

UNIVERZA V LJUBLJANI
FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Juš Lozej

**Razpoznavanje šarenice z uporabo
globokega učenja**

MAGISTRSKO DELO

MAGISTRSKI PROGRAM DRUGE STOPNJE
RAČUNALNIŠTVO IN INFORMATIKA

MENTOR: izr. prof. dr. Peter Peer

SOMENTOR: izr. prof. dr. Vitomir Štruc

Ljubljana, 2018

AVTORSKE PRAVICE. Rezultati magistrskega dela so intelektualna lastnina avtorja in Fakultete za računalništvo in informatiko Univerze v Ljubljani. Za objavlanje ali izkoriščanje rezultatov magistrskega dela je potrebno pisno soglasje avtorja, Fakultete za računalništvo in informatiko ter mentorja.

©2018 JUŠ LOZEJ

ZAHVALA

Najlepša hvala vsem, ki ste mi pomagali, me vzpodbujali in spremljali v času moje učne poti. Imate mojo večno hvaležnost.

Juš Lozej, 2018

”Precej je treba vedeti, preden opaziš, kako malo veš.”

— Karl Heinrich Waggerl

Kazalo

Povzetek

Abstract

1	Uvod	1
2	Sorodna dela	5
2.1	Tradicionalni cevovod	5
2.2	Globoko učenje	9
3	Nevronske mreže in globoko učenje	13
3.1	Nevronske mreže	13
3.2	Globoko učenje	16
3.3	Najpogostejši računski sloji	16
3.4	Optimizatorji	22
3.5	Funkcije napake	23
3.6	Konvolucijske nevrnske mreže	24
4	Globok cevovod za razpoznavo šarenice	25
4.1	Dosedanje globoke rešitve	25
4.2	Predlagana arhitektura	26
5	Implementacija	35
5.1	Segmentacija	35
5.2	Razpoznavna	36

KAZALO

5.3	Podatkovne zbirke	36
5.4	Učenje	39
6	Eksperimenti in rezultati	43
6.1	Segmentacija	43
6.2	Razpoznavna	51
7	Sklepne ugotovitve	61
	Priloge	63
A	Spletno orodje za označevanje slik	63
A.1	Čelni del	63
A.2	Zaledni del	64
B	Dodatni grafi	67
	Literatura	75

Seznam uporabljenih kratic

kratica	angleško	slovensko
CNN	Convolutional Neural Network	konvolucijske nevrnske mreže
SVM	Support Vector Machine	metoda podpornih vektorjev
IDO	Integro-Differential Operator	intergralno-diferencialni operator
RSM	Rubber Sheet Model	model prožne pojave
DNN	Deep Neural Network	globoka nevrnska mreža
NM	Neural Network	nevronske mreže
ReLU	Rectified Linear Unit	usmerjena linearna enota
SGD	Stochastic Gradient Decent	stohastičen gradientni spust
ADAM	Adaptive Moment Estimation	prilagodljiva ocena zagona
MSE	Mean Squared Error	povprečna kvadratna napaka
DSC	Depthwise Separable Convolution	globinsko ločljiva konvolucija
ASPP	Atrous Spatial Pyramid Pooling	razširjeno prostorsko piramidno združevanje
TF	Tensorflow	knjižnica Tensorflow
CPU	Central Processing Unit	centralna procesna enota
GPU	Graphics Processing Unit	grafična procesna enota
IOU	Intersection Over Union	preseki unij
CMC	Cumulative Match-Score Curve	krivulja kumulativne podobnosti
TPR	True Positive Rate	frekvenca pravih pozitivov
FPR	False Positive Rate	frekvenca napačnih pozitivov
BE	Back-End	zaledni del
FE	Front-End	čelni del

Povzetek

Naslov: Razpoznavanje šarenice z uporabo globokega učenja

Kljub izjemni rasti globokega učenja v zadnjih letih, do sedaj še ni bil razvit globok cevovod za razpoznavo šarenice. V nadaljevanju predstavljamo splošno arhitekturo za razpoznavo šarenice navdihnjeno po strukturi tradicionalnega cevovoda za razpoznavo šarenice. Naš cevovod je zaključena konvolucijska nevronska mreža sestavljena iz dveh visoko-nivojskih gradnikov: segmentacije in razpoznavne. Z množenjem izhoda segmentacijskega dela model izloči področja, ki ne pripadajo šarenici in jih poda razpoznavi. Razpoznavni del iz šarenice izlušči značilke, katere uporabimo pri razpoznavi oseb. Naša metoda je na testnih podatkovnih zbirkah dosegla visoke rezultate. Na zbirki Casia-Iris-Thousand je dosegla natančnost prvega ranga 95,12 % in na zbirki SBVPI natančnost 92,33 %. Implementirali smo tudi med-podatkovni model, naučen na vzorcih obeh zbirk, ki je na sklopu zbirk dosegel natančnost 88,53 %. Naša metoda je presegla uspešnost in hitrost tradicionalnega cevovoda. Naš cevovod, kolikor vemo, predstavlja prvo implementacijo globoke nevronske mreže, ki znotraj svoje strukture segmentira področje šarenice in to nato razpozna. Za razliko od trenutnih pristopov naš razpozna osebo na podlagi izvorne nenormalizirane slike očesa.

Ključne besede

Razpoznavanje šarenice, globoko učenje, konvolucijske nevronske mreže, biometrija

Abstract

Title: Iris recognition using deep learning

Despite the large increase of deep learning solutions in recent years, no deep learning iris pipelines have yet been developed. Inspired by conventional iris recognition pipelines, we present our general deep architecture for iris recognition. The presented deep iris pipeline is an end-to-end convolutional neural network consisting of two high-level blocks: segmentation and recognition. The segmentation part is tasked with the generation of binary mask, which corresponds with the surface of the iris. These masks are multiplied with the original iris image and then fed to the recognition part. The recognition part extracts meaningful iris features, which are then used for matching. Our model achieved high results on both testing datasets. On Casia-Iris-Thousand it achieved a Rank-1 accuracy of 95.12% and on SBVPI an accuracy of 92.33%. We also implemented a cross-database model, trained on samples from both dataset, which achieved an accuracy of 88.53%. Our deep pipeline outperformed a conventional iris pipeline in speed and accuracy. As far as we are aware, our pipeline is the first implementation of an end-to-end deep neural network, which is able to segment and recognize the iris image. As opposed to current deep models, which perform recognition on a pre-normalized iris image, our method uses original iris images.

Keywords

Iris recognition, deep learning, convolutional neural networks, biometrics

Poglavje 1

Uvod

Razpoznavanje šarenice je eden izmed najbolj zanesljivih in natančnih pristopov identifikacije posameznikov na podlagi biometričnih modalnosti. Razpoznavanje šarenice je področje raziskav že več kot dvajset let. Prvi splošno namenski cevovod za razpoznavo šarenice, katerega strukturo še danes uporablja večina rešitev, je leta 1993 predstavil John Daugman [1]. Identifikacijski sistemi, ki delujejo na podlagi razpoznave šarenice, se uporabljajo na veliko obsežnih namestitvah. Primeri takšnih so letališče Schipol v Amsterdamu [2], sistem Aadhar v Indiji [3], meja med ZDA in Kanado [4] ter mnogo drugih.

Na področju računalniškega vida se je v zadnjih letih za reševanje različnih problemov izjemno povečala raba konvolucijskih nevronske mreže (v nadaljevanju imenovanih CNN) ter njihovih derivatov. Za začetek trenda ponavadi označimo arhitekturo AlexNet [5], s katero je leta 2012 avtor Alex Krizhevsky podrl rekord dotedanje klasične napake tekmovanja ImageNet in to za velik delež (iz 26 % na 15 %). Izjemno globoki CNN modeli (kot so ResNet [6]) z več 100 sloji so zmožni preseči uspešnost razpoznave človeka.

Kljub temu je raba globokih arhitektur na področju razpoznave šarenice še vedno zelo omejena. Dosedanje rešitve se predvsem osredotočajo na izgradnjo značilik [7, 8, 9] in nekaj na detekcijo šarenice [10, 11]. V času pisanja te naloge še nismo zasledili implementacije celotnega globokega cevovoda za razpoznavo šarenice. Avtorji navajajo, da je glavni razlog za to pomanjkanje

obsežnih podatkovnih zbirk.

Na podlagi opisanih ugotovitev smo zasnovali splošen globoki cevovod za razpoznavo šarenice navdihnjen po izvornem Daugmanovem cevovodu. Zasnovani cevovod je sestavljen iz dveh visoko-nivojskih gradnikov. Prvi del je posvečen detekciji oz. segmentaciji šarenice. Rezultati tega se združijo z izvorno sliko in pošljejo v drugi del. Drugi del cevovoda je namenjen razpoznavi šarenice. Ta zgradi nabor značilk in filtrov na podlagi katerih model razpozna šarenico. Zaradi omenjenih težav, ki jih prinašajo majhne podatkovne zbirke, smo pri implementaciji bili zelo pazljivi s številom učnih parametrov končne mreže. Želeli smo tudi prikazati moč globokega učenja pri povezovanju različnih vrst podatkov, zato smo pristop testirali na dveh zelo različnih podatkovnih zbirkah Casia-Iris-Thousand [12] in SBVPI [13]. Želeli smo tudi, da bi bile naučene značilke dovolj splošne, da bi lahko na podlagi njih tudi razpoznali posameznike, ki niso del učne množice, kar bi tudi pomenilo, da je naš model uporaben tudi za obsežnejše namestitve.

Magistrska naloga je vključno z uvodom razdeljena na 7 poglavij. Poglavje 2 predstavi sorodna dela. V njem je govora tako o dosedanjih globokih rešitvah kot tudi o klasičnemu cevovodu razpoznave šarenice. V poglavju 3 predstavimo koncepte nevronske mreže in globokega učenja. V njem predstavimo delovanje nevronske mreže, kaj je globoko učenje ter nekaj najpogostejših računskih slojev, optimizatorjev ter funkcij napake. Nato v poglavju 4 predstavimo našo metodo za razpoznavo šarenice. Poglavje začnemo s pregledom trenutnih globokih rešitev za razpoznavo šarenice ter predstavimo obstoječe arhitekture, ki smo jih uporabili v naši končni rešitvi. V poglavju 5 govorimo o implementaciji naše metode. Poglavje govori o segmentaciji, razpoznavi, v njem navedemo informacije o uporabljenih podatkovnih zbirkah in opišemo postopek učenja. Poglavje 6 govori o eksperimentih in rezultatih naše metode. Poglavje je razdeljeno na segmentacijo in razpoznavo. Vsako podpoglavje predstavi ocenjevalne metrike, protokol in na koncu rezultate posameznega dela cevovoda. V poglavju 7 nato zaključimo s sklepnimi ugotovitvami. Priloga A opisuje našo spletno aplikacijo za označevanje se-

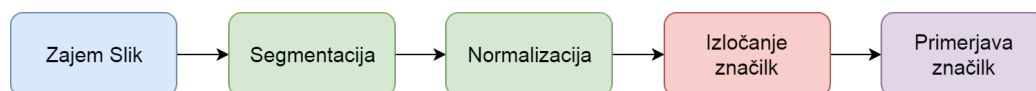
gmentacijskih mask šarenic. V priloga B se nahajajo dodatni grafi rezultatov.

Poglavje 2

Sorodna dela

2.1 Tradicionalni cevovod

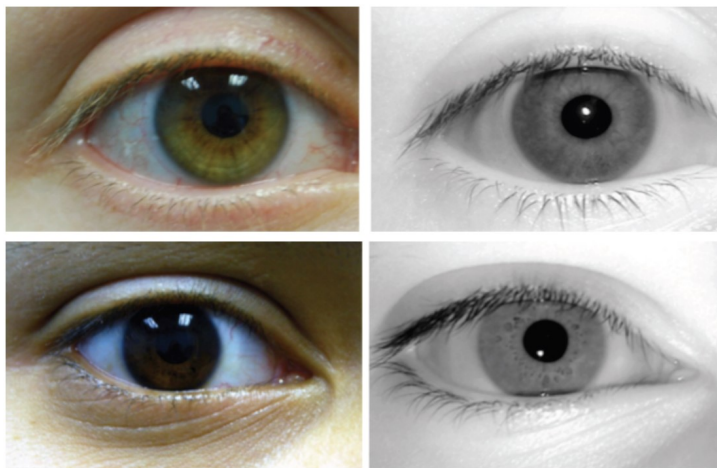
Prvi popolno avtomatiziran cevovod za razpoznavo šarenic (ilustriran na sliki 2.1) je leta 1993 predstavil John Daugman [1]. Cevovod je razdeljen na okvirno 5 korakov: zajem, segmentacijo, normalizacijo, izločanje značilnk in primerjavo. V koraku segmentacije cevovod zazna meje šarenice. Na podlagi lokalnega krožnega maksimuma določi lokaciji in radija mej šarenice. Nato z namenom minimizirati vpliv širjenja zenice, sistem šarenico normalizira. To doseže tako, da površino šarenice razvije na pravokotno ploskev. Iz normalizirane slike šarenice z uporabo filtrov Gabor zgradi binarno upodobitev šarenice, katere nato primerja z uporabo Hammingove razdalje. Struktura cevovoda še danes predstavlja osnovo za večino sodobnih sistemov za razpoznavo šarenice.



Slika 2.1: Shema cevovoda za razpoznavo šarenice.

2.1.1 Zajem slik

V tem koraku pridobimo slike očesa. Tukaj je pomembna predvsem kakovost slike, ki naj bodo idealno ostre, dobro osvetljene in z dobro vidno šarenico. Običajno se slika šarenice zajame pri bližnji infrardeči (angl. near-infrared) svetlobi. Na ta način se izravna vpliv svetlobe in različnih barv šarenice. Razlika je vidna na sliki 2.2. Sistemi za zajem slik ponavadi tudi zagotavljajo čim bolj konsistentno osvetljavo in pozo očesa.



Slika 2.2: Primerjava med slikami zajetimi z barvno in bližnjo infrardečo kamero [14].

2.1.2 Segmentacija

Namen segmentacije, včasih tudi imenovane lokalizacija ali detekcija, je določiti, katere slikovne točke slike pripadajo šarenici in katere ne. S tem zagotavljamo, da smo osebo identificirali le na podlagi šarenice in se pri tem tudi znebili elementov, ki bi lahko negativno vplivali na zaznavo. Pri segmentaciji najprej najdemo notranji in zunanji rob šarenice, nato zaznamo robove vek ter na koncu izločimo še trepalnice in svetleče odseve.

Integralno-diferencialni operator

Integralno-diferencialni operator (angl. Integro-differential operator, IDO), včasih imenovan tudi detektor krožnih robov, je funkcija, katero je Daugman [1] predstavil za namene iskanja robov šarenice. Funkcija išče koordinate maksimalnih vrednosti krožnega integrala glajenega radialnega odvoda izbrane slike. IDO predpostavi, da je na obeh robovih šarenice velik kontrast in da je šarenica krožne oblike.

Odstranjevanje veke

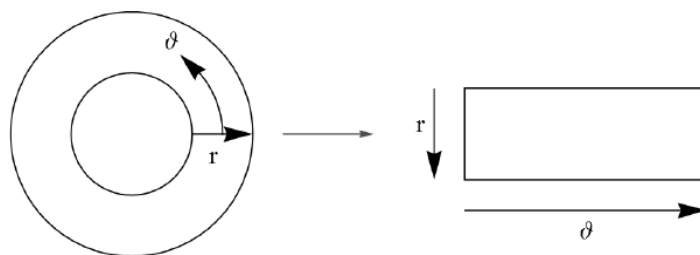
Z detektorjem paraboličnih robov lahko najdemo robove veke. Glede na smer ukrivljenosti parabole lahko ocenimo ali gre za spodnjo ali zgornjo veko, nato ustrezno izločimo slikovne točke nad ali pod njo.

Odstranjevanje trepalnic in svetlečih odsevov

Svetleče odseve se najlažje odstrani z upragovalno funkcijo (angl. thresholding), saj te predstavljajo najsvetlejši dele slike. Na isti način se lahko znebimo trepalnic, saj so te v bližnji infrardeči svetlobi zelo temne. Drugi pristopi odstranjevanja trepalnic uporabljajo detektorje skoraj vertikalnih črt, ki predstavljajo trepalnice.

2.1.3 Normalizacija

Normalizacija je postopek, ki pretvori slike v obliko, ki je neodvisna od položaja in razširjenosti zenice. Daugman v ta namen predlaga model prožne ponjave (angl. rubber sheet model, RSM). Ta krožno razvije površino šarenice na dvodimenzionalno površino, kot vidno na sliki 2.3. To doseže tako, da v radialni smeri enakomerno vzorči slikovne točke šarenice ter to ponavlja po celotnem obsegu očesa. S tem se znebimo razlik, ki so rezultat širjenja zenice. Pri razvijanju se tudi upošteva prej dobljena segmentacijska maska. Velikost dobljene normalizirane šarenice kontroliramo s številom vzorcev v smeri radija in v smeri tangente kroga.



Slika 2.3: Daugmanov model krožne ponjave [14].

2.1.4 Izločanje značilik

Normalizirano sliko se nato filtrira z naborom flitrov Gabor, kar pretvori posamezne slikovne točke v točke na kompleksni ravni. Faza teh točk se nato glede na kot kvantizira v binarna števila, katerih dolžina je odvisna od števila faznih kvantizacijskih polj. Torej v primeru, da kvantiziramo v štiri različna polja (0° - 90° , 90° - 180° , 180° - 270° , 270° - 360°), bodo naša dobljena števila 2 bitna. Števila nato združimo v eno samo zaporedje, ki predstavlja našo končno kodo šarenice. Gaborjevi filtri so skupina filtrov, ki simulirajo procesiranje človekovega vizualnega korteksa. Definirani so kot produkt sinusoide in Gaussove funkcije. Z različnimi frekvencami, smermi sinusoide ter standardnimi odkloni Gaussove funkcija lahko modeliramo različne filtre, ki so zmožni zaznati robove različnih orientacij in velikosti. Z linearno kombinacijo različnih filtrov lahko tako reproduciramo vrsto oblik.

2.1.5 Primerjava značilik

Dobljene binarne značilke primerjamo z uporabo Hammingove razdalje. Hammingova razdalja nam pove, kolikšen delež bitov dveh kod se razlikuje. Funkcija je tako omejena na interval $[0, 1]$. Pri primerjavi dveh kod se eno krožno zamika (angl. circular shift) in obdrži najmanjša razdalja z namenom, da najdemo rotacijo, ki najbolje poravnava dve zaporedji.

2.2 Globoko učenje

V zadnjih letih se je na področju računalniškega vida izjemno povečala uporaba globokega učenja, še posebej konvolucijskih nevronske mreže (angl. convolutional neural network, CNN). Trend je okvirno začel avtor Alex Krizhevsky z arhitekturo AlexNet [5], ki je na tekmovanju ImageNet podrl rekord dosedanje klasičarske napake (iz 26 % na 15 %). Kljub temu je do pred kratkim večina sistemov za razpoznavanje šarenice še vedno uporabljala ročno izdelane značilke [1, 15, 16] ali pa druge metode strojnega učenja [17, 18]. Prvi pristop za generiranje reprezentacij oziroma značilk šarenice z uporabo CNN [8] se je pojavil 2016. Avtorji so v članku predstavili dve mreži. Prva je predstavljala precej običajen CNN, druga je pa bila različica prve, pri kateri so del mreže nadomestili z moduli Inception [19]. Kot vhodne slike so avtorji uporabljali že vnaprej segmentirane in normalizirane slike šarenice. Za segmentacijo so uporabljali odprtokodni sistem Osiris [20], z uporabo modela prožne ponjave (angl. rubber sheet model) [1] pa so jih normalizirali. Kot predstavitev šarenic so uporabili izhod zadnjega polno povezanega sloja mreže. Mreže so ocenili na dveh zbirkah. Prvi sklop testov so naredili z vzorci iz zbirke, na kateri je bil model naučen, drugi na vzorcih obeh zbirk.

Avtorji članka [9] so arhitekturo predhodnega članka spremenili z namenom zmanjšati število učnih parametrov CNN, saj je zaradi manjšega števila učnih vzorcev globlje modele težje naučiti. Predlagali so tudi, da naj značilke predstavljajo bolj teksturne oz. oblikovne informacije površine šarenice kot pa samo izhod polno povezanega sloja. V ta namen so kot značilke uporabljali izhod zadnjega takšnega sloja. Velikosti posameznih filtrov posameznih konvolucijskih slojev so tudi povečali. Generirane značilke so z rabo ordinalne mere (angl. ordinal measure) binarizirali in jih primerjali z rabo Hammingove razdalje. Tako kot v prvem članku so na vходу uporabljali že normalizirane vzorce. Vzorce so segmentirali z uporabo pristopa avtorjev članka [21]. Njihova metoda uporablja kombinacijo gručenja (angl. clustering), transformacije Hough (angl. Hough transform) ter nabora različnih značilk za generiranje normalizirane slike ter maske šarenice.

V članku [7] avtorji predstavijo ogrodje za izločanje natančnih in splošnih značilk za razpoznavo šarenice. V delu predstavijo sklop mrež imenovanih UniNet. UniNet je razdeljen na dva dela: MaskNet in FeatNet. Obe mreži prejmeta na vhod že normalizirane slike šarenice. Pri tem avtorji normalizirane slike še pred-procesirajo z namenom, da povečajo kontrast površine šarenice. Naloga FeatNet je izločanje značilk. V svojem pristopu avtorji uporabljajo značilke iz različnih slojev FeatNet. Značilke poenotijo tako, da jih povečajo na enako velikost kot, je izvorna slika. Namen tega je sestaviti nabor značilk z različno grobimi predstavitvami izvorne slike. Enako kot pri prejšnjem članku jemljejo značilke iz konvolucijskih slojev mreže FeatNet. Naloga MaskNet je generirati grobe detekcije podane šarenice. To naredi na podoben način kot FeatNet, le da so izhodi verjetnostne distribucije, da posamezna slikovna točka pripada šarenici ali ne. Dobljene distribucije iz različnih slojev nato združijo, da prejmejo končno segmentacijsko masko. V delu so tudi predstavili posebno funkcijo napake imenovano razširjena napaka trojic (angl. extended triplet loss). Namen funkcije je usmeriti učenje na bolj teksturne/oblikovne značilnosti šarenice.

Avtorji članka [22] so predlagali, da se za izločanje značilk uporabljajo že obstoječe arhitekture in se naj ne razvija novih. V ta namen so izvedli teste na naboru znanih CNN arhitektur. V svojih testih so uporabljali že normalizirane in segmentirane slike šarenice. Izločene značilke so nato razpoznali z uporabo metode podpornih vektorjev (angl. support vector machine, SVM). Prišli so do zaključka, da se splošne arhitekture lahko uporabljajo za izločanje značilk in da je pri tem dobro uporabljati značilke iz vmesnih slojev modelov.

Nekateri avtorji so se lotili tudi segmentacije z uporabo globokega učenja. Prvi takšen pristop so avtorji predstavili v članku [11]. Avtorji članka so uporabil CNN, ki je bil namenjen razpoznavi obrazov, katerega so na slikah očeš doučili. Pri tem so uporabljali slike zajete pri vidni svetlobi.

Avtorji članka [10] so predstavili lastno globoko metodo za segmentacijo šarenice. V njem so predstavili relativno osnoven CNN z velikim poudarkom na predhodnih obogatitvah (angl. augmentation) podatkov. Z namenom,

da naredijo končno mrežo čim manj občutljivo, so učne slike spreminjali z naborom lastno razvitih obogatitev. Slikam so dodali umetne sence, jim reducirali kontrast ter jih razmazali, kot da bi se objekt gibal (angl. motion blur). Maske, na katerih so mrežo učili, so generirali z uporabo najmodernejših tradicionalnih segmentacijskih metod, vendar pri tem niso odstranili trepalnic in svetlečih odsevov.

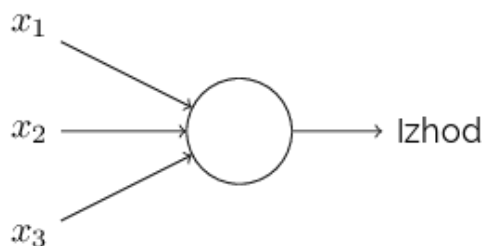
Poglavje 3

Nevronske mreže in globoko učenje

Zaradi velike uspešnosti pri reševanju raznolikih problemov, je v zadnjih letih izjemno naraslo število rešitev zasnovanih z globokimi nevronskimi mrežami (angl. deep neural network, DNN). Na različnih področjih so DNN izjemno izboljšale rezultate dotedanjih rešitev. Na področju računalniškega vida je postala različica DNN, imenovana konvolucijska nevronska mreža (angl. convolutional neural network, CNN), osnova večine najuspešnejših rešitev. V nadaljevanju bomo zato predstavili osnove nevronskih mrež ter globokega učenja (angl. deep learning) s poudarkom na konvolucijskih nevronskih mrežah.

3.1 Nevronske mreže

Nevronske mreže (v nadaljevanju NM) so računski sistemi namenjeni napovedovanju in klasifikaciji. NM sta leta 1943 zasnovala nevropsiholog in kibernetik Warren McCulloch ter logik Walter Pitts [23], ki sta želela zasnovati matematični model človeškega nevronskega sistema. Prvo implementacijo, imenovano Perceptron, je leta 1958 predstavil ameriški psiholog Frank Rosenblatt [24].



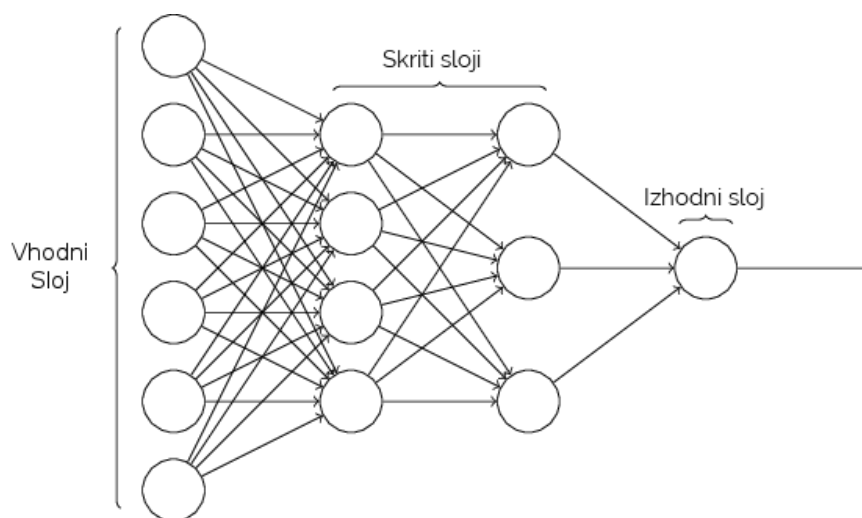
Slika 3.1: Preprosta shema nevrona [25].

Nevronske mreže, kot implicira ime, so sestavljene iz enot imenovanih nevroni. Vsak nevron (prikazan na sliki 3.1) sprejme enega ali več vhodov x_i , na podlagi katerega izračuna izhodno vrednost y . Vrednost izhoda se izračuna na podlagi enačbe (3.1).

$$y = \phi\left(\sum_i x_i w_i - b\right) \quad (3.1)$$

Pri tem w_i predstavljajo uteži, ki kontrolirajo pomembnost posameznega vhoda x_i , b pristranskost (angl. bias), ki določa nagib proti določenemu izhodu ter $\phi(\cdot)$ aktivacijsko funkcijo, ki definira nabor in obliko izhodnih vrednosti. Nekaj primerov aktivacijskih funkcij bomo tudi opisali v nadaljevanju. S spreminjanjem parametrov w_i in b lahko tako nadziramo izhod nevrona.

Nevrone organiziramo v strukture, sestavljene iz naborov med seboj povezanih nevronov, imenovanih sloji (angl. layers). Vsak nevron posameznega sloja je povezan z vsemi nevroni predhodnega sloja. Kako organiziramo in sestavimo strukturo mreže, imenujemo arhitektura (prikazano na sliki 3.2), instanca nevronska mreža z določeno nastavitvijo parametrov pa se imenuje model. Arhitekture so razdeljene na 3 glavne dele: vhodni sloj, skrite sloje in izhodni sloj. Vhodni sloj definira kakšne podatke bo mreža sprejemala. Na primer, da bi radi klasificirali črno-bele slike velikosti 32×32 slikovnih točk. V tem primeru bi vhodni sloj vseboval $32 \times 32 = 1024$ nevronov, enega za vsako slikovno točko. Izhodni sloj preslika rezultate predhodnega sloja v končni izhodni klasifikacijski vektor, torej če bi hoteli klasificirati katera



Slika 3.2: Shema arhitekture nevronske mreže [25].

števka se nahaja na sliki, bi imeli na izhodu 10 nevronov (eden za vsako števko). Skriti sloji so vsi sloji med vhomom in izhodom. Namen teh slojev je transformirati vhodne podatke v uporabne značilke (angl. features), na podlagi katerih lahko določimo izhodne razrede. Kako organiziramo skrite sloje tako najbolj vpliva na uspešnost končne mreže.

3.1.1 Učenje

Učenje je proces pri katerem iterativno spreminjamo uteži w_i in pristranskost b posameznih nevronov z namenom, da najdemo nabor parametrov, ki so zmožni najbolj natančno reševati definiran problem. V ta namen definiramo funkcijo, ki nam pove, kako uspešno mreža rešuje problem s trenutnimi parametri. Ta funkcija se imenuje funkcija napake (angl. loss function) in pove nam, kolikšno napako smo imeli pri napovedovanju vzorcev, torej več kot je napak, večja je. Funkcija napake primerja napovedan rezultat mreže z dejanskim pravim razredom posameznega vhodnega vzorca in pove kolikšen delež je nepravilnih. Pri učenju tako želimo na podlagi učnih vzorcev spreminjati učne parametre mreže, da minimizirajo funkcijo napake. To dosežemo

z uporabo gradientnega spusta po funkciji napake. Nekaj algoritmov za gradientni spust bomo opisali v poglavju 3.4. Učenje lahko izrazimo s funkcijo:

$$\Delta v = -\eta \nabla C \quad (3.2)$$

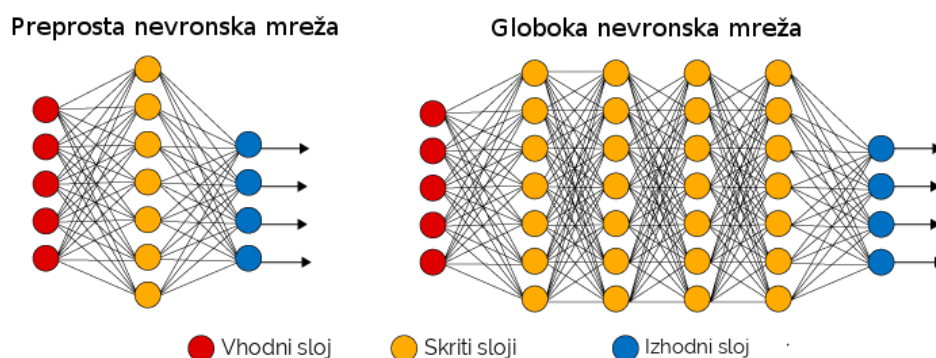
Pri tem je Δv sprememba skupka parametrov w_i in b vseh nevronov, ∇C gradientni vektor funkcije napake in η hitrost učenja, ki nadzira, za koliko se bodo učni parametri spreminjali.

3.2 Globoko učenje

Izraz globoko učenje se nanaša na učenje nevronske mreže z velikim številom skritih slojev. Take mreže se tudi imenuje globoke nevronske mreže (angl. deep neural network). Običajno se smatra vse nevronske mreže, ki imajo več kot en skriti sloj kot globoke (primer na sliki 3.3). Glavna prednost globokih nevronske mreže je modeliranje zelo kompleksnih nelinearnih odnosov med podatki. Zaradi visoke dimenzionalnosti učnega prostora je mreže zelo težko učinkovito učiti. V ta namen globoke mreže optimiziramo z uporabo porazdeljenega učenja. Zaradi visoke dimenzionalnosti je velika težava tudi izjemna sistemska zahtevnost, potreba po velikih podatkovnih zbirkah ter dolgi časi učenja. Vendar so se kljub temu globoke arhitekture izkazale kot uspešen pristop in so danes uporabljene v veliko, če ne večini, najsodobnejših metod v domenah računalniškega vida, razpoznavne govora ter obdelave naravnega jezika. Globoke nevronske mreže so ponavadi usmerjene (angl. feed-forward). Torej podatki potujejo od začetka do konca mreže brez kroženja znotraj njih. Različica nevronske mreže, ki vsebuje cikle, se imenuje povratna nevronska mreža (angl. recurrent neural network).

3.3 Najpogostejši računski sloji

V globokih nevronske mreže se vsaka matematična manipulacija smatra kot sloj. Torej ni nujno, da je vsak sloj učni, lahko je na primer samo



Slika 3.3: Primerjava med globoko in običajno nevronske mrežo.

transformacija med barvnimi prostori ali pa le množenje različnih vhodov. V nadaljevanju bomo predstavili nekaj najpogostejših slojev.

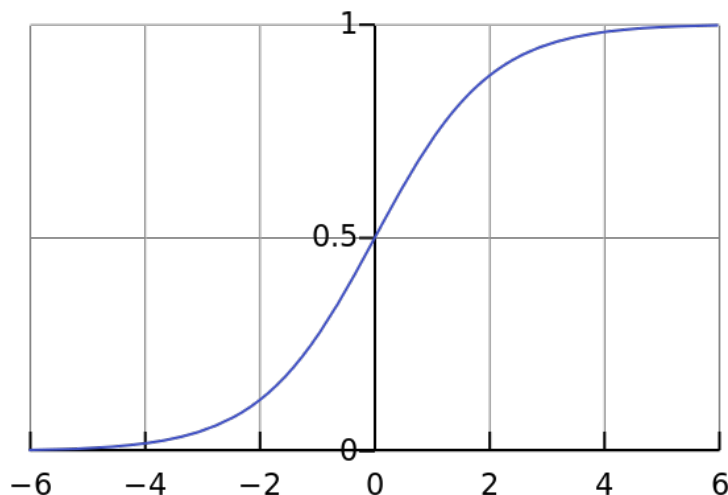
3.3.1 Aktivacijski sloj

Naloga aktivacijskega sloja je preslikati vhod z določeno aktivacijsko funkcijo $\phi(x)$. Naloga aktivacijske funkcije je oblikovati vrednosti izhodnih nevronov in pri tem dodajati nelinearnosti. Kot smo izvedeli v prejšnjih poglavjih, je izhod posameznega nevrona obtežena linearna kombinacija vhodov. Torej v primeru, da bi bile aktivacije vseh predhodnih nevronov linearne, bi lahko več slojev nadomestili z enim in odvod teh bi bil neodvisen od vhoda. Dodajanje nelinearnosti tako omogoča zlaganje večih slojev nevronov in s tem doseganje vse bolj kompleksnih značilk.

Sigmoid

Funkcija Sigmoid [24] je na videz (sliki 3.4) podobna zglajeni koračni (angl. step) funkciji. Definirana je z enačbo (3.3). Izhodne vrednosti so definirane med intervalom $[0, 1]$. Sigmoid se uporablja kot splošna aktivacija za nevronem, uporablja se pa tudi na izhodih segmentacijskih konvolucijskih nevronske mreže.

$$\phi(x) = \frac{1}{1 + e^{-x}} \quad (3.3)$$



Slika 3.4: Aktivacijska funkcija Sigmoid [25].

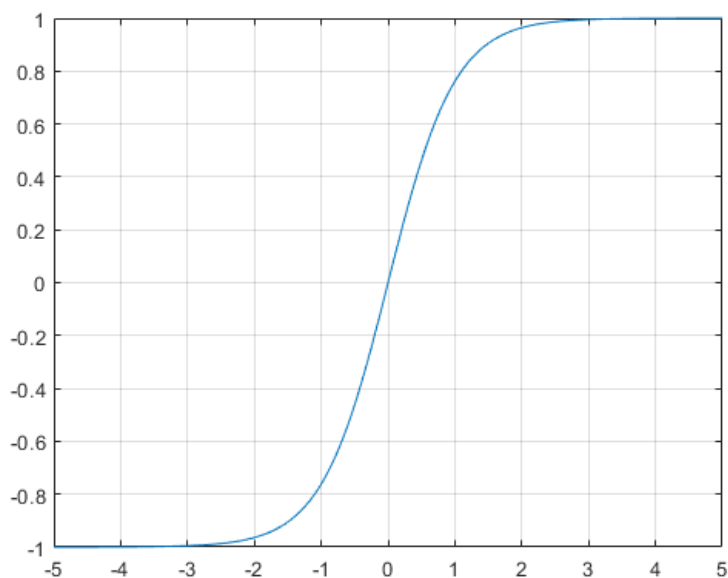
Tanh

Funkcija Tanh [26] (vidna na sliki 3.5) je transformacija funkcije Sigmoid. Največja razlika funkcije je, da omogoča negativne vrednosti in s tem seveda negativne izhode nevronov. Definirana je z enačbo (3.4).

$$\phi(x) = \text{Tanh}(x) = \frac{2}{1 + e^{-2x}} - 1 = 2 * \text{Sigmoid}(2x) - 1 \quad (3.4)$$

ReLU

Usmerjena linearna enota [27] (angl. Rectified Linear Unit, ReLU) je definirana z enačbo (3.5). Iz slike 3.6 lahko vidimo, da funkcija le obdrži pozitivne vrednosti, negativne pa nastavi na nič. Kljub linearnemu videzu zaradi ničelnega praga uvaja nelinearnosti v sistem. Funkcija tudi ni zgoraj omejena za razliko od Tanh in Sigmoid. Prednost ReLU je, da je zelo uspešen pri izklapljanju nepotrebnih nevronov, medtem ko Sigmoid in Tanh aktivirata



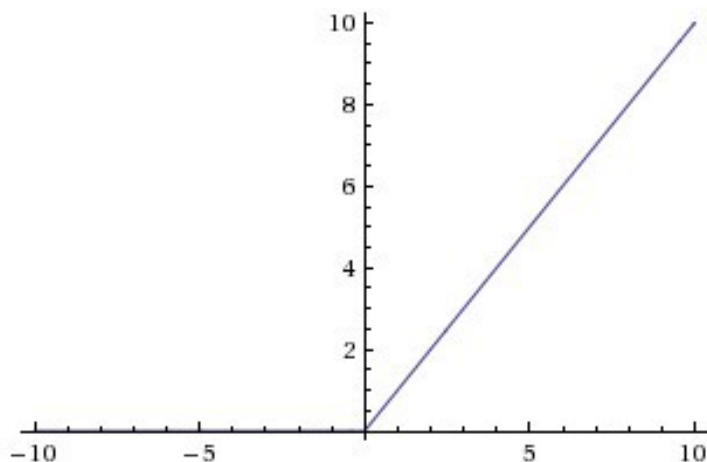
Slika 3.5: Aktivacijska funkcija Tanh [25].

skoraj vse nevrone (imajo neničelne izhode). ReLU je v primerjavi s Tanh in Sigmoid zelo računsko nezahteven. Največja težava ReLU je ničelni gradient negativnih vrednosti. Ker je gradient enak 0 se nevroni med gradientnim spustom prenehajo spreminjati. To težavo imenujemo problem umirajočih aktivacij (angl. dying ReLU problem). Težavo odpravimo tako, da namesto na 0 nastavimo negativne vrednosti na 0,01 in tako omogočamo majhne spremembe teh nevronov. Takšna aktivacija se imenuje puščajoči ReLU (angl. leaky ReLU).

$$\phi(x) = \max(0, x) \tag{3.5}$$

Softmax

Funkcija Softmax [28] je logistična funkcija, ki preslika realne vrednosti vektorja v vrednosti na intervalu $[0, 1]$, ki se seštejejo v 1. Funkcija pretvori vektor v normirano distribucijo svojega vhoda. Softmax se uporablja predvsem na izhodu nevronske mreže, da prejmemo verjetnosti posameznih izhodnih



Slika 3.6: Aktivacijska funkcija ReLU [25].

razredov. Izračunamo jo z enačbo (3.6).

$$\phi(\mathbf{x})_j = \frac{e^{x_j}}{\sum_{i=1}^D e^{x_i}} \quad (3.6)$$

Pri tem je \mathbf{x} vektor dolžine D , katerega sestavljajo elementi x_i .

3.3.2 Konvolucijski sloj

Konvolucijski sloj [26] (angl. convolutional layer) izračuna konvolucijo vhoda z notranjim naborom filtrov. Konvolucijski sloj je osnovni gradnik konvolucijskih nevronske mreže. Naloga sloja je se naučiti filtre, ki najboljše pripomorejo k uspešnosti zaznave. Parametri plasti so število filtrov, velikost filtrov in korak pri računanju odzivov filtrov.

Konvolucija

Konvolucija je matematična operacija na dveh funkcijah, ki nam pove, kako podobni so si segmenti teh dveh funkcij. V diskretnem primeru pokaže, kako podobni sta si dve sekvenci števil na določenem mestu. Dobljeno sekvenco podobnosti imenuje tudi odziv filtra na funkcijo. V našem primeru predstavlja ena funkcija sliko, druga pa filter. Ko sliko konvoliramo s filtrom,

nam rezultat pove, kateri del slike je najbolj podoben filtru. Izračunamo jo z enačbo (3.7).

$$(f * g)[n] = \sum_{m=-\infty}^{\infty} f[m]g[n - m] \quad (3.7)$$

3.3.3 Združevalni sloj

Združevalni sloj [26] (angl. pooling layer) je namenjen prostorskem združevanju vhodnih podatkov. Sloj podatke združi tako, da v drsnem oknu (angl. sliding window) računa povprečno ali maksimalno vrednost znotraj okna. Rezultat sloja je pomanjšanj vhod s povprečnimi ali maksimalnimi vrednostmi vseh oken. Glavni namen tega je zmanjšati dimenzionalnost vhoda in pri tem obdržati najpomembnejše informacije. Združevalni sloji imajo ponavadi nastavljivo velikost okna, korak okna (angl. stride) ter funkcijo združevanja (npr. maksimum, povprečje). Torej lahko z velikostjo okna in korakom kontroliramo, za koliko bo izhod manjši.

3.3.4 Polno povezani sloji

Polno povezani sloji [26] (angl. fully connected layer), včasih imenovani tudi gosti sloji (angl. dense layer), so običajni nevronske sloji, pri katerem so vsi nevroni novega sloja povezani z vsemi nevroni predhodnega sloja. Torej takšni sloji kot smo jih opisali v poglavju 3.1.

3.3.5 Izpustitveni sloj

Izpustitveni sloj [29] (angl. dropout layer) je sloj, katerega naloga je naključno izklapljanje nevronov predhodnega sloja. Namen tega je povečati rabo in splošno uporabnost vseh nevronov predhodnega sloja. Ker nevroni ne bodo mogli konstantno izkoriščati istih nevronov, jih bodo morali tvoriti iz večih naborov predhodnih nevronov in s tem povečali izkoriščenost vseh nevronov tega sloja. Rezultat tega so močnejše povezave med vsemi nevroni predhodnega sloja in s tem tudi boljše izhodne vrednosti. Koliko nevronov

predhodnega sloja se izklopi, lahko nastavimo s parametrom, ki kontrolira delež izklopljenih nevronov med vsako iteracijo.

3.4 Optimizatorji

Optimizatorji so algoritmi, ki v nevronske mreže izvajajo gradientni spust. Optimizatorji se razlikujejo v tem, kako prilagajajo hitrost učenja (angl. learning rate) z namenom, da se čim bolj učinkovito in uspešno najde rešitev ter pri tem ne obstojijo v slabših lokalnih minimumih.

3.4.1 Stohastični gradientni spust

Stohastični gradientni spust (angl. stochastic gradient decent, SGD) je različica običajnega gradientnega spusta, ki spreminja parametre po vsakem učnem vzorcu. Zaradi tega pa lahko med učenjem funkcija napake zelo niha, kar lahko oteži iskanje natančnega globalnega minimuma, se je pa zato, zmožen rešiti iz manjših lokalnih minimumov. Za bolj stabilno izvajanje algoritma je priporočljivo med učenjem manjšati hitrost učenja in je tako algoritem zmožen tudi najti boljše rešitve.

3.4.2 Adam

Prilagodljiva ocena zagona [30] (angl. Adaptive Moment Estimation, Adam) je metoda, ki adaptivno računa hitrost učenja za vsak parameter. Pri tem uporablja tudi eksponentno slabljeno povprečje predhodnih gradientov, da obdrži predhodno smer gibanja. Za razliko od ostalih algoritmov, ki delujejo na osnovi zagona, kovergira smer spusta optimizatorja Adam bistveno hitreje proti trenutnemu gradientu.

3.5 Funkcije napake

Funkcije napake nam povedo, kako uspešno je naša mreža naučena. Na podlagi napovedanih in dejanskih razredov izračuna, kolikšen delež napak je model naredil. Funkcije napake tako definirajo prostor, nad katerim izvajamo optimizacijo. Izbor pravilne funkcije tako omogoča učenje in željen izhod mreže.

3.5.1 Povprečna kvadratna napaka

Povprečna kvadratna napaka (angl. mean squared error, MSE) je ena izmed najbolj preprostih funkcij napake. MSE nam pove povprečno kvadratno razliko med napovedanim primerom p_i in pravim primerom y_i . Izračunamo jo z enačbo (3.8).

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (p_i - y_i)^2 \quad (3.8)$$

MSE se uporablja predvsem pri regresijskih problemih.

3.5.2 Križna entropija

Križna entropija (angl. cross entropy) ali logaritemska entropija je mera različnosti med dvema verjetnostnima distribucijama. Uporabljamo jo pri klasifikacijskih problemih, katerih izhod je verjetnostna distribucija razredov. Izračunamo jo z enačbo (3.9).

$$\text{CE} = - \sum_{c=1}^M y_{o,c} \log(p_{o,c}) \quad (3.9)$$

Pri tem predstavlja M število razredov, y binarni indikator, ki pove ali napoved o pripada pravemu razredu c , p je pa verjetnost, da napoved o pripada razredu c .

3.6 Konvolucijske nevronske mreže

Konvolucijske nevronske mreže (angl. convolutional neural network, CNN) so usmerjene različice globokih nevronskih mrež, namenjene procesiranju slik. Na področju računalniškega vida je CNN v zadnjih letih uporabljen v večini najbolj uspešnih rešitev.

CNN z zaporednimi konvolucijami slike izloči vizualne značilke, na podlagi katerih lahko razpoznajo objekte na njih. Glavna prednost konvolucijskih nevronskih mrež je, da se filtre, ki jih uporabljajo pri razpoznavi, naučijo na podlagi učnih vzorcev. Običajno so CNN sestavljeni iz večih skupkov konvolucijskih slojev zaključenih z združevalnim slojem. CNN so nato zaključeni s polno povezanim slojem in končnim klasifikacijskim slojem z aktivacijsko funkcijo Softmax, ki nam generira verjetnosti za posamezne razrede. Globlji kot je CNN, bolj bogate reprezentacije nosijo posamezni konvolucijski sloji.

CNN želijo modelirati človeški vizualni korteks, ki naj bi z linearno kombinacijo preprostih filtrov razpoznal objekte.

Poglavje 4

Globok cevovod za razpoznavo šarenice

V tem poglavju bomo predstavili naš globok cevovod za razpoznavo šarenice. Eden od osnovnih ciljev, ki smo si ga zadali pri tem magistrskem delu je, da bi bil naš cevovod v celoti implementiran znotraj ene mreže. Zasnovali smo arhitekturo zaključene konvolucijske nevronske mreže navdihnjene po strukturi izvornega Daugmanovega cevovoda. Naša arhitektura znotraj svoje strukture šarenico tako segmentira, kot tudi razpozna. V nadaljevanju bomo predstavili naš predlog zaključenega globokega cevovoda za razpoznavo šarenice, na kakšne načine so dosedanje globoke rešitve reševale problem razpoznavne in detekcij šarenice, kako smo njihova spoznanja v našem pristopu upoštevali ter kako se naš pristop razlikuje od njih. Predstavili bomo tudi nekaj znanih arhitektur, ki smo jih uporabili v naši končni implementaciji cevovoda.

4.1 Dosedanje globoke rešitve

Dosedanje rešitve, predstavljene v poglavju 2, se osredotočajo predvsem na uporabo globokih nevronskih mrež za izločanje robustnih reprezentacij šarenice, katere lahko nato uporabimo za primerjanje vzorcev. Zaradi pomanjkanja obsežnih podatkovnih zbirk so avtorji predlagali rabo bolj plitvih

arhitektur [9], ki so se zmožne naučiti na omejenem številu vzorcev. Tezo so avtorji članka [22] podprli s testiranjem različnih uspešnosti posameznih slojev znotraj znanih arhitektur. Avtorji člankov [7, 9] so tudi predlagali uporabo bolj geometričnih oz. teksturnih značilk. To so dosegli tako, da so kot značilke uporabljali binarizirane izhode konvolucijskih slojev. V nasprotju so avtorji člankov [8, 22] uporabljali bolj podatkovno usmerjen (angl. data-driven) pristop. Kot značilke so uporabljali sekvence števil pridobljenih iz polno povezanih slojev. V [22] so avtorji tudi predlagali uporabo že obstoječih arhitektur za namene razpoznavne. Vse dosedanje arhitekture za razpoznavo so na vhodu uporabljale normalizirane slike šarenic.

Na področju segmentacije so članki potrdili rabo obstoječih arhitektur [11]. Robustnost globokih segmentacij šarenice so avtorji članka [10] pokazali z učenjem z naborom različnih obogatitev. Kar se tiče predlaganih arhitektur, na tem področju ni bilo večjih novosti.

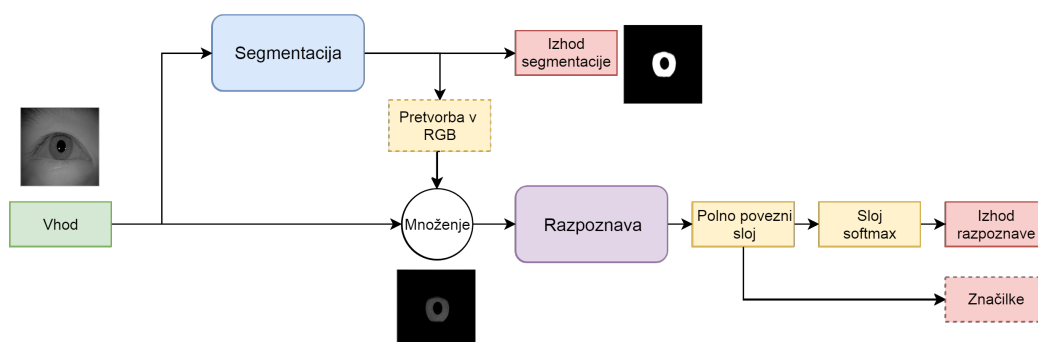
V času pisanja te naloge še niso obstajale implementacije zaključenih (angl. end-to-end) globokih cevovodov za razpoznavo šarenice. Najbližje temu so prišli avtorji članka [7], vendar so uporabljali že normalizirane slike šarenice in mreži nista bili zaključeni.

Glavni doprinos našega dela je predlog arhitekture za globok cevovod za razpoznavo šarenice, ki je zmožen segmentirati šarenico in jo razpoznati. Kot so predlagali avtorji [11, 22], smo pri tem uporabili znane arhitekture in pazili na to, da so te imele čim manjše število učnih parametrov. Od dosedanjih pristopov se tudi razlikuje v tem, da na vhod prejme izvirne, ne-normalizirane slike oči.

4.2 Predlagana arhitektura

Kot omenjeno v predhodnih poglavjih, predstavlja naša arhitektura splošno globoko arhitekturo cevovoda za razpoznavo šarenice. Shema arhitekture lahko vidimo na sliki 4.1. Cevovod ima en vhod in tri izhode. Na vhod prejme slike oči. Te so lahko zajete pri vidni ali bližnji infrardeči svetlobi.

Na vhodu bi lahko tudi uporabili že normalizirane slike, vendar v našem delu tega nismo testirali, ker smo želeli obdržati analogijo klasičnega cevovoda. Na izhodih vrača mreža segmentacijske maske (izhod segmentacije), verjetnosti posameznikov (izhod razpoznave) in značilke v primeru, da bi radi mrežo uporabljali kot izločevalec značilke (angl. feature extractor). Cevovod je razdeljen na dva visoko nivojska dela: segmentacijo in razpoznavo.



Slika 4.1: Shema predlagane arhitekture.

Naloga segmentacijskega dela je na podlagi slike očesa tvoriti črno-bele maske šarenice. Vrednosti slikovnih točk maske so omejene na intervalu med 0 in 1. Pri tem pomeni vrednost 1, da slikovna točka pripada šarenici in vrednost 0, da ne. Slikovne točke, za katere mreža ni prepričana ali pripadajo šarenici ali ne, zavzamejo vrednosti med 0 in 1. Ko dobljene maske točkovno pomnožimo z izvorno sliko, izločimo elemente, ki ne pripadajo šarenici. Segmentacijski del tako deluje kot model pozornosti (angl. attention model) za razpoznavni del. Če je naša podatkovna zbirka sestavljena iz slik zajetih pri vidni svetlobi, za segmentacijski del dodamo še pretvorbo v RGB prostor z namenom, da omogočimo pravilno množenje z izvorno sliko. Pri pretvorbi je vsak novi barvni kanal identičen izvornemu.

Razpoznavni del nato iz segmentiranih slik šarenice izloča značilke, na podlagi katerih je možno osebe razpoznati. Arhitektura se nato zaključi s polno povezanim slojem, katerega izhod lahko uporabimo kot značilko in slojem Softmax, ki izračuna verjetnosti, da vzorec pripada posamezni osebi. S

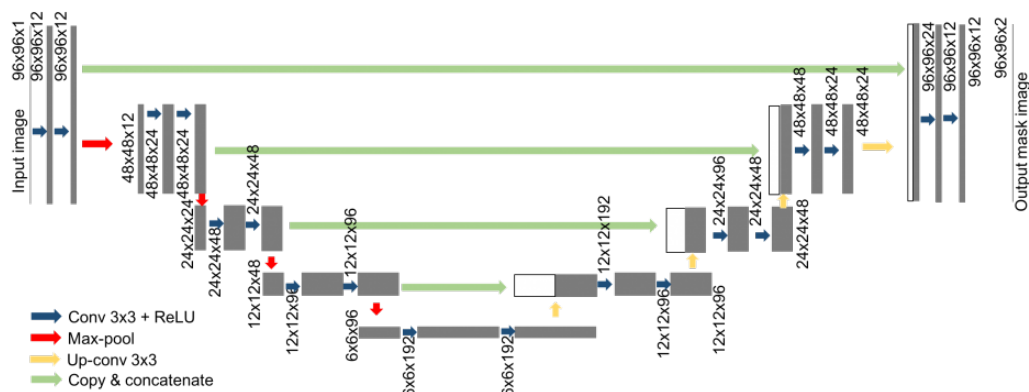
spreminjanjem velikosti Softmax sloja lahko tudi kontroliramo število razredov, ki jih vrača oz. na katerih se uči model.

Segmentacijski in razpoznavni del se lahko implementira s poljubno arhitekturo, pozorni moramo biti le na to, da so vhod in izhodi posameznih delov ustrezni. V nadaljevanju bomo predstavili nekaj znanih arhitektur, ki smo jih uporabili pri implementaciji našega cevovoda.

4.2.1 U-net

U-net [31] je ena izmed najbolj popularnih arhitektur namenjenih segmentaciji. Avtorji so mrežo razvili za segmentacijo biomedicinskih slik celic. Poleg tega je bila arhitektura uporabljana tudi v kopici drugačnih aplikacij [32, 33]. Glavna prednost tega modela je, da je zmožen doseči relativno natančne rezultate kljub manjšim podatkovnim zbirkam. Primarno smo želeli v naši končni implementaciji uporabiti model U-net, vendar se zaradi slabših rezultatov pri skupnem učenju segmentacije in razpoznave ter zaradi velikega števila parametrov, nismo odločili zanj. Kljub temu smo naše ugotovitve, kjer smo U-net uporabili za samostojno segmentacijsko metodo šarenice, objavili v članku [34].

U-net (ilustriran na sliki 4.2) uporablja arhitekturo koder-dekoder (angl. encoder-decoder). To pomeni, da je model razdeljen na dva nabora koderskih in dekoderskih konvolucijskih slojev. Koderski del uporablja običajno CNN arhitekturo, sestavljeno iz skupkov dveh 3×3 konvolucijskih slojev z ReLu aktivacijo, zaključenih z združevalnim slojem. Z vsakim zaporednim skupkom koder podvoji število značilk in zmanjša ločljivost za polovico. Dekoderski del nato razširi (angl. up-sample) značilke iz predhodnih slojev in jih združi z značilkami iz koderskega dela na enaki globini. S tem model obdrži različno grobe informacije koderskih slojev. Mreža je zaključena z 1×1 konvolucijskim slojem, ki generira končno segmentacijsko masko.

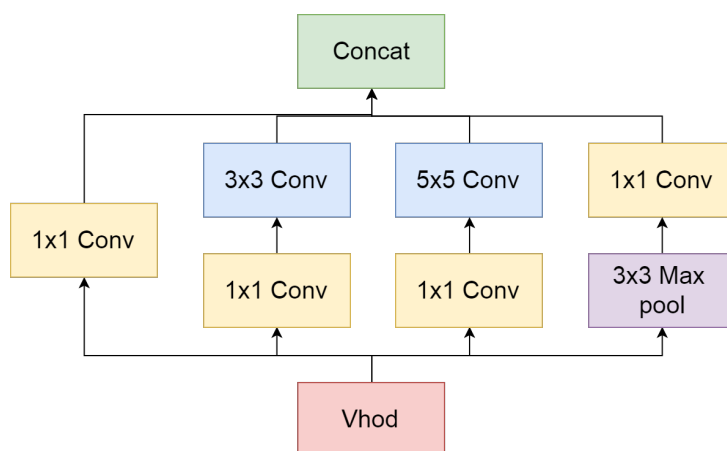


Slika 4.2: Shema arhitekture U-net z globino 4. Model uporablja arhitekturo koder-dekoder (angl. encoder-decoder). Z uporabo operacije kopiraj in poveži prenese informacije koderskih slojev na dekoderske [35].

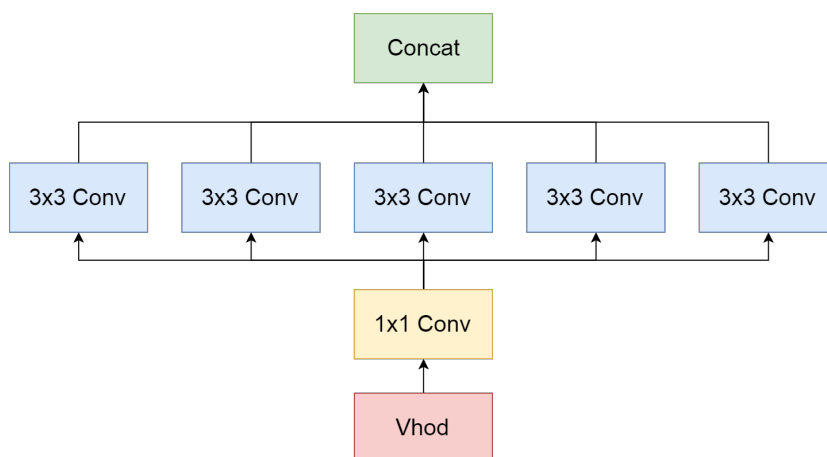
4.2.2 Xception

Xception [36] je arhitektura, razvita pri podjetju Google, namenjena razpoznavi objektov. Xception predstavlja nadaljnji razvoj njihove arhitekture Inception [37]. Inception uporablja za generiranje značilnik zaporedje večih modulov Inception. Modul Inception, prikazan na sliki 4.3, je sestavljen iz blokov različno velikih konvolucijskih slojev s predhodnimi 1×1 konvolucijskimi sloji. Poleg tega vsebuje modul tudi eno vejo, ki se začne z maksimalno združevalnim slojem. Filtri posameznih blokov so na koncu posameznega modula združeni. Namen posameznega bloka je razbrati različno grobe oblike in ker bloki niso vezani zaporedno, ostaja število parametrov relativno majhno. K nizkemu številu parametrov pripomore tudi začetni 1×1 konvolucijski sloj, ki zmanjša globinsko dimenzionalnost vhoda in pri tem obdrži izvorno informacijo. Torej moduli Inception razdelijo vhodne podatke v manjše podprostore, katere se želijo naučiti ločeno. Xception to delitev še razširi. Xception najprej izračuna prostorsko korelacijo na vsakem izhodnem kanalu ločeno, nato pa izvede še globinsko 1×1 konvolucijo, da zajame medkanalno korelacijo. Ta operacije se imenuje globinsko ločljiva konvolucija (angl. depthwise

separable convolution, DSC). Shema takšnega modula je ilustrirana na sliki 4.4. Operacijo si lahko razlagamo kot prvo iskanje korelacij v 2D prostoru, nato pa še iskanje teh po globini v 1D prostoru. Avtorji arhitekture predpostavljajo, da se je lažje naučit 2D in 1D korelacije ločeno kot 3D hkrati. Arhitektura je kljub podobnemu številu učnih parametrov izboljšala rezultate Inception.



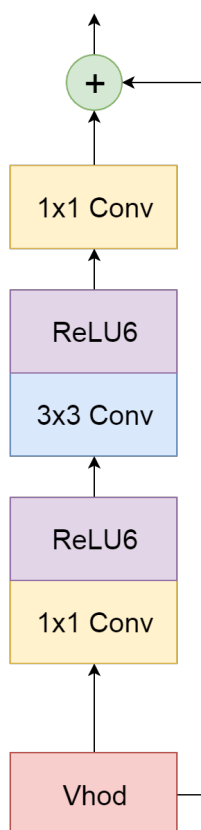
Slika 4.3: Shema modula Inception.



Slika 4.4: Shema modula Xception.

4.2.3 MobileNet

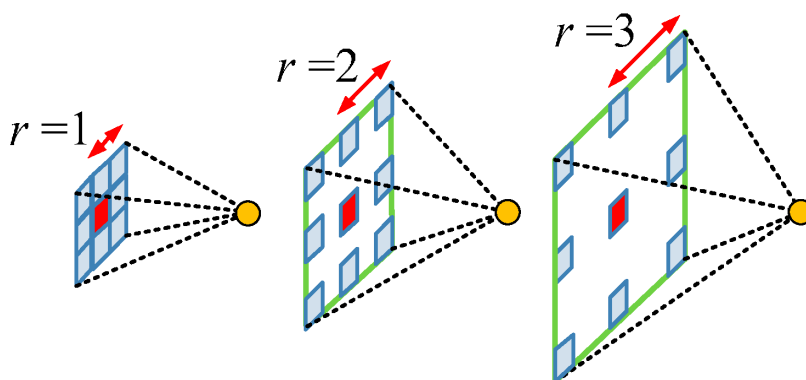
MobileNet [38] je globoka arhitektura, razvita pri podjetju Google, optimizirana za učinkovito izvajanje na mobilnih napravah. Njena glavna značilnost je izjemno nizko število učnih parametrov. Enako kot Xception ima MobileNet bločno strukturo, ki izkorišča globinsko ločljive konvolucije. Za razliko od Xception doda MobileNet na koncu svojega modula še en 1×1 konvolucijski sloj. Naloga tega sloja je stisniti izhod predhodnega sloja v manjši podprostor parametrov in tako učinkovito zmanjšati njihov nabor. Shema modula MobileNet je prikazana na sliki 4.5. K uspešnosti MobileNet prispeva tudi aktivacijska funkcija ReLU6. Funkcija ReLU6 je identična izvornemu ReLU, le da je ta omejen tudi zgoraj, in sicer na vrednost 6.



Slika 4.5: Shema modula MobileNet.

4.2.4 DeepLab

DeepLab [39] je Googlova najnovejša arhitektura za semantično segmentacijo. Naloga semantične segmentacije je posameznim slikovnim točkam dodeliti oznake, ki opisujejo vsebino tiste točke (npr. oseba, pes, drevo ipd.). Glavna značilnost arhitekture je raba razširjenih konvolucij (angl. dilated/atrous convolutions). Razširjene konvolucije, za razliko od običajnih, računajo odziv filtra na ne-sosednih slikovnih točkah, kot je ilustrirano na sliki 4.6. Z različnimi razmiki lahko filtri zajamejo večje površine slike, ne da bi povečali velikosti posameznih filtrov, kljub temu pa zajame podobno informacijo, kot bi jo večji filter.

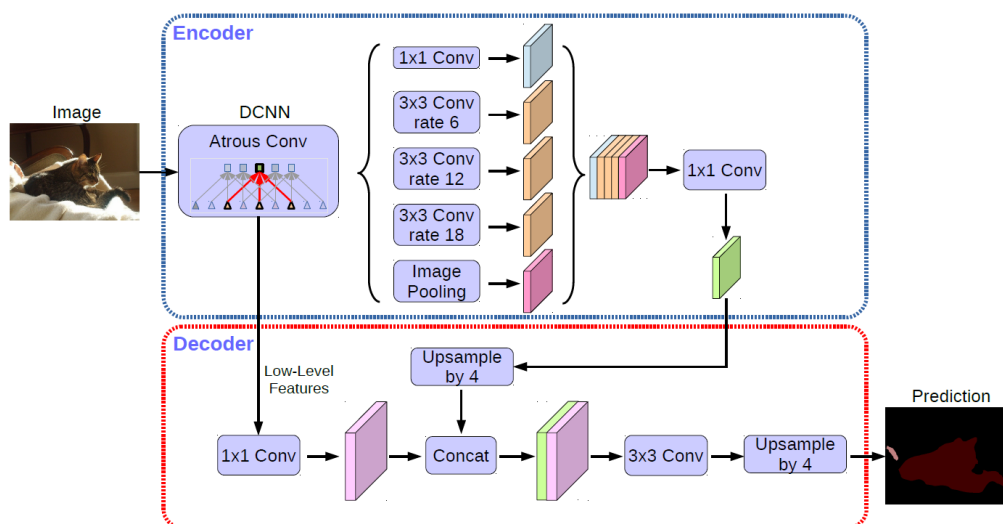


Slika 4.6: Primer izračuna razširjenih konvolucij. Odziv filtra se na slikah izračuna na razširjenem razmiku od središča. Ta razmik nastavimo s parametrom *rate*.

Za učinkovito združevanje značilk pri različnih velikostih so avtorji predlagali razširjeno prostorsko piramidno združevanje (angl. atrous spatial pyramid pooling, ASPP). ASPP uporablja nabor paralelnih razširjenih konvolucij z različnimi razmiki. Značilke, ki jih te zajamejo, se z uporabo polno povezanih slojev samostojno procesirajo in združijo.

V najnovejši različici DeepLabV3+ je arhitektura (ilustrirana na sliki 4.7) sestavljena iz dveh delov: koderja in dekoderja. Koder je sestavljen iz na-

bora konvolucijskih blokov in je zaključen z ASPP. Konvolucijski bloki uporabljajo razširjene konvolucije, da so zmožni zajeti podrobnosti na različnih velikostih. Organizirano strukturo teh blokov, imenujemo hrbtenica modela. Torej, če bi bloki vsebovali strukturo Xception bloka, bi takšno arhitekturo imenovali DeepLab s hrbtenico Xception. S parametrom izhodnega koraka (angl. output stride) lahko določimo, na kako majhne podrobnosti bo mreža pozorna. Z njim kontroliramo razmerje med prostorsko ločljivost slike in izhodne ločljivosti modela. Dekoderski del mreže združi značilke iz konca koderskega dela z nizko-nivojskimi značilkami konca hrbtenice. Te nato še procesira in poveča v končni izhod.



Slika 4.7: Shema arhitekture DeepLabV3+. Model je razdeljen na dva dela: koder in dekodek. Naloga koderja je z uporabo razširjenih konvolucij izločiti informacijo pri različnih velikostih, katere lahko dekodek izpopolni v končno segmentacijo.

Poglavje 5

Implementacija

Predstavljen cevovod smo implementirali v ogrodju Tensorflow [40] z visokonivojskim vmesnikom Keras [41]. Tensorflow (v nadaljevanju TF) je odprtokodno ogrodje za strojno učenje podjetja Google. TF omogoča razvoj v naboru programskih jezikov (Python, C++, Go, Java, JavaScript) in omogoča izvajanje tako na CPU kot GPU. Omogoča tako implementacije povratnih, kot tudi usmerjenih mrež. Pri implementaciji smo uporabljali arhitekturi podjetja Google Xception in DeepLabV3+. Glavni razlog za uporabo modelov je izjemno dobro razmerje med številom učnih parametrov in uspešnostjo modelov. Naš končni model je imel skupaj 29.494.217 parametrov od tega 2.196.033 na segmentacijskem in 27.298.184 na razpoznavnem delu. Zgrajeni modeli so bili okvirno veliki 320 MB. V nadaljevanju bomo predstavili, kako smo implementirali posamezna dela cevovoda in razložili, na kakšen način smo naš končni model naučili.

5.1 Segmentacija

Na segmentacijskem delu našega končnega modela smo uporabili arhitekturo DeepLabV3+. Uporabili smo implementacijo avtorja Bonlime [42], ki je pretvoril javno dostopno kodo izvirne Googlove implementacije iz Tensorflow v Keras. Implementacija omogoča uporabo dveh različnih hrbtenic:

Xception in MobileNet. Zaradi bistveno manjšega števila parametrov in zadovoljivih rezultatov, smo uporabili hrbtenico MobileNet. Implementacija omogoča tudi nastavitve različnih vhodnih tenzorjev, števila izhodnih razredov, izhodni korak ter parametrov posameznih hrbtenic. Model vrača za vsako slikovno točko logistično vrednost za posamezen razred. Ker smo v našem primeru potrebovali binarno masko, smo modelu nastavili en izhodni razred in tega procesirali z aktivacijskim slojem Sigmoid. Ostale parametre smo pustili privzete.

5.2 Razpoznavna

Na razpoznavnem delu smo uporabili Kerasovo implementacijo Xception, ki je verjetno zelo podobna izvorni, saj je François Chollet (avtor Xception) tudi avtor knjižnice Keras. Model omogoča nastavljanje vhodnega tenzorja in parametre izhoda modela. Z njimi lahko nastavimo, da konec modela predstavlja polno povezan sloj, izhod zadnjega konvolucijskega sloja ali pa maksimalno ali povprečno združen tenzor zadnjega konvolucijskega sloja. V našem primeru smo model nastavili, da vrača povprečno združene tenzorje zadnjega konvolucijskega sloja, katerega smo nato povezali v polno povezan sloj in tega zaključili s slojem Softmax z ustreznim številom nevronov.

5.3 Podatkovne zbirke

Podatkovne zbirke, ki jih uporabljamo pri globokem učenju, so ponavadi razdeljene na tri podmnožice: učno, potrjevalno in testno. Učna množica je nabor vzorcev zbirke, ki jih uporabljamo pri učenju. Potrjevalni del vsebuje nabor vzorcev, katere med učenjem uporabimo, da ocenimo, katere modele obdržati. Ta ne sme vsebovati učnih vzorcev. Po vsaki učni epohi testiramo trenutni model na vzorcih potrjevalne množice. S tem preverimo ali se je naš model naučil neko splošno rešitev ali je postal le poizvedovalna tabela učnih vzorcev. Ker se na podlagi potrjevalnih rezultatov odločamo, katere modele

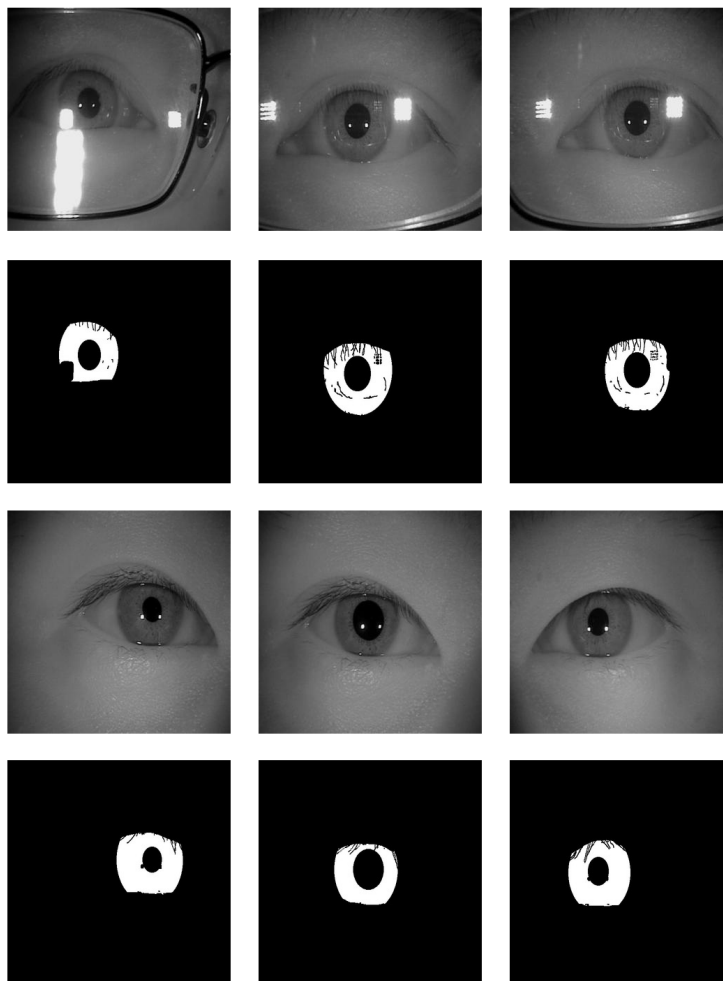
obdržimo, optimiziramo model tudi na njej. Da ne bi prišlo do prenasičenja (angl. *overfit*), je zato priporočljivo uporabiti še tretjo testno množico in na njej izvesti končne teste. Ta množica seveda ne vsebuje vzorcev učne in potrjevalne množice. Pri implementaciji in testih našega cevovoda smo uporabljali dve podatkovni zbirki: Casia-Iris-Thousand [12] in SBVPI [13]. V naslednjih podpoglavjih bomo predstavili splošne značilnosti posamezne zbirke. Kako smo zbirki razdelili na podmnožice ter ostale njune značilnosti, smo opisali v poglavju 6.

5.3.1 Casia-Iris-Thousand

Casia-Iris-Thousand (v nadaljevanju imenovana CASIA) je ena izmed javno dostopnih biometričnih zbirk kitajskega Centra za biometrijo in varnostni razvoj. Zbirka vsebuje slike oči zajetih pri bližnji infrardeči svetlobi s kamero IKEMB-100 podjetja IrisKing [43]. Kamera omogoča zajem obeh oči hkrati in nastavitev poze zajema posameznega očesa. Zbirka vsebuje 20.000 slik 1.000 posameznikov. Poleg informacij o posameznikih ima zbirka tudi slike razdeljene na levo in desno oko. Vsaka oseba ima tako 10 slik za vsako oko. Očala in svetleči odsevi v zbirki predstavljajo glavni izvor znotraj razrednih razlik. Ker zbirka CASIA ne vsebuje oznak šarenic, smo za namene učenja segmentacije ročno označili vzorce enajstih oseb (skupaj 220 vzorcev). Vzorce smo označevali z lastnim spletnim orodjem (opisanim v prilogi A), katerega smo tudi tekom te naloge za ta namen razvili. Segmentacijske maske imajo tudi filtrirane svetleče odboje in trepalnice. Nabor vzorcev zbirke CASIA lahko vidimo na sliki 5.1.

5.3.2 SBVPI

SBVPI je zbirka slik oči, ki se trenutno razvija na Univerzi v Ljubljani, katere namen je preučevanje razpoznavne na osnovi očesa, ne le šarenice. Zbirka vsebuje oznake žil beločnice, periokularne regije oči in šarenice. Iz tega tudi izhaja ime (angl. *Sclera Blood Vessels, Periocular and Iris, SBVPI*). Pri-



Slika 5.1: Primer vzorcev in oznak šarenice zbirke Casia-Iris-Thousand. Pod vsako sliko očesa se nahaja še naša oznaka šarenice.

mere vzorcev lahko vidimo na sliki 5.2. Podatkovna zbirka bo kmalu tudi javno dostopna na povezavi [13]. Podatkovna zbirka je razdeljena na dve podmnožici: vaskularni del in periokularni del. Za naše namene je prišel v poštev le vaskularni del. Vaskularni del vsebuje 1.959 barvnih slik, zajetih pri vidni svetlobi z zrcalno refleksno kamero Canon EOS 60D. Slike pripadajo 55-im različnim osebam. Glavna značilnost SBVPI je variacija smeri pogledov oči. Zbirko sestavljajo slike oči, ki gledajo direktno proti kameri ter levo, desno in nad njo. To naredi zbirko zelo zahtevno, saj na posameznih vzorcih šarenica ni v celoti razvidna. Še ena podrobnost, ki naredi zbirko precej zahtevno je, da jo sestavljajo slike različnih velikosti. Oznake regij šarenice so v SBVPI shranjene kot zelene regije preložene čez izvorno sliko očesa (sloj slike v programu GIMP). V ta namen smo pri naši rabi zbirke te regije predelali v binarne maske. Bolj podrobni opisi zbirke SBVPI se nahajajo v magistrskem delu Petra Rota [44], ki nam je omogočil dostop do zbirke.

5.4 Učenje

Učenje je potekalo v dveh korakih. V prvem koraku učimo segmentacijo. Segmentacijo učimo na način, da bi čim boljše pripomogla k uspešnosti razpoznave. To dosežemo tako, da učimo segmentacijo in razpoznavo hkrati. Pri učenju tako uporabljamo oba izhoda, vsakega s svojo funkcijo napake. Predpogoj za tak način učenja je, da imamo za vsak učni vzorec tako oznako šarenice kot tudi identiteto osebe. Da bo učenje uspešno, moramo imeti tudi dovolj vzorcev za posamezno osebo. Ker pri zbirki SBVPI tega nismo imeli, smo segmentacijo učili popolnoma ločeno. Torej brez hkratne razpoznave.

V drugem koraku smo učili razpoznavo. V tem koraku smo razpoznani del do-učili na večjemu številu razredov in vzorcev. Vzeli smo uteži naučene v segmentacijskem koraku in zamrznili uteži segmentacijskega dela. S tem smo onemogočili spreminjanje parametrov segmentacijskega dela in zagotovili, da se model uči le na informaciji šarenice. Primerno smo tudi spremenili število razredov na koncu mreže. Zaradi majhnega števila oseb, model po



Slika 5.2: Primer vzorcev in oznak šarenice zbirke SBVPI. Pod vsako sliko očesa se nahaja še oznaka šarenice.

prvem koraku ni mogel tvoriti dovolj opisnih značilk. V primeru, da bi imeli označene šarenice vseh oseb, bi bil ta korak nepotreben, saj bi lahko tudi razpoznavni del mreže uspešno v celoti naučili.

Pri obeh korakih smo se na podlagi vrednosti funkcije napake na potrjevalni podmnožici posameznih zbirk odločali, katere modele obdržati. Pri tem smo shranili nabor najboljših modelov, katerih rezultate smo primerjali v končnih eksperimentih. Učenje smo prekinili ročno ali če se funkcija napake na potrjevalni množici po določenem številu učnih epoh ni več izboljševala. Pri učenju smo pri posameznih epohah uporabljali manjše podmnožice učne in potrjevalne množice. Te množice so bile pred vsako epoho premešane, da so vsebovale bolj reprezentativne vzorce celote. Namen tega je bil zmanjšati direktno nasičenost na celotno potrjevalno množico. Pri obeh korakih smo uteži inicializirali z naključnimi vrednostmi. Podrobnosti učenja, kot so izbor optimizatorjev, funkcij napake ipd., so opisane v poglavju 6.

Poglavje 6

Eksperimenti in rezultati

V tem poglavju bomo predstavili rezultate naše metode. Poglavje je razdeljeno na dva dela. Prvi je posvečen oceni segmentacije, drugi oceni razpoznave. V njih bodo predstavljene metrike, na podlagi katerih smo ocenili naš model, značilnosti rabe posameznih podatkovnih zbirk, značilnosti postopka učenja ter evalvacija in rezultati posameznih eksperimentov.

6.1 Segmentacija

6.1.1 Metrike

Pri ocenjevanju segmentacije smo uporabljali 5 metrik: natančnost (angl. accuracy), presek unij (angl. intersection over union, IOU), točnost (angl. precision), priklic (angl. recall) ter krivuljo točnosti in priklica (angl. precision recall curve).

Natančnost

Natančnost je mera, ki nam pove delež pravilno napovedanih rezultatov. Pri segmentaciji nam mera pove, kolikšen delež slikovnih točk je model pravilno označil. Torej večja kot je, bolj uspešno označuje naš model področje

šarenice. Izračuna se jo z enačbo (6.1).

$$\text{Natančnost} = \frac{TP + TN}{N} \quad (6.1)$$

Pri tem je TP (angl. true positive) število pravilno napovedanih pozitivnih primerov (število pravilno označenih slikovnih točk, ki pripadajo šarenici), TN (angl. true negative) pa število pravilno napovedanih negativnih primerov (število pravilno napovedanih slikovnih točk, ki ne pripadajo šarenici) in N število vseh vzorcev, v našem primeru število vseh slikovnih točk.

Presek unij

Presek unij (angl. intersection over union, IOU) je razmerje med številom vzorcev preseka in unijo oznake šarenice in napovedane segmentacije. Pove nam, kako natančno napovedana segmentacija prekriva oznako šarenice. IOU se izračuna z enačbo (6.2).

$$\text{IOU} = \frac{TP}{TP + FP + FN} \quad (6.2)$$

Pri tem je FP (angl. false positive) število napačno napovedanih pozitivnih vzorcev (slikovne točke označene kot šarenica, vendar ji ne pripadajo), FN (angl. false negative) pa število napačno označenih negativnih vzorcev (slikovne točke, ki pripadajo šarenici, vendar niso bile označene s strani modela).

Točnost

Točnost nam pove, kolikšen delež pozitivnih vzorcev je dejansko pozitivnih. Torej z mero izvemo, koliko napovedanih slikovnih točk šarenice dejansko leži na področju šarenice. Izračuna se jo z enačbo (6.3).

$$\text{Točnost} = \frac{TP}{TP + FP} \quad (6.3)$$

Priklic

Komplementarna mera točnosti je priklic. Priklic pove, kolikšen delež vseh pozitivnih primerov je model označil. Torej iz njega izvemo, kolikšen delež šarenice model ni označil. Priklic izračunamo z enačbo (6.4).

$$\text{Priklic} = \frac{TP}{TP + FN} \quad (6.4)$$

Krivulja točnosti in priklica

Ker segmentacijsko masko sestavljajo zvezne vrednosti med nič in ena, se njena uspešnost razlikuje v odvisnosti od vrednosti praga, pri katerem te vrednosti binariziramo. Pri določanju praga si lahko pomagamo s krivuljo točnosti in priklica. Krivulja je sestavljena iz vrednosti točnosti in priklica pri različnih pragih. Pri tem predstavlja vertikalno komponento točnost in horizontalno priklic. Na podlagi krivulje lahko določimo prag, ki ima najboljše razmerje med priklicem in točnostjo ali ga pa nastavimo tako, da daje prednost posamezni meri. Kot mero uspešnosti lahko tudi uporabljamo ploščino, ki se nahaja pod krivuljo.

6.1.2 Eksperimentalni protokol

V naših eksperimentih segmentacije smo uporabljali dve podatkovni zbirki: CASIA-Iris-Thousand (v nadaljevanju imenovana CASIA) ter SBVPI, obe pa smo tudi kombinirali in tako dobili med-podatkovno (angl. cross-database) zbirko. Pri učenju smo na vseh zbirkah uporabljali optimizator Adam (poglavje 3.4.2) s hitrostjo učenja $\eta = 10^{-4}$. Slike smo poenotili tako, da smo jih zmanjšali na velikost 320×320 . Med učenjem smo vzorce vhoda in izhoda v realnem času tudi obogatili. Vzorce smo premikali za $\pm 20\%$ svoje velikosti, jih približevali za 20% ter jih rotirali za $\pm 20^\circ$. Z uporabo obogatitev smo tako umetno povečali število učnih vzorcev in model naredili manj ranljiv na navedene transformacije. Naučene modele smo shranjevali na podlagi potrjevalne funkcije napake. Od skupaj okvirno 40 modelov, smo obdržali 5 najboljših za posamezno podatkovno zbirko, katere smo nato uporabljali

pri testih. Učenje smo ustavili ročno ali pa če se za 15 učnih epoh funkcija napake ni izboljšala. Pri učenju segmentacije zaradi že majhnega števila vzorcev nismo uporabili ločene testne zbirke, ampak le potrjevalno.

CASIA

Zbirka CASIA, katere vzorce in opis lahko vidimo v poglavju 5.3.1, vsebuje 220 vzorcev pripadajočih 11 osebam, sestavljenih iz slike očesa in oznake šarenice. Slike oči so zajete pri bližnji infrardeči svetlobi. Od 220 vzorcev smo jih 154 uporabili za učenje in 66 za potrditev in teste. Slike oči v tej zbirki primarno gledajo proti kameri in glavne ovire predstavljajo trepalnice in sijoči odboji, ki so tudi izločeni iz anotacij šarenice.

Ker podatkovna zbirka vsebuje tako segmentacijske maske kot identitete oseb, smo na zbirki učili segmentacijo in razpoznavo hkrati. V našem primeru smo na segmentacijskem izhodu uporabljali binarno križno entropijo in na razpoznavnem izhodu kategorično križno entropijo. Pri tem smo večjo težo (0,7) dodelili segmentacijskemu izhodu. Tak način učenja prilagodi segmentacijske maske, da čim bolj pripomorejo k razpoznavi.

SBVPI

Zbirka SBVPI, katere vzorce in opis lahko vidimo v poglavju 5.3.2, vsebuje 332 vzorcev, od katerih smo jih 274 uporabili za učenje ter 58 za potrditev. Slike zbirke SBVPI so različnih dimenzij in vsebujejo nabor različnih pogledov posameznega očesa. Vzorci so zajeti pri vidni svetlobi, zato so na slikah razvidne tudi različne barve oči. V času pisanja te naloge je bil del zbirke za šarenico še v nastajanju, zato nismo imeli organiziranih identitet in mask. V ta namen smo segmentacijski model učili samostojno brez razpoznavnega dela.

Med-podatkovna zbirka

Med-podatkovno zbirko sestavljajo vsi vzorci zbirk SBVPI in CASIA. Namen zbirke je testirati splošnost modela na različne senzorje in tipe podatkov. V

našem primeru smo želeli testirati ali je model zmožen zaznati šarenice zajete pri vidni kot tudi pri bližnji infrardeči svetlobi. Zbirka predstavlja sklop obeh prej opisanih zbirk, brez informacije o identitetah iz zbirke CASIA. Enako kot pri SBVPI smo tukaj model učili samostojno.

6.1.3 Rezultati

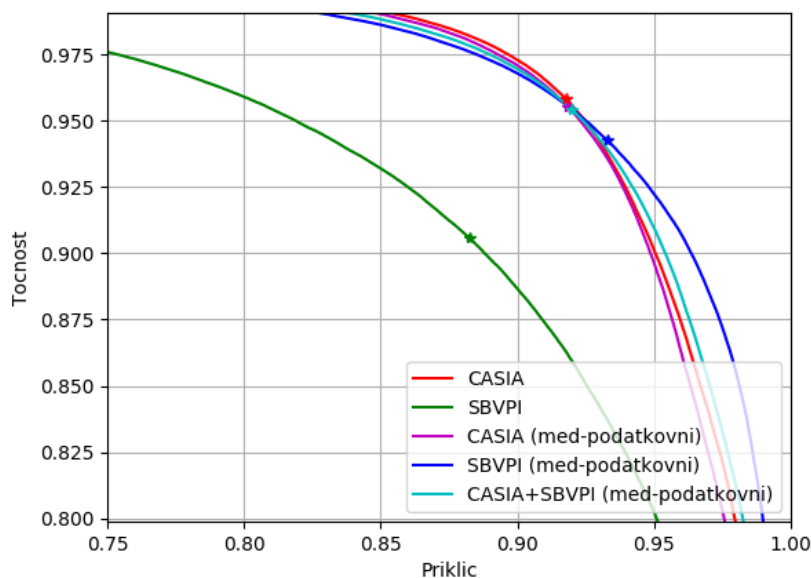
Rezultati segmentacije so vidni na slikah 6.1, 6.2 in 6.3. Iz slik vidimo, da so vsi modeli uspešno segmentirali šarenico. Največ težav pri tem je imel model naučen na vzorcih SBVPI. Predvidevamo, da je razlog za to večja znotraj razredna variacija, ki jo povzročijo raznolike poze in barve oči. Opazili smo tudi slabšo kakovost anotacij šarenic, katerih posledica so manj ostri robovi segmentacije. Težava ni prisotna na med-podatkovnem modelu, ki veliko bolj natančno in ostro označuje področje šarenice. Izboljšava je vidna tako na vzorcih zbirke SBVPI kot tudi CASIA. Predvidevamo, da je razlog za izboljšavo bolj konsistentna oblikovna informacija zbirke CASIA v konjunkciji z barvno informacijo zbirke SBVPI. Drugi razlog za izboljšavo je lahko večje število učnih vzorcev. Tako lahko takšen pristop predstavlja potencialno rešitev za pomanjkanje večjih podatkovnih zbirk. Primerjali smo tudi rezultate segmentacijskega modela, naučenega na zbirki CASIA, ki vzorcev SBVPI ni zmožen pravilno segmentirati. Iz slik lahko razberemo, da model ni zmožen segmentirati finih podrobnosti kot so trepalnice in manjši finih odsevi na površini očal. Model pa je zmožen zaznati večje svetleče odboje, vendar bi se teh zlahka znebili z upragovalno funkcijo.

Podobne rezultate lahko razberemo tudi iz kvantitativnih ocen. Na podlagi krivulje točnosti in priklica (sliki 6.1, B.1) smo določili najbolj uspešne pragove za posamezen model. Za najboljšega smo določili prag, ki ima najboljše razmerje med priklicem in točnostjo. Rezultati najboljših pragov so na krivulji označeni z *. S pridobljenimi pragovi smo izračunali prej opisane mere, katere so prikazane v tabeli 6.1. Na krivulji vidimo, da vsi modeli, razen tega naučenega na zbirki SBVPI, dosegajo približno enake rezultate. Numerični rezultati prikazujejo podobne rezultate. Najslabše se je odrezal mo-

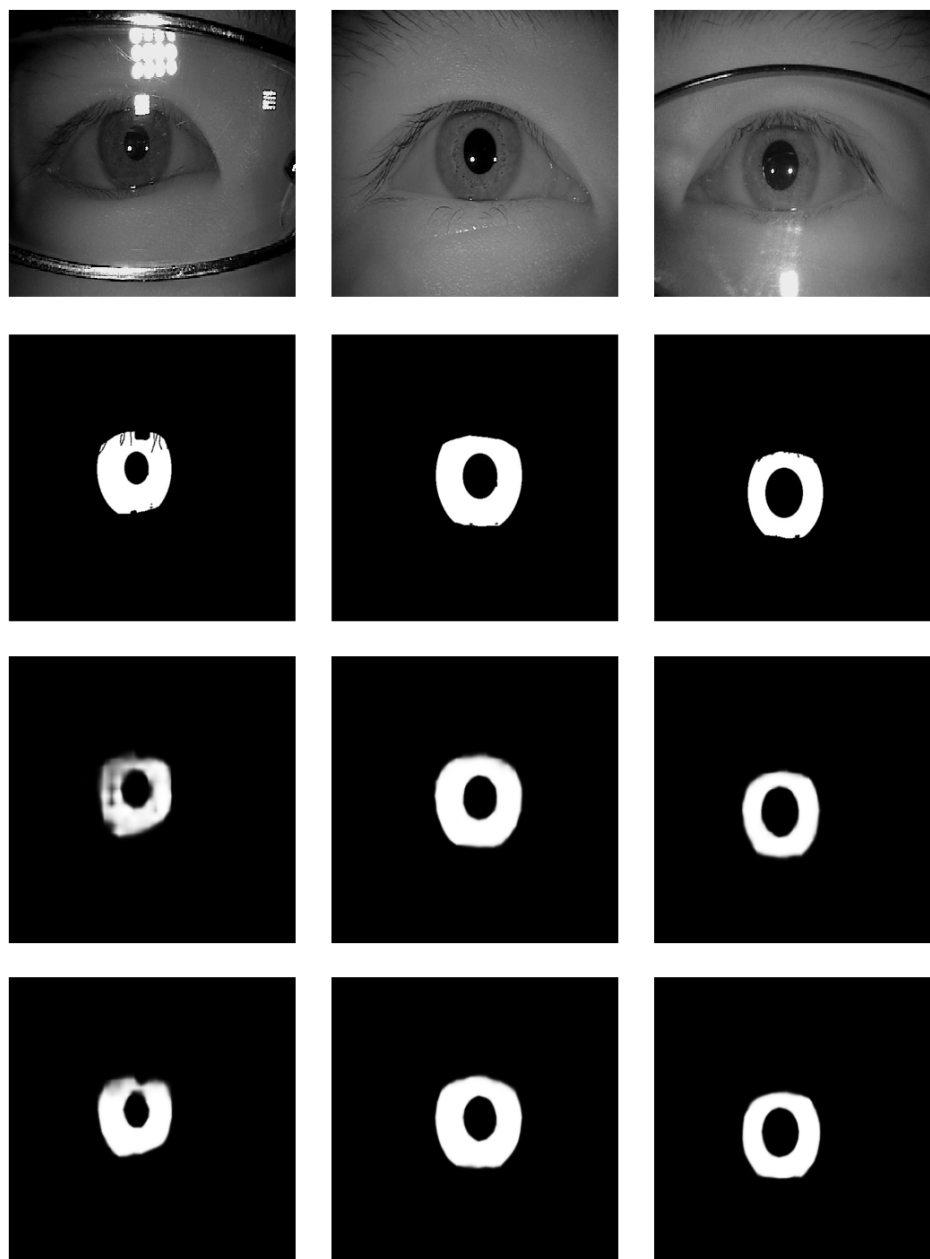
del naučen na zbirki SBVPI. Model je na splošno dosegel najslabše rezultate na vseh testih. Ta je tudi imel največje standardne deviacije, kar nakazuje tudi na nekonsistentnost rezultatov. Med-podatkovni model je pričakovano te izboljšal. Ostali modeli so med seboj pokazali približno enako uspešne rezultate.

Tabela 6.1: Rezultati segmentacije z najbolj uspešnimi pragovi potrjevalnih množic zbirk. V tabeli so prikazane povprečne vrednosti in standardni odklon vseh vzorcev potrjevalne množice za posamezno mero. Histogrami posameznih mer so prikazani na slikah B.2, B.3, B.4, B.5, B.6.

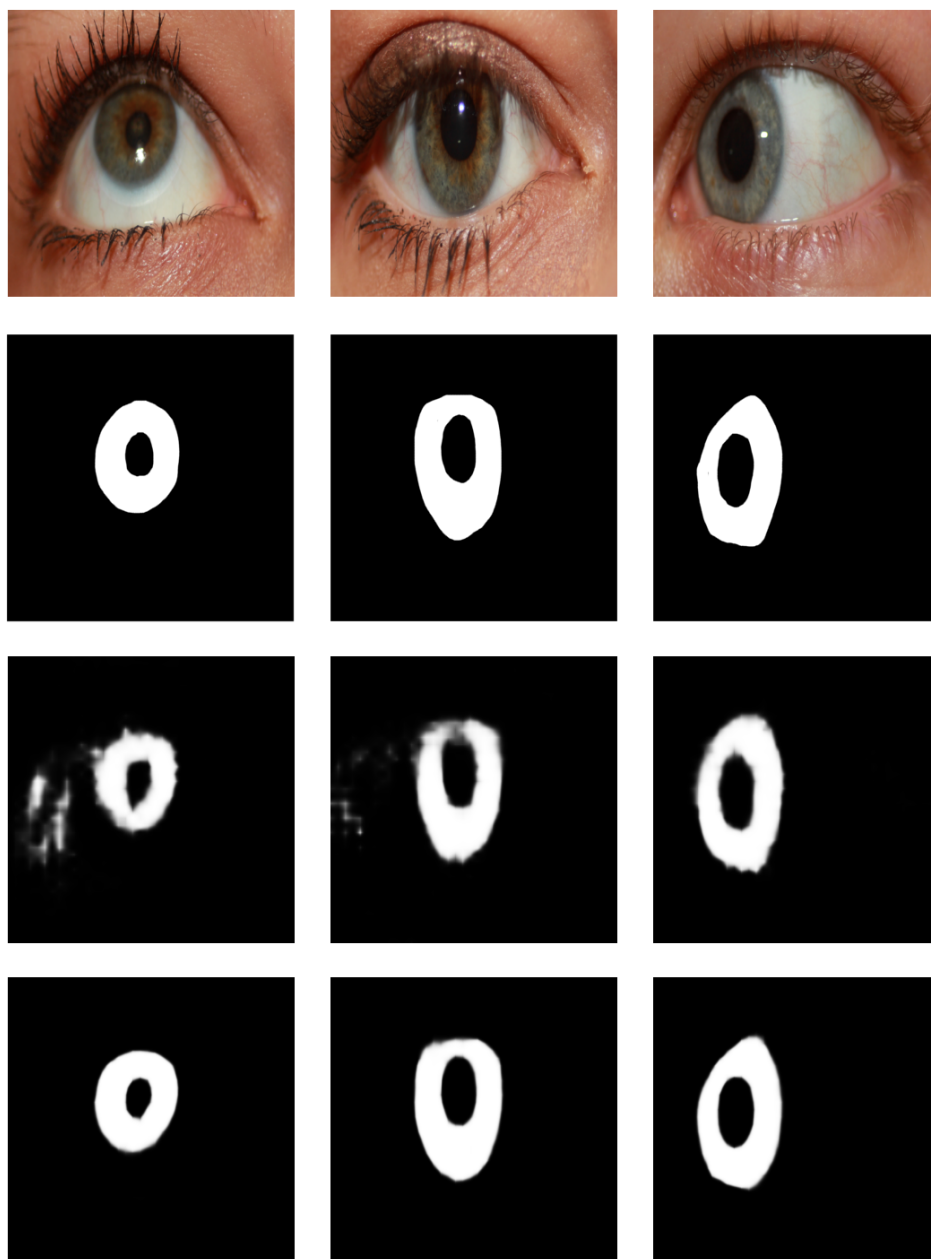
Zbirka	Naj. prag	Natančnost	Točnost	Priklic	IOU
CASIA	0,417	99,23 % \pm 0,25	91,79 % \pm 3,05	95,82 % \pm 2,32	88,26 % \pm 3,63
SBVPI	0,347	98,44 % \pm 0,54	88,26 % \pm 6,44	90,57 % \pm 7,00	80,89 % \pm 8,45
CASIA (med-podatkovni)	0,343	99,23 % \pm 0,25	91,83 % \pm 3,04	95,54 % \pm 2,41	88,07 % \pm 3,66
SBVPI (med-podatkovni)	0,442	99,11 % \pm 0,32	93,27 % \pm 4,68	94,27 % \pm 4,12	88,36 % \pm 6,16
CASIA + SBVPI	0,343	99,16 % \pm 0,29	91,97 % \pm 4,14	95,44 % \pm 3,08	88,14 % \pm 5,02



Slika 6.1: Približana krivulja točnosti in priklica. Posamezne krivulje prikazujejo uspešnosti posameznih zbirk. Simbol * prikazuje točko najboljšega praga. Nepribližana različica krivulje je vidna na sliki B.1.



Slika 6.2: Primeri segmentacij slik zbirke CASIA. V prvi vrsti se nahajajo izvirne slike, pod njimi so ročne oznake šarenic slik, v tretji vrsti se nahajajo segmentacije z modelom naučenim samo na vzorcih zbirke, spodnjo vrsto pa sestavljajo segmentacije med-podatkovnega modela.



Slika 6.3: Primeri segmentacij slik zbirke SBVPI. V prvi vrsti se nahajajo izvirne slike, pod njimi so ročne oznake šarenic slik, v tretji vrsti se nahajajo segmentacije z modelom naučenim samo na vzorcih zbirke, spodnjo vrsto pa sestavljajo segmentacije med-podatkovnega modela.

6.2 Razpoznav

6.2.1 Metrike

Pri ocenjevanju uspešnosti razpoznav smo uporabili 3 metrike: natančnost n -tega ranga (angl. rank- n accuracy), krivulja CMC (angl. cumulative match-score curve) ter krivulja ROC (angl. receiver operating characteristic curve).

Natančnost n -tega ranga in krivulja CMC

Pri problemu razpoznav predstavlja vsaka oseba svoj razred. Ker nam model vrača vektor verjetnosti za posamezen razred, lahko predvidevamo, da vzorec pripada najbolj verjetnemu razredu. Če ta ustreza pravi osebi, štejemo tak primer kot pravilno napoved. Ta postopek lahko ponovimo na vseh vzorcih podatkovne zbirke, seštejemo število pravih napovedi TP in jih delimo s številom vseh vzorcev N (enačba (6.5)). Dobimo delež pravilno napovedanih vzorcev, katerega imenujemo natančnost prvega ranga (angl. Rank-1 accuracy). V primeru, da štejemo kot pozitivno napoved vse primere, pri katerih katerikoli od n najbolj verjetnih razredov ustreza pravemu vzorcu, dobimo natančnost n -tega ranga. Z izračunom vseh natančnosti med rangom 1 in številom vseh razredov lahko izrišemo krivuljo. Dobljena krivulja se imenuje krivulja CMC (angl. cumulative match-score curve) in pove, kako se natančnost spreminja z zaporednimi rangi. Krivulja CMC se uporablja pri oceni identifikacije.

$$\text{Rang-}n = \frac{TP_n}{N} \quad (6.5)$$

Krivulja ROC

Krivuljo ROC dobimo na podoben način kot krivuljo točnosti in priklica. Razlika med njima je, da pri krivulji ROC predstavlja ordinatna os frekvenco pravih pozitivov ter abscisna os frekvenco napačnih pozitivov. Krivuljo dobimo tako, da iterativno računamo frekvenci na različnih pragovih vektorja

razredne verjetnosti. Frekvenca pravih pozitivov (angl. true positive rate) nam pove delež pravilno napovedanih pozitivnih vzorcev (enačba (6.6)), frekvenca napačnih pozitivov (angl. false positive rate) pa nam pove delež napačno napovedanih pozitivnih vzorcev (enačba (6.7)). S krivuljo si lahko pomagamo pri določitvi praga, ki minimizira vpliv napačnih pozitivov ali pa maksimiramo verjetnost pozitivnih napovedi. Torej v primeru, da bi želeli čim več oseb uspešno razpoznati, bi prag nastavili tako, da bi imeli čim večjo frekvenco pravih pozitivov in s tem tudi dovolili nekaj napačnih pozitivov. V nasprotnem primeru bi prag določili tako, da bi čim bolj precizno identificiral identitete oseb. V takem primeru bi želeli čim bolj omejiti frekvenco napačnih pozitivov. Pri oceni verifikacije uporabljamo krivuljo ROC.

$$\text{TPR} = \frac{TP}{TP + FN} \quad (6.6)$$

$$\text{FPR} = \frac{FP}{FN + TP} \quad (6.7)$$

6.2.2 Eksperimentalni protokol

Pri oceni razpoznavne smo uporabili 2 podatkovni zbirki: Casia-Iris-Thousand (v nadaljevanju imenovana CASIA) ter SBVPI. Tako kot pri segmentaciji, smo obe tudi kombinirali v med-podatkovno zbirko. Učenje globokih razpoznavnih modelov v osnovi poteka v načinu zaprte skupine (angl. closed set). To pomeni, da se mreža uči in identificira le osebe, katerih vzorci se nahajajo v učni množici. To pa pomeni, da bi z vsako dodatno osebo morali mrežo ponovno naučiti z novim naborom vzorcev, kar ni idealno. V ta namen je mreže dobro uporabiti tudi kot izločevalke značilk. Takšen način imenujemo razpoznavna odprte skupine (angl. open-set). Izločene značilke lahko z mero podobnosti primerjamo in določimo, katere so si najbližje. Pri tem je dobro, da testne zbirke vsebujejo tudi osebe, ki niso bile uporabljene pri učenju in potrditvi mreže. V našem primeru smo značilke jemali iz polno povezanega sloja na koncu razpoznavne in iz povprečno združevalnega sloja pred njim. Dobljena zaporedja smo primerjali z uporabo kosinusne razdalje (enačba (6.8)).

Pri tem je potrebno omeniti, da je težava odprte skupine bistveno težja.

$$\text{Razdalja (A, B)} = \frac{\sum_{i=1}^n A_i B_i}{\sqrt{\sum_{i=1}^n A_i^2} \sqrt{\sum_{i=1}^n B_i^2}} \quad (6.8)$$

Tako kot pri segmentaciji smo tudi pri razpoznavi uporabljali obogatitve z enakim naborom transformacij ter slike poenotili na velikost 320×320 . Brez njih se model pri večjih zbirkah, kot je CASIA, ni zmožen doseči višjih natančnosti. Testirali smo tudi uspešnost modela brez predhodne segmentacije, ki se tudi na večjih podatkovnih zbirkah ni bil zmožen naučiti do visoke mere uspešnosti.

Pri učenju razpoznavne smo najprej na segmentacijski del naložili uteži naučene v segmentacijskem koraku ter jim preprečili nadaljnje učenje. Tako smo zagotovili, da se bo razpoznavna učila le na področju šarenice. V primeru zbirke CASIA smo naložili tudi razpoznavni del in temu za večje število razredov primerno spremenili izhodni sloj.

Učenje je potekalo na enak način kot pri segmentaciji, le da smo v tem koraku uporabljali tudi tople ponovne zagone (angl. warm reset). Te izvedemo tako, da ponovno zaženemo učenje z do sedaj naučenimi parametri, ampak z nižjo hitrostjo učenja (v našem primeru $\eta = 10^{-6}$). Tako lahko najdemo še bolj fine nastavitve parametrov znotraj trenutne funkcijske okolice. Te ponovne zagone smo izvedli ročno.

CASIA

Zbirka CASIA za razpoznavo vsebuje vseh 20.000 slik izvirne zbirke opisane v poglavju 5.3.1. Razredi predstavljajo posamezne osebe, torej združili smo slike obeh oči v en razred. Zbirka vsebuje 14.000 učnih vzorcev, 4.000 potrjevalnih ter 2.000 testnih vzorcev. Testno množico sestavljajo osebe, ki se tudi nahajajo v učni množici (torej ne vzorci, ampak identitete), zato smo zbirko uporabili le v načinu zaprte skupine. Pri pregledu izvirne zbirke smo opazili, da so zaporedne slike pogosto zelo podobne. Od 10 slik za vsako oko posameznika sta potrjevalno množico sestavljali peta in sedma slika, testno pa druga.

SBVPI

Zbirka SBVPI vsebuje 3.708 slik. Od tega je 1.422 učnih in 656 validacijskih, ki pripadajo 120 razredom. Razredi so vsebovali delitev na levo in desno oko. Umetno smo podvojili število oseb tako, da smo zrcalili slike posamezne osebe. Testna podmnožica zbirke je vsebovala 1.630 slik oseb, ki se ne nahajajo v učni in validacijski zbirki. Množice vsebujejo vse različne poglede oči, razen periokularnega. Bolj podrobno razlago delitve zbirke si lahko preberete v magistrski nalogi [44].

Med-podatkovna

Med-podatkovna zbirka je tako kot prej sklop zbirk SBVPI in CASIA. Pri tem razrede predstavljajo identitete posameznih oseb, torej združili smo slike levega in desnega očesa zbirke SBVPI v en razred. Prilagodili smo tudi število vzorcev CASIA tako, da sta obe zbirki približno enako prisotni. Zbirka tako vsebuje 3.986 vzorcev pripadajočim 166 osebam (106 CASIA, 60 SBVPI), od tega 2.906 učnih, 1.080 potrjevalnih. Testno množico je sestavljal nabor preostalih vzorcev CASIA in SBVPI, katerih identitete niso bile del učne množice. Testna množica vsebuje 3.268 slik. Od tega 1.630 pripadajočih zbirki SBVPI in 1.638 zbirki CASIA. Skupaj je testna množica vsebovala 141 identitet.

6.2.3 Rezultati

Ocena razpoznavne je potekala v dveh ločenih skupinah testov. Pri prvi smo osebe razpoznali na podlagi verjetnosti za posamezen razred, ki smo jih prejeli iz zadnjega sloja mreže, torej razpoznavo zaprte skupine. Pri drugi skupini testov, smo osebo razpoznali na podlagi značilke zgrajene iz prejšnjih slojev modela. Z mero podobnosti značilke primerjamo in ocenimo ali najbolj podobne pripadajo isti osebi. V tem primeru gre za razpoznavo odprte skupine.

Zaprta skupina

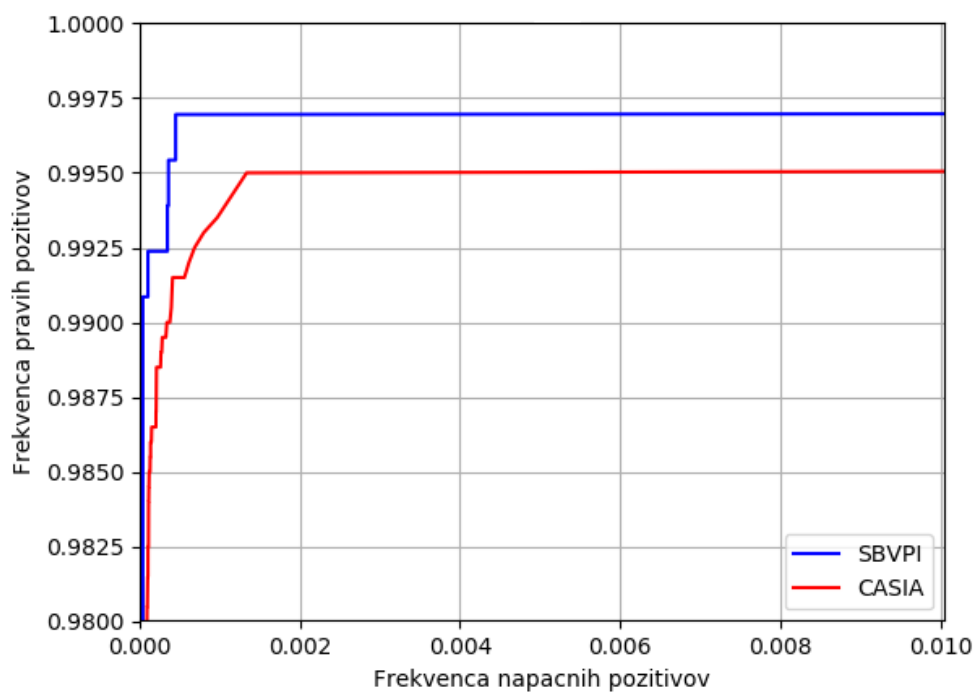
Pri razpoznavi zaprte skupine je bil model na obeh zbirkah zelo uspešen. Natančnost prvega ranga obeh modelov je presegla 95 % (tabela 6.2). Pri tem omenimo, da smo pri testih SBVPI uporabili vzorce potrjevalne skupine, saj je testna vsebovala le vzorce oseb, ki niso bile uporabljene pri učenju. Krivulja ROC (slika 6.4) tudi prikazuje izjemno dobre rezultate, saj je na videz skoraj kvadratna. Pri obeh modelih lahko pri 0,001 % napačnih razpoznav pričakujemo več kot 99 % pravih. Krivulja CMC (slika 6.5) kaže podobne rezultate. Vidimo lahko tudi, da se uspešnost modela CASIA zelo izboljša med rangom 1 in 2.

Tabela 6.2: Rezultati razpoznave zaprte skupine.

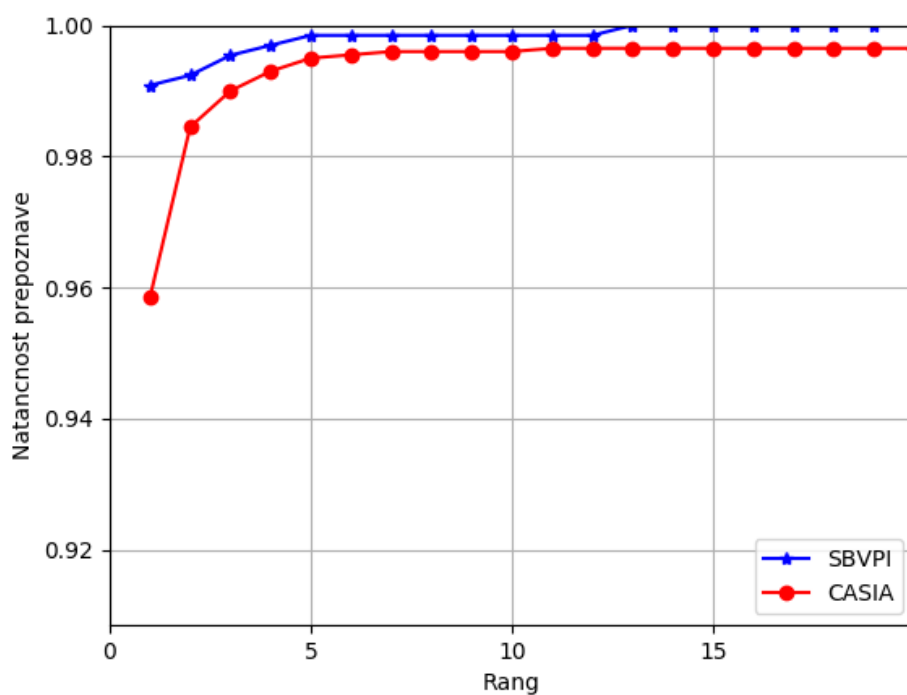
Zbirka	Rang-1	Rang-5
CASIA	95,85 %	99,50 %
SBVPI	99,09 %	99,85 %

Odprta skupina

Za vsako zbirko nam je uspelo naučiti model, katerega natančnost presega 90 % (tabela 6.3). Natančnosti različnih rangov lahko vidimo na krivulji CMC (slika 6.6). Na zbirki SBVPI je najbolje deloval model naučen na njenih vzorcih. Ta je dosegel natančnost prvega ranga 92,33 %. Pri zbirki CASIA je bil najbolj uspešen med-podatkovni model, ki je dosegel natančnost 95,12 %. Med-podatkovni model je na zbirki SBVPI dosegel bistveno slabšo natančnost 81,90 %. Predvidevamo, da je razlog za to večja znotraj-razredna variacija zbirke SBVPI. Poleg različnih barv šarenic se vzorci zbirke razlikujejo tudi v različnih pogledih in velikostih slik. Tako imajo lahko zaporedni vzorci pri poenotenju na velikost 320×320 slikovnih točk različne deformacije, medtem ko so vzorci zbirke CASIA relativno konsistentni. Predvidevamo, da se z večjim številom učnih ali pa z boljšo normalizacijo vzorcev natančnost



Slika 6.4: Močno približan nabor krivulj ROC. Posamezna krivulja pripada eksperimentu na različni zbirki. Uporabljeni modeli sta naučena na učnih množicah zbirk, katere ocenjujeta. Nepribližano različico si lahko ogledate v prilogi B.7.



Slika 6.5: Nabor CMC krivulj razpoznavne zaprte skupine. Vsaka krivulja prikazuje natančnost testov na različnih zbirkah, z različnimi modeli. V legendi je navedeno, kateri zbirki in kateremu modelu pripada krivulja. V primeru, da v oklepaju ni naveden model, uporablja model, naučen na navedeni zbirki.

še izboljšala. Sklop obeh zbirk je vendarle dosegel zadovoljivo natančnost 88,53 %.

Modele smo primerjali tudi z implementacijo konvencionalnega cevovoda za razpoznavo šarenice. Pri tem smo filtrirali vse rezultate, ki so imeli neveljavne detekcije. Uporabili smo javno dostopno implementacijo avtorja Liborja Maska [16]. Implementacija se od izvirne Daugmanove razlikuje v načinu detekcije robov šarenice. Namesto IDO ta uporablja Haughovo transformacijo. Vsi testirani modeli so dosegli višje rezultate kot Maskova metoda. Opazili smo tudi, da se je natančnost pri zaporednih rangih Maskove metode večala približno linearno, medtem ko je pri naši metodi naraščala približno logaritemsko. Še ena prednost, ki jih imajo globoke metode nad konvencionalnimi je hitrost izgradnje značilnk.

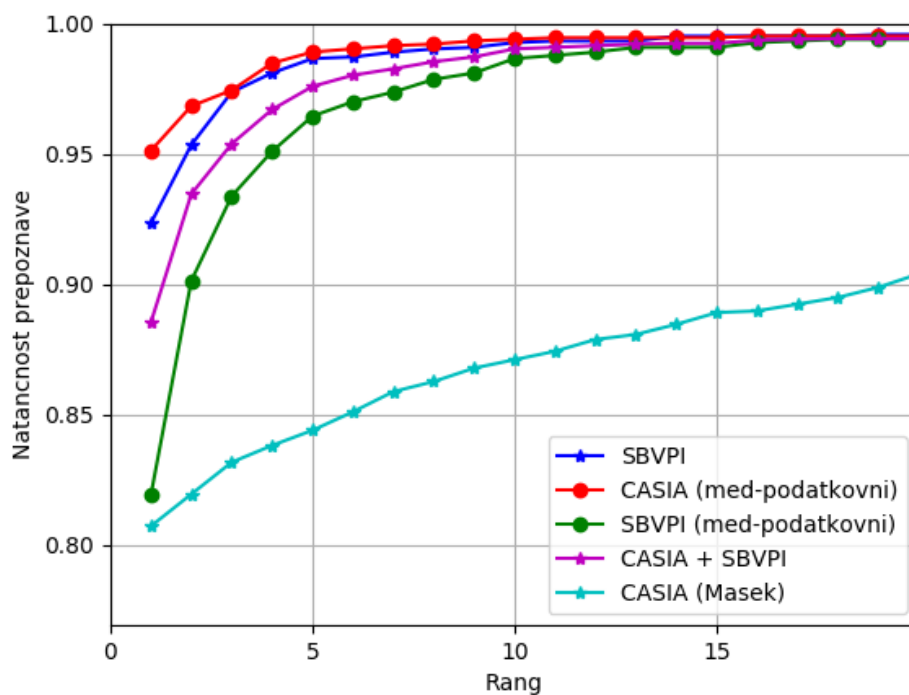
Testirali smo tudi razliko med uspešnostjo značilnk iz polno povezanega sloja in globalnega združevalnega sloja. V vseh primerih so bile značilke združevalnega sloja bolj uspešne. Predvidevamo, da je razlog za to bolj oblikovna/teksturna informacija združevalnega sloja.

Tabela 6.3: Rezultati razpoznave odprte skupine. Testna podmnožica zbirke CASIA vsebuje samo razrede, ki se pojavijo v učni množici, zato na modelu, naučenem izključno na njej, nismo naredil testov odprte skupine.

Zbirka	Rang-1	Rang-5
SBVPI	92,33 %	98,65 %
CASIA (med-podatkovni)	95,12 %	98,90 %
SBVPI (med-podatkovni)	81,90 %	96,44 %
CASIA + SBVPI	88,53 %	97,58 %
CASIA (Masek)	80,71 %	84,39 %

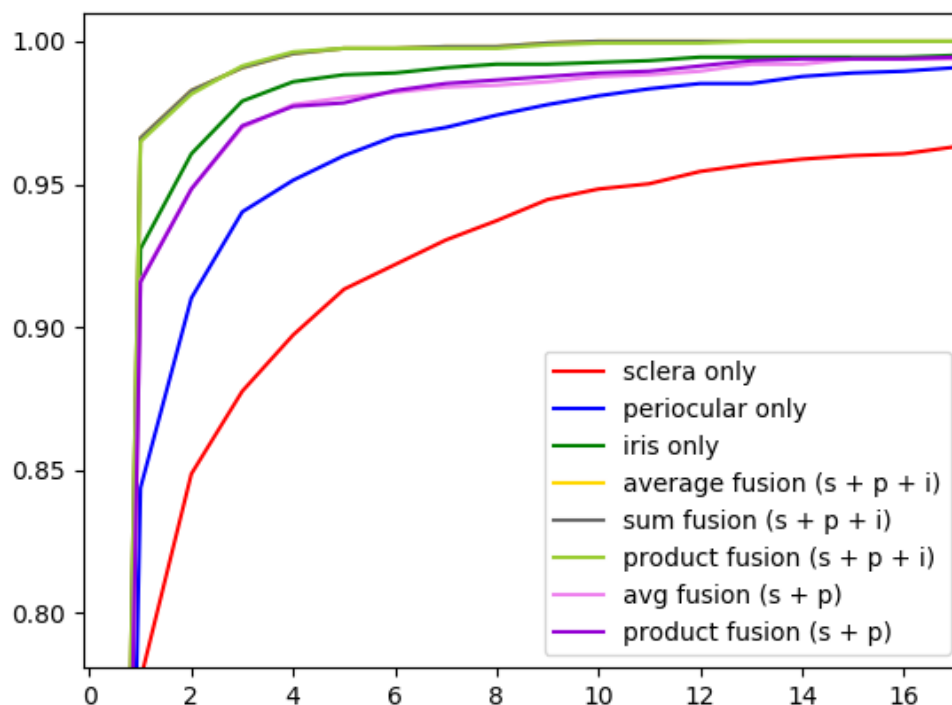
6.2.4 Fuzija

V sklopu te naloge smo sodelovali tudi s Petrom Rotom, ki je v svojem magistrskem delu [44] predstavljal razpoznavo na podlagi preostalih modalnosti



Slika 6.6: Nabor CMC krivulj razpoznavne odprte skupine. Vsaka krivulja prikazuje natančnost testov na različnih zbirkah, z različnimi modeli. V legendi je navedeno, kateri zbirki in kateremu modelu pripada krivulja. V primeru, da tega v oklepaju ni navedenega, je naučen na navedeni zbirki.

oči (žile beloočnice, periokularni del). Del njegove naloge je bila tudi fuzija različnih modalnosti, pri kateri je za razpoznavo šarenice uporabljal naš model. Krivulja CMC fuzije je prikazana na sliki 6.7. Iz rezultatov vidimo, da šarenica najbolj pripomore k razpoznavi. To je pričakovano, saj vsebuje najbolj opisno teksturo. Uspešnost te tudi preseže fuzijo ostalih dveh modalnosti. Kombinacija vseh treh doseže najvišje rezultate. Za bolj podroben opis postopka fuzije si preberite magistrsko delo [44].



Slika 6.7: Krivulje CMC fuzije [44].

Poglavje 7

Sklepne ugotovitve

V delu smo predstavili nov globoki cevovod za razpoznavo šarenice. Arhitektura predstavlja novost na področju globoke razpoznavne šarenice, kjer raziskovalci do sedaj še niso predstavili implementacije celotnega globokega cevovoda za razpoznavo šarenice znotraj ene zaključene mreže.

Avtorji člankov so kot glavno težavo za pomanjkanje rešitev navajali pomanjkanje obsežnih podatkovnih zbirk. Kot potencialno rešitev te težave predlagamo združevanje različnih podatkovnih zbirk. Pri tem predlagamo poenotenje slik različnih zbirk, tako da so si strukture (velikosti, poze šarenice ipd.) vseh zbirk tudi čim bolj podobne.

Naša splošna arhitektura globokega cevovoda za razpoznavo šarenice je sestavljena iz dveh visoko-nivojskih delov: segmentacije in razpoznavne. Vloga segmentacije v cevovodu je, da deluje kot model pozornosti razpoznavnega dela. Z množenjem izhoda segmentacije in izvorne slike označimo predele slike, ki pripadajo šarenici. Dobljene slike potem uporabimo v razpoznavi, ki iz teh izloči značilke na podlagi katerih osebe razpoznamo.

Avtorji članka [9] so predlagali rabo modelov z manjšim številom parametrov, zato smo se v naši implementaciji odločili za rabo arhitektur Xception, DeepLabV3+ (s hrbtenico MobileNet), ki imata zaradi inovativne bločne strukture izjemno dobro razmerje med številom učnih parametrov in uspešnostjo modelov. Skupaj je naš končni cevovod uporabljal okvirno 29

milijonov učnih parametrov.

Zaradi pomanjkanja zbirk za segmentacijo smo tudi označili 220 slik pripadajočih 11 osebam zbirke Casia-Iris-Thousand. Za označevanje smo tudi razvili spletno aplikacijo, ki omogoča sočasno označevanje iste centralizirane zbirke.

Za učenje modela smo uporabljali dvostopenjski pristop učenja, pri katerem prvo učimo segmentacijo in nato razpoznave. Pri učenju segmentacije smo uporabljali pristop, ki uči segmentacijo ter razpoznavo hkrati in tako prilagodi rezultat segmentacije, da čim bolj pripomore k razpoznavi.

Implementacijo našega cevovoda smo testirali na dveh podatkovnih zbirkah Casia-Iris-Thousand in SBVPI. Na obeh zbirkah nam je uspelo naučiti modele, ki so dosegali visoke rezultate tako pri segmentaciji kot tudi pri razpoznavi. Uspelo nam je tudi naučiti model, ki je zmožen uspešno razpoznati vzorce, zajete pri bližnji infrardeči in vidni svetlobi, kar nam klasični pristopi načeloma ne omogočajo.

Iz predstavljenega lahko tako zaključimo, da je globoka razpoznavna šarenice možna in omogoča posebno fleksibilnost, katero je težko doseči s tradicionalnimi metodami.

Nadaljnje delo bi posvetili razpoznavi in segmentaciji že vnaprej normaliziranih šarenic, saj predvidevamo, da bi tako lahko še povečali natančnost modela in hkrati tudi bistveno zmanjšali število učnih parametrov. Predlagali bi tudi raziskavo implementacij cevovoda z ostalimi že znanimi arhitekturami. S spreminjanjem izhodnega koraka bi bil model DeepLab mogoče zmožen tudi bolje segmentirati fine podrobnosti kot so trepalnice, zato bi tudi predlagali raziskavo tega. Na podlagi ugotovitev članka [22] bi predlagali testiranje uspešnosti značilk iz vmesnih slojev razpoznavnega dela, saj bi znali izboljšati uspešnost modela. Predlagali bi tudi testiranje binariziranih značilk konvolucijskih slojev.

Priloga A

Spletno orodje za označevanje slik

Za lažje in hitrejše označevanje binarnih mask šarenice smo med izdelavo naloge implementirali spletno orodje za označevanje slik. Aplikacija je razdeljena na zaledni (angl. back-end, BE) in čelni (angl. front-end, FE) del. FE omogoča risanje binarnih mask nad slikami, katere samodejno prejme iz BE. V nadaljevanju bomo podrobneje predstavili funkcionalnosti posameznega dela.

A.1 Čelni del

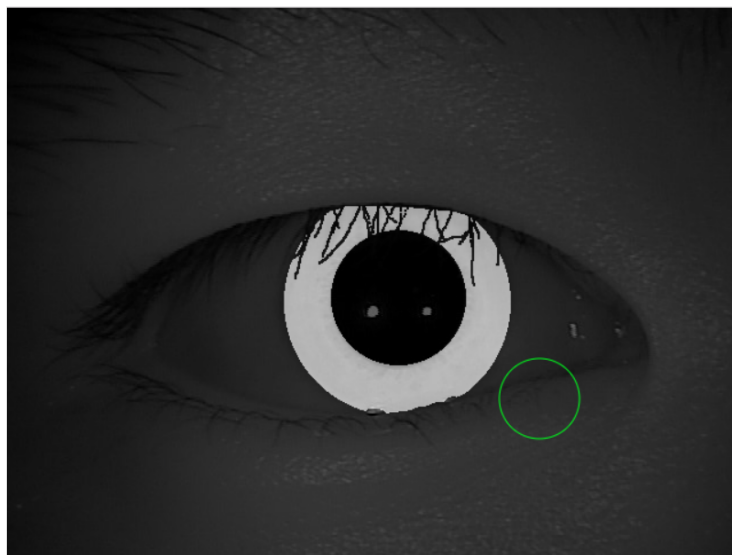
FE je razdeljen na dva modula: označevalni del in del namenjen pregledu mask. Označevalni del (viden na sliki A.1) samodejno iz BE prejme še neoznačene slike, katere lahko z risalnim orodjem nato označimo. Risalno orodje ima okrogel čopič, katerega lahko z miškinim kolesčkom večamo in manjšamo. S številčkama 1 in 2 lahko uporabnik izbira razred, katerega želi risati. Označevalni del ima tudi funkcionalnost razveljavitve (angl. undo) in ponovne izvedbe (angl. redo). FE izrisuje masko rahlo prozorno čez sliko, ki jo uporabnik označuje. S tipko *h* lahko masko tudi skrivamo, da spodaj ležečo sliko boljše vidimo. Masko lahko, kljub temu da je skrita, še naprej

spreminjamo. Ko smo z masko zadovoljni, jo lahko s tipko *Enter* pošljemo nazaj na BE, ki jo nato shrani.

Na pregledovalnem delu (viden na sliki A.2) preverimo ali je prišlo pri označevanju do napak. Vse maske morajo biti označene, preden jih lahko izvozimo. S pritiskom na gumba *correct* ali *wrong* se lahko pregledovalec odloči ali je maska pravilna ali nepravilna. V primeru nepravilne maske se maska izbriše in slika ponovno postane kandidat za označevanje.

A.2 Zaledni del

BE organizira izvajanje FE in omogoča branje podatkovnih zbirk ter izvoz končnih mask. BE indeksira podatkovne zbirke v podatkovno bazo ArangoDB. Pri tem si zapomni razred, pot do slike, velikost slike ter ostale značilnosti posameznih vzorcev. V bazi ima tudi zapis poti do maske in ali je bila slika že potrjena. BE omogoča tudi hkratno označevanje večih slik.



1: foreground, 2: background, h: hide mask, CTRL-Z: undo, CTRL-Y: redo, mouse wheel: control size, enter: submit

Slika A.1: Označevalni del aplikacije.



h-hide mask

Correct

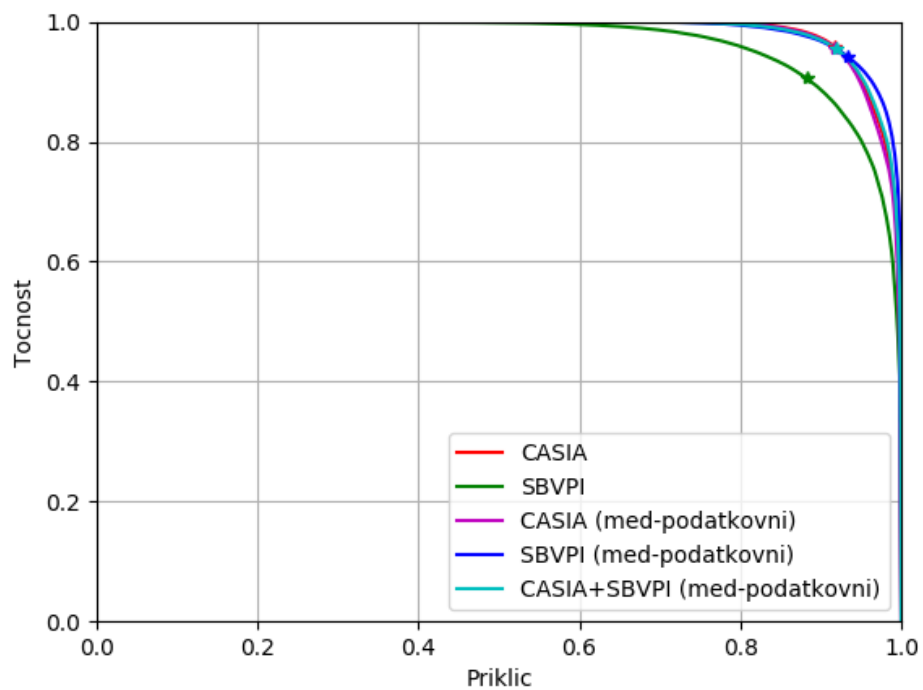
Wrong

Slika A.2: Čelni del aplikacije namenjen pregledovanju mask.

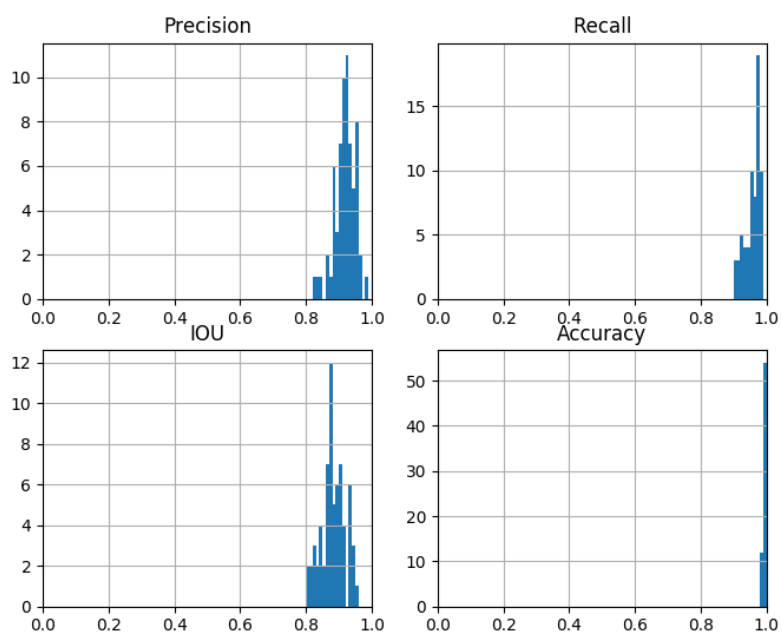
Priloga B

Dodatni grafi

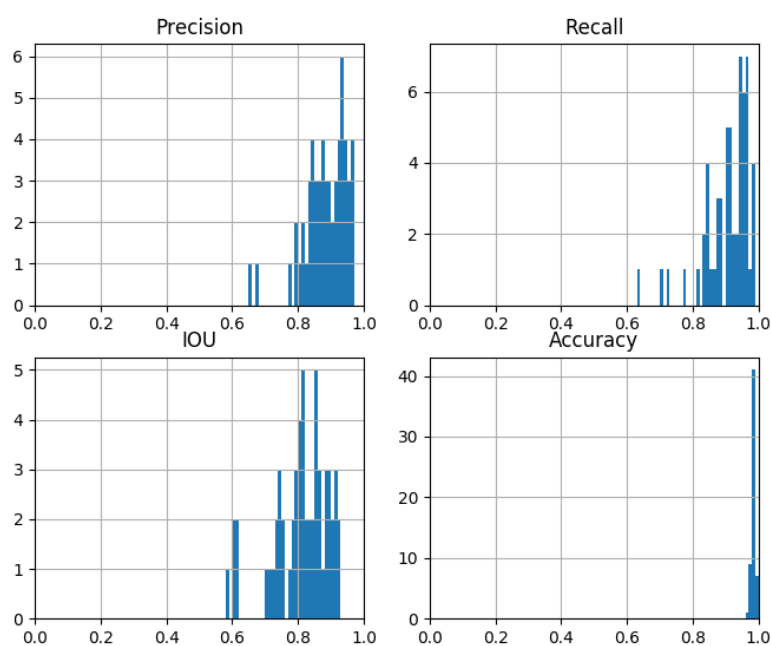
Slike B.1, B.2, B.3, B.4, B.5, B.6, B.7 prikazujejo dodatne rezultate iz poglavij 6.1.3, 6.2.3, ki osvetlijo še sekundarne vidike eksperimentov. Sklici na te slike so podani tudi v omenjenih poglavjih.



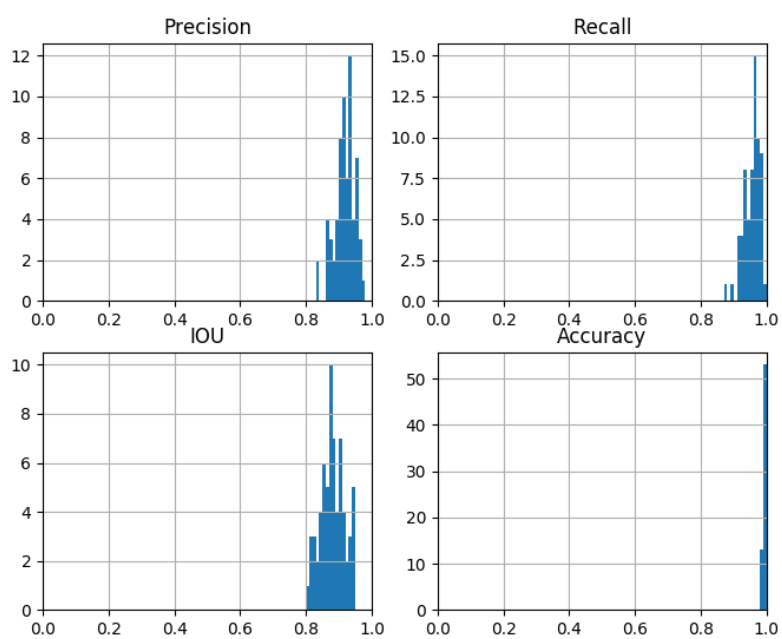
Slika B.1: Krivulja točnosti in priklica segmentacije.



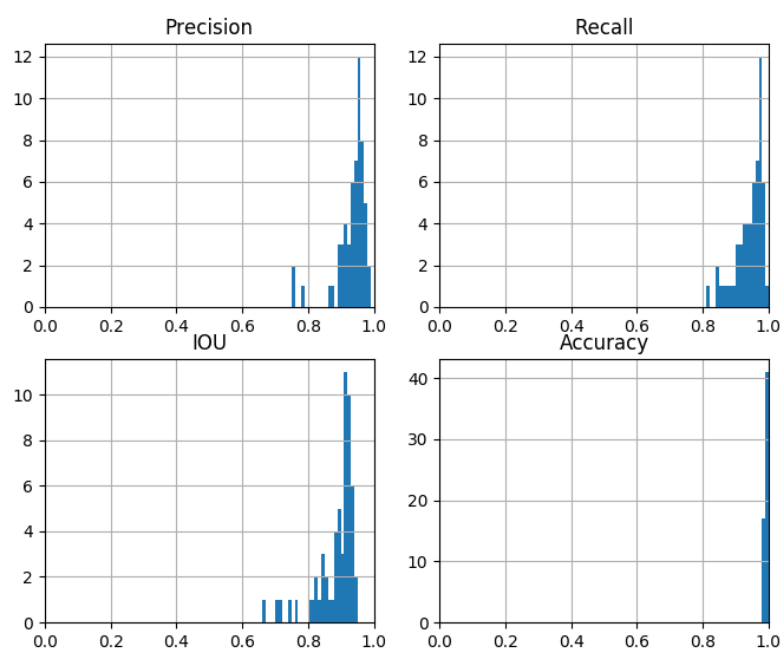
Slika B.2: Histogram testnih metrik segmentacije na zbirki CASIA z modelom naučenim na njenih vzorcih.



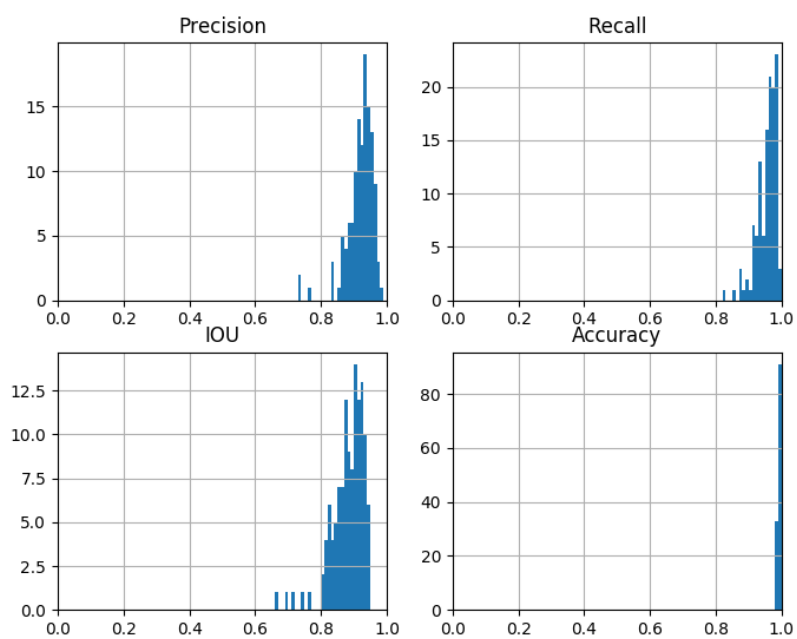
Slika B.3: Histogram testnih metrik segmentacije na zbirki SBVPI z modelom naučenim na njenih vzorcih.



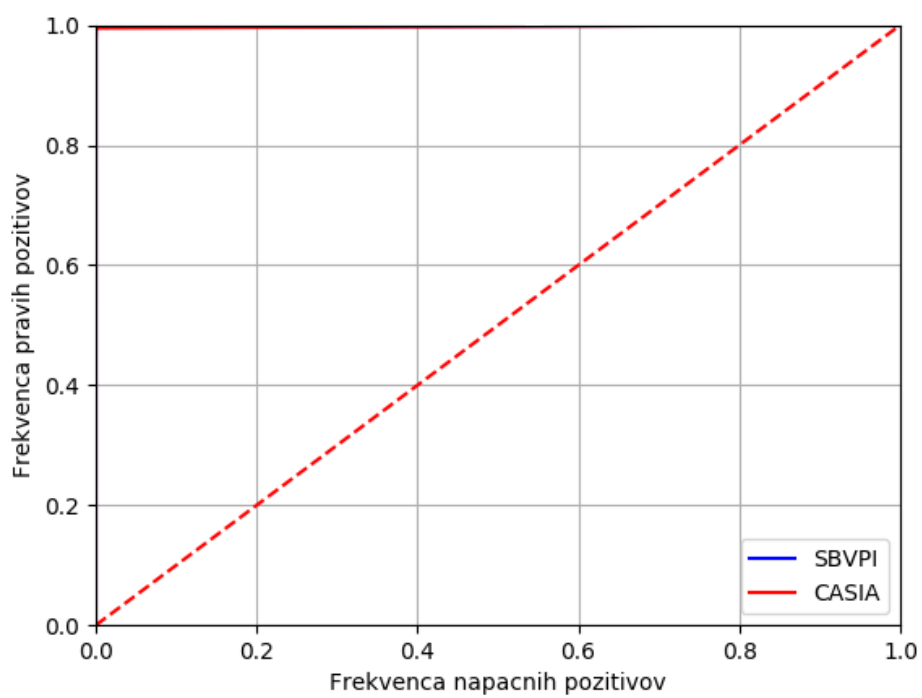
Slika B.4: Histogram testnih metrik segmentacije na zbirki CASIA z med-podatkovnim modelom.



Slika B.5: Histogram testnih metrik segmentacije na zbirki SBVPI z med-podatkovnim modelom.



Slika B.6: Histogram testnih metrik segmentacije na združenih zbirkah z med-podatkovnim modelom.



Slika B.7: Krivulja ROC razpoznavne na zaprti skupini.

Literatura

- [1] J. G. Daugman. High confidence visual recognition of persons by a test of statistical independence. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(11):1148–1161, Nov 1993.
- [2] J. Daugman. *Encyclopedia of Biometrics*, chapter Iris Recognition at Airports and Border-Crossings, pages 819–825. Springer US, Boston, MA, 2009.
- [3] Spletna stran AADHAAR. <https://uidai.gov.in>. Dostopano: 05.10.2018.
- [4] Navodila za prehod Nexus na meji med USA in Kanado. <http://usa.immigrationvisaforms.com/travel/nexus-lane-instructions-for-usa-canada-border>. Dostopano: 01.10.2018.
- [5] A. Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.
- [6] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.

-
- [7] A. Kumar Z. Zhao. Towards more accurate iris recognition using deeply learned spatially corresponding features. In *International Conference on Computer Vision, ICCV 2017*, pages 1–10, 2017.
- [8] A. Gangwar and A. Joshi. Deepirisnet: Deep iris representation with applications in iris recognition and cross-sensor iris recognition. In *2016 IEEE International Conference on Image Processing (ICIP)*, pages 2301–2305, Sept 2016.
- [9] T. Xingqiang, X. Jiangtao, and L. Peihu. Deep convolutional features for iris recognition. In *Biometric Recognition: 12th Chinese Conference, CCBR 2017, Shenzhen, China, October 28-29, 2017, Proceedings*, pages 391–400, Cham, 2017. Springer International Publishing.
- [10] S. Bazrafkan, S. Thavalengal, and P. Corcoran. An end to end deep neural network for iris segmentation in unconstrained scenarios. *Neural Networks*, 106:79 – 95, 2018.
- [11] M. Arsalan, H. G. Hong, R. A. Naqvi, M. B. Lee, M. C. Kim, D. S. Kim, C. S. Kim, and K. R. Park. Deep learning-based iris segmentation for iris recognition in visible light environment. *Symmetry*, 9:1–25, 11 2017.
- [12] Podatkovna zbirka CASIA v4. <http://biometrics.idealtest.org/dbDetailForUser.do?id=4>. Dostopano: 05.10.2018.
- [13] Zbirka SBVPI. sclera.fri.uni-lj.si. Dostopano: 06.10.2018.
- [14] P. Peer, Ž. Emeršič, and B. Meden. *Skripta Slikovna biometrija*. Fakulteta za računalništvo in informatiko, Univerza v Ljubljani, 2017.
- [15] C. Rathgeb, A. Uhl, and P. Wild. *Iris Recognition: From Segmentation to Template Security*. Springer Science & Business Media, 2012.
- [16] L. Masek. Recognition of human iris patterns for biometric identification. Magistrsko delo, The University of Western Australia, 2003.

-
- [17] C. Liu and M. Xie. Iris recognition based on dlda. In *18th International Conference on Pattern Recognition (ICPR'06)*, volume 4, pages 489–492, 2006.
- [18] H. Ali, M. J. E. Salami, and Wahyudi. Iris recognition system by using support vector machines. In *2008 International Conference on Computer and Communication Engineering*, pages 516–521, May 2008.
- [19] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. E. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. *CoRR*, abs/1409.4842, 2014.
- [20] N. Othman, B. Dorizzi, and S. G. Salicetti. Osiris: An open source iris recognition software. *Pattern Recognition Letters*, 82:124–131, 2016.
- [21] P. Li, X. Liu, L. Xiao, and Q. Song. Robust and accurate iris segmentation in very noisy iris images. *Image and Vision Computing*, 28(2):246–253, 2010. Segmentation of Visible Wavelength Iris Images Captured At-a-distance and On-the-move.
- [22] K. Nguyen, C. Fookes, A. Ross, and S. Sridharan. Iris recognition with off-the-shelf cnn features: A deep learning perspective. *IEEE Access*, 6:18848–18855, 2018.
- [23] M. Warren S. and P. Walter. Neurocomputing: Foundations of research. chapter A Logical Calculus of the Ideas Immanent in Nervous Activity, pages 15–27. MIT Press, Cambridge, MA, USA, 1988.
- [24] F. Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, pages 65–386, 1958.
- [25] M. A. Nielsen. *Neural Networks and Deep Learning*. Determination Press, 2015.

- [26] K. O’Shea and R. Nash. An introduction to convolutional neural networks. *CoRR*, abs/1511.08458, 2015.
- [27] A. F. Agarap. Deep learning using rectified linear units (relu). *CoRR*, abs/1803.08375, 2018.
- [28] D. Kaibo, K. S. Sathiya, C. Wei, S. S. Krishnaj, and P. A. Neow. Multi-category classification by soft-max combination of binary classifiers. In *Multiple Classifier Systems*, pages 125–134, Berlin, Heidelberg, 2003. Springer Berlin Heidelberg.
- [29] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958, 2014.
- [30] D. P. Kingma and J. L. Ba. Adam: a method for stochastic optimization. In *International Conference on Learning Representations*.
- [31] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. *CoRR*, abs/1505.04597, 2015.
- [32] Segmentacija satelitskih slik z arhitekturo U-net. <https://vooban.com/en/tips-articles-geek-stuff/satellite-image-segmentation-workflow-with-u-net/>. Dostopano: 05.10.2018.
- [33] Praktična segmentacija slik z arhitekturo U-net. <https://tuatini.me/practical-image-segmentation-with-unet/>. Dostopano: 05.10.2018.
- [34] J. Lozej, B. Meden, V. Štruc, and P. Peer. End-to-end iris segmentation model based on the u-net architecture. In *International Work Conference on Bioinspired Intelligence (IWOBI)*, 2018.

-
- [35] Segmentacija z U-net in Keras. <https://www.depends-on-the-definition.com/unet-keras-segmenting-images/>. Dostopano: 05.10.2018.
- [36] F. Chollet. Xception: Deep learning with depthwise separable convolutions. *CoRR*, abs/1610.02357, 2016.
- [37] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. E. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. *CoRR*, abs/1409.4842, 2014.
- [38] M. Sandler, A. G. Howard, M. Zhu, A. Zhmoginov, and L. C. Chen. Inverted residuals and linear bottlenecks: Mobile networks for classification, detection and segmentation. *CoRR*, abs/1801.04381, 2018.
- [39] L. C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. *CoRR*, abs/1802.02611, 2018.
- [40] Knjižnica TensorFlow. <https://www.tensorflow.org/>. Dostopano: 03.10.2018.
- [41] Python knjižnica Keras. <https://keras.io/>. Dostopano: 03.10.2018.
- [42] Implementacija DeepLabV3+ avtorja bonlime. <https://github.com/bonlime/keras-deeplab-v3-plus>. Dostopano: 05.10.2018.
- [43] Spletna stran podjetja IrisKing. <http://www.irisking.com>. Dostopano: 06.10.2018.
- [44] P. Rot. Metode globokega učenja za biometrično razpoznavo na podlagi očesa. Magistrsko delo, Fakulteta za računalništvo in informatiko, Univerza v Ljubljani, 2018.