Semantička segmentacija velikih satelitskih slika

Haralović, Marko

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 1651

SEMANTIČKA SEGMENTACIJA VELIKIH SATELITSKIH SLIKA

Marko Haralović

Zagreb, lipanj 2024.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 1651

SEMANTIČKA SEGMENTACIJA VELIKIH SATELITSKIH SLIKA

Marko Haralović

Zagreb, lipanj 2024.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

Zagreb, 4. ožujka 2024.

ZAVRŠNI ZADATAK br. 1651

Pristupnik:	**Marko Haralović (0036538058)**
Studij:	Elektrotehnika i informacijska tehnologija i Računarstvo
Modul:	Računarstvo
Mentor:	prof. dr. sc. Siniša Šegvić

Zadatak: **Semantička segmentacija velikih satelitskih slika**

Opis zadatka:

Semantička segmentacija važan je zadatak računalnog vida s mnogim važnim primjenama. U posljednje vrijeme vrlo zanimljive rezultate postižu duboki modeli koji znatan dio zaključivanja provode na poduzorkovanoj reprezentaciji. Ipak, jedna od glavnih prepreka prema boljoj generalizaciji i dalje je memorijska složenost algoritma backprop. Ovaj problem posebno je izražen kada želimo učiti na slikama vrlo visoke rezolucije kao što je to slučaj sa satelitskim snimkama. U okviru rada, potrebno je odabrati okvir za automatsku diferencijaciju te upoznati biblioteke za rukovanje matricama i slikama. Proučiti i ukratko opisati postojeće segmentacijske arhitekture utemeljene na konvolucijama i pažnji. Isprobati različite segmentacijske algoritme i istražiti njihov memorijski otisak. Uhodati učenje modela na neuravnoteženim podatcima. Validirati hiperparametre, vrednovati generalizacijsku izvedbu te prikazati i ocijeniti postignutu točnost. Radu priložiti izvorni i izvršni kod razvijenih postupaka, ispitne slijedove i rezultate te potrebna objašnjenja i dokumentaciju. Citirati korištenu literaturu i navesti dobivenu pomoć.

Rok za predaju rada: 14. lipnja 2024.

Ovim bih se putem htio zahvaliti mentoru prof.dr.sc. Siniši Šegviću te mag.ing. Marinu Kačanu, kao i ostalim članovima kabineta, na pomoći, savjetima i navođenju tokom izrade ovog završnog rada, kao i na predloženoj temi te omogućenim resursima.

Uvod

U ovom radu istražuje se područje dubokog učenja s posebnim naglaskom na primjenu u semantičkoj segmentaciji u računalnom vidu. Duboko učenje, kao podgrana strojnog učenja, revolucioniralo je mnoge aspekte umjetne inteligencije zahvaljujući svojoj sposobnosti da izvede zaključke iz složenih i obimnih podataka. Njegova primjena posebno je istaknuta u računalnom vidu, gdje se tehnike poput konvolucijskih neuronskih mreža (CNN) i algoritama utemeljenih na pažnji koriste za razumijevanje i analizu slika.

Semantička segmentacija slike je zadatak guste predikcije u računalnom vidu koji se temelji na dodjeljivanju semantičkih oznaka svakom pikselu slike. Ulazni parametarski prostor u ovom slučaju je slika, dok je očekivani rezultat dvodimenzionalna matrica iste veličine kao ulazna slika, gdje je svaki element te matrice klasa kojoj je pridružen odgovarajući ulazni piksel. Takva tehnologija nalazi primjene u raznim područjima, uključujući autonomna vozila, medicinsku dijagnostiku i udaljeno održavanje.

Rad započinje uvodom u osnove dubokog učenja, detaljno objašnjavajući strukturu i učenje dubokih modela. Posebna pažnja posvećuje se izazovima izračuna gradijenta te metodama koje se koriste za njihovo računanje, uključujući algoritme kao što je backpropagation. Učenje dubokih modela često uključuje složene tehnike optimizacije kako bi se poboljšala njihova učinkovitost i smanjila računalna zahtjevnost, što je ključno za obradu slika visoke rezolucije.

Nadalje, rad detaljno istražuje specifične primjene dubokog učenja u računalnom vidu, s fokusom na duboke modele trenirane na poduzorkovanoj reprezentaciji uzorka . Prikazane su različite arhitekture okvira za segmentaciju utemeljenih na pažnji te je dan detaljan pregled njihove arhitekture. Različite su segmentacijske arhitekture učene i testirane na javnom skupu podataka za semantičku segmentaciju te je dan detaljan pregled optimizacije hiperparametara treninga.

Opisane su tehnike i algoritmi koji se koriste za treniranje modela na neuravnoteženim skupovima podataka, kao što su tehnike za smanjenje memorijskog otiska, tehnike treniranja na teškim primjerima te tehnike prilagođenog otežavanja pojedinih klasa.

Zaključno, prikazani su rezultati vrednovanja različitih arhitektura modela te su rezultati uspoređeni sa modelima stanje tehnike na danom skupu podataka.

1. Uvod u duboko učenje

Strojno učenje jest programiranje računala na način da optimiziraju neki kriterij uspješnosti temeljem podatkovnih primjera ili prethodnog iskustva. Raspolažemo modelom koji je definiran do na neke parametre, a učenje se svodi na izvođenje algoritma koji optimizira parametre modela na temelju podataka ili prethodnog iskustva.[1]

Duboko je učenje grana strojnog učenja koja je posebno prikladna za rješavanje problema iz područja umjetne inteligencije. Specifično, model dubokog učenja jest tipično višeslojna neuronska mreža, poznata pod nazivom duboka neuronska mreža.

Algoritam strojnog (dubokog) učenja je definiran trojkom : model, gubitak te metoda optimizacije. [1] Karakteristika strojnog učenja jest razvijanje modela strojnog učenja koji unaprjeđuje svoju performansu u zadanom zadatku tokom vremena (tipično zvano proces učenja modela), skupljajući i učeći na podacima. Isto čini koristeći postupak obrade podataka te postupak optimiranja slobodnih parametara. Jasno, optimiranje slobodnih parametara iziskuje nužno postojanje cilja, odnosno kriterija cilja (kriterija optimizacije), na temelju kojega model uči i ažurira svoje slobodne parametre. Više u postupku učenja u sekciji **Učenje dubokih modela.**

Duboko učenje, kao i strojno učenje te umjetna inteligencija, postali su često susretani pojmovi te je raširenost modela dubokog učenja znatno veća nego prije do pred par desetljeća. Naime, tek su uspjesi modela u zadnjem desetljeću nadišli stanja tehnika na zadanim zadacima i postigli ljudsku preciznost.

Problemi dubokih modela jesu računalni resursi, nužnost velikih skupova podataka na kojima se trebaju učiti te nesigurnost tih postupaka. U praksi najčešće biva, iako je po definiciji potrebno svega 3 sloja za stvaranje primjerka dubokog modela, da je proces učenja i optimizacije modela računalno te memorijski zahtjevan proces zbog arhitekture modela, količine podataka te memorijske zahtjevnosti računanja gradijenata.

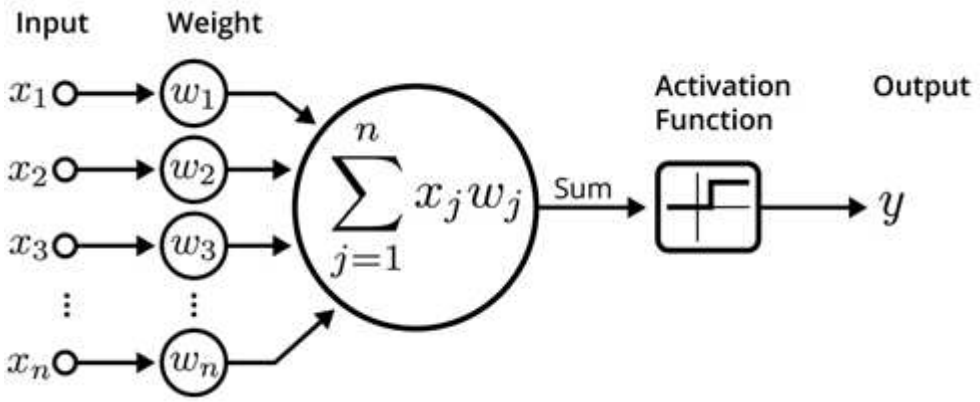
1.1. Duboki modeli

Definicija dubokog modela jest postojanje barem jednog skrivenog sloja. Ulazni i izlazni sloj su vidljivi slojevi, svaki ini sloj dodan u mrežu potpada u kategoriju skrivenih slojeva. Na ulazni se sloj šalju podaci, dok se s izlaznog sloja sakupljaju predikcije ili klasifikacije modela. Svaki sloj sadrži više čvorova zvanih neuronima te koristeći podatkovnu reprezentaciju podataka te parametre (težina i pristranosti), model uči obavljati zadatak koji mu je zadan.

Neuron, odnosno 'perceptron', kako ga je nazvao Frank Rosenblatt 1957. godine, jest najmanja jedinica u dubokom učenju koja se formalno može opisati sljedećom formulom:

$$y = f(\sum_{i=1}^{D} w_i x_i + b) \quad (1)$$

gdje je *f* aktivacijska funkcija, *D* dimenzija ulaznog prostora, *w* je vektor težina, *x* je vektor ulaznih podataka te *b* je skalar pristranosti, a vizualizacija je na Slika 1.



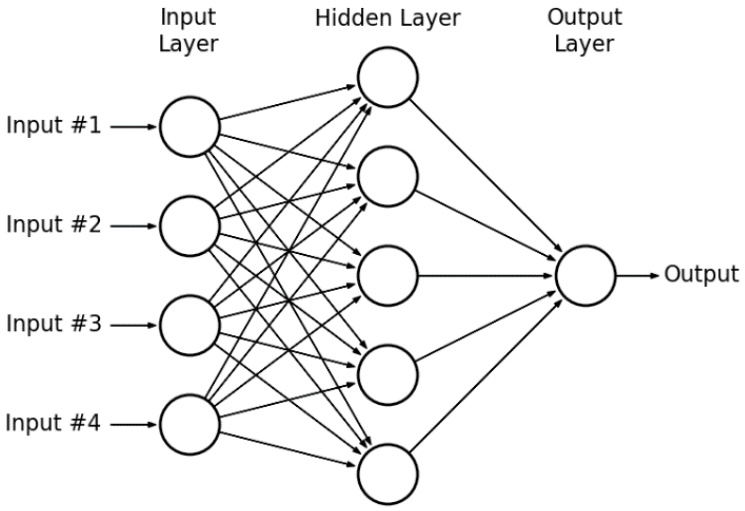
Slika 1. Vizualni prikaz modela neurona. [25]

Koncept učenja na razini perceptrona svodi se na ažuriranje težina i pristranosti tokom višestrukog prolaska podataka kroz slojeve neuronske mreže.

Aktivacijska funkcija (npr. sigmoidalna funkcija, zglobnica) jest nelinearna funkcija koja računa izlaz neurona. Važno svojstvo jest da je nelinearna u skrivenim slojevima, čime se sprječavanje građenja linearnog stroja koji bi bio sagrađen od linearnih perceptrona kao gradivnih jedinki.

Problem perceptrona jest diskriminacija linearno neodvojivih ulaznih prostora. Kako bi se takav problem razriješio, osmišljen jest *MLP*, eng. *Multi Layer Perceptron* (vidi Slika 2), odnosno kompleksniji sustav paralelno i linearno nadovezanih neurona, svakih sa svojim vektorima težina i skalara pristranosti. Upravo taj niz neurona omogućava veću 'perceptivnu'

moć modela, kako postoji značajno veći broj slobodnih parametara (svaki parametar koji se može algoritamski mijenjati tokom treninga jest slobodan parametar). Upravo je metrika veličine modela zadana preko broja slobodnih parametara u modelu.



Slika 2. Vizualni pregled MLP mreže [26]

Nedostatak prethodnog modela jest izostanak algoritma optimizacije. Zadaća metode optimizacije jest pronalazak parametara koji minimiziraju gubitak. Algoritmi optimizacije će biti opisan dodatno pod **Učenje dubokih modela.**

1.2. Učenje dubokih modela

Prostor učenja je skup podataka, primjerice skup slika, tekst, vremenski niz, i sl. Prostor učenja vektorski je prostor razapet vektorima primjera iz skupa podataka. Ukoliko postoji oznaka primjera, za isti kažemo da je označen, inače je neoznačen.

Ovisno o karakteristikama skupa podataka, označenosti te zadatku, razlikujemo nadzirano, nenadzirano i podržano učenje.

Nadzirano učenje podrazumijeva učenje na potpuno označenim podacima, podržano učenje na podacima za koje je dostupna povratna veza o kvaliteti međudjelovanja s okolinom te nenadzirano učenje koje podrazumijeva učenje na neoznačenim podacima, odnosno samo na podacima.

Model se sastoji od umreženih funkcija te kao što sam i naveo, slobodnih parametara. Cilj učenja parametarskog modela jest aproksimirati funkciju identiteta pretražujući prostor

stanja te mijenjajući slobodne parametre s ciljem najbolje aproksimacije prethodne funkcije, čime se učenje svodi na optimizacijski problem.

Mjera pogreške predikcija modela naziva se gubitak. Funkcija gubitka jest funkcija koja računa gubitak modela na skupu podataka ovisno o zadatku modela. U ovome se radu koristi unakrsna entropija na razini piksela koja ispituje gubitak na razini piksela, uspoređujući predikciju modela s oznakom piksela.

1.2.1. Unaprijedni i unazadni prolaz

Unaprijedni prolaz jest prolaz podataka iz ulaznog skupa podataka kroz mrežu te računanje izlaza. Ulaz se mapira na neurone u svakom sloju, koji koriste svoje težine i pristranosti za izračunavanje izlaza kojega šalju sljedećem sloju, dok je u zadnjem sloju izlaz aktivacijska funkcija ulaza predzadnjeg sloja, što je ovisno o zadatku. Unazadna propagacija, eng. back propagation, jest povratni prolaz kroz mrežu koji omogućava ažuriranje težina. Naime, za grešku izlaza mreže u odnosu na stvarnu vrijednost, računaju se gradijenti, odnosno gradijenti funkcije gubitka po svakom ulazu, odnosno težini te se na temelju tih vrijednosti model uči, koristeći gradijente gubitka po ulaznim parametrima te se u smjeru suprotno od njihova usmjerenja ažuriraju težine modela, s ciljem smanjenja gubitka.

1.2.2. Normalizacija grupe

Interni kovarijantni pomak [3] definiran je kao promjena u distribuciji aktivacija mreže nastale kao posljedice mijenjanja parametara mreže tokom treninga, zbog postojanja nasumičnosti u inicijalizaciji parametara i nasumičnosti i u samim ulaznim podacima. Normalizacija grupe je tehnika u treniranju dubokih neuronskih mreža koja normalizira ulaze mreže na srednju vrijednost 0 te varijancu 1. Kako u svakom prolasku modela na podacima nije moguće provesti sve podatke, na grupi podataka (eng. batch) se provodi normalizacija (stoga slijedi naziv eng. mini batch norm). Normalizacija na razini grupe omogućava korištenje više stope učenja bez pojave nestajućih ili eksplorirajućih gradijenata. Mreža teže divergira, manje konvergira u lokalne optimume te je teže da ulazi u mrežu koji znatno odudaraju od drugih 'odvuku' mrežu u globalno suboptimalnom smjeru.

1.2.3. Gradijentni spust

Optimizacije duboke neuronske mreže ostvaruje se minimizacijom funkcije greške. Algoritam gradijentnog spusta bazira se na iterativnom približavanju (konvergenciji) minimumu funkcije greške koristeći parcijalne derivacije funkcije pogreške po pojedinim težinama mreže.

Zadanu funkciju nazivamo funkcijom cilja (eng. objective, loss, cost, error function):

$$\mathrm{f}\colon \mathrm{R}(\mathrm{n}) \rightarrow \mathrm{R} \tag{2}$$

Osnovna formula za ažuriranje parametara u gradijentnom spustu je:

$$w_{t+1} = w_t - \eta \nabla E(w_t) \tag{3}$$

gdje je:

- w_t trenutna vrijednost parametara u vremenskom koraku t
- η stopa učenja,
- $\nabla E(w_t)$ gradijent funkcije gubitka E evaluiran u trenutnim parametrima w_t

Gradijent funkcije gubitka E u odnosu na parametre w obično se računa kao:

$$\nabla E(w) = \left(\frac{\partial E}{\partial w_1}, \frac{\partial E}{\partial w_2}, \ldots, \frac{\partial E}{\partial w_n}\right) \tag{4}$$

gdje n predstavlja broj parametara.

1.2.3.1 Stohastički gradijentni spust

Stohastički gradijentni spust (eng. Stochastic gradient descent, SGD) je optimizacijski algoritam koji za računanje pogreške u pojedinom koraku ne koristi cijeli skup podataka, već neki njegov manji dio.

Korak definiramo kao:

$$w_{t+1} = w_t - \eta \nabla E(w_t) \tag{5}$$

Pseudokod za SGD prikazan je na Slika 3

Algoritam 4 Algoritam za stohastički gradijentni spust

Inicijalizacija: $\boldsymbol{w}_0 \in \mathbb{R}^d$, $\alpha > 0$, $i_{max} \in \mathbb{N}$, $tol > 0$, $i = 0$, $\boldsymbol{w}_{-1} = \boldsymbol{w}_0$

while 1 **do**

1. Izmiješamo redosljed podataka za trening τ te dobijemo skup $\hat{\tau}$.

while $\hat{\tau} \neq \emptyset$ **do**

2. Provjerimo je li $i \leq i_{max}$. Ako nije, algoritam vraća $\boldsymbol{w}_{i-1}$.
3. Provjerimo je li $\|\boldsymbol{w}_i - \boldsymbol{w}_{i-1}\| > tol$. Ako nije, algoritam vraća $\boldsymbol{w}_{i-1}$.
4. Na slučajan način izaberemo točku $(\hat{\boldsymbol{x}}_i, \hat{y}_i) \in \hat{\tau}$.
5. Izračunamo gradijent funkcije f u toj točki: $\nabla f(\hat{\boldsymbol{x}}_i, \hat{y}_i)$.
6. Ažuriramo parametre: $\boldsymbol{w}_{i+1} = \boldsymbol{w}_i - \alpha \cdot \nabla f(\hat{\boldsymbol{x}}_i, \hat{y}_i)$.
7. Izbacimo točku $(\hat{\boldsymbol{x}}_i, \hat{y}_i)$ iz skupa za treniranje $\hat{\tau}$.
8. Povećamo broj napravljenih iteracija $i = i + 1$.

end while

end while

Slika 3. Pseudokod algoritma stohastički gradijentni spust [3]

Ukoliko se želi smanjiti računarska složenost algoritma može se koristiti manja grupa podataka (eng. mini-batch) za korekciju parametara, za koju se tehniku pokazalo da konvergira stabilnije jer se smanjuje varijanca osvježavanja parametara

1.2.4. Algoritam propagacije greške unazad

Propagacija pogreške unatrag (engl. backpropagation) je algoritam koji se koristi za efikasno izračunavanje gradijenta, tj. lokalne ovisnosti izlaza o pojedinim parametrima mreže. [4]

Generalni algoritam propagacije greške unazad mogao bi se iskazati ovako: [5]

1) Izračunaj unaprijedni prolaz za svaki par ulaza i izlaza $(\boldsymbol{x_d}, y_d)$ te spremi rezultat $\widehat{y_d}, a_j^k te\ o_j^k$ za svaki čvor j u sloju k prolaskom od ulaznog do izlaznog sloja mreže.
2) Izračunaj prolaz unatrag za svaki par ulaza i izlaza $(\boldsymbol{x_d}, y_d)$ te spremi rezultat $\frac{\partial E_d}{\partial w_{i,j}^k}$ za svaku težinu $w_{i,j}^k$ koja spaja čvor i k-1 sloja s čvorom n u sloju k prolaskom od zadnjeg ka ulaznog sloju
   a. Evaluiraj pogrešku zadnjeg sloja δ_1^m
   b. Propagiraj greške skrivenih slojeva δ_j^k, krenuvši od zadnjeg skrivenog sloja k = m-1
   c. Izračunaj parcijalne derivacije pojedine pogreške E_d u odnosu na težinu $w_{i,j}^k$

3) Za svaki ulazno-izlazni par $\frac{\partial E_d}{\partial w_{i,j}^k}$ sumiraj gradijente u ukupan zbroj $\frac{\partial E\,(X,\theta)}{\partial w_{i,j}^k}$ za cijeli skup ulazno-izlaznih parova
4) Ažuriraj težine modela koristeći stopu učenja η i ukupan gradijent $\frac{\partial E\,(X,\theta)}{\partial w_{i,j}^k}$

1.2.5. Eksplodirajući i nestajući gradijenti

Eksplodirajući gradijenti događaju se kada veličine gradijenata funkcije gubitka postaju izuzetno velike tijekom algoritma propagacije greške unazad. To može dovesti do prekomjernih, vrlo velikih ažuriranja težina, što može uzrokovati da se modelova stanja destabiliziraju. U praksi, to znači da težine mreže mogu postati prevelike ili nan vrijednosti, što često dovodi do modela koji daje besmislen izlaz i ne može konvergirati prema optimalnom rješenju.

Nestajući gradijenti događaju se kada veličine gradijenata postanu vrlo male, eksponencijalno brzo se smanjujući dok se propagiraju kroz mrežu, osobito u dubokim mrežama. To onemogućuje učinkovito ažuriranje težina u početnim slojevima jer gradijenti postaju premali da bi imali bilo kakav značajan utjecaj. Kao rezultat, težine u početnim slojevima ostaju gotovo nepromijenjene, što može spriječiti mrežu da nauči adekvatne značajke iz podataka i konvergira prema dobrom rješenju.

2. Duboko učenje u računalnom vidu

2.1. Računalni vid

Računalni vid (eng. computer vision) je grana umjetne inteligencije koja koristi modele strojnog i dubokog učenja u svrhu smislenog izvlačenja i razumijevanje informacija iz digitalnih slika, videa i ostalih vizualnih ulaza. Modeli se uče detekciji raspoznavanja uzoraka i značajki u podacima, primjerice rubova, oblika i kontura, odnosno temeljni je zadatak procesuiranje slike, raspoznavanje uzoraka te detekcija objekata.

U računalnom se vidu koriste i konvolucijske neuronske mreže koje ću u nastavku opisati.

2.2. Konvolucijska neuronska mreža

Konvolucijska neuronska mreža (eng. Convolutional Neural Network, CNN) je duboka neuronska mreža s barem jednim konvolucijskim slojem.

Konvoluciju definiramo kao skalarni produkt jedne funkcije s obzirom na posmaknutu i reflektiranu drugu funkciju.

Konvolucijski sloj ulaznu sliku apstrahira u mapu značajki (eng. feature map). Odziv ulaza u odnosu na konvoluciju ne mijenja se pomakom (ekvivarijantnost s obzirom na pomak), a još jedna je značajka i ta da se broj slobodnih parametara smanjuje, što smanjuje količinu i vrijeme potrebno da se slobodni parametri prilagode.

U konvolucijskom sloju, kao i u svim ostalim slojevima, svaki neuron dobiva kao ulaz izlaz iz prethodnog sloja. Pojam vezan za broj neurona prethodnog sloja koje neuron u sljedećem sloju 'vidi' naziva se receptivno polje (eng. receptive field). Receptivno polje neurona u unaprijednom sloju jest cijeli prethodni sloj, dok je kod konvolucijskih neurona riječ o NxN neurona iz prethodnog sloja. Receptivno polje isto tako raste dubinom. Na taj se način i modelira lokalno susjedstvo, a sve transformacije su lokalne, što znači kako lokalno susjedstvo piksela direktno utječe na izlazne piksele.

Konvolucijske mreže najčešće sadrže i sažimajuće slojeve (eng. Pooling layers), koji mogu biti lokalni ili globalni. Svrha tih je slojeva redukcija dimenzionalnosti podataka kombinirajući izlaze skupina neurona sloja u jedan neuron njemu slijednog sloja. Lokalni

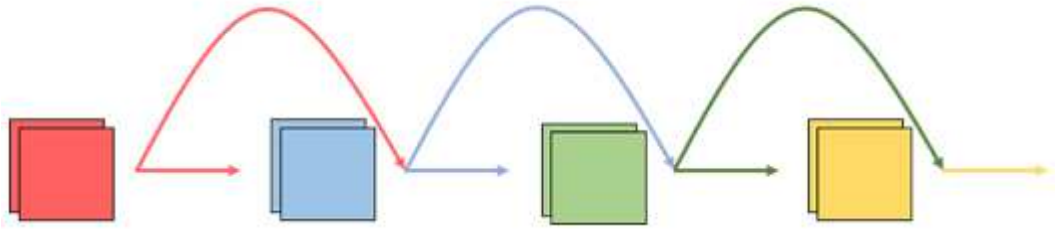
sažimajući sloj sažima male kvadratiće neurona (npr. 2x2), dok globalni sažimajući slojevi sažimaju na razini mapa značajki.

2.2.1. ResNet

ResNet[6] (od eng. Residual Neural Network) je obitelj konvolucijskih neuronskih mreža. ResNet je predstavljen u radu 'Deep Residual Learning for Image Recognition' iz 2015 te je posebnu pažnju privukao rješavajući problem nestajućih gradijenata. Naime, u pravilu je cilj uvijek bio omogućiti dubokim neuronskim mrežama da budu čim dublje, ne bi li se povećao parametarski prostor koji se može istražiti u svrhu povećanja sposobnosti modela. No, upravo tim produbljavanjem mreže pojavio se problem nestajanja gradijenata, odnosno male vrijednosti gradijenata su nakon ulančanog prolaza kroz mrežu bile smanjene do nule, što za direktnu posljedicu ima da se težine ne mijenjaju u smjeru suprotnom od gradijenta, već da ostaju iste te da se učenje duboke neuronske mreže u biti ni ne događa.

Arhitektura ResNet mreže omogućava mu da premosti problem nestajućih gradijenata. Naime, mreža je bazirana na rezidualnim blokovima, odnosno oformljena je naslagivanjem više rezidualnih blokova. Rezidualni blok je oformljen kada se aktivacija sloja u neuronskoj mreži poveže sa dubljim slojevima u mreži, pritom preskačući slojeve između.

Veza koja se koristi za prethodno opisano povezivanje naziva se rezidualna veza, a operacija dodavanja prethodnog izlaza ostvarena je tehnikom eng. skip connection (vidi Slika 4), koja izvodi mapiranje identiteta kako bi spojila ulaz i izlaz u podmrežu. Prethodno opisano mapiranje dovodi do propagacije signala i u unaprijednom i u povratnom prolazu kroz mrežu.

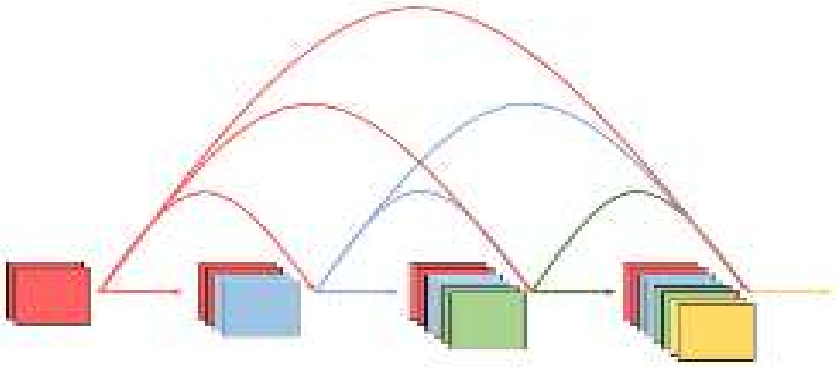


Slika 4. Vizualni pregled rezidualnih veza [6]

Rezidualne veze učinkovito zaglađuju funkciju gubitka te poboljšavaju tok gradijenta prema ranijim slojevima, time rješavajući problem nestajanja/eksplozije gradijenata.

2.2.2. DenseNet

DenseNet [7] (eng. Densely Connected Convolutional Networks) predstavlja tip konvolucijske neuronske mreže arhitekture u kojoj je svaki sloj direktno povezan s svakim drugim slojem u unaprijednom smjeru. Ova arhitektura omogućava svakom sloju da prima značajke svih prethodnih slojeva, što je značajno različito od tradicionalnih konvolucijskih mreža koje povezuju slojeve uglavnom samo s neposredno sljedećim slojevima. Ključna inovacija DenseNeta je gusta povezanost koja ublažava problem nestajućeg gradijenta, pojačava propagaciju značajki, potiče ponovnu upotrebu značajki i smanjuje broj parametara. U DenseNetu, broj direktnih veza među slojevima može se izračunati kao suma prvih L−1 prirodnih brojeva, gdje je L broj slojeva. Na primjer, za mrežu s četiri sloja postoji 6 direktnih veza što olakšava efikasnu razmjenu značajki i smanjuje redundanciju u učenju.



Slika 5: Prikaz direktnih veza u DenseNet arhitekturi [7]

2.2.3. MobileNet V2

MobileNetV2 [8] je arhitektura konvolucijske neuronske mreže koja teži dobrim performansama na mobilnim uređajima. Temelji se na obrnutoj rezidualnoj strukturi gdje su rezidualne veze između uskih slojeva (eng. bottleneck layers). Međusloj za širenje koristi lagane dubinske odvojive konvolucije za filtriranje značajki kao izvor nelinearnosti.

U cjelini, arhitektura MobileNetV2 sadrži početni potpuno konvolucijski sloj s 32 filtra, a zatim slijedi 19 rezidualnih uskih slojeva.

2.3. Vision tranformeri

Vision transformer (ViT) [9] je arhitektura za procesuiranje slika utemeljena na samo-pažnji (eng. self-attention). Arhitektura se sastoji od niza blokova transformatora. Svaki blok uključuje dva pod-sloja: sloj višestruke samo-pažnje te unaprijedni sloj. Sloj samo-pažnje izračunava težine pažnje za svaki piksel u slici temeljem njegove povezanosti s ostalim pikselima, dok unaprijedni sloj primjenjuje nelinearnu transformaciju na izlaz iz sloja samo-pažnje. Višestruka pažnja omogućava modelu istovremeno usmjeravanje na različite dijelove ulazne sekvence-slike.

ViT također uključuje dodatni sloj za ugrađivanje dijelova (eng. patches) slike , koji sliku dijeli na dijelove fiksne veličine i svaku mapira u visokodimenzionalnu vektorsku reprezentaciju. To radi kako transformeri iziskuju za ulaz niz tokena, dok je ovdje token dio originalne slike. Te ugrađene dijelove zatim šalju u blokove transformatora na daljnju obradu. Konačni izlaz arhitekture ViT je predikcija klase, dobivena prolaskom izlaza posljednjeg bloka transformatora kroz glavu za klasifikaciju, koja obično sadrži jedan potpuno povezani sloj.

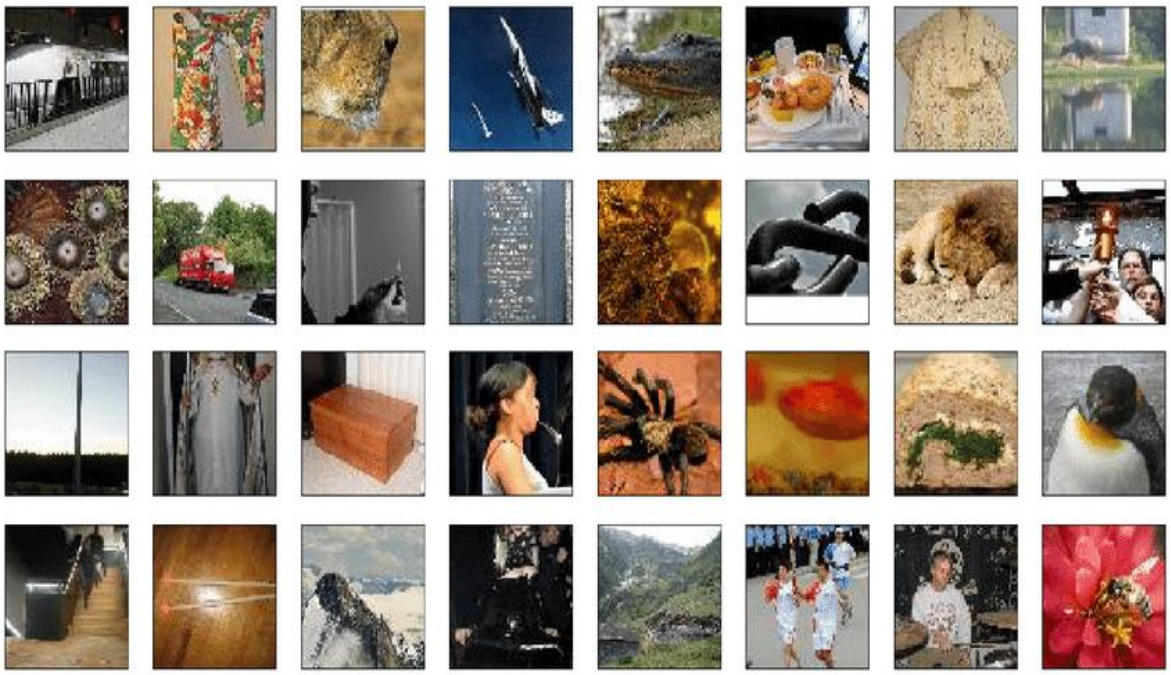
Za razliku od transformera u obradi prirodnog jezika, koji se sastoji od grana dekoder i enkodera, ViT koristi samo enkoder, a izlaz iz enkodera se šalje neuronskoj mreži za predikciju.

Performanse ViT-a na mnogim zadacima nadilaze konvolucijske neuronske mreže, primjerice na klasifikaciji slika na skupu podataka ImageNet prvo mjesto na temelju točnosti odnosi upravo model baziran na arhitekturi ViT-a.

3. Primjena dubokog učenja na ImageNetu

3.1. ImageNet

ImageNet (vidi Slika 6) je velika baza podataka slika dizajnirana za korištenje u istraživanju prepoznavanja objekata na slikama. Sadrži preko 14 milijuna slika koje su ručno označene s popisom objekata koji su na njima, dok ih oko milijun sadrži i pravokutne oznake za same objekte na slikama. Baza sadrži preko 20000 klasa te je javno dostupna.



Slika 6: Primjer slika iz baze podataka ImageNet

3.2. Predtreniranje na ImageNetu

Zanimljiva ideja je koristiti model treniran na ImageNetu, dakle klasifikacijski model, te ga prilagoditi drugom zadatku. Prilagodba znači da se koriste svi slojevi osim zadnjih nekoliko koji se odrežu, a ostatak mreže se spoji s prednjim krajem nove mreže koja će se koristiti na novom zadatku.

Takva je ideja interesantna iz više razloga, ponajprije jer su naslijeđeni slojevi klasifikacijske mreže, koja je naučena na ImageNet slikama, već naučeni, čime izravno utječemo na broj parametara koje je potrebno naučiti na novome zadatku. Time također smanjujemo potrebnu veličinu skupa podataka za učenje te ostvarujemo bržu konvergenciju i višu točnost.

Dodatno, takav prijenos znanja s većeg skupa podataka ima regularizacijski efekt u treningu mreže. [10]

Mreža čije se težine nasljeđuju naziva se predtrenirana (eng. pretrained) mreža, dok se proces učenja mreže koja sadrži slojeve predtrenirane mreže na novom skupu podataka naziva fino uglađivanje (eng. fine-tuning).

Naravno, korištenje ovog pristupa može imati svoje probleme. Ukoliko govorimo o premalom novom skupu podataka, postoji mogućnost da se model prenauči te da nauči šum ili specifične neznačajne značajke te da izgubi moć generalizacije, također je moguće prenijeti pristranost s originalnog skupa podataka, a ukoliko su domene dvaju zadataka znatno različite, model može imati lošije rezultate kao posljedicu slabije prilagodbe.

4. Tehnike segmentacije

4.1. Segmentacija slike

Segmentacija slike je zadatak u računalnom vidu, definiran kao proces analize slike koja se dijeli na različite segmente te se podaci u svakoj od tih regija (pikseli) klasificiraju.

Temeljna tri zadatka u segmentaciji slike su semantička segmentacija slike, panoptička segmentacija te segmentacija instanci. Svaki od zadataka temelji se na značajkama slike poput boja, kontrasta, pozicije unutar slike, i sl.

Segmentacija instanci fokusira se na razredima u slici koji mogu biti pobrojani, a izlaz je detekcija objekta te segmentacijska maska za svaki detektirani objekt.

Panoptička segmentacija se temelji na semantičkoj segmentaciji i na segmentaciji instanci, odnosno nakon semantičke analize slike se detektiraju i segmentiraju individualni objekti, dok se svim pikselima neprebrojivih razreda pridruži isti identifikator.

Detaljnije o semantičkoj segmentaciji slike slijedi.

4.2. Semantička segmentacija slike

Semantička segmentacija slike je zadatak guste predikcije (detaljnije vidi 4.2.1) u računalnom vidu koji se temelji na dodjeljivanju semantičkih oznaka svakom pikselu slike. Ulazni parametarski prostor u ovom slučaju je sama slika, dok je očekivani rezultat dvodimenzionalna matrica iste veličine kao ulazna slika, gdje je svaki element te matrice klasa kojoj je pridružen odgovarajući ulazni piksel.

Primjeri korištenja semantičke segmentacije slike su razni, navodim neke najzvučnije: autonomna vožnja, medicinska dijagnoza, analiza fotografija (biranje efekata i filtera na temelju semantički instanci u slici), udaljeno nadziranje površina (npr. monitoriranje javnih površina, farma, detekcija požara).

Gusto označene slike potrebne za trening dubokih modela, kojih je često potrebno pozamašan broj, uzimajući u obzir i činjenicu da iste mogu biti memorijski zahtjevne zbog visoke razlučivosti, dodatno otežavaju ovaj zadatak te ekstreman pritisak postavljaju na računalne resurse potrebne da se ovi modeli treniraju, evaluiraju te na koncu i koriste na krajnjim uređajima (od eng. edge devices). Takvi uvjeti zahtijevaju da mrežna arhitektura

bude što jednostavnija kako bi se smanjila latencija modela, ne bi li se vrijeme inferencije dostatno smanjilo za potrebe trenutne odluke u bilo kojoj domeni primjene.

Prikladni modeli za izlučivanje značajki iz slike su konvolucijske neuronske mreže te je njihovo korištenje rašireno, a i predmet su ovoga rad, tako da ću nadalje opisivati arhitekture koje koriste te modele.

U sljedećim ću poglavljima pokriti neke probleme guste predikcije, objasniti pojmove fuzije značajki te poduzorkovane reprezentacije uzorka, dati detaljniji pregled najpoznatijih javno dostupnih skupova podataka u domeni zadataka semantičke segmentacije slika visoke rezolucije, proći kroz najznačajnije modele za semantičku analizu slike te dati detaljniji pregled dvaju arhitektura koje su ekstenzivno korištene u ovom radu.

4.2.1. Problemi guste predikcije

Temeljni problemi su vremenska složenost, obnova ulazne rezolucije te povećanje receptivnog polja modela.

Zadaci guste predikcije, primjerice semantička segmentacija slike, sa sobom vuku iznimnu računarsku složenost. Naime, zadatak semantičke segmentacije zahtjeva visoku rezoluciju na ulazu te na izlazu. Korištenje konvolucijskih neuronskih mreža u semantičkoj analizi slike je neefikasno za originalne veličine slika, stoga je potrebno uzorak na ulazu poduzorkovati, čime se gubi rezolucija ulaza. Slojevi sažimanja u konvolucijskim neuronskim mrežama su od iznimne koristi za smanjenje računalne kompleksnosti. Cijena smanjenja računarske složenosti jest dakako gubitak određenih informacija (lokalnih), što je potrebno razriješiti prilikom naduzorkovanja poduzorkovane slike.

Pritom se javljaju i problem detekcije iznimno malih i iznimno velikih objekata. Mali objekti mogu ostati neprepoznati zbog niske rezolucije predikcija na razini piksela, što znači kako bi reprezentacija takvih objekata bila neznatna u mapi značajka koja bi nastala poduzorkovanjem ulazne slike. S druge strane, problem segmentacije velikih objekata je potreba za velikim receptivnim poljem, kako diskriminacija prostora nastala sažimanjem, odnosno lokalnim susjedstvima, nije dovoljna te je te objekte problematično raspoznati ukoliko veličina objekta (kontekst) nadilazi veličinu receptivnog polja.

Rješenja probleme raspoznavanja malih i velikih objekata usko je povezano s rješavanjem problema obnove rezolucije ulaznih podataka i povećanja receptivnog polja.

Jedan od načina za izbjegavanje smanjenja rezolucije je korištenje dilatirane konvolucije kojima se korak konvolucije u nekim blokovima smanji na 1 u svrhu izbjegavanja poduzorkovanja, npr. zamjena konvolucija s korakom 2 s konvolucijama bez koraka smanjuje gubitak informacija i povećava rezoluciju dubokih latentnih reprezentacija, iako to može značajno povećati računalnu složenost.

Također je korišteno prostorno piramidalno sažimanje (eng. spatial pyramid pooling, SPP) koje za ideju ima da se širi kontekst ugradi u lokalne značajke ne bi li se povećalo receptivno polje te omogućilo raspoznavanje objekata različitih veličina (ref. prethodno navedeni problem raspoznavanja velikih objekata), ljestvičasto naduzorkovanje gdje je cilj nadoknaditi poduzorkovanje korištenjem slojeva na različitim dubinama pomoću enkoder-dekoder arhitekture s bočnim vezama kombinirajući semantički bogate duboke slojeve s prostorno bogatim plićim slojevima. To smanjuje složenost dok istovremeno obnavlja rezoluciju predikcija, čime se omogućuje efikasnije prepoznavanje u stvarnom vremenu. Ove tehnike, iako povećavaju receptivno polje, mogu utjecati na generalizaciju zbog velikog kapaciteta modela.

4.2.2. Fuzija značajki

Fuzija značajki je tehnika kombiniranja značajki iz više slojeva unutar modela kako bi se poboljšala performansa modela. Cilj je stvoriti bogatiju reprezentaciju uzorka koristeći vrijednosti značajki iz više slojeva, gdje se apstraktnije značajke iz dubljih slojeva kombiniraju s površinskih značajkama koje sadrže prostorne detalje.

Pojam ću objasniti na primjeru enkoder-dekoder arhitekture. Zadaća je enkodera smanjiti prostornu dimenzionalnost ulazne slike, čime se dobivaju informacijski bogatije mape značajki. Dekoder pak postepeno povećava prostornu dimenzionalnost predikcija koristeći informacije enkodera, a bočni (lateralni) spojevi omogućavaju da se značajke iz različitih slojeva enkodera prenesu u odgovarajuće slojeve dekodera.

Prethodno opisane preskočne veze u ResNetu (eng. skip connections) slične su lateralnim spojevima.

4.2.3. Segmentacija utemeljena na pažnji

Semantička segmentacija slika utemeljena na pažnji koristi mehanizme pažnje za detaljnu analizu i razdvajanje različitih objekata unutar slike. Početni korak uključuje ekstrakciju značajki iz slike pomoću konvolucijskih neuronskih mreža. Mehanizmi pažnje zatim fokusiraju model na ključne dijelove slike, poboljšavajući točnost segmentacije. Pažnja se može primjenjivati na različite aspekte slike, uključujući prostorne značajke (eng. spatial features) ili kanale značajki (eng. feature channels). Nakon procesa pažnje, dekoder rekonstruira segmente slike, dodjeljujući svakom segmentu odgovarajuću klasu. Ovaj pristup omogućuje precizno razlikovanje objekata unutar slike, kao što su ljudi, vozila ili drveće. Krajnji rezultat je jasno segmentirana slika s objektima različitih klasa. Segmentacija utemeljena na pažnji posebno je korisna u primjenama gdje je potrebna visoka razina detalja, poput medicinske dijagnostike i autonomnih vozila. Metoda omogućava modelima da efikasno identificiraju i razumiju različite elemente u kompleksnim vizualnim scenama. Time se postiže naprednija analiza slika, što vodi boljem razumijevanju i interpretaciji vizualnih informacija.

4.3. Skupovi podataka za semantičku segmentaciju

Skupovi podataka sastoje se od skupa slika (odnosno više skupova ukoliko su slike podijeljene na slike za treniranje, testiranje i validaciju modela) te popratnih semantičkih oznaka slika.

Temeljno su pregledani skupovi podataka s visoko rezolutnim slikama koji odgovaraju prethodnom opisu te koji su javno dostupni.

Pregledano je više skupova podataka koji bi bili zanimljivi za zadatak semantičke segmentacije, fokusirajući se pritom na broj razreda (taksonomiju), broj slika u skupovima podataka, veličinama slika u piksela, ukupnom broju piksela te rezoluciji slika u m/pikselu. U tablici slijedi detaljniji prikaz.

Tablica 1.: Pregled skupova podataka korištenih u zadacima semantičke klasifikacije.

SKUP PODATAKA	**BROJ RAZREDA**	**BROJ SLIKA**	**VELIČINA SLIKE U PIXELIMA**	**UKUPAN BROJ PIKSELA/MIL**	**REZOLUCIJA m/pixel**
DeepGlobe Road Extraction	2	6626 RGB	1024x1024	6948	0,5
Deep Globe Land Cover Classification	*7*	*803 RGB*	*2448X2448*	*4812*	*0,5*
Deep Globe Building Detection	2	24.586 RGB	650x650	10.4	0,3 te 1,24 *
URUR	8	3008 UHR	5120X5120	79.000	0,2
Inria Aerial	8	360 RGB	5000x5000	25	0,3
UUD6	6	141 RGB	4000 × 3000 ili 4096 × 2160	8.9	0,3
Massachusetts buildings dataset	2	151 RGB	1500x1500	340	1,0
Massachusetts roads dataset	2	1171 RGB	1500x1500	2493	1,0
DSTL satellite imagery	11	25** RGB	3348x3392	284	0,31,1,24, 7,5 ***
BSB-Aerial	17	3400 RGB	512x512	1380	0,24
Potsdam Dataset	6	38 RGB	6000x6000	1368	0,05
Vaihingen Dataset	6	33 RGB	2494x2494	205	0,08
UAVid	9	420 RGB	3840x2160 ili 4096x2160	2500	0,5

* rezolucija 30cm/pixel za pankromatsku sliku , 1.24 za multispektralnu sliku

** dvije vrste slika; s 3 te s 16 bandova

*** rezolucija: pankromatska 0.31 m/piksel, multispektralna 1.24m/piksel, SWIR 7.5m/piksel

4.3.1. DeepGlobe land cover classification dataset

DeepGlobe je izazov iz 2018 koji se sastoji od 3 zadatka: izazov izvlačenja cesta, izazov detekcije zgradi te izazov klasifikacije zemljišnog pokrova.

Izazov klasifikacije zemljišnog pokrova predstavlja izazov automatske klasifikacije tipova zemljišnog pokrova. Ovaj problem definiran je kao zadatak segmentacije više klasa za otkrivanje područja urbanih, poljoprivrednih, pašnjačkih, šumskih, vodenih, pustih i nepoznatih površina.

Skup podataka sadrži 803 satelitske snimke u RGB formatu, veličine 2448x2448 piksela. Snimke su prikupljene satelitom tvrtke DigitalGlobe i imaju rezoluciju piksela od 50 cm.

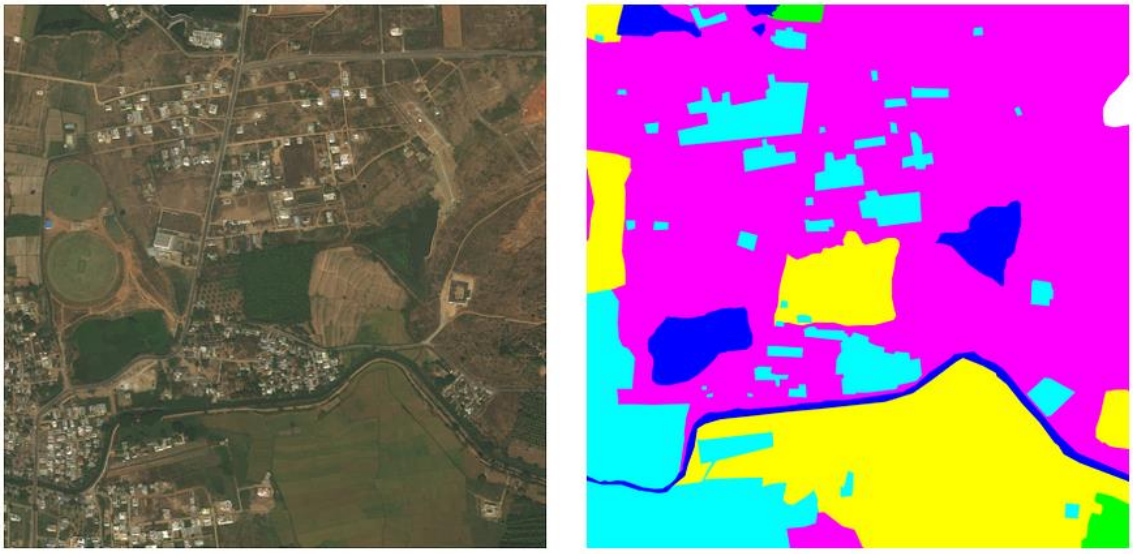
Svaka satelitska slika uparena je s maskom za anotaciju zemljišnog pokrova. Maska je RGB slika s 7 klasa oznaka, koristeći kodiranje bojama (R, G, B) kako slijedi:

- Urban Land: 0,255,255 -umjetne, izgrađene površine s ljudskim artefaktima
- Agricultural land: 255,255,0 - farme, planirane plantaže, obradive površine, voćnjaci, vinogradi, rasadnici i ukrasne hortikulturne površine
- Range Land: 255,0,255 - nešumovita, nefarmerska, zelena površina, trava
- Forest: 0,255,0 - zemljište s x% gustoće krošnji drveća plus čistina
- Water: 0,0,255 - rijeke, oceani, jezera, močvare, ribnjaci
- Barren land 255,255,255 - planine, zemlja, stijene, pustinje, plaže, bez vegetacije
- Other : 0,0,0 - oblaci i ostalo

Tablica 2.: Prikaz zastupljenosti pojedinih klasa u DeepGlobe land cover skupu podataka

Klasa	Broj piksela u milijunima	Udio skupa podataka
Urban Land	461.19	11.27%
Agricultural land	2379.65	58.14%
Range Land	343.12	8.38%
Forest	444.92	10.87%
Water	138.39	3.38%
Barren land	323.39	7.90%
Other	2.35	0.06%

Slika 7: Primjer slike (lijevo) te označene maske (desno) iz DeepGlobe land cover skupa podataka



5. Napredne tehnike i optimizacije

5.1. Online Hard Example Mining – OHEM

OHEM [11] je algoritam korišten kod treniranja dubokog modela, primarno u zadacima detekcije objekata. Algoritam se temelji na odabiru 'teških' primjera tokom treninga, odnosno primjera na kojima model pogrešno predviđa ili ima nisku razinu pouzdanosti u svoje predikcije. Cilj OHEM-a je optimizirati performanse modela tako što će ga natjerati da se usredotoči na te teške primjere, umjesto da jednako tretira sve primjere u skupu za treniranje.

U procesu treniranja, za svaki se primjer u mini grupi izračuna gubitak te se na temelju najviših gubitka biraju teški primjeri za nastavak treninga na kojima se ponavlja prolaz modela, čime se na težim primjerima računaju gradijenti i slijedno mijenjaju težine.

Kod segmentacije slike, unakrsna se entropija računa za svaki piksel, što rezultira mapom gubitka koja se zatim maskira na temelju vrijednosti gubitka po pikselu veće ili manje od neke predefinirane granice (stvara se binarna maska). Maska se pomnoži s gubicima po pikselu te se produkt koristi u povratnom prolazu (eng. backpropagation). Drugim riječima, samo se za 'teške' piskele računaju gradijenti.

5.2. Inverse Frequency Weighting- IFW

Problem nebalansiranosti klasa česta je pojava u zadacima strojnog (dubokog) učenja. Takav je slučaj i na DeepGlobe Land Cover skupu podataka, gdje klasa *agrikulturna zemlja* predstavlja 58% oznaka, dok klasa *vode* tek 3.38% svih oznaka te klasa *ostalo* skoro neznatnih 0.06%. Generalno, nebalansiranost u broju primjeraka klase vrlo lako može dovesti do pristranosti modela gdje na učenje modela primarno utječe većinska klasa, a manja je klasa zanemarena, pogotovo ukoliko se provodi učenje gubitkom koji jednake težine daje svim klasama i oznakama klasa.

U nekim zadacima taj problem možda nije toliko važan, a uz odabir metrika koje zorno ne prikazuju generalizira li model po klasama već koje samo plastično prikazuju ukupno stanje predikcija modela, primjerice točnost, moguće je niti ne primijetiti ovaj problem.

No, znajući da je klasa u pitanju od interesa te uzimajući mIoU (vidi 8.1) kao primarnu metriku evaluacije, jasno je kako bi prethodni problem idealno trebao biti premošten.

Inverse frequency weighting je metoda računanja težina klasa na temelju broja uzoraka svake klase u podacima, odnosno svakoj se klasi pridjeli težina obrnuto proporcionalnu njenoj učestalosti u skupu podataka. Tada se težina klase može izračunati kao

$$w_c = \left(\frac{P}{P_{i=c}}\right) * \alpha \quad (6)$$

Ovako izračunate težine moguće je poslati kao parametar modula unakrsne entropije (eng. cross entropy)

5.3. Prijenos podataka na GPU

U kontekstu operacijskih sustava, straničenje je metoda upravljanja memorijom gdje se podaci premještaju iz sekundarnog spremišta u glavnu memoriju u obliku blokova iste veličine, poznatih kao "stranice". Straničenje omogućava efikasnije korištenje memorije i optimizaciju pristupa podacima. [11]

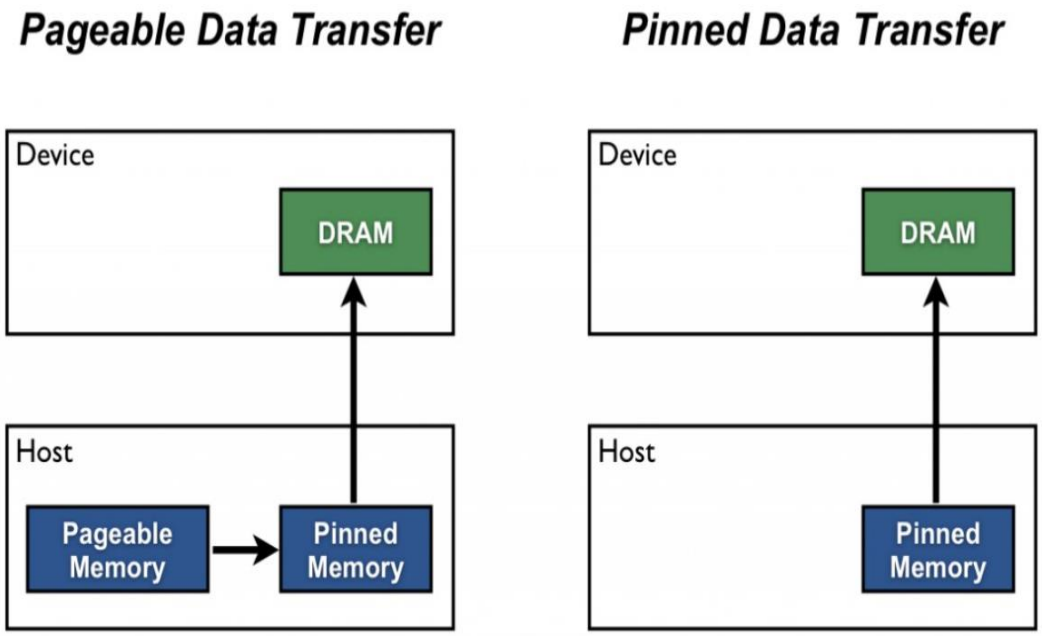
Prijenos podataka s CPU-a na GPU uključuje određene vremenske troškove, stoga je važno minimizirati te prijenose kako bi se optimizirala ukupna performansa. Budući da GPU ne može izravno pristupiti podacima u straničenoj memoriji CPU-a, potrebno je podatke prvo premjestiti u nepodijeljenu (eng. non-paged) ili prikvačenu (eng. pinned) memoriju. Prikvačena memorija omogućava DMA (eng. Direct Memory Access), čime se ubrzava prijenos podataka na GPU jer operacijski sustav ne može straničiti ovu vrstu memorije.[12]

Proces prijenosa podataka uključuje sljedeće korake:

1. *Alokacija prikvačene memorije*: CUDA driver ili aplikacija mora prvo alocirati prostor u prikvačenoj memoriji na CPU-u.
2. *Kopiranje podataka:* Podaci se kopiraju iz straničene memorije CPU-a u prikvačenu memoriju.
3. *Prijenos na GPU:* Nakon kopiranja, podaci se prebacuju s prikvačenog niza u memoriju GPU-a.

Ako se podaci izravno alociraju u prikvačenoj memoriji, smanjuje se vrijeme potrebno za prijenos podataka jer se izbjegava dodatni korak kopiranja iz straničene u prikvačenu memoriju. Ova metoda omogućava brže i efikasnije korištenje GPU resursa, što je ključno za performanse aplikacija koje intenzivno koriste grafiku i računanje.

Slika 8.: Prikaz straničene i prikvačene memorije te prijenos s CPU na GPU [12]



Prethodno ubrzanje moguće je ostvariti postavljanjem parametra pin_memory na True prilikom inicijalizacije DataLoadera, gdje se viši tenzori slika izravno postavljaju u prikvačenu memoriju. [27]

U eksperimentima koje smo pokretali, korištenje prethodnog principa je za određene veličine grupa javljalo greške te smo morali koristili manje grupe. Razrješenje je problema bilo postavljanja pin_memory na False te je veličina grupe koja stane u memoriju porasla. Detaljnije u 378.2.2

6. Arhitekture i modeli

6.1. SwiftNet

Swiftnet je arhitektura modela opisana u radu [10] dizajnirana za efikasno segmentiranje slike, koristeći se prednostima prijenosa znanja (eng. transfer learning) i efikasnog naduzorkovanja te obrade u stvarnom vremenu. Model se oslanja na tri gradivna bloka: enkoder za prepoznavanje, dekoder za uzorkovanje te modul za povećavanje receptivnog polja. SwiftNet se implementira u dvije varijante: model jednostrukog mjerila (eng. Single Space Model) i model s međusobno povezanom fuzijom piramida (eng. Interleaved Pyramid Fusion Model).

6.1.1. Enkoder za prepoznavanje

Enkoder se temelji na modelima prethodno treniranim na ImageNetu, primjerice ResNet18 te MolbileNetV2 arhitekturama. Specifično su ovi modeli korišteni zbog umjerene dubine, mogućnosti interferencije u stvarnom vremenu te rezidualne strukture.

6.1.2. Dekoder za uzorkovanje

Dekoder transformira semantički bogate vizualne značajke iz enkodera natrag u rezoluciju ulazne slike koristeći niz modula za uzorkovanje s lateralnim vezama. Modula je više te autori koriste naziv ljestvičastog stila (eng. ladder style) za opis njihova dizajna, gdje se značajke niske rezolucije prvo uzorkuju bilinearnom interpolacijom do rezolucije lateralnih značajki, a zatim se miješaju zbrajanjem elementima, nakon čega slijedi fuzija pomoću 3x3 konvolucije.

6.1.3. Modul za povećanje receptivnog polja

SwiftNet koristi prethodno opisani SPP (eng. Spatial Pyramid Pooling), odnosno prostorno piramidalno sažimanje kojime se zadržava brzina obrade u stvarnom vremenu. SPP blok prikuplja značajke enkodera na različitim razinama prostornog sažimanja. Takve reprezentacije su slike varirajućih razina detalja te je time modelu omogućeno obrađivanje različitih rezolucija.

6.1.4. Single space model

Single space model transformira ulaznu sliku u guste (semantičke) predikcije koristeći prethodno opisane dekodere i enkodere, dok su značajke na zadnjem izlazu enkodera poslane u SPP sloj zbog povećanja receptivnog polja. Zbog nesimetričnosti enkodera i dekodera koriste se lateralne veze s 1x1 konvolucijom.

Modul za naduzorkovanje slijedi u 3 koraka; 1) značajke prostorno niske razlučivosti su bilinearno naduzorkovane, 2) zatim su sumirane lateralnim vezama te 3) izmiješane s 3x3 konvolucijom.

6.1.5. Interleaved pyramid fusion model

Razlika u odnosu na Single space model jest korištenje drugog enkodera. Drugi enkoder je primijenjen na originalnoj slici niže rezolucije. Ideja je time povećati receptivno polje, specifično aktivacije nižih rezolucija piramide slike te otkloniti potrebu za modelom većeg kapaciteta zbog dijeljenja parametara između enkodera.

Vektori značajki iste prostorne razlučivosti obaju enkodera su spojeni na različitim razina poduzorkovanja te su te značajke, kao i u Single space modelu, prenijete na ulaze dekodera pomoću 1x1 konvolucija.

6.2. MagNet

MagNet [13] je segmentacijski okvir s više razina dizajniran za obradu slika velike rezolucije. Sastoji se od više razina arhitekture mreže, gdje je svaka razina zadužena za određenu razinu (eng. scale) slike.

Okvir se sastoji od segmentacijskog modula te modula uglađivanja. Ulazna slika se analizira na više razina, počevši od najgrublje pa sve do najfinije razlučivosti.

Za svaku razinu skale, ulazna slika se dijeli na dijelove odgovarajuće veličine (eng. patch), na kojima se zatim izvodi semantička segmentacija. Ove operacije su prikazane u modularnom procesu gdje se ulazne slike i rezultati segmentacije slijedno obrađuju i poboljšavaju kroz niz obradbenih faza.

MagNet koristi mrežu za segmentaciju za proizvodnju predikcija specifičnih za razinu, dok modul za uglađivanje selektivno zaglađuje grube predikcije iz prethodnih faza na temelju

lokalnih predikcija. Ovaj pristup omogućuje detaljnije i preciznije rezultate segmentacije, optimizirajući tako procesiranje slika visoke rezolucije.

6.2.1. Segmentacijski modul

Ovaj modul može biti bilo koji osnovni segmentacijski model sposoban za isporuku mape segmentacije s procjenama neizvjesnosti. Modul segmentacije na svakoj razini proizvodi specifične mape segmentacije za tu razinu.

6.2.2. Modul za uglađivanje

Koristi se za uglađivanje pojedinih dijelova segmentirane mape na različitim razinama. Kao ulaz modulu šalju se segmentacijske mape, odnosno kumulativni rezultat iz prethodnih etapa i rezultat dobiven od segmentacijskog modula specifičnog za trenutnu razinu.

7. Programski aspekti i alati

Zadatak implementacije jest usporediti performanse MagNeta te SwiftNeta na DeepGlobe land cover classification datasetu. Autori [13] navode kako MagNet ostvaruje 72.96% mIoU na skupu za testiranje te je cilj rekreirati te rezultate. Autori SwiftNeta [10] nisu učili i testirati svoju arhitekturu na tom skupu podataka te je cilj omogućiti pokretanje eksperimenata na tom datasetu te optimizirati učenje modela.

Za ostvarenje zadatka korišteni su javno dostupni repozitoriji na Githubu na kojima su objavljeni kodovi za radove [13] te [10], dostupni na javnim GitHub[1] repozitorijima [28],[29]. Na odgovarajuća mjesta ugrađena je podrška za korištenje skupa podataka korištenog u ovome radu te su dodane logike u kodu na za to predviđena mjesta, primjerice IFW opisan u poglavlju 5.2 te OHEM opisan u poglavlju 5.1. Oba repozitorija nalaze se modificirana na mome javnom GitHub repozitoriju, a ostatak koda, koji se sastoji od bilježnica i skripti u Pythonu, priloženi su ovome radu.

7.1. Korišteni alati i tehnologije

Korišten je programski jezik Python[2] v. 3.10.9. Naravno, kako govorimo o dvama različitim projektima i repozitorijima (za rad [13] repozitorij [28] te za [10] repozitorij [29]), postoje različiti zahtjevi biblioteka. Unija zahtijevanih biblioteka jesu PyTorch[3] v. 1.13.1, NumPy[4] v. 1.24.1, Torchvision[5] v. 0.14.1, Pillow[6] v. 9.4.0, OpenCV[7] v. 4.7 te scikit-learn[8] v. 1.2.0.

Za potrebe treniranja mreže korištene su grafičke kartice s instaliranim CUDA[9] driverima. Korišteni su GeForce GTX 1070 te RTX 2080 Ti.

[1] https://github.com/
[2] https://www.python.org/
[3] https://pytorch.org/
[4] https://numpy.org/
[5] https://pytorch.org/vision
[6] https://numpy.org/
[7] https://python-pillow.org/
[8] https://scikit-learn.org/
[9] https://en.wikipedia.org/wiki/CUDA

7.2. Programska izvedba

DeepGlobe land cover dataset sadrži 803 anotirane slike. Prateći radove [13],[14] skup je podijeljen na skup za učenje, testiranje i evaluaciju. 454 slike su u skupu za učenje, 207 ih je u validacijskom skupu te 142 u skupu za testiranje.

Augmentacija u treningu SwiftNeta uključuje sljedeće augmentacije: nasumično zrcaljenje te tehnika izrezivanja kvadrata na slici te promjena njegove veličine, dok augmentacija kod MagNeta uključuje rotaciju, horizontalno i vertikalno zrcaljenje.

SwiftNet je treniran s ResNet18 mrežom kao okosnicom, predtreniranom na ImageNetu, koristio se SemsegCrossEntropy gubitak te obje inačice modela (6.1.4 te 6.1.5). Koristio se optimizacijski algoritam ADAM, stopa učenja je bila 0.0004, koristio se cosine annealing za smanjenje stope učenje tokom učenja na minimum od 1e-6 . Veličina grupe je varirala zbog korištenja različitog GPU-a, problema s prikvačenom memoriju ref. 5.3 te zbog različite veličine izreza slike. Korištena je veličina izreza od 768 piksela te od 960 piksela. Trening se izvodio u 250 epoha.

U eksperimentima s MagNetom koristio sam ResNet18 te ResNet50 uz SGD. Trenirao sam modul za uglađivanje kroz 50 epoha, a stopa učenja smanjena je deset puta na epohama 20, 30, 40 i 45. Koristio sam unakrsnu entropiju kao funkciju gubitka za treniranje modula segmentacije i uglađivanja.

Sve se metrike usporedbe gledaju prilikom evaluacije modela na skupu podataka za testiranje.

8. Eksperimenti i evaluacija

8.1. Metrike uspješnosti

TP (eng. true positive) : broj ispravno identificiranih pozitivnih slučajeva.

FP (eng. false positive): broj pogrešno identificiranih pozitivnih slučajeva.

TN (eng. true negative): broj ispravno neoznačenih primjera.

FN (eng. false negative): broj stvarnih pozitivnih slučajeva koji nisu identificirani.

IoU (eng. intersection over union) je metrika koja kvantizira preciznost predikcija modela u odnosu na oznake. Formulom je zadaka kao :

$$IoU = \frac{TP}{\text{TP+FP+FN}} \tag{7}$$

Točnost (eng. accuracy) piksela je najjednostavnija metrika koja mjeri ukupni udio piksela koji su ispravno klasificirani u cijeloj slici ili u skupu podataka.

Preciznost (eng. precision) je mjera točnosti pozitivnih predikcija modela. Formula je također točno navedena:

$$Precision = \frac{TP}{\text{TP+FP}} \tag{8}$$

Odziv (eng. recall) ili osjetljivost predstavlja sposobnost modela da identificira sve relevantne instance unutar skupa podataka.

$$Recall = \frac{TP}{\text{TP+FN}} \tag{9}$$

8.2. Eksperimenti

8.2.1. MagNet

Na službenom su git repozitoriju dostupne težine naučenih modela, odnosno segmentacijskog modula te modula za uglađivanje. Izviješteno postižu mIoU od 67.22% te 72.10% te je vrlo lako u isto se uvjeriti evaluacijom modela na validacijskom skupu.

Cilj je bio izviještene rezulatate rekreirati treningom iz nule. Početni eksperimenti provedeni su na GeForce GTX 1070 grafičkoj kartici. Izvođeni su treninzi s različitim okosnicama, odnosno s ResNet18 te ResNet50.

Trening s ResNet50 kao okosnicom traje oko 7 sati za 484 epohe, a daje niže rezultate od izviještenih. Očekivano, gore je rezultate zabilježio model s okosnicom manjeg kapaciteta.

Sljedeći korak bio je dodati metode IFW ref. 5.2 te OHEM ref. 5.1 Ponovljeni su treninzi koristeći ResNet50 kao okosnicu, detaljniji je pregled u tablici.

Tablica 3.: Prikaz rezultata grube (eng. coarse) i uglađene segmentacije koristeći razne tehnike opisane u 5

Okosnica	Tehnika	Gruba segmentacija IoU (%)	Uglađena segmentacija IoU (%)
ResNet50	-	65.11	68.97
ResNet18	-	64.16	67.31
(Izviješten model) ResNet50	-	**67.22**	**72.10**
ResNet50	IFW (alfa=0.5)	64.12	67.13
ResNet50	IFW (alfa=1.0)	64.48	65.71
ResNet50	IFW (alfa=0.25)	64.43	65.54
ResNet50	OHEM	66.88	69.07
ResNet50	OHEM + IFW	NaN	NaN

Zaključak je kako okosnica postiže izvještene rezultate koristeći OHEM uz unakrsnu entropiju, no nije bilo moguće rekreirati izvještene rezultate za modul uglađivanja. Korišteni su hiperparametri za trening izvješteni u radu te je veličina grupe odgovara onoj u radu.

Korištenjem OHEM-a podigla je performanse modela, dok isti zaključak nije zapažen korištenjem IFW-a, gdje je za različite faktore alfa MagNet ostvarivao slične rezultate kao i bez te tehnike.

Korištenjem OHEM-a u kombinaciji s IFW-om dovodi do eksplozije gradijenata za različite faktore alfa.

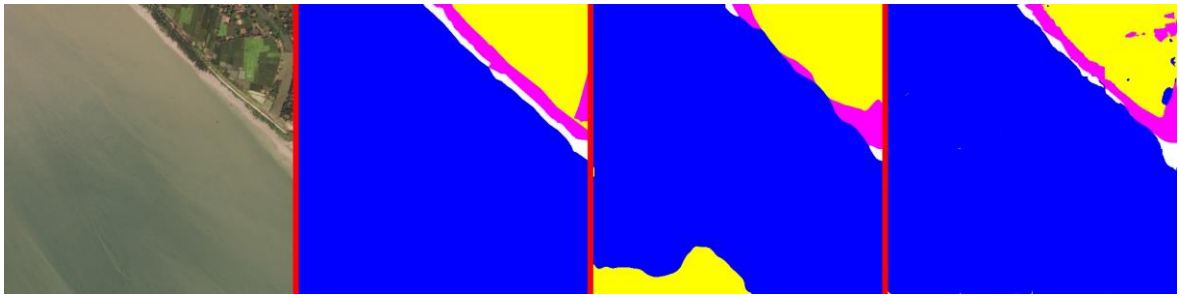
Tehnika uglađivanja predložena od autora ostvaruje viši mIoU prilikom validacije, ali ne u jednako značajnoj mjeri kao što je izvješteno u radu.

8.2.1.1 Primjeri segmentacije najboljeg modela

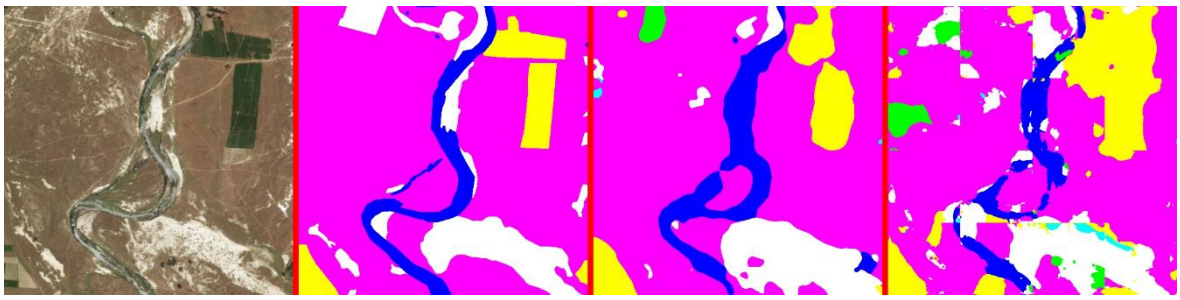
Ovdje navodim primjere segmentacijskih mapa generiranih grubljim predikcijama segmentacijskog te modula za uglađivanje, s ResNet50 kao okosnicom.

S lijeva na desno nalaze se originalna slika, semantička oznaka slike, predikcija segmentacijskog te predikcija modula za uglađivanje.

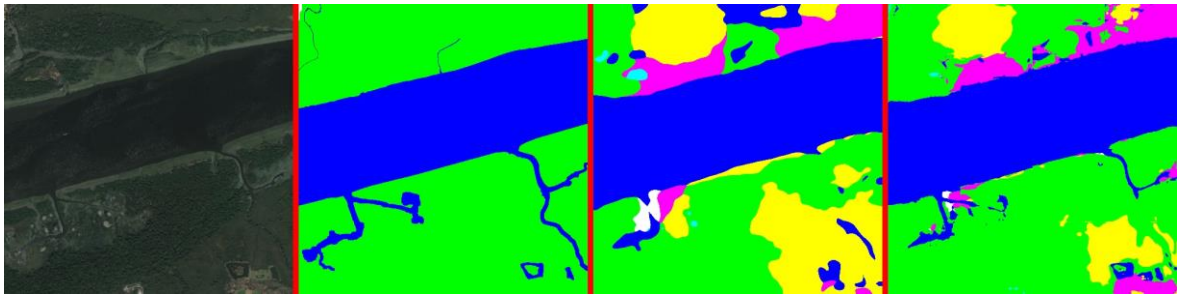
Slika 9: 7892_sat.jpg, segmentacija utreniranim modelom okosnice ResNet50 te uglađivanjem.



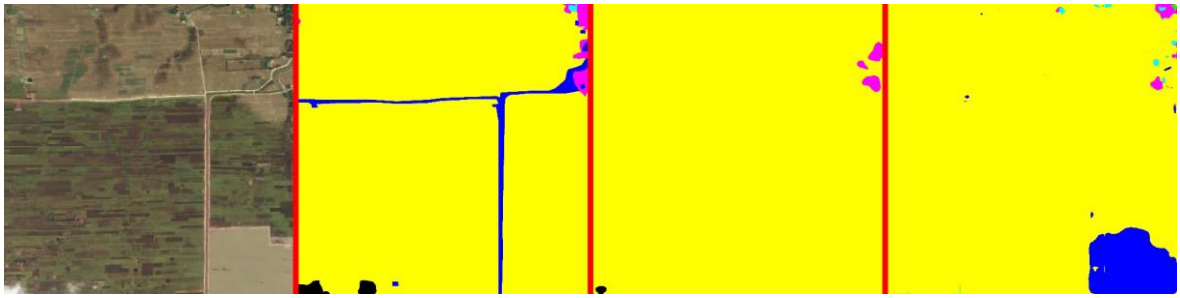
Slika 10: 28935_sat.jpg, segmentacija utreniranim modelom okosnice ResNet50 te uglađivanjem.



Slika 11: 170535_sat.jpg, segmentacija utreniranim modelom okosnice ResNet50 te uglađivanjem.



Slika 12: 987381_sat.jpg, segmentacija utreniranim modelom okosnice ResNet50 te uglađivanjem.

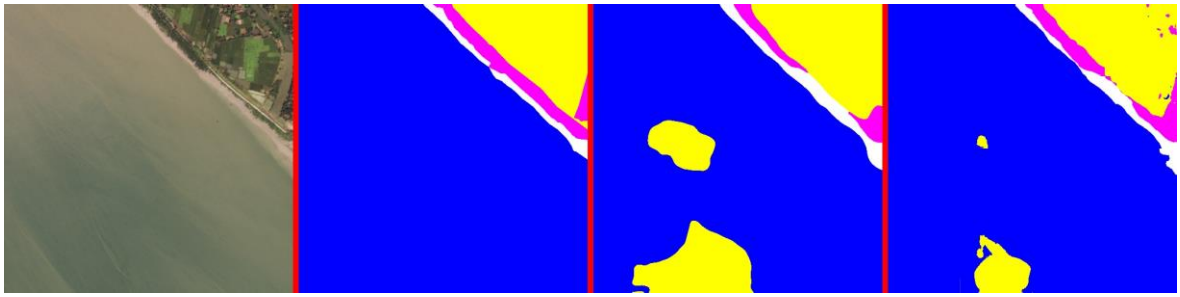


8.2.1.2 Usporedba s modelom s ResNet18 okosnicom

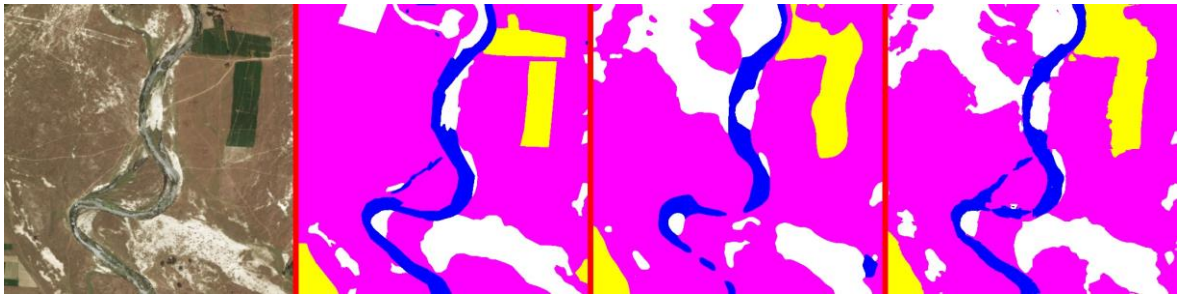
Kako bih pokazao razliku korištenja okosnica različitih kapaciteta, specifično ResNet18 te ResNet50 modela, ovdje ću priložiti nekoliko segmentacijskih mapa generiranih kao izlazima iz MagNeta s ResNet18 mrežom kao osnovnim segmentacijskim modulom.

S lijeva na desno nalaze se originalna slika, semantička oznaka slike, predikcija segmentacijskog te modula za uglađivanje.

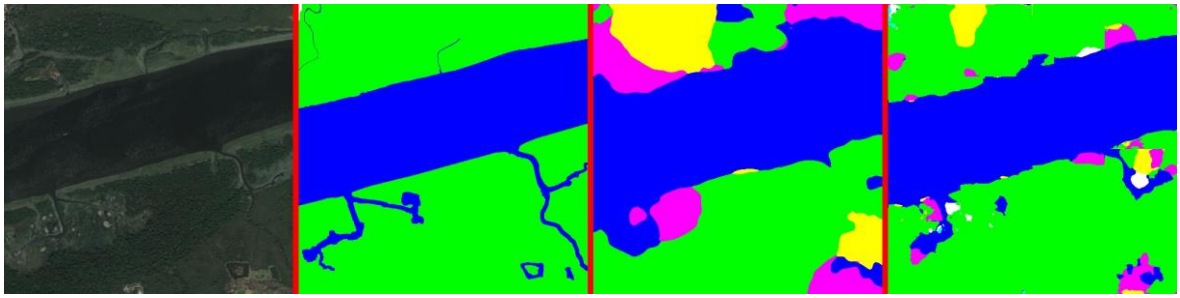
Slika 13: 7892_sat.jpg, segmentacija utreniranim modelom okosnice ResNet18 te uglađivanjem.



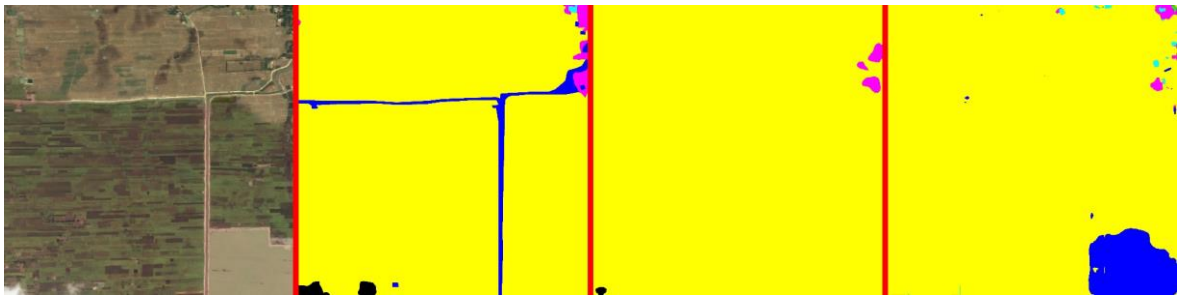
Slika 14: 28935_sat.jpg, segmentacija utreniranim modelom okosnice ResNet18 te uglađivanjem.



Slika 15: 170535_sat.jpg, segmentacija utreniranim modelom okosnice ResNet18 te uglađivanjem.



Slika 16: 987381_sat.jpg, segmentacija utreniranim modelom okosnice ResNet18 te uglađivanjem.



8.2.2. SwiftNet

Prvotni trening obavljen je na GeForce GTX 1070 grafičkoj kartici, s izrezom slike od 768x768 piksela. Model je treniran u 250 epoha, s evaluacijom svake 4 epohe. Ukupno je trajanje treninga 4 sata i 45 minuta, fps (eng. frames per second) je u rangu 12.14. Ovaj trening je obilježila loša konvergencija modela na evaluacijskom skupu. Isto je posljedica male veličine grupe koja je stala u memoriju GPU-a, naime ista je bila tek 2. Sljedeći je cilj bio prenaučiti model na manjem skupu podataka kako bi se istražilo ima li model dostatan broj informacija za diskriminaciju između piksela na ulazima koje prima. Niti taj zadatak nije bio uspješan zbog malog broja slika koje su mogle stati u memoriju.

Jedno je rješenje bio i prelazak s GeForce GTX 1070 na RTX 2080 Ti. No, isto je tako rješenje oslobađajuće po memoriju bilo i postavljanje parametra pin_memory u pozivu torch.utils.data.DataLoader na false. U službenoj dokumentaciji [15], parametar funkcije je opisan ovako

> *"pin_memory (bool, optional):*
>
> *If ``True``, the data loader will copy Tensors into device/CUDA pinned memory before returning them. If your data elements are a custom type, or your :attr:`collate_fn` returns a batch that is a custom type, see the example below.*

U poglavlju 5.3 opisao sam prijenos podataka na GPU, a problemi [17] ,[18] ,[19] opravdali su postavljanje parametra na False.

Posljedica je bilo povećanje maksimalne veličine grupe na 6 te su zabilježeni najbolji rezultati treninga s mIoU od 73%.

Posljednja je etapa treniranja bila na RTX 2080 Ti, gdje je maksimalna veličina grupe, uz prethodno postavljeno na False, bila 56. Pokrenuto je nekoliko treninga s različitim veličinama grupa te različitim veličinama izreza slika, a sve rezultate prilažem u tablici 4.

Tablica 4.: Sažeti prikaz testiranja single scale SwiftNeta uz različite parametre te GPU uređaje.

GPU	PIN MEM.	VEL. GRUPE	VEL. ISJEČKA	mIoU /%	mRecall /%	mPrecision /%	mAccuracy /%
GeForce GTX 1070	True	2	768x768	64.66	78.68	75.86	84.35
GeForce GTX 1070	False	6	768x768	**73.00**	85.21	**82.49**	88.02
RTX 2080 Ti	False	14	768x768	69.55	82.67	79.49	86.60
RTX 2080 Ti	False	50	768x768	72.54	**86.26**	81.15	**88.20**
RTX 2080 Ti	False	16	768x768	71.09	83.01	81.13	86.90
RTX 2080 Ti	False	16	960x960	72.24	83.45	82.24	87.68

Za svaki model prikazujem i IoU za svaku klasu u skupu podataka DeepGlobe land cover. Redak u tablici ispod odgovara retku u gornjoj tablici.

Tablica 5.: Sažeti prikaz pojedinih klasnih IoU u testiranju SwiftNet modela:

Urban Land IoU /%	Agriculture Land IoU /%	Rangeland IoU /%	Forest Land IoU /%	Water IoU /%	Barren Land IoU /%
78.11	83.87	28.58	70.80	73.33	53.26
78.10	86.94	**48.21**	75.62	82.52	**66.61**
76.31	86.23	35.19	79.04	79.73	60.79
75.72	87.05	44.50	75.88	77.85	74.21
77.70	86.53	40.81	78.20	82.20	61.13
78.29	**87.39**	40.55	**81.37**	**83.29**	62.55

Zatim su provedeni eksperimenti na RTX 2080 Ti s veličinom grupe od 16, s veličinama isječaka od 768 te 960 piksela, koristeći 6.1.5 piramidalnu inačicu modela. Rezultati su slični 6.1.4. inačici SwiftNeta.

Tablica 6.: Sažeti prikaze evaluacija piramidalne inačice SwiftNeta uz različite veličine isječka.

GPU	PIN MEM.	VEL. GRUPE	VEL. ISJEČKA	mIoU /%	mRecall /%	mPrecision /%	mAccuracy /%
RTX 2080 Ti	False	6	768x768	70.47	80.83	82.37	86.71
RTX 2080 Ti	False	16	768x768	71.08	81.31	82.65	87.11
RTX 2080 Ti	False	16	960x960	71.68	82.37	82.37	87.39

Tablica 5.: Sažeti prikaz pojedinih klasnih IoU u evaluaciji 6.1.5 inačice SwiftNeta

Urban Land IoU /%	Agriculture Land IoU /%	Rangeland IoU /%	Forest Land IoU /%	Water IoU /%	Barren Land IoU /%
78.43	86.80	38.14	78.75	81.31	59.36
79.33	87.21	37.81	78.52	82.31	61.32
79.22	87.25	39.38	79.76	83.29	61.18

8.2.3. Pregled najboljeg modela

8.2.3.1 Službeni poredak na izazovu DeepGlobe land cover classification

Najbolje rezultate na validacijskom skupu odnio je single scale SwiftNet s veličinom isječka od 768x768, s veličinom grupe od 6 , treniran 250 epoha na GeForce GTX 1070.

Ljestvica [24] nudi poredak najboljih modela evaluiranih na testnom skupu podataka. Komponirajući rezultate ovog rada, tablica izgleda ovako.

Tablica 6.: Prikaz stanja tehnike na skupu DeepGlobe land cover classification

Model	mIoU/%	Rad
WSDNet	74.10	[20]
FCtL	73.22	[21]
SwifNet (naš)	73.00	[10]
MagNet *	72.96	[13]
GLNet	71.60	[22]
MagNet (naš)**	69.07	[13]
CascadePSP	68.5	[23]

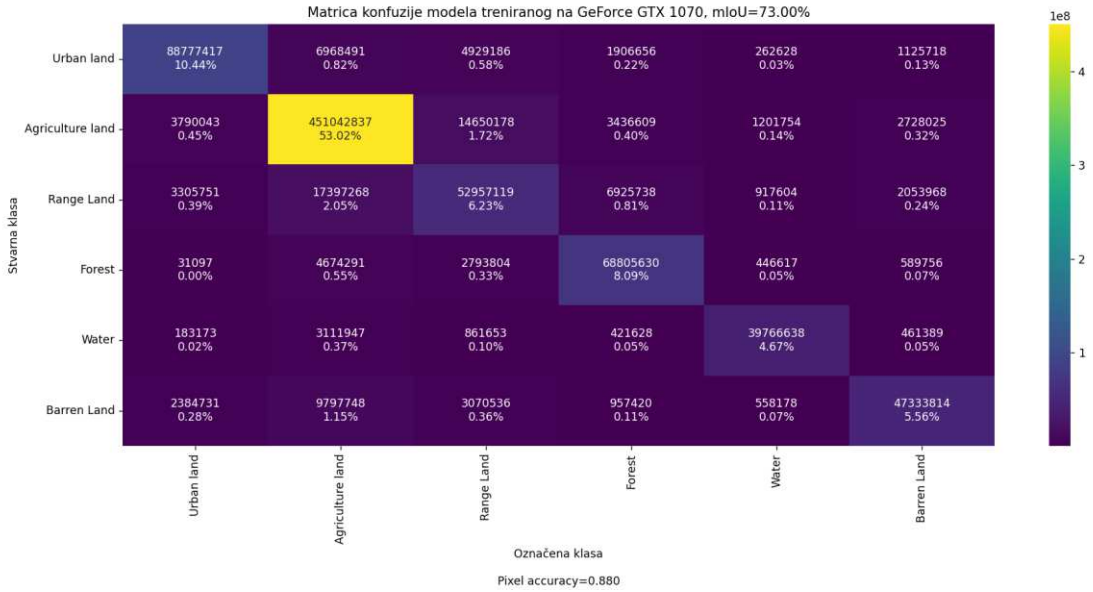
*službena verzija modela

**model utreniran u ovome radu

8.2.3.2 Matrica konfuzije

Za prethodno opisani model, ovdje prilažem matricu konfuzije dotičnog modela na skupu za testiranje.

Slika 11. Matrica konfuzije SwiftNeta s mIoU=73.00%

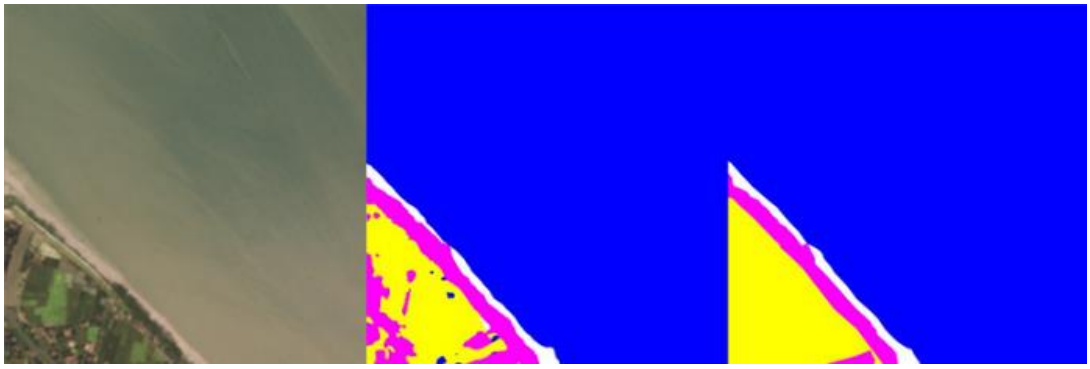


Zanimljivo, klasa rangeland svim je modelima predstavljala problem za segmentacijom, iako je više zastupljena od klasa water, barren land. Pretpostavka bi bila semantička sličnost i/ili prostorna bliskost s kategorijama koje su znatno zastupljenije, primjerice agriculture land, čime model češće odlučuje klasificirati piksele na razdiobi dviju klasa kao piksele češće klase.

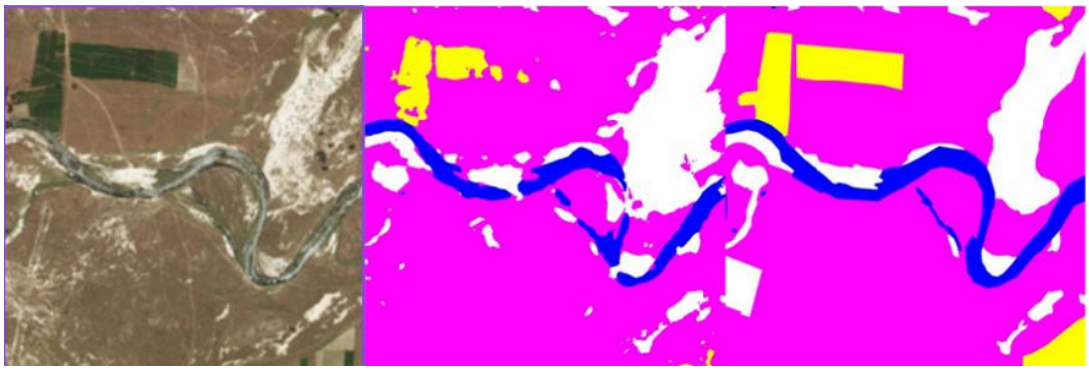
8.2.3.3 Primjeri segmentacije

S lijeva na desno nalaze se originalna slika, predikcija modela te semantička oznaka slike.

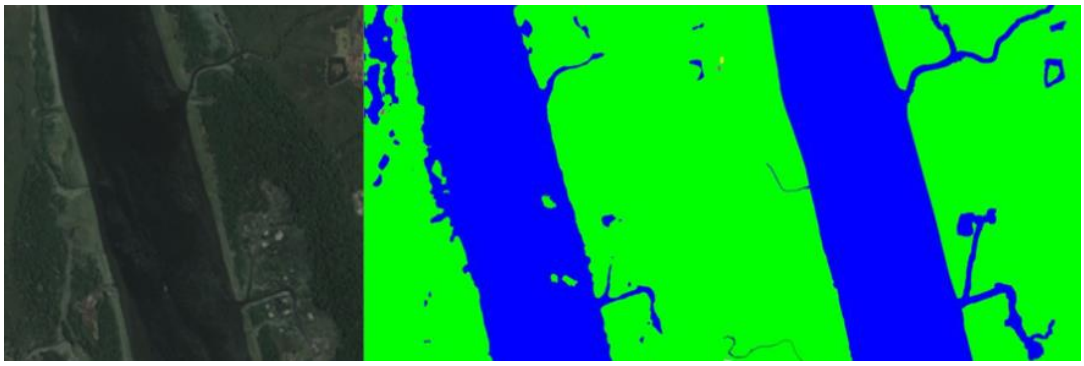
Slika 17.: Segmentacija slike 7892_sat.jpg Single Space SwiftNeta s ResNet18 okosnicom



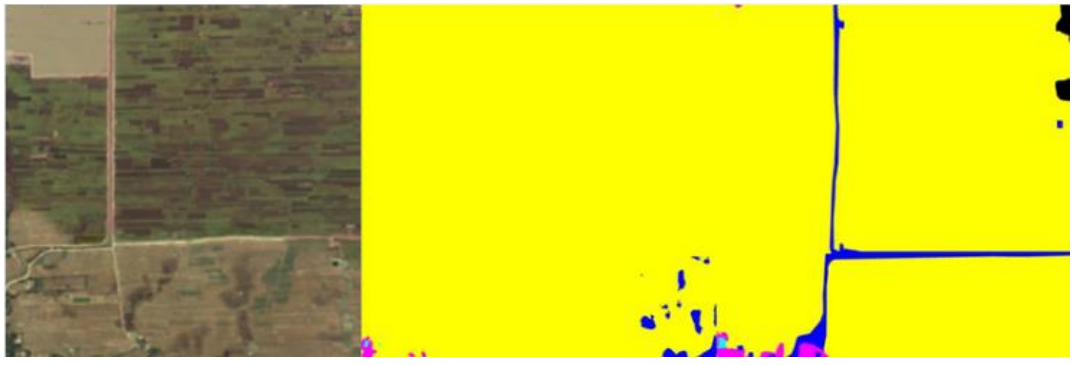
Slika 18.: Segmentacija slike 28935_sat.jpg Single Space SwiftNeta s ResNet18 okosnicom



Slika 19.: Segmentacija slike 170535_sat.jpg Single Space SwiftNeta s ResNet18 okosnicom



Slika 20.: Segmentacija slike 987381_sat.jpg Single Space SwiftNeta s ResNet18 okosnicom



Zaključak

Tema ovog reda je bila semantička segmentacija slika. Dan je detaljan pregled područja dubokog učenja, računalnog vida, segmentacije slike, dubokih neuronskih mreža te su pobliže opisani algoritmi propagacije greške unatrag.

Priložen je detaljan pregled dvaju segmentacijskih okvira, MagNeta te SwiftNeta. Opisani su najvažniji pojmovi vezano uz oba te su temeljito opisani postupci učenja i vrednovanja modula na DeepGlobe land cover classification datasetu.

Upogonjena je postojeća te dodatno razvijena specifična programska podrška za semantičku segmentaciju obaju arhitektura na DeepGlobe land cover classification datasetu.

Istraženi su koncepti poput Online hard example mininga, Inverse frequency weightinga, prijenosa podataka između CPU te GPU uređaja te problemi povezani s tim prijenosom te je ostvaren dobar osjećaj o segmentaciji slike koristeći napredne tehnike segmentacije.

Na spomenutom skupu podataka ostvaren je jako dobar rezultat u odnosu na stanje tehnike prilikom testiranja modela. Slijedi i kako je SwiftNet brži u učenju i ima niže vrijeme inferencije u odnosu na MagNet. Dapače, s rekreiranjem rezultata navedenih u [13] bilo je dosta poteškoća, da bi posljedično uspjeli rekreirati samo rezultate osnovnog segmentacijskog modula, dok se pokazalo kako je korištenje modula za uglađivanje složen proces s neizvjesnim ishodom.

SwiftNet ostvaruje mIoU od 73.00% na testnom skupu koristeći pritom Resnet18 kao okosnicu, dok je izvješteni uglađeni model MagNeta ostvario mIoU od 72.96%. Doduše, najbolji ostvareni rezultat prilikom našeg rekreiranja rezultata je bio značajno niži te je mIoU iznosio 69.07%. Model stanja tehnike WSDNet [20] (na ovom skupu podataka) ostvaruje mIoU od 74.10%.

Rezultat od 73.00% mIoU na testnom skupu dostatno je za treće mjesto službene ljestvice [24] na DeepGlobe land cover classification izazovu.

Uspoređujući arhitekture SwiftNeta s onom od MagNeta, ostvarivanje konvergencije je znatno lakše na SwiftNetu te je posljedično i ostvaren bolji rezultat i od inačice MagNeta utrenirane iz nule, ali i od izvještenog rezultata na testnom skupu.

Literatura

[1] Dalbelo Bašić,B.,Šnajder,J. *Uvod u strojno učenje*. Poveznica:

https://www.fer.unizg.hr/_download/repository/SU-2016-01-Uvod.pdf,pristupljeno:7.lipanj 2024.

[2] Ioffe, S., Szegedy, C. *Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift,* arXiv preprint arXiv:1502.03167v3 [cs.LG]. Dostupno na: https://doi.org/10.48550/arXiv.1502.03167.

[3] Bošnjak, D. *Varijante gradijentnog spusta u strojnom učenju s primjerima*. Diplomski rad. Sveučilište u Zagrebu, Prirodoslovno-matematički fakultet,2023

[4] Grubišić,I. *Semantička segmentacija slika dubokim konvolucijskim mrežama*. Završni rad. Sveučilište u Zagrebu Fakultet elektrotehnike i računarstva, 2016.

[5] *Backpropagation*. Brilliant.org. Poveznica: https://brilliant.org/wiki/backpropagation/, pristupljeno 8.lipnja 2024.

[6] He, K., Zhang, X., Ren, S., Sun, J. *Deep Residual Learning for Image Recognition*, arXiv preprint arXiv:1512.03385, dostupno na: https://doi.org/10.48550/arXiv.1512.03385, (2015, prosinac).

[7] Huang, G., Liu, Z., van der Maaten, L., Weinberger, K. Q. *Densely Connected Convolutional Networks*, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017. arXiv:1608.06993v5 [cs.CV]. Dostupno na:

https://doi.org/10.48550/arXiv.1608.06993.

[8] Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.-C. *MobileNetV2: Inverted Residuals and Linear Bottlenecks*, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), str. 4510-4520, 2018. Dostupno na:

https://doi.org/10.48550/arXiv.1801.04381. arXiv preprint arXiv:1801.04381v4 [cs.CV].

[9] Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N. *An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale*, u: Proceedings of the International Conference on Learning Representations (ICLR), 2021. Dostupno na:https://arxiv.org/pdf/2010.11929

[10] Oršić, M., Krešo, I., Bevandić, P., Šegvić, S. *In Defense of Pre-trained ImageNet Architectures for Real-time Semantic Segmentation of Road-driving Images*, Proceedings of the

IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2019. arXiv:1903.08469v2 [cs.CV]. Dostupno na: https://doi.org/10.48550/arXiv.1903.08469.

[11] *Stranična datoteka*. Poveznica: https://hr.wikipedia.org/wiki/Strani%C4%8Dna_datoteka, pristupljeno 8.lipnja 2024.

[12] Mark Harris, *How to optimize dana transfer sin CUDA C/C+*. Poveznica: https://developer.nvidia.com/blog/how-optimize-data-transfers-cuda-cc/, pristupljeno 8.lipnja 2024.

[13] Huynh, C., Tran, A., Luu, K., Hoai, M. *Progressive Semantic Segmentation* Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition CVPR), 2021. arXiv:2104.03778 [cs.CV]. Dostupno na: https://doi.org/10.48550/arXiv.2104.03778

[14] Chen, W., Jiang, Z., Wang, Z., Cui, K., Qian, X. (2019). *Collaborative Global-Local Networks Memory-Efficient Segmentation of Ultra-High Resolution Images*. U: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). arXiv:1905.06368 [cs.CV]. Poveznica: https://doi.org/10.48550/arXiv.1905.06368

[15] *Source code for torch.utils.data.dataloader*. Poveznica : https://mmcv.readthedocs.io/en/master/_modules/torch/utils/data/dataloader.html, pristupljeno 7.lipnja 2024.

[16] Petrač, T. *Polunadzirana semantička segmentacija utemeljena na pseudooznačavanju*. Diplomski rad. Sveučilište u Zagrebu Fakultet elektrotehnike i računarstva, 2021.

[17] StackOverflow. *Pytorch*. *How does pin_memory work in DataLoader*. Poveznica: https://stackoverflow.com/questions/55563376/pytorch-how-does-pin-memory-work-in-dataloader, pristupljeno 7. lipnja 2024.

[18] Zhoung,K. *When to set pin_memory to true?* Poveznica: https://discuss.pytorch.org/t/when-to-set-pin-memory-to-true/19723/22, pristupljeno 7. lipnja 2024.

[19*]* *RuntimeError: Pin memory thread exited unexpectedly*. Poveznica: RuntimeError: Pin memory thread exited unexpectedly · Issue #4 · Runinho/pytorch-cutpaste · GitHub, pristupljeno 7. lipnja 2024.

[20] Ji, D., Zhao, F., Lu, H., Tao, M., Ye, J. *With the increasing interest and rapid development of methods for Ultra-High Resolution (UHR) segmentation*, Proceedings of the IEEE Conference on

Computer Vision and Pattern Recognition (CVPR), 2023. arXiv:2305.10899 [cs.CV]. Dostupno na: https://doi.org/10.48550/arXiv.2305.10899.

[21] Liu, W., Li, Q., Lin, X., Yang, W., He, S., Yu, Y. *Ultra-high Resolution Image Segmentation via Locality-aware Context Fusion and Alternating Local Enhancement*, arXiv preprint arXiv:2109.02580 [cs.CV]. Dostupno na: https://doi.org/10.48550/arXiv.2109.02580.

[22] Chen, W., Jiang, Z., Wang, Z., Cui, K., Qian, X. *Collaborative Global-Local Networks for Memory-Efficient Segmentation of Ultra-High Resolution Images*, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2019. arXiv:1905.06368v3 [cs.CV]. Dostupno na: https://doi.org/10.48550/arXiv.1905.06368.

[23] Cheng, H. K., Chung, J., Tai, Y.-W., Tang, C.-K. *CascadePSP: Toward Class-Agnostic and Very High-Resolution Segmentation via Global and Local Refinement*, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2020. arXiv:2005.02551 [cs.CV]. Dostupno na: https://doi.org/10.48550/arXiv.2005.02551.

[24] *Land Cover Classification on DeepGlobe*. Poveznica : https://paperswithcode.com/sota/land-cover-classification-on-deepglobe, pristupljeno 7. lipnja 2024.

[25]McCullum,N. *Deep Learning Neural Networks Explained in Plain English*. Poveznica: https://www.freecodecamp.org/news/deep-learning-neural-networks-explained-in-plain-english/, pristupljeno 7.6.2024.

[26] Mohamed et al., 2015. Poveznica: https://www.researchgate.net/figure/A-hypothetical-example-of-Multilayer-Perceptron-Network_fig4_303875065, pristupljeno 7. lipnja 2024.

[27] *Memory Pinning*. Poveznica: https://pytorch.org/docs/stable/data.html#memory-pinning, pristupljeno 7. lipnja 2024.

[28]Huynh,C. Tran,A. Luu,K. Hoai,M. *MagNet.* Poveznica:

https://github.com/VinAIResearch/MagNet, pristupljeno 7. lipnja 2024.

[29] *Oršić,M.* Swiftnet. Poveznica: https://github.com/orsic/swiftnet, pristupljeno 7. lipnja 2024.

Sažetak

Semantička segmentacija velikih satelitskih slika

U ovome je radu dan pregled arhitektura i performansi dubokih modela koji koriste poduzorkovanu reprezentaciju uzorka za učenje u zadacima guste predikcije, odnosno na zadacima semantičke segmentacije slike. Opisani su provedeni eksperimenti te su sažeto prikazana vrednovanja različitih segmentacijskih arhitektura.

Ključne riječi: računalni vid, duboko učenje, semantička segmentacija, poduzorkovana reprezentacija.

Summary

Semantic Segmentation of Large Satellite Images

This paper provides an overview of the architectures and performances of deep models that use a subsampled image representation for learning in dense prediction tasks, specifically in image semantic segmentation tasks. The experiments conducted are described, and evaluations of various segmentation architectures are presented.

Keywords: computer vision, deep learning, semantic segmentation, subsampled image representation.