SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 1163

ALGORITMI ZA SEMANTIČKU SEGMENTACIJU RADIOLOŠKIH SLIKA

Petra Renić

Zagreb, lipanj 2023.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 1163

ALGORITMI ZA SEMANTIČKU SEGMENTACIJU RADIOLOŠKIH SLIKA

Petra Renić

Zagreb, lipanj 2023.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

Zagreb, 10. ožujka 2023.

ZAVRŠNI ZADATAK br. 1163

Pristupnica:	**Petra Renić (0036536102)**
Studij:	Elektrotehnika i informacijska tehnologija i Računarstvo
Modul:	Računarstvo
Mentor:	prof. dr. sc. Siniša Šegvić

Zadatak: **Algoritmi za semantičku segmentaciju radioloških slika**

Opis zadatka:

Semantička segmentacija važan je zadatak računalnog vida s mnogim zanimljivim primjenama. Veliki napredak u posljednje vrijeme postižu modeli utemeljeni na slojevima pažnje. Ipak, konvolucijski modeli su i dalje kompetitivni, posebno u slučajevima kada je računski budžet ograničen. Ovaj rad razmatra primjene vezane uz analizu medicinskih slika dobivenih različitim dijagnostičlim postupcima. U okviru rada, potrebno je odabrati okvir za automatsku diferencijaciju te upoznati biblioteke za rukovanje matricama i slikama. Proučiti i ukratko opisati postojeće pristupe za semantičku segmentaciju. Odabrati slobodno dostupni skup slika te oblikovati podskupove za učenje, validaciju i testiranje. Odabrati prikladnu arhitekturu te uhodati postupke učenja modela i validiranja hiperparametara. Primijeniti naučene modele te prikazati i ocijeniti postignutu točnost. Radu priložiti izvorni i izvršni kod razvijenih postupaka, ispitne slijedove i rezultate, uz potrebna objašnjenja i dokumentaciju. Citirati korištenu literaturu i navesti dobivenu pomoć.

Rok za predaju rada: 9. lipnja 2023.

Zahvaljujem mentoru prof.dr. sc. Siniši Šegviću

na prenesenom znanju, korisnim savjetima i strpljenju.

Sadržaj

Uvod

Računalni vid grana je umjetne inteligencije koja proučava i razvija algoritme i tehnologije za interpretaciju i obradu vizualnih podataka. Osiguravanje pouzdane i učinkovite medicinske dijagnostike i terapije jedan je od najvažnijih izazova u zdravstvenoj skrbi. U tu svrhu, računalni vid danas igra važnu ulogu u medicini, omogućavajući analizu medicinskih snimaka na temelju piksela. Prije pojave računalnog vida, analiza medicinskih snimaka zahtijevala je visoku razinu stručnosti ljudskih stručnjaka i bila je skupa i vremenski zahtjevna. Razvoj računalnog vida olakšao je ovaj proces, omogućavajući brzu, preciznu i pouzdanu analizu medicinskih slika, što u konačnici dovodi do bolje dijagnostike i terapije. S druge strane, računalni vid također se primjenjuje u drugim područjima, poput autonomne vožnje automobila, prepoznavanja neispravne robe na proizvodnoj traci i drugima, što otvara nove tehnološke mogućnosti u različitim industrijama.

Semantička segmentacija slika je postupak koji omogućuje označavanje svakog piksela na slici s odgovarajućom klasom i pripadajućim značenjem. To je proces koji se oslanja na algoritme i strojno učenje kako bi automatski prepoznali različite dijelove slike i klasificirali ih u specifične kategorije. U ovom radu, primijenit ću semantičku segmentaciju na radiološke slike abdomena. Cilj je izolirati dijelove slike koji se odnose na različite organe i pomoći medicinskim stručnjacima u označavanju specifičnih anatomskih struktura abdomena, kao i patoloških promjena ili lezija.

1. Duboko učenje

1.1. Umjetne neuronske mreže

Neuronske mreže predstavljaju računalne modele inspirirane ljudskim mozgom, a temelje se na konceptu umjetnih neurona. Imaju slojevitu strukturu i građene su od neurona koji su povezani. Uloga neurona u mreži je da obrađuju podatke ulaza na način da obavljaju matematičke operacije nad njima i prosljeđuju dobivene vrijednosti drugim neuronima. Neuroni su, kao i u biološkom modelu, povezani sinaptičkim težinama koje određuju važnost veze.

1.1.1. Umjetni neuron

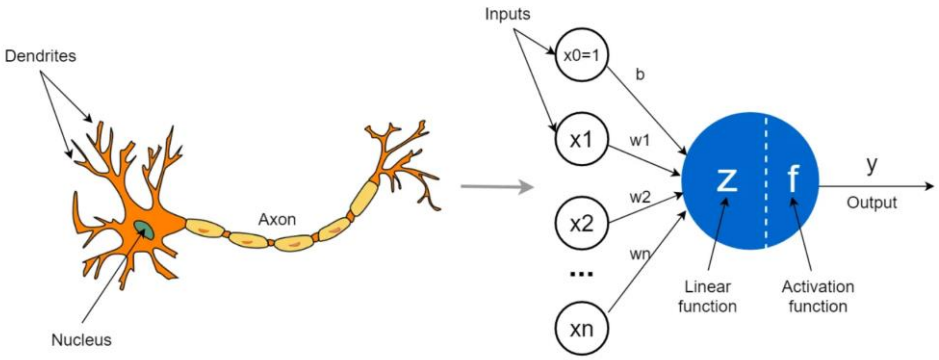
Umjetni neuron predstavlja osnovni građevni blok neuronske mreže. On prima ulazne podatke sa ulaza ili kao izlaze drugih čvorova putem veza sa pridruženim težinama. Težine nam omogućavaju da pojedini ulazi više ili manje utječu na izlaz neurona, te tako odražavaju važnost ulaznih signala. Ulaz u neuron se množi s težinom veze te se dobiveni produkti zatim zbrajaju sa pristranostima kako bi se na kraju dobila ukupna suma. Rezultat sume se obrađuje kroz aktivacijsku funkciju čime nastaje izlaz neurona.

Obrada podatka u neuronu se može definirati kao:

$$f(x) = \sum_{i=1}^{n} w_i x_i \; + \; w_0 \qquad (1)$$

Gdje su w_i težine veze, x_i vrijednost ulaza, a w_0 pristranost.

Inspiracija za umjetni neuron proizlazi iz neurona u ljudskom mozgu gdje ulaze u umjetni neuron možemo usporediti sa dendritima (eng. *dendrites*) koji primaju signale iz okolnih neurona u mozgu, tijelo biološkog neurona (eng. *nucleus*) sa funkcijom sumiranja i aktivacijskom funkcijom te akson (eng. *axon*) sa vezom i prijenosom informacija prema ostalim neuronima. Usporedbu građe možemo vidjeti na slici 1.



Slika 1. Građa biološkog (lijevo) i umjetnog neurona (desno), preuzeto iz [22]

1.1.2. Aktivacijske funkcije

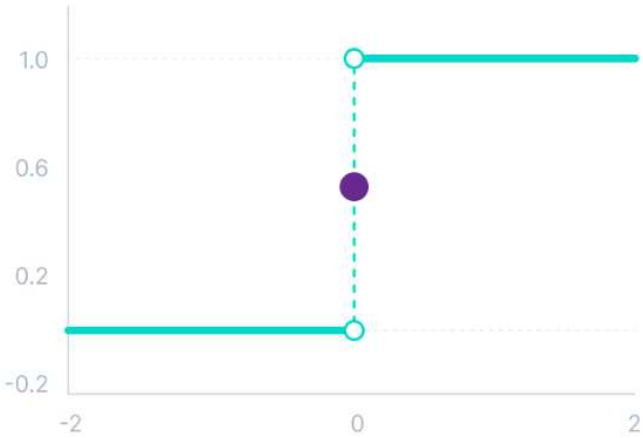
Kompozicija linearnih funkcija je uvijek linearna funkcija, bez obzira na broj slojeva u modelu, pa se iz tog razloga koriste aktivacijske funkcije kako bi se uvela nelinearnost u model. Kad se nebi koristile aktivacijske funkcije, izlaz neurona posljednjeg sloja bio bi linearna kombinacija ulaza, što bi ograničilo kapacitet mreže za rješavanje složenijih problema i sposobnost učenja.

Funkcija skoka

Funkcija skoka ovisi o graničnoj vrijednosti kojom se određuje hoće li se neuron aktivirati ili ne. Ulazna vrijednost aktivacijske funkcije se uspoređuje sa graničnom vrijednošću, te ako je veća od granice, neuron se aktivira, a inače se njegova vrijednost ne prenosi u idući sloj. Funkcija skoka može se koristiti u binarnoj klasifikaciji u posljednjem sloju mreže, ali se u ostalim zadacima danas rijetko koristi jer današnje neuronske mreže za učenje koriste propagaciju pogreške unatrag, a kako je derivacija funkcije skoka 0 svugdje osim u točki $x = 0$, ona nije dobar izbor za takvu vrstu učenja.

Funkcija skoka je aktivacijska funkcija koja se definira kao:

$$f(x) = \begin{cases} 0, & x < 0 \\ 1, & x \geq 0 \end{cases} \qquad (2)$$



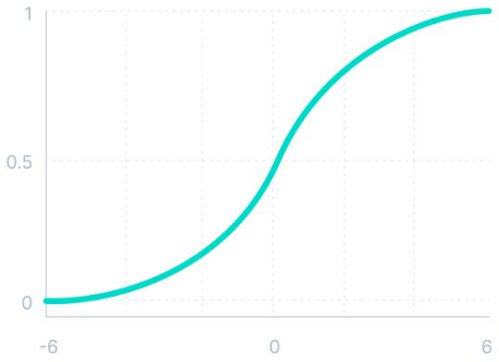
Slika 2. Graf funkcije skoka, preuzeto sa [7]

Sigmoidalna funkcija

Sigmoidalna funkcija je aktivacijska funkcija koja transformira vrijednosti ulaza u vrijednosti u intervalu od 0 do 1, što ju čini pogodnom za modeliranje vjerojatnosti. Derivacija sigmoidalne funkcije uvijek se može izraziti u smislu same funkcije, što omogućava jednostavniji postupak računanja gradijenta u postupku propagacije pogreške unatrag.

Sigmoidalna funkcija je aktivacijska funkcija koja se definira kao:

$$f(x) = \frac{1}{1+e^{-x}} \tag{3}$$



Slika 3. Graf sigmoidalne funkcije, preuzeto iz [7]

A njena derivacija je:

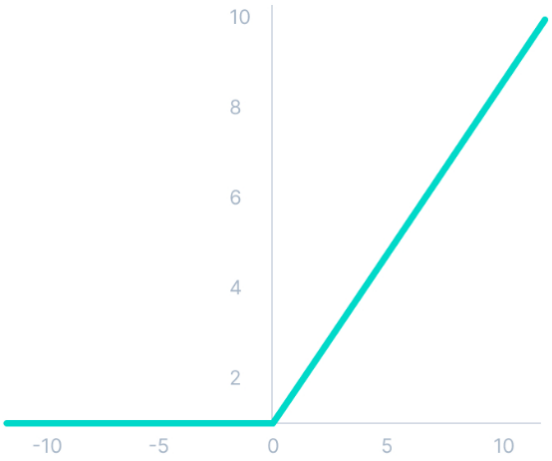
$$f'(x) = f(x)(1 - f(x)) \tag{4}$$

Zglobnica

Zglobnica ili ReLU je aktivacijska funkcija koja uzima vrijednost 0 za sve negativne vrijednosti ulaza, a za pozitivne vrijednosti ulaza vraća iste. Funkcija zglobnice je neprekidna, a njezina derivacija postoji u svim točkama osim u $x = 0$. Zglobnica je jedna od često korištenih aktivacijskih funkcija zbog brze računalne izvedbe, no također i zbog njenog svojstva da rješava problem nestajućeg gradijenta. Ograničenje zglobnice vidljivo je kada velik broj neurona ima negativne ulaze pa stoga, s obzirom da zglobnica negativne ulaze preslikava u nulu, u toj situaciji ti neuroni ne prenose informacije dalje.

Zglobnica je aktivacijska funkcija koja se definira kao:

$$f(x) = \begin{cases} 0, & x < 0 \\ x, & x \geq 0 \end{cases} \tag{5}$$



Slika 4. Graf zglobnice, preuzeto iz [7]

A njena derivacija je:

$$f'(x) = \begin{cases} 0, & x < 0 \\ 1, & x \geq 0 \end{cases} \tag{6}$$

GELU

GELU (Gaussian Error Linear Unit) je aktivacijska funkcija koja je glatka i kontinuirana te njezina derivacija postoji u svim točkama. U usporedbi s ReLU funkcijom, GELU zadržava veću količinu informacije u negativnom rasponu vrijednosti.

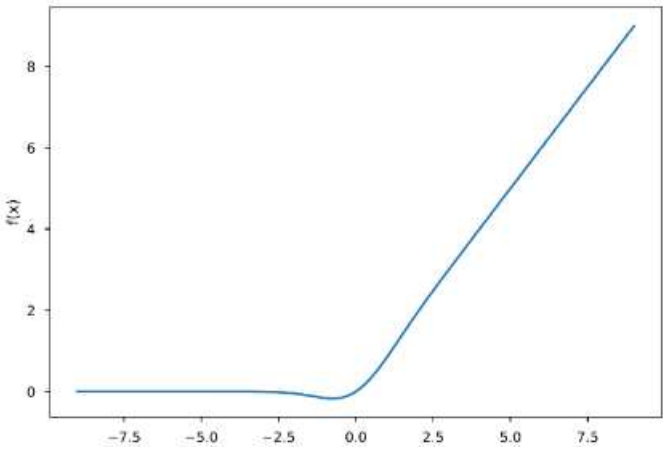
GELU funkcija se definira kao:

$$f(x) = xP(X \leq x) = x\varphi(x) \tag{7}$$

Gdje je $\varphi(x)$ kumulativna distribucijska funkcija standardne normalne distribucije. Ona se koristi za izračunavanje vjerojatnosti da će slučajna varijabla iz standardne normalne distribucije biti manja ili jednaka određenoj vrijednosti x. Kumulativna distribucijska funkcija pruža informaciju o tome koliki je udio vrijednosti manjih ili jednakih od određenog x.

GELU aktivacijska funkcija se aproksimira kao:

$$f(x) = 0.5x\left(1 + \tanh\left[\sqrt{\frac{2}{\pi}}(x + 0.044715x^3)\right]\right) \tag{8}$$



Slika 5. Graf GELU funkcije, preuzeto iz [22]

A njena derivacija je:

$$f'(x) = \varphi(x) + xP(X = x) \tag{9}$$

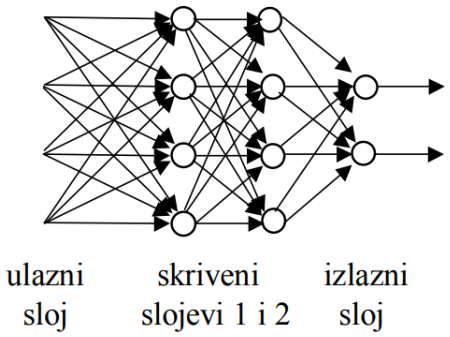
Gdje je $P(X = x)$ funkcija gustoće vjerojatnosti.

1.1.3. Propagacija pogreške unatrag

Propagacija pogreške unatrag je metoda koja se koristi prilikom učenja umjetnih neuronskih mreža kako bi se težine mreže prilagodile na način da razlika (pogreška) između predviđenog i stvarnog rezultata bude što manja. U procesu učenja mreže, modelu se daje ulazna slika na temelju koje on generira izlaz. Izlazne vrijednosti se zatim uspoređuju sa stvarnim vrijednostima i računa se razlika među njima koja predstavlja pogrešku predviđanja. Algoritmom propagacije pogreške unatrag pogreška se propagira kroz model od izlaza prema ulazu na način da se za svaku težinu u slojevima određuje koliko je ta težina doprinijela ukupnoj pogrešci te se, u skladu s time, vrijednosti težina ažuriraju kako bi se pogreška smanjila. Ovaj postupak ponavlja se više puta kroz učenje kako bi mreža davala predviđanja što sličnija stvarnim rezultatima.

1.2. Višeslojni perceptron

Višeslojni perceptron je vrsta neuronske mreže građena od više slojeva neurona. Mreža je sagrađena od ulaznog, izlaznog i jednog ili više skrivenih slojeva. U svakom sloju skup neurona prima i obrađuje ulazne podatke i prosljeđuje ih u slijedeći sloj. Svi neuroni u slojevima povezani su sa svim susjednim neuronima u prethodnom i slijedećem sloju kao što je vidljivo u prikazu perceptrona na slici 6. Neuron obrađuje podatke kao što je opisano u 1.1.1 te na dobivene vrijednosti primjenjuje aktivacijsku funkciju kako bi se uvela nelinearnost u model.



Slika 6. Višeslojni perceptron sa 4 neurona u svakom sloju, osim u izlaznom koji se sastoji od 2 neurona, preuzeto iz [18]

1.3. Konvolucijske neuronske mreže

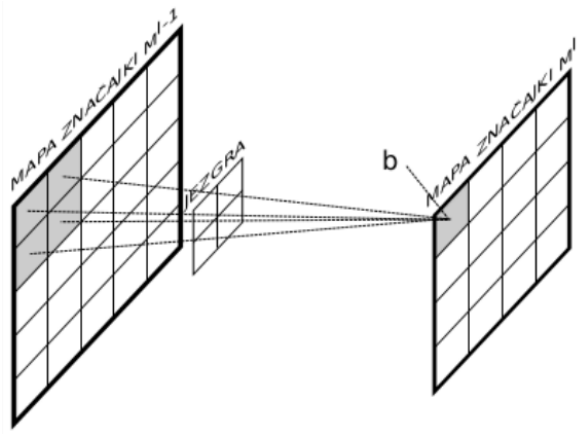
Konvolucijske neuronske mreže su pogodne za klasifikaciju podataka koji se prikazuju kao mreža točaka koje su međusobno povezane, što predstavlja izazov za obične neuronske mreže jer takvi podaci su visoko dimenzionalni i sadrže mnogo značajki. One su dizajnirane na način da mogu efikasno obraditi te podatke i izdvojiti relevantne informacije, što ih čini idealnim za obradu slika, zvukova ili senzorskih podataka. Konvolucijske mreže koriste kovolucijske slojeve za lokalno izdvajanje značajki transformiranjem ulaznih podataka sa nizom naučljivih filtera. Različitim filtrima mreža može naučiti izdvajati različite značajke iz slike. Slojevima za uzorkovanje smanjuje se veličina ulaznih podataka tako da se uzima prosječna ili maksimalna vrijednost u lokalnom području. Tim postupkom se smanjuju prostorne dimenzije ulaza, ali se zadržavaju njihove bitne značajke. Konvolucijska mreža dijeli naučljive parametre u svojim konvolucijskim slojevima, čime se smanjuje ukupan broj parametara u mreži. Ova svojstva čine konvolucijsku mrežu računalno učinkovitijom i lakšom za treniranje te joj omogućavaju da efikasnije obrađuje nove podatke.

1.3.1. Konvolucijski sloj

Konvolucijski sloj primjenjuje operaciju konvolucije na ulazne podatke kako bi se izlučile značajke ulaza. Za izlučivanje značajki koriste se filtri koji su predstavljeni kao matrice težina i čiji se parametri uče tijekom treninga. Filter prelazi preko visine i širine ulaznog podatka i na svakoj prostornoj poziciji se računa skalarni produkt između elemenata filtra i ulaznog podatka. Prelaskom filtra preko cijele ulazne slike generira se izlazna mapa značajki koja predstavlja mjeru prisutnost neke značajke na svakom od lokalnih područja ulazne slike.

Filter je dvodimenzionalan i prostorne dimenzije su mu najčešće puno manje od ulazne slike. On se pomiče za unaprijed određeni pomak (eng. *stride*) i po potrebi se ulazna slika nadopunjava (eng. *padding*) kako se nebi izgubile informacije na rubovima.

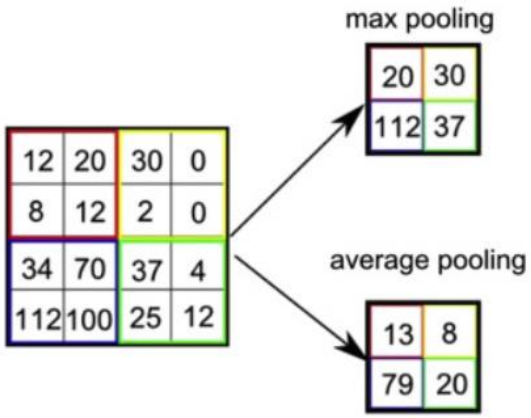
Slika 7 prikazuje konvoluciju koju možemo prikazati kao jezgru (filter) koja prelazi preko ulazne mape značajki i za svako lokalno područje skalarnim produktom sa tim lokalnim područjem generira izlaz. Možemo reći da svaki piksel izlaza ovisi o lokalnom području piksela ulaza na koje je primijenjen filter.



Slika 7. Konvolucijski sloj, preuzeto iz [3]

1.3.2. Sloj sažimanja

Sloj sažimanja (eng. *pooling layer*) je sloj koji preslikava prostorno bliske značajke iz ulaznih podataka u jednu značajku izlaza. On se često koristi u konvolucijskim neuronskim mrežama s ciljem da se smanji dimenzionalnost prostora ulaznih vrijednosti radi smanjenja potrebne računalne snage za obradu podataka bez gubitka informacija. Sloj sažimanja dijeli ulazne podatke na manja područja kao što je prikazano na slici 8 i nad svakim od područja obavlja funkciju sažimanja. Na slici su prikazane srednja vrijednost (eng. *average pooling*) i maksimalna vrijednost (eng. *max pooling*). Srednja vrijednost za svako područje ulaznog podatka računa prosječnu vrijednost svih podataka na tom području i tu vrijednost postavlja na odgovarajuće mjesto u izlazu. Maksimalna vrijednost za svako područje ulaznog podatka pronalazi najveću vrijednost na tom području i ta ona se postavlja na odgovarajuću poziciju na izlazu.



Slika 8. Sloj sažimanja, preuzeto is [16]

1.4. Transformeri

Transformeri predstavljaju arhitekturu dubokih modela koja se temelji na slojevima pažnje. Originalni transformer model sastoji se od kodera i dekodera. Zadaća svakog sloja kodera je odrediti značajke koje pokazuju koji dijelovi ulaznih podataka su relevantni za određene ostale dijelove ulaznih podataka, dok dekoder kao ulaz dobiva kontekstualne informacije iz kodera i na temelju njih generira izlaz.

Ulaz u transformer razlikuje se od ulaza konvolucijskih mreža. Konvolucijske mreže rade sa ulaznim podacima u obliku višedimenzionalnih matrica, gdje su za obradu slika tipični ulazi trodimenzionalne matrice kojima su dimenzije visina, širina i dubina slike. Zbog svog načina obrade ulaznih podataka konvolucijske mreže su efikasne u prepoznavanju prostornih i lokalnih ovisnosti na slikama. S druge strane, Transformeri koji se koriste za segmentaciju slika slike prvo pretvaraju u niz tokena. Svaki token predstavlja dio slike koji se tretira kao jedan element niza. Za razliku od konvolucijskih neuronskih mreža, transformeri ne prepoznaju lokalne prostorne ovisnosti različitih tokena ulaznog niza. Kako bi se zadržale prostorne informacije unutar slike, oni koriste pozicijska ugrađivanja (eng. *positional embeddings)* i slojeve pažnje.

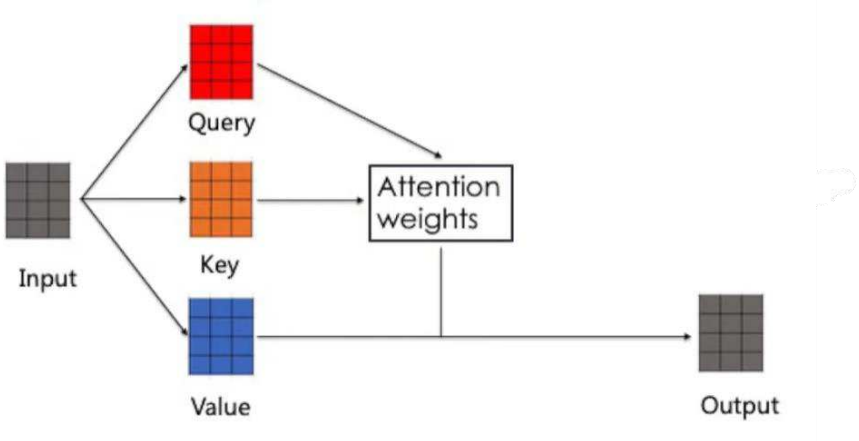
1.4.1. Slojevi pažnje

Mehanizmi slojeva pažnje mreži omogućavaju mreži da se fokusira na važne elemente ulaznih podataka prilikom procesiranja i donošenja predviđanja. Ti mehanizmi najčešće se koriste u modelima za obradu prirodnog jezika.

Primjer takve uporabe je prevođenje rečenice s hrvatskog na engleski jezik, gdje su veze između riječi ključne pa tako njihova međusobna ovisnost utječe na rezultate prevođenja. U obradi prirodnog jezika se, recimo, za svaku riječ računaju vrijednosti koje određuju koliko je ona relevantna za svaku drugu riječ u rečenici, pa tako slojevi pažnje daju veću težinu riječima koje su važnije za onu riječ koju trenutno prevodimo. U obradi slika, pažnja se primjenjuje na same piksele ili područja unutar slike.

Za obradu podataka u slojevima pažnje ključne su tri komponente: upiti (eng. *Queries*), ključevi (eng. *keys*) i vrijednosti (eng. *values*). Ulazni podaci u sloj predstavljeni su kao tenzori ili vektori značajki koji predstavljaju kanale ili dijelove slike. Na svaki kanal

primjenjuje se linearna transformacija kojom se generiraju vektori upita, ključeva i vrijednosti. U idućem koraku računa se sličnost između svih parova upita i ključeva njihovim matričnim umnoškom, te se na izlaznu matricu primjenjuje *softmax* funkcija za generiranje skupa težina koje reprezentiraju relevantnost svakog ključa za svaki upit. Te težine se zatim množe s vrijednostima i predstavljaju izlaz iz slojeva pažnje. Opisani postupak je prikazan na slici 9.



Slika 9. Sloj pažnje, preuzeto iz [17]

Funkcija kojom se računa izlaz sloja pažnje je definirana kao:

$$paznja(q,k,v) = softmax(\frac{qk^T}{\sqrt{d_k}})v \tag{10}$$

Gdje je d_k dimenzija ključeva.

1.4.2. Pozicijsko kodiranje

Kod transformerskih modela slojevi pažnje računaju važnost svakog elementa u ulaznim podacima za predviđanje, ali ne uzimaju u obzir redoslijed tih podataka na početnoj slici. Kako bi se uzela u obzir informacija o prostornom rasporedu, koristi se pozicijsko kodiranje (eng. *position embedding*). Pozicijsko kodiranje je tehnika koja se koristi kako bi svakom elementu ulaznog niza bila dodijeljena reprezentacija njegove pozicije u nizu. Ova tehnika omogućava modelu da razumije prostorni odnos objekata koji se nalaze na slici ili u videu

koji se obrađuje. Svakoj poziciji u ulaznom podatku pridružena je jedinstvena reprezentacija u obliku vektora čije vrijednosti opisuju položaj tog elementa u odnosu na druge elemente u ulaznom nizu. Model tako može naučiti razlikovati objekte koji su blizu jedan drugome ili koji se nalaze na određenim lokacijama na slici. Vektori pozicijskog kodiranja se mogu učiti tijekom treninga ili se mogu koristiti sinusoidalne funkcije pomoću kojih se generira jedinstvena oznaka za svaku poziciju u ulaznom podatku.

1.5. Semantička segmentacija

Segmentacija u računalnom vidu je proces u kojem se svakom pikselu unuar slike pridružuje odgovarajuća kategorija. Ova tehnika omogućuje računalima da obradom slika razumiju njihov sadržaj na razini pojedinačnih objekata i njihovih dijelova. Razlikujemo semantičku segmentaciju i segmentaciju instanci. Segmentacija instanci klasificira svaki piksel na slici i prepoznaje kojem konkretnom objektu na slici taj piksel pripada što je vidljivo na slici 10 (desno) gdje je svakom objektu na slici pridružena maska, a pozadina nema pridjeljenu masku. Za razliku od segmentacije instanci, semantička segmentacija ne razlikuje pojedine objekte što je vidljivo na slici 10 (lijevo), gdje je svakom pikselu na slici pridjeljena određena klasa i zbog toga maska prokriva cijeli ulazni podatak. U ovom radu bavit ću se semantičkom segmentacijom.

Slika 10. Semantička segmentacija (lijevo) i segmentacija instanci (desno), preuzeto iz [7]

2. Skup podataka

2.1. Synapse multi-organ

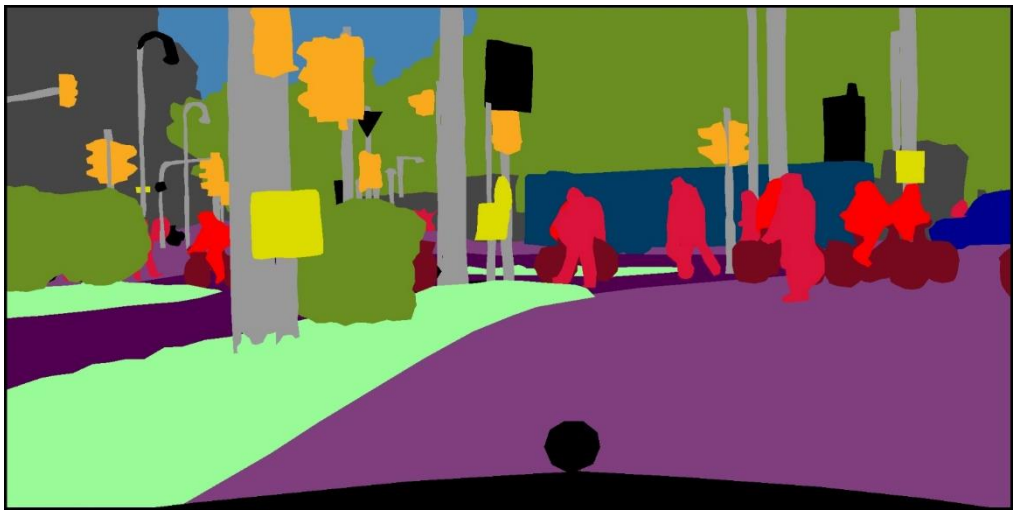
Skup podataka korišten za treniranje i testiranje u ovom radu je skup podataka korišten na natjecanju MICCAI 2015 Multi-Atlas Abdomen Labeling Challenge. Pod nadzorom Institutional Review Board-a (IRB) odabrano je 50 CT skenova abdomena iz snimaka kontinuiranog kliničkog ispitivanja kemoterapije za kolorektalni karcinom i retrospektivne studije ventralne hernije. Standardni registracijski podaci generirani su pomoću Nifty-Reg-a. U skupu podataka 13 organa je ručno označeno na snimkama od strane radiologa koristeći MIPAV softver. Tih 13 organa uključuju: desni bubreg, lijevi bubreg, žučni mjehur, jednjak, jetra, želudac, aorta, donja šuplja vena, portalna vena, slezena, gušterača, desna nadbubrežna žlijezda i lijeva nadbubrežna žlijezda. Na nekim od snimaka nema definiranog desnog bubrega ili žučnog mjehura jer ga pacijenti na kojima je provedeno istraživanje nemaju. S obzirom da u originalnom skupu postoji 13 kategorija u koje nije uključena kategorija pozadine, u svrhu zadatka semantičke segmentacije 5 kategorija (jednjak, donja šuplja vena, portalna vena, desna nadbubrežna žlijezda i lijeva nadbubrežna žlijezda) se tretira kao kategorija pozadine, dok su preostalih 8 kategorija zadržane kao zasebne.

S obzirom da maske skupa slika za testiranje nisu javno dostupne, prikazani su rezultati na skupu za validaciju. Skup za validaciju u ovom radu izdvojen je iz skupa za trening tako da se skup za trening sastojao od 2211 slika, a skup za validaciju od 1568 slika, po uzoru na [2].

2.2. Cityscapes

Cityscapes je veliki skup podataka koji se često koristi u zadacima semantičke segmentacije. Skup prikazuje gradske ulične scene u 50 njemačkih gradova kroz nekoliko mjeseci u godini u proljeće, ljeto i jesen u povoljnim vremenskim uvjetima. Slike su dimenzija 1024×2048 piksela. Za potrebe ovog rada korišten je podskup gtFine preuzet sa službene stranice [23]. GtFine se sastoji od fino označenog skupa za trening sa 2975 slika, skupa za validaciju sa 500 slika i skupa za testiranje sa 1525 slika. Skup sadrži 33 klase za objekte na slikama, a u ovom radu je korišteno 19 osnovnih . Neke od klasa su automobil, kamion, semafor, čovjek, zgrada…

Slika 11. Primjer segmentacijske maske Cityscapes



Slika 12. Primjer segmentacijske maske Cityscapes

3. Model

3.1. SegFormer

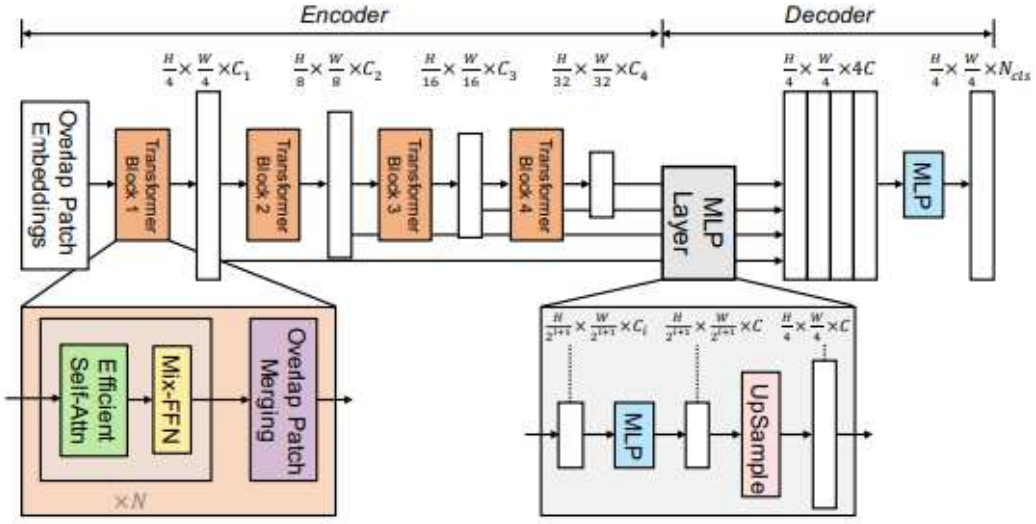
Model korišteni u ovom radu je model predstavljen u SegFormer: Simple and Efficient Design for Semantic Segmentation with Transformers verzije SegFormer-B0. SegFormer model je arhitektura dubokog učenja koja objedinjuje transformatorsku arhitekturu s potpuno povezanim modulom(MLP). Ovaj model koristi hijerarhijski dizajn s koderom i dekoderom. Koder koristi transformatorske slojeve za globalni kontekst i višeskalne značajke iz ulazne slike. Dekoder agregira informacije iz različitih slojeva, te tako kombinira globalnu i lokalnu pažnju. Arhitektura SegFormera koristi relativno kodiranje pozicija i povezivanje između kodera i dekodera kako bi se obuhvatili detalji ulaznih slika i zadržala prostorna konzistentnost u segmentacijskim predikcijama.

Kralježnica (eng. *backbone*) modela se sastoji od transformerskih blokova kod kojih se na ulazu u blok slike pretvaraju u okna (eng. *patch*) pomoću konvolucijskog sloja. Nakon pretvaranja u okna slijedi jedan ili više manjih blokova koji se sastoje od sloja pažnje i sloja pozicijskog ugrađivanja. U sloju pažnje izlaz se računa prema postupku opisanom u 1.4. Izlaz sloja pažnje prosljeđuje se sloju pozicijskog ugrađivanja koji se sastoji od dva potpuno povezana modula između kojih se koristi 3×3 konvolucija i aktivacijska funkcija. 3×3 konvolucija omogućava da se umjesto pozicijskog kodiranja koristi popunjavanje nulama (eng. *zero padding*) kako se ne bi izgubile informacije o prostornim ovisnostima. Na ovu arhitekturu može se gledati i kao na spoj transformerske i konvolucijske arhitekture jer se jednadžba sloja pozicijskog ugrađivanja (11) može predstaviti kao rezidualna konvolucija s uskim grlom jer je MLP zapravo 1×1 konvolucija. Također, između ulaza u sloj pozicijskog ugrađivanja i izlaza dodaje se izravna veza kako bi se zadržale informacije o prostornim ovisnostima. Sloj pozicijskog ugrađivanja može se definirati na slijedeći način:

$$X_{izlaz} = \text{MLP}\,(\text{GELU}\,(\text{Conv3}\times3\,(\text{MLP}\,(x_{ulaz})))) + x_{ulaz} \qquad (11)$$

Izlazi iz svakog od transformerskih blokova se prosljeđuju dekoderu. S obzirom da su mape značajki na izlazu iz svakog od transformerskih blokova drugačijih dimenzija, potrebno ih je prvo svaku zasebno proslijediti na ulaz potpuno povezanog sloja, te zatim

povećano uzorkovati do dimenzije $\frac{H}{4} \times \frac{W}{4}$, gdje su H i W visina i širina ulaznog podatka. Te mape značajki se u dekoderu konkateniraju, prosljeđuju potpuno povezanom modulu radi spajanja dimenzija kanala konkateniranih mapa značajki te se izlaz iz njega prosljeđuje završnoj 1×1 konvoluciji koja stvara segmentacijsku masku. Opisana arhitektura SegFormera prikazana je na slici 13 na kojoj je prikazan SegFormer koji se sastoji od 4 transformerska bloka od koji se svaki od transformerskih blokova sastoji od pretvaranja ulaza u okna i jednog manjeg bloka sloja pažnje i pozicijskog ugrađivanja. Na poveznici između transformerskog bloka i dekodera prikazana je dimenzija proslijeđenog podatka. Potpuno povezani modul je na slici označen sa FC i možemo vidjeti da se nakon povećanog uzorkovanja mape značajki konkateniraju i nakon prolaska kroz još jedan potpuno povezani sloj i konvoluciju na izlazu dobivamo segmentacijsku masku.



Slika 13. arhitektura modela Segformer preuzeta iz [1]

4. Programska izvedba

Za potrebe programske izvedbe ovog rada korišten je projekt otvorenog koda MMSegmentation [20] za semantičku segmentaciju u računalnom vidu. Projekt je izgrađen na platformi open-mmlab (MultiMediaLab). MMSegmentation je napisan u Pythonu i koristi biblioteke kao što su PyTorch, torchvision, mmcv, numpy i sickit-image. Projekt pruža bogatu kolekciju prethodno konfiguriranih modela i komponenti, sa alatima za učitavanje, predobradu, treniranje i evaluaciju podataka . MMSegmentation koristi iterativno orijentiran trening (eng. *iteration based runner*) pa će stoga u nastavku duljina treninga biti izražena u broju iteracija. Model je treniran na računalu sa grafičkom procesnom jedinicom NVIDIA GeForce GTX 1650.

Skupovi podataka su preuzeti sa javno dostupnih repozitorija za Cityscapes [5] i Synapse [6] te su pripremljeni pomoću skripti dostupnih na javnom repozitoriju MMSegmentationa [20].

Za vizualizaciju maski predviđanja i iscrtavanje grafa dice vrijednosti korišteni su moduli MMSegmentation dostupni u javnom repozitoriju [20].

Upute za korištenje MMSegmentationa kako bi se ponovio moj eksperiment mogu se pronaći na [24].

5. Eksperimentalni rezultati

U prvom dijelu eksperimentalnog rada model SegFormer-B0 koji je predtreniran na skupu ImageNet i zatim naučen na skupu Cityscapes koji je opisan u 2.2. validiramo na skupu Cityscapes. Model je treniran od strane autora modela nabrojanih u [1] te su težine modela preuzete sa javnog repozitorija mmsegmentation platforme[20].

U drugom dijelu SegFormer-B0 treniramo na skupu podataka Synapse opisanom u 2.1. Model treniramo 3 puta uz različite parametre. Kod prvog učenja je veličina grupe 2, a učenje je trajalo 20000 iteracija, kod drugog učenja korišteni su isti parametri kao i kod prvog, ali uz inicijalizaciju težina na težine modela predtreniranog na ImageNet skupu. Kod trećeg učenja težine su inicijalizirane nasumično te je učenje trajalo 80000 iteracija uz veličinu grupe 4.

5.1. Mjere točnosti

Mjere točnosti su brojčane metrike pomoću kojih ocjenjujemo koliko dobro model radi na način da uspoređuju dobivene rezultate sa očekivanima. Ove metrike omogućuju objektivnu procjenu performansi, što nam je potrebno kako bismo mogli usporediti uspješnost obavljanja zadatka pojedinih modela i izabrati onaj koji daje najoptimalnije rezultate.

5.1.1. Dice koeficijent

Dice koeficijent predstavlja mjeru preklapanja između izlazne segmentacijske maske i stvarne maske. Sličnost te dvije mape se računa na temelju presjeka i unije segmentiranih područja. Kao mjera točnosti u ovom radu za evaluaciju korišten je srednji dice koeficijent koji je izračunat kao prosječna vrijednost dice koeficijenata svih klasa.

Dice koeficijent sličnosti za skupove X i Y je definiran kao:

$$DSC = \frac{2|X \cap Y|}{|X| + |Y|} \tag{12}$$

5.1.2. Srednja točnost

Srednja točnost predstavlja ukupnu točnost u pikselima izlazne segmentacijske maske u usporedbi s pravom maskom.

Srednja točnost se definira kao:

$$SrednjaTocnost = \frac{1}{K}\sum_{i=0}^{K}\frac{n_{ii}}{t_i} \tag{13}$$

Gdje je n_{ii} reprezentira broj točno pozitivno klasificiranih piksela klase i, a t_i označava ukupan broj piksela.

5.1.3. mIoU

Omjer presjeka i unije (engl. *Intersection over Union*, IoU) je metrika kojom se mjeri koliko mreža može precizno segmentirati objekte na slici. Mjera se računa tako da se podijeli veličina preklapanja predviđene i stvarne segmentacijske mase sa unijom predviđene i stvarne maske. IoU metrika se mjeri u intervalu od 0 do 1, gdje 1 predstavlja potpuno preklapanje, a 0 da nema nimalo preklapanja.

Mjera uspješnosti se izražava kao srednji omjer presjeka i unije (engl. *mean intersection over union*, mIoU), što se računa tako da se zbroje omjeri presjeka i unije za sve klase i to se podijeli sa brojem klasa.

Omjer presjeka i unije se definira kao:

$$mIoU = \frac{1}{K}\sum_{i=0}^{K}\frac{TP_i}{FP_i+FN_i+TP_i} \tag{14}$$

Gdje je TP broj točno pozitivno klasificiranih piksela, FN broj netočno negativnih i FP broj netočno pozitivnih, a i predstavlja indeks klase.

5.2. Rezultati

Kao pripremu za eksperimentalni dio rada model SegFormer-B0 koji je predtrenirane na skupu ImageNet i zatim naučen na skupu Cityscapes od strane autora modela [1] validiramo na skupu podataka Cityscapes. Model je od strane autora modela treniran kroz 160000 iteracija uz veličinu grupe 4. Korišten je optimizator Adam gdje se stopa učenja kroz prvih 0.9% od ukupnog broja iteracija linearno povećava od 1e-6 do 0.00006, te se zatim do kraja treninga polinomijalno smanjuje do minimalne vrijednosti 0.0. Koder se sastoji od 4 transformerska bloka te su veličine okna u svakom od blokova redom 7, 3, 3, 3, pomak kod ugrađivanja okna 4, 2, 2, 2, te broj kanala izlaza 32, 64, 160, 256. Broj slojeva kodera je redom 2, 2, 2, 2, omjer smanjenja (eng. *reduction ratio*) 8, 4, 2, 1, i broj glava u slojevima pažnje 1, 2, 5, 8. Prigušenje težina (engl. *weight decay*) postavljeno je na 0.01.

Kako bi se spriječila prenaučenost i poboljšala sposobnost generalizacije modela autor modela je slike za učenje nasumično rotirao sa vjerojatnošću 50% za maksimalni kut od 20 stupnjeva i nasumično zrcalio s vjerojatnošću 50%.

	Dice	**Točnost**	**IoU**
Srednja vrijednost	86.06	83.86	76.54

Tablica 1. Validacija modela SegFormer-B0 treniranog od strane autora modela na skupu Cityscapes, parametri su preuzeti

Klasa	Dice	Točnost	IoU
kolnik	99.00	98.95	98.03
nogostup	91.32	91.92	84.03
zgrada	95.96	96.73	92.23
zid	74.02	65.28	58.76
ograda	72.47	66.19	56.82
stup	76.90	72.51	62.59
semafor	82.23	81.00	69.82
prometni znak	87.46	84.88	77.72
vegetacija	96.13	96.64	92.55
zemljište	78.15	73.71	64.13
nebo	97.43	98.22	94.98
čovjek	89.33	90.19	80.72
vozač	72.93	69.63	57.39
automobil	97.08	97.59	94.33
kamion	81.91	73.78	69.36
autobus	91.23	89.76	83.88
vlak	86.14	81.34	75.66
motocikl	78.74	77.38	64.93
bicikl	86.54	87.73	76.27

Tablica 2. Validacija modela SegFormer-B0 treniranog od strane autora modela na Cityscapes skupu po pojedinačnoj klasi, parametri su preuzeti sa [25]

U drugom dijelu eksperimenta treniramo SegFormer na Synapse skupu. Kod svakog od učenja za koja ću navesti rezultate korišteni su slijedeći parametri koji su isti u svakom učenju na Synapse skupu.

Koristimo Adam optimizator, a stopa učenja se kroz prvih 0.9% od ukupnog broja iteracija linearno povećava od 1e-6 do 0.00006, te se zatim do kraja treninga polinomijalno smanjuje do minimalne vrijednosti 0.0. Za prigušenje težina (eng. *weight decay*) postavljena je vrijednost 0.01, a kao funkcija gubitka korištena je unakrsna entropija. Koder se sastoji od 4 transformerska bloka te su veličine okna u svakom od blokova redom 7, 3, 3, 3. Pomak kod ugrađivanja okna je redom 4, 2, 2, 2, a broj kanala izlaza 32, 64, 160, 256. Broj slojeva kodera je redom 2, 2, 2, 2 u transformerskim blokovima, omjer smanjenja (eng. *reduction ratio*) 8, 4, 2, 1,a broj glava u slojevima samopažnje 1, 2, 5, 8. Postotak ispuštenih neurona (eng. *dropout ratio*) 0.1. Svi ostali hiperparametri koji nisu navedeni postavljeni su na vrijednosti kao u radu autora modela [1].

Najveća moguća veličina grupe je na resursima koji su mi bili dostupni je 4.

Kako bi se spriječila prenaučenost i poboljšala sposobnost generalizacije modela koristimo nasumično rotiranje slika za vrijeme treninga sa vjerojatnošću 50% za maksimalni kut od 20 stupnjeva. Rezolucija ulazne slike je 512×512 piksela.

5.2.1. Učenje kroz 20000 iteracija uz nasumično inicijalizirane težine

SegFormer treniramo na Synapse skupu 20000 iteracija uz veličinu grupe 2. Kod učenja uz inicijalizaciju nasumičnim težinama učenje je trajalo 2 sata i 2 minute . Brzina obrade slike 33.91 slike / s. Za učenje je korišteno 2182 MiB memorije grafičke procesne jedinice.

5.2.2. Učenje kroz 20000 iteracija uz težine inicijalizirane parametrima modela predtreniranog na ImageNet

U idućem dijelu eksperimenta je SegFormer treniramo na Synapse skupu 20000 iteracija uz veličinu grupe 2. Za inicijalizaciju su koristimo težine predtreniranog modela na ImageNet preuzete iz javnog repozitorija mmsegmenatation platforme, a objavljene od strane autora modela.

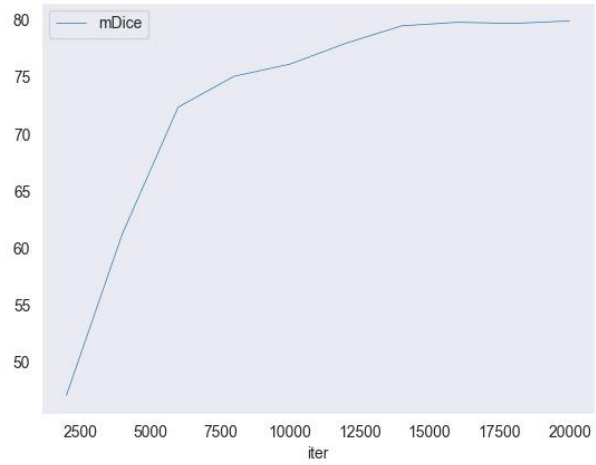
Učenje je trajalo 2 sata i 2 minute. Brzina obrade slike iznosi 33.16 slike / s. Za učenje je bilo potrebno 2328 MiB memorije grafičke procesne jedinice.

Inicijalizacija težina	**Dice koeficijent [%]**	**Srednji omjer presjeka i unije (mIoU) [%]**	**Srednja točnost [%]**
nasumično	80.09	69.36	78.30
ImageNet	79.94	69.13	77.24

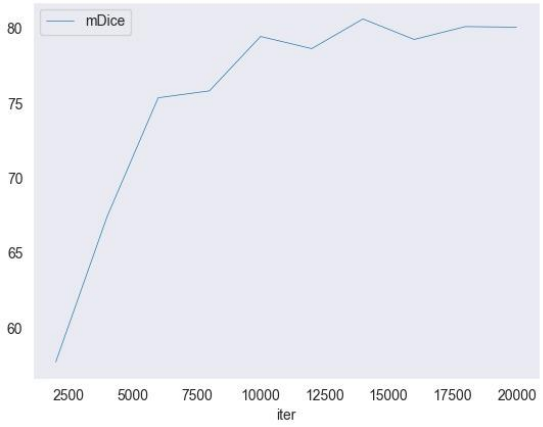
Tablica 3. Rezultati modela koji je predtreniran na ImageNet(P) i nije predtreniran(N), oba trenirani kroz 20000 iteracija, veličina grupe 2

	Dice		**Točnost**		**IoU**	
Klasa	**P**	**N**	**P**	**N**	**P**	**N**
aorta	78.02	**82.67**	69.47	**77.48**	63.96	**70.46**
žučni mjehur	62.63	**64.79**	67.29	**71.42**	45.59	**47.92**
lijevi bubreg	**90.37**	88.36	89.79	**90.90**	**82.44**	79.15
desni bubreg	**82.87**	82.04	**79.79**	78.90	**70.76**	69.54
jetra	**94.83**	94.78	95.59	**96.07**	**90.17**	90.08
gušterača	**49.40**	47.99	**38.70**	38.64	**32.80**	31.57
slezena	88.83	**89.78**	**89.67**	89.18	79.91	**81.46**
želudac	**72.89**	70.81	**65.17**	64.42	**57.35**	54.82

Tablica 4. Rezultati modela koji je predtreniran na ImageNet(P) i nije predtreniran(N), oba trenirani kroz 20000 iteracija, veličina grupe 2



Slika 14. Srednji dice koeficijent izračunat nakon svakih 2000 iteracija za model predtreniran na ImageNet-u



Slika 15. Srednji dice koeficijent izračunat nakon svakih 2000 iteracija za model kojem su težine prije učenja nasumično inicijalizirane

Tablica 3 pokazuje da model uspješnije obavlja zadatak semantičke segmentacije ako se koriste nasumične vrijednosti za inicijalizaciju težina umjesto težina dobivenih treniranjem na ImageNet-u. Prosječna vrijednost dice koeficijenta je veća za 0.15%, prosječna točnost za 1.06%, a mIoU za 0.23%. Iz slike 14 i slike 15 koje prikazuju vrijednosti srednjeg Dice koeficijenta za vrijeme treninga može se vidjeti da model koji je predtreniran ima stabilniji rast vrijednosti tijekom učenja što je i očekivano jer predtrenirani modeli mogu "prenijeti" već stečeno znanje za neke značajke, čime se stabilizira učenje. Prilikom korištenja težina modela koji je predtreniran na ImageNet-u i zatim treniran na Synapse nije došlo do napretka u treningu, vjerojatno zato jer ImageNet sadrži slike čija je domena nije pretjerano bliska Synapse skupu.

5.2.3. Učenje kroz 40000 iteracija uz nasumično inicijalizirane težine

U idućem dijelu eksperimenta SegFormer treniramo na Synapse skupu 40000 iteracija uz veličinu grupe 4. Za inicijalizaciju su korištene nasumično generirane težine.

Učenje je trajalo 7 sati i 27 minuta. Brzina obrade slike i predikcija iznosi 34.17 okvira / s. Za učenje je bilo potrebno 3292 MiB memorije grafičke procesne jedinice.

U slijedećim tablicama bit će uspoređene performanse modela treniranog na Synapse skupu uz nasumično inicijalizirane težine kroz 20000 iteracija i grupu veličine 2 te 40000 iteracija i grupu veličine 4.

Broj iteracija	**Dice koeficijent [%]**	**Srednji omjer presjeka i unije (mIoU) [%]**	**Srednja točnost [%]**
20000	80.09	69.36	78.30
40000	81.31	70.77	79.48

Tablica 6. Rezultati Segformer-B0 treniranog na Synapse

	Dice		Točnost		IoU	
Klasa	**20k**	**40k**	**20k**	**40k**	**20k**	**40k**
aorta	82.67	**85.63**	77.48	**81.42**	70.46	**74.87**
žučni mjehur	64.79	**66.82**	**71.42**	71.40	47.92	**50.17**
lijevi bubreg	**88.36**	88.02	90.90	**91.81**	**79.15**	78.60
desni bubreg	82.04	**84.22**	78.90	**81.44**	69.54	**72.74**
jetra	94.78	**95.35**	96.07	**96.38**	90.08	**91.11**
gušterača	47.99	**52.75**	38.64	**41.12**	31.57	**35.82**
slezena	**89.78**	88.79	89.18	**91.28**	**81.46**	79.83
želudac	**70.81**	70.55	**64.42**	60.75	**54.82**	54.50

Tablica 7. Rezultati na pojedinačnoj klasi za model treniran na Synapse skupu uz nasumično inicijalizirane težine, 20k označava trening kroz 20000 iteracija, 40k kroz 40000 iteracija

Iz tablice 6 je jasno vidljivo da model puno uspješnije obavlja zadatak semantičke segmentacije nakon učenja kroz 40000 iteracija uz veličinu grupe 4 od modela naučenog kroz 20000 iteracija uz veličinu grupe 2. Prosječna vrijednost dice koeficijenta je veća za 1.22%, prosječna točnost za 1.41%, a mIoU za 1.18%, što je i očekivano. Na razini pojedinačne klase lijevi bubreg, slezena i želudac imaju veći IoU i Dice kod modela koji je kraće treniran no sa minimalnim razlikama u odnosu na model koji je dulje treniran, dok kod svih ostalih klasa model treniran kroz 40000 iteracija ima bolje rezultate te su najveće razlike vidljive za aortu (Dice – 2.96%, IoU – 3.94%) i gušteraču (Dice – 4.76%, IoU – 4.25%). Moguće je da je to posljedica toga što su gušterača i aorta manji organi, pa se rjeđe pojavljuju na slikama s obzirom da su slike plošne snimke rendgena abdomena, a također je njihova površina na slikama manja od površina ostalih organa, pa je modelu teže naučiti prepoznati te organe.

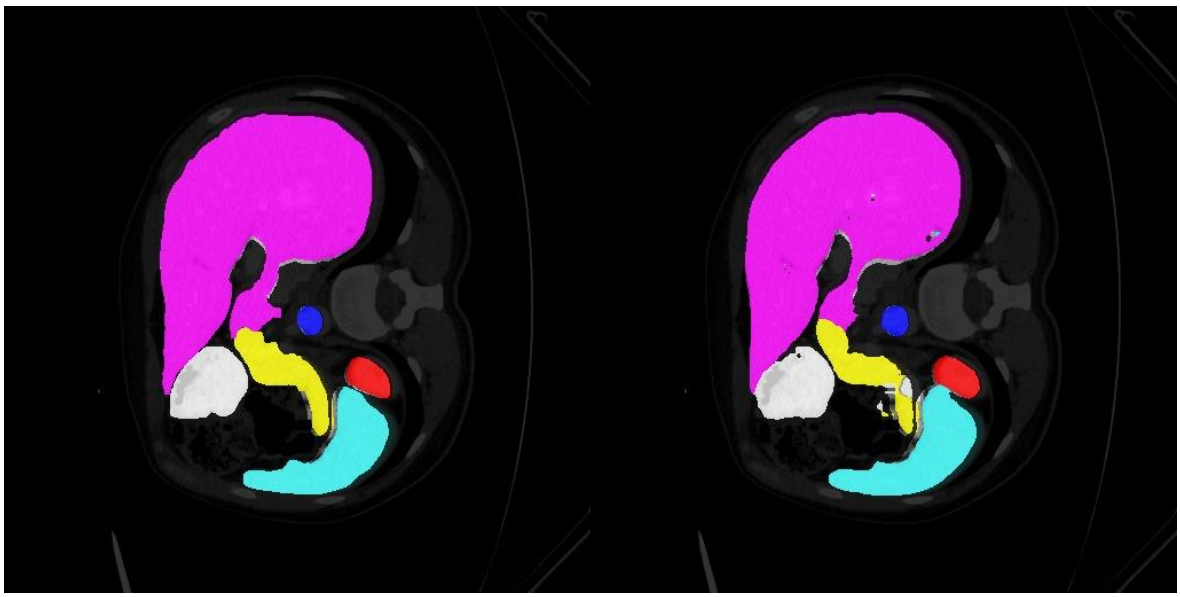
Usporedba rezultata na Synapse skupu sa drugim modelima

Model	Dice
nnUNet [9]	**88.80**
PHTrans [10]	**88.55**
nnFormer [11]	**86.57**
MERIT [12]	**84.90**
MISSFormer [2]	**81.96**
PVT-CASCADE [13]	81.06
FocalUNet [14]	80.81
SETR [15]	79.60
SegFormerB0	**81.31**

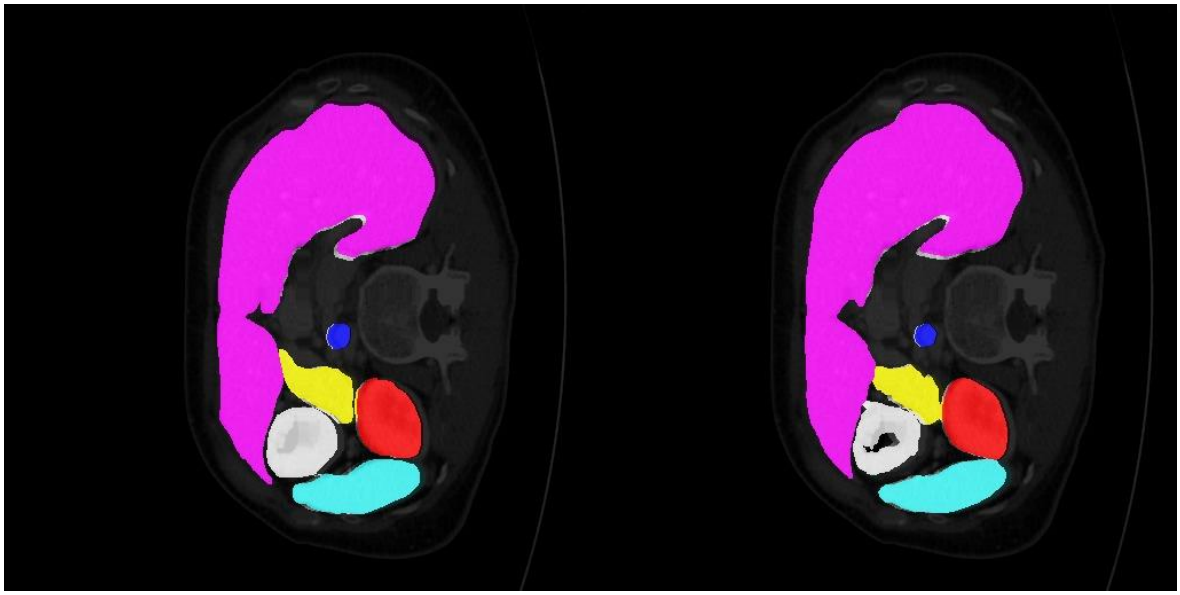
Tablica 8. Rezultati modela na Synapse-u

Na natjecanju MICCAI 2015 Multi-Atlas Abdomen Labeling Challenge opisanom u 2.1 su ponuđeni modeli s boljim rezultatima od SegFormera kojeg sam ja trenirala, no s obzirom da je zbog računalnih resursa dostupnih za ovaj eksperiment korišten najmanji model SegFormera (B0) , rezultati su zadovoljavajući.

Primjeri rezultata SegFormer B-0 treniranog kroz 20000 iteracija, inicijaliziranog nasumičnim težinama na Synapse skupu



Slika 16. Lijevo - oznake, desno – predviđanje



Slika 17. Lijevo - oznake, desno – predviđanje

Zaključak

Ovaj rad bavi se problemom semantičke segmentacije medicinskih rendgenskih snimaka. U radu su opisana osnovna načela dubokog učenja, konvolucijskih neuronskih mreža, trnsformerske arhitekture i slojeva pažnje, te semantičke segmentacije. Opisani su skupovi podataka Synapse i Cityscapes.

Također je predstavljen transformer model SegFormer i opisana arhitektura njegove verzije B0 koji postiže odlične rezultate u zadacima semantičke segmentacije.

U praktičnom dijelu već istrenirani model je preuzet sa javnog repozitorija i validiran na Cityscapes skupu, te je u drugom dijelu model uz različite hiperparametre učen na skupu Synapse. Prikazani su rezultati učenja su objašnjeni i uspoređeni sa rezultatima drugih radova koji su se bavili istim zadatkom.

Za praktični dio rada korištena je biblioteka MMSegmentation koja sadrži osnovne module za izgradnju modela, trening i validaciju, obradu podataka te vizualizaciju rezultata na temelju kojih su za potrebe ovog rada izrađeni konkretni moduli za učenje i validaciju.

Rezultati su pokazali da model treniran kroz 20000 iteracija inicijaliziran nasumičnim težinama postiže dice koeficijent veći za 0.15%, prosječnu točnost za 1.06% i mIoU za 0.23% od modela treniranog kroz 20000 iteracija uz težine inicijalizirane treniranjem na ImageNetu. Također je vidljivo da model treniran kroz 40000 iteracija uz grupu veličine 4 postiže dice koeficijent veći za 1.22%, prosječnu točnost za 1.41% i mIoU za 1.18% od modela treniranog kroz 20000 iteracija i grupu veličine 2, što je bilo i očekivano.

Detaljnijom analizom bi se moglo utvrditi zašto model predtreniran na ImageNet skupu postiže lošije rezultate od modela sa slučajno inicijaliziranim težinama, te bi se učenja mogla ponoviti uz drugačiju stopu učenja što bi možda moglo dati još bolje rezultate.

Literatura

[1] Xie, E., Wang, W., Yu, Z., Anandkumar, A., M. Alvarez, J., Luo, P. *SegFormer: Simple and Efficient Design for Semantic Segmentation with Transformers,* NeurIPS 2021.

[2] Huang, X., Deng, Z., Li, D., Yuan, X., *MISSFormer: An Effective Medical Image Segmentation Transformer,* arXiv preprint arXiv:2109.07162, 2021

[3] Siniša Šegvic Josip Krapac, Konvolucijski modeli, 'http://www.zemris.fer.hr/~ssegvic/du/du2convnet.pdf'. Pristupljeno: 25. svibnja 2023.

[4] Justin Johnson, Deep Learning for Computer Vision , 'https://web.eecs.umich.edu/~justincj/slides/eecs498/498_FA2019_lecture07.pdf', Pristupljeno: 29. svibnja 2023.

[5] https://www.cityscapes-dataset.com, Pristupljeno: 22. travnja 2023.

[6] https://www.synapse.org/#!Synapse:syn3193805/wiki/89480, Pristupljeno: 26. travnja 2023.

[7] https://www.v7labs.com/blog/neural-networks-activation-functions, pristupljeno 29. svibnja 2023.

[8] Justin Johnson, Deep Learning for Computer Vision , 'https://web.eecs.umich.edu/~justincj/slides/eecs498/WI2022/598_WI2022_lecture15.pdf', Pristupljeno: 29. svibnja 2023.

[9] Isensee, F., Petersen, J., Klein, A., Zimmerer, D., Jaeger, P.F., Kohl, S., Wasserthal, J., Köhler, G., Norajitra, T., Wirkert, S., & Maier-Hein, K.H., *nnU-Net: Self-adapting Framework for U-Net*-Based *Medical Image Segmentation,* 2018, arXiv:1809.10486

[10] Liu, W., Tian, T., Xu, W., Yang, H., Pan, X., Yan, S., & Wang, L., *PHTrans: Parallelly Aggregating Global and Local Representations for Medical Image Segmentation., arXiv* 2022, arXiv:2203.04568

[11] Zhou, H-Y., Guo, J., Zhang, Y., Han, X., Yu, L., Wang, L., & Yu, Y., *nnFormer: Volumetric Medical Image Segmentation via a 3D Transformer,* arXiv preprint arXiv:2109.03201, 2021.

[12] Rahman, M., Marculescu, R., *Multi-scale Hierarchical Vision Transformer with Cascaded Attention Decoding for Medical Image Segmentation,* arXiv preprint arXiv:2303.16892, 2023.

[13] Rahman, M., Marculescu, R., *Medical Image Segmentation via Cascaded Attention Decoding,* IEEE/CVF Winter Conf. Appl. Comput. Vis. (WACV), Waikoloa, HI, USA, Jan. 2023

[14] Naderi, M., Givkashi, M.H., Piri, F., Karimi, N., Samavi, S., *Focal-UNet: UNet-like Focal Modulation for Medical Image Segmentation,* arXiv:2212.09263, 2022.

[15] Zheng, S., Lu, J., Zhao, H., Zhu, X., Luo, Z., Wang, Y., Fu, Y., Feng, J., Xiang, T., Torr, P.H.S., & Zhang, L, *Rethinking Semantic Segmentation from a Sequence-to-Sequence Perspective with Transformers,* CVPR, 2021.

[16] https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53, Pristupljeno: 7. lipnja 2023.

[17] https://www.youtube.com/watch?v=5T38-2J5CcY, Pristupljeno: 7. lipnja 2023.

[18] https://www.aes.hr/_download/repository/06-ViseslojniPerceptron-1s.pdf, Pristupljeno: 8. lipnja 2023.

[19] Tang, X., Wang, W., Tu, Z., Liu, M., Li, D., Fan, X. *Automatic Detection of Coseismic Landslides Using a New Transformer Method,* Remote Sens.,2022.

[20] https://github.com/open-mmlab/mmsegmentation/tree/main, Pristupljeno: 7. lipnja 2023.

[21] https://alaaalatif.github.io/2019-04-11-gelu/, Pristupljeno: 7. lipnja 2023.

[22] https://towardsdatascience.com/the-concept-of-artificial-neurons-perceptrons-in-neural-networks-fab22249cbfc, Pristupljeno: 9. lipnja 2023.

[23] https://www.cityscapes-dataset.com/login/, Pristupljeno: 10. lipnja 2023.

[24] https://github.com/pecra/Zavrsni, Pristupljeno: 11. lipnja 2023.

[25] https://download.openmmlab.com/mmsegmentation/v0.5/segformer/segformer_mit-b0_8x1_1024x1024_160k_cityscapes/segformer_mit-b0_8x1_1024x1024_160k_cityscapes_20211208_101857-e7f88502.pth, Pristupljeno: 12. lipnja 2023.

Algoritmi za semantičku segmentaciju radioloških slika

Sažetak

Semantička segmentacije predstavlja važan zadatak računalnog vida koji se primjenjuje u mnogim područjima. U posljednje vrijeme veliki napredak u tom zadatku postižu modeli koji se temelje na slojevima pažnje. Ovaj rad razmatra primjene semantičke segmentacije vezane uz analizu medicinskih slika. U okviru rada predstavljen je i opisan transformerski model SegFormer te je provedena evaluacija na Cityscapes skupu slika i učenje i evaluacija na Synapse skupu slika. Dobiveni rezultati su objašnjeni i uspoređeni sa rezultatima drugih radova koji se bave istim zadatkom. U radu su opisane osnove neuronskih mreža, kovolucije, transformerskih modela i slojeva pažnje. Opisan je i javno dostupan projekt koji je korišten za potrebe programske implementacije.

Ključne riječi: računalni vid, semantička segmentacija, slojevi pažnje, SegFormer, transformer, kovolucijske neuronske mreže

Algorithms for Semantic Segmentation of Radiological Images

Abstract

Semantic segmentation represents an important task in computer vision that is applied in various fields. Recently, significant progress has been achieved in this task using models based on attention layers. This paper discusses the applications of semantic segmentation related to the analysis of medical images. Within the paper, the SegFormer transformer model is presented and described, followed by an evaluation on the Cityscapes dataset and training and evaluation on the Synapse dataset. The obtained results are explained and compared with the results of other works that address the same task. The basics of neural networks, convolutions, transformer models, and attention layers are described in the paper. Additionally, a publicly available project used for the purpose of program implementation is mentioned.

Keywords: computer vision, semantic segmentation, attention layers, SegFormer, transformer, convolutional neural networks