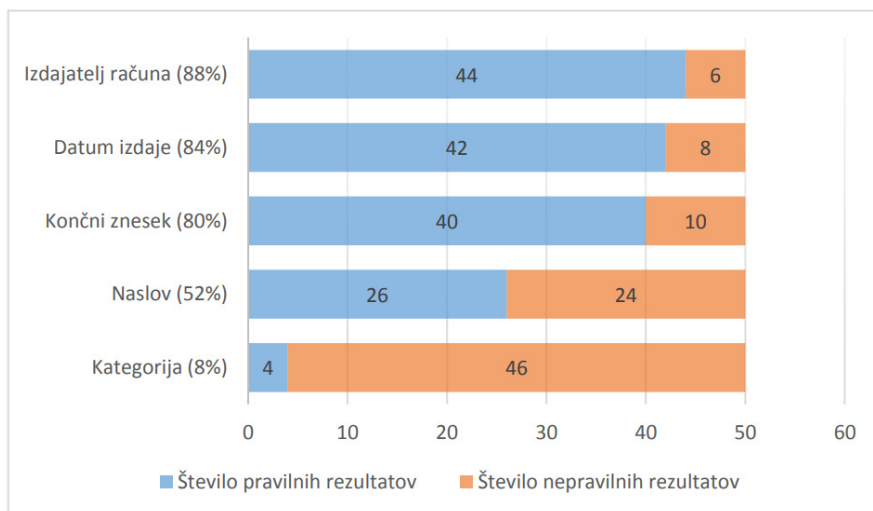
2.2.5 Kategorija

Kategorijo v katero bi uporabnik umestil skenirani račun, ocenimo glede na podatek o izdajatelju računa. Algoritem poišče vse kategorije, ki so do trenutka skeniranja bile povezane na izdajatelja računa, ter vzame tisto, s katero je bil izdajatelj povezan največkrat. Če izdajatelja še ni v bazi podatkov, mora za kategorijo poskrbeti uporabnik.

2.3 Učinkovitost rešitve

Pri testiranju algoritmov smo uporabili 50 računov, ki med razvojem aplikacije niso bili uporabljeni v namen prilagajanja skeniranja. Testiranje je bilo izvedeno tako, da smo slike vseh 50 računov skenirali s pomočjo aplikacije, ter si shranili rezultate, katerih pravilnost smo preverjali ročno.

Končni znesek računa je bil pravilno prebran v 80 odstotkih (40 računov), datum izdaje v 84 odstotkih (42 računov), izdajatelj računa v 88 odstotkih (44 računov), lokacija oziroma naslov na katerem je bil račun izdan v 52 odstotkih (26 računov), pri kategoriji pa je odstotek pravilnosti znašal le 8 odstotkov (4 računi). Tukaj velja pripomniti da je umeščanje računa v kategorijo odvisno od obstoječih podatkov v bazi, pravilnost prikazanega podatka pa je subjektivno mnenje posameznika. Dokaj nizek odstotek pravilnosti lokacije izdaje računa, pa je posledica dejstva, da se na računu v določenih primerih nahajata naslov podjetja in naslov prodajne enote na kateri je bil račun izdan, saj je 50 odstotkov napačno prebranih naslovov, posledica ravno tega, da je aplikacija namesto naslova prodajne enote, izbrala naslov podjetja. Slika 2 prikazuje graf uspešnosti skeniranja po postavkah računa.



Slika 2: Učinkovitost algoritmov skeniranja računov po atributih

Z uporabo aplikacije je predvideno tudi povečanje odstotka uspešnosti skeniranja računov, saj je uspešnost iskanja nekaterih elementov računa odvisna tudi od obstoječih podatkov v bazi.

3 ZAKLJUČEK

Skeniranje računa s pomočjo tehnologije OCR je proces, katerega učinkovitost skorajda ne more biti sto odstotna. Glavni razlog za to je, da izdan račun nima nekega predpisanega formata na katerega bi se algoritmi za iskanje podatkov lahko zanašali. Za izboljšavo učinkovitosti skeniranja bi bilo potrebno zgraditi občutno večjo bazo testnih primerov, kot smo jo uporabljali pri izdelavi aplikacije, ki bi služila za prilagajanje algoritmov iskanja podatkov na računih. Bazo testnih računov bi lahko nadgrajevali uporabniki aplikacije, tako da bi opcijsko pošiljali podatke povezane s skeniranjem računa s pomočjo katerih bi potem lahko tudi izboljšali storitev.

Literatura

1. *Google Play. Small Receipts.* Pridobljeno na https://play.google.com/store/apps/details?id=wb.receipts&hl=en_US
2. *Google Play. Money Lover: Money Manager, Budget Expense Tracker.* Pridobljeno na <https://play.google.com/store/apps/details?id=com.bookmark.money>
3. *Google Play. Receipt Bank: Auto Bookkeeping & Receipt Scanner.* Pridobljeno na <https://play.google.com/store/apps/details?id=com.receiptbank.android>
4. *Google Developers. Mobile Vision.* Pridobljeno na <https://developers.google.com/vision/introduction>
5. *Google Maps Platform. Developer Guide.* Pridobljeno na <https://developers.google.com/maps/documentation/geocoding/intro>

Subdivided Ear Recognition

DAVID ROMERO, MATEJ VITEK, BLAŽ MEDEN, PETER PEER
& ŽIGA EMERŠIČ

Abstract The present paper addresses the performance of different ear feature extractors on partial ear images. The main goal is to find out which parts of the ear have major influence on successful recognition for each extractor. In this sense, a whole ear recognition pipeline has been simulated using Annotated Web Ears (AWE) Toolbox and dataset [1]. Ears have been divided in up, down, internal and external parts and results have been compared. It has been demonstrated the existence of performance gaps between different ear parts and extractors. Trying to exploit that, a score level distance fusion approach has been tested combining separately obtained distances by means of weighted averaging.

Keywords: • Ear Recognition • Biometrics • Computer Vision • AWE dataset • Texture features •

CORRESPONDENCE ADDRESS: David Romero, University of Ljubljana, Faculty of Computer and Information Science, Večna pot 113, 1000 Ljubljana, Slovenia, e-mail: dr7289@student.uni-lj.si. Matej Vitek, PhD, Associate Professor, University of Ljubljana, Faculty of Computer and Information Science, Večna pot 113, 1000 Ljubljana, Slovenia, e-mail: matej.vitek@fri.uni-lj.si. Blaž Meden, Assistant, University of Ljubljana, Faculty of Computer and Information Science, Večna pot 113, 1000 Ljubljana, Slovenia, e-mail: blaz.meden@fri.uni-lj.si. Peter Peer, PhD, Associate Professor, University of Ljubljana, Faculty of Computer and Information Science, Večna pot 113, 1000 Ljubljana, Slovenia, e-mail: peter.peer@fri.uni-lj.si. Žiga Emeršič, PhD Student, University of Ljubljana, Faculty of Computer and Information Science, Večna pot 113, 1000 Ljubljana, Slovenia, e-mail: ziga.emersic@fri.uni-lj.si.

1 INTRODUCTION

Image based biometric subject recognition has always been an active research topic in computer vision field. During the years, several physical metrics (face, iris...) and feature extractor algorithms have been explored. The present work is focused on ear recognition, a promising topic that still has several research issues unsolved.

According to [1], feature extraction, used for ear recognition, can be divided into four different categories: geometric, holistic, local and hybrid. In this paper local methods have been addressed. They are based on the principle of evaluating the whole ear texture without looking for specific points or shapes. Because of that, it is interesting to address if different ear parts have different influence on recognition performance of these extractors.

This information is especially useful when facing one of the most typical issues: occlusion. In some cases, clear images of ears cannot be acquired and recognition must be performed only on ears occluded by hair, glasses or earrings. Furthermore, depending on the occluded ear part recognition systems could suffer different performance degradation rates. Thus, it is important to investigate if these differences actually exist and if so, how big they are for each feature extractor.

Despite helping to better overcome occlusion it is also interesting to address if this performance asymmetry between different ear parts could be somehow exploited to improve recognition rates when whole ear images are available.

2 SUBDIVIDED EAR RECOGNITION

We perform recognition using two different subdivision approaches: up-down and internal-external. For internal-external subdivision the whole dataset has been first divided into left-right images. The idea is to simulate separately up-down and external-internal parts of left and right ears and look for asymmetries in the performance.

In order to get an overall and consistent picture of the topic, simulations have been carried out using several feature extractors. The following ones have been tested: Local Binary Patterns (LBP) [2], Binarized Statistical Image Features (BSIF) [3],[4], Local Phase Quantization (LPQ) [5], Rotation Invariant LPQ (RLPQ) [6], Patterns of Oriented Edge Magnitudes (POEM) [7], Histogram of Oriented Gradients (HOG) [8],[9] and Dense Scale Invariant Feature Transform (DSIFT) [10]. All of them are already implemented in AWE toolbox and belong to local approach category.

For score level fusion distances from different parts of the ear have been combined based on a simple weighted averaging (1) where α defines the averaging weight and goes from 0 to 1.

$$d_{comb} = \alpha d_1 + (1 - \alpha)d_2 \quad (1)$$

3 RESULTS

To obtain partial ear images for the experiments, we used AWE dataset. It contains 1000 images of 100 subjects that have been subdivided following the criteria shown in Figure 1. Identification results based on the AWE dataset and ear division is described in the continuation of this section.

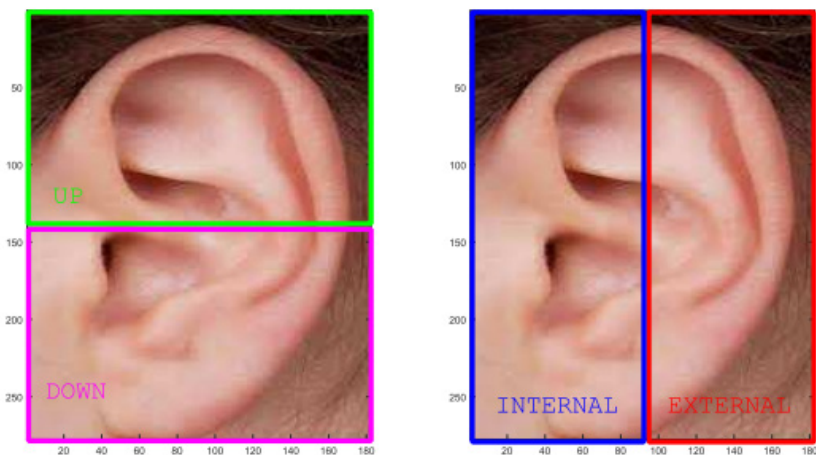


Figure 1: Ear subdivision approaches.

3.1 Up-Down comparison

For up-down comparison Protocol 2 of AWE toolbox has been used. A previously defined distribution of a train set of 600 images has been used to test the performance of each algorithm based on 5 K-Fold and chi square distance.

Obtained Rank-1 absolute and degradation values are displayed in Table 1. The Gap column shows the difference between upper and down parts. At the same time, Cumulative Match Characteristic (CMC) curves are plotted in Figure 2.

Table 1: Rank-1 values for Up-Down comparison

<i>DATASET</i>	<i>AWE</i>	<i>Up</i>	<i>Deg.</i>	<i>Down</i>	<i>Deg.</i>	<i>Gap</i>
LBP	43.6 ±7.2	36.3±5.0	7.3	34.7±7.5	8.9	+1.6
BSIF	48.5±6.9	42.9±3.5	5.6	36.8±9.4	11.7	+6.1
LPQ	43.3±7.7	39.2±5.1	4.1	32.4±9.0	10.9	+6.8
RILPQ	43.5±9.3	36.2±5.1	7.3	31.2±7.4	12.3	+5.0
POEM	49.0±6.9	40.1±5.7	8.9	36.4±7.9	12.6	+3.7
HOG	43.5±8.0	30.9±3.3	12.6	34.0±8.0	9.5	-3.1
DSIFT	43.6±8.5	34.1±6.1	9.5	33.9±13.5	9.7	+0.2

It is clear that, for most of the algorithms, upper part of the ear has a higher influence in successful recognition. HOG and DSIFT are the only descriptors where upper part is equal or worse than down part. Especially for HOG, the performance degradation for both up and down parts is highly noticeable.

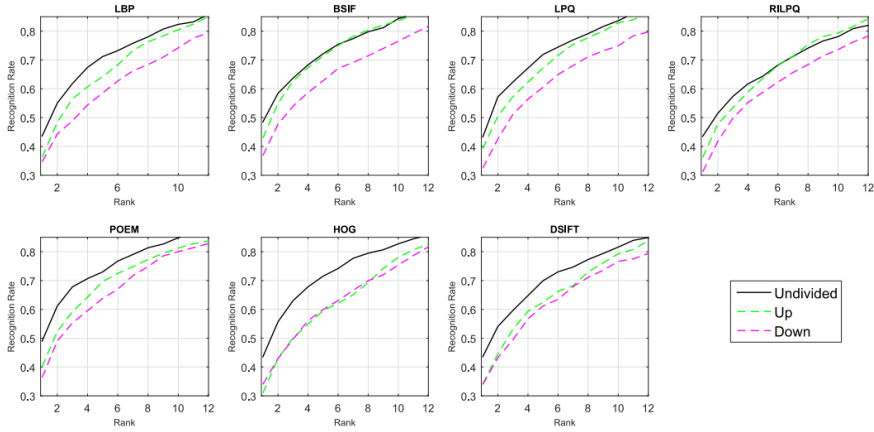


Figure 2: CMC curves for Up-Down comparison

On the other hand, for BSIF, LPQ and RILPQ obtained upper part curves are surprisingly close to ones with the whole ear.

3.1 External-Internal comparison

In this case, internal and external parts of left and right ears have been separately tested following AWE Protocol 1. Thus, the entire datasets have been tested with a random distribution, 5 K-Fold evaluation and chi square distance. Obtained Rank-1 results for both left and right ears are shown in Tables 2 and 3 respectively.

Table 2: Rank-1 values for Left Internal-External comparison

<i>DATASET</i>	<i>Left</i>	<i>Left Int.</i>	<i>Deg.</i>	<i>Left Ext.</i>	<i>Deg.</i>	<i>Gap</i>
LBP	57.1±6.3	55.0±7.1	2.1	42.9±4.0	14.2	+12.1
BSIF	59.4±6.9	62.7±6.8	-3.3	44.4±4.6	15.0	+18.3
LPQ	58.2±6.0	54.0±6.8	4.2	38.6±4.9	19.6	+15.4
RILPQ	56.7±6.5	47.2±5.8	9.5	40.6±5.3	16.1	+6.6
POEM	62.7±4.5	47.7±6.4	15.0	43.5±6.3	19.2	+4.2
HOG	57.9±5.3	41.7±5.0	16.2	41.9±4.8	16.0	-0.2
DSIFT	56.9±5.2	51.7±6.9	5.2	44.4±2.8	12.5	+7.3

According to the results, internal part appears to be more relevant than external part. All extractors have achieved better or at least equal recognition rates for this part of the ear and this difference is consistent for both left and right ears.

The performance gap is not constant for all the extractors. HOG shows again practically no gap while BSIF and LPQ show the biggest one. Again, BSIF achieves almost the same performance for internal part and whole ear images. LPQ and DSIFT also get quite close to the whole ear curve.

Table 3: Rank-1 values for Right Internal-External comparison

<i>DATASET</i>	<i>Right</i>	<i>Right Int.</i>	<i>Deg.</i>	<i>Right Ext.</i>	<i>Deg.</i>	<i>Gap</i>
LBP	52.4±9.9	48.3±2.1	4.1	38.5±1.9	13.9	+9.8
BSIF	57.7±6.3	55.2±3.7	2.5	38.7±2.7	19.0	+16.5
LPQ	52.9±5.2	48.8±4.3	4.1	34.2±2.2	18.7	+14.6
RILPQ	49.4±6.5	42.9±4.0	6.5	34.4±3.6	15.0	+8.5
POEM	58.9±7.1	44.2±3.6	14.7	39.4±3.7	19.5	+4.8
HOG	54.8±8.1	40.0±3.2	14.8	36.5±4.7	18.3	+3.5
DSIFT	51.2±7.5	47.3±3.0	3.9	30.2±3.9	21.0	+17.1

3.1 Distance fusion

Separately obtained distances for up and down parts have been combined according to equation (1) for three different values of α : 0.5, 0.65 and 0.8. Obtained CMC curves for up-down are depicted in Figure 3. As it is shown, proposed distance combination always works equal or better than partial ear images separately. However, in general these improvements are not big enough to outperform recognition rates obtained with whole ear. Interestingly, BSIF is the only feature extractor that achieves a global improvement. Same distance fusion approach has been applied to internal-external ear parts with similar results.

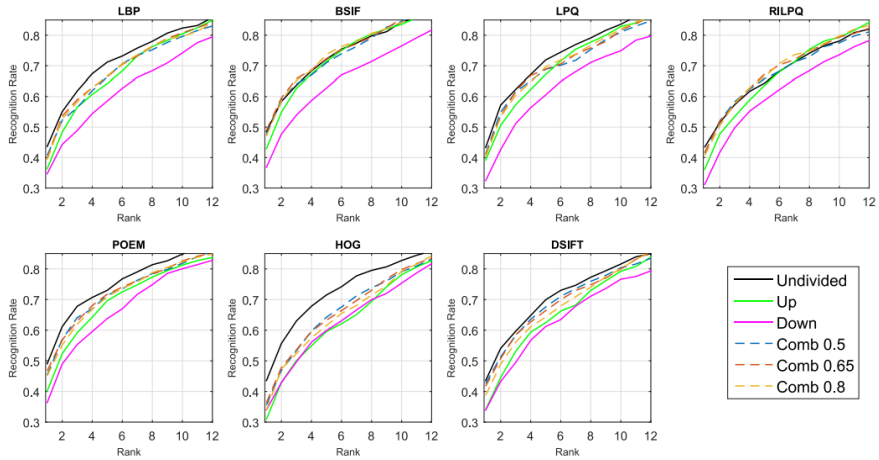


Figure 3: CMC curves for Up-Down distance combining

4 DISCUSSION AND FUTURE WORK

Results show that, according to Rank-1, there is a performance gap of up to +7 between upper and down parts of the ear. At the same time, the gap between internal and external parts is up to +18.3. However, these differences are variable between different feature extractors. The different nature of each extractor could be an explanation for these differences.

Although simulated weighted averaging distance fusion method showed small improvements, the existence of a performance gap suggests that it could be possible to exploit it to improve the overall performance. In this sense, some future research steps can be identified. First, smaller and more complex ear parts could be tested. Secondly, new fusion and feature extractor methods could be developed to exploit performance asymmetry between ear parts. Finally, feature extractors that have shown smaller degradation should be strongly considered in future occluded ear recognition systems.

5 CONCLUSIONS

Throughout this paper, the influence of different ear parts over several feature extractor algorithms has been addressed. Based on AWE toolbox and dataset, simulations have been carried out with images of only one part of the ear (up, down, external or internal). According to obtained results, we can conclude that, for most of the feature extractors, upper and internal parts have stronger influence than down and external parts. This performance gap between internal-external parts is bigger than the one for up-down parts. Both performance gap and degradation differ significantly depending on used feature extractor. We propose to use here presented degradation values as a metric of the sensitivity to occlusion of a feature extractor.

At the same time, separately calculated distances have been combined based on a basic weighted averaging. A small improvement has been achieved mostly for BSIF extractor but results show that, for most of the cases there is no significant improvement. However, this should be considered as a first step to exploit existing performance gap since a lot of research work remains to be done: different subdivision strategies or development of new feature extractors or fusion models based on obtained results.

Literatura

1. Ž. Emeršič, V. Štruc, and P. Peer, "Ear recognition: More than a survey," *Neurocomputing*, 2017.
2. Y. Guo and Z. Xu, "Ear recognition using a new local matching approach," in *Image Processing, 2008. ICIP 2008. 15th IEEE International Conference on*. IEEE, 2008, pp. 289–292.
3. A. Benzaoui, A. Hadid, and A. Boukrouche, "Ear biometric recognition using local texture descriptors," *Journal of Electronic Imaging*, vol. 23, no. 5, p. 053008, 2014.
4. A. Benzaoui, N. Hezil, and A. Boukrouche, "Identity recognition based on the external shape of the human ear," in *2015 International Conference on Applied Research in Computer Science and Engineering (ICAR)*. IEEE, 2015, pp. 1–5.
5. A. Pflug, C. Busch, and A. Ross, "2d ear classification based on unsupervised clustering," in *Biometrics (IJCB), 2014 IEEE International Joint Conference on*. IEEE, 2014, pp. 1–8.
6. V. Ojansivu, E. Rahtu, and J. Heikkilä, "Rotation invariant local phase quantization for blur insensitive texture analysis," in *Pattern Recognition, 2008. ICPR 2008. 19th International Conference on*. IEEE, 2008, pp. 1–4.

7. N.-S. Vu and A. Caplier, "Face recognition with patterns of oriented edge magnitudes," in European conference on computer vision. Springer, 2010, pp. 313–326.
8. A. Pflug, P. N. Paul, and C. Busch, "A comparative study on texture and surface descriptors for ear biometrics," in Security Technology (ICCST), 2014 International Carnahan Conference on. IEEE, 2014, pp. 1–6.
9. N. Damer and B. Führer, "Ear recognition using multi-scale histogram of oriented gradients," in Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP), 2012 Eighth International Conference on. IEEE, 2012, pp. 21–24.
10. Križaj, V. Štruc, and N. Pavešić, "Adaptation of sift features for robust face recognition," in International Conference Image Analysis and Recognition. Springer, 2010, pp. 394–404.

Comparing performance of biometric models trained on different groups with Bayesian statistics

ALEN AJANOVIĆ, PETER PEER & ŽIGA EMERŠIČ

Abstract When developing a decision model, data bias can have large unwanted consequences. In this paper we explore whether a trained ear detection model performs better on female or male images depending on what its training set constituted. We use three separately trained convolutional neural networks (CNN) based on the U-Net architecture as well as Haar cascades for comparison. The three distinct training sets consist of a balanced set of females and males, a set of all females, and a set of all males. After being trained on each, they attempted to produce a mask locating an ear of both groups. With Bayesian statistics, we modelled a normal distribution of their intersection over union scores on similarly sized testing sets. Results show that models tend to prefer one group over the other, but that this difference is too marginal to be practically relevant. The overall success achieved is related to the size of the training set and the larger it is, the better it seems to perform on both genders. In our case, a model that was trained on only males performed equally well classifying females as the one trained on both genders if their training sets were similarly large.

Keywords: • Ear segmentation • Neural networks • Bayesian statistics • Bias • U-Net achitecture •

CORRESPONDENCE ADDRESS: Alen Ajanovič, University of Ljubljana, Faculty of Computer and Information Science, Večna pot 113, 1000 Ljubljana, Slovenia, e-mail: aa7688@fri.uni-lj.si. Peter Peer, PhD, Associate Professor, University of Ljubljana, Faculty of Computer and Information Science, Večna pot 113, 1000 Ljubljana, Slovenia, e-mail: peter.peer@fri.uni-lj.si. Žiga Emeršič, PhD Student, University of Ljubljana, Faculty of Computer and Information Science, Večna pot 113, 1000 Ljubljana, Slovenia, e-mail: ziga.emersic@fri.uni-lj.si.

1 INTRODUCTION

With the advent of automatic biometric approaches, the need for unbiased data and derived models which are able to perform well on a wide variety of groups has increased as well. Developing a decision model built on data that contains certain biases is problematic as this can then reflect in biased decisions made by such a model. This is especially important in biometrics, where allowing an intruder to enter a system could have very adverse safety effects. Observations were made in face recognition models where the skin tone had a large influence on the accuracy of a developed system [1]. However, to the best of our knowledge, no such tests have been performed on ear images. In the work of Emeršič et al. [2], the authors analysed performance of their PED-CED approach on gender-specific data but did not train the models separately to observe any possible implications. To explore if and to what extent gender causes bias in ear detection, we evaluate different state-of-the-art approaches on two different groups - males and females. We use Haar cascades [3] and a U-Net based deep neural network [4]. We measure union-over-intersection accuracy of our predictions and observe whether there is discrepancy between differently trained models, groups, or systems in general. Finally, we perform the analysis with aid of Bayesian statistics to see whether there is some inherent bias in our system.

2 METHODOLOGY

To analyse performance, we used a specific CNN architecture for segmentation and Haar cascades. Architectures were selected due to their performance reports in various literature. We selected the latter, Haar cascades, due to their popularity before neural networks had started actively outperforming them.

2.1 Haar cascades

For ear detection, we used Python and the OpenCV library, which already contains everything required to run the model (cascades, optimisations, the algorithm, etc.). The proposed algorithm is based on the work of Viola and Jones [3]. In our implementation we have chosen the parameters to be 1-nearest neighbour for classification, and a scaling factor of 1,05.

2.2 Convolutional neural networks

In our work, we have trained three separate different networks in order to analyse how they will perform on males vs. females. In each case, we were working with a relatively small number of images for a neural network, so for each model, we have additionally created 20,000 artificial images. These are just the original subjects which are transformed in some way, e.g. the image is skewed, flipped, stretched, sheared, or any random combination of those. The same process is applied to the mask, so it can still accurately represent the ear in the original image as the network is trained.

The used loss function was binary cross entropy. In our tests, a smaller learning rate performed significantly better, and experimentally we set it on $1e-05$ as a trade-off between time taken to train the model and accurately reproducing a mask. The generated mask represents each output as an independent model where each pixel is a predicted likelihood of it belonging to an ear in the accompanying image.

The majority of code and the architecture was based on the U-Net architecture as uploaded by Zhixuahao [5]. Some slight modifications were made to the pipeline and some parameters were changed, but the general approach remains the same. We have trained each of the models on 20 epochs and 2,000 iterations. On a GTX 980 graphics card, the networks trained for about 3-4 hours each.

As a neural network trains, over time, the mask converges to what is shown on Figure 1, creating a proper rectangular mask. As we are trying to measure bias of the system, we did not perform any additional operations on the mask to clean it up, but rather left it as is for a better comparison of end results.



Figure 1: Masks as created by the neural network after 10,000 iterations.

3 BAYESIAN ANALYSIS

To approximate a statistical distribution for e.g. females, after we train a model, we randomly sample 200 predicted masks for females. We calculate the average intersection over union for this sub-sample and mark it as our first reading. We repeat this process 250 times to obtain a vector of 250 readings, where each reading is an average of 200 random samples.

This means that we sampled a total of 50,000 (250 x 200) images and that some may repeat. This gives us an indication of how accurate our model was in predicting masks for that group. Each reading is a mean as by the central limit theorem that guarantees the underlying distribution will be normal. In a Gaussian distribution, we are estimating two parameters - the mean μ and the variance σ . The model is as a 2-parameter distribution, which means that both parameters are considered and modelled independently.

Based on preliminary tests, we have set the priors for these two parameters to be:

$$\mu_p \sim N(70, 20) \tag{1}$$

$$\sigma_p \sim U(0, 1) \tag{2}$$

In other words, we believe that our posterior distribution has a mean that is normally distributed around 70 with the standard deviation of 20. Unlike the mean, we have no prior opinion about the variance and we express that by setting the prior to be uniformly distributed.

With this we express that we believe that our posterior distribution is:

$$P_d = N(\mu_p, \sigma_p) \quad (3)$$

Based on these findings we then introduce a sample vector to our beliefs. The process considers our prior beliefs and "updates" them with actual data we provide it. The result is a posterior distribution of our data samples. We repeat the process for both groups and this results in two separate distributions which we can compare to each other. In this case, the intersecting area of distributions will tell us just how similar the intersection over union is. We may interpret this as confidence in these two distributions being the same if area is 1 (both distributions intersect in their entirety) and 0 if they are different (they do not intersect at all). For Bayesian analysis we have used R with the package Rstan [6].

4 EXPERIMENTS AND RESULTS

To evaluate the models, we have used the intersection over union metric which measures the total area that intersects between the ground truth and our predicted masks. For further reference, we will name the neural network that contains a mix of males and females in the training set as 7030NN, and the neural networks containing just females and males as FNN and MNN respectively.

4.1 Data

All ear images and ground truth masks were collected by students as part of the 2018/19 Image Based Biometry course at Faculty of Computer and Information Science, University of Ljubljana. The images contain photographs of famous people and were taken from the internet to ensure real-life

conditions with variable environmental conditions. Each image has an accompanying mask marking the location of an ear. We have used these rectangular masks to compare both the Haar cascade approach and the trained neural networks. The inputs were images of people and the outputs of the models were masks where white pixels denote the ear region and black pixels denote everything else.

4.2 Separation and model training

If we want to measure effects on groups of trained model predictions, it is important to properly split the data beforehand. This only applies to the neural networks as Haar cascades merely serve as some baseline considering we use the pre-trained Haar-like filters provided in the OpenCV [7] library.

Table 1: Model gender distribution and accompanying size of the training set

	<i>Female/Male dist</i>	<i># of training images</i>
Neural network 1 (7030NN)	70/30	10.214
Neural network 2 (FNN)	100/0	1.977
Neural network 2 (MNN)	0/100	5.984

Table 1 showcases three differently trained models and sizes of their training sets. While differences may occur due to this, we are mostly interested in how each model performs separately on each group. It is also important to note that some subjects in the first training set are the same as subjects in the other two. This is something we should take into account when evaluating the final performance of our models.

4.3 Detection Results

Table 2: Total intersection over union for all predicted cases

<i>Haar</i>	<i>7030NN</i>	<i>FNN</i>	<i>MNN</i>
0,255	0,338	0,236	0,333

Table 3: Relative number of samples where IoU for females is higher than the one for males (areas of non-intersection)

<i>Haar</i>	<i>7030NN</i>	<i>FNN</i>	<i>MNN</i>
0,939	0,2175	0,882	0,851

Table 2 showcases total accuracy of each trained model. FNN performs the worst, but that is most likely due to its smaller training set and not because it would favour one gender over the other. MNN performs only slightly worse than 7030NN even though its training set consists of only males and is half as large.

This does not indicate whether any model discriminates a certain group. To establish that, we have sampled an intersection over union vector for females and males separately for each model. After modelling this and calculating their posterior distributions, we arrive to plots shown on Figure 2.

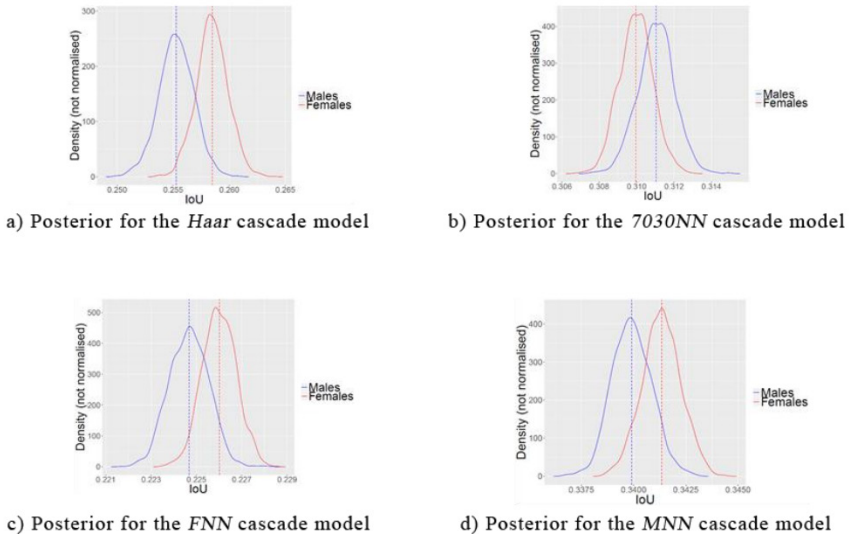


Figure 2: Posterior distributions for different model.

Results reveal that both groups are relatively similarly distributed, existing within less than 1% of each other. While practically this a negligible margin, it is statistically significant. In fact, all models perform better on one specific group with areas of non-intersection being above 75% (Table 3). While the distributions of 7030NN and FNN match the training set, this does not hold true for MNN. Even though the model was trained on only males, predicted masks for females seem to match them better. These results seem to indicate that there exists a small bias in a pre-trained network, which does not seem to be directly correlated to the training set but is rather a result of some other unaccounted factors.

5 CONCLUSION

In most models, selection of a proper and diverse training set is an important step to building a generalized system. In the case of ear segmentation, this seems to matter less as ears tend to be very similar across different genders. Our results seem to show that some inherent bias towards one group exists, but that it is practically negligible as these differences seem to be very minor. We have also shown that a model's success is also somewhat related to the size of its training set, regardless of the gender, as even the neural network that was trained on only males performed practically equally well to the one trained on both females and males. As most models tend to perform very slightly better on females it would be interesting to see whether that is due to some biological factor or just due to some inherent variance introduced in our procedure.

Bibliography

1. S. K. Singh, D. Chauhan, M. Vatsa, and R. Singh, "A robust skin color based face detection algorithm," *Tamkang Journal of Science and Engineering*, vol. 6, no. 4, pp. 227-234, 2003.
2. Ž. Emeršič, L. L. Gabriel, V. Štruc, and P. Peer, "Convolutional encoder-decoder networks for pixel-wise ear detection and segmentation," *IET Biometrics*, vol. 7, no. 3, pp. 175-184, 2018.
3. P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, 2001, vol. 1, pp. 1-1.
4. S. E. Umbaugh, *Computer Vision and Image Processing: A Practical Approach Using Criptools with Cdrom*. Prentice Hall PTR, 1997, p. 528

5. O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234-241.
6. *GitHub*. *U-Net*. Pridobljeno na <https://github.com/zhixuhao/unet>
7. *RStan*. *The R interface to Stan*. Pridobljeno na <https://mc-stan.org/users/interfaces/rstan>
8. *The OpenCV*. Pridobljeno na <https://www.opencv.org/>

Spletni sistem in javna podatkovna baza USOVA3D za nepristrano validiranje postopkov detektiranja jajčnih mešičkov na 3D ultrazvočnih posnetkih jajčnikov

JURIJ MUNDA, BOŽIDAR POTOČNIK, MILAN RELJIČ, KSENIJA RAKIČ,
JURE KNEZ, VELJKO VLAISAVLJEVIČ, GAŠPER SEDEJ, BORIS CIGALE,
ALEŠ HOLOBAR IN DAMJAN ZAZULA

Povzetek V tem prispevku predstavljamo spletni sistem in javno podatkovno bazo USOVA3D, namenjena nepristranemu ocenjevanju uspešnosti algoritmov detektiranja mešičkov na 3D ultrazvočnih posnetkih jajčnikov. Podatkovna baza vsebuje 44 zapisov, pri čemer vsak zapis sestoji iz ultrazvočnega posnetka jajčnika pacientke ter volumenskih datotek z označbami jajčnika in mešičkov. Vsak posnetek sta ločeno označila dva neodvisna eksperta ginekologa. Vsi podatki so popolnoma anonimizirani ter so shranjeni v volumenskih datotekah v formatu VTK. Spletni sistem zagotavlja dostop do podatkovne baze USOVA3D, hkrati pa natančno predpisuje protokol validiranja in nudi programske rutine za nepristrano vrednotenje algoritmov detektiranja mešičkov. Ta sistem tudi beleži in omogoča vizualiziranje rezultatov validiranja.

Ključne besede: • 3D ultrazvočni posnetki jajčnikov •
detektiranje jajčnih mešičkov • javna podatkovna baza •
nepristrano validiranje algoritmov • spletni sistem USOVA3D •

NASLOVI AVTORJEV: Jurij Munda, Univerza v Mariboru, Fakulteta za elektrotehniko, računalništvo in informatiko, Koroška cesta 46, 2000 Maribor, Slovenija, e-pošta: jurij.munda@um.si. dr. Božidar Potočnik, izredni profesor, Univerza v Mariboru, Fakulteta za elektrotehniko, računalništvo in informatiko, Koroška cesta 46, 2000 Maribor, Slovenija, e-pošta: bozidar.potocnik@um.si. dr. Milan Reljič, dr. med., izredni profesor, Univerzitetni klinični center Maribor, Ljubljanska ulica 5, 2000 Maribor, Slovenija, e-pošta: ivf.mb@mb.si. Ksenija Rakič, dr. med., asistentka, Univerzitetni klinični center Maribor, Ljubljanska ulica 5, 2000 Maribor, Slovenija, e-pošta: ivf.mb@mb.si. dr. Jure Knez, dr. med., docent, Univerzitetni klinični center Maribor, Ljubljanska ulica 5, 2000 Maribor, Slovenija. dr. Veljko Vlaisavljevič, redni profesor, IVF ADRIA consulting, Ljubljanska ulica 9, 2000 Maribor, Slovenija, e-pošta: vlai@ab-mb.si. Gašper Sedej, mag. inž., tehniški sodelavec, Univerza v Mariboru, Fakulteta za elektrotehniko,

računalništvo in informatiko, Koroška cesta 46, 2000 Maribor, Slovenija, e-pošta: gasper.sedej@um.si. dr. Boris Cigale, LogicData, Ulica Kneza Koclja 22, 2000 Maribor, Slovenija. dr. Aleš Holobar, redni profesor, Univerza v Mariboru, Fakulteta za elektrotehniko, računalništvo in informatiko, Koroška cesta 46, 2000 Maribor, Slovenija, e-pošta: ales.holobar@um.si. dr. Damjan Zazula, redni profesor, Univerza v Mariboru, Fakulteta za elektrotehniko, računalništvo in informatiko, Koroška cesta 46, 2000 Maribor, Slovenija, e-pošta: damjan.zazula@um.si.

1 UVOD

V lanskem letu je minilo 40 let, odkar se je rodil prvi otrok, ki ni bil spočet po naravni poti. Hiter razvoj postopka za oploditev z biomedicinsko pomočjo (OBMP) je posebej v preteklih dveh desetletjih omogočil tisočem neplodnih parov, da so postali starši svojim otrokom. Vse faze postopka so močno odvisne od ultrazvočnih pregledov ženskih jajčnikov, v katerih dozorevajo jajčni mešički z jajčeci, primernimi za oploditev.

Pravilno ovrednotenje jajčnih mešičkov je temeljni kamen za uspešno izpeljano OBMP. Jajčni mešički so sferične oblike in so napolnjeni s tekočino, ki ščiti jajčece. V jajčniku se razvijajo dalj časa, da s premerom 1 do 2 mm dosežejo t. i. antralno stopnjo. V naravnem ciklusu nato v nekaj dneh zrastejo do 5 mm in se prelevijo v dominante mešičke. Ti se povečujejo približno za 1 mm na dan in tik pred ovulacijo dosežejo velikost med 18 in 30 mm. Pri OBMP je običajno, da rast mešičkov stimulirajo s hormoni, kar povzroči hkratno zorenje več antralnih mešičkov.

Doziranje hormonov in spremljanje razvoja mešičkov je za OBMP ključnega pomena. Ultrazvočne preiskave opravljajo v razmikih od 1 do 3 dni in medicinski eksperti za ultrazvok morajo sproti analizirati množico ultrazvočnih posnetkov. Delo je zamudno in utrudljivo, saj traja vsaj 10 minut po opravljenem ultrazvočnem pregledu, pri 3D-ultrazvočnih posnetkih pa lahko še več.

Avtomatična detekcija jajčnih mešičkov v ultrazvočnih posnetkih je torej zelo dobrodošla, vendar pa mora biti po uspešnosti primerljiva z ekspertovimi odčitki. Avtomatični računalniški algoritmi se pojavljajo vzporedno z razvojem ultrazvočnih medicinskih snemalnik in izkoriščajo dejstvo, da se jajčni mešički pri delovanju naprav v B-načinu vidijo kot temna območja na svetlejši podlagi. Sprva so se avtomatične analize izvajale na ločenih računalnikih, danes pa so detekcijski programi praviloma vgrajeni že v ultrazvočne snemalnike.

Pred leti so ultrazvočne naprave omogočale le 2D-vpogled v človeško telo, tako da so medicinskemu ekspertu prikazale ravninski presek skozi tkivo. Ultrazvočna sonda odda zvočno valovanje, ki se razširi kot ozek snop v smeri, ki jo določa usmerjenost sonde. Različna struktura tkiv bolj ali manj vsrka ali pa odbija zvočno energijo. Odbiti valovi se vrnejo do sonde, in ker poznamo hitrost potovanja zvoka, lahko izračunamo razdaljo od sonde do reflektorja, torej telesne strukture, ki je valovanje odbila. Običajno vidimo na ultrazvočnih snemalnikih strukture, ki močno reflektirajo zvok (hiper odmevne), kot svetle točke, tiste pa, ki zvočno energijo vsrkajo (hipo odmevne), kot temne točke.

Zaporedje več 2D-presekov lahko sestavimo v prostorsko predstavitev preiskovanca, vendar z ročnim premikanjem sonde v tretji prostorski dimenziji ne moremo zagotoviti enakomerne in popolnoma vzporedne lege nanizanih 2D-slik. Pravilne kartezične razmere dosežemo lahko le s 3D-sodno, ki oddaja in lovi zvok v pravilni 3D-mreži. Najnovejše ultrazvočne naprave so v celoti prilagojene 3D-snemanju in vsebujejo računalniške algoritme, ki ekspertom pomagajo razpoznati iskane 3D-strukture.

V klinični praksi se je najbolj uveljavil detekcijski program SonoAVC, ki je vgrajen v ultrazvočnih snemalnikih firme General Electric in v 3D-ultrazvočnih posnetkih jajčnikov segmentira jajčne mešičke. Njegovo delovanje ni čisto avtomatično, saj mora ekspert najprej omejiti območje, v katerem išče mešičke, nato pa mora ročno izločiti tista območja, ki jih je program napačno označil kot mešičke, in pokazati na mešičke, ki jih je program spregledal. Čeprav že to prinaša pomembno pomoč, je vendarle moči pričakovati od naprednejših algoritmov še večjo uspešnost in predvsem več avtonomije [1].

Zato se mnogi raziskovalci trudijo izboljšati računalniško detekcijo jajčnih mešičkov [1, 3]. Vsak zase si pridobijo nabor ultrazvočnih posnetkov jajčnikov, v katerih medicinski eksperti najprej ročno označijo območja mešičkov. Te segmentacije služijo kot referenca za dokazovanje, kako uspešna je razvita računalniška analiza. A pri takšnem pristopu se soočamo vsaj z dvema problemoma:

- detekcijski rezultati, dobljeni z različnimi računalniškimi algoritmi, niso primerljivi, ker ne izhajajo iz istega nabora ultrazvočnih posnetkov in ročno določenih referenc;
- uporabljene so različne metrike za ugotavljanje, kako uspešno in natančno je algoritem razpoznal območja, ki pomenijo referenčne odčitke mešičkov.

Če želimo zagotoviti objektivno primerjavo uspešnosti različnih računalniških postopkov za razpoznavanje jajčnih mešičkov, morajo vsi uporabiti isto bazo anotiranih ultrazvočnih posnetkov. Ker takšne baze doslej raziskovalci še niso javno ponudili, smo se na Fakulteti za elektrotehniko, računalništvo in informatiko (UM FERI) odločili, da z eksperti za OBMP v Univerzitetnem kliničnem centru (UKC) v Mariboru zasnujemo in realiziramo takšno podatkovno bazo.

O njej in o možnostih, da do nje dostopamo s spletnimi storitvami ter da na ta način preverjamo tudi rezultate svojih detekcijskih algoritmov, govorimo v tem prispevku. Baza je v fazi nastajanja in je sestavni del spletnega sistema za nepristrano validiranje algoritmov za detekcijo jajčnih mešičkov v 3D-ultrazvočnih posnetkih jajčnikov.

2 PODATKOVNA BAZA ULTRAZVOČNIH SLIK JAJČNIKOV Z MEŠIČKI – USOVA3D

Podatkovno bazo 3D ultrazvočnih slik jajčnikov z referenčnimi označbami jajčnikov oz. mešičkov (v nadaljevanju jo označujemo z USOVA3D) sta konstruirala in jo dopolnjujeta ter vzdržujeta ekipa ginekologov iz UKC Maribor ter raziskovalci iz Laboratorija za sistemsko programsko opremo

(LSPO) iz UM FERI. UKC zagotavlja 3D ultrazvočne posnetke jajčnikov ter referenčne označbe, medtem ko LSPO zagotavlja infrastrukturo za hranjenje podatkovne baze ter orodja in protokol za označevanje jajčnikov in mešičkov.

Vsaka pacientka, vključena v raziskavo, je s pisnim privoljenjem dovolila uporabo in dodajanje njenih ultrazvočnih posnetkov v podatkovno bazo USOVA3D. Vsi podatki v bazi so popolnoma anonimizirani. Opisana raziskava ima vsa potrebna dovoljenja Komisije za medicinsko etiko, UKC Maribor.

Trenutna različica podatkovne baze USOVA3D vsebuje 44 označenih 3D ultrazvočnih posnetkov. Večina ultrazvočnih posnetkov je bila zajeta z ultrazvočno napravo Volusonom 730, nekaj z napravo Medison Accuvix XQ, novejši posnetki pa so zajeti z napravo Voluson 630. Vsi 3D ultrazvočni posnetki imajo prostorsko ločljivost 0,2 mm po vsaki osi oz. velikost voksla je 0,2 mm × 0,2 mm × 0,2 mm, se pa posnetki razlikujejo po velikosti. V podatkovni bazi so posnetki od velikosti 149 × 103 × 115 vokslov pa vse do 185 × 193 × 223 vokslov. Velikost posnetkov je po osi x med 101 in 229, po osi y med 89 in 193, medtem ko je velikost po osi z med 115 in 247.

En zapis v podatkovni bazi USOVA3D sestoji iz ultrazvočnega posnetka, označbe jajčnika ter označb jajčnih mešičkov. Vsak 3D ultrazvočni posnetek sta vedno ločeno obdelala ter označila dva neodvisna eksperta ginekologa iz UKC. LSPO je vzpostavil protokol in orodja za označevanje posnetkov. Vsa označevanja so potekala z orodjem ITK-SNAP, ki bere in izpisuje podatke v formatu VTK. Vse označbe so tvorjene volumsko in so shranjene v dveh ločenih volumskih datotekah tipa VTK. V prvi datoteki so voksli oštevilčeni z 0 za ozadje in z zaporednimi številkami za ločene mešičke (od 1 do števila mešičkov), v drugi pa imajo voksli ozadja vrednost 0, medtem ko so voksli, ki pripadajo jajčniku, vpisani z 1. Oba volumna se po velikosti popolnoma ujemata z originalnim volumnom oz. 3D ultrazvočnim posnetkom jajčnika.

Podatkovna baza USOVA3D trenutno vsebuje 44 zapisov. Vsak zapis torej sestoji iz petih datotek, in sicer: 3D ultrazvočnega posnetka, dveh datotek z označbami jajčnika (ena datoteka za vsakega označevalca) ter dveh datotek z

označbami jajčnih mešičkov (ena datoteka za vsakega označevalca). Vse datoteke so shranjene volumensko v formatu VTK.

Zavedamo se, da 44 zapisov za podatkovno bazo ni veliko, a ker je raziskava še v teku, pričakujemo, da bo velikost baze narasla vsaj na 100 zapisov.

3 SPLETNI SISTEM USOVA3D

Razvijamo spletni portal, ki bo sprva omogočal dostop do podatkovne baze s podatki o jajčnikih in mešičkih, kasneje pa bo nadgrajen z možnostjo nepristranskega validiranja uspešnosti postopkov za detektiranje jajčnih mešičkov. Portal bo omogočal rangiranje rezultatov validiranih postopkov, s čimer bo omogočena direktna primerjava med različnimi detekcijskimi postopki.

Prototipna verzija portala teče v okolju linux, a je načeloma neodvisna od operacijskega sistema in arhitekture. Uporabljena je kombinacija strežnika Apache, podatkovne baze SQL in sistema za upravljanje z vsebinami WordPress. Za validiranje se trenutno uporablja modificiran sistem za obdelavo posnetkov v oblaku [4].

3.1 Dostop do podatkovne baze

Dostop do podatkovne baze bo omogočen zgolj avtoriziranim uporabnikom. Uporabniki se v spletni portal prijavijo s pomočjo storitve Google Sign-On, lahko pa zaprosijo tudi za lokalno geslo. V razvoju je še dodatna možnost prijave preko Arnes AAI.

Ker gre za medicinske podatke, ki so bili zbrani v raziskovalnih projektih, morajo uporabniki najprej sprejeti raziskovalno licenco. Ker obstaja možnost za zlorabo uporabniških imen in s tem neupravičenega dostopa do podatkovne baze, želimo hraniti podatke o prijavah v portal ter prenosih vsebine, za kar v skladu z uredbo GDPR potrebujemo uporabnikovo privolitev. Pričakujemo, da portal vsaj na začetku ne bo beležil velikega obiska. Obstaja pa verjetnost, da bo obisk čez čas poskočil, zato želimo tudi statistično spremljati obisk

avtoriziranih uporabnikov na naši spletni strani, da bomo lahko ustrezno uravnavali sistemske vire za portal.

Uporabniki bodo bazo videli razdeljeno na dva dela. Prvi bo predstavljal t. i. učno množico, ki bo nudila dostop do vseh petih datotek, torej do 3D ultrazvočnega posnetka, dveh datotek z označbami jajčnika in dveh datotek z označbami jajčnih mešičkov (po ena datoteka za vsakega označevalca). Drugi del baze bo vseboval testno množico posnetkov, od katerih bo uporabnikom omogočen dostop le do 3D ultrazvočnih posnetkov, referenčne anotacije pa bodo služile za validiranje rezultatov, ki jih bodo uporabniki pošiljali v preverjanje. Uporabniki bodo tako lahko svoje algoritme za razpoznavanje jajčnih mešičkov razvijali s pomočjo učnega dela podatkovne baze, njihovo uspešnost pa bodo potem potrjevali z zapisi v testnem delu baze. Obseg obeh delov bomo z rastjo baze prilagajali glede na njeno velikost in vsebino.

3.2 Validiranje rezultatov

Portal bo sprva omogočal zgolj validiranje podatkov na naši strani (uporabniki bodo morali naložiti na portal rezultate algoritmov v predpisani obliki), za kar bo uporabljal že obstoječo infrastrukturo [4]. V kasnejšem razvoju bi radi implementirali še t. i. "prenosni" validacijski sistem, ki ga bi uporabnik prenesel na svoje računalnike, v njem zagnal validacijske teste, sistem bi rezultate digitalno podpisal, uporabnik pa bi jih nato naložil na naš portal.

3.3 Beleženje rezultatov

Pričakujemo, da se bo podatkovna baza občasno dopolnjevala z novimi podatki, zato se bodo rezultati validacijskih testov rangirali za vsako verzijo podatkovne baze posebej. Načrtujemo, da bo podatkovna baza dopolnjena približno enkrat letno.

V tabeli rezultatov bodo zbrani podatki o detekcijskem postopku (avtorji, ime postopka, DOI, če obstaja) ter njegovi uspešnosti. Objavili bomo vse merjene metrike. Postopki bodo razvrščeni glede na skupno uspešnost postopka.

4 PROTOKOL ZA VALIDIRANJE REZULTATOV

Kadar skuša več neodvisnih opazovalcev opisati dogodke ali stvari samo z dvema nasprotnima pojmom, lahko ujemanje njihovih ugotovitev dobro kvantificiramo z uveljavljenimi statističnimi indeksi. S tem pa seveda še ne izvemo, kdo od opazovalcev se moti in kdo ne. Če hočemo preveriti pravilnost njihovih zaznav, moramo poznati dejansko stanje oziroma t. i. absolutno resnico.

Pri razpoznavanju jajčnih mešičkov v ultrazvočnih posnetkih se srečamo s prav takšno situacijo. Opazujemo lahko le slike, na katerih se mešički pojavljajo praviloma kot temnejša območja na svetlejšem ozadju. Ultrazvočni posnetki vsebujejo celo vrsto različnih motilnih pojavov, ki lahko močno zabišejo jasne meje med mešički in okolico. Njihove dejanske oblike, velikosti in položaja torej na ta način ne moremo določiti absolutno natančno, ampak jih lahko opredelimo le s subjektivno oceno. To velja seveda tako za medicinskega strokovnjaka, ki označi vsebino ultrazvočnega posnetka, kot za vsak računalniški razpoznavalni postopek.

Baza USOVA3D z zbirkami ultrazvočnih posnetkov in njihovih anotacij je zasnovana dinamično. Sčasoma se bo število posnetkov v njej večalo. Tako bodo statistične primerjave, ki bodo temeljile na vpisih v bazi, postajale bolj nepristranske. Ne glede na to pa seveda ne bomo mogli vedeti, kako dobro se kakšna ocena ujema z absolutno resnico, ker te pri medicinskih ultrazvočnih posnetkih nikoli ni mogoče poznati brez invazivnih posegov, tj. biopsije.

Trenutno še vedno velja, da je medicinski izvedenec uspešnejši ocenjevalec struktur v ultrazvočnih posnetkih kot doslej znani računalniški algoritmi. Zato jemljemo ročne anotacije jajčnih mešičkov kot približke absolutni resnici: s primerjavo ocen več neodvisnih označevalcev izboljšamo statistično pomembnost in znižamo pristranskost. V bazi USOVA3D sta za vsak ultrazvočni posnetek jajčnika prisotni po dve neodvisni oznaki mešičkov in jajčnikov. Oznake se lahko razlikujejo tako po številu razpoznanih mešičkov kot po njihovi označeni obliki oz. velikosti. Ker gre za neodvisne oznake medicinskih izvedencev, jim lahko zaupamo tem bolj, čim bolj se med seboj

statistično ujemajo. Istočasno pa se lahko primerjava z njimi uporabi tudi za ugotavljanje uspešnosti računalniških postopkov, s katerimi želimo avtomatično odkrivati mešičke v ultrazvočnih posnetkih jajčnikov.

Predstavljamo merila, s katerimi bomo ugotavljali ujemanje anotacij vzporednih označevalcev ali detekcij iz računalniških postopkov z anotacijami označevalcev. Oboje bo vključevalo statistične indikatorje uspešnosti in točnosti razpoznav.

Ujemanje anotacij dveh neodvisnih označevalcev bo uporabljeno kot ocena za kakovost anotacij v podatkovni bazi. Ocena bo objavljena in se bo spreminjala z vsako spremembo v bazi, sestavljena pa bo iz statističnih meril za ujemanje anotacij in mer za ugotavljanje geometrijskih razlik med njimi.

Pri statistiki smo delno omejeni, ker ne moremo priti do podatkov o pravilno negativnih odločitvah, kar pomeni, da ne moremo ugotoviti, katere strukture v ultrazvočnem posnetku bi označevalec lahko napačno razpoznal kot mešiček, pa jih ni. Odvisni smo torej le od primerjav označenih območij dveh označevalcev, pri čemer enega izberemo za referenčnega, ali razpoznav računalniškega postopka z enim od označevalcev oz. s presekom oznak obeh označevalcev, kar se jemlje kot referenca.

Če se primerjano območje vsaj nekoliko ujema z enim samim referenčnim, to poveča število pravilno pozitivnih izidov (PP). Če primerjano območje ne sovpada z nobenim referenčnim, s tem poveča število lažno negativno odločitev (LN). Kadar primerjano območje prekriva več referenčnih, štejemo največje med referenčnimi k PP , ostale pa k LN . In nazadnje, če referenčno območje pokriva več primerjanih območij, štejemo največje med njimi kot PP za dano referenco, vse ostale pa kot lažno pozitivne (LP).

Iz danih vrednosti lahko izračunamo občutljivost O in natančnost N :

$$O = \frac{PP}{PP+LN}, \quad N = \frac{PP}{PP+LP}. \quad (1)$$

Gotovo pa ni pomembno samo, da so mešički zaznani, ampak da so območja, ki jih predstavljajo, čim točneje orisana. Primerno informacijo o tem posreduje

produkt $\rho_1\rho_2$, uveden v [2]. Dobimo ga tako, da primerjamo volumne, ki ustrezajo hkrati primerjanemu in referenčnemu mešičku. Vrednost ρ_1 primerja volumen preseka med označenima območjema istega mešička z volumnom referenčnega območja, ρ_2 pa volumen preseka z volumnom primerjanega območja oz. mešička. Čim bolj se njun produkt približa 1, tem bolj se ujemata območji dveh oznak istih mešičkov. Vrednost $\rho_1\rho_2 = 0,65$, recimo, pomeni 80-odstotno, vrednost $\rho_1\rho_2 = 0,86$ pa že 93-odstotno ujemanje območij.

Uvajamo še tri mere točnosti, in sicer i) razmerje med skupnim volumnom primerjanih mešičkov (V_p) in skupnim volumnom vseh referenčnih mešičkov (V_r):

$$r_V = \frac{V_p}{V_r}, \quad (2)$$

ii) povprečno evklidsko razdaljo med površinami primerjanih in referenčnih mešičkov:

$$\bar{e} = \frac{1}{M} \sum_{i \in (\text{vsi površinski voksl})} E(\mathbf{p}_p(i) - \mathbf{p}_r(i)), \quad (3)$$

pri čemer pomeni M število vseh ujemajočih se površinskih vokslav na ujemajočih se mešičkih, E operator za računanje evklidske razdalje, $\mathbf{p}_p(i)$ koordinate primerjanega in $\mathbf{p}_r(i)$ koordinate referenčnega i -tega voksla, ter iii) povprečno absolutno razliko med premeri ekvivalentnih sfer, ki volumsko ustrezajo primerjanim in referenčnim mešičkom:

$$\bar{d} = \frac{1}{N} \sum_{i \in (\text{vsi mešički})} |d_p(i) - d_r(i)|, \quad (4)$$

pri čemer pomeni N število vseh ujemajočih se mešičkov, $d_p(i)$ ekvivalentni premer primerjanega, $d_r(i)$ pa ekvivalentni premer referenčnega i -tega mešička.

Površinske vokse na mešičkih določimo tako, da od celotnega območja mešička odštejemo morfološko skrčeno območje tega mešička. Potem jih

uparimo, tako da se vsak iz površine primerjanega mešička poveže z najbližjim referenčnim vokslom.

5 SKLEP

V prispevku smo predstavili spletni sistem in javno podatkovno bazo USOVA3D, ki sta namenjena nepristranemu ocenjevanju uspešnosti algoritmov detektiranja mešičkov na 3D ultrazvočnih posnetkih jajčnikov. Pri razvoju spletnega sistema smo se zgledovali po uveljavljenih, javno dostopnih podatkovnih bazah s področja prepoznavanja obraznih značilnosti in človeških emocij. Natančno smo predpisali protokol za validiranje rezultatov, s čimer omogočamo "pošteno" in nepristrano medsebojno primerjavo detekcijskih algoritmov. Spletni sistem je v fazi prototipiranja, zato še pričakujemo določene spremembe v implementaciji.

Podatkovna baza USOVA3D je zasnova tako, da jo je mogoče enostavno periodično dopoljevati z novimi posnetki in označbami. Načrtujemo, da bomo do dvakrat letno podatkovno bazo dopolnili z novimi podatki iz UKC. Vzpostavljamo še protokol za dopolnjevanje podatkovne baze USOVA3D s podatki (tj. posnetki in označbami) drugih raziskovalnih skupin. Pri tem bomo zahtevali, da bodo nove označbe lahko dodajali zgolj relevantni eksperti. Pred dodajanjem podatkov v bazo USOVA3D bodo posnetke in označbe potrdili eksperti iz UKC.

V prihodnosti želimo spletni sistem nadgraditi in uporabiti tudi za tekmovanje algoritmov detektiranja mešičkov na 3D ultrazvočnih posnetkih jajčnikov.

Literatura

1. B. Cigale, D. Zazula (2019), Directional 3D Wavelet Transform Based on Gaussian Mixtures for the Analysis of 3D Ultrasound Ovarian Volumes, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, št. 1, str. 64-77, doi: 10.1109/TPAMI.2017.2780248.
2. B. Potočnik, D. Zazula (2001), Assessing the Efficiency of Segmentation Algorithms for Ultrasound Images, *Electrotech. Rev.*, vol. 68, št. 2-3, str. 97-104.
3. B. Potočnik, B. Cigale, D. Zazula (2012), Computerized detection and recognition of follicles in ovarian ultrasound images: a review, *Medical & biological engineering & computing*, vol. 50, št. 12, str. 1201-1212, doi: 10.1007/s11517-012-0956-y.

4. S. Šinjur, J. Munda (2014), Iskanje podobnih video posnetkov kot spletna storitev v oblaku, *Zbornik triindvajsete mednarodne Elektrotehniške in računalniške konference ERK 2014*, Zv. B, str. 64-67.

Javni sklad Republike Slovenije za podjetništvo (skrajšano Slovenski podjetniški sklad)
Ulica kneza Koclja 22
2000 Maribor
T: +386 (0)2 234 12 60
E: info@podjetniskisklad.si
www.podjetniskisklad.si

Slovenski podjetniški sklad (SPS) je javna finančna institucija Republike Slovenije, ustanovljena z namenom zagotavljanja boljšega dostopa mikro, malim in srednje velikim podjetjem do ugodnih finančnih virov na trgu.

Oblike finančnih spodbud SPS-a:

- 1. Garancije za zavarovanje bančnih kreditov s subvencijo obrestne mere** omogočajo MSP-jem hitrejšo, lažjo, predvsem pa cenejšo pridobivanje bančnih kreditov za obratna sredstva ali večje investicije. Bančni kredit zavarovan z garancijo SPS-a je za podjetja ugodnejši zaradi nižjih zahtev po zavarovanju, nižjih obrestnih mer, daljše ročnosti kredita in možnosti koriščenja moratorija pri vračilu kredita, stroški odobritve garancije pa se ne zaračunavajo.
- 2. Mikrokrediti** predstavljajo neposredne kredite SPSa manjših vrednosti, ki so namenjeni za manjše investicije in obratna sredstva ter na ta način omogočijo nemoteno tekoče poslovanje. Njihova pridobitev je enostavna in hitra, prav tako so zelo ugodni (nižja obrestna mera, možnost koriščenja moratorija, daljša ročnost) in brez stroškov odobritve.
- 3. Zagonske spodbude** so na voljo start up podjetjem oziroma tistim, ki svojo podjetniško pot še le začnejo. Namen tovrstne finančne linije je, da se spodbudi ustanovitev in zagon mladih podjetij z nepovratnimi oblikami spodbud, ki so najprimernejše za nastajajoča podjetja v zgodnji, najbolj ranljivi fazi.
- 4. Semenski in tvegan kapital** sta namenjena mladim visokotehnoškimi podjetjem s potencialom globalne rasti.
- 5. Posebne spodbude** so namenjene specifičnim ciljnim skupinam, pri katerih se izkazuje vrzel pri dostopu do potrebnih finančnih sredstev.

Spremljajte razpise in preverite ugodnosti posameznih produktov na:
http://www.podjetniskisklad.si/sl/razpisi_

Prijavite se na e-novice in obveščali vas bomo o novostih!





Univerza v Mariboru

Fakulteta za elektrotehniko,
računalništvo in informatiko

LSPO

Laboratorij za sistemsko
programsko opremo

<http://rosus.feri.um.si/rosus2019>
rosus.feri@um.si

