

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 6686

**DUBOKO UČENJE RELAYNET ARHITEKTUROM ZA
SEGMENTACIJU SLOJEVA RETINE**

Antun Herkov

Zagreb, lipanj 2020.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 6686

**DUBOKO UČENJE RELAYNET ARHITEKTUROM ZA
SEGMENTACIJU SLOJEVA RETINE**

Antun Herkov

Zagreb, lipanj 2020.

Zagreb, 13. ožujka 2020.

ZAVRŠNI ZADATAK br. 6686

Pristupnik: **Antun Herkov (0036507724)**

Studij: Računarstvo

Modul: Računarska znanost

Mentor: prof. dr. sc. Sven Lončarić

Zadatak: **Duboko učenje ReLayNet arhitekturom za segmentaciju slojeva retine**

Opis zadatka:

Dijagnostika i terapija bolesti očne pozadine zahtijeva snimanje korištenjem optičke koherentne tomografije (OCT). U okviru diplomskog rada potrebno je proučiti metode za segmentaciju anatomskih slojeva u slikama očne pozadine dobivenih OCT snimanjem. Posebnu pažnju potrebno je posvetiti metodama segmentacije slika temeljenim na dubokim neuronskim mrežama s ReLayNet arhitekturom. Potrebno je ostvariti programsku implementaciju odabrane duboke neuronske mreže za rješenje problema segmentacije slojeva retine te provesti učenje pomoću danih skupova OCT slika. Točnost segmentacije slojeva treba ispitati korištenjem ispitnih skupova OCT slika.

Rok za predaju rada: 12. lipnja 2020.

Sadržaj

Uvod.....	1
1. Opis problema.....	3
1.1. Pozadinske informacije.....	3
1.2. Postojeće metode.....	4
1.3. Problem segmentacije.....	5
2. Osnove neuronskih mreža.....	7
2.1. Gradijentni spust.....	8
2.2. ReLU.....	8
2.3. Softmax.....	9
2.4. Epohe.....	9
2.5. Veličina grupe.....	10
3. Metodologija.....	11
3.1. Arhitektura.....	12
3.1.1. Enkoderi.....	12
3.1.2. Dekoderi.....	13
3.1.3. Klasifikator.....	13
3.2. Treniranje modela.....	14
3.2.1. Funkcija pogreške.....	14
3.2.2. Metrika točnosti.....	15
4. Baza podataka.....	17
4.1. Prilagodba dataseta ReLayNet arhitekturi.....	18
4.2. Augmentacija podataka.....	18

5. Eksperimenti i rezultati.....	19
5.1. Model 1: unakrsna entropija s težinama.....	19
5.2. Model 2: Diceov gubitak.....	21
5.3. Model 3: udružena pogreška.....	23
Zaključak.....	25
Literatura.....	26
Sažetak.....	28
Summary.....	29
Skraćenice.....	30
Privitak.....	31

Uvod

Optička koherentna tomografija spektralne domene (engl. *spectral domain optical coherence tomography*, SD-OCT) je neinvazivni način snimanja koji se obično koristi za dobivanje snimki poprečnog presjeka visoke rezolucije (6 μm) za biološka tkiva s dovoljnom dubinom penetracije snimanja (0,5 - 2 mm) [1]. Koristi se princip stvaranja mrljica kroz koherentno opažanje fotona koji su raspršeni unutar optičkog medija poput biološkog mekog tkiva. Primjenjuje se u raznim medicinskim snimkama od patologije mrežnice do kožnih snimanja radi praćenja zacjeljivanja rana i intravaskularnog snimanja za učinkovito postavljanje stenta, detekcije lumena i otkrivanja plaka. OCT je preferirani način za slike presjeka mrežnice zbog visoke rezolucije koja prioritizira jasnu vizualizaciju različitih sastavnih slojeva mrežnice [1] [3].

Raspoznavanje uzoraka u snimkama retine je problem prilagođen za primjenu umjetne inteligencije za analizu slike. Koristi se u dijagnozama raznih bolesti kao što su dijabetička retinopatija, retinopatija nedonoščadi te senilna makularna degeneracija [2].

Makula je središnji dio retine koji omogućuje centralni vid, čitanje i razlikovanje detalja. Starosna odnosno senilna degeneracija makule (engl. *age-related macular degeneration*, AMD) je bolest koja se javlja kod starijih odraslih ljudi, a može rezultirati gubitkom vida u središtu vidnog polja [3]. Budući da je AMD bolest makule, vid neće sasvim nestati niti kad bolesnik u cijelosti izgubi centralni vid, već će se tada koristiti perifernim dijelovima retine [16]. AMD se može očitovati u obliku odvajanja pigmentnog epitela retine što motivira segmentaciju i prepoznavanje slojeva na OCT snimkama kao neinvazivan način dijagnoze degeneracije makule [3]. Kod dijagnoze AMD-a bitna je pravovremenost liječenja s obzirom na to da se centar makule može u kratkom vremenu znatno oštetiti i dovesti do trajnih posljedica za vid [16]. Potrebno je pravilno nadziranje morfologije retinalnog sloja kako bi liječnici mogli pomoći pacijentima sa zaustavljanjem progresije bolesti i oporavkom vidne oštine.

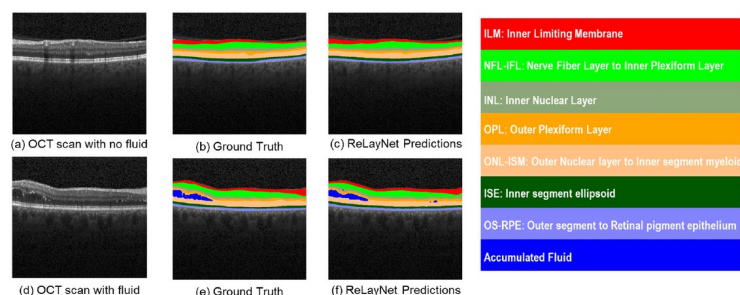
U sklopu ovog završnog rada, bilo je potrebno prilagoditi metodu predloženu u članku domaćoj bazi podataka i sukladno tome segmentacijski problem je tretiran kao višeklasni segmentacijski zadatak s četiri klase.

U prvom poglavlju dan je kraći pregled samog problema segmentacije i postojećih metoda koje se bave istom problematikom. U drugom poglavlju opisano je nekoliko glavnih koncepata neuronskih mreža koji su korišteni u implementaciji metode. U trećem poglavlju detaljnije je opisana sama arhitektura ReLayNeta, proces učenja modela kroz udruženu funkciju pogreške koju čine unakrsna entropija i Diceov gubitak te metrika točnosti kojom kvantificiramo uspješnost segmentacije slojeva. Četvrto poglavlje posvećeno je opisu baze podataka i prilagodbi iste za implementaciju ReLayNeta. Isto tako, opisana je augmentacija podataka korištena pri učenju modela. Peto poglavlje se bavi s nekoliko provedenih eksperimenata s osvrtom na razlike i rezultate pojedinih treniranih modela.

1. Opis problema

1.1. Pozadinske informacije

OCT snimanje mrežnice centrirano na optički živac i foveu je vrlo izazovno zbog prisutnosti mikro-skadičnih pokreta (titraja) oka što rezultira gibanjem artefakata, varijacijama u nagibu tkiva u odnosu na površinu koherencije vala i lošim odnosom signala i buke s povećanjem dubine snimanja [1]. Snimanje je posebno teško u slučajevima izrazito kratkovidnih očiju (jer su miopične odnosno kratkovidne oči izduljene i samim time je snimanje otežano različitim dubinama slojeva). Ovi svojstveni izazovi povezani s načinom snimanja čine tumačenje OCT slike izazovnim i često vrlo različitim između stručnjaka [1]. Točnije, izazovno je odrediti slojeve na OCT snimku zbog vrlo difuzne prirode granica između slojeva mrežnice. To čini ručno označavanje granica između slojeva vrlo subjektivnim i dugotrajnim. Upravo to je motivacija iza velikog broja istraživanja u svrhu razvoja automatiziranih metoda za segmentiranje različitih slojeva mrežnice na temelju OCT snimki s ciljem olakšanja pri postavljanju točne dijagnoze uz minimalnu varijaciju između stručnjaka [4]. U tu svrhu je u radu na kojemu se ovaj završni rad zasniva predložena implementacija koja se temelji na učenju modela za segmentaciju više slojeva mrežnice i prepoznavanje područja nakupljene tekućine u OCT snimkama oka, nazvana ReLayNet (engl. *Retinal Layer segmentation network*, mreža za segmentaciju slojeva retine).



Slika 1.1: Primjer OCT snimki s označenim slojevima iz ReLayNet članka [4]

1.2. Postojeće metode

Većina pristupa problemu segmentacije slojeva retine **nisu cjelovite metode**. U tim pristupima segmentacija se najčešće postiže kroz nekoliko koraka koji između ostalog uključuju i **fazu pretprocesiranja (smanjenje šuma, pročišćavanje slike i ostalo)**. To nikako ne umanjuje korisnost tih pristupa, ali činjenica je da samim time **zahtijevaju veće poznavanje problematike i ručno modeliranje različitih procjena**. OCT je često predstavljen kao graf (engl. *graph construction*, GC) i pridjeljivanje oznaka slojevima se rješava pristupima dinamičkog programiranja [4], ali postoje i radovi koji se bave semantičkom segmentacijom koristeći metode dubokog učenja. Chiu et al. u radu "*Automatic segmentation of seven retinal layers in SDOCT images congruent with expert manual segmentation*" koriste gradijente i algoritam najkraćeg puta za procjenu granica slojeva [4]. Slično, Karri et al. predlažu ojačanje konstrukcija grafa učenjem granica specifičnih svakom sloju koristeći strukturirane slučajne šume u radu "*Learning layer-specific edges for segmenting retinal layers with large deformations*" [4]. Fang et al. su u radu "*Automatic segmentation of nine retinal layer boundaries in OCT images of non-exudative AMD patients using deep learning and graph search*" udružili konvolucijske neuronske mreže s metodama pretraživanja grafa za segmentaciju devet granica slojeva retine [4].

Long et al. autori su jednog od najranijih radova na temu potpuno konvolucijske semantičke segmentacije "*Fully convolutional networks for semantic segmentation*" u kojem prilagođavaju mreže namijenjene za klasifikaciju u potpuno konvolucijske mreže podešene za problem segmentacije [19]. Predstavili su **koncept prečaca** (engl. *skip connections*) [19] koji koriste **više informacija pri zadržavanju segmentacijskih detalja**. Daljnji napredak postigli su Noh et al. u sklopu rada "*Learning Deconvolution Network for Semantic Segmentation*" (popularno zvanog **DeconvNet**) korištenjem **enkodera i dekodera** [18] te **slojeva za proširivanje (engl. *unpooling*)** za održavanje prostorne konzistentnosti umjesto interpolacije [18].

Iako postoje pristupi dubokog učenja koji se fokusiraju na segmentaciju, pojavljuje se problem da takve metode ne uzimaju u obzir nakupine tekućine u oku što može dovesti do pogreški. Uz već navedena ograničenja, testna faza metoda koje se temelje na pretraživanju stabla je spora zbog optimizacije grafova koja se mora provesti. To onemogućava njihovu upotrebu u vremenski ograničenim situacijama kao što su hitne intervencije.

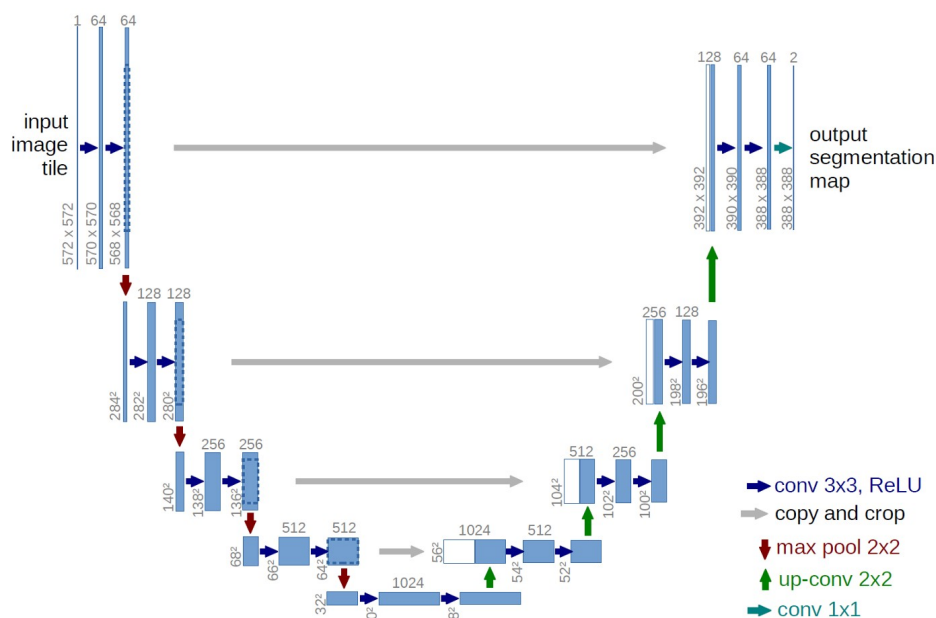
Iz tog razloga je predložena ReLayNet arhitektura kao state-of-the-art metoda za rješavanje problema segmentacije slojeva retine.

1.3. Problem segmentacije

Glavni cilj segmentacije je podijeliti sliku na nekoliko dijelova, gdje svaki od njih predstavlja određeni entitet. Čovjeku se segmentacija slika čini kao vrlo jednostavan zadatak jer ga vrlo lako činimo: vidimo prizore iz svakidašnjice i jasno razaznajemo različite objekte na njima, razlikujemo auto od ostatka ulice, komad papira od ostatka stola itd., ali treba razmisliti koliko je takav zadatak kompliciran za računalo. Ono što nama predstavljaju slike, računalo vidi kao matrice i kombinacije običnih brojki. Razvojem umjetne inteligencije, poglavito raznih arhitektura zasnovanih na **konvolucijskim neuronskim mrežama** (engl. *Convolutional Neural Network*, CNN) [5] postignut je veliki napredak u rješavanju problema koji su se prethodno činili prekompleksnima za računalo. **AlexNet** je naziv jedne takve mreže čiji je rezultat pokazao da je **dubina modela vrlo važna** za visoke performanse, što je računski skuplje, ali nadoknađeno korištenjem grafičkih procesora [10].

Jedan takav problem je upravo segmentacija slike. Segmentacija se koristi u brojnim područjima kao što je industrija samovozećih automobila ili čak u satelitima [12]. Također je vrlo značajna u području medicinskog snimanja [12] jer izlučuje i analizira područja interesa automatiziranim procesom [13]. Različiti detalji, variranja i suptilnosti u medicinskim snimkama mogu biti izrazito komplicirani i njihovo razlikovanje ponekad izazovno čak i za školovane liječnike [4]. U tom području provode se brojna istraživanja jer bi program koji bi bio sposoban razlikovati takve detalje i identificirati bitna područja imao znatan utjecaj u području medicinskih snimanja i medicine općenito.

CNN su u početku davale dobre rezultate primijenjene na lakše segmentacijske zadatke, ali su bile **nedovoljno dobre u kompliciranijim zadacima** [12]. Pokazala se potreba za **drukčijim pristupom** ovakvom problemu - Olaf Ronneberger, Philipp Fischer i Thomas Brox su u radu “**U-Net: Convolutional Networks for Biomedical Image Segmentation**” [6] predložili simetričnu U-Net arhitekturu koja je ime dobila po svojoj sličnosti slovu *U*. U-Net je prvotno razvijen s ciljem primjene u području segmentacije biomedicinskih snimki, ali su rezultati bili toliko napredniji od do tada poznatih pristupa da je arhitektura dobila primjenu i u brojnim drugim poljima [6].

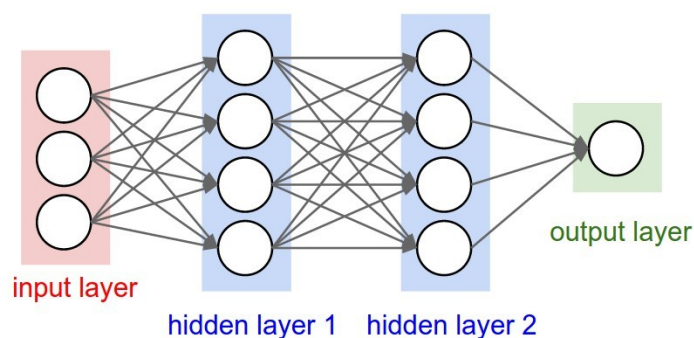


Slika 1.2: Predložena U-Net arhitektura [6]

Kod problema klasifikacije, učenje značajki slike radi vrlo dobro jer se slika pretvara u vektor koji se koristi u daljnjoj klasifikaciji [10], ali kod problema segmentacije nije dovoljno samo pretvoriti sliku u vektor, već iz tog vektora **treba rekonstruirati sliku**. To nas dovodi do glavnog koncepta U-Net arhitekture (i sličnih arhitektura kao što je ReLayNet koji je njime inspiriran), **korištenje istih značajki koje smo dobili kontrahiranjem slike u vektor pri ekspaniranju u sliku**. Ovo **čuva strukturalni integritet** same slike i **smanjuje distorziju** [12].

2. Osnove neuronskih mreža

Neuronska mreža sastoji se od međusobno povezanih neurona, a veze između neurona imaju određenu težinu koja predstavlja važnost te povezanosti [7]. Svaki neuron ima aktivacijsku funkciju koja definira izlaz neurona i uvodi nelinearnost u modeliranje mreže. Treniranje neuronske mreže možemo promatrati kao proces učenja kroz iteriranje u oba smjera kroz različite slojeve mreže koji se sastoje od neurona.



Slika 2.1: Primjer jednostavne neuronske mreže [5]

Prolazak unaprijed (engl. *forward propagation*) je prostiranje informacija kroz mrežu – mreži je dodijeljen početni skup informacija za trening koje prođu čitavu mrežu kako bi njihove predikcije bile izračunate, dok prolazak „unazad” predstavlja postupak propagacije pogreške unatrag (engl. *error backpropagation*) – počevši od zadnjeg sloja, informacija o gubitku (engl. *loss*) odnosno pogrešci se propagira neuronima koji direktno doprinose tom rezultatu. Neuroni ne dobivaju cijelu informaciju o pogrešci već samo njen dio, ovisno o stupnju njihovog utjecaja na rezultat [7].

Funkcija gubitka se koristi kako bi imali informaciju o pogrešci te kako bi se ona kvantificirala što nam omogućuje usporedbu dobivene predikcije i točnog rezultata.

Dakle, **treniranjem modela, težine određenih veza se postepeno prilagođavaju** u nadi poboljšanja predikcija. Radimo sljedeće – trudimo se smanjiti gubitak odnosno pogrešku. Za to koristimo tehniku zvanu gradijentni spust koja mijenja težine u malim inkrementima uz pomoć računanja derivacije (gradijenta) funkcije gubitka iz koje možemo vidjeti u kojem smjeru se trebamo „spuštati” prema globalnom minimumu [7].

2.1. Gradijentni spust

Gradijentni spust je osnova nekoliko optimizatora i jedan od najčešćih optimizacijskih algoritama s primjenom u strojnom i dubokom učenju. Gradijentni spust koristi prvu derivaciju (gradijent – **nagib funkcije** u određenoj točki) funkcije pogreške pri **ažuriranju parametara** [7]. Gradijent je uvijek smjera u kojem se vrijednost funkcije pogreške povećava, stoga smjer smanjenja funkcije pogreške dobivamo korištenjem negativnog gradijenta. Algoritam gradijentnog spusta ponavlja ovaj proces približavajući se minimumu dok vrijednost parametra ne dođe do točke u kojoj se funkcija pogreške ne može dodatno smanjiti.

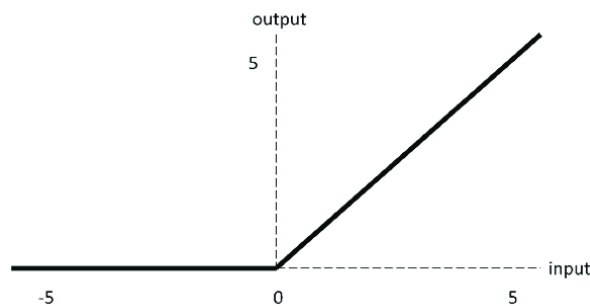
Jedan od optimizatora u kojem se koristi gradijentni spust je Adam [8], koji se koristio u treniranju implementacije ReLayNeta koja se koristila za analizu performansi.

2.2. ReLU

ReLU (engl. *rectified linear unit*) je **aktivacijska funkcija** koja **aktivira određeni neuron ako je vrijednost ulaza iznad određenog praga**. U praksi to zapravo znači da je izlaz jednak nuli dok god je ulaz manji od nule, a ako je ulaz pozitivan, izlaz linearno ovisi o ulazu:

$$f(x) = x \quad [7]$$

ReLU se pokazala kao aktivacijska funkcija primjerena za veliki broj scenarija i trenutno je u širokoj uporabi upravo zbog svojih dobrih rezultata.



Slika 2.2: ReLU funkcija [7]

2.3. Softmax

Softmax aktivacijska funkcija vraća **raspodjelu vjerojatnosti po ciljnim klasama** [7]. Izlazne vrijednosti su u granicama $[0, 1]$ što **omogućuje izbjegavanje binarne klasifikacije** i rad s proizvoljnim brojem klasa i dimenzija u našem modelu. Iz tog razloga se Softmax ponekad naziva i multinomijalna logistička regresija. Softmax pretvara numerički izlaz iz zadnjeg sloja mreže u vjerojatnosti uzimajući eksponent svakog izlaza i provođenjem normalizacije zbrojem tih eksponenata tako da je ukupan zbroj izlaza jednak 1, zato što suma vjerojatnosti matematički mora biti 1.

$$\text{Softmax}(x_i) = \frac{\exp(x_i)}{\sum_j x_j} \quad [7]$$

2.4. Epohe

Broj epoha je informacija koja nam govori **koliko puta je cjelokupni skup podataka za treniranje prošao kroz neuronsku mrežu** u procesu učenja iste. Dobra praksa je povećavati broj epoha dok se metrika točnosti u usporedbi sa skupom podataka za validaciju ne počne smanjivati, čak i kada metrika točnosti skupa podataka za učenje još uvijek raste [15].

Tako možemo detektirati potencijalni problem prenaučene mreže, odnosno mreže koja izvrsno radi na podacima za učenje, ali gubi svojstvo generalizacije [15] te to nastojimo izbjeći.

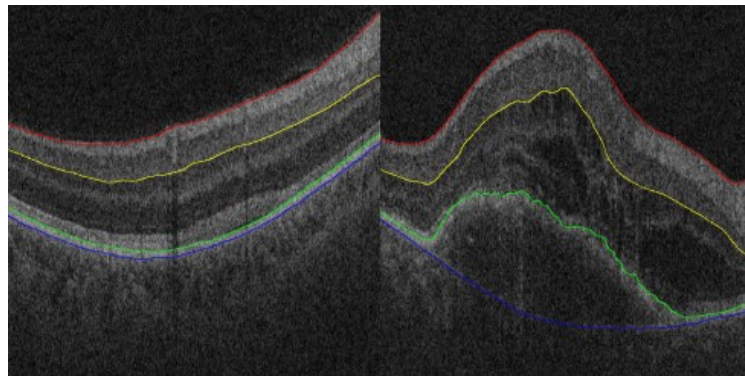
2.5. Veličina grupe

Imamo mogućnost podijeliti skup podataka za treniranje u **manje grupe podataka za prolaz kroz mrežu** – takav parametar nazivamo veličina grupe (engl. *batch size*). Optimalna veličina ponajviše ovisi o dostupnoj memoriji računala koje vrši izračune.

U provedenim eksperimentima korišten je *batch size* od četiri snimke.

3. Metodologija

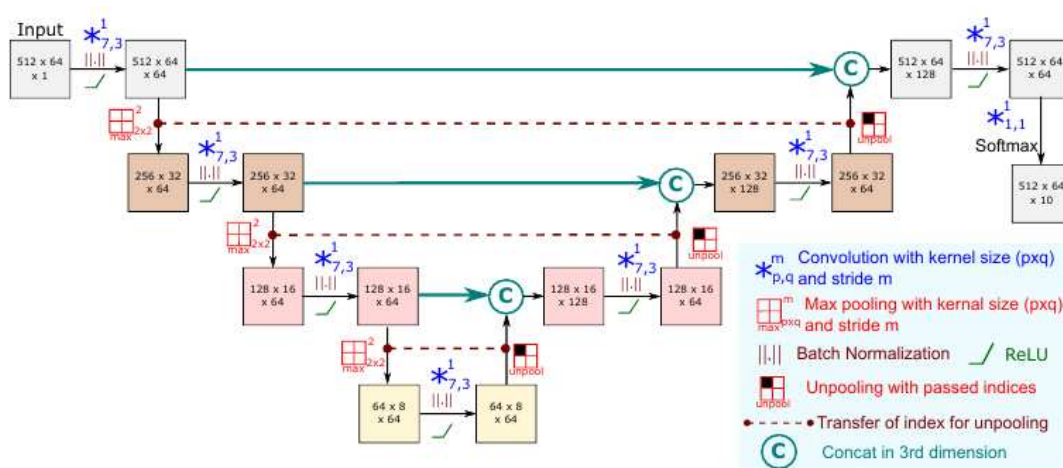
Zadatak je na dobivenoj OCT snimci retine, svakom pikselu na lokaciji $x=(i,j)$ dodijeliti oznaku l iz skupa oznaka $L=\{1,\dots,K\}$ gdje K označava broj klasa. U sklopu ovog završnog rada ovaj zadatak tretiramo kao višeklasni segmentacijski problem s četiri klase. Četiri klase u našem problemu su sljedeće: pozadina, područje između granice ILM i granice IPL/INL, područje između granice IPL/INL i granice RPE te pigmentni epitel retine odnosno njegovo odvajanje (engl. retinal pigment epithelium, RPE te pigment epithelial detachment, PED) koje je prisutno kod pacijenata koji boluju od senilne makularne degeneracije ili dijabetičke retinopatije [3].



Slika 3.1: Primjer anotiranih *B-scanova* iz domaće baze podataka

3.1. Arhitektura

Arhitektura ReLayNeta sastavljena je od **kontrahirajuće sekvence konvolucijskih blokova za enkodiranje** (u daljnjem tekstu enkodera), kojega slijedi **ekspanzivna sekvenca konvolucijskih blokova za dekodiranje** (u daljnjem tekstu dekodera) s poveznicama između svakog enkodera i pripadnog mu dekodera kroz slojeve konkatencije [4]. Zadnji sloj arhitekture je klasifikacijski sloj koji određuje pripadnost pojedinog piksela jednoj od četiri navedene klase.



Slika 3.2: predložena arhitektura ReLayNet mreže [4]

3.1.1. Enkoderi

Svaki konvolucijski blok za enkodiranje sastoji se od četiri sloja, redom: konvolucijski sloj, normalizacijski sloj, ReLU te sloj za sažimanje maksimalnom vrijednošću (engl. *max pooling*) [4]. Konvolucijski sloj koristi jezgre veličine 7 x 3 za konzistentnost s dimenzijama OCT snimki. Normalizacijski sloj slijedi nakon konvolucijskog kako bi izbjegli prenaučenosť tijekom učenja, a već spomenuta ReLU funkcija u naš model uvodi nelinearnost. ReLU nam je važna jer bi bez nje imali mrežu koja bi se ponašala kao jedan neuron neovisno o broju slojeva u našoj arhitekturi - jer je zbroj linearnih funkcija opet linearan. Max pooling sloj smanjuje dimenzije, a indeksi koji su korišteni pri sažimanju su spremljeni i ponovno iskorišteni u sloju proširivanja (engl. *unpooling*) pripadnog dekodera kako bi se očuvala prostorna konzistentnost [4].

3.1.2. Dekoderi

Svaki konvolucijski blok za dekodiranje sastoji se od pet slojeva, redom: sloj proširivanja, konkatencija, konvolucijski sloj, normalizacijski sloj te ReLU aktivacijska funkcija [4].

Unpooling sloj vrši naduzorkovanje (engl. *upsampling*) mape značajki (engl. *feature map*) prethodnog dekodera u veću rezoluciju i pritom koristi spremljene indekse korištene u sažimanju pripadnog enkodera, dok na ostala mjesta postavlja nule. Motivacija ovakvog sloja je **sačuvanje prostorne informacije** koja bi se izgubila kad bi koristili npr. interpolaciju za proces naduzorkovanja [4]. Unpooling sloj u ovakvom segmentacijskom problemu je vrlo bitan za točnu segmentaciju slojeva jer postoje slojevi koji su često tanki samo nekoliko piksela, što bi rezultiralo u vrlo difuznim granicama kad bi koristili bilinearnu interpolaciju i samim time nepouzdanom procjenom debljine sloja. Nakon konkatencije slijedi konvolucijski sloj, normalizacijski sloj i ReLU. Njihova uloga u dekodernu je takva da **obogate mape značajki** dobivene unpoolingom. Kao i kod enkodera, koristi se jezgra konstantne veličine 7×3 .

3.1.3. Klasifikator

Zadnji dekodern slijedi konvolucijski sloj s jezgrom veličine 1×1 . Ovakav sloj služi kako bi smanjili broj kanala mape značajki bez da promijenimo prostorne dimenzije [4]. Zatim uparujemo dobivenu mapu značajki s četverokanalnom mapom značajki (za četiri klase našeg segmentacijskog problema). **Softmax** funkcija kao završni sloj procjenjuje **pripadnost piksela jednoj od četiri klase** za sve piksele snimke.

3.2. Treniranje modela

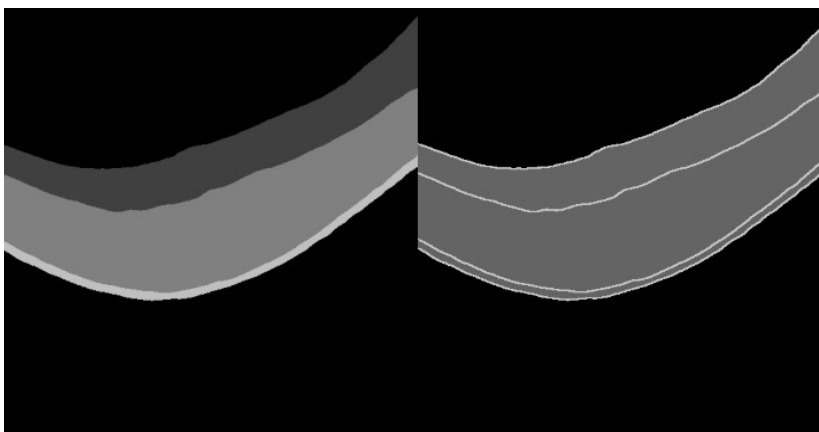
3.2.1. Funkcija pogreške

ReLayNet model je treniran zajedničkom optimizacijom dvije funkcije pogreške. Jedna od njih je višeklasna logistička pogreška koja uzima u obzir težine piksela. Unakrsna entropija (engl. *cross entropy*) nam daje vjerojatnosnu sličnost između prave oznake (engl. *ground truth*) i predviđene vrijednosti pri trenutnom stanju mreže. Prosječna unakrsna entropija svih klasa definira logističku pogrešku koja kažnjava devijaciju predviđene vjerojatnosti $p(x)$ od oznake l za svaki piksel [4].

$$f_{\logloss} = - \sum_{x \in \Omega} \omega(x) g_l(x) \log(p_l(x)) \quad [4]$$

Koristimo težine u izračunu logističke pogreške kako bi kompenzirali za različite stupnjeve zastupljenosti pojedinih slojeva u cijeloj snimci retine te kako bi segmentacija bila posebno osjetljiva na prijelaze odnosno granice između slojeva [4]. Kako bi povećali osjetljivost mreže na granice, množimo vrijednosti piksela na granicama slojeva faktorom ω_1 . Slojevi retine su također vrlo podzastupljeni s obzirom na prevladavajuću klasu pozadine što se pokušava uravnotežiti množenjem vrijednosti na području slojeva faktorom ω_2 .

$$\text{Pozadina: } \omega(x)=1 \quad \text{Sloj: } \omega(x)=1+\omega_2 \quad \text{Granica: } \omega(x)=1+\omega_1 \quad [4]$$



Slika 3.3: Primjer označenih slojeva OCT snimke i pripadnih težina gdje svjetlija boja označava veću težinu piksela

Druga komponenta funkcije pogreške je **Diceov gubitak** (engl. *Dice loss*), funkcija koja razmatra **prostorno preklapanje predikcija sa stvarnim oznakama** (*ground truthom*). Korišten je diferencijabilni oblik Diceovog gubitka čija je osnova Sørensen-Diceov koeficijent, statistički alat za mjerenje sličnosti između dva uzorka.

$$f_{dice} = 1 - \frac{2 \sum_{x \in \Omega} p_l(x) g_l(x)}{\sum_{x \in \Omega} p_l^2(x) + \sum_{x \in \Omega} g_l^2(x)} \quad [4]$$

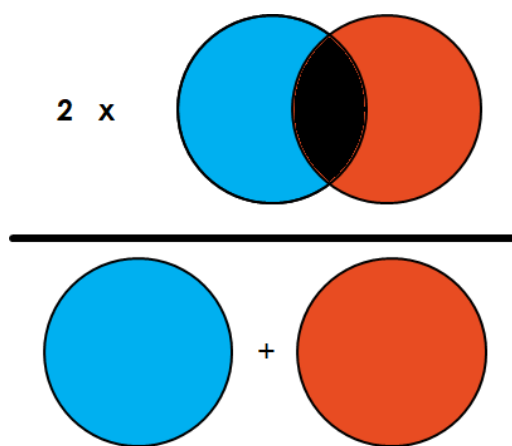
3.2.2. Metrika točnosti

Kako bi kvantificirali uspješnost segmentacije naših modela, trebamo metriku koja će na neki način pratiti točnost segmentacije. Početno je kao metrika točnosti odabrana metrika točnosti pojedinih piksela. Ta metrika je konceptualno vrlo jednostavna i izražava se u postotku piksela slike koji su ispravno klasificirani te se na prvi pogled čini kao dobar pokazatelj točnosti našeg treniranog modela. Međutim, usporedbom rezultata metrike te stvarnim predikcijama modela, pokazalo se kako ovakva metrika nije najbolji pokazatelj točnosti kod segmentacijskih problema poput našeg, gdje postoji neuravnoteženost između zastupljenosti pojedinih klasa.

Primjera radi, uzmimo sliku dimenzija 100 x 100 piksela gdje razlikujemo objekte i pozadinu, a pozadina čini 90% cijele površine. Primjenom modela na sliku, dobijemo predikciju da je cijela slika pozadina, odnosno model sve piksele svrsta u klasu pozadine. Primjenom metrike točnosti pojedinih piksela dobili bi točnost od 90%, a rezultat predikcije bio bi praktički beskoristan jer niti jedan objekt nije uspješno segmentiran. Vidimo da visoka točnost piksela nije uvijek pokazatelj dobre sposobnosti segmentacije modela te se kao prikladnija metrika točnosti za problem segmentacije pokazao Dice koeficijent [9].

Dice koeficijent pojedinog sloja jednak je vrijednosti dvostruke površine preklapanja predikcije modela i *ground trutha* podijeljenoj s ukupnim brojem piksela obje slike [9], ilustrirano na slici niže.

Primijenimo metriku Dice koeficijenta na prethodni primjer: za pozadinski sloj preklapanje iznosi 90% te dobijemo koeficijent jednak $(2 * 0.9 * 100 * 100) / (2 * 100 * 100) = 0.9$, dok za klasu koju predstavljaju objekti dobijemo Dice koeficijent 0, jer nema preklapanja. To znači da je prosječni Dice koeficijent i mjera točnosti predikcije našeg modela jednaka srednjoj vrijednosti koja iznosi $0.45 = 45\%$ i vidimo kako je za ovakav segmentacijski zadatak puno informativnija uzevši u obzir prirodu problema.



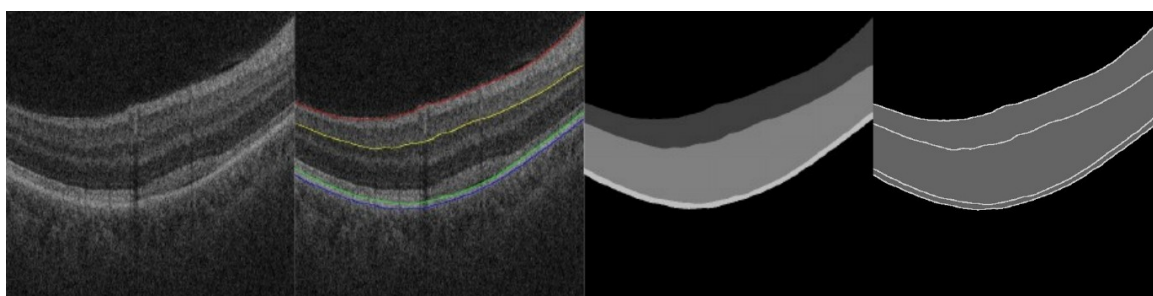
Slika 3.4: Vizualizacija izračuna Dice koeficijenta [9]

4. Baza podataka

Koristimo bazu podataka s anotiranim slojevima iz KBC Sestre milosrdnice. Baza podataka sastoji se od volumena (engl. *C-scanova*) od 25 pacijenata, a svaki *C-scan* sadrži 128 *B-scanova*. Za snimanje je korišten uređaj Zeiss Cirrus HD OCT 4000.

Slike iz baze podataka su OCT snimke oka s 10 slojeva i 11 granica. Na slikama je crvenom bojom označen ILM - prijelaz između staklovine i retine, granica između IPL i INL žutom bojom, gornja granica RPE sloja zelenom bojom te Bruchova membrana plavom bojom. Bruchova membrana se kod zdravih osoba nalazi odmah uz RPE, dok je kod patoloških promjena odignuta.

Liječnici su anotirali slojeve za koje smatraju da bi dobivene predikcije segmentacije najviše pomogle pri dijagnozama pacijenata s nekom vrstom oboljenja koja utječe na retinu oka. Pri anotiranju naglasak je stavljen na ILM i RPE kao rubne granice retine. Kod određivanja intraretinalnih i subretinalnih nakupina tekućine, IRF i SRF, važna je njihova pozicija relativna granicama. Granica IPL i INL sloja odabrana je jer se u središtu spaja s ILM i može pomoći pri određivanju fovee. Bruchova membrana je odabrana jer se njenim položajem vidi postoji li PED između nje i RPE.



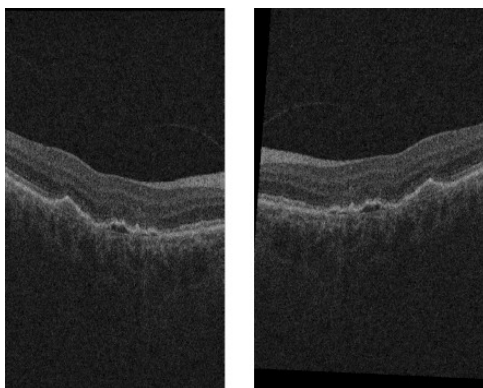
Slika 4.1: Primjer OCT snimke iz dataseta: originalna snimka, anotirana snimka, *ground truth* maska te mapa težina

4.1. Prilagodba dataseta ReLayNet arhitekturi

Kako bi prilagodili skup podataka ReLayNetu, bilo je potrebno napisati poseban razred koji nasljeđuje `torch.utils.data.Dataset`, apstraktni razred koja predstavlja skup podataka te implementirati metode za dohvat veličine baze podataka te pojedinih elemenata. U metodi za dohvat pojedine snimke bilo je potrebno napraviti izmjenu originalne slike tako da se trokanalna RGB slika pretvori u jednokanalnu *grayscale* sliku te vrijednosti normaliziraju u granicama $[0,1]$. Kako se baza podataka sastoji od originalnih snimki i snimki s označenim granicama, bilo je potrebno ručno napraviti *ground truth* slike koje označavaju četiri sloja te normalizirati njihove vrijednosti na cijele brojeve $\{0,1,2,3\}$ koji predstavljaju klase. Isto tako, težine granica i slojeva za unakrsnu entropiju su dodane kao dio skupa podataka u obliku slika (Slika 4.1).

4.2. Augmentacija podataka

Pri dohvatima pojedine snimke uvodi se i augmentacija podataka u obliku nasumičnih translacija u prostoru, blagih rotacija te horizontalnog zrcaljenja slike. Kako je bilo potrebno primijeniti istu augmentaciju na sve tri slike u jednom elementu dataseta, korištena je `transforms.functional` biblioteka iz paketa `torchvision` koja omogućuje veću kontrolu nad procesom transformacije u smislu da se isti parametri pojedine metode transformacije primijene na sve tri slike - original, masku te mapu težina.



Slika 4.2: originalni B-scan prije i poslije nasumične augmentacije

5. Eksperimenti i rezultati

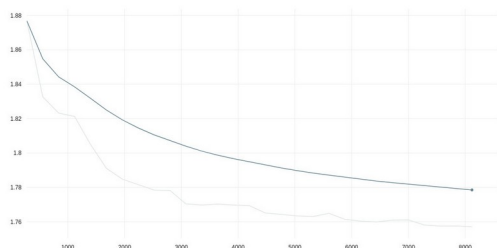
U sklopu ovog završnog rada provedeno je nekoliko eksperimenata kroz treniranje modela ReLayNet arhitekture neuronske mreže koja je implementirana u PyTorchu [14]. U svim eksperimentima kao optimizator pri učenju korišten je Adam optimizator [8] uz snižavanje stope učenja faktorom 0.5 svakih 5 epoha. U treniranju je korišten tzv. *early stopping* tako da se učenje zaustavlja ako metrika točnosti na validacijskom skupu podataka ne raste prethodnih nekoliko epoha te su svi eksperimenti pušteni da se izvode dok metrika točnosti raste. Svi eksperimenti su provedeni na istoj bazi podataka, podijeljenoj na skup podataka za učenje (kojega čini 70% pacijenata), validaciju (20% pacijenata) i testiranje (10%). Dodatno, skup za učenje je svakim eksperimentom izmiješan. Kao alat za lakše praćenje i vizualizaciju brojnih podataka o modelima i eksperimentima korišten je Comet - platforma koja omogućava lakše praćenje i usporedbu eksperimenata u području strojnog učenja [20].

5.1. Model 1: unakrsna entropija s težinama

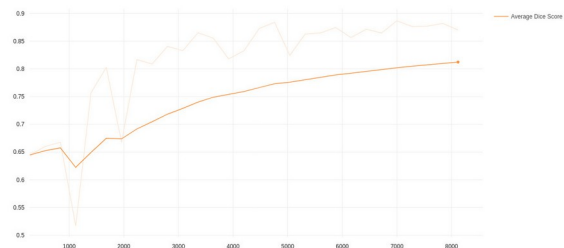
Kao prvi eksperiment provedena je varijanta modela koji koristi unakrsnu entropiju (logističku pogrešku) kao jedinu komponentu cjelokupne funkcije pogreške. Eksperiment je konvergirao (učenje je zaustavljeno) nakon 29 epoha. Od provedenih eksperimenata, ovaj eksperiment je rezultirao najnižim rezultatima točnosti segmentacije slojeva.

Pozadina	0.9854077237528751
Područje između ILM i granice IPL/INL	0.9516428549643670
Područje između granice IPL/INL i granice RPE	0.9390008965269274
Pigmentni epitel retine	0.5280693914314474
Prosječni Dice koeficijent	0.8510302166689041

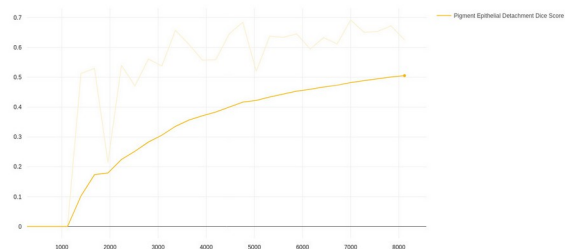
Tablica 5.1: Dice koeficijenti modela primijenjenog na testni skup podataka



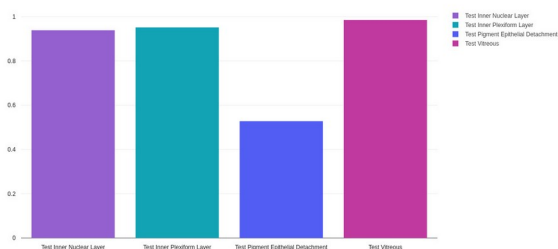
Slika 5.1: Pogreška po epohama učenja



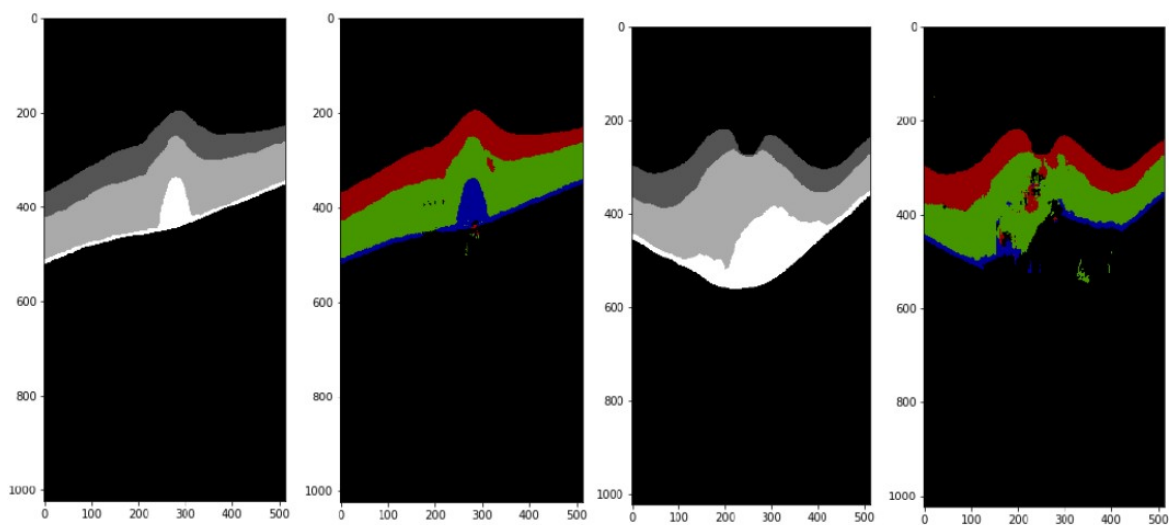
Slika 5.2: Prosječni Dice koeficijent po epohama na validacijskom skupu podataka



Slika 5.3: Dice koeficijent sloja pigmentnog epitela retine na validacijskom skupu podataka



Slika 5.4: Dice koeficijenti slojeva na testnom skupu podataka. S lijeva na desno: područje između ILM i granice IPL/INL, područje između granice IPL/INL i granice RPE, pigmentni epitel retine, pozadina



Slika 5.5: Kvalitativna usporedba *ground truth* maske i predikcije boljeg i lošijeg primjera

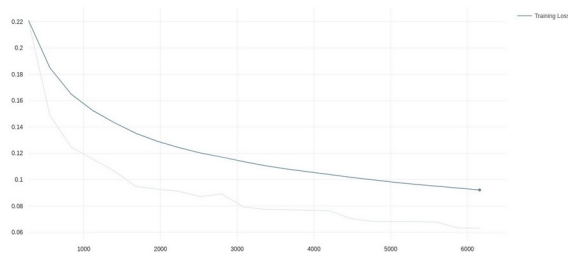
5.2. Model 2: Diceov gubitak

Kao drugi eksperiment provedena je varijanta modela koji koristi Dice pogrešku kao jedinu komponentu funkcije pogreške. Korištenjem unakrsne entropije statistička raspoređenost klasa igra veliku ulogu u uspješnosti učenja modela - što su klase neuravnoteženije to će treniranje biti otežanije [17]. Težine se u unakrsnoj entropiji koriste kako bi olakšale učenje, ali to ne rješava generalni problem pogreške. U unakrsnoj entropiji, pogreška se računa kao prosječna vrijednost po pojedinačnom pikselu [17] što znači da unakrsna entropija pogrešku ne razmatra na dovoljno globalnoj razini što je bitan faktor za predikcije na razini cijele slike.

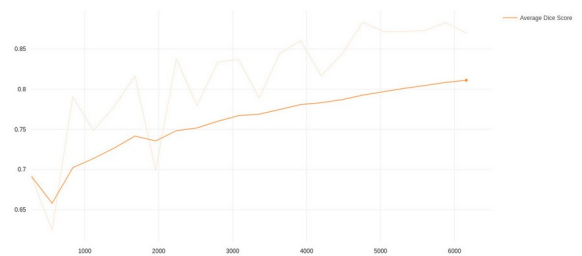
Kao što je već navedeno u dijelu ovoga rada koji opisuje Dice koeficijent, brojnik koeficijenta predstavlja preklapanje dva skupa na lokalnoj razini dok je nazivnik jednak ukupnom broju piksela na globalnoj razini. Dakle, Diceov gubitak uzima u obzir informaciju o pogrešci lokalno i globalno, što je važno za veću točnost segmentacije. Eksperiment je konvergirao nakon 22 epohe.

Pozadina	0.9848121086474871
Područje između ILM i granice IPL/INL	0.9550577056872942
Područje između granice IPL/INL i granice RPE	0.9295930821590238
Pigmentni epitel retine	0.5465539185934577
Prosječni Dice koeficijent	0.8540042037718156

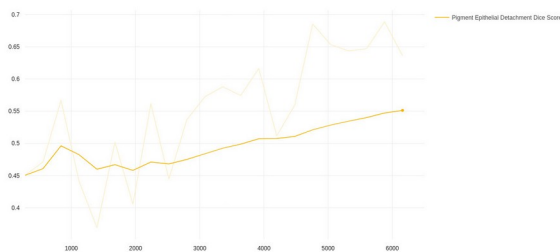
Tablica 5.2: Dice koeficijenti modela primijenjenog na testni skup podataka



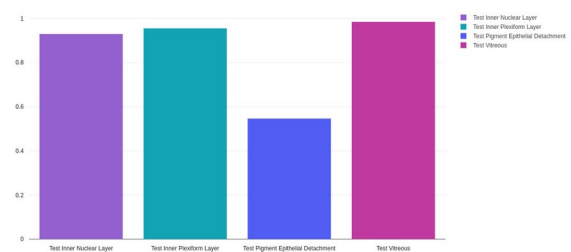
Slika 5.6: Pogreška po epohama učenja



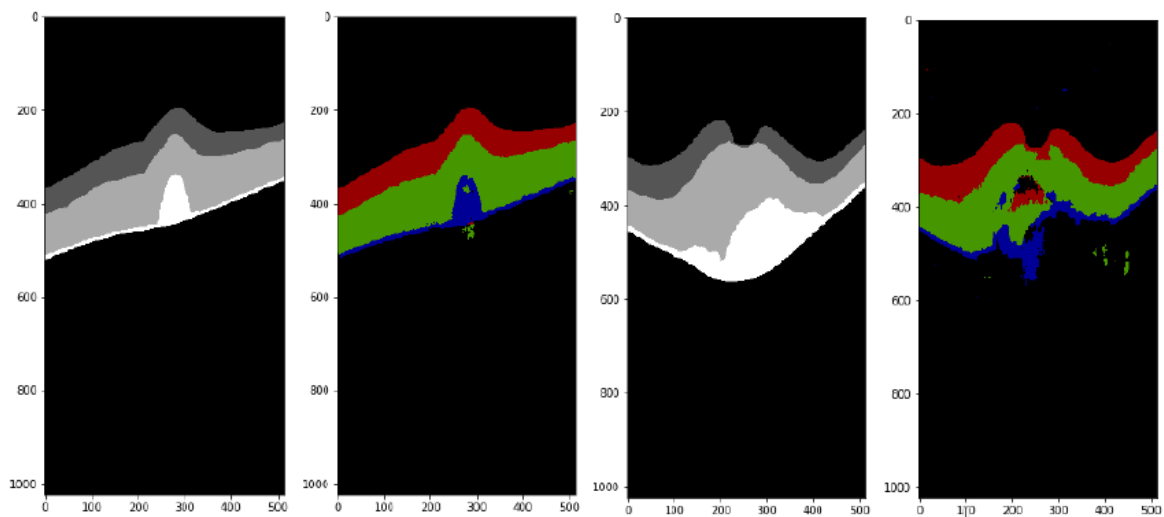
Slika 5.7: Prosječni Dice koeficijent po epohama na validacijskom skupu podataka



Slika 5.8: Dice koeficijent sloja pigmentnog epitela retine na validacijskom skupu podataka



Slika 5.9: Dice koeficijenti slojeva na testnom skupu podataka. S lijeva na desno: područje između ILM i granice IPL/INL, područje između granice IPL/INL i granice RPE, pigmentni epitel retine, pozadina



Slika 5.10: Kvalitativna usporedba *ground truth* maske i predikcije boljeg i lošijeg primjera

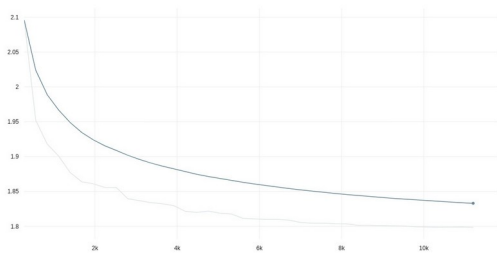
5.3. Model 3: udružena pogreška

U originalnom *ReLayNet* radu predloženo je korištenje udružene funkcije pogreške, one koje u obzir uzima unakrsnu entropiju s težinama i Diceov gubitak. Logika je sljedeća: pojedina komponenta pogreške prioritizira različite značajke naših snimki. Diceov gubitak stavlja veći značaj na neuravnoteženost klasa dok unakrsna entropija favorizira generalnu točnost po pikselima. To znači da bi segmentacija nekih slojeva mogla davati lošije rezultate ako nisu dovoljno zastupljeni da bi imali utjecaja na unakrsnu entropiju. Zbrajanjem ove dvije komponente trudi se imati koristi od točnosti segmentacije slojeva putem *Dice* koeficijenta, ali i točnije određenim granicama između slojeva korištenjem težina u unakrsnoj entropiji. Eksperiment je konvergirao nakon 40 epoha.

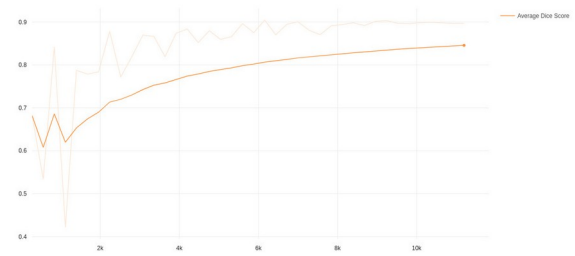
Ovaj model rezultirao je najboljim rezultatima segmentacije slojeva te predikcijama primijenjen na testni skup podataka.

Pozadina	0.986851585342877
Područje između ILM i granice IPL/INL	0.9635760961026184
Područje između granice IPL/INL i granice RPE	0.9428518091067815
Pigmentni epitel retine	0.6059256519718582
Prosječni Dice koeficijent	0.8748012856310339

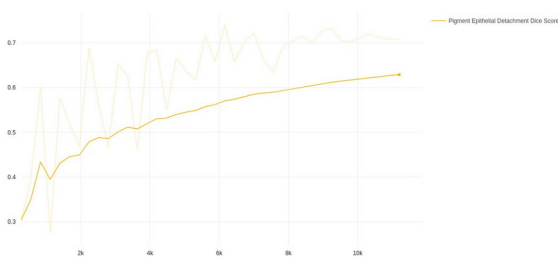
Tablica 5.3: Dice koeficijenti modela primijenjenog na testni skup podataka



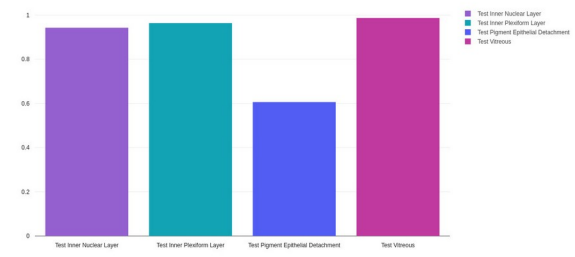
Slika 5.11: Pogreška po epohama učenja



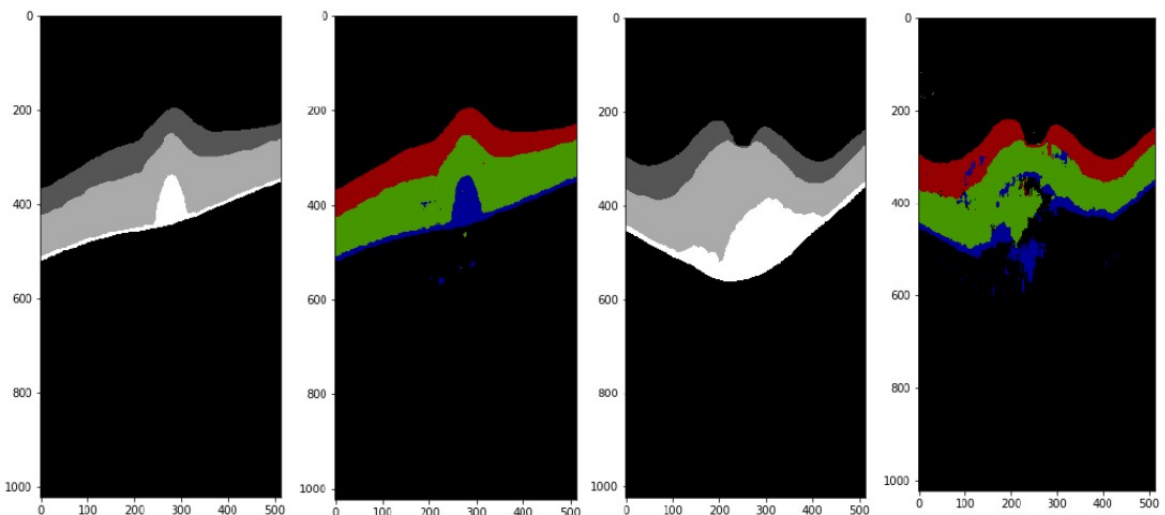
Slika 5.12: Prosječni Dice koeficijent po epohama na validacijskom skupu podataka



Slika 5.14: Dice koeficijent sloja pigmentnog epitela retine na validacijskom skupu podataka



Slika 5.13: Dice koeficijenti slojeva na testnom skupu podataka. S lijeva na desno: područje između ILM i granice IPL/INL, područje između granice IPL/INL i granice RPE, pigmentni epitel retine, pozadina



Slika 5.15: Kvalitativna usporedba *ground truth* maske i predikcije boljeg i lošijeg primjera

Zaključak

Ovim radom nastojalo se prikazati rezultate primjene ovakve metode na domaću bazu podataka pacijenata sa senilnom makularnom degeneracijom. Opisan je način rada i nekoliko osnovnih koncepata i svojstava neuronskih mreža te specifičnosti arhitektura s naglaskom na problem semantičke segmentacije poput ReLayNet arhitekture koja se temelji na U-Netu. Razvijena je implementacija ReLayNeta korištenjem biblioteke PyTorch pomoću koje je definirana duboka konvolucijska neuronska mreža te maksimalno iskorištena efikasnost učenja koristeći grafičke procesore u svrhu obrade podataka.

Duboko učenje ReLayNet arhitekturom za segmentaciju slojeva dalo je vrlo dobre rezultate za pozadinski sloj, unutarnji mrežasti sloj i unutarnji zrnati sloj dok se pokazalo kako segmentacija daje lošije rezultate za sloj pigmentnog epitela retine ako je sloj deformiran, odnosno ako postoji odvojenje Bruchove membrane - što je slučaj kod patoloških promjena. Unatoč tome, udruživanje unakrsne entropije i Diceovog gubitka dalo je najbolje rezultate.

Lošija segmentacija epitela retine bi se mogla pripisati činjenici da je taj sloj najpodložniji deformacijama i varijacijama, ali i tome da je skup podataka iz KBC-a Sestre milosrdnice anotiran s četiri sloja, iako ih u retini ima sveukupno deset. Pokazalo se kako manji broj klasa može imati negativan utjecaj na rezultate segmentacije jer se s manjim brojem klasa smanjuje točnost [11]. Slične značajke između slojeva na snimci mogu rezultirati lažno pozitivnim predikcijama i samim time netočnom detekcijom. Možemo pretpostaviti da bi segmentacija bila uspješnija kad bi imali skup podataka s anotiranim svim slojevima i granicama za učenje te bi u daljnjem radu to valjalo i provjeriti.

Literatura

- [1] N. Nassif, B. Cense, B. H. Park, S. H. Yun, T. C. Chen, B. E. Bouma, G. J. Tearney, J. F. de Boer. *In vivo human retinal imaging by ultrahigh-speed spectral domain optical coherence tomography*. Optics Letters, 29 (5), 480-482, (2004)
- [2] E. D. Cole. *Artificial Intelligence in Retina*, Američka oftalmološka akademija. Residents and Fellows (2019)
- [3] A. Garcia-Layana, G. Ciuffo, J. Zarranz-Ventura, A. Alvarez-Vidal. *Optical Coherence Tomography in Age-related Macular Degeneration*. Ophthalmology Department. Clínica Universidad de Navarra. Pamplona. Spain (2017)
- [4] A. Guha Roy, S. Conjeti, S.P.K. Karri, D. Sheet, A. Katouzian, C. Wachinger, N. Navab. *ReLayNet: Retinal Layer and Fluid Segmentation of Macular Optical Coherence Tomography using Fully Convolutional Networks*. Biomed. Opt. Express 8, 3627-3642 (2017)
- [5] *CS231n: Convolutional Neural Networks for Visual Recognition*. Stanford University
- [6] O. Ronneberger, P. Fischer, and T. Brox. *U-net: Convolutional networks for biomedical image segmentation*. International Conference on Medical Image Computing and Computer-Assisted Intervention (2015)
- [7] J. Torres. *Learning Process of a Deep Neural Network*. Medium, Towards Data Science (2020). Poveznica: <https://towardsdatascience.com/learning-process-of-a-deep-neural-network-5a9768d7a651>
- [8] D. P. Kingma, J. Ba. *Adam: A Method for Stochastic Optimization*. 3rd International Conference for Learning Representations, San Diego, (2015)

- [9] E. Tiu. *Metrics to Evaluate your Semantic Segmentation Model*. Medium (2019).
Poveznica: <https://towardsdatascience.com/metrics-to-evaluate-your-semantic-segmentation-model-6bcb99639aa2>
- [10] A. Krizhevsky, I. Sutskever, G. E. Hinton. *ImageNet Classification with Deep Convolutional Neural Networks*. ImageNet Visual Recognition Challenge (2012)
- [11] J. Hamwood, D. Alonso-Caneiro, S. A. Read, S. J. Vincent, M. J. Collins. *Effect of patch size and network architecture on a convolutional neural network approach for automatic segmentation of OCT retinal layers*. Biomed. Opt. Express 9 (2018)
- [12] H. Hanksara. *UNet - Introducing Symmetry in Segmentation*. Medium (2019).
Poveznica: <https://towardsdatascience.com/u-net-b229b32b4a71>
- [13] Y. Guo, A. S. Ashour. *Neutrosophic sets in dermoscopic medical image segmentation*. Academic Press (2019)
- [14] *PyTorch* - open source machine learning framework. Poveznica: <https://pytorch.org/>
- [15] M. Čupić. *Umjetna inteligencija - Uvod u strojno učenje*. Prvo izdanje (2020).
- [16] M. Jelčić. *Senilna makularna degeneracija ili degeneracija žute pjege*. Optical Express. Poveznica: <https://www.dioptrija.hr/bolesti-oka/senilna-makularna-degeneracija/>
- [17] J. Jordan. *An overview of semantic image segmentation* (2018). Poveznica: <https://www.jeremyjordan.me/semantic-segmentation/>
- [18] H. Noh, S. Hong, B. Han. *Learning Deconvolution Network for Semantic Segmentation*. Open Access, Computer Vision Foundation (2015)
- [19] J. Long, E. Shelhamer, T. Darrell. *Fully convolutional networks for semantic segmentation*. Proceedings of Computer Vision and Pattern Recognition (2015)
- [20] *Comet* - machine learning platform. Poveznica: <https://www.comet.ml/site/>

Sažetak

Duboko učenje ReLayNet arhitekturom za segmentaciju slojeva retine

U ovom je radu opisana state-of-the-art metoda za segmentaciju slojeva retine i motivacija iza nje. Kroz prilagodbu metode domaćoj bazi podataka analizirane su performanse modela primijenjenog na bazu podataka pacijenata koji boluju od senilne makularne degeneracije. U sklopu rada provedeno je nekoliko eksperimenata te analiziran značaj pojedinih komponenti funkcije pogreške i težina pridijeljenih slojevima i granicama između slojeva retine.

Ključne riječi: OCT, neuronske mreže, duboko učenje, konvolucijske neuronske mreže, ReLayNet, višeklasna segmentacija, računalni vid, senilna makularna degeneracija

Summary

Deep Learning Using ReLayNet for Retinal Layer Segmentation

In this paper, the motivation behind the problem of retinal layer segmentation is explained and a state-of-the-art method for this problem is described. Through adjustment of the method to our dataset, the performance of the model was analysed when applied to a dataset of B-scans of patients suffering from age-related macular degeneration. A number of experiments were conducted as part of this paper and the importance of different loss function components and weights given to retinal layers and layer borders was analysed.

Keywords: OCT, neural networks, deep learning, convolutional neural networks, ReLayNet, multiclass segmentation, computer vision, age-related macular degeneration

Skraćenice

OCT	<i>optical coherence tomography</i>	optička koherentna tomografija
AMD	<i>age-related macular degeneration</i>	senilna makularna degeneracija
SD	<i>spectral domain</i>	spektralna domena
DR	<i>diabetic retinopathy</i>	dijabetička retinopatija
CNN	<i>convolutional neural network</i>	konvolucijska neuronska mreža
ILM	<i>inner limiting membrane</i>	unutrašnja granična membrana
IPL	<i>inner plexiform layer</i>	unutarnji mrežasti sloj
INL	<i>inner nuclear layer</i>	unutarnji zrnati sloj
PED	<i>pigment epithelial detachment</i>	odvojenje pigmentnog epitela retine
RPE	<i>retinal pigment epithelium</i>	pigmentni epitel mrežnice
IRF	<i>intraretinal fluid</i>	intraretinalna tekućina
SRF	<i>subretinal fluid</i>	subretinalna tekućina

Privítak

Definícija mreže

ReLayNet

```
((encode1): EncoderBlock(
  (conv): Conv2d(1, 64, kernel_size=(7,3), stride=(1,1), padding=(3,1))
  (batchnorm): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True)
  (prelu): PReLU(num_parameters=1)
  (maxpool): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1)
)
)
encode2): EncoderBlock(
  (conv): Conv2d(64,64, kernel_size=(7,3), stride=(1,1), padding=(3,1))
  (batchnorm): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True)
  (prelu): PReLU(num_parameters=1)
  (maxpool): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1)
)
)
encode3): EncoderBlock(
  (conv): Conv2d(64,64, kernel_size=(7,3), stride=(1,1), padding=(3,1))
  (batchnorm): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True)
  (prelu): PReLU(num_parameters=1)
  (maxpool): MaxPool2d(kernel_size=2, stride=2, padding=0, dilation=1)
)
)
```

```

(bottleneck): BasicBlock(
  (conv): Conv2d(64,64, kernel_size=(7,3), stride=(1,1), padding=(3,1))
  (batchnorm): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True)
  (prelu): PReLU(num_parameters=1)
)
(decode1): DecoderBlock(
  (conv): Conv2d(128,64, kernel_size=(7,3), stride=(1,1), padding=(3,1))
  (batchnorm): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True)
  (prelu): PReLU(num_parameters=1)
  (unpool): MaxUnpool2d(kernel_size=(2,2), stride=(2,2), padding=(0,0))
)
(decode2): DecoderBlock(
  (conv): Conv2d(128,64, kernel_size=(7,3), stride=(1,1), padding=(3,1))
  (batchnorm): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True)
  (prelu): PReLU(num_parameters=1)
  (unpool): MaxUnpool2d(kernel_size=(2,2), stride=(2,2), padding=(0,0))
)
(decode3): DecoderBlock(
  (conv): Conv2d(128,64, kernel_size=(7,3), stride=(1,1), padding=(3,1))
  (batchnorm): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True)
  (prelu): PReLU(num_parameters=1)
  (unpool): MaxUnpool2d(kernel_size=(2,2), stride=(2,2), padding=(0,0))
)
(classifier): ClassifierBlock(
  (conv): Conv2d(64, 4, kernel_size=(1, 1), stride=(1, 1))
  (softmax): Softmax2d()
))

```