

Segmentacija stanica na mikroskopskim slikama zasnovana na modelima dubokog učenja

Ćošković, Hrvoje

Master's thesis / Diplomski rad

2021

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Science / Sveučilište u Zagrebu, Prirodoslovno-matematički fakultet**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:217:645201>

Rights / Prava: [In copyright](#)/Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2025-03-29**



Repository / Repozitorij:

[Repository of the Faculty of Science - University of Zagreb](#)



Segmentacija stanica na mikroskopskim slikama zasnovana na modelima dubokog učenja

Ćošković, Hrvoje

Master's thesis / Diplomski rad

2021

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Science / Sveučilište u Zagrebu, Prirodoslovno-matematički fakultet**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:217:645201>

Rights / Prava: [In copyright](#)/Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-06-20**



Repository / Repozitorij:

[Repository of the Faculty of Science - University of Zagreb](#)



SVEUČILIŠTE U ZAGREBU
PRIRODOSLOVNO–MATEMATIČKI FAKULTET
MATEMATIČKI ODSJEK

Hrvoje Čošković

**SEGMENTACIJA STANICA NA
MIKROSKOPSKIM SLIKAMA
ZASNOVANA NA MODELIMA
DUBOKOG UČENJA**

Diplomski rad

Voditelj rada:
dr. sc. Tomislav Šmuc

Zagreb, studeni, 2020.

Ovaj diplomski rad obranjen je dana _____ pred ispitnim povjerenstvom u sastavu:

1. _____, predsjednik
2. _____, član
3. _____, član

Povjerenstvo je rad ocijenilo ocjenom _____.

Potpisi članova povjerenstva:

1. _____
2. _____
3. _____

Posvećujem ovaj rad mojim roditeljima, mojoj braći, mojoj sestri i mojoj djevojci. Hvala Vam na bezuvjetnoj podršci.

Zahvaljujem se dr. sc. Tomislavu Šmucu i dr. sc. Tomislavu Lipiću na prenesenim znanjima i vodstvu u izradi ovoga diplomskog rada.

Sadržaj

Sadržaj	iv
Uvod	6
1 Segmentacija stanice	7
1.1 Opis problema i motivacija	7
2 Duboko učenje	11
2.1 Modeliranje problema	13
2.2 Učenje neuronske mreže	18
2.3 Modeli dubokog učenja	28
3 Segmentacija jezgre stanice na mikroskopskim slikama: 2018 Data Science Bowl	41
3.1 Rezultati	43
3.2 Skup podataka	47
3.3 Referentni model	50
3.4 Najučinkovitija rješenja	52
3.5 TTA	59
3.6 Zaključak	61
4 Modeliranje vlastitog modela	65
4.1 Uvod	65
4.2 Skup podataka	65
4.3 Modeli i učenje	67
4.4 Rezultati	70
Bibliografija	75

Uvod

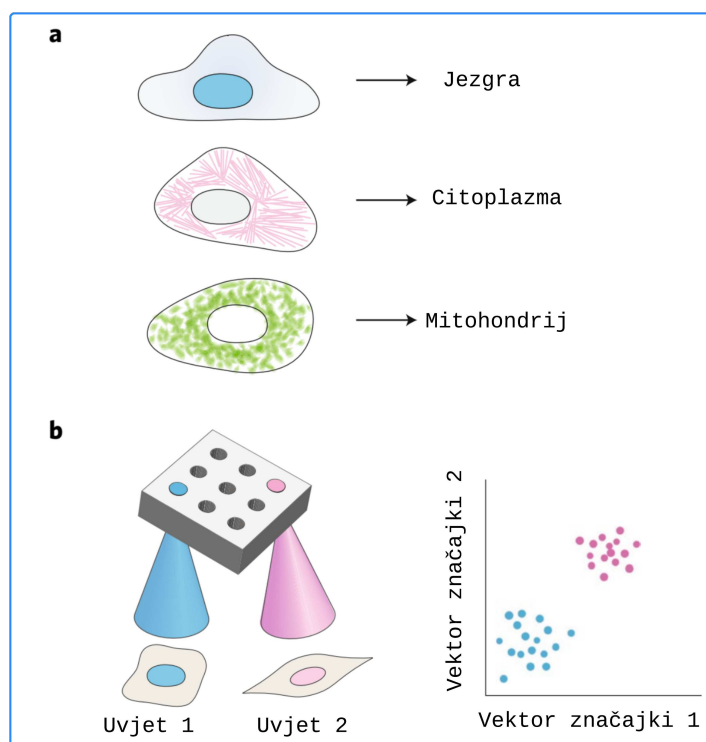
Napredci u području računalnog vida i strojnog učenja bili su preduvjet za razvoj velikog broja algoritama koji imaju sposobnost raspoznavanja sadržaja slike. Ovi algoritmi potpomognuti metodama dubokog učenja, našli su svoju primjenu i u računalnoj biologiji i medicini. Pomoću njih, analiza i interpretacija slika je znatno brža, što znanstvenicima ostavlja više vremena za istraživanje. Takvi algoritmi predstavljaju potencijalnu jedinstvenu metodologiju za automatizirano, učinkovito i precizno rješavanje zadatka kvantitativne analize slike. Postupci augmentacije podatkovnog skupa za učenje, prijenos znanja prethodno naučenih modela i njihova prilagodba na drugu domenu pokazali su se kao uspješan pristup primjene modela dubokih neuronskih mreža u problemima gdje je otežano prikupljanje velikog broja precizno označenih primjera potrebnih za uspješno učenje.

Razvoj modernih fotografskih tehnika zaslužan je za transformaciju biologije i medicine. Taj razvoj je omogućio znanstvenicima pristup vremenskim i prostornim značajkama živih bića. Napredak na području optike, iznjedrio je mikroskope koji imaju mogućnost fotografirati prostorni raspon veličine stanice, pa sve do čitavog organizama. Istovremeno, ispitivanja pomoću fluorescentnih sondi proteina poboljšavaju svjetlinu, stabilnost slike i spektralni raspon boja u molekulama. Zajedno, ova dva faktora omogućavaju široku domenu istraživanja živih stanica, od dugoročnog praćenja samo jedne molekule, simultanih mjerenja pomoću više biosenzora, do promatranja i praćenja razvoja cijelog organizma. Istovremeno s navedenim otkrićima i napredcima, dolazi do povećane potrebe za analizom slike. Tipični zadaci uključuju:

- klasifikaciju slike — cilj ovog zadatka je pridružiti značajnu oznaku slici. Klasični primjer je određivanje prikazuje li slika psa ili mačku. Primjer iz biologije bi mogao biti odluka je li se matična stanica sa slike promijenila ili nije. Mnoga današnja istraživanja su usredotočena na poboljšanje rezultate klasifikacije na javno dostupnim skupovima podataka poput ImageNet-a¹, jer su modeli koji se koriste za klasifikaciju u biologiji slični, ako ne i isti kao u komercijalnim aplikacijama. Zbog nedostatka podataka i otežanog i skupog prikupljanja istih, modeli za segmentaciju bioloških slika se treniraju na takvim standardnim javno dostupnim skupovima slika,

¹<http://www.image-net.org/>

te se zatim u završnoj fazi učenja (završnom sloju neuronske mreže) prilagođavaju za rad na konkretnom zadatku tako što se proces učenja ponovno pokreće na željenim označenim biološkim slikama. Klasifikatori slike se često koriste u određivanju morfoloških promjena u stanici. Pomoću fluorescentnih markera određuju je li se dogodila promjena u morfologiji stanice ili nije, te zatim treniraju modele na takvim slikama koji daju izravan odgovor na to pitanje, zaobilazeći markiranje [7]. Duboko učenje se također primjenjuje u određivanju prostornih uzoraka, poput otkrivanja proteina u kvascu ili čak ljudskom tkivu. Suvremena istraživanja [11] su iznjedrila uređaje za raspoznavanje stanica, koji pomoću modela dubokog učenja za klasifikacije slike u realnom vremenu sortiraju niz slika stanica.

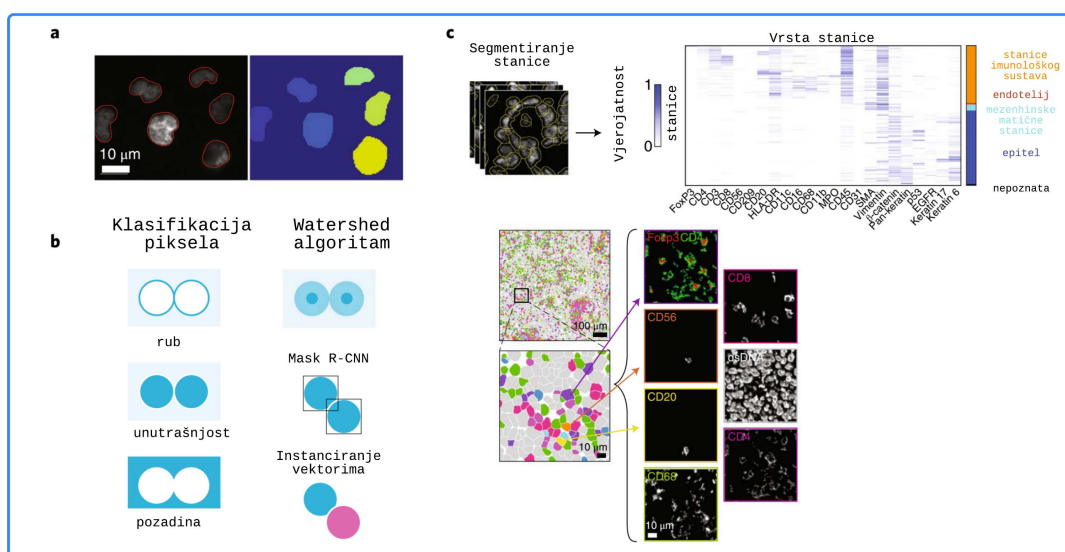


Slika 0.1: Izvor: **Klasifikacija slike u biologiji.**

a: Klasifikator pomoću dubokog učenja nad podacima dobivenim fluorescentnim markiranjem točno raspoznaje dijelove stanice.

b: Suvremeni uređaji pogonjeni algoritmima dubokog učenja mogu uspješno razaznati promjene u morfologiji stanice. Ovi modeli se modeliraju kao klasifikatori promjena, zatim iz podataka na kojima su ustanovili promjene ekstrahiraju vektore značajki, koji se zatim klasteriraju i koriste kako bi se utvrdili fenotipi novih stanica. Izvor: [6].

- segmentaciju slike — segmentacija slike je određivanje dijelova ili objekata sa slike. Tipičan biološki primjer je određivanje pojedinih stanica na mikroskopskoj slici tkiva. Važno je napomenuti da postoji dva podzadatka segmentacije slike: semantička segmentacija i segmentacija instanci. Semantička segmentacija svim pikselima koji prikazuju isti objekt pridružuje istu oznaku, dok segmentacija instanci razlikuje instance iste klase. Segmentacije jedne stanice promatramo kao zadatak segmentacije instanci. Tipičan primjer razlike ova dva podzadatka je taj, da semantička segmentacija nebi razdvojila dvije susjedne jezgre stanice, dok segmentacija instanci bi.



Slika 0.2: Segmentacija slike u biologiji.

a: Segmentacija instanci identificira svaku instancu klase, poput jezgra stanica.

b: Koraci segmentiranja instanci. Klasifikacija na razini piksela pokušava točno razaznati granice objekata, dok metoda slijeva (*watershed*) razaznaje preklapanja, računa granice objekata te označava piksele različitih instanci različitim oznakama.

c: Primjena segmentacije dubokim učenjem na problemu raka dojke u istraživanju [10]. Segmentacijski filteri su korišteni kako bi izmjerili jačinu signala za svaku stanicu pomoću kojih je određen tip stanice. Nakon toga te su stanice klasterirane, iz njih su ekstrahirani vektori značajki te su pomoću njih određivani tipovi novih stanica i njihova stanja. Znanstvenici su pomoću tih saznanja mjerili rast tumora u mikro-okruženju. Izvor: [6].

Znanstvenici u novijim istraživanjima [?] koriste modele poput U-Net-a² i DeepCell-

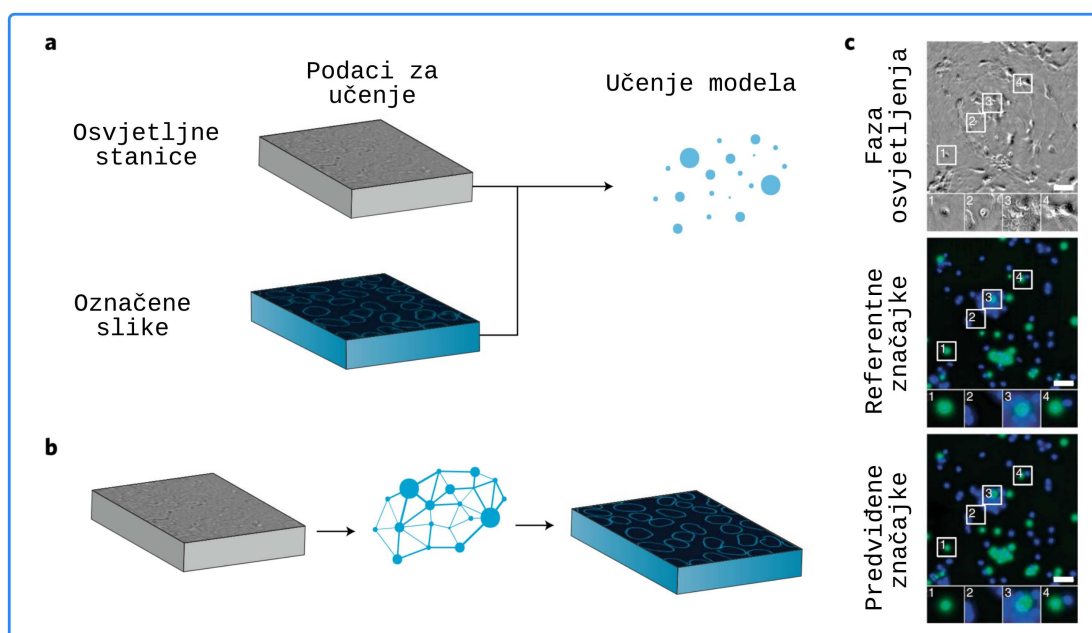
²<https://arxiv.org/abs/1505.04597>

a³ te tretiraju segmentaciju stanice kao problem semantičke segmentacije, te za svaki piksel pretpostavljaju vjerojatnost pripadaju li unutrašnjosti stanice, rubu ili pozadini. Zatim u zadnjem sloju prelaze na filtriranje pomoću metode slijeva iz scikit-image⁴ biblioteke, koji razdvaja instance, te tako dobivamo završnu masku početne slike stanice. Ovakav pristup predstavlja sintezu semantičke i segmentacije instanci, te također pokazuje obećavajuće rezultate. Potencijal segmentacije slike je ogroman, te ćemo se u ovome radu koncentrirati na segmentaciju stanice pomoću ansambla od 32 U-Net modela pomoću kojega su pobjednici Data Science Bowl 2018 [9]. uspješno segmentirali stanice na čak 15 različit skupova podataka bez rekonfiguracije rješenja. Također bitno je napomenuti kako se modeli dubokog učenja koriste u patološkim istraživanjima interakcije stanica tumora i imunoloških stanica.

- praćenje objekta sa slike — zadatak praćenja objekta u video sekvenci. Biološka primjena bi mogla biti praćenje razvoja embrija u video isječku. Analiza jedne stanice zahtjeva praćenje i prostorno određivanje pozicije stanice, kako bi se mogle povezati u vremenu, ili čak predvidjeti daljnji tijek. Iako iznimno zahtjevan (objekti se mogu spojiti i ili razdvojiti i nestati, može ih biti jako puno po slici i slično), ovaj zadatak je ključan u razumijevanju kretanja i razvoja stanica, što je važno kod bakterijski infekcija i napredovanja tumora.
- augmentacije mikroskopskih podataka — zadatak ekstrakcije latentnih informacija iz bioloških slika, poput određivanje pozicije jezgre stanice metodama osvjetljavanja stanica. Iako takve metode mogu generirati informacije vezane za biološku strukturu, poput informacija o jezgri i citoplazmi, iznimno je računalno teško dobiti te informacije. Duboko učenje se još jednom pokazalo kao rješenje problema. Znanstvenici su koristili fluoroscentne slike stanice kao referencu za nadzirano učenje. Kako bi kreirali 3D realne morfološke modele membrane i jezgre stanice, znanstvenici na Allen institutu u Seattleu su koristili generativni model dubokog učenja. Njihov model se sastojao od dva jedinstvena podmodela: jednog koji je učio varijacije u jezgri i obliku stanice, te drugog koji je uočavao relacije među podstaničnim dijelovima.

³<https://www.biorxiv.org/content/10.1101/505032v1>

⁴scikit-image.org



Slika 0.3: **Augmentacija mikroskopskih podataka.**

a: Duboko učenje pomoću osvjetljavanja različitim svjetlosnim valovima otkriva skrivene podatke u biološkim slikama, te pomoću tih saznanja možemo poboljšati podatke.

b,c: Model određuje koji su neuroni mrtvi ili živi direktno iz mikroskopske slike. Izvor: [6].

Sukladno ovim potrebama, znanstvenici i tvrtke su razvili različite programske alate i pakete. MATLAB⁵ je jedan od prvih programskih alata koji je pružio podršku i algoritme za računalni vid, dok se nedavno Python popularizirao razvojem javno dostupnih modula i paketa poput: NumPy⁶, SciPy⁷, Pandas⁸, scikit-image⁴, scikit-learn⁹, matplotlib¹⁰, Jupyter¹¹, itd. Oba programska jezika nude spremne implementirane biblioteke za računalni vid koje znanstvenici mogu iskoristiti kako bi napisali programska rješenja za određene probleme. Kako je analiza slike postala sve učestalija, razvijeni su i alati koji nude grafička sučelja kako bi bili pristupačniji.

⁵www.mathworks.com/products/matlab.html

⁶www.numpy.org

⁷www.scipy.org

⁸pandas.pydata.org

⁹scikit-learn.org/stable

¹⁰matplotlib.org

¹¹jupyter.org

Cilj ovog rada je proučiti pristupe koji su zasnovani na modelima dubokog učenja za rješavanje problema segmentacije stanica na mikroskopskim slikama, oslanjajući se na najuspješnije modele razvijene na problemu natjecanja Data Science Bowl 2018., koje generalno rade za različite tipove slika i eksperimentalne postavke, kao i najuspješnije specifične modele za zadatak CAMELYON17¹² i ostale povezane zadatke na histološkim slikama u sklopu izazova Grand Challenges in Biomedical Image Analysis¹³. Za izabrane modele evoluirat će se mogućnosti njihovog izravnog prijenosa znanja i domenske prilagodbe na zadatke precizne segmentacije stanica neurona na digitalnim histopatološkim slikama citoarhitektonike ljudske moždane kore kao i segmentacija tumorskih stanica na histološkim slikama tumora dojke.

¹²camelyon17.grand-challenge.org

¹³grand-challenge.org

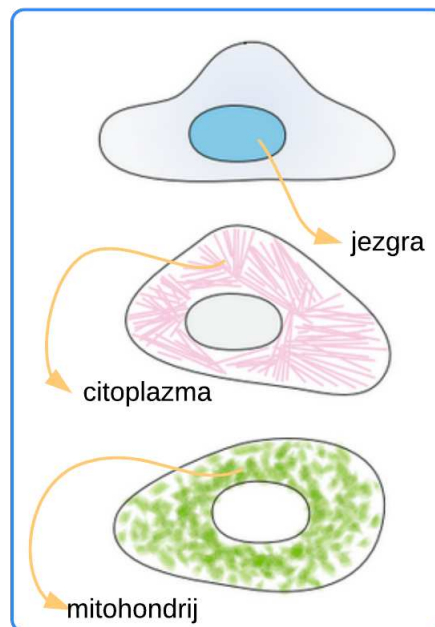
Poglavlje 1

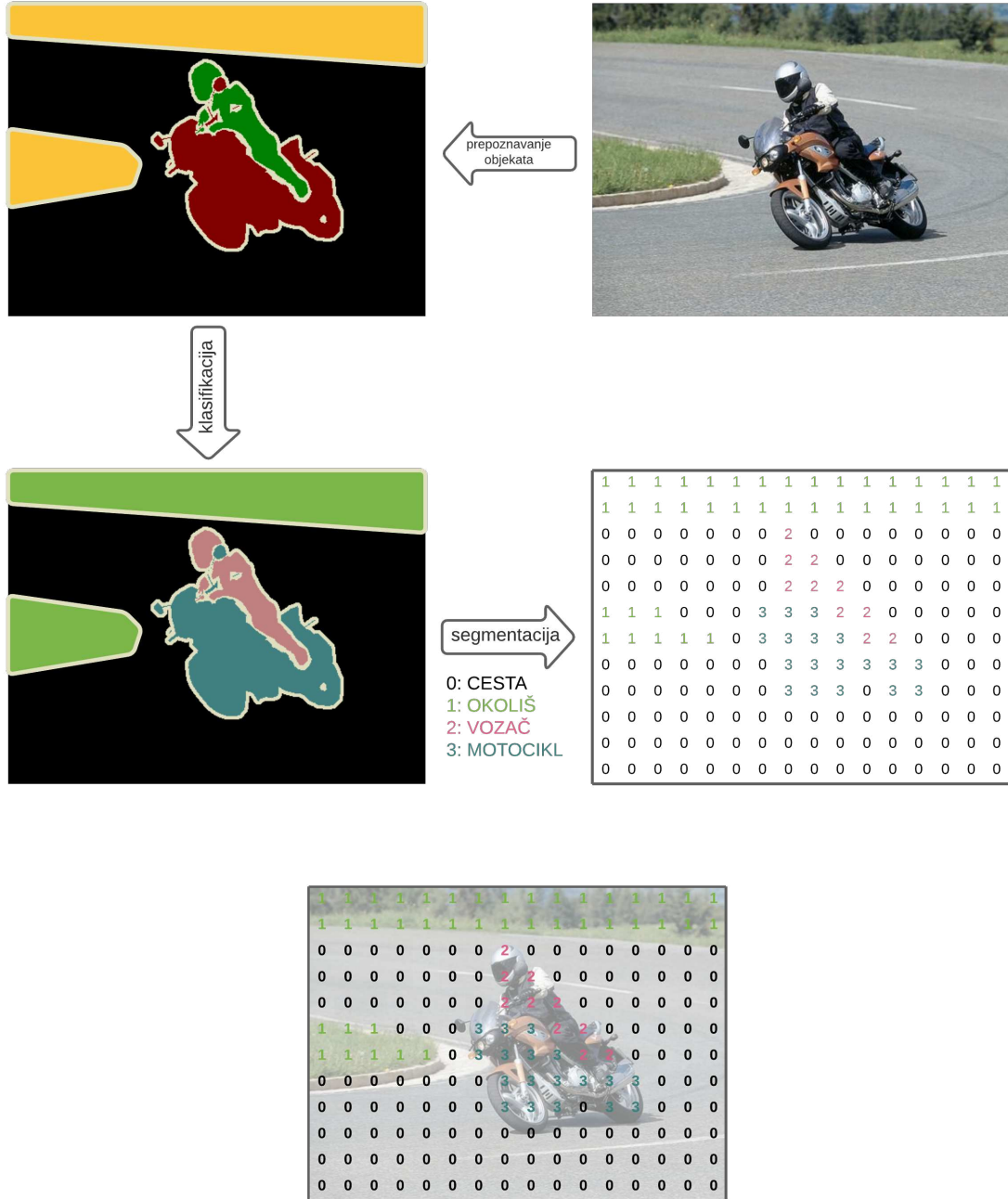
Segmentacija stanice

1.1 Opis problema i motivacija

Segmentacija slike stanice predstavlja specijalni slučaj segmentacije instanci slike, problema računalnog vida u kojemu je potrebno označiti specifične regije slike, tj. dodijeliti im oznake (eng. *label*) koje će indicirati pripadnost klasama, a istovremeno razdvojiti različite instance iste klase. Ako se spustimo na razinu piksela, potrebno je svakom pikselu pridružiti oznaku tako da pikseli koji prikazuju isti objekt imaju iste oznake (Slika 1.1). Krajnji cilj segmentacije je matrica koja svaki piksel preslikava u odgovarajuću cjelobrojnu vrijednost koja pak reprezentira određeni objekt sa slike.

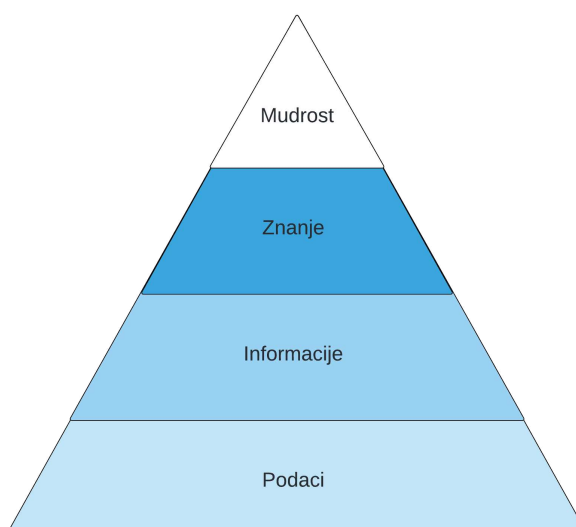
točnost od 0pt preciznost





Slika 1.1: Matrica sa označenim pikselima je $M \in \mathbb{N}^{12 \times 16}$, odnosno prikazana je u manjoj rezoluciji radi jednostavnosti. U stvarnom svijetu, ona bi trebala biti $M \in \mathbb{N}^{\text{visina} \times \text{širina}}$, tj. njene dimenzije bi trebale odgovarati stvarnoj rezoluciji ulazne slike, jer za svaki piksel moramo označiti kojoj klasi pripada. Izvor ulazne slike: [1].

Promatranje igra veliku ulogu u medicini i biološkim znanostima, stoga ogromnu važnost ima slika jer zarobljava trenutak, te ga iz nepovratno izgubljenoga čuva u beskonačnosti vremena. Segmentacija medicinskih slika oduvijek je bila jedan od zahtjevnijih problema. Odrediti ozlijeđeno tkivo ili razaznati organe na CT¹ ili MRI² slikama, ljudskome oku nije nipošto trivijalan zadatak. Uspješno rješavanje navedenog problema daje ključne informacije o obliku i volumenu tkiva ili organa. Prijašnja automatizirana rješenja su se oslanjala na tradicionalne matematičke metode i filtere za detekciju rubova. Strojno učenje predstavlja potencijalno rješenje koje bi moglo revolucionirati shvaćanje unutrašnjih procesa živih bića. Pojavom dubokog učenja, ekstrakcija željenih značajki baca tradicionalne metode u drugi plan. Ubrzanim razvojem računalnog sklopovlja početkom novog tisućljeća, složenost suvremenih pristupa dubokog učenja prestaje biti ograničenje, te se otvara put ka razvoju rješenja za automatiziranu segmentaciju slike u računalnoj medicini i biologiji, koja će iz njih ekstrahirati željene značajke i ubrzati analizu istih. Ubrzo, metode dubokog učenja demonstriraju svoje sposobnosti u procesiranju slike te zbog obećavajućih pristupa postaju primarna opcija u segmentaciji slike, posebno u medicini. Prethodnih nekoliko godina, segmentacija slike koja se temelji na tehnikama dubokog učenja privukla je veliku pozornost. U sljedećem poglavlju ćemo opisati na što se zapravo odnosi pojam dubokog učenja, te proći kroz arhitekture neuronskih mreža koje se primjenjuju medicini.



Također, trenutno se odvija nova zlatna groznica — groznica rudarenja podataka. Mnogi znanstvenici i istraživači žele suvremene metode dubokog učenja primijeniti na svoje po-

¹kompjuterska tomografija, eng. *Computed Tomography*

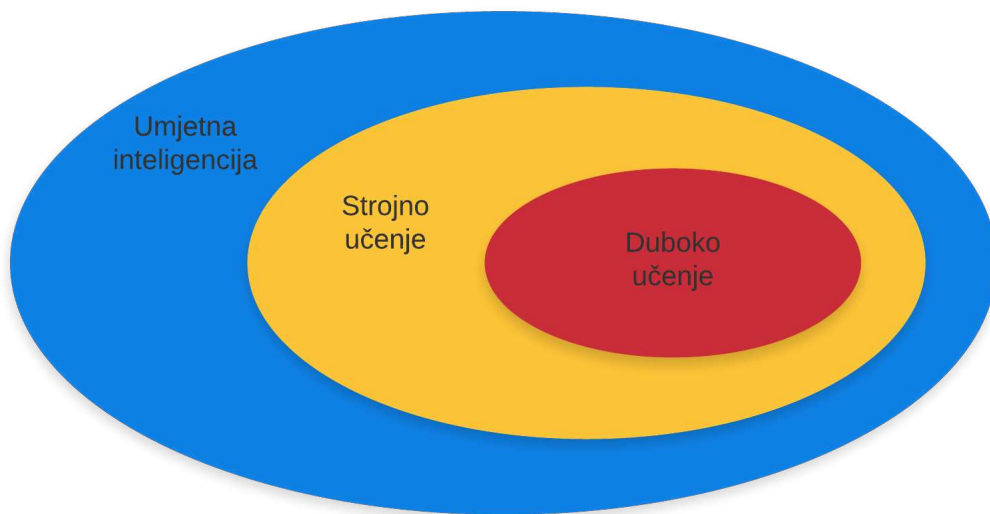
²magnetska rezonanca, eng. *Magnetic Resonance Imaging*

datke kako bi iz njih izvukli skrivena znanja te došli do novih spoznaja. Piramidalna shema iznad pokazuje odnos podatka, informacije, znanja i mudrosti. Neobrađeni i neorganizirani podaci bez konteksta nemaju nikakvu vrijednost ni značenje. Interpretiranjem podataka i dodavanjem konteksta dobivamo informaciju. Znanje predstavlja kombinaciju podataka, ekspertnog mišljenja, vještina i iskustva, dok mudrost možemo opisati kao razboritu primjenu znanja [12]. Duboko učenje obuhvaća proces transformacije podataka u korisne informacije i znanje. Ipak, još uvijek nije općeprihvaćeno u medicini. Revolucija u zdravstvu će nastupiti kada se dubokog učenje budu rutinski primjenjivalo u svakodnevnim zadaćama [3].

Poglavlje 2

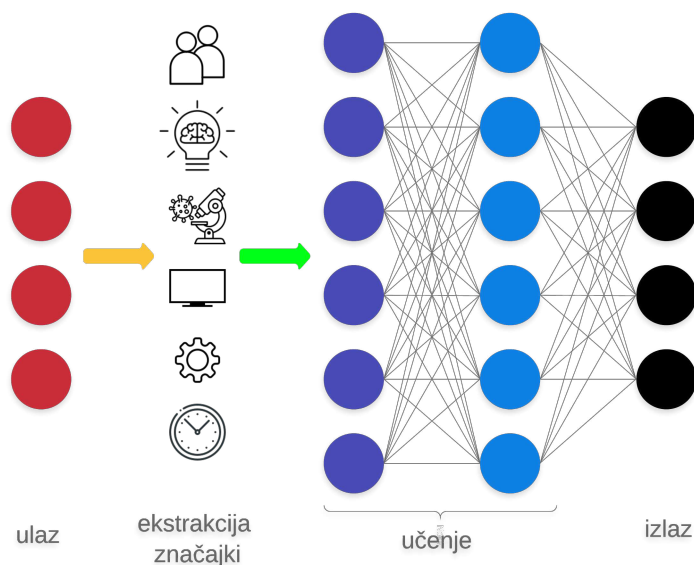
Duboko učenje

Duboko učenje je proširilo domenu rješivih problema računalnog vida. Naravno, kada govorimo o dubokom učenju, govorimo o grani strojnog učenja, neuronskim mrežama koje su u mogućnosti naučiti razaznati različite reprezentacije podataka (čak i one čovjeku izrazito apstraktne) te aproksimirati vrlo kompleksne funkcije.

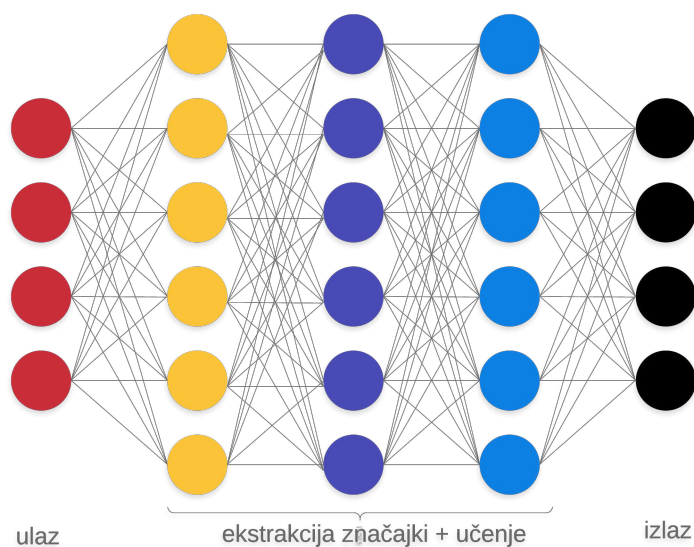


Slika 2.1: Odnos umjetne inteligencije, strojnog učenja i dubokog učenja.

Primijetimo razliku između tradicionalnog strojnog učenja, gdje se reprezentacije podataka ručno određuju kroz proces inženjeringa značajki (eng. *feature engineering*), dok suvremeni pristup kroz nadzirano ili nenadzirano duboko učenje omogućava klasifikatoru



Slika 2.2: Tradicionalni pristup u kojemu model uči na unaprijed određenim značajkama



Slika 2.3: Suvremeni pristup u kojemu model sam odabire koje su mu značajke važne i pomoću njih optimizira svoju funkciju troška

da sam nauči koje su značajke najvažnije. Taj proces nazivamo *ekstrakcija značajki*. Modeli nadziranog učenja koji rade na unaprijed označenom skupu podataka, za cilj imaju

naučiti funkciju koja najbolje aproksimira relaciju ulaznih i izlaznih podataka te postižu zavidne rezultate. Nenadzirani pristup učenja se koristi kako bi se rekonstruirali skrivene strukture i uzorci u podacima nakon preslikavanja u nižedimenzionalne prostore bez eksplicitno navedenih značajki.

Iako su ove metode postojane u matematičkom svijetu već desetljećima, veću pažnju uspijevaju privući 2012. kada je algoritam baziran na metodama dubokog učenja pobijedio na natjecanju prepoznavanja i klasifikacije velikog broja slika (otprilike 10M), punog imena: 2012 ImageNet Large Scale Visual Recognition Challenge¹. Otada, dolazi do povećanja domene problema koji se rješavaju metodama dubokog učenja. Dodatno, razvojem kompleksnijeg računalnog sklopovlja i programskih okvira (eng. *frameworks*), duboko učenje postaje pristupačnije programskim inženjerima. Dok se duboko učenje naširoko primjenjuje u komercijalne svrhe, pojavljuje se znatno veća potreba za primjenama u fizici, kemiji, medicini i biološkim znanostima, u primjenama nad slikama, ali i drugim tipovima podataka.

Ovo poglavlje će se sastojati od četiri glavna djela:

- modeliranja problema — opisuje i obuhvaća pristupe, odnosno arhitekturu neuronske mreže
- tehnika učenja — procesa učenja mreže kojim dolazi do željenih rezultata
- modela — koji su visoko zastupljene u segmentiranju slike, njihove prednosti i mane
- izazova i njihovih suvremenim rješenjima — modeliranje podataka, mreža i učenje

Prije nego krenemo na arhitekture neuronskih mreža, najprije ćemo pokušati objasniti što je neuronska mreža, tj. što ona predstavlja.

2.1 Modeliranje problema

Neuron

Osnovna gradivna jedinica svake neuronske mreže je **neuron**. Neuron ćemo matematički modelirati funkcijom te ćemo pokušati slijedno uvoditi funkcije i dodatne vektore kako ih budemo trebali.

Ulazni vektor $\vec{x} \in \mathbb{R}^n$, činit će prvi sloj neuronske mreže. Svaki neuron iz prethodnog sloja povezat ćemo sa neuronima iz sljedećeg, te im pridružiti neku proizvoljnu vrijednost jačine veze, težinu $w_i \in [-1, 1]$. Pozitivne i negativne vrijednosti možemo interpretirati kao smjer vektora. Sada imamo dva vektora, $\vec{x}, \vec{w} \in \mathbb{R}^n$, te ćemo u neurone spremati njihov

¹<http://image-net.org/challenges/LSVRC/2012/>

skalarni produkt kako bi indicirali njihovu povezanost. Skalarnim množenjem dobivamo skalar koji može biti bilo koja realna vrijednost. Za učenje neuronske mreže najpoželjnije su vrijednosti iz segmenta $[0, 1]$, jer će takva normalizirana vrijednost predstavljati aktivaciju neurona u mreži, te ako je ona iz određenog spektra, lakše ćemo moći uspoređivati jačine aktivacija neurona u mreži. Zbog prethodno navedene tvrdnje, uvodimo nelinearnu aktivacijsku funkciju, koja će nam dati dvije ključne stvari:

1. normalizirat će vrijednost skalarnog produkta \vec{x} i \vec{w}
2. omogućit će učenje kompleksnijih ciljnih funkcija od linearnih

Aktivacijske funkcije

Slijedi nekoliko aktivacijskih funkcija i njihove formule, nakon formalne definicije svih gradivnih dijelova mreže te metoda učenja, vratit ćemo se na navedene funkcije i reći nešto o svakoj:

1. Identiteta $\mathbb{1} : \mathbb{R} \rightarrow \mathbb{R}$

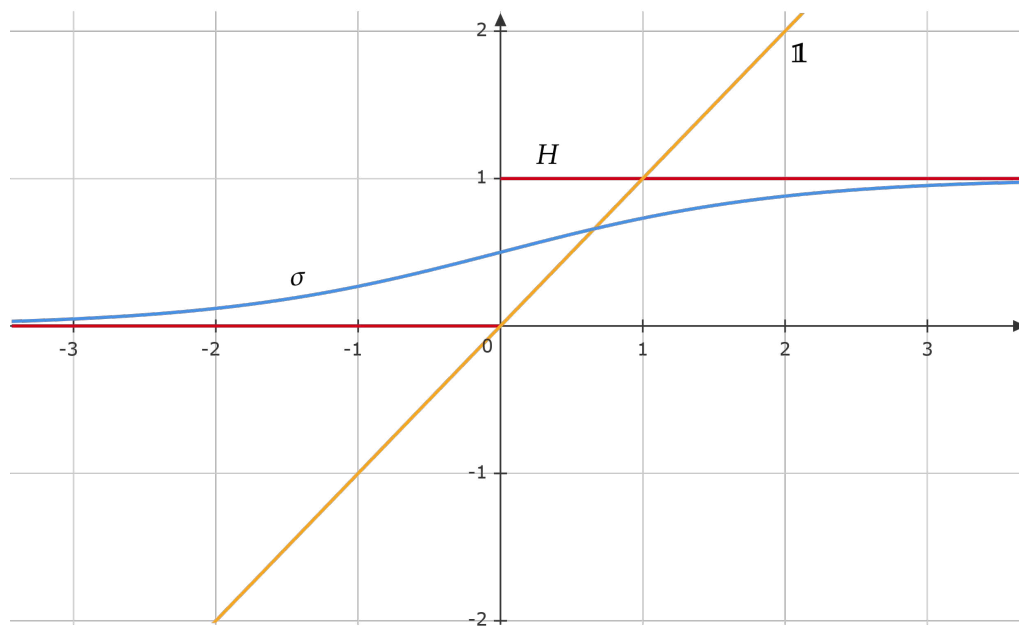
$$\mathbb{1}(z) = z \quad (2.1)$$

2. Funkcija praga ili *Heavisideova* funkcija skoka $H : \mathbb{R} \rightarrow \{0, 1\}$

$$H(z) = \begin{cases} 0, & z < 0 \\ 1, & z \geq 0 \end{cases} \quad (2.2)$$

3. Sigmoidna funkcija $\sigma : \mathbb{R} \rightarrow \langle 0, 1 \rangle$:

$$\sigma(z) = \frac{1}{1 + e^{-z}} \quad (2.3)$$



Slika 2.4: Graf identitete, funkcije skoka i Sigmoidne funkcije

4. Tangens hiperbolni $\text{th} : \mathbb{R} \rightarrow \langle -1, 1 \rangle$:

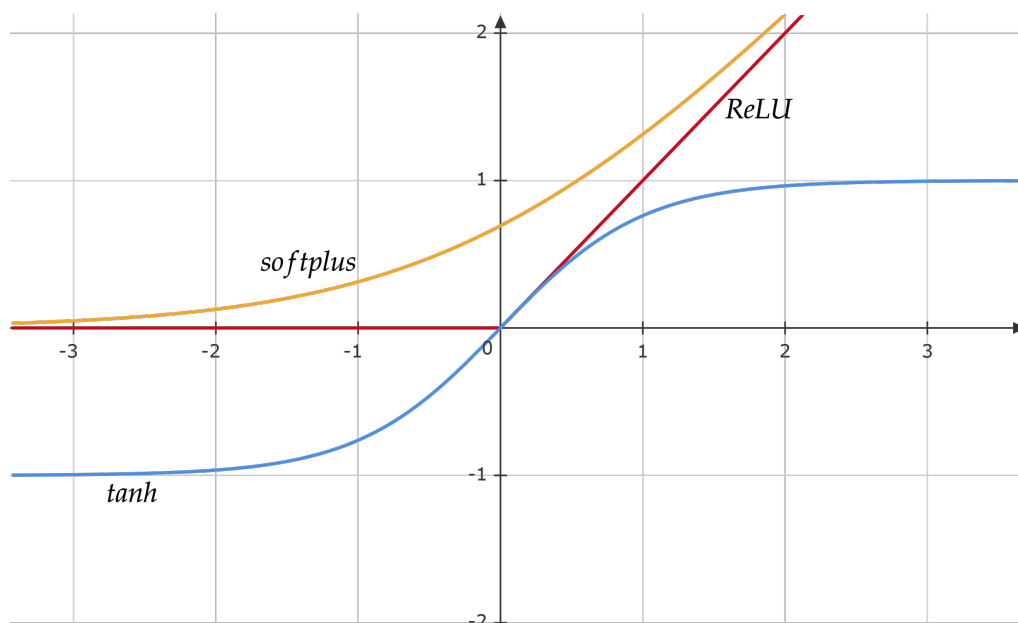
$$\text{th}(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}} \quad (2.4)$$

5. $\text{ReLU} : \mathbb{R} \rightarrow [0, \mathbb{R})$:

$$\text{ReLU}(z) = \max(0, z) \quad (2.5)$$

6. $\text{softplus} : \mathbb{R} \rightarrow \langle 0, \infty \rangle$:

$$\text{softplus}(z) = \ln(1 + e^z) \quad (2.6)$$



Slika 2.5: Graf tangensa hiperbolnog, ReLU i softplus funkcije

7. Softmax $\sigma : \mathbb{R}^K \rightarrow [0, 1]^K$:

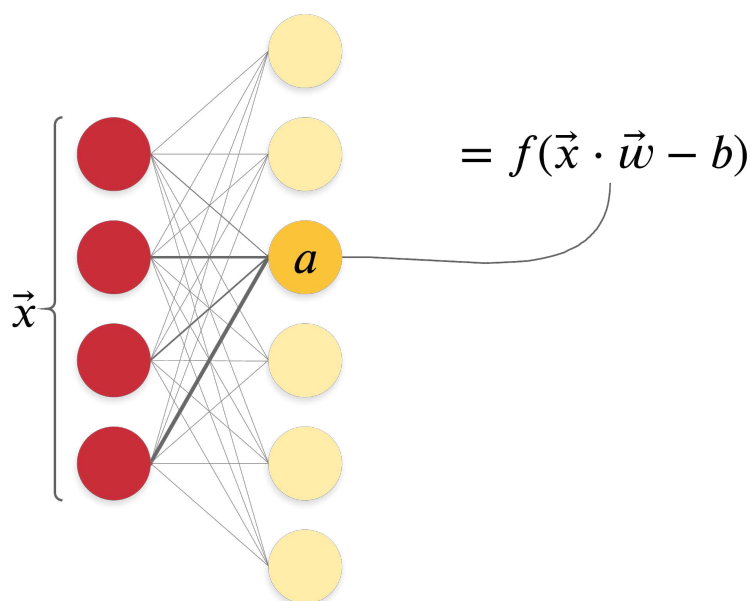
$$\sigma(\vec{z})_i = \frac{e^{z_i}}{\sum_{j=1}^K e^{z_j}} \quad (2.7)$$

Možda nam ne odgovara da nam se funkcija aktivira točno za vrijednost skalarnog produkta vektora \vec{x} i \vec{w} ? Možemo dodati prag b , odnosno oduzeti ga od skalarnog produkta te onda primijeniti aktivacijsko preslikavanje. Napokon imamo sve potrebno kako bi definirali neuron [14]:

Definicija 2.1.1. Neka je $n \in \mathbb{N}$, te $A \subseteq \mathbb{R}$. Funkciju $N^f : A^n \rightarrow A$ definiranu sa:

$$N^f(x_1, \dots, x_n) = f\left(\sum_{i=1}^n x_i w_i - b\right)$$

nazivamo **neuron**. Realne brojeve nazivamo w_1, w_2, \dots, w_n **težine**, realan broj b **prag**, a funkciju $f : \mathbb{R} \rightarrow A$ **aktivacijska funkcija**.



Slika 2.6: Grafički prikaz prva dva sloja s naglaskom na neuron. Oznake težina nisu prikazane, ali debljina brida indicira njihov iznos.

Sada ćemo još definirati i sloj neuronske mreže, te naposljetku i samu neuronsku mrežu [14]:

Definicija 2.1.2. Neka su $N_1^f, \dots, N_m^f : A^n \rightarrow A$ neuroni. Funkciju

$$S(\vec{x}) = (N_1^f(\vec{x}), \dots, N_m^f(\vec{x})), \quad \vec{x} \in A^n$$

nazivamo **sloj** neuronske mreže. Ako su neuroni $N_1^f, \dots, N_m^f : A^n \rightarrow A$ identitete, onda S nazivamo **ulazni sloj**.

Na slici (2.3) vidimo 5 slojeva: ulazni, 3 skrivena i izlazni sloj neuronske mreže.

Definicija 2.1.3. Neka je S_0 ulazni sloj, te S_1, \dots, S_L slojevi neurona s aktivacijskom funkcijom f tako da je

$$N = S_L \circ \dots \circ S_0$$

dobro definirana. Funkciju N zovemo **neuronska mreža**. Sloj S_L izlazni sloj, a uređenu $(L - 1)$ -torku (S_1, \dots, S_{L-1}) nazivamo **skriveni sloj**.

2.2 Učenje neuronske mreže

Sada kada smo definirali arhitekturu modela, trebamo reći nešto o procesu učenja modela, odnosno neuronske mreže. Tri su osnovna pristupa učenju: nadzirano, nenadzirano, polunadziranog te transfer znanja. Nadzirani pristup koristimo kada imamo označeni skup ulaznih podataka koji su nam dostupni za učenje. Problem zna predstavljati prikupljanje takvih podataka. Ono zna biti vremenski i financijski skupo. Na slici (2.2) vidimo da umjesto drugog sloja imamo znanstvenike ili eksperte koji troše svoje vrijeme na promatranje ulaznog skupa podataka kako bi odlučili koje su značajke najvažnije. Taj problem se rješava nenadziranim metodama učenja u kojima klasifikator pokušava naučiti skrivene uzorke među ulaznim podacima. Polunadzirano učenje se koristi kada imamo ulazni skup podataka koji nije sasvim označen. O učenju pomoću transfera znanja će biti više riječi kasnije. Prisjetimo se sada proizvoljnih težina, realnih pragova i aktivacijskih funkcija.

1. Grupiramo težine po slojevima. Neka su S_i i S_{i+1} slojevi neuronske mreže N sa L slojeva te neka w_l označava sve težine povezane sa l -tim slojem, b^l vektor pomaka u l -tom sloju. Sa:

$$\mathbb{W} = \{w^1, \dots, w^L\} \quad (2.8)$$

označimo sve težine u N .

2. Označimo sa \mathbb{B} skup svih vektora pragova:

$$\mathbb{B} = \{b^1, \dots, b^L\} \quad (2.9)$$

3. Posljednje, sa:

$$\mathbb{F} = \{f^1, \dots, f^L\} \quad (2.10)$$

označimo sve aktivacijske funkcije u slojevima.

4. Sve te slobodne parametre objedinimo u jedan skup:

$$\Theta = \{\mathbb{W}, \mathbb{B}, \mathbb{F}\} \quad (2.11)$$

Neka imamo skup od N ulaznih podataka oblika $\{(x^{(1)}, y^{(1)}), \dots, (x^{(N)}, y^{(N)})\}$, takvih da je $x^{(i)}$ vektor značajki i -tog ulaznog primjera iz vektorskog prostora X , a $y^{(i)}$ oznaka ili pripadna oznaka ulaznog vektora. Prema [4], svaki algoritam strojnog učenja možemo opisati pomoću tri stvari:

- **modela** H koji predstavlja skup **hipoteza**,

$$H = \{h(X | \Theta)\}_{\Theta} \quad (2.12)$$

pri čemu je hipoteza određena sa slobodnim parametrima. Učenje neuronske mreže se svodi na problem potrage "pravog" skupa Θ^* funkcije $\hat{y} = H|_{\Theta^*}$, tj. prave hipoteze koja najbolje aproksimira **funkciju cilja** $y = J(x)$, koja opisuje stvarnu relaciju ulaznih podataka i izlazih, te za koju pretpostavljamo da postoji. To je iterativni proces u kojemu mreža ažurira navedene parametre. Ali kako znati koja je hipoteza *prava*? Uvodimo sljedeću funkciju.

- **funkcije troška** C koja će imati ulogu metrike između naše **aproksimacije** \hat{y} i **funkcije cilja** J ,

$$J(h | X) = \frac{1}{N} \sum_{i=1}^N C(x^{(i)}, h(x^{(i)} | \Theta)). \quad (2.13)$$

Kako bi model bio što uspješniji, želimo funkciju cilja minimizirati tako što ćemo minimizirati funkciju troška po svim ulaznim vrijednostima. Funkcija troška specifična je domeni zadatka koji se želi riješiti.

- **Procesa optimizacije** kojima minimiziramo funkciju cilja, tj. dolazimo do željenih parametra Θ^* u skupu hipoteza H . Taj proces se još naziva **učenje**:

$$\min_h J(h | X) = \frac{1}{N} \sum_{i=1}^N C(x^{(i)}, h(x^{(i)} | \Theta^*)) = \min_{\Theta} \frac{1}{N} \sum_{i=1}^N C(x^{(i)}, h(x^{(i)} | \Theta)) \quad (2.14)$$

Proces učenja je bitan zbog toga što učenjem nad dostupnim (trening) podacima, model uči kako bi stekao svojstvo **generalizacije**, odnosno kako bi mogao rješavati zadatke i na "neviđenim" (novim) podacima.

Kod dubokog učenja i neuronskih mreža, dva su osnovna pristupa učenju: učenje **metodom gradijentnog spusta** i učenje **propagacijom greške unatrag**.

Metoda gradijentnog spusta

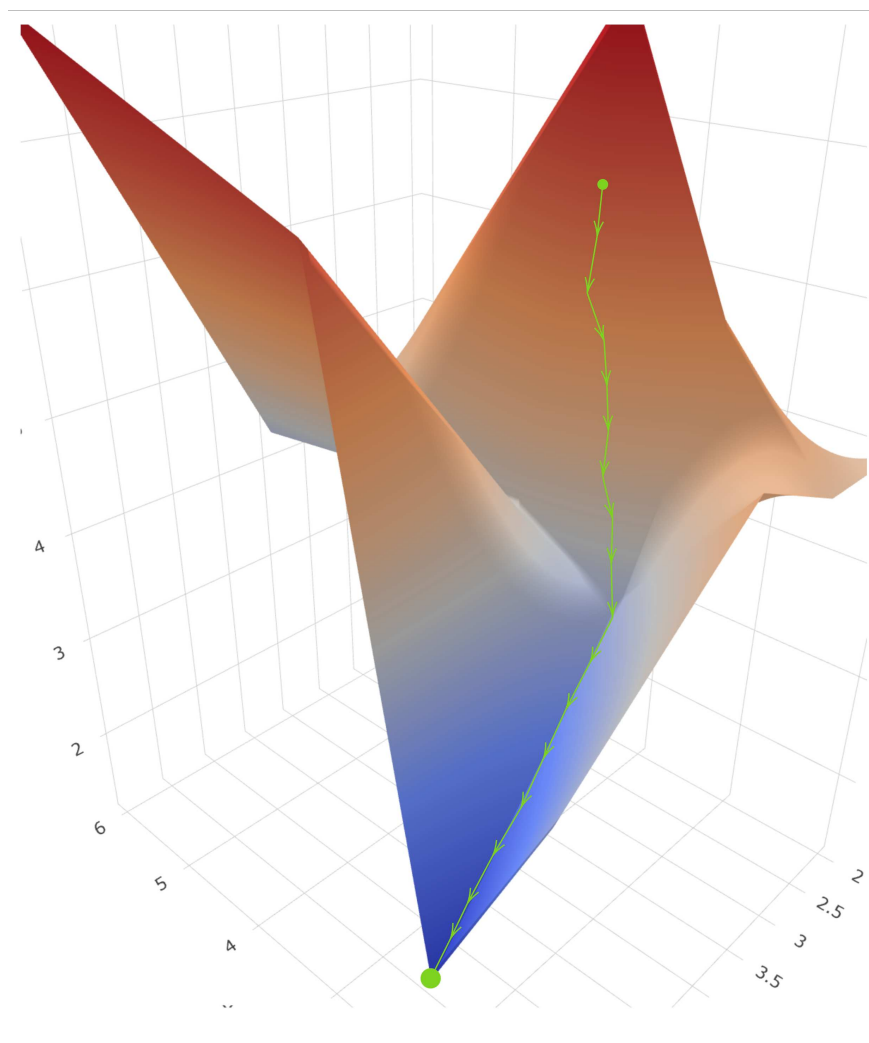
Kao što smo već naveli, strategija učenja je da se u svakoj iteraciji ažuriraju slobodni parametri (izbor aktivacijskih funkcija je unaprijed određen, pa ćemo pod procesom ažuriranja parametara smatrati samo težine i pragove). Također naveli smo da nam je cilj minimizirati funkciju cilja (2.14) tako što ćemo minimizirati funkciju troška po svakom ulaznom vektoru značajki. Proces optimizacije vršimo suprotnom smjeru gradijenta funkcije troška C , jer poznavajući gradijent funkcije koji predstavlja smjer i iznos najbržeg rasta funkcije, promjenom smjera možemo se približiti točki njenog minimuma. Pretpostavimo da funkcija troška C ovisi o parametrima $w_1, \dots, w_\zeta, b_1, \dots, b_\nu$ tada se njen gradijent računa:

$$\nabla C = \left(\frac{\partial C}{\partial w_1}, \dots, \frac{\partial C}{\partial w_\zeta}, \frac{\partial C}{\partial b_1}, \dots, \frac{\partial C}{\partial b_\nu} \right). \quad (2.15)$$

Tada ćemo se točki globalnog (ili lokalnog, što je jedan od problema ove metode jer ne garantira konvergenciju u globalni minimum) minimuma približavati iterativno:

$$\Theta^{(t+1)} = \Theta^{(t)} - \eta \nabla C(\Theta^{(t)}) \quad (2.16)$$

gdje je η **stopa učenja**, za koju je poželjno da "dovoljno" mala kako bi se osigurala postupna minimizacija funkcije troška C . Ukoliko je premala, proces optimizacije će sporo konvergirati, a ukoliko je prevelika, mogao bi oscilirati oko optimalnih parametara.

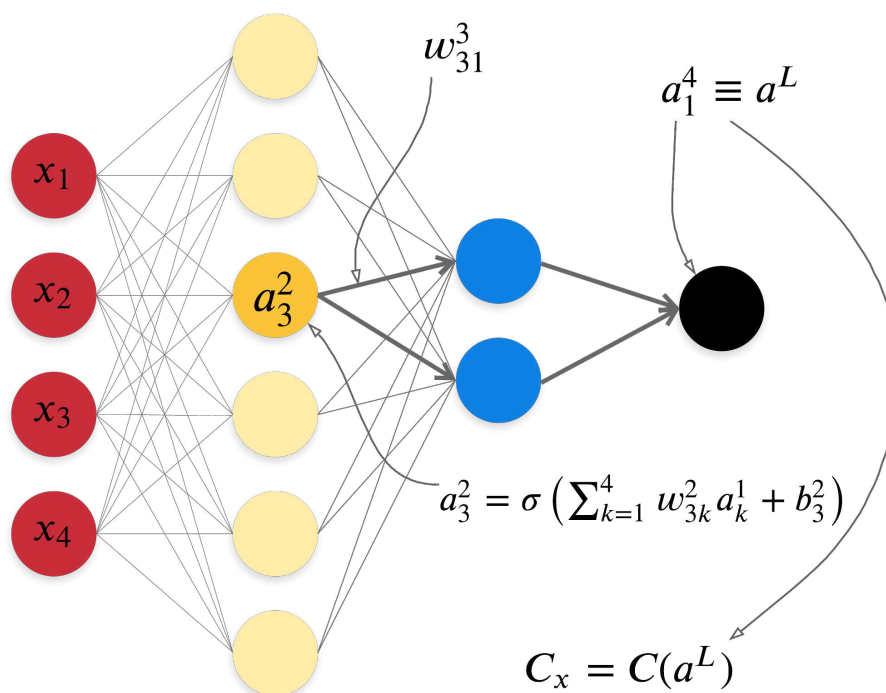


Slika 2.7: Intuitivni prikaz funkcije troška C kao trodimenzionalne plohe, te metode gradijentnog spusta koja se "malim" koracima spušta niz hiperravninu dok ne konvergira u minimum.

Propagacija greške unazad

Drugi pristup je algoritam propagacije greške unazad. Ovaj algoritam datira još iz 70-tih godina prošlog stoljeća, ali važnost mu raste tek nakon znanstvenog rada *Learning representations by back-propagating errors*². U navedenom radu, autori opisuju nekoliko arhitektura neuronskih mreža u kojima algoritam propagacije greške unazad radi daleko brže nego u prijašnjim pristupima, te omogućava modelima da riješe probleme koji su do tada bili nerješivi. Danas je propagacija greške unazad neizostavan faktor svih procesa učenja modela. Koristi se za efikasno računanje gradijenta te mu je ideja promatrati funkciju troška u svakome neuronu. Fokus je na parcijalnim derivacijama $\partial C/\partial w$ i $\partial C/\partial b$ funkcije troška C . One nam govore kako brzo se trošak mijenja s obzirom na promjene pragova i težina. Iako naizgled kompleksan, algoritam propagacije greške unazad daje dobar uvid u to kako promjena težina i pragova utječe na ponašanje cjelokupnog modela.

Prvo uvodimo oznake koje ćemo koristiti. Sa w_{jk}^l ćemo označavati težinu koja povezuje k -ti neuron $l-1$ -tog sloja sa j -otim neuronom l -tog sloja. Za pomak j -tog neurona u l -tom sloju, koristit ćemo oznaku b_j^l , dok ćemo za aktivaciju istog tog neurona koristimo oznaku a_j^l , te je ona definirana sa:



²David Rumelhart, Geoffrey Hinton i Ronald Williams, 1986.

$$a_j^l = \sigma \left(\sum_k w_{jk}^l a_k^{l-1} + b_j^l \right), \quad (2.17)$$

gdje suma iterira po svim neuronima u prethodnom sloju. Oznake su objašnjene u poglavlju 2.2, valja samo napomenuti još da a^l označava vektor aktivacija l -tog sloja. Sada 2.17 možemo zapisati sa:

$$a^l = \sigma \left(w^l a^{l-1} + b^l \right), \quad (2.18)$$

te izraz u zagradi možemo pridružiti oznaku $z^l \equiv w^l a^{l-1} + b^l$, koji ćemo referencirati kao težinski ulaz u neruone sloja l . z^l predstavlja vektor čija svaka komponenta z_j^l predstavlja težinski ulaz u j -ti neuron sloja l , $z_j^l = \sum_k w_{jk}^l a_k^{l-1} + b_j^l$. Ovaj zapis nam daje puno jasniji uvid kako aktivacije iz jednog sloja utječu na sljedeći, a i obratno. U moru indeksa teško je uočiti stvarnu bit, odnosno ne želimo od "drveta" ne vidjeti "šumu".

Kao što smo u uvodu ovog potpoglavlja rekli, naglasak je na parcijalnim derivacijama $\partial C / \partial w_{jk}^l$ i $\partial C / \partial b_j^l$. Ali da bi njih izračunali, uvodimo oznaku δ_j^l koja nam bilježi grešku j -tog neurona u l -tom sloju. Također potrebne su nam dvije pretpostavke:

1. prva je ta da se funkcija troška C može napisati kao aritmetička sredina funkcija troška svakog pojedinog ulaznog vektora značajki x :

$$C = \frac{1}{n} \sum_x C_x,$$

gdje je C_x zapravo predstavlja trošak u ovisnosti o parametru zadnje aktivacije, $C_x = \frac{1}{n} \sum_x C(a^L)$.

2. dok je druga ta da su moguće parcijalne derivacije za svaki x , tj. da su $\partial C_x / \partial w$ i $\partial C_x / \partial b$ dobro definirane.

Ideja algoritma je prvo izračunati odziv mreže za određeni ulaz (**propagirati ga unaprijed kroz mrežu**), odnosno aktivacije svih neurona u mreži, te nakon toga izračunati iznos funkcije troška. Kada znamo koliko ona iznosi, računamo greške izlaznog sloja, pa za prethodni sloj izračunamo koliko je svaki neuron utjecao na sljedeći. Na kraju je potrebno još izračunati gradijent po težinama i pomacima koje povezuju ta dva sloja i ažurirati ih. Postupak se nastavlja između preostalih slojeva, sve dok ne dođemo do ulaznog sloja, **propagiramo unatrag**.

Pa krenimo sa formalnim postupkom:

1. izračunamo odziv mreže, tj. aktivacije svih neurona u mreži za ulazni vektor
2. računamo greške u svakom neuronu počevši od zadnjeg sloja:

$$\delta_j^L = \frac{\partial C}{\partial a_j^L} \sigma'(z_j^L) \quad (2.19)$$

ili u matricnom zapisu:

$$\delta^L = \nabla_a C \odot \sigma'(z^L), \quad (2.20)$$

gdje je $\nabla_a C$ gradijent funkcije troška po aktivacija u trenutnom sloju, a \odot Hadamardov produkt³.

3. zatim računamo grešku u neuronima prijašnjeg sloja, izraženu pomoću izračunatih parametara trenutnog:

$$\delta^l = \left((w^{l+1})^T \delta^{l+1} \right) \odot \sigma'(z^l), \quad (2.21)$$

gdje $(w^{l+1})^T$ predstavlja transponiranu matricu težina w^{l+1} . Ako znamo greške u prednjem sloju, možemo grešku propagirati na prethodni pomoću aktivacija trenutnog.

4. Sada kada imamo sve potrebno, odredimo parcijalne derivacije po težinama:

$$\frac{\partial C}{\partial w_{jk}^l} = a_k^{l-1} \delta_j^l, \quad (2.22)$$

i pomacima:

$$\frac{\partial C}{\partial b_j^l} = \delta_j^l, \quad (2.23)$$

te ih ažuriramo ih u (2.16).

Korake 2., 3. i 4. ponavljamo dok ne dođemo do ulaznog sloja tako da nakon što ažuriramo težine između zadnjeg i predzadnjeg sloja, isto to radimo za predzadnji i njegov prethodni sloj. Cjelokupni postupak (sve korake) ponavljamo dok ne postignemo zadovoljavajuću vrijednost funkcije troška po svim podacima za učenje, što je zapravo proces učenja.

³definiran za matrice A, B jednakih dimenzija, tada je $(A \odot B)_{ij} = A_{ij}B_{ij}$

Dodatak o aktivacijskim funkcijama

Vratimo se sada na navedene aktivacijske funkcije. Identiteta (2.1) je najprimitivniji oblik aktivacijske funkcije. Zbog njene linearnosti, učenje mreže je ograničeno jer je kompozicija linearnih funkcija opet linearna funkcija, te bi svaku mrežu sa k slojeva i m izlaza bilo moguće zamijeniti sa jednim slojem od m neurona. Primjenjuje se samo u jednostavnim ili u regresijskim problemima gdje nismo ograničeni na vrijednost koju trebamo predvidjeti. Kod funkcije praga (2.2) vidimo da će neuron biti aktiviran samo i ako samo je izraz $\vec{x} \cdot \vec{w} - b$ veći ili jednak 0. Problem kod ove funkcije je taj da nije diferencijabilna u 0, te je učenje gradijentnim metodama otežano. \tanh (2.4) i *sigmoid*-na funkcija (2.3) pokazuju bolja svojstva. Tangens hiperbolni će negativne vrijednosti preslikati u izrazito negativne, dok će vrijednosti blizu 0 biti preslikane blizu 0. Kodomena *sigmoid*-ne funkciju sugerira korištenje kada treba prepostaviti vjerojatnost pripadnosti nekoj klasi. Izrazito je korisna u binarnoj klasifikaciji. *softmax* (2.7) je generalizirana *sigmoid*-na funkcija, djeluje na cijeli vektor za razliku od *sigmoid*-ne koja djeluje na skalar. Za svaku komponentu vektora predviđa normaliziranu vrijednost pripadnosti nekoj klasi te zbroj svih tih vrijednosti iznosi 1. To nam sugerira da je *softmax* pogodna za korištenje kod višeklasne klasifikacije, te će imati izrazito važnu primjenu u segmentaciji slike. Kod binarne klasifikacije, *sigmoid*-na i *softmax* funkcija su ekvivalentne. *ReLU* (2.5) je trenutno jedna od najčešće korištenih aktivacijskih funkcija. Mogući problem kod ove funkcije je da negativne vrijednosti preslikava u 0, što umanjuje sposobnost modela da uči iz podataka. Zbog toga se uvodi *softplus* (2.6), te ona predstavlja glatku aproksimaciju *ReLU* funkcije.

Navest ćemo često korištene funkcije troška za segmentaciju slike [5]:

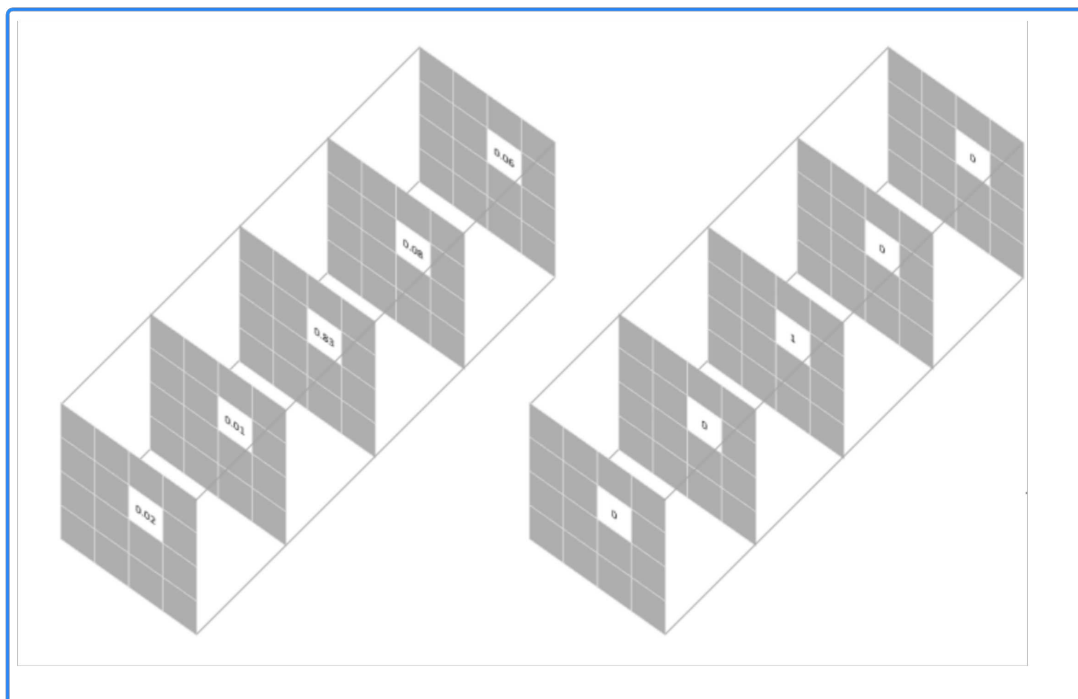
1. unakrsna entropija — ako za višeklasnu klasifikaciju koristimo *softmax* (2.7), gdje kao izlaz iz sloja dobivamo vjerojatnosnu razdiobu pripadnosti nekoj klasi, unakrsna entropija je najpogodnija funkcija troška u tome slučaju:

$$C = -\frac{1}{n} \sum_{i=1}^n \sum_c y_{ic} \ln a_{ic}^L, \quad (2.24)$$

gdje je y_{ic} označava vjerojatnost pripadnost stvarnoj klasi, dok a_{ic}^L predstavlja vjerojatnost pripadnosti klasi c koju je model izračunao.

2. negativni logaritam izglednosti — ako y_{ic} označava istinitost, tj. 1 ako piksel pripada klasi c , 0 inače onda se (2.24) svodi na:

$$C = -\frac{1}{n} \sum_{i=1}^n \ln a_{ic}^L, \quad y_{ic} = 1 \quad (2.25)$$



Slika 2.8: Primjer izlaza neuronske mreže kod segmentacije slike. Dimenzija oba vektora su određene brojem klasa na slici. Desna slika predstavlja vektor čija je samo jedna komponenta jednaka 1 za određenu klasu (eng. one-hot encoded vector), dok lijeva slika pokazuje izlaz modela koji pretpostavlja pripadnost piksela svakoj klasi. Poveznica između ta dva vektora može biti argmax funkcija.

3. Relaksirani Dice-ov⁴ koeficijent — još jedna popularna funkcija troška za segmentaciju slike koja se bazira na Dice-ovom koeficijentu, koji je izvorno mjera sličnosti dva uzorka. Mjera je iz segmenta $[0, 1]$, gdje 1 označava da su uzorci isti, odnosno da se savršeno preklapaju. Računa se formulom:

$$Dice = \frac{2|A \cap B|}{|A| + |B|} \quad (2.26)$$

gdje $|A \cap B|$ označava zajedničke elemente skupova A i B , dok $|A|$ označava kardinalnost skupa A (2 je u nazivniku zbog toga što ih nazivnik broji dva puta također).

⁴Lee Raymond Dice, američki ekolog i genetičar. Definirao je ovu mjeru 1945. Tri godine nakon, nezavisno o njegovome radu, isto je učinio i danski botaničar Thorvald Sørensen pa se ovaj koeficijent naziva i Sørensen-Diceov koeficijent

Vratimo se sada na sliku (1.1) i promotrimo matricu sa oznakama klasa za pojedini piksel. Ako svaku oznaku razdvojimo u matricu jednakih takvih dimenzija, dobivamo maske, odnosno filtere za naš problem. Sada $|A \cap B|$ u segmentaciji možemo aproksimirati pomoću množenja po elementima:

$$\begin{aligned}
 |A \cap B| &= \begin{bmatrix} 0.01 & 0.03 & 0.02 & 0.02 \\ 0.05 & 0.12 & 0.09 & 0.07 \\ 0.89 & 0.85 & 0.88 & 0.91 \\ 0.99 & 0.97 & 0.95 & 0.97 \end{bmatrix} \odot \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} \\
 &= \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0.89 & 0.85 & 0.88 & 0.91 \\ 0.99 & 0.97 & 0.95 & 0.97 \end{bmatrix} \xrightarrow{\text{sumiramo ih}} 7.41 \\
 |A| &= \begin{bmatrix} 0.01 & 0.03 & 0.02 & 0.02 \\ 0.05 & 0.12 & 0.09 & 0.07 \\ 0.89 & 0.85 & 0.88 & 0.91 \\ 0.99 & 0.97 & 0.95 & 0.97 \end{bmatrix} \rightarrow 7.82 \\
 |B| &= \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} \rightarrow 8
 \end{aligned} \tag{2.27}$$

Ako matricu A promatramo kao predviđene vrijednosti pripadnosti klasi, a B kao ciljne (ovo će važiti i za sljedeće funkcije), onda efektivno poništimo piksele koji se nisu aktivirali pomoću našeg filtera. Da bi formulirali funkciju troška koju možemo minimizirati, koristimo $1 - Dice$ i nazivamo ju relaksiranim Dice-ovim koeficijentom zbog toga što koristimo vjerojatnosti umjesto aktiviranja i konvertiranja u binarne vrijednosti. Zapišimo još $Dice$ jezikom segmentacije, odnosno u ovisnosti o pikselima (iteriramo po njima):

$$C = 1 - \frac{2 \sum_{i=1}^n y_i \hat{y}_i}{\sum_{i=1}^n y_i + \sum_{i=1}^n \hat{y}_i} \tag{2.28}$$

I to radimo za svaku klasu pojedinačno, te na kraju računamo prosječnu vrijednost kako bi dobili krajnju mjeru greške. Treba naglasiti kako neki znanstvenici kao $Dice$ koriste:

$$Dice = \frac{2|A \cap B|}{|A|^2 + |B|^2} \tag{2.29}$$

što možemo obrazložiti time kako bi veće vjerojatnosti zadržali visokim, a manje još minorizirali.

4. IoU (eng. *Intersection over Union*) — još se naziva Jaccard-ov⁵ indeks

$$IoU = \frac{|A \cap B|}{|A \cup B|} \quad (2.30)$$

5. preciznost — najjednostavnija mjera, omjer točno klasificiranih i ukupnih

$$P = \frac{|A \cap B|}{|A|} \quad (2.31)$$

6. odziv — vrlo važna mjera kod neuravnoteženih skupova podataka, jer uzima u obzir veličinu ciljane klase

$$R = \frac{|A \cap B|}{|B|} \quad (2.32)$$

7. F_1 -mjera — definira se pomoću prijašnje dvije.

$$F_1 = 2 \cdot \frac{P \cdot R}{P + R} \quad (2.33)$$

Radi jednostavnosti, formule svih mjera su napisane za pojedinu klasu. Za krajnju točnost bi trebali naći srednju vrijednost pojedinih mjera svih klasa.

8. Posljednja, ali ne i najmanje važna mjera će biti izvedena mjera **S** koja je korištena kao metrika u *Data Science Bowl 2018*. natjecanju koje će kasnije biti detaljnije opisano. Strategija ocjenjivanja temeljila se na utvrđivanju pogrešaka na razini objekta. To je postignuto podudaranjem maski ciljnog objekta s predviđenim objektima koje su podnijeli sudionici, a zatim izračunavanjem stvarnih pozitivnih rezultata (**TP**)⁶ i lažnih pozitivnih rezultata (**FP**)⁷. Kako bi se uskladile ciljane maske i predviđeni objekti, za sve parove objekata korištena je mjera IoU gdje su A i B dva objekta.

Za identificiranje ispravno segmentiranih objekata odabran je minimalni IoU prag t , gdje je bilo koja druga predviđena maska segmentacije ispod praga smatrana pogreškom. Uz sve stvarne pozitivne (**TP**), lažne pozitivne (**FP**), stvarne negativne (**TN**)⁸ i lažne negativne (**FN**)⁹, radi usporedbi računate su već spomenute mjere uz fiksni prag t :

⁵Paul Jaccard, švicarski botaničar. U svome radu "*Distribution de la flore alpine dans le bassin des Dranses et dans quelques régions voisines*. *Bulletin de la Société Vaudoise des Sciences Naturelles*" je definirao indeks sličnosti koji je kasnije dobio ime po njemu

⁶TP = ishod u kojemu model točno predvidi pozitivnu klasu, npr. jezgru stanice

⁷FP = ishod u kojemu model netočno predvidi pozitivnu klasu, npr. nije označio jezgru stanice tamo gdje je trebao

⁸TN = ishod u kojemu model točno predvidi negativnu klasu, npr. tamo gdje nije jezgra i ne označi jezgru

⁹FN = ishod u kojemu model krivo predvidi negativnu klasu, npr. tamo gdje nema jezgre označi da se nalazi jezgra

$$P(t) = \frac{TP(t)}{TP(t) + FP(t)},$$

$$R(t) = \frac{TP(t)}{TP(t) + FN(t)},$$

$$F_1(t) = \frac{2TP(t)}{2TP(t) + FP(t) + FN(t)}.$$

Korištene su sve veći IoU pragovi za procjenu točnosti uparivanja oblika. Kada je segmentacija savršeno prekrila ciljnu masku, rezultat IoU-a bio je 1 i objekt je ispravno otkriven bez obzira koji je prag korišten. U praksi, segmentacije mogu samo aproksimirati stvarni oblik objekta, pa je na određenom pragu pokrivenosti objekt model propustio dobro segmentirati oblik. Ovim je ispitivanjem procijenjeno koliko su se dobro segmentacije podudarale s oblikom ručno definiranih ciljnih maski. Zatim je službeni rezultat natjecanja definiran metrikom S definirana primjenom više pragova IoU-a:

$$S = \frac{1}{|T|} \sum_{t \in T} \frac{TP(t)}{TP(t) + FP(t) + FN(t)}, \quad (2.34)$$

gdje je $t \in T = \{0.10, 0.15, \dots, 0.95\}$.

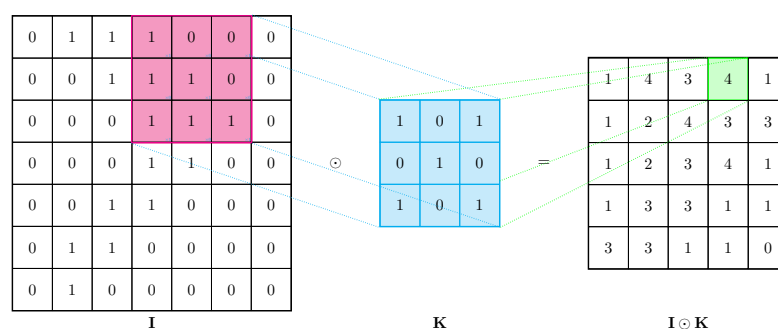
2.3 Modeli dubokog učenja

Prije nego krenemo na konkretne modele dubokih neuronskih mreža koji se koriste u biološkoj segmentaciji slike, navest ćemo neke njihove gradivne dijelove i karakteristike.

Konvolucija

U prijašnjem opisu općenite neuronske mreže vidimo da je svaki neuron povezan sa svim neuronima iz prethodnog sloja (2.6) te je aktivacija neurona predstavljena skalarnom vrijednosti. Primijenimo li takvu mrežu na probleme računalnog vida, čiji su ulazi monokromatske ili višedimenzionalne slike (npr. RGB slike), pojavio bi nam se problem prenaučnosti mreže te bi ona imala prevelik broj parametara, što bi izravno utjecalo na trajanje i preformanse učenja. Zbog toga se uvode konvolucijske neuronske mreže koje umjesto neurona mogu sadržavati i matrice ili čak i tenzore, kako bi iskoristile lokalne ovisnosti u slikama, jer je svaki piksel povezan samo sa okolnim pikselima, te nema potrebe povezivati ulazni neuron koji ga reprezentira sa svima ostalima. Zato se uvodi konvolucijski sloj, sloj kojeg čini rezultati Hadamardovog produkta (**konvolucije**) podmatrica ulaza i matrice $K_{n \times n}$ (gdje je n najčešće neparan) koje zovemo **jezgrom** ili **filterom** (eng. *kernel*).

Elementi jezgre su težine, koje model uči, a ulazno polje na kojemu se vrši konvolucija nazivamo **receptivno** polje. Ukoliko je ulaz višedimenzionalan (poput RGB ulaza), jezgra također može biti višedimenzionalna. Rezultat se sprema u izlaznu matricu, koju nazivamo **mapom značajki** ili (eng. *feature map*), te ona predstavlja ulaz u sljedeće slojeve. Na elemente mape značajki primjenjujemo aktivacijsku funkciju pa se često mape značajki nazivaju i **aktivacijske mape**. Konvolucijski sloj igra ulogu ekstraktora značajki.



Također, spomenimo još dilatiranu konvoluciju ili dilataciju. To je oblik konvolucije u kojoj se jezgra proširuje sa 0-ma, te se time istovremeno proširuje receptivno polje u sljedećim slojevima, bez povećanja parametara mreže i složenosti učenja. Zbog toga se često primjenjuje aplikacijama koje segmentiraju sliku u realnom vremenu, gdje si ne možemo priuštiti dodatne konvolucijske slojeve.

Primjer 2.3.1. *Primjer proširenja jezgre. Množenjem ulazne podmatrice sa ovakom jezgrom dobili bi rezultat veće dimenzije, a time povećano receptivno polje u sljedećim slojevima.*

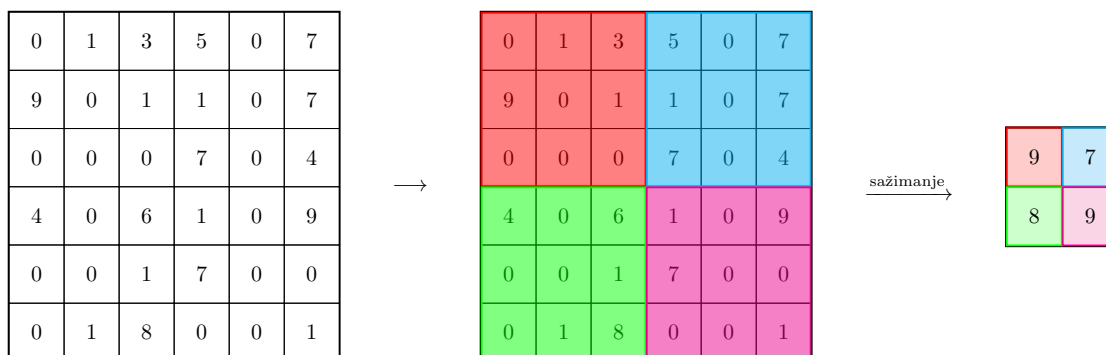
$$\begin{bmatrix} w_{00} & w_{01} & w_{02} \\ w_{10} & w_{11} & w_{12} \\ w_{20} & w_{21} & w_{22} \end{bmatrix} \rightarrow \begin{bmatrix} w_{00} & 0 & w_{01} & 0 & w_{02} \\ 0 & 0 & 0 & 0 & 0 \\ w_{10} & 0 & w_{11} & 0 & w_{12} \\ 0 & 0 & 0 & 0 & 0 \\ w_{20} & 0 & w_{21} & 0 & w_{22} \end{bmatrix} \quad (2.35)$$

Sažimanje

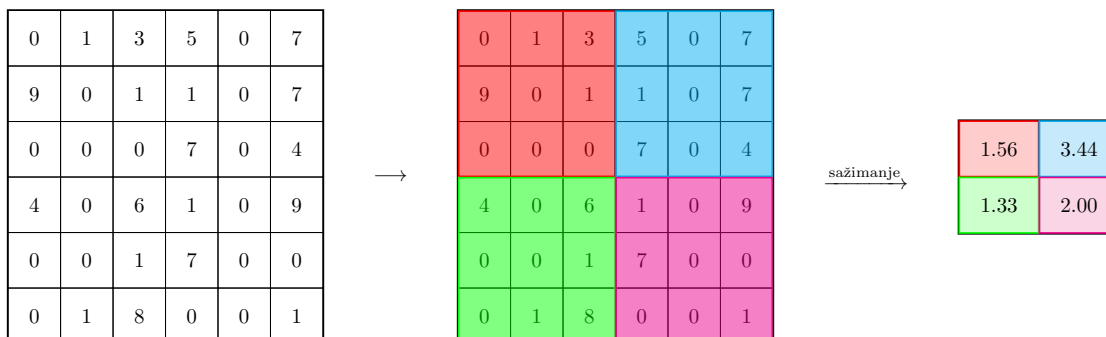
Kako bi dodatno smanjili parametre mreže, uvodimo slojeve sažimanja. Time izravno ubrzavamo učenje, ali i sklonost prenaučivosti. Ulazna mapa značajki dijeli na nepreklapajuće podmatrice proizvoljne dimenzije, zatim se određenom tehnikom računa vrijednost koja sažima određenu podmatricu. Postoji mnogo različitih načina sažimanja, a kao jedan od najefikasnijim se pokazalo sažimanje maksimalnom vrijednošću (eng. *max-pooling*). Skicirat ćemo ga na primjeru sažimanje maksimalnom vrijednošću i sažimanje prosječnom

vrijednošću, a ostale tehnike poput sažimanja Gaussovom jezgrom ili L^p normom ostavljamo čitatelju na proučavanje.

- sažimanje maksimalnom vrijednošću



- sažimanje prosječnom vrijednošću



Dekonvolucija

Kako kod klasične konvolucije i sažimanja malo po malo od slike visoke rezolucije dobivamo manje i manje aktivacijske mape, odnosno smanjujemo rezoluciju. Pomoću te dvije operacije dobivamo samo odgovor na pitanje što se nalazi na slici, ali ne i odgovor gdje se nalazi. Da bi riješili pitanje lokalizacije, moramo nekako iz smanjenih aktivacijskih mapa dobiti početno polje interesa. To radimo dekonvolucijama, operacijama inverznim konvoluciji, te tako sliku niske rezolucije vraćamo u početnu dimenzionalnost.

Razdvojene konvolucije

Također jedan od oblika smanjenja složenosti učenja mreže u kojemu razdvajamo jezgru u dva dijela, što je moguće ako i samo ako je ona ranga $r = 1$. Ako je matrica reda (m, n) , umjesto $m \times n$ množenja u konvoluciji, u razdvojenoj konvoluciji imamo $m + n$ množenja. Ukoliko imamo k konvolucija u konvolucijskom sloju, to je znatno smanjenje složenosti.

Primjer 2.3.2. *Primjer matrice ranga $r = 1$, koju možemo razdvojiti.*

$$\begin{bmatrix} 3 & 6 & 9 \\ 4 & 8 & 12 \\ 5 & 10 & 15 \end{bmatrix} = \begin{bmatrix} 3 \\ 4 \\ 5 \end{bmatrix} \times \begin{bmatrix} 1 & 2 & 3 \end{bmatrix} \quad (2.36)$$

Primjer 2.3.3. *Također jedna od važnijih jezgri koja se koristi u detekciji ruba je Sobelova¹⁰ jezgra.*

$$\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} \times \begin{bmatrix} -1 & 0 & 1 \end{bmatrix} \quad (2.37)$$

Potpuno povezani sloj

Potpuno povezani sloj je sloj na kraju konvolucijske neuronske mreže. Sastavljen je od neurona u potpunosti povezanih svakim elementom aktivacijske mape iz prethodnog sloja. Nije neuobičajeno da se koristi više potpuno povezanih slojeva kraju. Na te neurone se obično primjenjuje 2.7 ukoliko je riječ o klasifikacijskom problemu. Potpuno povezani sloj zapravo igra ulogu klasifikatora koristeći značajke koje su ekstrahirali prijašnji slojevi.

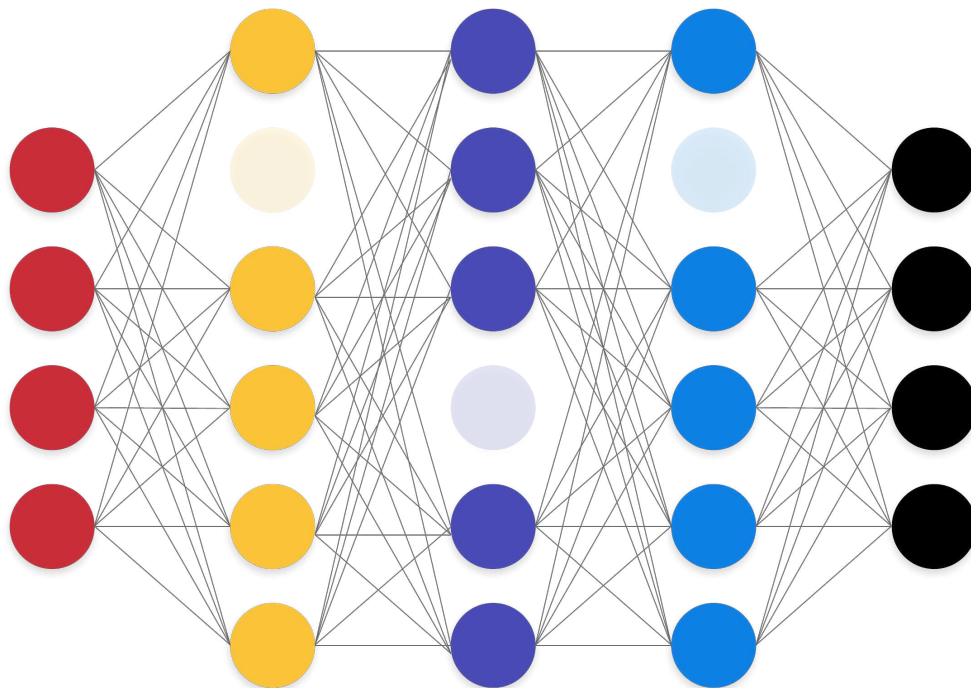
Dodatne metode optimizacije učenja

Prenaučenost definiramo kao preveliku prilagođenost modela skupu za učenje. Ne želimo model koji će imati savršenu točnost na skupu za učenje, već model koji će dobro generalizirati, odnosno dobro se ponašati na neviđenim podacima. Isto tako, ne želimo ni model koji će imati lošu mjeru točnosti na skupu za učenje, jer takav model sigurno neće imati svojstvo generalizacije, ukoliko nije savladao ni trening podatke. Zato je bitno znati zaustaviti učenje u pravom trenutku. Učenje se obično odvija u epohama. Neke od tehnika sprječavanja prenaučivosti su:

- grupna normalizacija — omogućava brže učenje mreže tako što normalizira vrijednosti unutar slojeva, omogućava korištenje većih stopa relativizira lošu inicijalizaciju početnih težina.

¹⁰Irwin Sobel, znanstvenik koji se bavi istraživanjima na području digitalne obrade slike i računalnog vida.

- isključivanje neurona — uklanja mogućnost da neki neuron ima prevelik utjecaj u modelu, tako što ga u nekim fazama učenja isključi zajedno sa svim njegovim konekcijama, tako tjera ostale neurone da nauče bitne značajke, umjesto da se oslanjaju na isključene neurone. Ovaj proces direktno utječe na poboljšanje generalizacije.



- L^p -regularizacija — ukratko, tehnika koja dodatnim parametrima regulira funkciju troška C :

$$C_r = C + \lambda \|w\|_p^p, \quad (2.38)$$

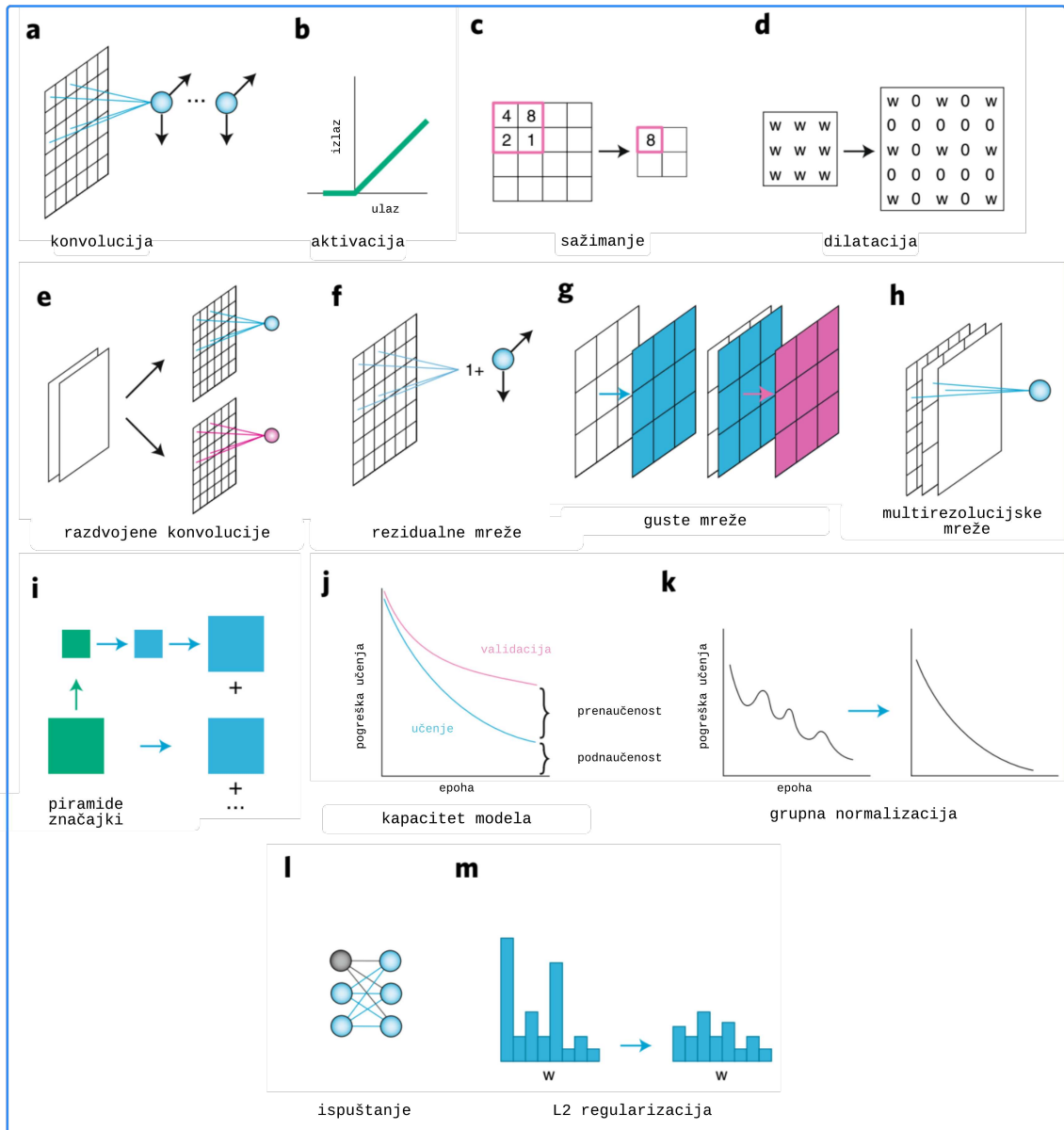
gdje $\|w\|_p^p$ predstavlja L^p normu vektora težina w :

$$\|w\|_p = \left(\sum_i w_i^p \right)^{\frac{1}{p}}. \quad (2.39)$$

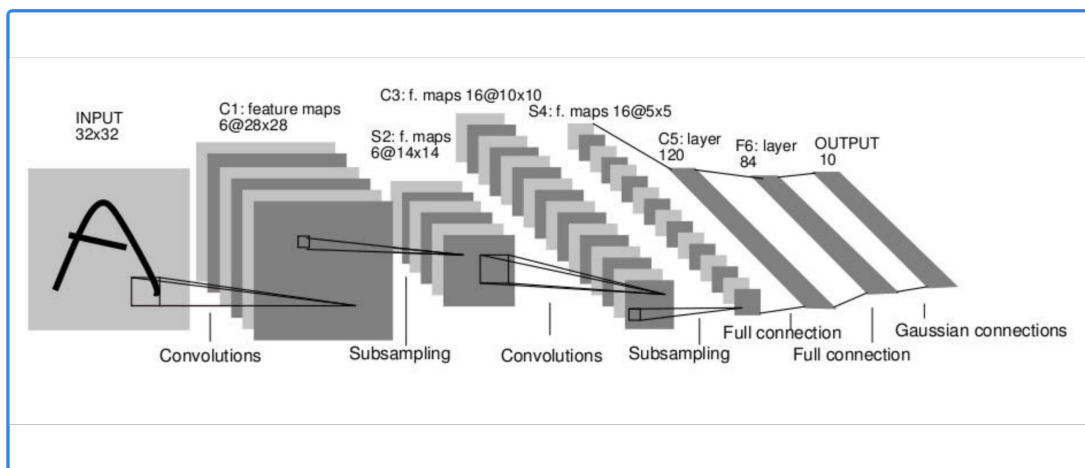
a λ hiperparametar kojim *penaliziramo* težine, te tjeramo model da traži uzorke u ulaznim podacima, a ne da ih pamti i tako postigne optimalnu vrijednost funkcije cilja.

- dopunjavanje — mape značajki dopunjavamo oko rubova (najčešće nulama). Na taj način povećavamo dimenzije i složenost, ali zapravo nam ova tehnika znatno

pomaže jer će rubna receptivna polja više puta sudjelovati u konvoluciji te tako više pridonijeti otkrivanju objekata koje je slika možda dijelom zahvatila.



Slika 2.9: Kratak sažetak arhitekturnih principa i metoda ubrzavanja učenja. Izvor: [6].



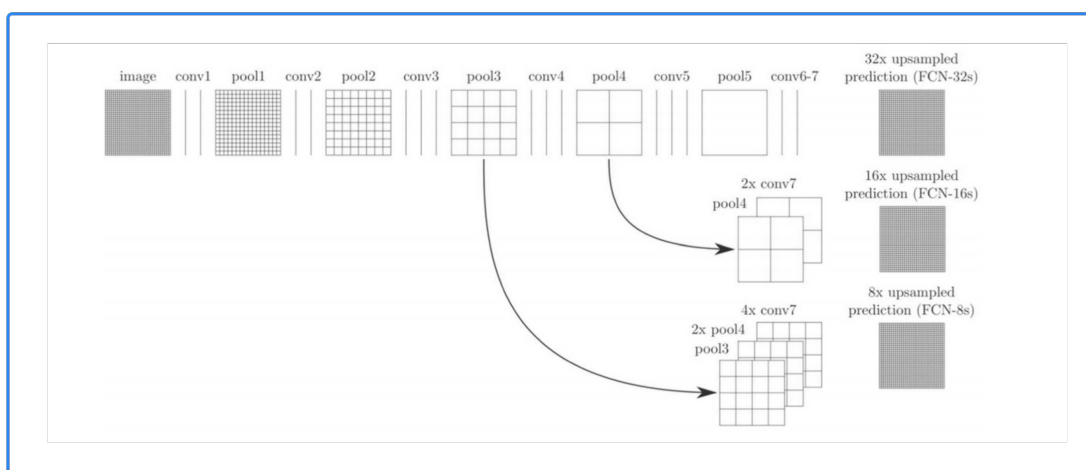
Slika 2.10: Struktura CNN. Izvor: [2]

CNN

Konvolucijska neuronska mreža (eng. *Convolutional Neural Network*) je arhitektura neuronske mreže koja se sastoji se od većeg broja slojeva od kojih svaki izvodi određenu operaciju, npr. konvoluciju, aktivaciju, sažimanje, izračun gubitka itd. Svaki od međuslojeva dobiva izlaz prethodnog sloja kao ulaz. Početni sloj je sloj koji je izravno povezan s ulaznom slikom, te se sastoji od broja neurona jednakom broju piksela u ulaznoj slici. Sljedeći sloj je konvolucijski sloj koji prikazuje rezultate konvolucije s određenom jezgrom te kao njihov rezultat nastaju mape značajki. Jezgre su proizvoljne veličine, definirane od strane dizajnera mreže. Svaki neuron reagira samo na određeno područje prethodnog sloja, koje smo nazvali receptivnim poljem. Izlaz svakog konvolucijskog sloja smatra se aktivacijskom mapom, koja je rezultat primjene određenog filtra (jezgre) na ulaz. Nakon konvolucijskih slojeva obično slijede aktivacijski slojevi kako bi se na aktivacijske mape primijenila nelinearnost. Sljedeći sloj može biti sloj sažimanja koji pomaže u smanjenju dimenzionalnosti konvolucijskih izlaza. Za sažimanje postoji nekoliko strategija koje smo naveli, te one ovise o domeni i aplikaciji problema, baš kao i jezgre. Na poslijetku, visoka razina apstrakcije se ekstrahira potpuno povezanim slojevima. Težine neuronskih veza i jezgri se kontinuirano ažuriraju tijekom propagacije greške unazad tokom učenja. Navedena struktura poznata je kao konvencionalna CNN [8].

FCN

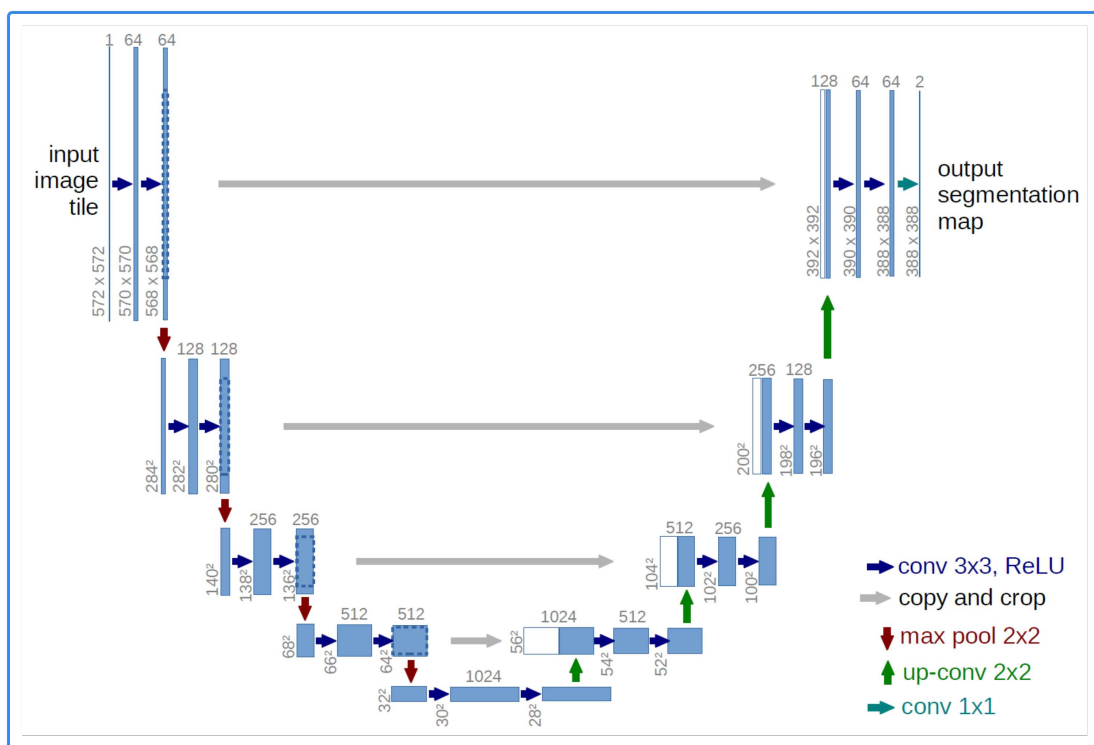
Potpuna konvolucijska mreža (eng. *Fully Convolutional Network*) je arhitektura neuronske mreže u kojoj je posljednji potpuno povezani sloj zamijenjen sa potpunim konvolucijskim slojem. Ovo poboljšanje omogućuje mreži da klasificira na razini piksela. Kako bi se postigla bolja lokalizacija, aktivacijske mape visoke rezolucije kombiniraju se s uvećanim mapama značajki koje se dobiju dekonvolucijom te se propuštaju kroz slojeve konvolucije kako bi se pridobio točniji rezultat. Ovo poboljšanje omogućuje modelu predviđanje na razini piksela iz slike pune rezolucije umjesto predviđanja na temelju filterirane mape značajki u prethodnom sloju. Ovakav model je u mogućnosti provesti predviđanje za cijelu sliku u samo jednom prolazu kroz mrežu.



Slika 2.11: Struktura FCN. Izvor: [8].

U-Net

Jedna od najpoznatijih arhitektura neuronske mreže za segmentaciju slike na području medicine. Izgrađen je na strukturi FCN. Osim povećane dubine modela na 19 slojeva, U-Net je unaprijeđena jer ju krasi svojstva rezidualne mreže. Veliki napredak u segmentaciji ostvaruje pomoću veza među udaljenim slojevima, što modelu omogućava gledanje u glatke u grube značajke slike. Koristi već spomenute modifikacije poput dekonvolucije kako bi prevladao probleme između lokalizacije i veličine receptivnog polja, koji se javljaju kada sliku visoke rezolucije konvolucijama i sažimanjem spustimo u nižedimenzionalne prostore. U-Net se sastoji od dva glavna sloja:



Slika 2.12: Struktura U-Net modela. Izvor: [16].

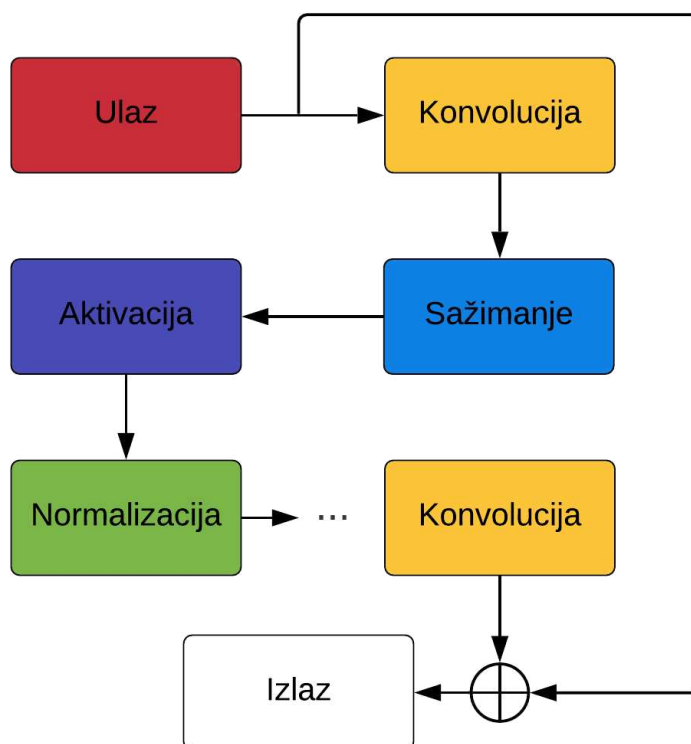
1. analitičkog sloja ili sloja kontrakcije (eng. *encoder*) — obavlja prepoznavanje objekata sa slike, odnosno daje odgovor na pitanje što se na slici nalazi. Sastoji se od konvolucijskih i slojeva slojeva sažimanja.
2. sintetičkog sloja ili sloja ekspanzije (eng. *decoder*) — obavlja lokalizaciju objekta, odnosno daje odgovor na pitanje gdje se prepoznati objekt sa slike nalazi. Sastoji se od dekonvolucijskih slojeva i sličnih slojeva koji povećavaju rezoluciju.

Najvažnije svojstvo U-Net modela je su postojanje veza između slojeva jednake rezolucije, koje kreću iz analitičkog sloja prema sintetičkom sloju iste rezolucije. Te veze omogućuju da esencijalne značajke visoke rezolucije budu dostupne dekonvolucijskim slojevima, te da im olakšaju lokalizaciju.

CRN

Konvolucijska rezidualna mreža (eng. *Convolutional Residual Network*) je arhitektura neuronske mreže koja može sadržavati i rezidualne blokove. Teoretski, dokazano je da

dublje mreže imaju veću sposobnost učenja, ali dublje mreže ne pate samo od problema iščezavajućeg gradijenta, već se suočavaju i s važnijim pitanjem degradacije. To znači kako se dubina povećava, točnost se zasićuje i potom brzo degradira. Kako bi iskoristili dublju strukturu mreže, He i suradnici [13] uveli su rezidualne mreže koje su prvotno razvijane za segmentaciju 2D slika. U ovom modelu, umjesto uzastopnog propuštanja mape značajki kroz naslagane slojeve konvolucije i sažimanja, rezidualna mapa se propušta kroz samo nekoliko slojeva. Drugim riječima, rezidualne mape su preskakajuće veze koje omogućavaju mreži da preusmjeri mape značajki na neke slojeve preskačući ostale. Ovaj dizajn pomaže mreži da iskoristi točnosti naučene u dubljim slojevima.



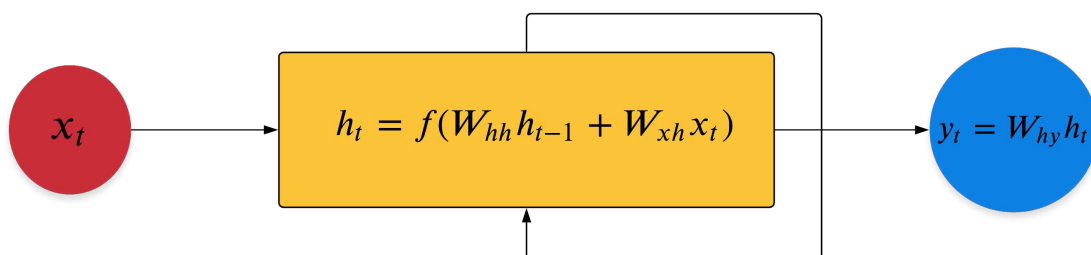
Slika 2.13: Primjer strukture rezidualnog bloka neuronske CRN. Rezidualni blok može imati različite kombinacije i broj slojeva.

Rezidualne neuronske mreže omogućavaju ulaznom vektoru da simultano preskoči i prođe kroz neke slojeve (u najčešćem slučaju konvolucijski i sloj sažimanja). Preskakanje slojeva može ubrzati učenje mreže, što pomaže kod problema iščezavajućeg gradijenta, problema gdje je gradijent toliko mali da sprječava promjenu vrijednosti novih težina u

formuli 2.16, što zaustavlja učenje daljnjih slojeva. Uvođenje više takvih konekcija među udaljenim slojevima, omogućava mreži da gleda u sve prijašnje slojeve te takve mreže nazivamo gustim konvolucijskim neuronski mrežama (eng. *DNN*). Povezivanjem klasifikacijskih neurona zadnjeg sloja sa prijašnjim konvolucijskim slojevima, omogućavamo modelu da vidi i grube i glatke uzorke, što nam daje multi-rezolucijsku konvolucijsku neuronsku mrežu (eng. *MNN*). Na taj način mreža stvara piramide značajki, što daje skalabilnost u detekciji objekata. Guste mreže omogućavaju istraživanje šireg spektra značajki, dok rezidualne mreže omogućuju ponovnu uporabu već otkrivenih značajki u kasnijim slojevima.

RNN

Rekurentna neuronska mreža (eng. *Recurrent Neural Network*) je arhitektura neuronske mreže osnažena rekurentnim vezama koje omogućuju mreži da zapamti uzorke iz posljednjeg unosa. Činjenica je da se područje interesa u medicinskim snimkama obično distribuira na više susjednih rezova (npr. na CT ili MRI), to rezultira korelacijom u uzastopnim rezovima. U skladu s tim, RNN modeli mogu iz ulaznih isječaka izvući uzorke iz sekvencijalnih ulaza.

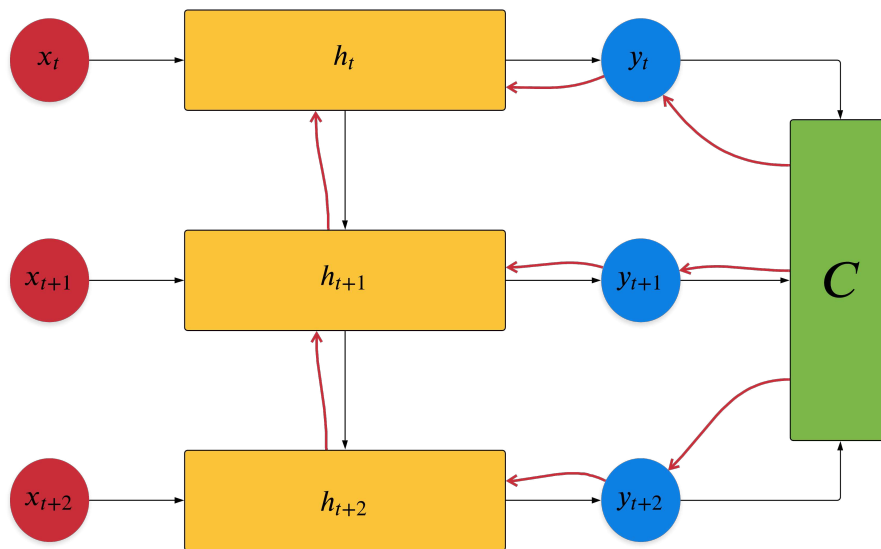


Slika 2.14: Rekurentni blok se sastoji od ulaznog sloja, rekurentnog sloja i izlaznog sloja. Ulazni sloj uzima vektor x_t u trenutku t . Rekurentni sloj množi ulazni vektor x_t sa matricom težina između ulaznog sloja i skrivenog sloja W_{xh}^T te pribraja produkt matrice težina vektora stanja skrivenog sloja h_{t-1} iz prošlog trenutka $t - 1$. Na njihov zbroj se primjenjuje aktivacijska funkcija. Izlat y_t predstavlja vektor predikcije, koji se dobije kao produkt izlaza iz rekurentne ćelije i matrice težina između skrivenog i izlaznog sloja. Važno za uočiti je to da RNN sadži tri različite matrice težina čije elemente treba ažurirati tokom učenja.

RNN možemo predstaviti kao neuronsku mrežu koja ima internu memoriju. Posjeduje mogućnost zadržavanja informacije koje je izvukla iz prijašnjeg ulaza. Rekurentne neuronske mreže su izrazito dobre u aplikacijama prevođenja i predikcije. RNN možemo zamisliti kao normalnu potpuno povezanu neuronsku mrežu. U svakom unaprijednom prolazu, stanje se pamti i istovremeno prosljeđuje sljedećem prolazu. Na taj način RNN može

zapamtiti stanje prošlog prolaza. Osnovna gradivna jedinica rekurentne neuronske mreže je rekurentni blok.

Problem kod rekurentnih neuronskih mreža je taj da pate od učenja dalekosežnih veza. Tijekom vremena, stariji ulazi postaju manje značajni te RNN ne uspijeva zadržati kontekst iz ranijih koraka. Kao rješenje toga se uvode LSTM (eng. *Long Short Term Memory*) mreže koje koriste vrata (eng. *gate*) kako bi regulirale tok podataka koji prolazi kroz njih. Vrata diktiraju koji su podaci važni, a koji se trebaju odbaciti.



Slika 2.15: Propagacija greške unatrag kroz vrijeme u RNN. Funkcija troška se računa u svakom koraku t i sumira kako bi se dobila ukupna vrijednost troška učenja mreže. Zatim se propagira greška unatrag kako bi se mogle ažurirati težine.

Poglavlje 3

Segmentacija jezgre stanice na mikroskopskim slikama: *2018 Data Science Bowl*

Segmentacija jezgre stanica na mikroskopskim snimkama često je prvi korak u kvantitativnoj analizi podataka za biološku i biomedicinsku primjenu. Mnogi alati za analizu slike mogu prepoznati jezgre u slikama, ali prije ikakvog rada treba odabrati i podesiti pravi alat za svaki eksperiment. *Data Science Bowl 2018*. privukao je 3 891 timova iz cijelog svijeta kako bi iznjedrio pravu metodu segmentacije jezgre stanice koja bi se mogla primijeniti na bilo koju dvodimenzionalnu sliku svjetlosne mikroskopije, bez ljudske interakcije. Najbolje rangirani sudionici izazova uspjeli su u ovom zadatku razviti modele temeljene na dubokom učenju koji su identificirali jezgre stanica na različitim vrstama slika i eksperimentalnim uvjetima bez potrebe za ručnom prilagodbom parametara segmentacije. Ovo predstavlja važan korak prema softverskim alatima za analizu bioslike bez potrebe rekonfiguracije.

Mikroskopija je središnja tehnologija biomedicinskih istraživanja, spomenuta u preko milijun znanstvenih radova indeksiranih na PubMed-u¹. Sve više, dobivene slike analiziraju se kvantitativno. Različite tehnike mikroskopije omogućuju prikupljanje strukturnih i funkcionalnih svojstava sustava biološkog modela, uključujući uzgojene stanice, tkiva i organoide. Kako mikroskopija napreduje u prikupljanju takvih detalja takvih sustava te kako razvoj novih analiza otkriva složenija svojstva živih organizama, potreba za robusnom i jednostavnom uporabom metoda analize mikroskopskih slika postaje esencijalna da bi se odgovorilo na sve širi spektar bioloških pitanja. Mnogi analitički tokovi slika uključuju identifikaciju (segmentiranje) jezgri stanica kao prvi korak za ekstrakciju značajnih bioloških signala. Istraživanja mogu uključivati brojanje stanica, praćenje pokretnih popu-

¹<https://pubmed.ncbi.nlm.nih.gov/>

lacija, lokalizaciju proteina i klasifikaciju fenotipova te afirmaciju liječenja. U svim tim slučajevima jezgra je pouzdana referenca za identifikaciju pojedinačnih stanica na mikroskopskim snimkama. Međutim, odabir strategije za segmentiranje jezgre nije lagan zadatak za nestručne korisnike u regularnim biološkim laboratorijima. Većina postojećih alata za analizu bioslike prilagođenih korisniku identificira jezgre koristeći algoritme klasične segmentacije kao što su metoda praga (eng. *thresholding*), metoda slijeva ili metoda aktivnih kontura. Takve metode je potrebno prilagoditi za svako istraživanje kako bi se uzeli u obzir različiti modaliteti mikroskopije, skale i eksperimentalni uvjeti, što često zahtijeva veliku stručnost za odabir algoritma koji odgovara problemu i prilagodbu njegovih parametara. Čak i za napredne korisnike izbor također može biti zapanjujuće težak, s obzirom da se stotine radova objavljuje svake godine predstavljajući nove metode za segmentaciju stanica i jezgri. Čak i pod kontroliranim eksperimentalnim uvjetima, niti jedan kombinacija parametara ne može pravilno segmentirati sve vrste slika, jer se klasični algoritmi ne mogu prilagoditi heterogenosti bioloških uzoraka i mogu biti osjetljivi na tehničke značajke opreme iz koje su podaci dobiveni. Sve u svemu, zbog potrebnog vremena i ekspertize usporava se tempo istraživanja i sprječava usvajanje novih tehnologija za snimanje. Ovdje istražujemo ideju stvaranja modela za segmentaciju koji može automatski identificirati jezgru stanica u raznolikom skupu obojenih dvodimenzionalnih (2D) svjetlosnih mikroskopskih slika bez ljudske interakcije. Takav model mogao bi omogućiti budućim robotičkim mikroskopima širok spektar bioloških primjena: pronalaženje i brojanje jezgra u slikama u realnom vremenu usprkos varijacijama u podacima (različitim tipovima stanica i veličinama). Jedinstvenim obučeni modelom poboljšalo bi se iskustvo biologa i ubrzalo njihovo istraživanje. Klasični algoritmi za identifikaciju jezgri u mikroskopskim slikama slijede vrlo slične računalne strategije, s različitim parametrima ili konfiguracijama. Cilj nam je bio istražiti mogu li ikakva moderna rješenja, kao što su modeli dubokog učenja velikog kapaciteta, pružiti jedinstveno rješenje bez potrebe za ručnom konfiguracijom.

Biološka segmentacija slike na bazi strojnog učenja već postoji u korisničkom *software-u*, kao što su Ilastik² i ImageJ³, a novija istraživanja potvrđuju korisnost ovog pristupa. Duboko učenje pokazalo je velik potencijal za rješavanje teških problema u analizi staničnih slika, a postoje i modeli neuronskih mreža za segmentaciju slike. Međutim, postojeća rješenja zahtijevaju od korisnika izradu modela koji su prilagođeni svakom eksperimentu, uz dovoljno vremena za pripremu oznaka, učenje modela i konfiguriranje algoritama. Umjesto toga, cilj natjecanja je stvoriti generički, uporabljivi model koji se jednom trenira i može se ponovno koristiti i provoditi na različitim eksperimentima fluorescencijske mikroskopije bez dodatne intervencije korisnika. Cilj je imati slične programske alate za segmentaciju jezgre koji se mogu koristiti s istom lakoćom i robusnošću kao i detektori lica na slikama u prirodnom okruženju, koji rade pod različitim svjetlosnim i scenskim uvjetima bez da

²<https://www.ilastik.org/>

³<https://imagej.nih.gov/ij/>

korisnici moraju trenirati modele ili konfigurirati postavke.

U ovom poglavlju prikazuju se rezultati natjecanja *Data Science Bowla 2018.*, koji je sudionike potaknuo da segmentiraju jezgre na različitim 2D svjetlosnim mikroskopskim snimkama bez potrebe za ručnom interakcijom ili prilagodbom. Natjecanje je sudionicima pružilo podatke za treniranje modela koji su se sastojali od slika koje sadrže jezgre, kao i odgovarajućih oznaka (segmentnih maska za jezgre) i testnih grupa slika za koje su morali generirati segmentacijske modele koristeći protokol dvostupanjske procjene. Važno je napomenuti da se izostavljeni skup podataka za evaluaciju modela sastojao od 15 različitih vrsta slika iz bioloških eksperimenata koji nisu bili prisutni u podacima za učenje, kako bi se realno procijenilo koliko dobro algoritmi funkcioniraju u različitim eksperimentalnim uvjetima. Ovo je prvi put da su modeli segmentacije jezgre izazvani na slijepo generaliziranje — predviđanje na neviđenim biološkim eksperimentima bez djelovanja korisnika ili dodatnog označavanja podataka i učenja.

3.1 Rezultati

Za natjecanje je prikupljen skup podataka sa 37 333 ručno označene jezgre na 841 2D slici iz više od 30 eksperimenata na različitim uzorcima, staničnim linijama, mikroskopskim instrumentima, uvjetima snimanja, operatorima, istraživačkim objektima i protokolima obojenja. Oznake je ručno izradio tim stručnih biologa koji su surađivali, tako što bi svaki biolog označio jednu sliku, tj. stvorio "ciljanu masku" za tu sliku, dok bi preostali pregledali njegove oznake. Taj je skup podataka javno dostupan na stranicama *Broad Bioimage Benchmark Collection*⁴.

Izazov se odvijao tijekom ukupno 3 mjeseca u kojima su sudionici imali pristup podacima za treniranje modela (s ciljnim maskama) i testnim podacima prve faze (bez ciljnih maski). Procjena predviđanja natjecatelja sa izostavljenim maskama prvog kruga ispitivanja odredila je rangiranje na tablici, a u posljednjem tjednu natjecanja objavljeni su podaci za testiranje u drugom krugu, opet s izostavljenim ciljnim maskama kako bi se odredili pobjednici izazova. Ovom evaluacijom u dvije faze željela se procijeniti robusnost modela za segmentiranje novih slika iz novih eksperimenata te procijeniti sposobnost modela za potpuno autonomnu segmentaciju bez interakcije korisnika. Kako bi se spriječila ručna intervencija na slikama, skup za testiranje u drugoj fazi je imao je 3 200 slika s približno 100 000 jezgri koju su trebale biti segmentirane unutar 7 dana, od kojih je mali dio imao popratne ručno definirane ciljne masku i koji se koristio za stvarno ocjenjivanje. Sudionici su postavili maske za segmentaciju na Kaggle-ov poslužitelj⁵, koji je testirao njihove podatke na stvarnim maskama koje nisu bile javno dostupne, te koristeći taj rezultat rangirao

⁴<https://bbbc.broadinstitute.org/BBBC038>

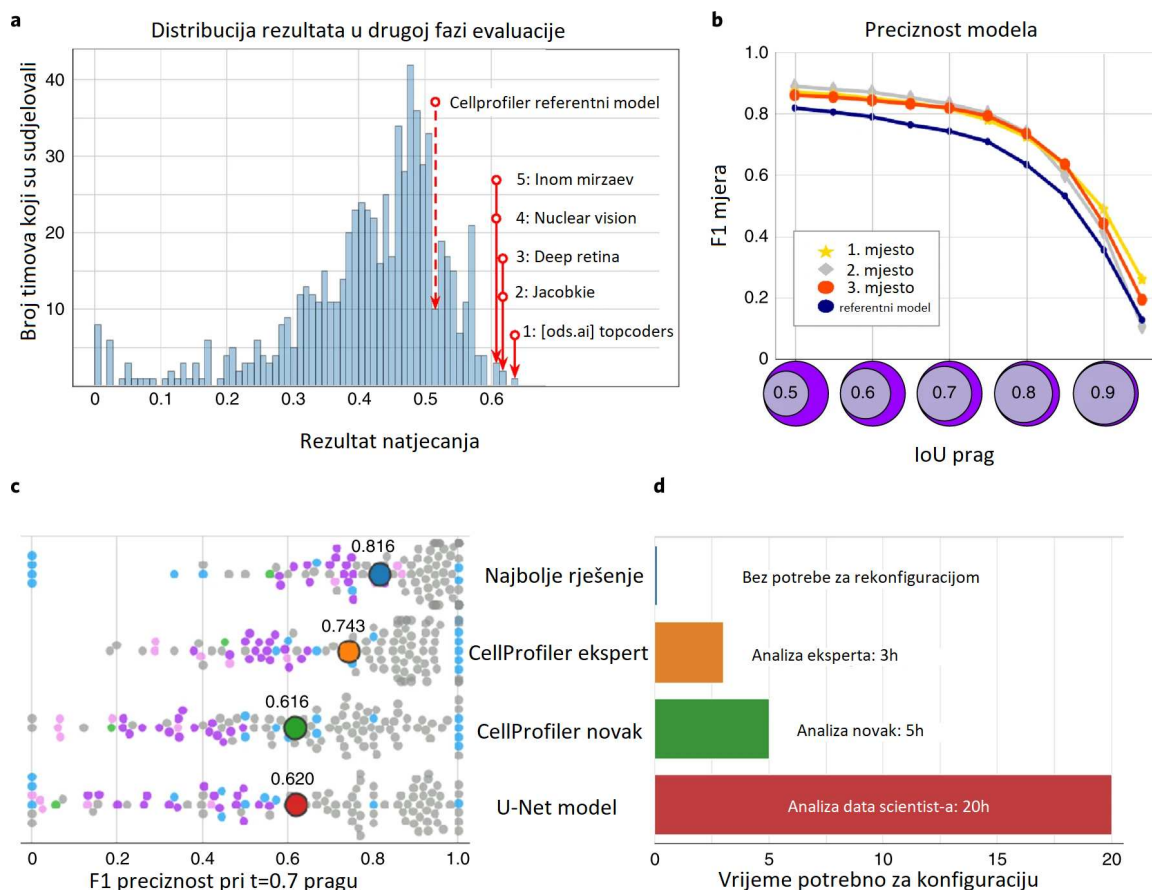
⁵<https://www.kaggle.com/c/data-science-Bowl-2018>

je sudionike. Ukupno 17 929 natjecatelja se prijavilo na natjecanje unutar 3 891 timova u prvoj etapi, a 739 timova ušlo je u drugu etapu natjecanja, natjecajući se za 170 000 američkih dolara u gotovini i nagradama. Ovo natjecanje potaknulo je razvoj novih metoda doprinosima znanstvenika iz područja obrade podataka diljem svijeta koji obično ne rade na mikroskopskim snimkama, ali su ipak uspjeli pridonijeti inovacijama na ovome području. Tri glavna sudionika, među brojnim konkurentima u izazovu, svoja su rješenja učinila javno dostupnim, što će olakšati njihovo usvajanje i prihvaćanje šire znanstvene zajednice.

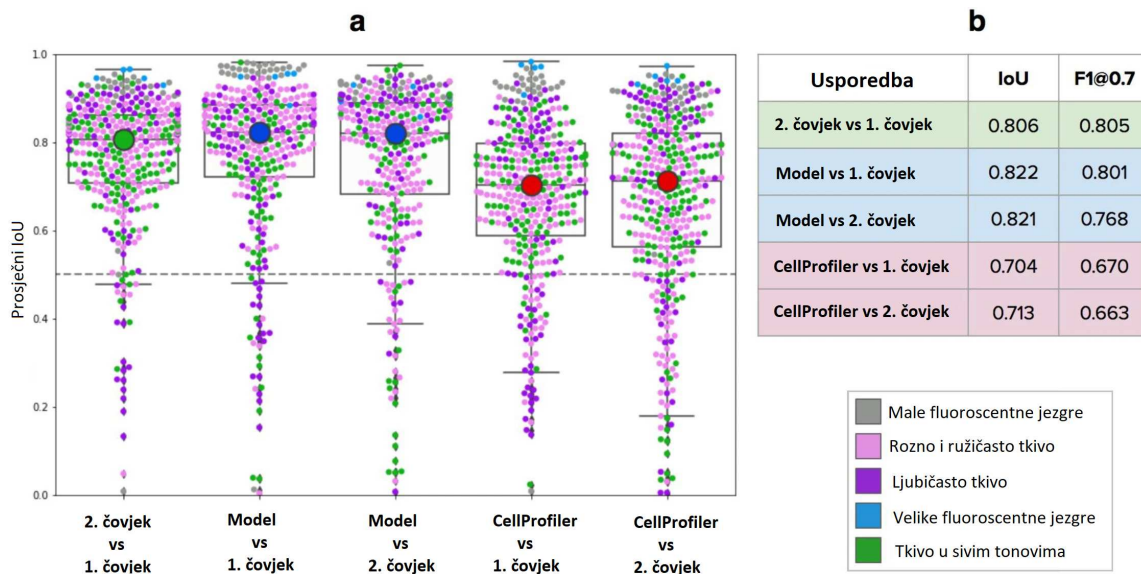
Najbolja rješenja poboljšavaju točnost i uporabljivost segmentacije jezgre

U drugoj fazi ocjenjivanja konkurentima nije bilo dopušteno rekonfigurirati modele za specifičnu vrstu slike. Kao rezultat toga, dobiveni modeli rezultirali su velikim poboljšanjem uporabljivosti u odnosu na trenutnu praksu segmentacije mikroskopskih slika, koja zahtijeva ili odabir algoritama i podešavanje (klasične metode) ili ručno označavanje podataka (za metode strojnog učenja) za različite skupove slika. U usporedbi s referentnom segmentacijom dobivenom klasičnim tehnikama obrade slike koja je bila prilagođena svakoj vrsti izostavljenih skupova podataka, koristeći minimalnu intervenciju korisnika, ustanovilo se da 85 algoritama kandidata daje bolju točnost (slika 3.1a) od referentnog modela koji ćemo kasnije opisati. Posebno, tri najbolja rješenja su uvelike nadmašila te minimalno podešene klasične algoritme, što je dovelo do boljih segmentacija na svim pragovima pokrivenosti (slika 3.1b).

Točnost segmentiranja je ocijenjena pomoću uobičajenih već spomenutih metrika u istraživanju računalnog vida za segmentaciju objekta. Važno je istaknuti da je za segmentiranje klasičnim algoritmima za obradu slika, za razliku od metoda u izazovu, potrebna manualna konfiguracija. Ne postoje klasični algoritmi za koje bi se moglo tvrditi da daju razumne rezultate na 15 različitih skupova slika bez intervencije korisnika. Nasuprot tome, znanstveni novak bez prethodnog znanja na području biološke analize je imao znatno lošije rezultate od eksperta na tom području, a i najboljih modela dubokog učenja (slika 3.1c).



Slika 3.1: **Točnost i uporabljivost segmentacije.** **a:** histogram prikazuje natjecateljske timove ($n = 739$) koji su ušli u drugu fazu natjecanja. Prvih pet natjecatelja je označeno, kao i referentna segmentacija koju je modelirao stručni analitičar koristeći CellProfiler. **b:** točnost triju najboljih rješenja izmjerena pomoću F_1 mjere na višestrukim pragovima IoU-a. Os x na histogramu **a** (Rezultat natjecanja) je u korelaciji s područjem ispod krivulje rezultata F_1 razmatrajući različite IoU pragove. Prva tri modela imala su slične performanse s blagim razlikama na repovima krivulja. **c:** distribucija točnosti u drugoj fazi procjene za najbolji model i tri referentna rješenja. Distribucija F_1 rezultata na jednom IoU pragu ($t = 0.7$) pokazuje točke ($n = 106$) gdje svaka predstavlja točnost segmentacije jedne slike u skupu od 106 označenih slika druge faze procjene. Boja točke odgovara skupini slika definiranih za evaluaciju (slika 3.4). Prosjek distribucije označen je većom točkom povezanom odgovarajućom prosječnom točnošću. **d:** procijenjeno vrijeme potrebno za konfiguriranje modela segmentacije. Izvor: [9].



Slika 3.2: Estimacija varijacije oznaka promatrača (subjektivna pristranost). **a:** svaka mala točka na grafu odgovara jednom objektu. Boja malih točaka odgovara vrsti slike iz koje objekt dolazi (legenda u donjem desnom kutu). Os y prikazuje prosječan IoU rezultat između uspoređenih slika, dok os x prikazuje parove ispitanika ili modela koji se uspoređuju. Velike točke su medijan svih rezultata. Jezgre iz slika tkiva u sivim tonovima teže je segmentirati pomoću računalnih metoda, dok se ljudski analitičari općenito slažu s njihovim maskama. Ljubičaste točke češće pokazuju neslaganje, neovisno o uspoređivanju parova. **b:** parove se uspoređuju s mjerenjima slaganja u preklapanju (IoU) i točnosti (F_1 rezultat pri $t = 0.7$ IoU). Ljudski anotatori (zeleni red) postignu visoku razinu slaganja u oznakama, ali se najbolji model (plavi redovi) bolje slaže u oznakama s oba čovjeka nego što se oni međusobno slažu. Međutim, model ima nešto više neslaganja s ljudima glede točnosti, što znači da model češće propusti nekoliko predmeta nego ljudi. CellProfiler referenca pokazuje izrazito neslaganje s ljudima i u oznakama i točnosti. Izvor: [9].

Novak je uložio 5 sati dodatnog rada, dok je ekspert uložio 3 sata rada kako bi postigao odgovarajuće rezultate segmentacije. Uz korisničko sučelje koje omogućava *upload* podataka, najboljem modelu natjecanja nije bilo potrebno vrijeme za rekonfiguriranje. Klasične metode ispitane su prvo organiziranjem slika u pet grupa vizualnim pregledom, zatim su kreirani analitički mehanizmi u računalnom programu otvorenog koda *CellProfiler*⁶. Ti mehanizmi predstavljaju široko usvojene tehnike nedavno istražene u literaturi segmentacije mikroskopskih slika [17], iako su to vjerojatno suboptimalna rješenja, jer tehnike nisu u potpunosti optimizirane za svaki od 15 skupova slika zasebno kako bi se spriječila

⁶<https://cellprofiler.org/>

prenaučenost i kako bi bili u skladu s uvjetima zadanim na rekonfiguraciju. Osim referentnih segmenata klasičnim tehnikama, autori članka [9] su odabrali model dubokog učenja koji su modelirali na istim razdvojenim podacima. Opisali su i trenirali model temeljen na U-Net-u, već opisanom modelu neuronske mreže koji je izrazito popularan za rješavanje problema segmentacije mikroskopskih slika, uključujući segmentaciju jezgre. Povukli su paralelu sa najboljim rješenjima natjecanja koristeći ga kao referentnu točku pri mjerenju performansi. Kapacitet učenja regularnog U-Net modela nije bio dovoljan da nauči eksperimentalne varijacije u izazovu, stoga je skup podataka podijeljen na pet podskupova i obučeno je pet modela kako bi se smanjila varijanca, kao što je to slučaj kod klasičnih algoritama, te su primijenjeni postupci za poboljšanje i obradu slika specifični za svaki skup podataka. Rezultati pokazuju da čak i nakon trošenja oko 20 sati praktičnog vremena za konfiguraciju (ne uključujući vrijeme za razvoj i trening modela), takav model nije bio konkurentan u odnosu na najbolja rješenja (slika 3.1d). Ovom rezultatu pridonijelo je nekoliko čimbenika:

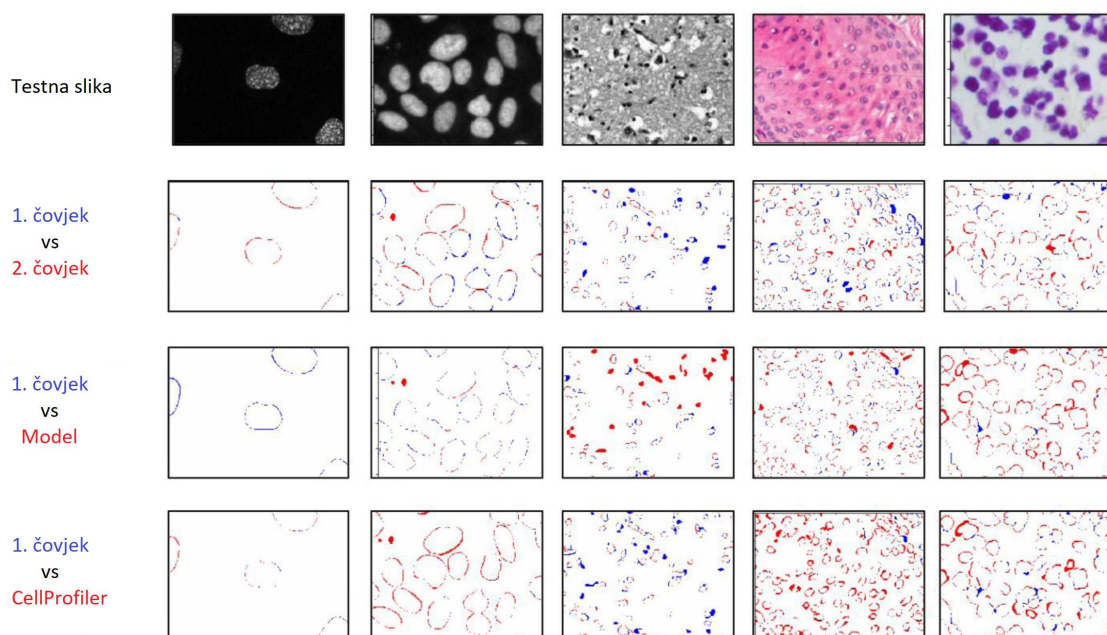
- ograničeni kapacitet učenja odabrane U-Net mreže u odnosu na najbolje modele
- smanjen broj uzoraka za učenje nakon podijele u pet skupina
- eksperimentalna varijabilnost ispitnih podataka

Konačno, nakon što su od analitičara dodatno zatražili da stvore ciljne maske za podskup slika u skupu za testiranje, uočena je varijabilnost među njihovim maskama (slika 3.2 i 3.3).

Zanimljivo je da se model pobjednika natjecanja više podudara u rubnim oznakama sa svim ostalim timova, nego što se itko od njih podudara međusobno (slika 3.2). To upućuje na to da je model naučio glatke granice (slika 3.8) koje su bile blizu ruba jezgre, dok "ručne" oznake mogu biti pristrane zbog subjektivnog šuma. Svojstva U-Net modela također su bila sličnija ljudima nego klasičnim algoritmima u smislu točnosti segmentacije. Iako ovaj rezultat nije temeljito istražen, on sugerira da bi najbolja rješenja mogla doseći performanse nalik ljudskom anotatoru sa sličnim stopama pogreške.

3.2 Skup podataka

Podaci su donirani od strane brojnih laboratorija koji su proučavali različite aspekte stanične biologije. Cilj je bio prikupiti što više neovisnih bioloških eksperimenata kako bi se stvorili resursi koji sadrže dovoljno tehničke i biološke varijabilnosti za obučavanje generičkih modela segmentacije jezgri. Ukupno, skup podataka sadržavao je slike iz više od 30 različitih bioloških eksperimenata, koje su podijeljene u 16 eksperimenata za trening (670 slika) i procjenu u prvom stupnju (65 slika) te točno 15 eksperimenata za procjenu u

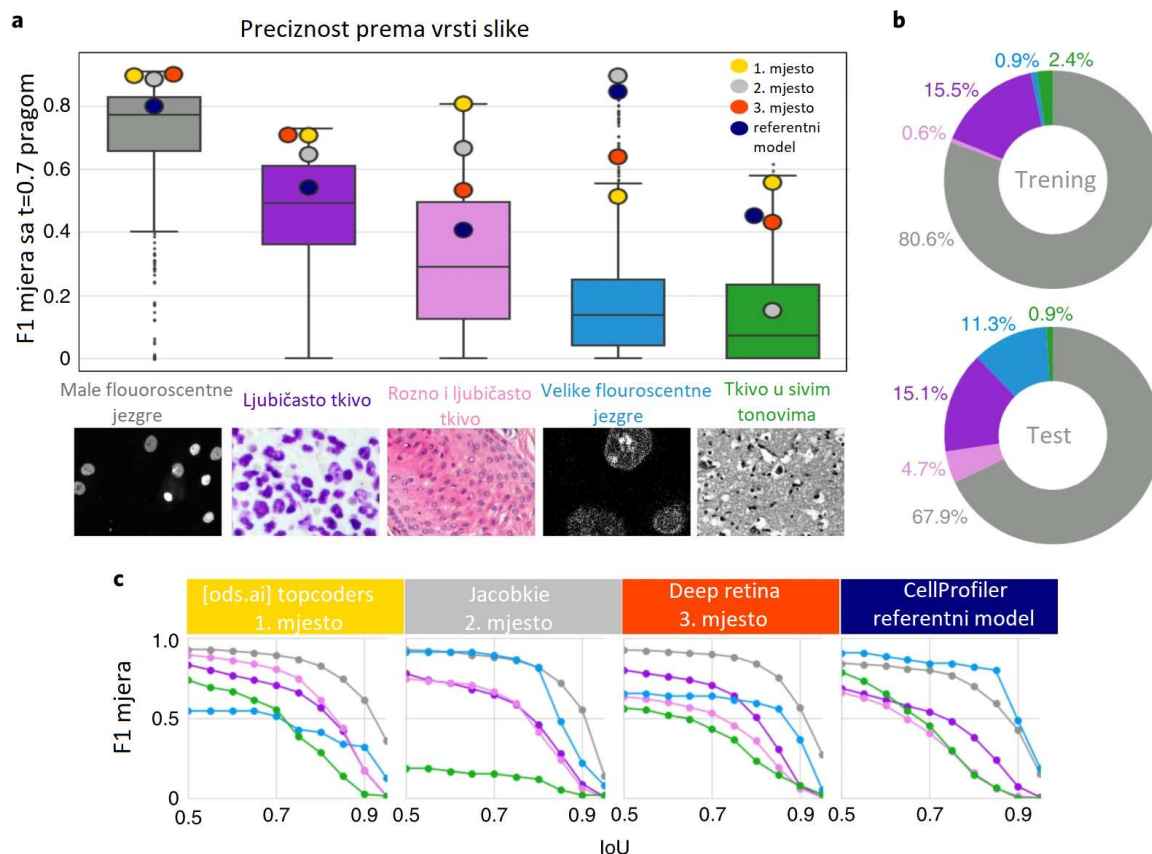


Slika 3.3: **Razlike u oznakama pet odabranih slika.** Gornji redak: izvorne slike. Sljedeći redci: Usporedba segmentacije između dva subjekta ili metoda, jedan označen plavom, drugi crvenom bojom. Plavi i crveni objekti ili strukture označavaju oznake koje je predao odgovarajući promatrač, a koje ne odgovaraju objektu ili oznakama drugog promatrača. U idealnom slučaju, graf bi trebao biti potpuno bijel, što bi značilo da su oznake savršeno postavljene. Izvor: [9].

drugom stupnju (106 slika). Najzahtjevniji aspekt natjecanja bio je taj što se izostavljeni skupu za testiranje sastojao od mikroskopskih slika iz 15 različitih bioloških eksperimenata, uključujući različite 2D tipove svjetlosne mikroskopije, prikupljene pomoću različite opreme i iz različitih bioloških uvjeta.

To je u suprotnosti s prethodnim istraživanjima koja optimiziraju metode segmentacije jezgre pojedinačno za svaki tip slike. Vizualno gledano, autori su identificirali pet grupa slika koje se sastoje od vrlo različitih jezgri, uključujući dvije bitne vrste svjetlosne mikroskopije (slika 3.4): mikroskopija fluorescencije uglavnom uzgojenih stanica i mikroskopija svjetlosnog polja uzoraka obojenog tkiva.

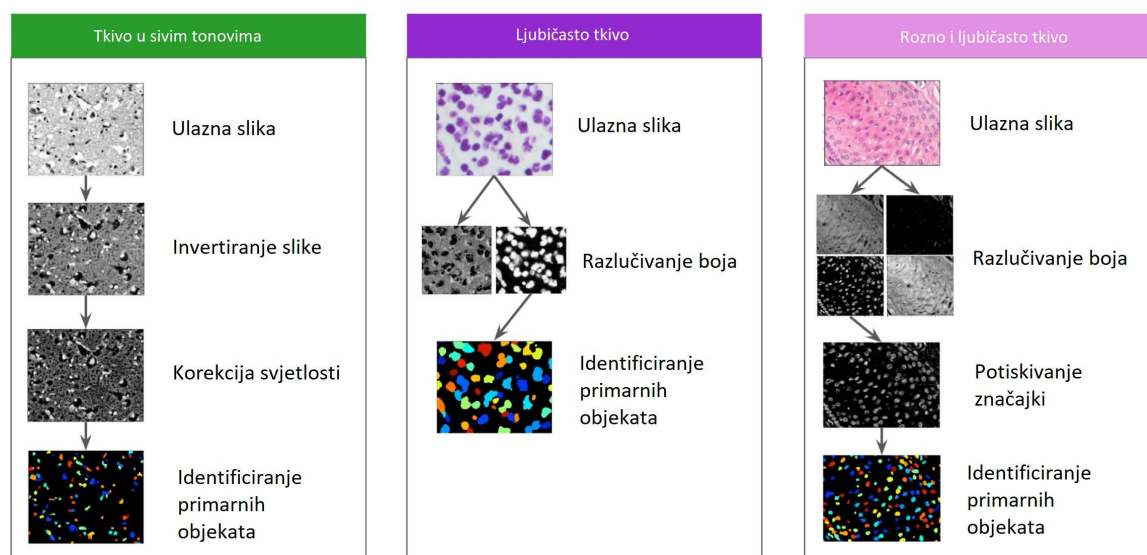
Uzorci tkiva tipično su zahtjevniji zadatak obrade slike zbog nepravilnih pojava jezgri i njihova prenapučenog izgleda. Male fluorescentne jezgre vrlo su česte u biomedicinskim istraživanjima i najzastupljenije su u skupovima za učenje i testiranje. Cijeli skup podataka obuhvatio je 31 različiti eksperiment (16 za trening i evaluaciju u prvoj fazi, 15 za evaluaciju u drugoj fazi), koji predstavljaju 22 vrste stanica, 15 različitih rezolucija slike i pet skupina vizualno sličnih slika, što je rezultiralo već spomenutim brojem od 841 slike



Slika 3.4: **Preformanse modela na različitim vrstama slika.** **a:** primjeri različitih vrsta slika prikazani na dnu, a na grafikonu je prikazan raspon F_1 rezultata za sve timove koji su ušli u drugu fazu natjecanja. F_1 rezultat je mjereno s obzirom na prag $t = 0.7$. Na grafu srednja crta predstavlja medijan, granice obojenog okvira predstavljaju gornji i donji kvartil. Obojene točke predstavljaju najbolja tri tima. **b:** distribucija različitih vrsta slika prikazana je bojom kodirana kao u **a**. Najbolji konkurenti segmentirali su sve vrste slika s visokom točnošću unatoč neravnoteži primjera u skupu za učenje. **c:** detalji rezultata točnosti s obzirom na vrstu slika i pragove pokrivenosti objekata (IoU). Os x prikazuje pragove IoU-a, a os y predstavlja točnost izmjerenu pomoću F_1 mjere. Za svakog sudionika pograf prikazuje pet krivulja koje opisuju točnosti segmentacije na različitim pragovima pokrivenosti objekata. Izvor: [9].

i 37 333 ručno označenih jezgra. Najbolji timovi su se istaknuli izradom modela koji dobro generalizirali različite vrste slika i eksperimentalnih varijacija (slika 3.4a) i unatoč vrlo neuravnoteženom skupu podataka (slika 3.4b). Pristranost skupa podataka može modele strojnog učenja dovesti u zabludu. Manje zastupljene vrste slike doista su bile izazov za segmentaciju za prosječne sudionike (slika 3.4a). U najvećem podskupu slika, koji sadrži 80% primjera za učenje, najbolji sudionici dostigli su maksimalnu točnost od 0.9 na skupu za testiranje, a s najmanjom skupinom, koja sadrži 0.6% primjera, dostigli su maksimalnu točnost od 0.55. U svim slučajevima njihova je izvedba s velikom razlikom premašila referentnu segmentaciju CellProfiler-a, kao i prosječnog sudionika.

3.3 Referentni model



Slika 3.5: Grafovi izračunavanja (cjevovodi) za tri vrste slika. Svi cjevovodi završavaju s IPO modulom (Identificiraj primarne objekte) koji koristi metodu praga, transformacije udaljenosti i metodu slijeva na slici u sivim tonovima proizvedenoj prethodnim koracima. Moduli prije IPO-a imaju za cilj pretvoriti ulazne slike u matricu sivih tonova prikladnu za segmentiranje. Izvor: [9].

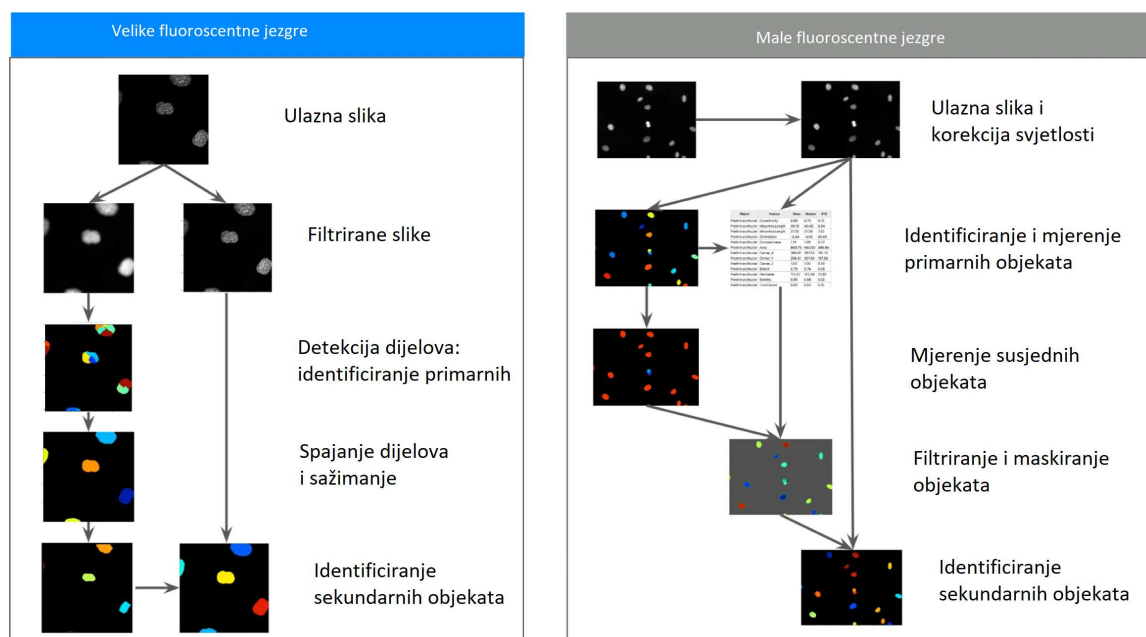
Cilj modeliranja referentnog modela je bila ocjena doprinosa metoda segmentacije jezgre koje su predložili sudionici izazova, s obzirom da su te metode radile na različitim eksperimentima bez intervencije korisnika. Najprikladnija baza bila je postojeća strategija koja bi se mogla primijeniti na bilo koju sliku i dati točne rezultate bez interakcije ljudi. S obzirom da takva metoda ne postoji, korišten je pristup koji je koristio što ma-

nje ljudskog vremena. Segmente dobivene ovim pristupom ne nazivamo baznim segmentima, već referentnim segmentima. Odabran je CellProfiler v.3.1.5 kao alat za stvaranje referentnih segmenata s obzirom na njegovu fleksibilnost u konfiguraciji robusnih grafova izračunavanja (cjevovoda, eng. *pipeline*, koji simboliziraju tok izračunavanja) na temelju provjerenih algoritama, uz ulaganje minimalnog vremena. CellProfiler je moćan alat otvorenog koda za mikroskopsku analizu slika koji uključuje niz algoritama temeljne obrade slika organiziranih u module. Algoritmi se mogu organizirati u cjevovodu koji uzima slike kao ulaze i proizvodi različite vrste izlaza. Važno je napomenuti da je cjevovod definiran sučeljem prilagođenim korisniku i da može pokrenuti složene operacije bez potrebe za pisanjem jednog retka koda. Ta svojstva činila su CellProfiler dobrom referencom za modele u konkurenciji, jer je cilj izazova bio istražiti generičke metode segmentacije jezgri uz minimalnu interakciju korisnika. Modelirano je pet prilagođenih cjevovoda, jedan za svaku vrstu slike u skupu podataka za testiranje, s ciljem ocjenjivanja klasičnih algoritama primijenjenih u CellProfileru. Očekivano je će pojedini cjevovodi klasične segmentacije loše funkcionirati na različitim slikama sadržanim u skupu za testiranje, stoga su prilagođeni za svaku skupinu slika. Takav pristup imao je prednost pred modelima dubokog učenja koji su testirani tijekom natjecanja jer su skupovi slika ručno organizirani i obrađeni posebnim metodama prema vrsti. S druge strane, očekivalo se da modeli dubokog učenja nemaju "ručnu" konfiguraciju od strane korisnika. Cjevovodi su projektirani u tri glavna uzastopna koraka:

1. prethodnom obradom slike u matricu sivog tona, gdje su jezgre opažene kao relativno glatki bijeli oblici na crnoj pozadini
2. segmentacijom slike sivog tona pomoću metode praga (eng. *thresholding*), transformacije udaljenosti (eng. *distance transforms*) i metode slijeva (eng. *watershed*) na temelju približne očekivane veličine jezgre
3. revizijom segmentacije korištenjem metode slijeva temeljene na presjeku prethodne segmentacije i dodatnih veličina jezgri

Koraci 2. i 3. temeljili su se na CellProfiler modulima identifikacije primarnih i sekundarnih objekata. Slike 3.5 i 3.6 ilustriraju odgovarajuće računalne grafove za svaki od pet projektiranih cjevovoda.

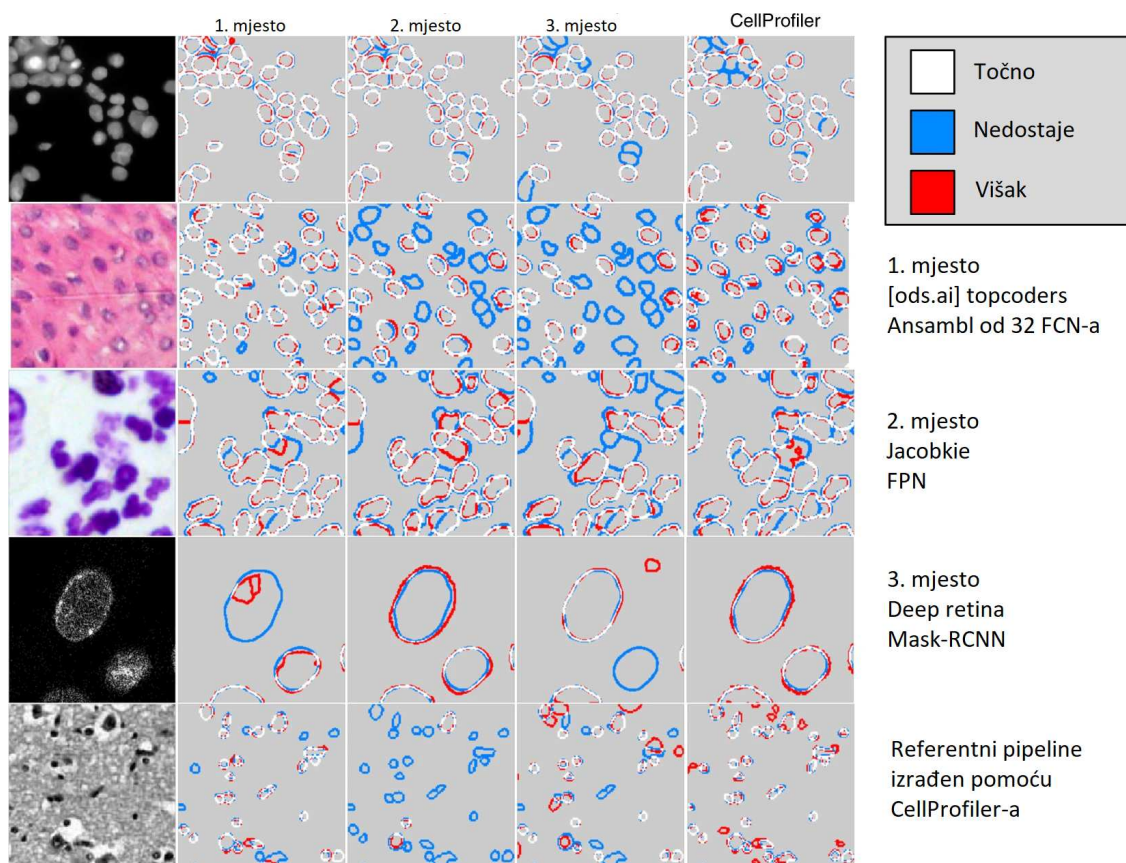
Važno je napomenuti da parametri algoritma nisu optimizirani za svaki eksperiment posebno, već su za svaku grupu metode prilagođene kako bi model bio što generičniji i automatičniji. U tom smislu, rješenje može biti suboptimalno, ali može poslužiti kao reprezentativni model analizu slike. Možda se neke pogreške mogu ispraviti podešavanjem parametara za pojedine slike ili pokuse, ali s obzirom da je cilj minimizirati ručni ručne konfiguracije, taj dio se neće razmatrati.



Slika 3.6: Grafovi izračunavanja (cjevovodi) za fluorescentne vrste slika. Oba cjevovoda koriste modul za identifikaciju primarnih objekata (IPO) i modul za identifikaciju sekundarnih objekata (ISO). Oba modula bazirana su na *thresholding* i *watershed* metodama. Potrebni su zbog velike varijabilnosti eksperimenata i fenotipova jezgri prisutnih u podacima. Drugi moduli imaju za cilj smanjiti i filtrirati šum kako bi se slika pripremila za segmentaciju. Izvor: [9].

3.4 Najučinkovitija rješenja

Prema rezultatima natjecanja i drugim parametrima pomoću kojih su rezultati analizirani, rješenja triju najboljih timova bila su znatno bolja od referentnih segmentacija (slika 3.7). Rezultat natjecanja bila je agregirana metrika koja je uzela u obzir višestruke čimbenike kvalitete segmentacije, uključujući preciznost, odziv i Jaccard-ov indeks, a takva metrika bi se mogla razdijeliti na više slučajeva kako bi se razumjela učinkovitost i pogreške. Prvo, ocijenjene su performanse sa Jaccard-ovim indeksom na jednom objektu kako bi se interpretirale razlike u točnosti između modela. Kada je izabran prag $t = 0.5$ (uobičajen u prethodnim radovima), najuspješniji model postigao je $F_1 = 0.889$, u usporedbi s 0.819 referentnog CellProfiler modela. Treba uzeti u obzir da su ti rezultati bili prosjek svih slika iz 15 različitih eksperimenata u drugoj fazi procjene. Kada se razmotre fluorescentne slike sa samo malim jezgrama, F_1 točnost modela najboljeg rješenja dosegla je vrijednost od 0.932, dok je CellProfiler postigao vrijednost od 0.844. Primjenom praga od $t = 0.7$ (slika 3.4a), zahtjeva se veća minimalna pokrivenosti objekata od modela. Primijetili smo



Slika 3.7: Mape segmentacije različitih vrsta slika dobivene od tri najbolja sudionika i CellProfiler reference. Mape segmentacije pokazuju oznake po pikselima između ciljnih maski segmentacije i predviđenih segmentacija od strane modela. Ako se maske pravilno poravnaju, rubni pikseli su bijele boje. Ako nedostaje ciljna maska ili njezin dio, rubni pikseli su obojeni plavom bojom. Ako je predviđena segmentacija u području bez stvarnog objekta, rubni pikseli su crveni. Izvor: [9].

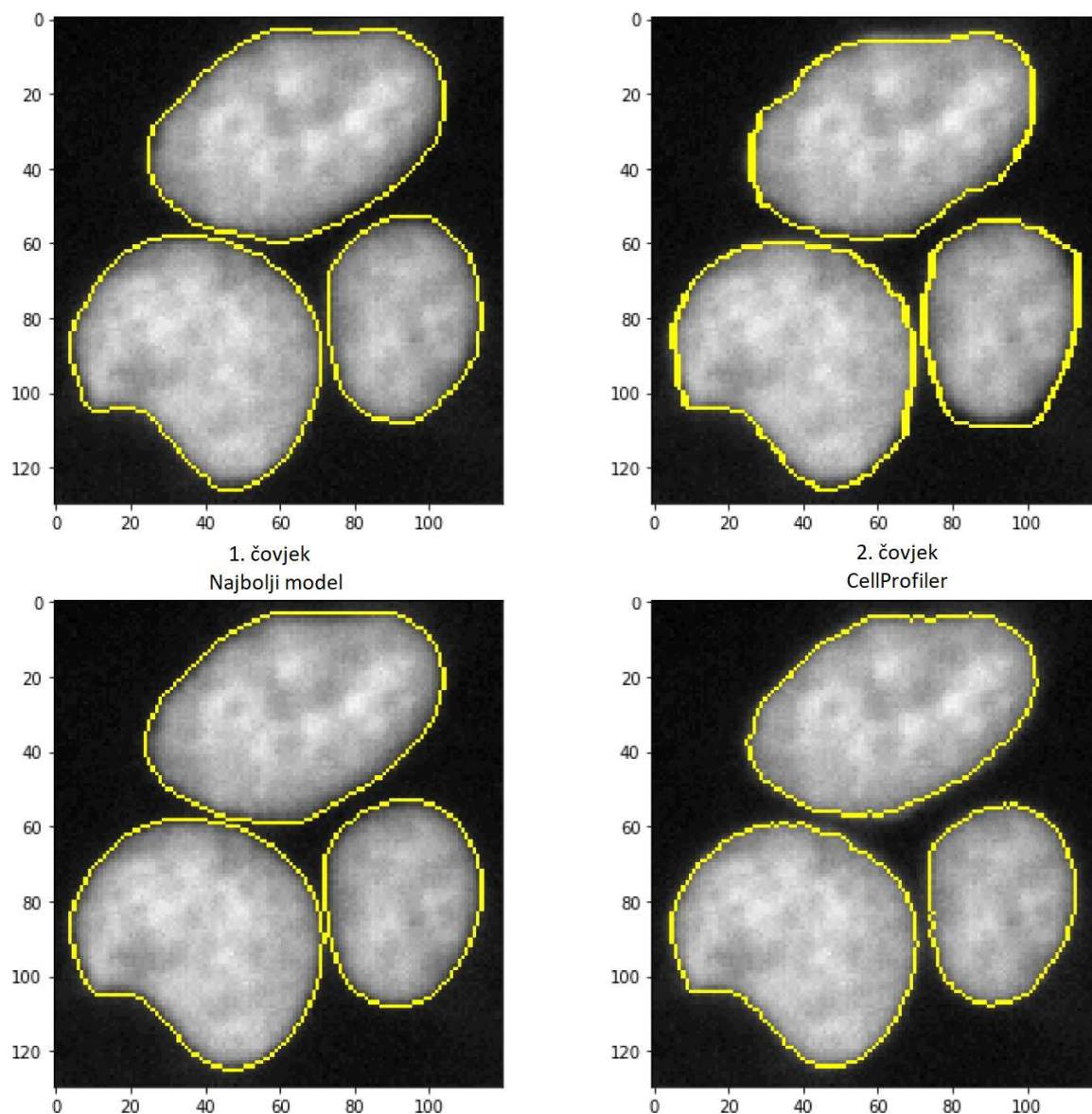
da su najbolja tri modela nadmašila referentnu vrijednost CellProfiler-a za tri vrste slike (male fluorescentne jezgre, ljubičasto tkivo, te slike roznog i ljubičastog tkiva). Za ostale dvije vrste slika (velike fluorescentne i tkivo u sivim tonovima), svi osim jednog modela bili su lošiji od referentnog CellProfiler-a, prvenstveno zbog ograničenog broja primjera u skupu za učenje (slika 3.4a). Općenito gledano, prva tri modela postigla su slične ukupne performanse, ali su pokazala drugačije ponašanje i pogreške. Rezultati točnosti mogu se razvrstati prema slikama po višestrukim pragovima pokrivenosti objekata (slika 3.4c), što nam omogućuje da utvrdimo prednosti i slabosti svake strategije. Primjerice, rješenje tima na drugom mjestu najbolje je za velike fluorescentne jezgre (plava linija), ali slabije za tkivo u sivim tonovima (zelena linija). Nakon analize ponašanja prva tri modela u odnosu na referentnu vrijednost CellProfiler-a (slika 3.9), uočavaju se smanjene stope pogrešaka modela koji su propustili segmentirati manje objekata, te su ti modeli uspješno razdvojili spojene jezgre te poboljšali preciznosti i odziv. Sva ta opažanja podupiru ideju da najbolja tri rješenja mogu segmentirati 2D obojene jezgre u realnom vremenu bez konfiguracije.

Pozadina algoritama

Većina sudionika koristila je duboke konvolucijske neuronske mreže (CNN), popularne za rješavanje zadataka računalnog vida, kao i različitih problema povezanih sa mikroskopijom i patologijom. Za segmentaciju slike može se koristiti široka paleta CNN arhitektura, pa su tako i sudionici dizajnirali kreativna rješenja za poboljšanje točnosti segmentacije. Zanimljivo je da su tri najbolja sudionika koristila vrlo različita rješenja: ansambl U-Net modela, potpunu konvolucijsku piramidalnu mrežu značajki (FPN) i Mask-RCNN model. Njihova učinkovitost i glavne karakteristike svakog modela opisane su u nastavku. Na slici 3.7 prikazani su primjeri segmenata dobivenih tim modelima s referentnom segmentacijom koju je dobio CellProfiler.

Najbolje rješenje

Tim [*ods.ai*] *topcoders*, čiji su članovi *A. Buslaev*, *V. Durnov* i *S. Seferbekov*, predstavili su visoko optimiziran model sa sofisticiranom augmentacijom podataka i naknadnom obradom podataka. Njihovo rješenje se sastojalo od ansambla mreža. Koordinacija svih tih elemenata u uspješnom rješenju bila je veliko postignuće jer modeli s većim kapacitetom učenja mogu biti prenaučeni i na kraju neuspješni u radu s novim slikama. Umjesto toga, rješenje je dobro generalizirano u odnosu na 15 skupova slika. U smislu računalnih zahtjeva, to je bilo najzahtjevnije rješenje, jer jednu sliku trebaju obraditi 32 različite neuronske mreže koristeći grafičke kartice (GPU-eve). Osim toga, postprocesni koraci trebaju provjeriti i kombinirati predviđene objekte iz sva 32 izlaza. Sve u svemu, ovaj sustav je bio najtočniji, iako po cijenu visoke izračunljivosti i složenosti. Sustav je bio zasnovan



Slika 3.8: **Razlike oznake granica jezgri najboljeg modela, ljudskih anotatora i referentnog modela.** Oznake ljudskih anotatora unose subjektivni šum u rubnim pikselima. Najbolji model je naučio proizvoditi glatke krivulje koje su blizu ruba objekta. Ovaj primjer pokazuje kako se granice vrhunskog modela mogu češće slagati s oba anotatora, dok se oni možda ne slažu oko sitnih detalja. Segmenti CellProfiler-a, proizvedeni pomoću metode slijeva, nisu tako glatki na rubovima. Izvor: [9].

na strategiji ansambla od osam potpunih konvolucijskih mreža (FPN). Za svaki od njih trenirana su četiri replicirana modela, što je rezultiralo sa ukupno 32 naučene neuronske mreže u konačnom rješenju. Svih osam osnovnih arhitektura slijedilo je načelo kodiranja-dekodiranja za obradu ulazne slike i generiranje segmentacijske mape na izlazu. Šest od osam bazičnih arhitektura koristilo je dekodere nalik na U-Net, a druga dva su koristila shemu FPN dekodiranja. Među enkoderima korištene su rezidualne mreže, DPN mreže (eng. *Dual Path Network*, vrsta mreže koja predstavlja sintezu gustih i rezidualnih mreža, koristeći najbolje od svake) i inception-Resnet (rezidualne mreže koje kombiniraju različite veličine jezgri kako bi postigle skalabilnost u detekciji objekata). Tim je konstatirao kako je pravilno modeliranje ciljnih maski za obuku U-Net modela ili FPN modela kritično za postizanje najboljih rezultata. U konačnom su rješenju ugradili pristup temeljen na maskama jezgri koje su sadržavale umjetno-generirane granice. Zatim je zadatak segmentacije bio klasificirati piksele u tri tipa: pozadinske piksele, unutrašnjost stanica i rubne piksele. Najbolji učinak ostvaren je kada su rubni pikseli označeni samo između stanica koje se dodiruju. U prethodnim radovima također je uzeto u obzir modeliranje ciljnih maski na sličan način, što je jednako pristupu semantičke segmentacije za odvojene slučajeve.

Kombinacija izlaza 32 mreže provedena je u tri koraka:

1. agregiranje predviđenih maski koristeći srednju vrijednost svih
2. rangiranja za filtriranje loših predviđanja
3. primjena metode slijeva kako bi se redefinirale granice

Model rangiranja drugog koraka koristio je klasične morfološke značajke ekstrahirane iz svakog kandidata za jezgru. Te su značajke korištene za obučavanje regresijskog modela (stabla pojačana gradijentom) koji je naučio predviđati IoU rezultate iz evaluacijskih primjera. Tijekom ispitivanja, svaki potencijalni objekt je naknadno obrađen na ovaj način kako bi se procijenilo koliko je usklađen s potencijalno stvarnim objektom. Ova strategija omogućila je "pogađanje" mnogih segmentacijskih maski i njihovo rangiranje kao obećavajućih, što je bilo korisno za identificiranje i uklanjanje lažnih predviđanja. Tim se usredotočio na sprječavanje prenaučivosti dvjema strategijama:

1. korištenjem neuronskih mreža pretreniranih na popularnoj bazi podataka ImageNet kao enkodere značajki za svih osam arhitektura
2. augmentacijom podataka kako bi se što efikasnije iskoristili primjeri za učenje

Ukupno 24 augmentacijske strategije, uključujući inverziju boja i kopiranje objekata, korištene su za obuku svih modela. Tim je također koristio dodatne mikroskopske slike iz javno dostupnih baza podataka kako bi proširio skup primjera obuke, uključujući WikiMedia slike, koje su ručno obilježene.

Drugo najbolje rješenje

M. Jiang (tim *Jacobkie*) predstavila je precizno i brzo rješenje, a za obradu novih slika korištena je samo jedna neuronska mreža. Njezino rješenje uvelo je nekoliko inovacija koje se mogu usvojiti u drugim modelima, kao što je funkcija gubitka koja penalizira pogreške uzimajući u obzir veličinu objekta, korištenje mapa udaljenosti umjesto binarnih maski kao cilja za učenje, pretreniranje na slikama u prirodnom okruženju i skupovima podataka za detekciju objekata.

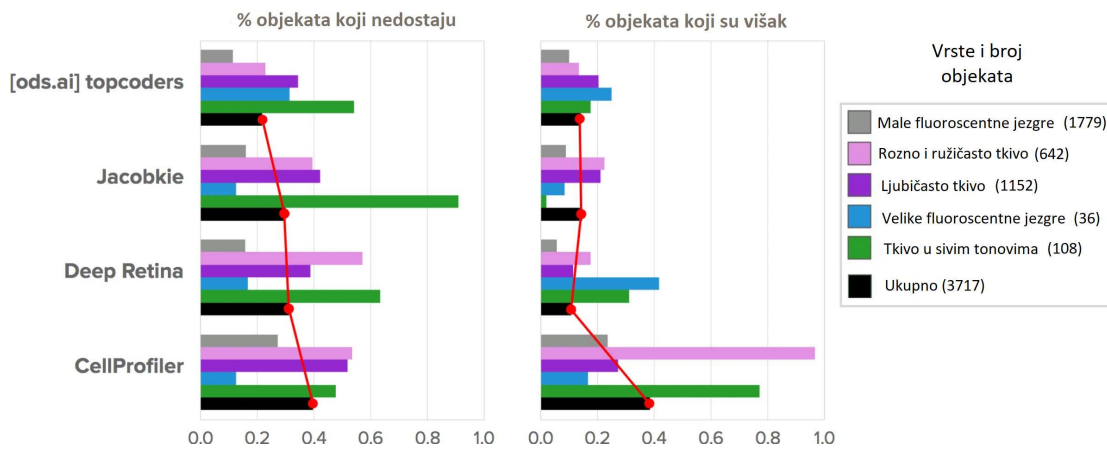
Sustav je sačinjavala jedna neuronska mrežna temeljena na FPN arhitekturi. Rješenje je uvelo dva prilagođena izlazna sloja, od kojih svaki proizvodi maske relativnog položaja koje procjenjuju udaljenost svake jezgre do njenih granica u četiri smjera (vertikalni, horizontalni, 45 stupnjeva i 135 stupnjeva). Maske relativnog položaja slične su „deltama”, tj. udaljenostima piksela s obzirom na referentne točke u određenim regijama. Važno je napomenuti da su koordinatne mape grubo izračunate za svaki piksel u unutrašnjosti jezgre, dok su za piksele u pozadini postavljeni na nulu. Također, maske relativnog položaja su polije obrađene i transformirane u granice, redefinirane pomoću metode slijeva i rangirane prema konzistentnosti između lokalnih i globalnih rezultata kako bi se odabrao konačni skup maski koje se ne preklapaju. Okosnica FPN-a u ovom rješenju je pretreniranje na ImageNet i COCO⁷ skupovima podataka primjenom Matterport implementacije maska-RCNN⁸ programskog okvira. Dva izlazna sloja obučavana su korištenjem okvira za više zadataka. Uvedena je nova funkcija gubitka kako bi se kaznile pogrešne instance prema veličini predmeta, balansirajući doprinos grešaka manjih objekata u odnosu na velike. Tijekom treninga i ispitivanja primijenjene su različite tehnike augmentacije podataka i nisu korišteni vanjski podaci. Povećanje podataka za vrijeme ispitivanja sastojalo se od predviđanja transformiranih verzija slika za testiranje (kao što su skaliranje ili rotacija) i zatim integracije tih predviđanja u jedan izlaz.

Treće najbolje rješenje

A. Lopez-Urrutia (tim *Deep Retina*) predstavio je rješenje temeljeno na jednoj neuronskoj mreži koja je umjesto potpunog konvolucijskog pristupa obrađivala područja od interesa. Baza modela poznat je kao Mask-RCNN, popularna arhitektura za detekciju objekata i segmentaciju instance na slikama u prirodnom okruženju. Uz rezultat, jednostavnost rješenja je ono što se ističe, jer postoje razne implementacije takvog rješenja koje su se mogle prilagoditi problemu segmentacije stanice prekvalifikacijom izlaznih slojeva s pravim podacima. Osim toga, model Mask-RCNN je aktivno istraživano potpodručje računalnog vida, te se inovacije lako mogu primijeniti problemu segmentacije jezgre.

⁷<https://cocodataset.org>

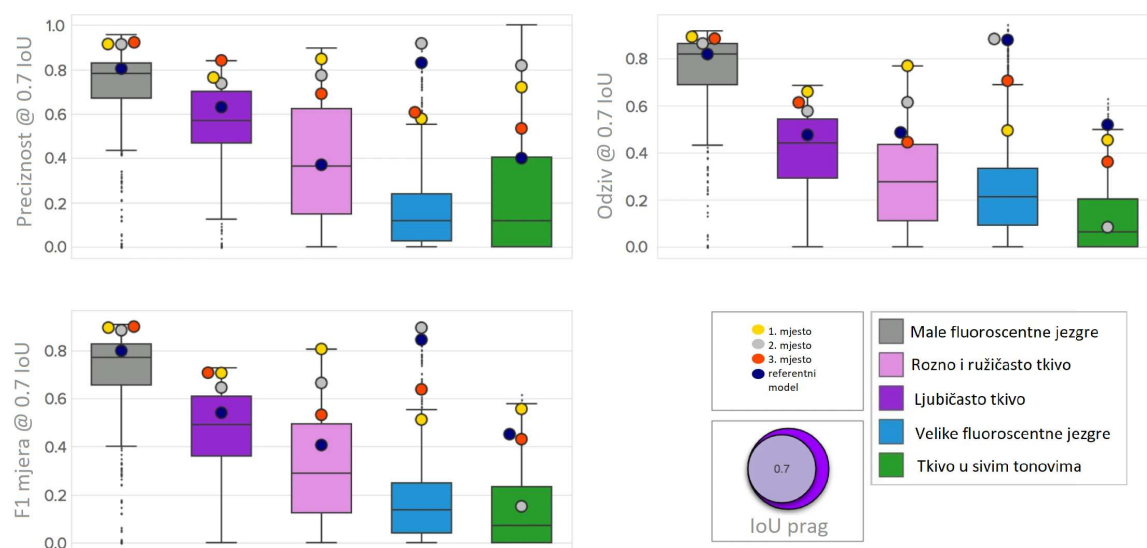
⁸https://github.com/matterport/Mask_RCNN



Slika 3.9: Usporedba stopa pogrešaka triju najuspješnijih rješenja s referentnim segmentima CellProfilera. Crni stupci i crvene točke prikazuju općenite performanse za sudionika, dok obojeni stupci označavaju performanse za različite grupe slika. Izvor: [9].

Mask-RCNN model je prenaučan na već spomenutom COCO skupu podataka za detekciju i segmentiranje objekata na slikama u prirodnom okruženju. Rješenje je obuhvatilo augmentaciju skupa podataka koje je izrazito značajno u biološkom kontekstu, uključilo je umjetne simulacije skaliranjem slika, imitirajući rad mikroskopa. Također su korištene modifikacije omjera proporcija, pomaci i rotacije. Skup podataka za učenje bio je uravnotežen s obzirom na vrste slika, iako za određivanje vrste slike nije korištena posebna analiza, samo se veličina slike uzimala u obzir za stvaranje nedovoljno zastupljenih slika sa nasumičnim augmentacijama. Tijekom treninga primijenjene su dodatne augmentacije podataka, a model nije ponovno učen za drugu fazu s dodatnim podacima nakon prve. To ostavlja prostor za istraživanje uloge više umjetno stvorenih podataka kakve je koristio ovaj model.

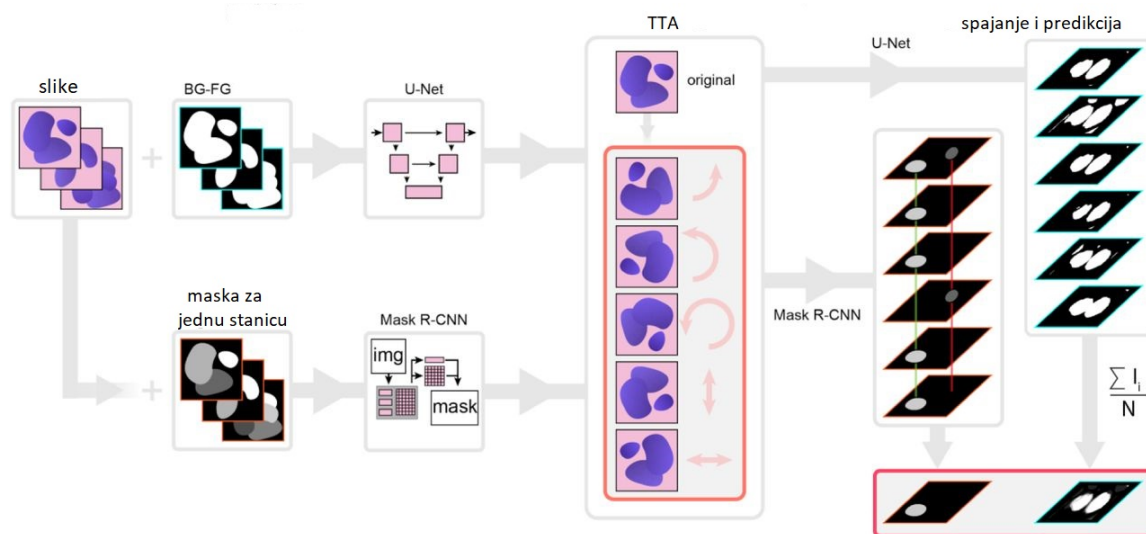
Kako bi generirao segmentacije za nove slike, sudionik je uveo dodatne augmentacije, proučavanjem testnih podataka pri različitim transformacijama i agregiranjem predviđanja u jedan izlaz. Ovaj pristup se naziva *test-time augmentation*, skraćeno TTA. Te su transformacije uključivale rotacije sa različitim kutovima, skaliranje slika i promjene boje. To je bila jedna od razlika u pristupu sudionika naspram drugih koji su bez približnog uspjeha koristili Mask-RCNN. Sudionik je također objavio da jednostavnije tehnike za kasniju obradu, kao što je morfološka dilatacija, mogu dostići slične performanse te da se konfiguracija parametara i povećanje podataka tijekom treninga čine važnijima u skladu s njegovim eksperimentima.



Slika 3.10: **Distribucija različitih metrika svih sudionika koji su prošli u drugu fazu natjecanja.** Točke na grafu predstavljaju tri najbolja sudionika i CellProfiler referencu. Sva se metrika mjeri sa pragom $t = 0.7$, što možemo protumačiti kao zahtjev da se objekti moraju simetrično preklapati najmanje sa 70% svojeg područja, kako bi se mogli računati kao stvarno pozitivni (TP). Izvor: [9].

3.5 TTA

Nedavni napredak u dubokom učenju unjeo je revoluciju u način obrade mikroskopskih slika stanica. Arhitekture mreže dubokog učenja imaju velik broj parametara i stoga, kako bi se postigla visoka točnost, zahtijevaju ogromnu količinu označenih podataka. Ustaljeni način poboljšanja točnosti temelji se na umjetnom povećanju podataka za učenje primjenom različitih tehnika povećavanja. Rijetko se modeli oslanjaju na povećanju skupa podataka tokom ispitivanja (**TTA**) koje rezultira transformiranim verzijama slike za predviđanje i agregiranjem rezultata. Čak i korištenjem jednostavnih algoritama, kao što su rotacija i odgovarajućih metoda agregiranja, autori [15] su pokazali da može doći do značajnog poboljšanja točnosti predviđanja. Koristili su slike tkivnih i staničnih kultura sa natjecanja i iz drugih izvora. Uz to, pokazali su da uz TTA mogu dodatno poboljšati rezultat na DSB-u.

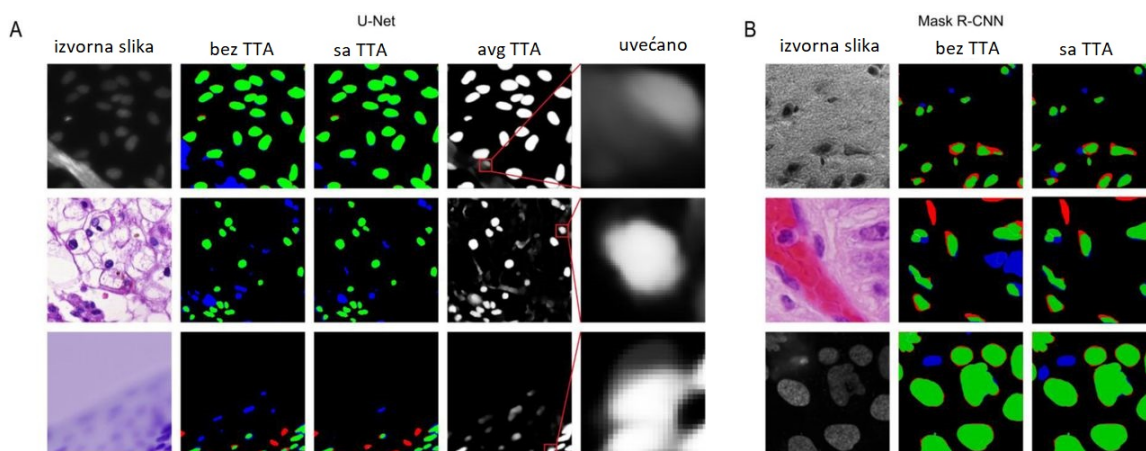


Slika 3.11: **TTA**. Predikcija se vrši na nekoliko i ispitnih slika, a rezultati su transformirani natrag i spojeni. U slučaju U-Net-a primijenjeno je većinsko glasovanje po pikselima, dok je za Mask R-CNN korištena kombinacija podudaranja predmeta i većinskog glasovanja. Izvor: [15].

TTA uključuje četiri postupka:

1. augmentaciju
2. predviđanje
3. disaugmentaciju
4. spajanje

Prvo se primjenjuje augmentacija na testnoj slici. Zatim se radi predikcija na originalnoj i izmjenjenim slikama, te potom se slika vrati u početno stanje, a predviđanja zadrže. Taj se proces naziva disaugmentacija. Primjerice, kada je predikcija izvršena na simetričnoj ili rotiranoj slici, dobiveno slika se vraća u prvotnu orijentaciju. Završni korak spajanja nije izravan u slučaju Mask-RCNN jer je arhitektura raspoznaje instance, stoga metoda spajanja mora rukovati instancama. Autori su razvili proširenu metodu spajanja inspiriranu modelom koji je ostvario peti najbolji rezultat na DSB 2018. (slika 3.11). Za svaki detektirani objekt s izvorne slike, isti objekt nalazi se u slikama dobivenim augmentacijom i detektiraju ga računajući IoU između maski. Minimalni IoU prag koji se koristi za odlučivanje jesu li pronađeni objekti isti je $t = 0.5$. Iterirali su sve otkrivene objekte da bi pronašli najbolju podudarnost. Objekt bi trebao biti prisutan u većini slika koje će biti uključene kao



Slika 3.12: **Primjeri predikcije. A: U-Net predikcija.** Prvi stupac - originalna slika, drugi stupac - predviđanja bez TTA u odnosu na stvarnu vrijednost, treći stupac - predviđanja sa TTA u odnosu na stvarnu vrijednost. Crvene oznake ukazuju na FN, zelene oznake ukazuju na TP, dok plave oznake ukazuju na FP. Četvrti stupac – prosječna predviđanja TTA prije *thresholding*, peti stupac – uvećani segmenti iz prethodnog stupca. **B: Mask R-CNN predikcija.** Stupci su prva kao i tri stupca u A. Izvor: [15].

završna maska. Zatim su provjeravali prvu sliku dobivenu augmentacijom, kako bi provjerali za preostale neotkrivne objekte (mogući scenarij je da objekt nije otkriven u originalnoj slici je otkriven u bilo kojoj slici iz procesa augmentacije). Tražili su odgovarajuće neotkrivene objekte i na drugim augmentacijama. Zatim na drugu augmentiranu sliku obavljaju iste operacije. Proces se ponavlja sve dok većina kriterija za glasovanje ne bude teoretski zadovoljena. Prosječna binarna maska objekta stvara se glasovanjem većine piksela na uparenim objektima.

Za U-Net proces spajanja je jednostavan jer ne raspoznaje instance, tako da se jednostavno zbroji i računa prosječna vjerojatnost da je tu jezgra. To rezultira slikom čiji su pikseli decimalne vrijednosti i koju treba pretvoriti u binarnu masku. Jednostavnom metodom praga, čija je vrijednost 0.5, slika se pretvara u binarnu masku (slika 3.11 desno).

3.6 Zaključak

Data Science Bowl 2018. predstavio je izazov automatskog pronalaženja jezgri u velikom broju neviđenih mikroskopskih slika, bez dodatne konfiguracije. To je prvi dokumentirani pokušaj izrade modela koji bi mogao segmentirati obojene jezgre stanica iz 15 bioloških eksperimenata, kroz eksperimentalne uvjete, iz laboratorijskih podataka. Glavni cilj izazova bio je istražiti generičke strategije segmentacije koje bi se mogle automatski

primijeniti na mnoge eksperimente snimanja bez daljnje intervencije korisnika. Tim bi se pristupom moglo skratiti vrijeme za obradu slika, čime bi se buduće generacije biologa rasteretile pri provođenju kvantitativnih eksperimenata snimanja za istraživanje i kliničku praksu.

Obučavanje automatiziranih alata za segmentaciju jezgri primjenom modernih pristupa strojnog učenja zahtijeva prikupljanje označenih primjera. Spomenuto natjecanje iznjedrilo je resurse raznovrsnih slika koje su donirali brojni biološki laboratoriji i koje su ručno obilježili stručni biolozi u institutima širom svijeta. Svi ti podaci sada su javno dostupni s dozvolama za javno dobro kako bi se olakšala buduća znanstvena istraživanja i industrijski razvoj. Nadamo se da će drugi u široj biološkoj zajednici dati više slika i primjedbi za rast ovog resursa s dodatnim eksperimentalnim varijacijama, uključujući elektronsku mikroskopiju, kao i mnoge druge uobičajene modalitete slika koji nisu bili uključeni u ovo istraživanje.

Izazov je privukao sudjelovanje različitih timova znanstvene zajednice, koji su dali različite vrste doprinosa, učili zajedno i surađivali kako bi bolje razumjeli problem i postigli napredak i ostvarili zajednički cilj. Rješenja, koja su predstavili neki sudionici, u kojima su postigli da jedan modela bez intervencije može segmentirati različite mikroskopske slike, su zadivljujuća. Eksperimentalni rezultati upućuju na to da bi segmentacija jezgre mogla biti u potpunosti automatizirana, što znači da se ne zahtijeva ručno postavljanje ili ekspertiza za obradu slika od strane korisnika, a istovremeno osigurava veću točnost u odnosu na ocijenjene alate. Veća točnost može biti moguća kroz veći, raznolikiji skup podataka za učenje i ugradnjom najnovijih dosega strojnog učenja i istraživanja računalnog vida.

Najuspješniji sudionici predstavili su rješenja temeljena na potpunim konvolucijskim mrežama (U-Net i FPN) i rezidualnim mežama. Ta dva pristupa bila su rasprostranjena tijekom natjecanja; ono što je razlikovalo pobjednike bila je kombinacija tehnika predobrade i postobrade, kao i primjena najboljih metoda tijekom učenja (uravnotežavanje i uvećavanje skupa podataka). Uobičajeno među najboljim konkurentima bila je uporaba augmentacije podataka tijekom treninga i testiranja, uključujući izmjene boja kako bi mreže bile invarijantne, te metode skaliranja za rješavanje izazova veličine objekta. Zanimljivo je da su sva tri najbolja rješenja koristila strategiju rangiranja najboljih segmentacija od nekoliko predviđenih pomoću osnovnih modela. Iako je to uobičajena praksa za modele nalik RCNN-u (treće rješenje), prva dva modela su stvorila i vlastite strategije za postizanje sličnog učinka pomoću potpunih konvolucijskih mreža.

Rezultati predstavljaju uspješan dokaz da je duboko učenje uistinu sposobno pružiti točne rezultate bez interakcije korisnika. Međutim, iako su najbolji modeli javno dostupni, još uvijek ima mjesta za dodatnu primjenu ekspertnih znanja. Alat prilagođen korisniku potreban je za premošćivanje jaza između najsuvremenijih rješenja i svakodnevne biomedicinske prakse. Također, pokazano je da je dostupnost podataka ograničen je za postizanje

vrhunskih performansi za različite vrste slika, stoga su potrebni dodatni naponi za prikupljanje i označavanje više podataka kako bi se proširila primjenjivost budućih sustava. Sposobnost generalizacije modela može se procijeniti i na drugim skupovima podataka koji se nisu koristili tijekom izazova, kao što su podaci sa izazov praćenja stanica i drugi. Međutim, preostaje pitanje rješavanja drugih aspekata uporabljivosti. Primjerice, ako postoje pogreške u segmentaciji, kako modeli mogu učinkovito i jednostavno uzimati povratne informacije od ljudi kako bi ispravili pogreške u segmentaciji? Rezultati istraživanja Science Bowl 2018. prvi su korak prema stvaranju generičkog sustava za segmentiranje jezgre stanica na svakoj mikroskopskoj slici. Budući rad mogao bi proširiti skup podataka tako da obuhvati vrste mikroskopskih snimaka koje nisu bile obuhvaćene, kao što su i trodimenzionalne slike. Slijedeći ovdje navedenu strategiju, mogu se izraditi i modeli za segmentiranje ostalih staničnih struktura uz jezgru, kao što su stanične membrane i organeli.

Poglavlje 4

Modeliranje vlastitog modela

4.1 Uvod

Kako bi vidjeli kako nastaje model u praksi, za potrebe diplomskog rada implementirao sam vlastiti model pomoću Tensorflow¹ biblioteke na Google Colab I Kaggle platformama. Tijekom modeliranja i učenja, postupno sam ga poboljšavao i dodavao nove slojeve, sve kako bi razvio model koji bi automatski segmentirao nakupine tumora na centralnom živčanom sustavu miša.

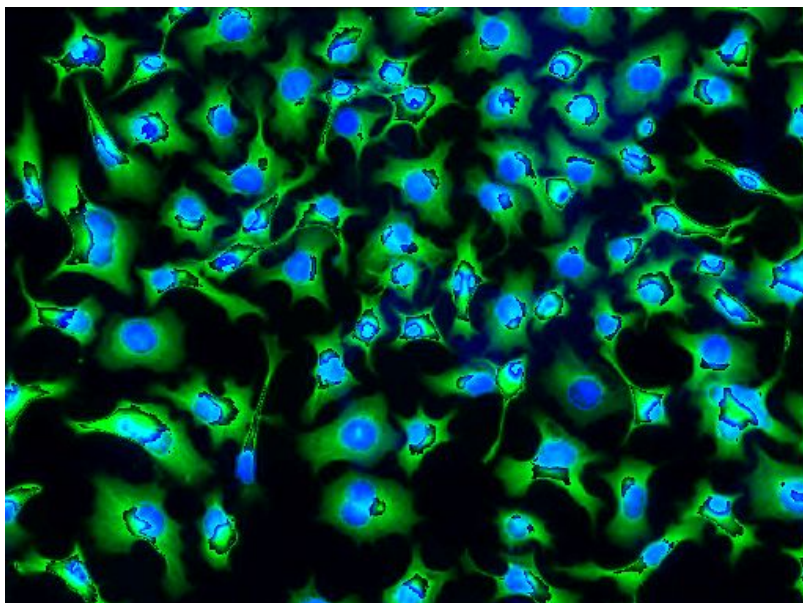
4.2 Skup podataka

Kao skup podataka za učenje koristio sam slike nakupina neurona miša objavljene na *CellImageLibrary.org*². Ovaj se skup podataka sastoji od epifluorescentnih snimaka nakupina neurona miša s citoplazmatskim i staničnim mrljama (DAPI) te niza ručno stvorenih segmentacijskih maski neuronskih i staničnih granica koji se mogu koristiti kao za razvoj algoritama i modela segmentacije. Problem koji se ističe kod ovih podataka je subjektivnost anotatora, jer velika većina slika ima generirane dvije datoteke koje predstavljaju segmentacijsku masku, dok su dva uzorka imala čak i treću masku generiranu. Iako samo jedna prikazuje većinu neurona, a ostale su nadopune i/ili moguće slučajeve, subjektivna pristranost i neodlučnost može biti utjecajan čimbenik u učenja modela. Skup slika sadrži 200 slika koje su generirane pomoću elektromikroskopije iz 4 kulture uzgojene u različitim uvjetima:

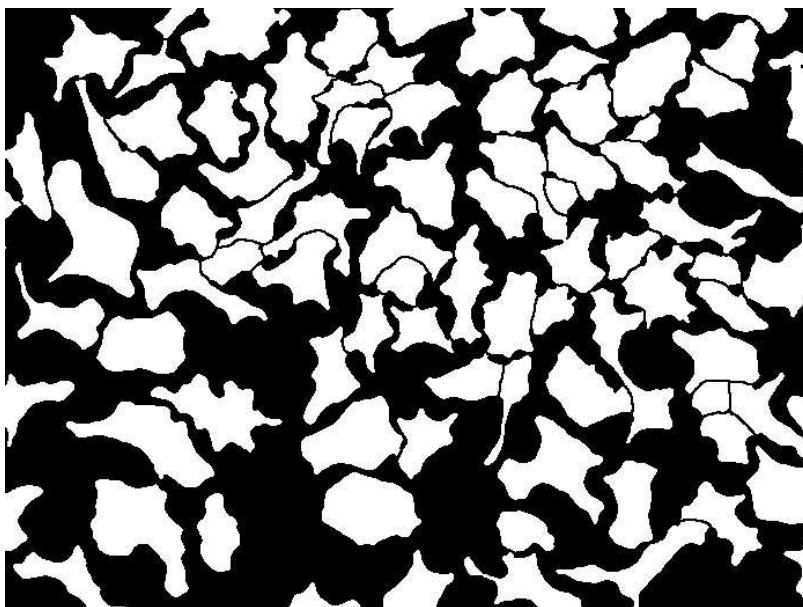
1. pomoću "serum starvation" metode molekularne biologije na 50 000 uzoraka stanica,

¹<https://www.tensorflow.org>

²http://www.cellimagelibrary.org/images/CCDB_6843



Slika 4.1: Uzorak iz dataset-a na kojemu je model treniran. Slika je dimenzija 1392×1040 .

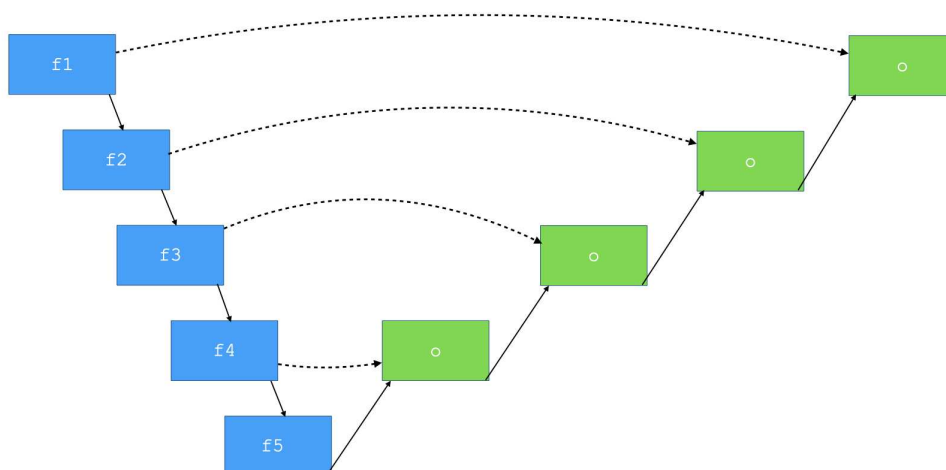


Slika 4.2: Maska generirane za istu ulaznu sliku prikazanu 4.1.

2. pomoću "serum starvation" metode molekularne biologije na 30 000 uzoraka stanica,
3. transfekcijom stanica pomoću proteina Toca-1 prenesen na 50 000 uzoraka stanica i
4. transfekcijom stanica pomoću proteina Toca-1 prenesen na 30 000 uzoraka stanica.

Međutim, s obzirom da ti uvjeti nisu relevantni za potrebe modeliranja modela dubokog učenja, svi podaci su objedinjeni u jedan skup podataka. Slike su dimenzija 1392×1040 i prikazuju neuroblastom (vrsta tumora) na centralnom živčanom sustavu kućnog miša (lat. *mus musculus*). Uz njih su generirane maske dimenzija 696×520 .

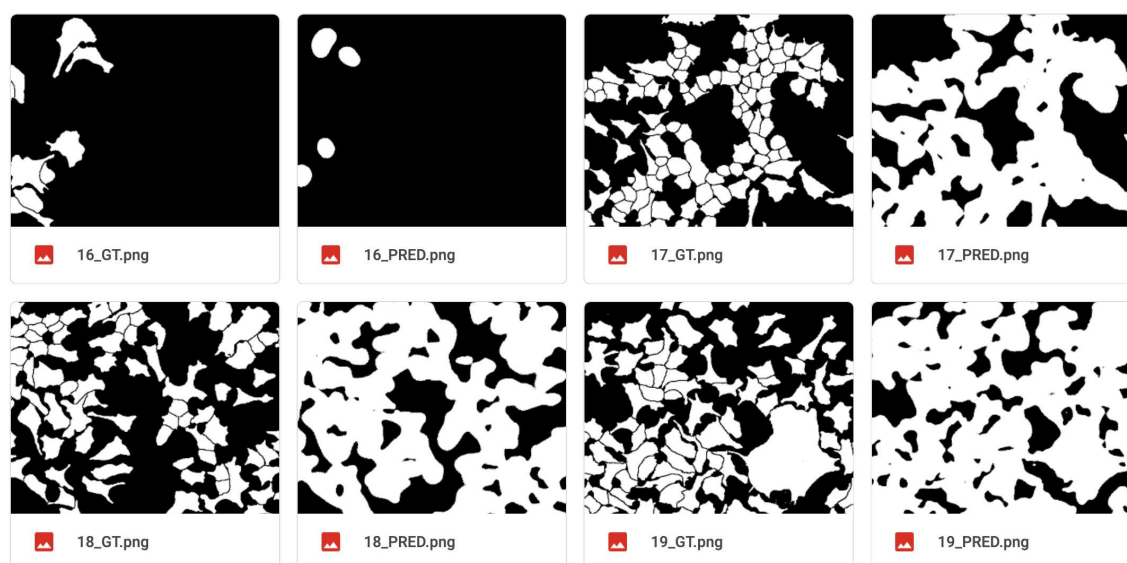
4.3 Modeli i učenje



Slika 4.3: Inicijalni model. Svaki plavi i zeleni blok se sastoji od sljedećih operacija redom navedenih: konvolucije, normalizacije, aktivacije pomoću ReLU funkcije, ispuštanja, ponovne konvolucije, normalizacije i aktivacije. Puna strelica prema dolje predstavlja sažimanje, puna strelica prema gore predstavlja dekonvoluciju. Isprekidane veze predstavljaju concatenaciju izlaza odgovarajućeg sloja encodera (plavi pravokutnici) na ulaz u odgovarajući sloj decodera (zeleni pravokutnici).

Kako je ovo bio moj prvi zadatak iz segmentacije slike, odlučio sam krenuti sa jednostavnijim model. Pokušao sam i uspio implementirati klasičnim UNet model, implementirajući

sloj po sloj, blok po blok. Jedan od problema s kojima sam se susreo su dimenzije slike, gdje polovina dimenzije (smanjio sam veličinu ulazne slike kako bi odgovarala dimenziji maski) nije višekratnik 2^5 , te se operacija *sažimanja* u jednom trenutku našla u situaciji gdje je imala neparnu dimenziju koju je trebala prepoloviti. Taj problem sam zaobišao dopunjavanjem nulama oko donjeg ruba slike, a također trebalo je pripaziti na ovaj detalj u povratnom procesu kada se u decoderu multiplicira dimenzija za faktor 2. Nakon 20 epoha učenja i treniranja modela na početnom datasetu, rezultati nisu bili ni približno dobri, jedino je lokalizacija bila prihvatljivo.

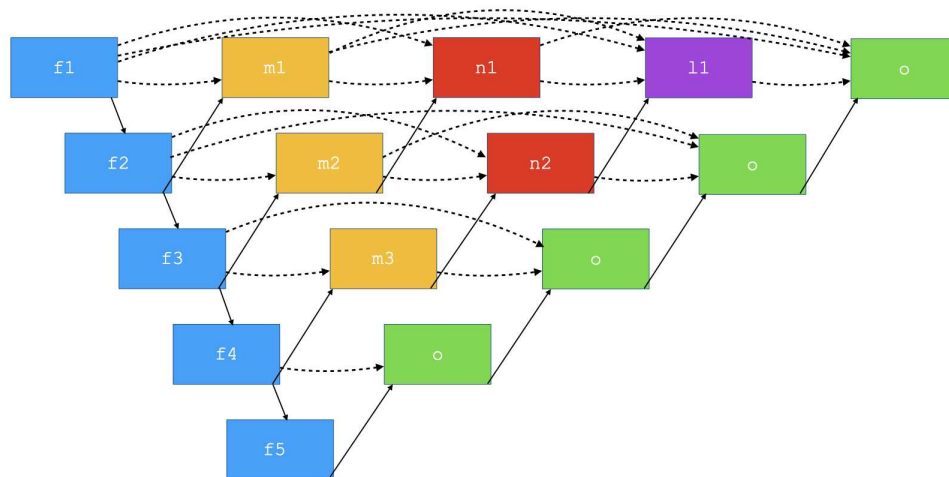


Slika 4.4: Rezultati predikcije prvoga modela na uzorku iz skupa za validaciju.

Slika:	F_1 mjera
12 :	0.8607841511936339
13 :	0.9487013704686118
14 :	0.9725519451812555
15 :	0.9707918877099911
16 :	0.9606349469496022
17 :	0.7782548629531388
18 :	0.7151000221043324
19 :	0.7060344827586207

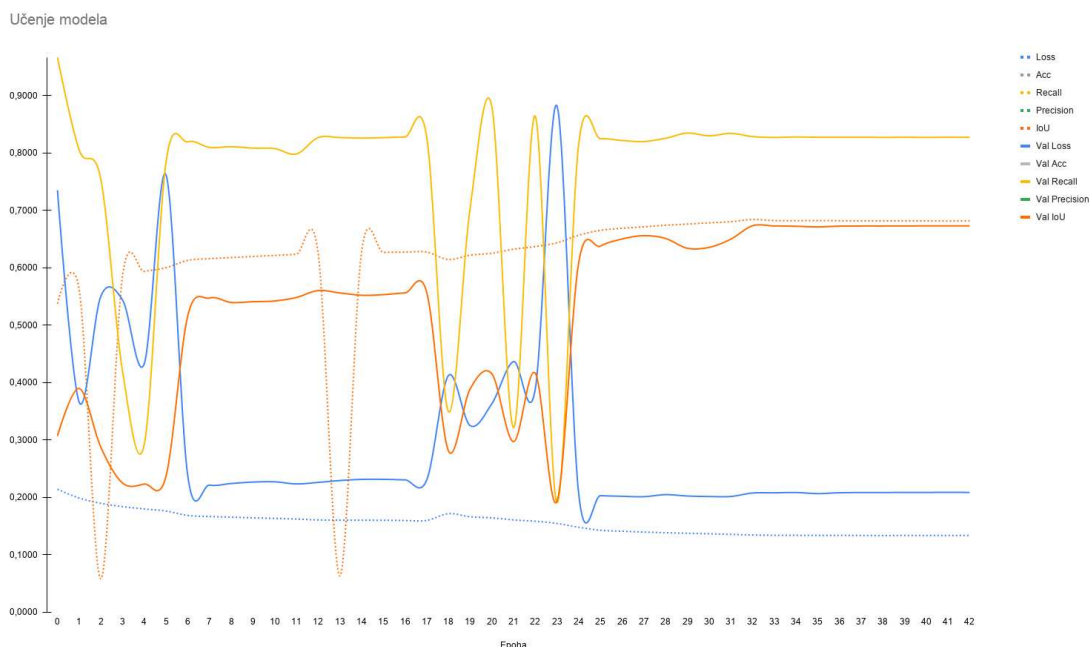
Tablica 4.1: Rezultati inicijalnog modela. Uspoređujući tablicu i sliku, vidimo da je na slikama koje sadrže više stanica za segmentaciju, F_1 mjera oko 70%.

Zatim sam odlučio napraviti augmentaciju podataka te sam od prvotnih 200 slika iz *dataseta* različitim metodama iz *tf.image* modula generirao 800 slika. Odlučio sam učiniti prvotne plave f_i blokove rezidualnim i dodao sloj zbrajanja ishoda prve i druge aktivacije u svaki taj blok. Istraživajući, unaprijedio sam model dodatnim operacijama. Nakon svaki sloja nakon prvog sažimanja sam dekonvoluirao kako bi model mogao steći znanje iz tih slojeva. I svaki taj sloj sam povezao sa po formatu odgovarajućim slojevima u neuronskoj mreži, čineći više povezanim.



Slika 4.5: Inicijalni model unaprijeđen dodatnim decoderima u središnjem dijelu. Strelice odgovaraju istim operacijama kao i u slici 4.3.

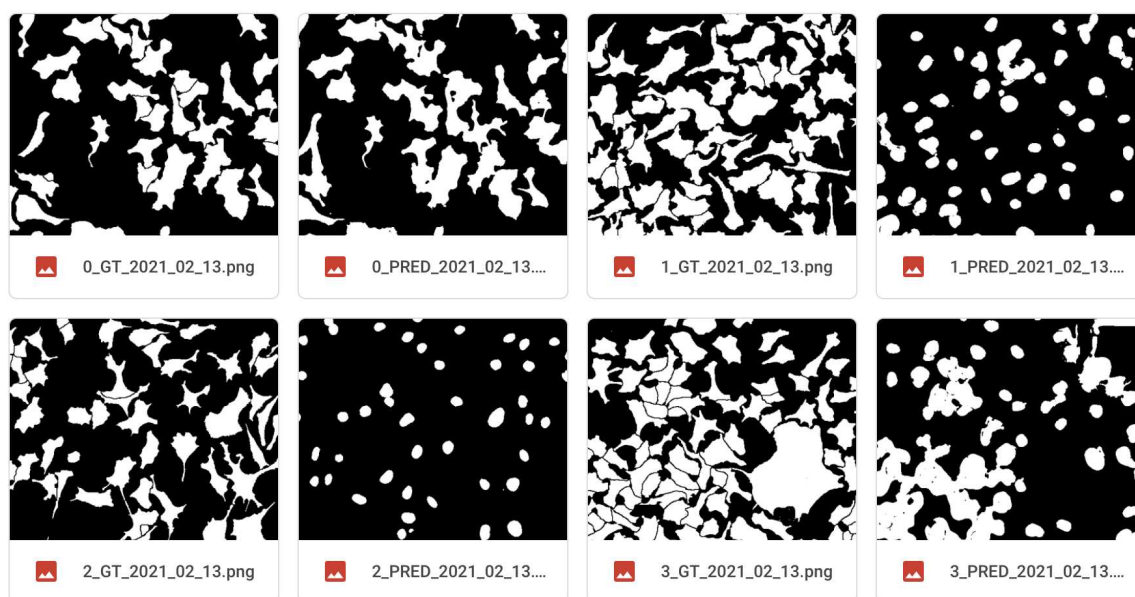
Nakon toga uslijedilo je ponovno treniranje modela iz početka, u dva odvojena pokretanja od 20-ak etapa svako.



Slika 4.6: Prikaz važnijih mjera tokom učenja modela. Prve isprekidane crte su mjere tijekom treninga, a pune su dobivene iz validacije nakon svake epohe.

4.4 Rezultati

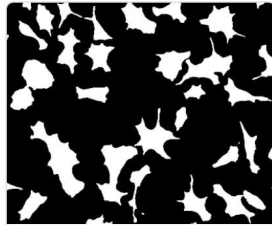
Rezultati nakon zadnjeg učenja modela su prikazani ispod, tablično i slikovno. Model je na datasetu za evaluaciju ostvario vrijednost funkcije troška od 30,04%, točnost od 90,19%, preciznost 87,13%, odziv 61,03%, IoU 45,02% i F_1 mjeru 58,21%. Ispod su prikazani umanjeni vizualni rezultati iz skupa za evaluaciju. Nakon mnogih testiranja, uočavam da model ipak nije potpuno naučio razaznati sitne granice između stanica, te bi za takvo znanje trebalo ipak razmotriti složenije modele. Ipak mogućnosti modela dubokog učenja su izvanredne! Zamislimo samo model koji u realnom vremenu pomoću segmentacije slike prati nastanak novih malignih stanica i detektira neželjne stanice, te kakvu pomoć takvi modeli ponudili medicini, te u kranjem slučaju koliko bi ubrzali detekcije tumora i koliko bi života spasili.



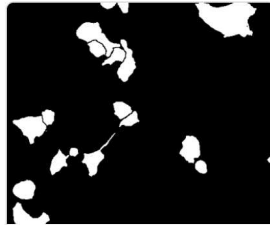
Slika:	F ₁ mjera	Slika:	F ₁ mjera
0 :	0.9708968832891247	10 :	0.9489168877099912
1 :	0.6944131299734748	11 :	0.9078636162687886
2 :	0.8214633068081344	12 :	0.801353890362511
3 :	0.7099441865605659	13 :	0.9350298408488064
4 :	0.9850104995579133	14 :	0.9713141025641026
5 :	0.9756189213085765	15 :	0.958612400530504
6 :	0.9682112068965517	16 :	0.9415146993810787
7 :	0.994570623342175	17 :	0.9575431034482759
8 :	0.9657355216622459	18 :	0.9472286693191866
9 :	0.8191451149425287	19 :	0.8834272767462422



4_GT_2021_02_13.png



4_PRED_2021_02_13...



5_GT_2021_02_13.png



5_PRED_2021_02_13...



6_GT_2021_02_13.png



6_PRED_2021_02_13...



7_GT_2021_02_13.png



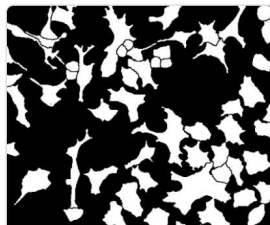
7_PRED_2021_02_13...



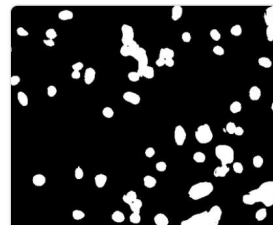
8_GT_2021_02_13.png



8_PRED_2021_02_13...



9_GT_2021_02_13.png



9_PRED_2021_02_13...



10_GT_2021_02_13.p...



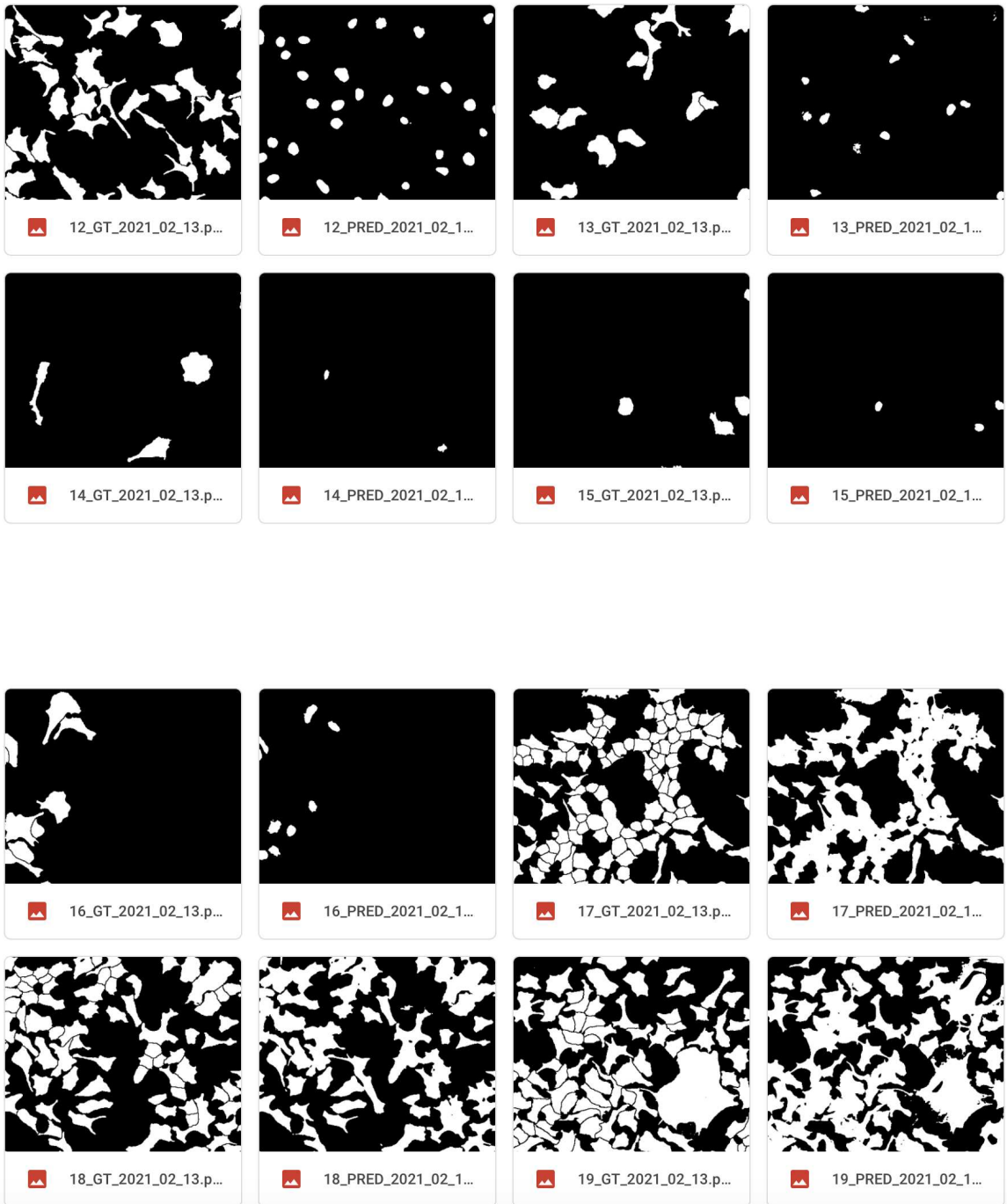
10_PRED_2021_02_1...



11_GT_2021_02_13.p...



11_PRED_2021_02_1...



Bibliografija

- [1] *Visual Object Classes Challenge 2012*, (2012.), <http://host.robots.ox.ac.uk/pascal/VOC/voc2012/>.
- [2] Y. LeCun L. Bottou Y. Bengio i P. Haffner, *Gradient-based learning applied to document recognition*, Proceedings of the IEEE, 86(11):2278–2324, 1998.
- [3] Jennifer Bresnick, *What Is Deep Learning and How Will It Change Healthcare?*, (2018.), <https://healthitanalytics.com/features/what-is-deep-learning-and-how-will-it-change-healthcare>.
- [4] Alpaydin E., *Introduction to Machine Learning: Supervised Learning*, London: The MIT Press, 2004.
- [5] Bajcsy et al., *Survey statistics of automated segmentations applied to optical imaging of mammalian cells*, (2015.).
- [6] E. Moen et al., *Deep learning for cellular image analysis.*, (2019.), <https://www.nature.com/articles/s41592-019-0403-1>.
- [7] F. Buggenthin et al., *Prospective identification of hematopoietic lineage choice by deep learning.*, (2017.), <https://doi.org/10.1038/nmeth.4182>.
- [8] Hesamian et al., *Deep Learning Techniques for Medical Image Segmentation: Achievements and Challenges.*, (2019.).
- [9] J. C. Caicedo et al., *Nucleus segmentation across imaging experiments: the 2018 Data Science Bowl.*, (2019.), <https://www.nature.com/articles/s41592-019-0612-7>.
- [10] L. Keren et al., *A structured tumor-immune microenvironment in triple negative breast cancer revealed by multiplexed ion beam imaging.*, (2018.).
- [11] N. Nitta et al., *Intelligent Image-Activated Cell Sorting.*, (2018.), <https://www.sciencedirect.com/science/article/pii/S0092867418310444>.

- [12] Louis Fernandes, *The journey from data to knowledge and wisdom*, (2014.), <https://www.linkedin.com/pulse/20140722053326-3033591-the-journey-from-data-to-knowledge-and-wisdom/>.
- [13] Kaiming He, Xiangyu Zhang, Shaoqing Ren i Jian Sun, *Deep Residual Learning for Image Recognition*, 2015.
- [14] Tomislav Levanić, *Neuronske mreže*, (2016.), <https://web.math.pmf.unizg.hr/nastava/matsoft/DobreDZ/2015-16/HTML/TomislavLevanic/>.
- [15] Nikita Moshkov, Botond Mathe, Attila Kertesz-Farkas, Reka Hollandi i Peter Horvath, *Test-time augmentation for deep learning-based cell segmentation on microscopy images*, bioRxiv (2019), <https://www.biorxiv.org/content/early/2019/10/23/814962>.
- [16] Olaf Ronneberger, Philipp Fischer i Thomas Brox, *U-Net: Convolutional Networks for Biomedical Image Segmentation*, 2015.
- [17] Vladimír Ulman, Martin Maška, Klas Magnusson, Olaf Ronneberger, Carsten Haubold, Nathalie Harder, Pavel Matula, Petr Matula, David Svoboda, Miroslav Radojevic, Ihor Smal, Karl Rohr, Joakim Jaldén, Helen Blau, Oleh Dzyubachyk, Boudewijn Lelieveldt, Pengdong Xiao, Yuexiang Li, Siu Yeung Cho i Carlos Ortiz-de Solorzano, *An Objective Comparison of Cell Tracking Algorithms*, *Nature Methods* **14** (2017).

Sažetak

U uvodu ovoga diplomskog rada mogu se pronaći valjani razlozi okretanja strojnom i dubokome učenju, njegove sve veće prednosti nad tradicionalnim metodama, ali i pokoja mana. Zatim smo teoretski skupa prošli cijeli proces dubokog učenja, od modeliranja problema, samog procesa učenja pa i do najsitnijih gradivnih jedinica svake neuronske mreže, njenih slojeva i principa rada i učenja. Nakon toga smo se okrenuli konkretnoj primjeni dubokog učenja u medicini, segmentiranju slika stanica nastalih svjetlosnom elektromikroskopijom stanica iz različitih vrsta tkiva koja su se našla na natjecanju Data Science Bowlu 2018. Vidjeli smo i opisali najuspješnije algoritme i metode koje su natjecatelji koristili, a slobodno možemo reći i najsuvremenija rješenja koja su neki od njih implementirali. Zatim smo došli i do najvažnijeg dijela ovoga rada, procesa modeliranja autorovog modela koji se okušao u segmentaciji slika tumora na centralnom živčanom sustavu kućnog miša. Zadnje poglavlje obuhvaća sve korake i procese u nastanku istoga, te rezultate koji su na nekim uzorcima na jako dobrom tragu, ali kod pojedinih ipak sugeriraju da bi se trebali okrenuti kompleksnijim modelima većeg kapaciteta učenja.

Summary

In the introduction of this Master work, one can find valid reasons for turning to machine and deep learning, its increasing advantages over traditional methods, but also a few defects. Then, theoretically, we went through the entire process of deep learning, from problem modeling, the learning process itself to the tiniest building units of each neural network, its layers and principles of work and learning. After that, we turned to the concrete application of deep learning in medicine, segmentation of images of cells produced by light electromicroscopy of cells from different types of tissues found in the Data Science Bowl competition in 2018. We have seen and described the most successful algorithms and methods used by competitors, and we can freely say state-of-the-art solutions that some of them have implemented. Then we came to the most important part of this paper, the modelling process of the author's model that was tested in segmentation of tumor images on the central nervous system of the mouse. The last chapter covers all the steps and processes in the creation of the same, as well as the results that are on some samples on a very good track, but in some cases suggest that they should turn to more complex models of greater learning capacity.

Životopis

Rođen sam u Slavonskom Brodu, osnovnu školu sam pohađao malom slavonskom mjestu Štitar, pokraj Županje. Srednjoškolsko obrazovanje sam nastavio u Županji, gdje sam završio Opću gimnaziju i upisao se na nastavnički preddiplomski sveučilišni studij Matematike Prirodoslovno-matematičkog fakulteta. Nakon završenog preddiplomskog studija, upisujem diplomski sveučilišni studij Računarstva i matematike na istom fakultetu. U trenutku nastanka rada radim kao razvojni programer u tvrtci Ericsson Nikola Tesla d.d. kao član internacionalnog tima.