

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 10

**SEMANTIČKA SEGMENTACIJA ORGANA U SLIKAMA
MAGNETSKE REZONANCIJE**

Luka Merćep

Zagreb, lipanj 2021.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 10

**SEMANTIČKA SEGMENTACIJA ORGANA U SLIKAMA
MAGNETSKE REZONANCIJE**

Luka Merćep

Zagreb, lipanj 2021.

ZAVRŠNI ZADATAK br. 10

Pristupnik: **Luka Merćep (0036513765)**
Studij: Elektrotehnika i informacijska tehnologija i Računarstvo
Modul: Računarstvo
Mentor: prof. dr. sc. Siniša Šegvić

Zadatak: **Semantička segmentacija organa u slikama magnetske rezonancije**

Opis zadatka:

Semantička segmentacija važan je zadatak računalnog vida s mnogim zanimljivim primjenama. U posljednje vrijeme vrlo zanimljive rezultate postižu konvolucijski modeli s ljestvičastim naduzorkovanjem. Ovaj rad razmatra primjenu tog postupka za segmentaciju organa u medicinskim slikama koje su dobivene magnetskom rezonancijom. 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 klasifikaciju slike. Odabrati slobodno dostupni skup slika te oblikovati podskupove za učenje, validaciju i testiranje. Predložiti prikladnu arhitekturu dubokog modela za semantičku segmentaciju organa. 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: 11. lipnja 2021.

Zahvaljujem mentoru prof. dr. sc. Siniši Šegviću i mag. ing. Mateju Grciću na prenesenom znanju, pomoći pri pisanju ovog rada kao i savjetima pri implementaciji ovog rada.

SADRŽAJ

1. Uvod	1
2. Duboko učenje	2
2.1. Umjetni neuron	2
2.1.1. Aktivacijske funkcije	3
2.2. Neuronska mreža	5
2.3. Učenje neuronske mreže	6
2.3.1. Funkcije gubitka	7
2.3.2. Gradijentni spust	8
2.3.3. Optimizacijski algoritmi	9
2.4. Konvolucijska neuronska mreža	11
2.4.1. Konvolucijski sloj	12
2.4.2. Slojevi sažimanja	13
2.4.3. Normalizacijski sloj	14
2.4.4. Rezidualni modeli	14
2.4.5. Konvolucijski modeli s ljestvičastim naduzorkovanjem	15
3. Semantička segmentacija	18
4. Skupovi podataka	20
4.1. Cambridge-driving Labeled Video Database (CamVid)	20
4.2. MICCAI Grand Challenge: Prostate MR Image Segmentation 2012	21
5. Programska implementacija	23
5.1. Korištene tehnologije	23
5.2. Arhitektura mreže	23
5.2.1. Model ResNet-18 u semantičkoj segmentaciji	24
5.2.2. Model s ljestvičastim naduzorkovanjem	24

6. Rezultati	26
6.1. Korištene mjere	26
6.1.1. Točnost	26
6.1.2. Srednji omjer presjeka i unije	26
6.2. Rezultati na skupu podataka CamVid	27
6.3. Rezultati na skupu podataka PROMISE12	31
7. Zaključak	35
Literatura	36

1. Uvod

Povijest umjetnih neuronskih mreža počinje 1943. godine radom Warren McCulloch i Walter Pitts [1]. Nakon tog otkrića dolazi do popularizacije neuronskih mreža i područje počinje dobivati na popularnosti. Nakon tog razdoblja popularnosti dolazi do razdoblja u kojem neuronske mreže ne napreduju, a popularnost im pada. Kasijim otkrivanjem gradijentnog spusta kao i prosljeđivanja greške unatrag (engl. backprop) neuronske mreže počinju dobivati na značenju i njihova popularnost ponovno raste.

Zahvaljujući novim algoritmima, povećanju računalnih resursa i pristupu velikim količinama podataka neuronske mreže postaju sve dublje (imaju veći broj slojeva), odnosno dolazi do početka dubokog učenja (engl. deep learning). Konvolucijske neuronske mreže jedna su od poznatih arhitektura dubokih neuronskih mreža koje imaju odlične rezultate u području računalnog vida.

Računalni vid područje je računarstva koje se uglavnom bavi klasifikacijom ili segmentacijom slika. Klasifikacija slika je određivanje sadržaja slike, odnosno određivanje klase slike. Segmentacija slike je određivanje klase svakog od piksela te slike.

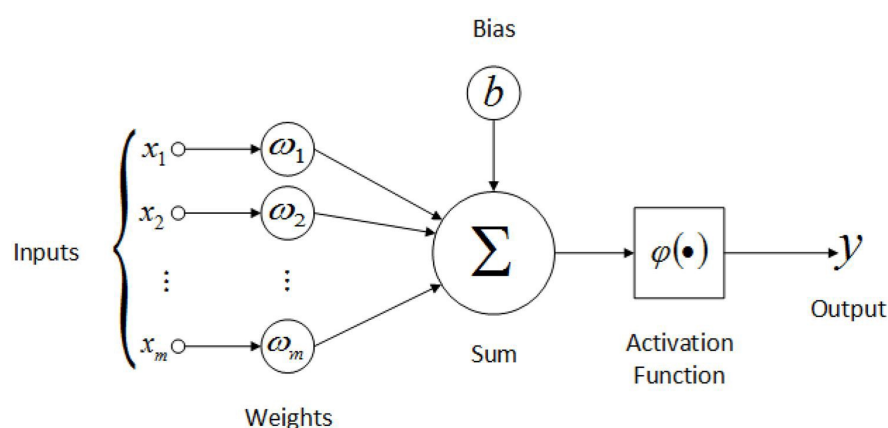
U nastavku ovog rada bit će opisane osnove dubokih neuronskih mreža kao što su neuron, određeni slojevi neuronske mreže, konvolucijske neuronske mreže. Opisat ću također rezidualne modele kao i modele koji koriste ljestvičasto naduzorkovanje.

Na kraju ću opisati programsku implementaciju rezidualnih modela i modela s ljestvičastim naduzorkovanjem na skupu podataka koji sadrže slike magnetske rezonancije prostate koji su korišteni na "MICCAI Grand Challenge: Prostate MR Image Segmentation 2012" [2] natjecanju. Za validaciju arhitekture mreža za segmentaciju koristio sam "Cambridge-driving Labeled Video Database (CamVid)" [3] [4].

2. Duboko učenje

2.1. Umjetni neuron

Umjetni neuron osnovna je građevna jedinica umjetne neuronske mreže. Umjetni neuroni složeni su u slojevima neuronske mreže gdje svaki sloj neuronske mreže kao ulaz u trenutni neuron uzima izlaze prethodnog sloja, obavlja nad njima nelinearnu transformaciju i tako nastaje izlaz iz trenutnog neurona, odnosno kada ovu ideju proširimo na cijeli sloj tada dobivamo izlaz trenutnog sloja neuronske mreže.



Slika 2.1: Građa umjetnog neurona. Preuzeto 04.06.2021. s:

<https://dzone.com/articles/the-artificial-neural-networks-handbook-part-4>

Na gornjoj slici su ulazi u trenutni neuron (koji mogu biti izlazi iz prethodnog sloja ili ulazni podatci u neuronsku mrežu ako se radi o prvom sloju mreže) označeni sa x_1 do x_m . Svaki od ulaznih parametara množi se sa pripadajućom težinom, odnosno x_i množi se sa ω_i , svi ti rezultati se sumiraju i dodaje im se pomak (engl. bias) b . Rezultat sume propušta se kroz aktivacijsku funkciju koja neuron čini nelinearnom funkcijom i to je konačno izlaz neurona. Dakle izlaz neurona možemo prikazati i kao:

$$y = f\left(b + \sum_{i=1}^m \omega_i * x_i\right) \quad (2.1)$$

gdje je f neka od aktivacijskih funkcija.

Danas ovaj pogled na duboke modele više nije relevantan. Moderan pogled na duboke modele jest da su to kompozicije diferencijalnih funkcija sa slobodnim parametrima. Skalarnе aktivacije u takvim modelima možemo zvati i neuronima.

2.1.1. Aktivacijske funkcije

Kao što je već opisano aktivacijske funkcije zadužene su za nelinearnost neurona, a samim time i za nelinearnost sloja neuronske mreže pa i same neuronske mreže. Izostavljanjem nelinearnost iz neuronske mreže duboki model bi se mogao sažeti u samo jedan neuron koji bi bio linearna funkcija. Neki od najčešćih oblika aktivacijske funkcije [5] [6]:

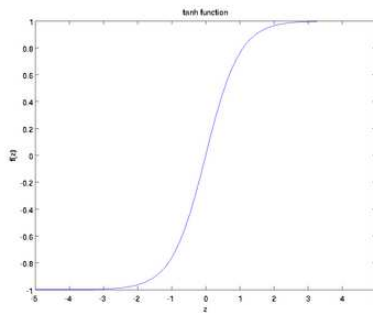
Funkcija identiteta je najjednostavnija aktivacijska funkcija čiji je izlaz jednak njezinu ulazu

$$f(x) = x \quad (2.2)$$

I takva funkcija je linearna te je svojstvena za model umjetnog neurona ADALINE (Adaptive Linear Element). Funkcija identiteta ne koristi se kao aktivacijska funkcija neurona jer njenom primjenom neuron ostaje linearan, a ovdje je navedena jer se koristi u rezidualnim modelima koji će biti objašnjeni kasnije.

Sigmoidalna funkcija derivabilna je aktivacijska funkcija. Kasnije će biti objašnjeno zašto je derivabilnost bitno svojstvo pri odabiru aktivacijske funkcije u neuronskim mrežama.

$$f(x) = \frac{1}{1 + e^{-x}} \quad (2.3)$$



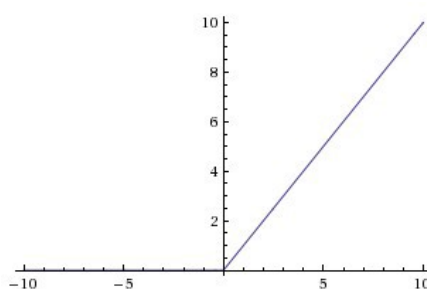
Slika 2.2: Sigmoidalna funkcija. Preuzeto 04.06.2021. s:

<https://medium.com/the-theory-of-everything/understanding-activation-functions-in-neural-networks-9491262884e0>

Prednost sigmoidalne funkcije u usporedbi s linearnom funkcijom je što je kodomena ograničena na $(0,1)$ pa ne dolazi do eksplozije pri nadovezivanju neurona koji koriste ovu aktivacijsku funkciju.

Zglobnica (engl. ReLU, Rectified Linear Unit) je najčešće korištena aktivacijska funkcija u dubokim modelima. Zglobnica je nelinearna aktivacijska funkcija koju računalo računa brže od sigmoide ili tangensa hiperboličnog [6]. Zbog svoje brzine izvođenja pogodan je izbor za aktivacijsku funkciju dubokih modela. Bitno je napomenuti kako zglobnica nije derivabilna, ali to u praksi nije problem jer nije derivabilna samo u jednoj svojoj točki ($x=0$).

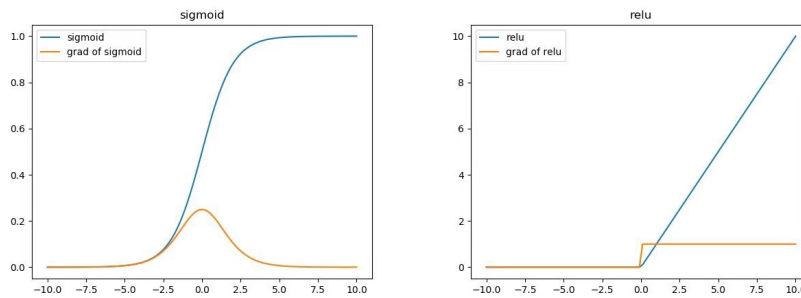
$$f(x) = \max(0, x) \quad (2.4)$$



Slika 2.3: Zglobnica (engl. ReLU). Preuzeto 04.06.2021. s:

<https://medium.com/the-theory-of-everything/understanding-activation-functions-in-neural-networks-9491262884e0>

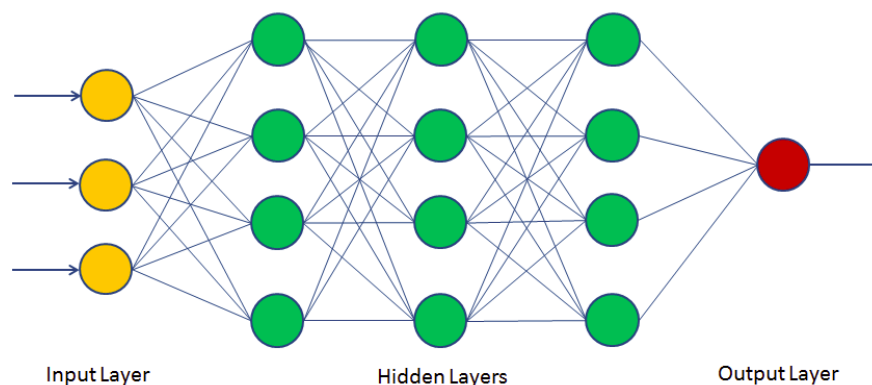
Glavna prednost zglobnice u usporedbi sa sigmoidalnom funkcijom je što ne guši gradijente što se vidi iz gornje slike gdje vidimo da je derivacija sigmoidalne funkcije vrlo blizu nule izvan dijela domene $(-5, 5)$.



Slika 2.4: Graf sigmoidalne funkcije i zglobnice i njihovih derivacija. Preuzeto 11.06.2021. s: <https://programmer.help/blogs/building-l-layer-neural-network-from-zero.html>

2.2. Neuronska mreža

Umjetne neuronske mreže nakupina su umjetnih neurona koji su posloženi u slojevima pri čemu svaki sloj kao svoj ulaz uzima izlaz prethodnog sloja, iznimka je početni sloj koji kao ulaz prima ulazne podatke cijele mreže. Kao što je ranije opisano neuroni su zapravo nelinearne funkcije čijim postavljanjem u lanac (izlaz jedne nelinearne funkcije je ulaz u drugu) se dobivaju vrlo kompleksne nelinearne funkcije koje onda mogu obrađivati nelinearne odnose uzoraka [5] koji su česti u stvarnom svijetu kao što su slike, tekst, zvuk i slično.



Slika 2.5: Neuronska mreža. Preuzeto 04.06.2021. s: <https://jacqui.sh/understanding-backpropagation-in-neural-networks/>

Kao što je vidljivo na gornjoj slici neuroni jednog sloja kao svoj ulaz primaju izlaze svih neurona prethodnog sloja što je odlika potpuno povezanog sloja (engl. fully connected layer) te kao svoj ulaz primaju samo izlaze sloja koji se nalazi neposredno prije trenutnog sloja. U kasnijem dijelu rada bit će opisani i modeli neuronskih mreža čije se ponašanje razlikuje od dva ranije opisana svojstva, primarno su to konvolucijske mreže gdje trenutni sloj kao svoj ulaz ne koristi izlaze iz svih neurona prethodnog sloja.

Rezidualna neuronska mreža (engl. residual neural network) i gusto povezana neuronska mreža (engl. densely connected convolutional networks) važni su pretstavnici modela koji kao svoj ulaz koriste i izlaze slojeva koji im nisu neposredni prethodnici, a u nastavku rada bit će detaljnije opisani rezidualni modeli.

Slojevi neuronskih mreža dijele se u tri skupine prema poziciji u neuronskoj mreži. Ulazni sloj (prvi sloj neuronske mreže) prima ulazne podatke mreže i broj neurona prvog sloja uglavnom je određen ulaznim podacima, primjerice ako neuronska mreža radi predikciju na temenju nekih N varijabli tada će najčešće prvi sloj imati N neurona. Izlazni sloj (zadnji sloj neuronske mreže) kao svoj izlaz daje izlaz cijele neuronske mreže i broj neurona izlaznog sloja uglavnom je određen željenim izlazom iz mreže, primjerice ako se radi o klasifikaciji podataka u M klasa neuronska mreža će uglavnom imati M neurona u svom zadnjem sloju. Skriveni slojevi (engl. hidden layers) mogu imati proizvoljan broj neurona te neuronska mreža može imati proizvoljan broj skrivenih slojeva. Također broj neurona u skrivenim slojevima može se međusobno razlikovati. Kratki zapis arhitekture neuronske mreže ranije prikazanog modela prikazuje se i kao $3 \times 4 \times 4 \times 4 \times 1$ gdje svaki od brojeva odgovara broju neurona određenog sloja.

2.3. Učenje neuronske mreže

Jednom kada je duboki model određen, odnosno kada je određen broj slojeva i broj neurona u svakom od slojeva te kada je svakom od neurona pridružena neka od aktivacijskih funkcija kreće proces učenja neuronske mreže.

Ovdje će se opisivati samo primjena neuronskih mreža na označenim podacima (engl. labeled data), odnosno podacima koji dolaze u parovima (x, y) gdje je x ulaz neuronske mreže nad kojima se radi predikcija, a y je oznaka danog ulaza gdje je y vrijednost ako se radi o regresijskom problemu ili klasa ulaza ako se radi o klasifikacijskom problemu.

Težini svakog od neurona kao i pomaku (engl. bias) svakog od neurona u početku je pridjeljena neka slučajna vrijednost. Kada ulazne podatke propagiramo kroz slojeve do izlaza neuronske mreže dobivamo predikciju. U nastavku će biti opisano kako se na temelju ulaznih podataka i predikcija neuronske mreže za te iste podatke izračunava funkcija gubitka te kako model uči (daje sve bolje i bolje predikcije) algoritmom gradijentnog spusta (engl. gradient descent).

2.3.1. Funkcije gubitka

Potrebno je uvesti funkcije gubitka kako bi se model u nekom trenutku mogao evaluirati. U nastavku će biti objašnjene dvije najčešće korištene funkcije gubitka korištene u dubokim modelima.

Srednja kvadratna pogreška (engl. MSE, Mean Square Error) računa se tako da se uprosječi kvadratna pogreška (razlika odstupanja predikcija i oznaka). Srednja kvadratna pogreška računa se:

$$\text{MSE} = \frac{1}{n} \cdot \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (2.5)$$

gdje je n broj parova (x_i, y_i) na kojima se model trenira, y_i je oznaka izlaza za ulazni primjer x_i , a \hat{y}_i predikcija modela za ulazni primjer x_i .

Svojtstno srednje kvadratne pogreške je da zbog kvadriranja pogreške puno više penalizira velika odstupanja od označene vrijednosti u usporedbi s manjim odstupanjima u vrijednosti. Srednja kvadratna vrijednost izvrsna je funkcija gubitka za slučajeve u kojima se greška može opisati kvadratnom razdiobom, a primjer toga su regresijski problemi.

Unakrsna entropija (engl. Cross-Entropy) koristi se kao funkcija gubitka za klasifikacijske probleme. Neka je izlaz klasifikacijskog modela koji klasificira ulaz u jednu od m klasa vektor veličine m gdje je j -ti član vektora vjerojatnost da se dani ulaz klasificira kao j -ta od m klasa. Također, neka su oznake primjera dane u vektoru veličine m takvom da ako trenutni primjer pripada j -toj, onda vektor za j -ti član ima 1, a svi ostali članovi vektora su 0. Unakrsna entropija koristi se i u slučaju mekih oznaka (oznake zadane distribucijom preko m poznatih klasa). Unakrsna entropija računa prema formuli:

$$J = -\frac{1}{n} \cdot \left(\sum_{i=1}^n \mathbf{y}_i \cdot \log(\hat{\mathbf{y}}_i) \right) \quad (2.6)$$

Minus u gornjoj jednadžbi prisutan je kako bi funkcija gubitka bila pozitivna te se ona onda minimizira, u slučaju izostavljanja minusa bi se funkcija gubitka maksimizirala što je kontraintuitivno, ali bi bilo matematički korektno.

Unakrsna entropija korigira se balansiranom unakrsnom entropijom ([7]) ako postoji nesrazmjer u zastupljenosti klasifikacijskih klasa u danom skupu podataka.

Unakrsna entropija za binarnu klasifikaciju (engl. Binary Cross Entropy) poseban je slučaj unakrsne entropije pri klasifikaciji podataka u binarnoj klasifikaciji (klasifikaciji podataka u dvije klase). Glavna razlika binarne klasifikacije u usporedbi sa klasifikacijom sa više klasa je u tome što se klasa može efikasno zapisati i koristiti ako je rezultat klasifikacije i oznaka jedna varijabla, dok je u slučaju s više klasa potrebno imati vektor s oznakom klase. Dakle y će poprimiti jednu od vrijednosti 0 ili 1 za tvrde oznake, odnosno vrijednosti $[0, 1]$ za meke oznake.

Unakrsnu entropiju za binarnu klasifikaciju računa se prema sljedećoj formuli:

$$J = -\frac{1}{n} \cdot \left(\sum_{i=1}^n y_i \cdot \log(\hat{y}_i) + (1 - y_i) \cdot \log(1 - \hat{y}_i) \right) \quad (2.7)$$

2.3.2. Gradijentni spust

Gradijentni spust (engl. Gradient descent) je optimizacijski algoritam čiji je zadatak pronalazak lokalnog minimuma diferencijabilne funkcije [8]. Gradijentni spust provodi se iterativno i u svakom koraku iteracije određuje smjer najvećeg nagiba (najveće derivacije) i pomiče se u tom smjeru prema minimumu funkcije.

Prije svakog koraka učenja (jednog koraka iteracije gradijentnog spusta) provodi se predikcija nad podacima, odnosno jedan korak propagacije unaprijed (engl. forward propagation) sa trenutnim parametrima mreže (težinama i pomacima). Jednom kada imamo predikcije \hat{y} nad skupom podataka nad kojim radimo trenutnu iteraciju izračunavamo gubitak nekom od funkcija gubitka. Sada svaki od parametara mreže (težine i pomaci) osvježavamo [9]:

$$w'_{i,j,k} = w_{i,j,k} - \alpha \cdot \frac{\partial J}{\partial w_{i,j,k}} \quad (2.8)$$

$$b'_{i,j} = b_{i,j} - \alpha \cdot \frac{\partial J}{\partial w_{i,j,k}} \quad (2.9)$$

gdje je sa w označena težina, b oznaka pomaka, i indeks sloja neuronske mreže, j indeks neurona u i -tom sloju, k indeks neurona prethodnog sloja čiji se izlaz množi s težinom $w_{i,j,k}$, α je korak učenja (engl. learning rate), J oznaka pogreške, a $w'_{i,j,k}$ i $b'_{i,j}$ su nove vrijednosti težine i pomaka.

Bitno je napomenuti kako se ranije opisana parcijalna derivacija funkcije gubitka ne računa za svaku težinu ispočetka, već se pomoću pravila ulančavanja parcijalna derivacija za i -ti sloj računa pomoću parcijalne derivacije sloja s indeksom $i + 1$ pomoću jednadžbe:

$$\frac{\partial y}{\partial x} = \frac{\partial y}{\partial u} \frac{\partial u}{\partial x} \quad (2.10)$$

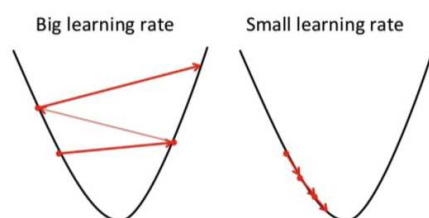
Nova ideja se koristi u algoritmu prosljeđivanja greške unatrag (oblik dinamičkog programiranja) gdje se nove vrijednosti parametara mreže u svakoj iteraciji računaju počevši od zadnjeg sloja neuronske mreže te se ti rezultati parcijalnih derivacija koriste za računanje sloja prije toga. Odnosno svaki sloj osim zadnjeg računa parcijalnu derivaciju funkcije gubitka u ovisnosti o težinama trenutnog sloja pomoću pravila ulačavanja iz parcijalne derivacije sloja koji mu je neposredni sljedbenik. Danas se algoritam prosljeđivanja greške unatrag provodi automatski, primjenom programskih okvira za automatsku diferencijaciju.

2.3.3. Optimizacijski algoritmi

Ranije opisani postupak gradijentnog spusta i prosljeđivanja greške unatrag opisan je tako da se u svakoj iteraciji učenje odvija na svim podacima skupa za treniranje. I osvježavanje parametara mreže odvija se tek kada se izračuna funkcija gubitka na svim podacima skupa za treniranje.

U praksi izračun svih predikcija i osvježavanje parametara mreže na cijelom skupu podataka za treniranje može trajati predugo da bi se napravio mali pomak prema traženom minimumu. Također, kako je potrebno pamtiti latentne aktivacije (aktivacije skrivenih slojeva) svih podataka može doći do iscrpljivanja memorije računala, primjerice u slučaju treniranja dubokog modela na velikom skupu podataka.

Jednostavno rješenje bilo bi povećanje stope učenja, no odabirom prevelike stope učenja dolazi do divergencije koja je prikazana na donjoj slici lijevo.



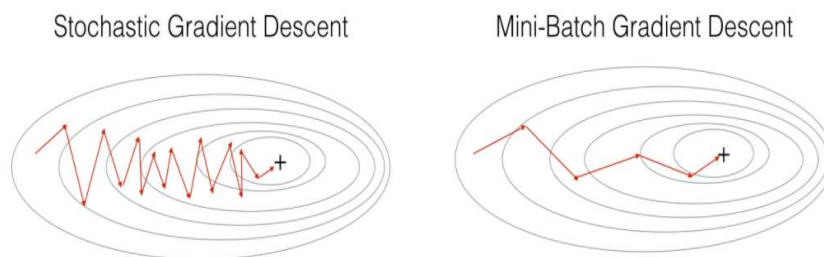
Slika 2.6: Primjer konvergencije s odgovarajućom stopom učenja (desno) i divergencije s prevelikom stopom učenja (lijevo). Preuzeto 05.06.2021. s:

<https://www.educative.io/edpresso/learning-rate-in-machine-learning>

Neki od optimizacijskih algoritama koji rješavaju ovaj problem opisani su u nastavku, a generalna ideja je podijeliti skup za treniranje na manje grupe i osvježavati

parametre mreže nakon iteracije nad svakom od grupa algoritmom gradijentnog spusta (engl. Mini-Batch Gradient Descent).

Stohastički gradijentni spust (engl. SGD, Stochastic Gradient Descent) poseban je slučaj gradijentnog spusta s manjim grupama u kojem grupa ima samo 1 element. Odnosno, nakon svakog primjera iz skupa za treniranje radi se osvježavanje parametara mreže.



Slika 2.7: Ilustracija učenja modela stohastičkim gradijentnim spustom (lijevo) i gradijentnim spustom s malim grupama (desno) gdje su grupe veće od jednog člana. Preuzeto 05.06.2021.

s:

<https://www.i2tutorials.com/explain-brief-about-mini-batch-gradient-descent/>

Odabirom manjih grupa (ili ekstremom ovog primjera stohastičkim gradijentnim spustom) dobiva se više šuma u treniranju, odnosno time što se odlučujemo za smanjenje veličine grupe, koraci učenja ne idu direktno prema točki lokalnog minimuma što je dobro prikazano i gornjom slikom. Glavna prednost ovog pristupa je češće ažuriranje parametara modela.

Adam (ime je derivirano od engl. "adaptive moment estimation" [10]) je optimizacijski algoritam koji radi na načelu zadržavanja zaleta, odnosno u trenutnoj iteraciji pri donošenju odluke o smjeru osvježavanja nekog parametra mreže koriste se i ranije izračunati gradijenti tog parametra. Osvježavanje prvog i drugog zaleta gradijenta korištenjem Adama kao optimizacijskog algoritma [11]:

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t \quad (2.11)$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2 \quad (2.12)$$

gdje je g_t oznaka gradijenta parametra neuronske mreže izračunat u trenutnoj iteraciji, m i v su varijable zaleta definirane Adamom, a β_1 i β_2 su hiperparametri definirani u

Adamu [10], a varijable zaleta inicijaliziraju se na 0.

Potrebno je napraviti korekciju tih parametara i osvježiti parametre mreže [11]:

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t} \quad (2.13)$$

$$\hat{v}_t = \frac{v_t}{1 - \beta_2^t} \quad (2.14)$$

$$w'_{i,j,k} = w_{i,j,k} - \frac{\alpha}{\sqrt{\hat{v}_t + \epsilon}} \hat{m}_t \quad (2.15)$$

Napomena: zaleti m i v se računaju posebno za svaki parametar mreže.

[10] predlažu sljedeće vrijednosti hiperparametara Adama:

$$\beta_1 = 0.9 \quad \beta_2 = 0.999 \quad \epsilon = 10^{-8}$$

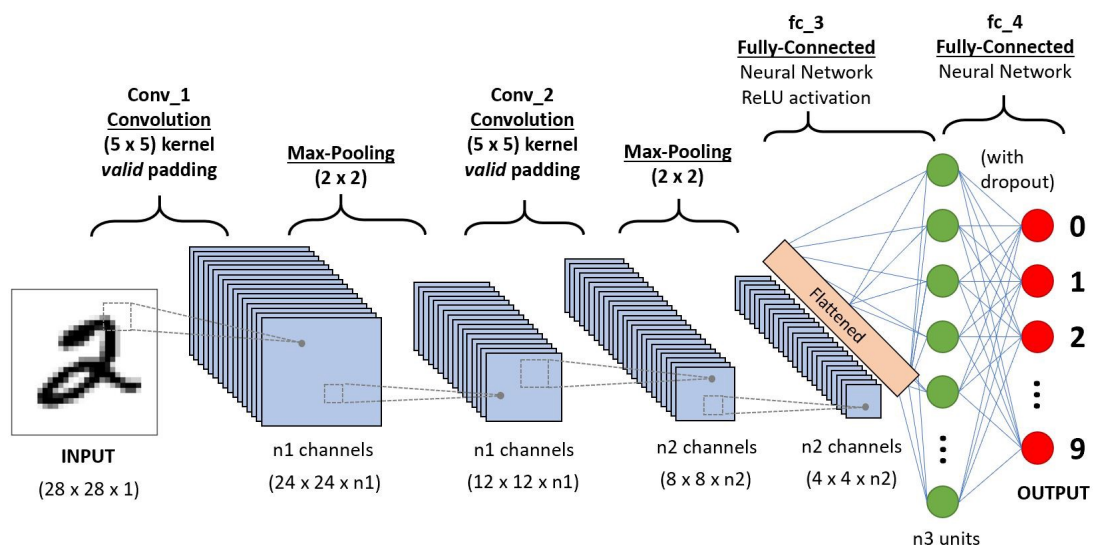
2.4. Konvolucijska neuronska mreža

Veliki napredak računalnog vida postignut je korištenjem dubokih neuronskih mreža, a model koji predvodi taj napredak su upravo konvolucijske neuronske mreže [12]. Ranije u ovom radu opisan je koncept potpuno povezanih neuronskih mreža i rečeno je da će u nastavku biti prezentiran model konvolucijskih neuronskih mreža koje u svim slojevima nemaju potpuno povezane slojeve.

Kako su u analizi slika bitniji prijelazi između jednolikih površina (rubovi slike) te postojanje određenih značajki (primjerice dijelovi lica pri klasifikaciji ljudi) nego njihov točan položaj unutar mreže razvijen je model konvolucijske neuronske mreže koji zadovoljava ove zahtjeve. Također, kako je ranije opisano konvolucijski slojevi nisu potpuno povezani s prethodnim slojem pa imaju bitno manje parametara od potpuno povezanog sloja.

Gornja slika primjer je konvolucijske neuronske mreže za klasifikaciju rukom pisanih crno-bijelih znamenki veličine 28x28 piksela. Gornja mreža sastoji se od konvolucijskog sloja, sloja sa sažimanje maksimalnom vrijednosti, funkcije izravnavanja (engl. flattened) koja je zadužena za pretvaranje svog ulaza u vektor za svaki primjer skupa podataka te na kraju dolaze dva potpuno povezana sloja.

Slojevi mreže prije funkcije izravnavanja služe za stvaranje mapi značajki koje onda dva potpuno povezana sloja koriste za klasifikaciju znamenaka. Kao što je i ranije opisano izlaz mreže za klasifikaciju jednak je broju klasa klasifikacije, a u gornjem primjeru je to 10 (znamenke od 0 do 9).



Slika 2.8: Primjer konvolucijske neuronske mreže za klasifikaciju znamenaka. Preuzeto 05.06.2021. s:

<https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53>

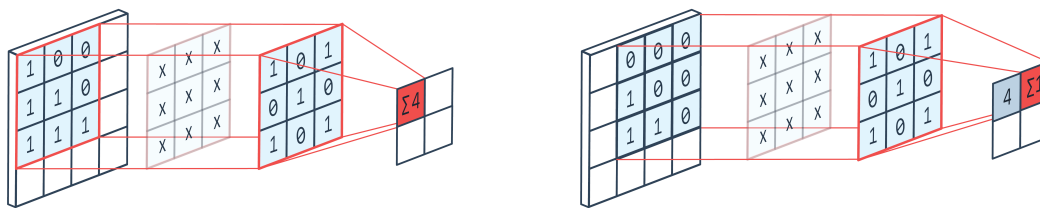
U nastavku će biti objašnjeni osnovni slojevi konvolucijskih mreža.

2.4.1. Konvolucijski sloj

Prije samog opisa konvolucijskog sloja bitno je ukratko objasniti koncept kanala (engl. channel) u digitalnom zapisu slika, a isti koncept koristi se i u konvolucijskom sloju. Slike ćemo zapisivati u matricama dimenzija $C \times H \times W$ gdje je C broj kanala slike, H visina slike, a W širina slike. Broj kanala slike na ulazu u neuronsku mrežu uglavnom iznosi 1 za crno-bijele slike ili 3 za slike u boji (RGB format).

Ulazni broj kanala u svaki od konvolucijskih slojeva definiran je prethodnim slojem ili ulazom ako se radi o prvom konvolucijskom sloju, a izlaz svakog od konvolucijskih slojeva konfigurabilan je parametar (slično kao broj neurona u potpuno povezanom skrivenom sloju neuronske mreže). Svakom od izlaznih kanala odgovara jedna jezgra (engl. kernel).

Gore je prikazano ponašanje jezgre konvolucijskog sloja nad danim ulazom tog sloja, dakle jezgra se fiksira iznad nekog okna ulaznog tenzora i odradi se operacija konvolucije, odnosno pomnoži se svaki element jezgre sa pripadajućim elementom ulaza i rezultat se zbroji te tako dobivamo izlaznu aktivaciju, nakon ovoga se okno klizno pomiče te se postupak određivanja izlazne aktivacije ponavlja.



Slika 2.9: Primjer konvolucijskog sloja. Preuzeto 05.06.2021. s:

<https://datascience.stackexchange.com/questions/80436/understanding-how-convolutional-layers-work>

Konfigurabilni elementi konvolucijskog sloja su broj izlaznih kanala (broj jezgri), veličina koraka (engl. stride) koji određuje udaljenost dviju fiksnih pozicija jezgre (u gornjem primjeru je korak jednak 1) te nadopunjavanje (engl. padding) koji dodaje elemente oko ulaza kako rubni elementi ne bi imali manju važnost u konvoluciji.

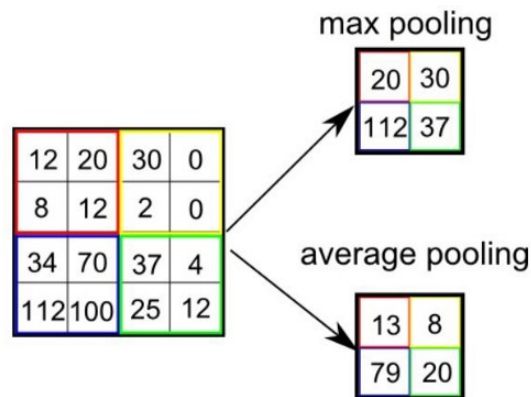
2.4.2. Slojevi sažimanja

Drugi bitan sloj konvolucijskih neuronskih mreža su slojevi sažimanja, njihova uloga je smanjenje prostorne veličine čime se dobiva na brzini jer se smanjuje broj parametara mreže te se ekstrahiraju dominantne značajke [12]. U nastavku će biti objašnjena dva tipa slojeva sažimanja.

Jezgra sažimanja ponaša se vrlo slično jezgri konvolucijskog sloja, jezgra se fiksira na neku poziciju u matrici te na toj poziciji izvodi svoju funkciju, a glavna je razlika što sloj sažimanja nema parametre mreže koji se mijenjaju (uče) za vrijeme treniranja mreže.

Sažimanje srednjom vrijednosti (engl. Average pooling) jednom kada je jezgra fiksirana računa prosječnu vrijednost značajki u odgovarajućem oknu ulaza.

Sažimanje maksimalnom vrijednosti (engl. Max pooling) jednom kada je jezgra fiksirana računa maksimalnu vrijednost značajki u odgovarajućem oknu ulaza.



Slika 2.10: Primjer sloja sažimanja srednjom vrijednosti i sloja sažimanja maksimalnom vrijednosti. Preuzeto 05.06.2021. s: <https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53>

2.4.3. Normalizacijski sloj

Sloj normalizacije grupa (engl. batch normalization, batch norm) radi normalizaciju podataka koje dobije na svom ulazu. Ulaz normalizacijskog sloja izlaz je prethodnog sloja mreže. Normalizacija se provodi tako da se od ulaznih podataka oduzme srednja vrijednost i rezultat se podijeli sa standardnom devijacijom ulaza. Bitno je napomenuti kako sloj normalizacije grupa ne radi sa stohastičkim gradijentnim spustom, odnosno veličina grupe treba biti veća od 1.

Sloj normalizacije grupa povećava brzinu ućanja modela te služi kao regularizacijski sloj.

2.4.4. Rezidualni modeli

Duboke neuronske mreže pa i duboke konvolucijske neuronske mreže teže se treniraju te nakon određenog broja slojeva počinju davati lošije rezultate od plićih mreža slične arhitekture.

Iz grafova rada [13] vidi se kako se dodatnim dodavanjem slojeva neuronskoj mreži kvaliteta predikcija krene smanjivati. Isti rad predlaže uvođenje rezidualnih blokova čijim se uvođenjem broj slojeva neuronske mreže može značajno povećati.

Generalna ideja rezidualnih modela je uvođenje veza (engl. skip connections) koje zbrajaju izlaz jednog bloka sa ulazom nekog bloka koji mu nije neposredni sljedbenik u arhitekturi mreže.

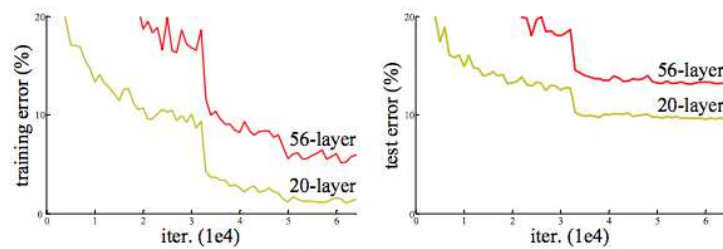
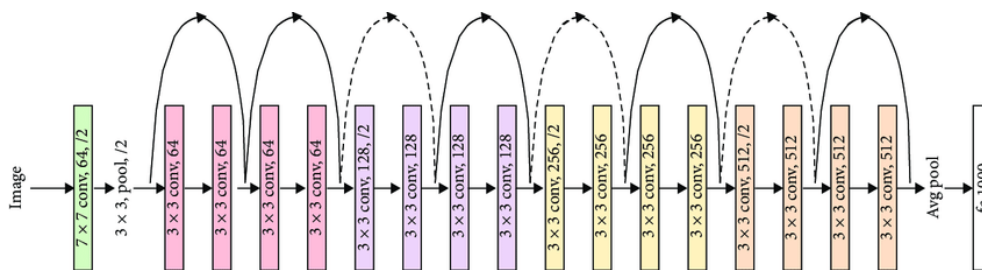


Figure 1. Training error (left) and test error (right) on CIFAR-10 with 20-layer and 56-layer “plain” networks. The deeper network has higher training error, and thus test error. Similar phenomena on ImageNet is presented in Fig. 4.

Slika 2.11: Grafovi greške skupa za treniranje i skupa za testiranje mreže s 20 i mreže s 56 slojeva. Preuzeto iz [13]



Slika 2.12: Arhitektura mreže ResNet-18 (rezidualnog modela s 18 slojeva). Preuzeto 06.06.2021. s:

https://www.researchgate.net/figure/ResNet-18-model-architecture-10_fig2_342828449

Na gornjoj slici prikazana je arhitektura rezidualne neuronske mreže s 18 slojeva. Također, vide se dva tipa rezidualnih veza (engl. skip connections), rezidualne veze označene punom linijom obavljaju funkciju identiteta nad ulaznim podacima, odnosno podaci na izlazu funkcije jednaki su onima na ulazu. Drugi tip rezidualne veze označen je isprekidanom linijom na gornjoj slici, a ona označava propuštanje podataka kroz jedan konvolucijski sloj s jezgrama veličine 1x1 i korakom veličine 2, a broj izlaznih kanala istovjetan je broju kanala s kojim će se izlaz ove rezidualne veze zbrojiti. Konačno rezultat ove konvolucije propušta se kroz normalizacijski sloj.

2.4.5. Konvolucijski modeli s ljestvičastim naduzorkovanjem

Nedostatak ranije opisanih modela analize slike (konvolucijskih modela i rezidualnih modela) je što ulaznu sliku pretvaraju u znatno manju mapu značajki. Ako je zadatak modela raditi semantičku segmentaciju slike u tom slučaju izlaz modela treba za svaki

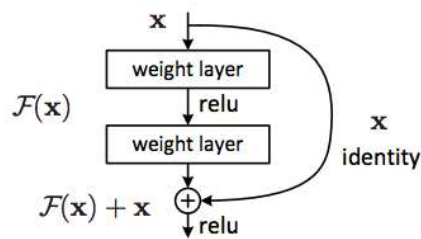
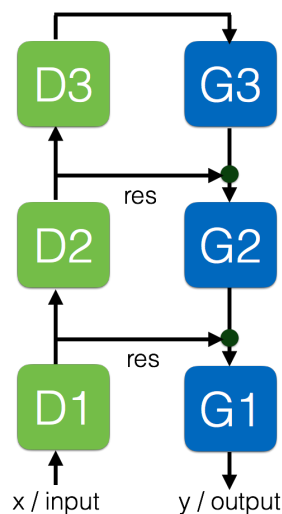


Figure 2. Residual learning: a building block.

Slika 2.13: Rezidualni blok sa funkcijom identiteta kao rezidualnom vezom. Preuzeto iz [13]

ulazni piksel odrediti klasu tog piksela. Konvolucijski modeli i rezidualni modeli u semantičkoj segmentaciji trebaju iz znatno manje mape značajki dobiti izlaz jednake veličine kao i ulazna slika te zbog toga gube na točnosti.

Konvolucijski modeli s ljestvičastim naduzorkovanjem rješavaju ovaj problem uvođenjem lateralnih (bočnih) veza koje sadrže podatka iz nekoliko međufaza pretvorbe originalne slike u znatno manju mapu značajki.



Slika 2.14: Konvolucijski modeli s ljestvičastim naduzorkovanjem. Preuzeto 11. 6. 2021. s: <https://towardsdatascience.com/a-new-kind-of-deep-neural-networks-749bcde19108>

Rezidualne veze na slici oznašene s "res" u modelima s ljestvičastim naduzorkovanjem zovu se još i lateralne veze. Na slici prikazan model sastoji se primarno od 2 različite komponente čijim se ponavljanjem gradi model.

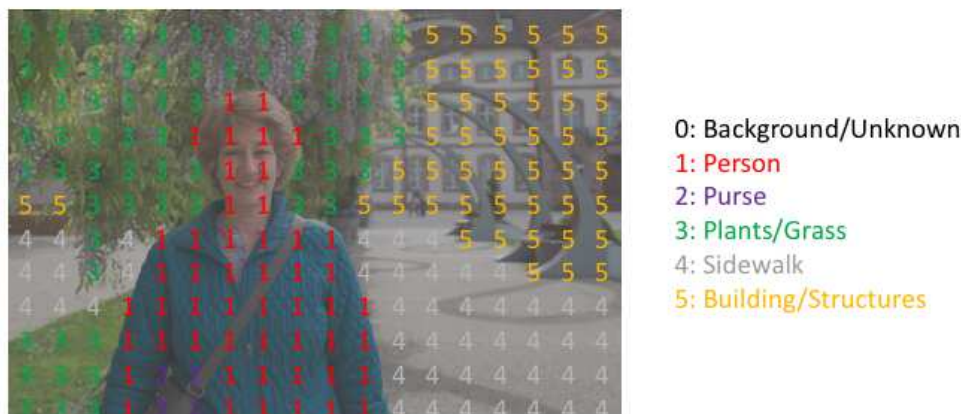
Enkoderi (na slici prikazani zelenom bojom) su jedna od tih komponenti, njihova uloga je iz ulazne primjernom enkodera stvoriti znatno manju mapu značajki. Kasnije će biti opisana arhitektura modela koji koristi ResNet-18 kao bazu za izgradnju svojih enkodera.

Dekoderi (na slici prikazani plavom bojom) su druga komponenta čija je uloga spajanje mapa značajki niže rezolucije sa mapama značajki veće rezolucije koji su izlaz pripadnog enkodera u ljestvičastoj arhitekturi modela [14]. Dekoderi rade povećanje rezolucije mape značajki niže rezolucije primjerice bilinearnim naduzorkovanjem te se izlaz bilinearnog naduzorkovanja zbraja sa lateralnom vezom pripadnog enkodera. Rezultat sumacije se zatim može propustiti kroz konvolucijski sloj koji također može biti dio dekodera.

Ovakva ljestvičasta arhitektura izvrstan je odgovor na nedostatke ranije opisanih konvolucijskih i rezidualnih modela.

3. Semantička segmentacija

Semantička segmentacija slike jedan je od poznatih zadataka računalnog vida kojem je zadatak pridijeliti značenje svakom pikselu slike (svrstati ga u jedno od predefiniranih klasa). Semantička segmentacije jedan je od oblika guste predikcije (engl. dense prediction).



Slika 3.1: Primjer semantičke segmentacije. Preuzeto 06.06.2021. s: <https://www.jeremyjordan.me/semantic-segmentation/>

Gornja slika primjer je semantičke segmentacije slike u jednu od 5 klasa (pozadina/ostalo, osoba, torbica, pločnik i zgrada), u semantičkoj segmentaciji slike čest je primjer postojanja jedne klase u koju se svrstavaju svi pikseli koji ne pripadaju jednoj od češćih klasa, a pogotovo je to primjer u segmentaciji slike autonomnih vozila.

Bitno je napomenuti kako ovdje opisana semantička segmentacija ne razlikuje objekte iste klase, odnosno ako dva različita objekta iste klase imaju susjedne piksele semantička segmentacija ih neće nikako razlikovati, time se bavi segmentiranje primjeraka (engl. instance segmentation) [15].

Česta područja primjene semantičke segmentacije slike [16] su autonomna vožnja (prepoznavanje pješaka, prometnih znakova, svjetla kočenja automobila i drugo), me-

dicina (mjerjenje volumena tkiva, lokacija tumora i drugih patogena, planiranje operacije i drugo) i satelitske snimke (ceste, šume, usjevi i drugo). U ovom radu semantička segmentacija slike korištena je u području autonomne vožnje na skupu podataka CamVid te u području medicine na skupu podataka PROMISE12, a oba skupa podataka bit će opisana u nastavku.

4. Skupovi podataka

4.1. Cambridge-driving Labeled Video Database (CamVid)

CamVid (Cambridge-driving Labeled Video Database) [3] [4] korišten je kao skup podataka za validaciju modela neuronske mreže, odnosno svaka nova arhitektura modela prije njenog treniranja nad slikama magnetske rezonancije prostate validirana je na CamVid skupu podataka. CamVid je skup podataka koji sadrži slike slikane iz perspektive samovozećeg vozila i svaka slika ima svoj pripadni par oznaka za semantičku segmentaciju, odnosno svakom pikselu slike dodijeljena je jedna od 32 klase. Slike CamVid skupa podataka su u boji (imaju 3 kanala), visina slike je 720 piksela i široke su 960 piksela. CamVid skup podataka dolazi u već podijeljenim podskupovima podataka za treniranje, validaciju i testiranje mreže.

Pri treniranju mreže na ovom skupu podataka smo prepolovili slike, odnosno korištene su slike rezolucije 3x360x480. Također, iako originalni skup podataka dolazi sa oznakama za 32 klase, pri treniranju modela na ovom skupu podataka korišten je reducirani broj klasa. Korišteno je 11 originalnih klasa: biciklist (engl. bicyclist), zgrada (engl. building), automobil (engl. car), stup (engl. column pole), ograda (engl. fence), pješak (engl. pedestrian), cesta (engl. road), trotoar (engl. sidewalk), prometni znak (engl. sign symbol), nebo (engl. sky) i stablo (engl. tree). Korištena je i jedna dodatna klasa koja je bila unija svih izbačenih klasa. Neke od izbačenih klasa: životinja (engl. animal), child (engl. dijete), tunel (engl. tunnel), most (engl. most) i druge.

Svaka od klasa u CamVid-u dolazi sa odabranom bojom za lakši prikaz semantički segmentiranih slika.



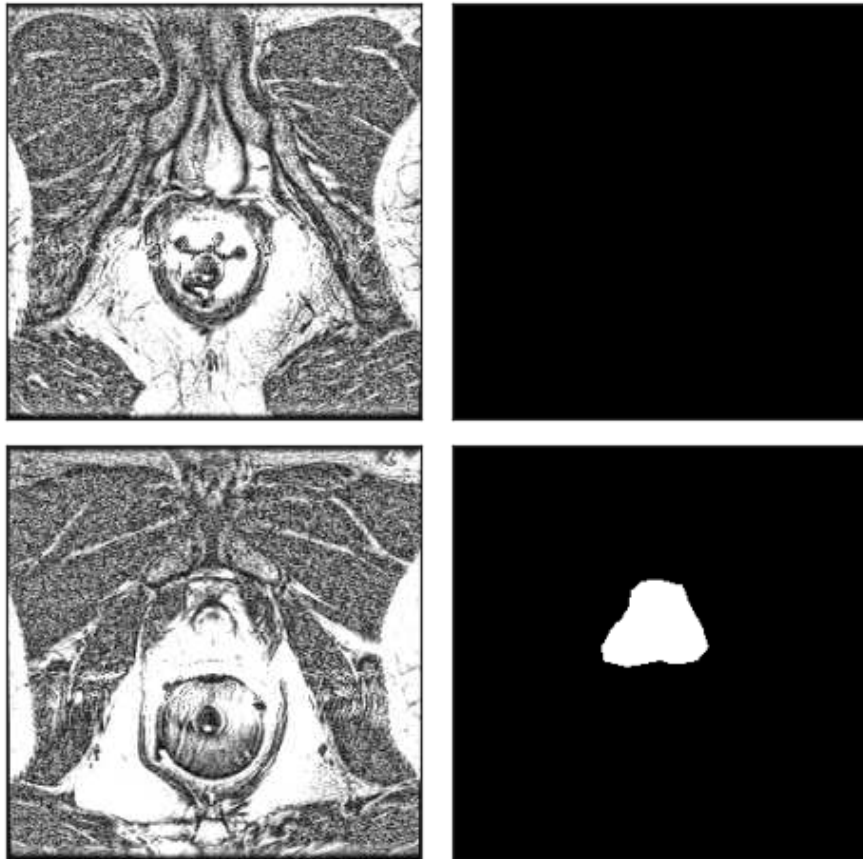
Slika 4.1: Primjeri slika iz CamVid skupa podataka (lijevo) i primjeri oznaka klasa svakog od piksela (desno).

4.2. MICCAI Grand Challenge: Prostate MR Image Segmentation 2012

Skup podataka PROMISE12 (MICCAI Grand Challenge: Prostate MR Image Segmentation 2012) [2] originalno je korišten i za istoimeno natjecanje i sadrži slike crno-bijele slike magnetske rezonancije područja prostate, također svaka slika dolazi sa pripadnim parom oznaka za semantičku segmentaciju, odnosno za svaki piksel svake od slika dolazi i oznaka 1 za tkivo prostate, odnosno 0 za tkivo koje ne pripada prostati.

Rezolucija slika je $1 \times 256 \times 256$ i originalne slike su korištene pri treniranju mreže, a svaka arhitektura neuronske mreže prethodno je validirana na CamVid skupu podataka.

Na slikama oznaka semantičke segmentacije prostate je prikazana bijelom bojom, a okolno tkivo crnom. Također bitno je napomenuti kako 44% slika u skupu podataka nema oznaku prostate, odnosno da na pripadnim slikama magnetske rezonancije nema tkiva prostate.



Slika 4.2: Primjeri slika iz PROMISE12 skupa podataka (lijevo) i primjeri oznaka klasa svakog od piksela (desno).

5. Programska implementacija

5.1. Korištene tehnologije

Programski jezik Python 3 korišten je pri izradi ovog rada za sljedeće dijelove: učitavanje i manipulacija podataka, definiranje arhitekture modela, treniranje i evaluacija modela. Kao podršku za rukovanje velikom količinom vektorskih i matričnih podataka koristio sam knjižnicu NumPy¹ koja je predviđena za rukovanje i manipulaciju vektora i matrica u programskom jeziku Python. NumPy sam koristio pri učitavanju podataka te kao pomoć pri pisanju funkcija za evaluaciju modela. Također sam koristio knjižnicu PyTorch² koji je također specifično razvijen za programski jezik Python. PyTorch sam koristio za definiranje modela neuronske mreže kojeg sam složio od gotovih slojeva koje nudi PyTorch (konvolucijski sloj, sloj maksimalnog sažimanja, normalizacijski sloj, potpuno povezani sloj i drugi), također jednom kada je model definiran i korisnik napiše funkciju za propagaciju unaprijed ova knjižnica može sama provoditi propagiranje greške unatrag (autodiff). Također, PyTorch uključuje optimizacijske algoritme kao, funkcije računanja gubitka trenutnog modela te nudi transparentno računanje na GPU-ovima (graphics processing unit). Učenje i razvijanje modela provedeno je u sustavu Google Colab³ koji može interpretirati Python skripte te nudi CPU i GPU resurse na besplatno korištenje.

5.2. Arhitektura mreže

U ovom poglavlju bit će opisane dvije arhitekture koje su korištene pri izradi ovog rada. Jednostavnija arhitektura zasnovana je na modelu ResNet-18, a naprednija arhitektura zasnovana je na ideji konvolucijskih modela s ljestvičastim naduzorkovanjem.

¹<https://numpy.org/>

²<https://pytorch.org/>

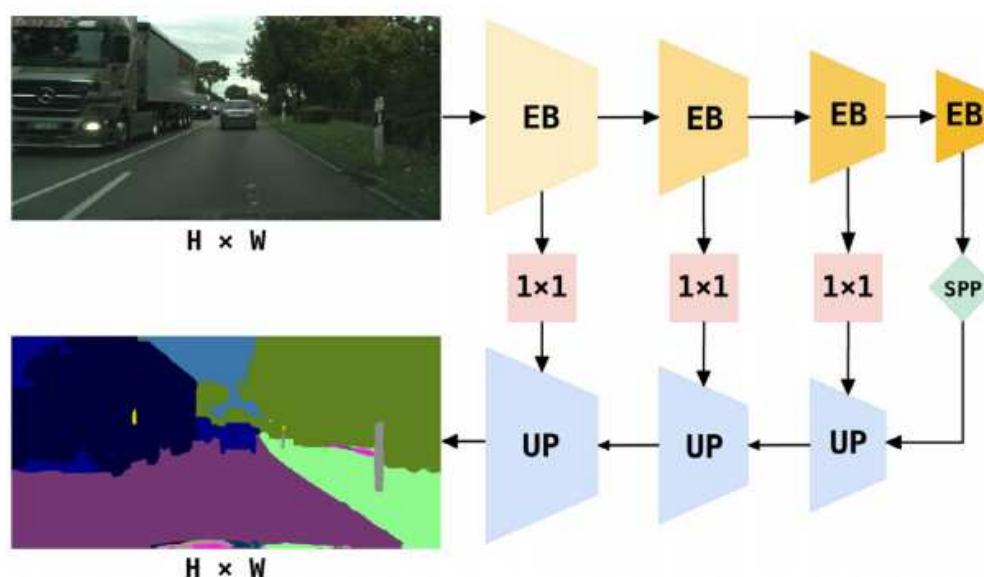
³<https://colab.research.google.com/>

5.2.1. Model ResNet-18 u semantičkoj segmentaciji

Kao polazni (engl. benchmark) model korišten je ResNet-18 implementiran u sklopu knjižnice PyTorch kojem smo maknuli zadnja dva sloja: sloj sažimanja srednjom vrijednosti i potpuno povezani sloj. Umjesto zadnja dva sloja koji se koriste pri klasifikaciji slike stavljeni su slojevi korišteni za semantičku segmentaciju: konvolucijski sloj i sloj za bilinearano naduzorkovanje. Dodani konvolucijski sloj ima jezgru veličine 1×1 , korak 1 te mu je broj izlaznih kanala jednak broju klasa u zadanom problemu semantičke segmentacije. Sloj za bilinearano naduzorkovanje koristi se za povećanje rezolucije ulaza na rezoluciju slika dobivenih na ulazu modela.

5.2.2. Model s ljestvičastim naduzorkovanjem

Druga arhitektura modela korištena u ovom radu originalno je opisana u radu [14].

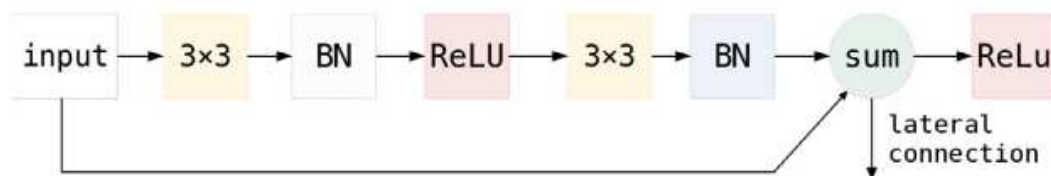


Slika 5.1: Arhitektura modela s ljestvičastim naduzorkovanjem. Preuzeto iz [14]

Arhitektura SwiftNet sastoji se od modula za izlučivanje značajki (odnosno koder) sastoji se od 4 procesna bloka (EB, sl. 5.1). Procesni blokovi provode konvolucije na rezolucijama $/4$, $/8$, $/16$ i $/32$. Modul za naduzorkovanje (odnosno dekođer) sastoji se od 3 procesna bloka (UP, sl. 5.1) koji podižu rezoluciju na $/16$, $/8$ odnosno $/4$. Izlazi iz koderu propušeni su kroz jedan sloj konvolucije te je izlaz te konvolucije korišten kao lateralni ulaz u dekođer. Na slici arhitekture modela prikazan je i sloj prostorno piramidalnog sažimanja (engl. SPP, spatial pyramid pooling) prikazan zelenom bojom. Sloj prostornog piramidalnog sažimanja izostavljen je iz modela za skup

podataka CamVid kao i iz modela za skup podataka PROMISE12. Sloj za prostorno piramidalno sažimanje posebno je koristan kada su slike velike (preko megapiksela) i kada se objekti u sceni javljaju na različitim mjestima. Kako su slike iz skupa podataka PROMISE12 niske rezolucije (256x256) sloj za prostorno piramidalno sažimanje je izostavljen iz arhitekture tog modela.

Koder SwiftNet-a odgovara prednjem dijelu arhitekture modela ResNet-18. Blokovi kodera u ovoj arhitekturi su prilagođeni ResNet-18 modeli iz kojih smo maknuli zadnja dva sloja (sloj sažimanja srednjom vrijednost i potpuno povezani sloj). Izlaz lateralne veze svakog od blokova kodera je rezultat sumiranja u zadnjem rezidualnom bloku ResNet-18 modela, odnosno ulaz u funkciju zglobnice (ReLU).



Slika 5.2: Arhitektura zadnjeg rezidualnog bloka enkodera. Preuzeto iz [14]

Rezolucija izlazne mape značajki prvog enkodera je 4 puta manja od ulazne slike, a rezolucija izlaznih mapa značajki svakog od preostalih modela 2 puta je manja od ulaznih mapa značajki.

Blokovi za naduzorkovanje kao svoj ulaz uzimaju izlaz prethodnog dekodera te lateralnu vezu sa pripadnim enkoderom. Iznimka je dekodek koji kao ulaz koristi izlaz iz zadnjeg enkodera ili sloja za prostorno piramidalno sažimanje. Izlaz prethodnog enkodera propušta se kroz sloj za bilinearano naduzorkovanje koji povećava rezoluciju svog ulaza na rezoluciju pripadne lateralne veze. Izlaz sloja za bilinearano naduzorkovanje zbraja se s izlaznom konvolucije lateralne veze i rezultat se propušta kroz dodatan konvolucijski sloj s jezgrom veličine 3x3. Izlaz zadnjeg dekodera propušta se kroz još jedan konvolucijski sloj i sloj za bilinearano naduzorkovanje (slično kao i u prethodno opisanom modelu ResNet-18 za semantičku segmentaciju).

6. Rezultati

6.1. Korištene mjere

U ovom dijelu ukratko će biti objašnjene dvije mjere korištene pri evaluaciji rezultata rada.

6.1.1. Točnost

Točnost (engl. accuracy) jedna je od najjednostavnijih načina procjene uspješnosti modela. Točnost se računa kao postotak dobro klasificiranih piksela i u kontekstu semantičke segmentacije računa se:

$$\frac{\text{broj dobro klasificiranih piksela}}{\text{ukupan broj piksela}} \quad (6.1)$$

Glavni nedostatak točnosti je što daje jako lošu ocjenu uspješnosti modela ako postoje velike razlike u zastupljenosti klasa. Primjer ovoga je izrada modela koji na temelju određenog ulaza klasificira pacijente u skupine koja ima i skupinu koja nema rak. Kako je pojava raka u medicini statistički rijetka model koji kao svoj izlaz uvijek daje predikciju da pacijent nema rak bi imao vrlo veliku točnost, ali korisnost ovog modela bila bi zanemariva.

6.1.2. Srednji omjer presjeka i unije

Srednji omjer presjeka i unije (engl. mIoU, mean Intersection over Union) jedna je od mjera koje rješavaju ranije opisani problem. Za opis srednjeg omjera presjeka i unije potrebno je prvo objasniti omjer presjeka i unije na jednoj klasi. Jednom kada je IoU (omjer presjeka i unije) izračunat za sve klase mIoU (srednji omjer presjeka i unije) računa se kao aritmetička sredina svih IoU-ova svih klasa korištenim u danom problemu semantičke segmentacije.

Uzevši u obzir rezultata klasifikacije modela i pravih oznaka klasa piksele možemo podijeliti u 4 skupine [17]:

- Točno pozitivno klasificirani pikseli (engl. TP, true positive) su oni pikseli koji pripadaju promatranoj klasi i model ih je klasificirao kao primjerak promatrane klase
- Netočno pozitivno klasificirani pikseli (engl. FP, false positive) su oni pikseli koje je model klasificirao da pripadaju promatranoj klasi, ali oni pripadaju nekoj drugoj klasi
- Točno negativno klasificirani pikseli (engl. TN, true negative) su oni pikseli koje je model klasificirao da pripadaju nekoj drugoj klasi i oni pripadaju nekoj drugoj klasi
- Netočno negativno klasificirani pikseli (engl. FN, false negative) su oni pikseli koje je model klasificirao kao primjerke neke druge klase, ali oni pripadaju promatranoj klasi

Omjer presjeka i unije računa se prema jednadžbi:

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

Jednom kada je omjer presjeka i unije izračunat za sve promatrane klase srednji omjer presjeka i unije (mIoU) računa se kao aritmetička sredina svih izračunatih omjera presjeka i unije te se izražava u postotcima.

6.2. Rezultati na skupu podataka CamVid

Na skupu podataka CamVid provedeno je validiranje arhitekture dvaju korištenih modela: modela ResNet-18 bez ljestvičastog naduzorkovanja i modela ResNet-18 s ljestvičastim naduzorkovanjem za semantičku segmentaciju koji su ranije opisani u ovom radu.

Model	Skup podataka	Točnost [%]	Srednji omjer presjeka i unije (mIoU) [%]
Model ResNet-18 bez ljestvičastog naduzorkovanja	skup za učenje	87.43	58.08
	skup za validaciju	84.17	50.25
	skup za testiranje	81.52	46.79
Model ResNet-18 s ljestvičastog naduzorkovanja	skup za učenje	90.87	68.85
	skup za validaciju	87.27	57.86
	skup za testiranje	83.01	51.00

Tablica 6.1: Rezultati modela na skupu podataka CamVid.

Prvo je bilo pokrenuto učenje modela ResNet-18 bez ljestvičastog naduzorkovanja, a rezultati i hiperparametri modela sada će biti objašnjeni. Kao veličina grupe za treniranje korišteno je 8 primjera po grupi, korak učenja na početku je bio $3 \cdot 10^{-4}$ koji se nakon 15 epoha smanjio na 10^{-4} te je treniranje trajalo 28 epoha. Kao kraj treniranja odabrana je epoha u kojoj je rezultat modela na skupu za validaciju po mjeri srednjeg omjera presjeka i unije (mIoU) bio najbolji. Kao optimizacijski algoritam korišten je Adam koji je ranije opisan u ovom radu te su korištene preporučene vrijednosti hiperparametara.

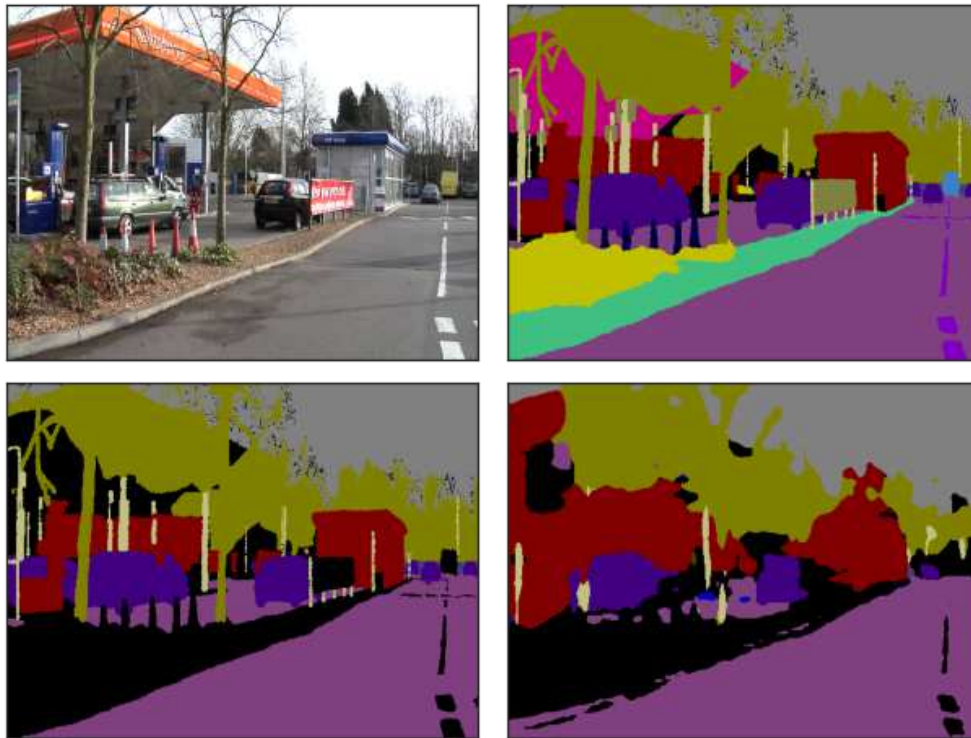
Ovime su dobiveni rezultati od 81.52% točnosti i 46.79% srednje vrijednosti presjeka i unije na skupu za testiranje. Ovime je model ResNet-18 bez ljestvičastog naduzorkovanja bio spreman za učenje na skupu podataka PROMISE12.

Nakon ovoga pokrenuli smo učenje modela za semantičku segmentaciju s ljestvičastim naduzorkovanjem opisanim ranije u radu. Bitno je napomenuti kako pri treniranju na ovom skupu podataka nije korišten sloj prostornog piramidalnog sažimanja. Kao veličina grupe za treniranje korišteno je 4 primjera po grupi, korak učenja na početku je bio $3 \cdot 10^{-4}$ koji se svikih 15 epoha smanjio za faktor 3 te je treniranje trajalo 90 epoha, a najbolji rezultat na validacijskom skupu postignut je nakon 75 epoha. Kao optimizacijski algoritam korišten je Adam koji je ranije opisan u ovom radu te su korištene preporučene vrijednosti momenata.

Ovime su dobiveni rezultati od 83.01% točnosti i 51.00% srednje vrijednosti presjeka i unije na skupu za testiranje.

Kao potencijalna poboljšanja modela ResNet-18 s ljestvičastim naduzorkovanjem predložimo nekoliko opcija:

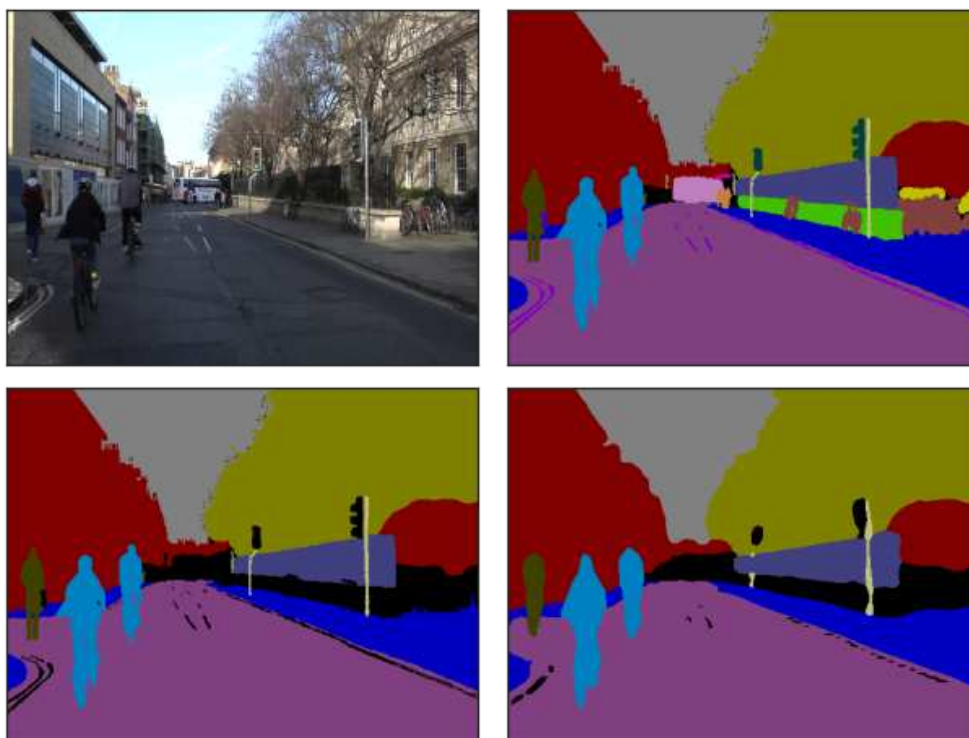
- Korištenje sloja za prostorno piramidalno sažimanje.
- Korištenje modela koji je predtreniran na skupu podataka ImageNet.



Slika 6.1: Primjeri loše segmentacije slike korištenjem modela s ljestvičastim naduzorkovanjem. Ovdje je prikazana originalna slika (gore lijevo), originalne oznake klasa (gore desno), originalne oznake klasa nakon smanjivanja broja klasa s 32 na 11 (dolje lijevo) i izlaz modela s ljestvičastim naduzorkovanjem (dolje desno).

Klasa	Omjer presjeka i unije (IoU) [%]
Klasa nastala kao unija svih izbačenih klasa	39.93
Zgrada	71.06
Stablo	66.70
Nebo	89.49
Automobil	66.55
Prometni znak	21.89
Cesta	88.99
Pješak	18.14
Ograda	30.08
Stup	12.04
Trotoar	71.50
Biciklist	35.59

Tablica 6.2: Rezultati omjera presjeka i unije (IoU) za svaku klasu na testnom skupu CamVid-a korištenjem modela ResNet-18 s ljestvičastim naduzorkovanjem.



Slika 6.2: Primjeri dobre segmentacije slike korištenjem modela s ljestvičastim naduzorkovanjem. Ovdje je prikazana originalna slika (gore lijevo), originalne oznake klasa (gore desno), originalne oznake klasa nakon smanjivanja broja klasa s 32 na 11 (dolje lijevo) i izlaz modela s ljestvičastim naduzorkovanjem (dolje desno).

6.3. Rezultati na skupu podataka PROMISE12

Prethodno validiran model ResNet-18 bez ljestvičastog naduzorkovanja sada je učen na podacima skupa podataka PROMISE12. Kao veličina grupe za treniranje korišteno je 8 primjera po grupi, korak učenja na početku je bio $3 \cdot 10^{-4}$ te je treniranje trajalo 40 epoha, a najbolji rezultat na validacijskom skupu podataka postignut je nakon 14 epoha. Kao kraj treniranja odabrana je epoha u kojoj je rezultat modela na skupu za validaciju po mjeri srednjeg omjera presjeka i unije (mIoU) bio najbolji. Korak učenja smanjivao se svakih 15 epoha za faktor 3. Optimizacijski algoritam i dalje je bio Adam sa preporučenim vrijednostima hiperparametara.

Model	Skup podataka	Točnost [%]	Srednji omjer presjeka i unije (mIoU) [%]
Model ResNet-18 bez ljestvičastog naduzorkovanja	skup za učenje	98.42	76.15
	skup za validaciju	98.00	65.68
	skup za testiranje	97.49	59.05
Model ResNet-18 s ljestvičastim naduzorkovanjem	skup za učenje	99.32	87.04
	skup za validaciju	99.24	81.07
	skup za testiranje	99.00	74.47

Tablica 6.3: Rezultati modela na skupu podataka PROMISE12.

Ovime su dobiveni rezultati od 97.49% točnosti i 59.05% srednje vrijednosti presjeka i unije na skupu za testiranje. Ovi rezultati bili su zadovoljavajući znajući da je referentna vrijednost mIoU na istom skupu podataka za model UNet $65.4\% \pm 0.8\%$ [18].

Nakon ovoga korišten je model s ljestvičastim naduzorkovanjem ranije opisan u ovom radu i validiran na skupu podataka CamVid. Kao veličina grupe za treniranje korišteno je 8 primjera po grupi, korak učenja na početku je bio $3 \cdot 10^{-4}$ te se on smanjivao za faktor 3 svakih 15 epoha. Treniranje je trajalo 50 epoha, a najbolji rezultat srednje vrijednosti omjera presjeka i unije (mIoU) na validacijskom skupu podataka postignut je nakon 28 epoha. Također, ponovno je korišten Adam kao optimizacijski algoritam sa preporučenim vrijednostima hiperparametara.

Ovime su dobiveni rezultati od 99.00% točnosti i 74.47% srednje vrijednosti presjeka i unije na skupu za testiranje. Ovdje je nadogradnja polaznog modela sa ResNet-18 arhitekturom na opisani model s ljestvičastim naduzorkovanjem dala kvalitetniji model. Ovaj model dao je bolje rezultate od modela UNet na istom skupu podataka (mIoU $65.4\% \pm 0.8\%$) [18].

Klasa	Omjer presjeka i unije (IoU) [%]
Tkivo koje ne pripada prostati	98.99
Prostata	49.96

Tablica 6.4: Rezultati omjera presjeka i unije (IoU) za svaku klasu na testnom skupu PRO-MISE12 korištenjem modela ResNet-18 s ljestvičastim naduzorkovanjem.

Vjerujemo da bi ovakav model s ljestvičastim naduzorkovanjem dao još bolje rezultate primjenom nekih od sljedećih opcija:

- Korištenje modela koji je predtreniran na skupu podataka ImageNet.
- Korištenje modela koji je predtreniran na nekom drugom skupu podataka sa slikama magnetske rezonancije.



Slika 6.3: Primjeri loše segmentacije slike korištenjem modela s ljestvičastim naduzorkovanjem. Ovdje je prikazana originalna slika (gore), originalne oznake klasa (sredina) i izlaz modela s ljestvičastim naduzorkovanjem (dolje).



Slika 6.4: Primjeri dobre segmentacije slike korištenjem modela s ljestvičastim naduzorkovanjem. Ovdje je prikazana originalna slika (gore), originalne oznake klasa (sredina) i izlaz modela s ljestvičastim naduzorkovanjem (dolje).

7. Zaključak

U ovom radu napravljen je uvod u područje dubokog učenja s naglaskom na duboke modele korištene u analizi slike kao što su ResNet-18 i model s ljestvičastim naduzorkovanjem rađenim po uzoru na [14]. Objasnjeni su osnovni koncepti kao što su neuron, neuronska mreža, aktivacijske funkcije, optimizacijski algoritmi te gradijentni spust.

Opisani su osnovni slojevi konvolucijske neuronske mreže kao i rezidualnih konvolucijskih neuronskih mreža. Objasnjen je i koncept konvolucijskih modela s ljestvičastim naduzorkovanjem.

Cilj rada bio je upotrijebiti jednostavni model ResNet-18 i model s ljestvičastim naduzorkovanjem na skupu podataka PROMISE12. Kako bi se validirala arhitektura ovih modela korišten je skup podataka CamVid (skupu podataka korištenim za razvoj samovozećih vozila) te su se validirane arhitekture koristile na skupu podataka PROMISE12 (skupu podataka slika magnetske rezonancije prostate i sve slike su labelirane, odnosno za svaki piksel svake od slika zabilježeno je radi li se o tkivu prostate ili okolnom tkivu).

Platforma korištena za učenje modela je Google Colab te su korišteni GPU resursi ponuđeni na ovoj platformi. Programska implementacija pisana je u programskom jeziku Python 3 te je korištena knjižnica PyTorch za izgradnju modela, učenje modela te transparentno korištenje GPU resursa.

Prikazani su rezultati modela ResNet-18 s i bez ljestvičatog naduzorkovanja na skupu podataka CamVid te jednom kada je proces učenja modela uhodan na skupu podataka CamVid isti modeli korišteni su na skupu podataka PROMISE12. Dobi-vene rezultate potencijalno je moguće poboljšati korištenjem predtreniranih modela na skupu podataka ImageNet ili na nekom od srodnih skupova podataka skupu podataka PROMISE12.

LITERATURA

- [1] Jaspreet, “A concise history of neural networks,” Apr 2019. URL <https://towardsdatascience.com/a-concise-history-of-neural-networks-2070655d3fec>.
- [2] G. Litjens, R. Toth, W. van de Ven, C. Hoeks, S. Kerkstra, B. van Ginneken, G. Vincent, G. Guillard, N. Birbeck, J. Zhang, R. Strand, F. Malmberg, Y. Ou, C. Davatzikos, M. Kirschner, F. Jung, J. Yuan, W. Qiu, Q. Gao, P. dwards, B. Maan, F. van der Heijden, S. Ghose, J. Mitra, J. Dowling, D. Barratt, H. Huisman, and A. Madabhushi, “Evaluation of prostate segmentation algorithms for mri: The promise12 challenge,” *Medical Image Analysis*, vol. 18, no. 2, pp. 359–373, 2014.
- [3] G. J. Brostow, J. Fauqueur, and R. Cipolla, “Semantic object classes in video: A high-definition ground truth database,” *Pattern Recognition Letters*, vol. xx, no. x, pp. xx–xx, 2008.
- [4] G. J. Brostow, J. Shotton, J. Fauqueur, and R. Cipolla, “Segmentation and recognition using structure from motion point clouds,” in *ECCV (1)*, pp. 44–57, 2008.
- [5] prof. dr. sc. Bojana Dalbelo Bašić, mr. sc. Marko Čupić, and mr. sc. Jan Šnajder, *Umjetne neuronske mreže*. 2008.
- [6] A. S. Avinash Sharma V, “Understanding activation functions in neural networks,” *Medium*, Mar 2017. <https://medium.com/the-theory-of-everything/understanding-activation-functions-in-neural-networks-9491262884e0>.
- [7] J. Vandit, “Handling class imbalanced data using a loss specifically made for it,” *Medium*, Sep 2019. URL <https://towardsdatascience.com/>

handling-class-imbalanced-data-using-a-loss-specifically-made-for-it-6e58fd65ffab.

- [8] J. Shadforth, “Understanding backpropagation in neural networks,” *Jacqui Shadforth*, Feb 2021. URL <https://jacqui.sh/understanding-backpropagation-in-neural-networks>.
- [9] N. Donges, “Gradient descent: An introduction to 1 of machine learning’s most popular algorithms,” *Built In*. URL <https://builtin.com/data-science/gradient-descent>.
- [10] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” 2014.
- [11] S. Ruder, “An overview of gradient descent optimization algorithms,” *Sebastian Ruder*, Mar 2020. URL <https://ruder.io/optimizing-gradient-descent/index.html#adam>.
- [12] S. Saha, “A comprehensive guide to convolutional neural networks - the eli5 way,” *Medium*, Dec 2018. URL <https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53>.
- [13] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” 2015.
- [14] M. Oršić and S. Šegvić, “Efficient semantic segmentation with pyramidal fusion,” *Science Direct*, 2019.
- [15] J. Jordan, “An overview of semantic image segmentation.,” *Jeremy Jordan*, Nov 2020. URL <https://www.jeremyjordan.me/semantic-segmentation/>.
- [16] Wikipedia, “Image segmentation — Wikipedia, the free encyclopedia.” <http://en.wikipedia.org/w/index.php?title=Image%20segmentation&oldid=1026516450>, 2021. [Online; accessed 06-June-2021].
- [17] “Classification: True vs. false and positive vs. negative,” *Google*, 2021. URL <https://developers.google.com/machine-learning/crash-course/classification/true-false-positive-negative>.

- [18] X. Yan, W. Jiang, Y. Shi, and C. Zhuo, “Ms-nas: Multi-scale neural architecture search for medical image segmentation,” 2020.

Semantička segmentacija slika prostate u slikama magnetske rezonancije

Sažetak

Primjena tehnologije u medicini sve je češća i ovaj rad bavi se segmentacijom tkiva prostate od okolnog tkiva u slikama magnetske rezonancije. Semantička segmentacija jedan je od zadataka računalnog vida. Ovaj rad primjenjuje tehnike dubokog učenja koje su se pokazale uspješnima u rješavanju problema semantičke segmentacije. Primarno su to rezidualni konvolucijski modela i konvolucijski modeli s ljestvičastim naduzorkovanjem. U ovom radu korišteni su ovi modeli u području semantičke segmentacije slika magnetske rezonancije na skupu podataka PROMISE12. Skup podataka CamVid korišten je kao skup podataka za validaciju arhitekture modela prije no što su isti modeli bili korišteni na skupu podataka PROMISE12.

Ključne riječi: neuronske mreža, duboko učenje, semantička segmentacija, magnetska rezonancija, prostata

Semantic segmentation of prostate images in images of magnetic resonance

Abstract

Application of technology in medicine is becoming more common and this paper is exploring segmentation of prostate tissue from the surrounding tissue in magnetic resonance images. Semantic segmentation is one of the problems of computer vision. This paper applies techniques of deep learning that have proved to be successful in semantic segmentation problems. Primarily those are residual models and models with ladder-style upsampling. In this paper, those models were used in semantic segmentation of magnetic resonance images on the PROMISE12 data set. CamVid dataset was used as a dataset for model architecture validation before using those models on the PROMISE12 dataset.

Keywords: neural networks, deep learning, semantic segmentation, magnetic resonance, prostate