Zahvaljujem prof. dr. sc. Siniši Šegviću na pomoći i prenesenom znanju te mag. ing. Josipu Šariću i Dariu Oreču na savjetima pri izradi rada. Također zahvaljujem svojoj obitelji na podršci tijekom cijelog školovanja.

SADRŽAJ

1. Uvod

Računalni vid područje je umjetne inteligencije koje se bavi izvlačenjem i obradom korisnih informacija iz vizualnih ulaza kao što su slike i video zapisi. U početcima računalnog vida najviše se pažnje pridavalo prebrojivim objektima (engl. *things*). Kasnije je primijećena važnost raspoznavanja i neprebrojivih razreda (engl. *stuff*) koji predstavljaju površine slične teksture ili materijala. Na primjerima slika iz prometa primjeri prebrojivih razreda su automobili, bicikli, pješaci i slično, a primjeri neprebrojivih razreda su cesta, pločnik, nebo, travnate površine i slično.

Raspoznavanjem neprebrojivih razreda bavi se semantička segmentacija (engl. *semantic segmentation*) u kojoj svakom pikselu slike pridružujemo oznaku razreda kojem pripada. Segmentacija primjeraka (engl. *instance segmentation*) nastoji raspoznati prebrojive razrede te svakom prebrojivom objektu pridružiti okvir ili segmentacijsku masku. U proteklom desetljeću, semantička segmentacija i segmentacija primjeraka doživjele su značajan napredak, ali su se razvijale nezavisno jedna od druge te se nametnulo pitanje mogu li se prebrojivi i neprebrojivi razredi promatrati zajedno. Zadnjih nekoliko godina, razvija se područje panoptičke segmentacije (engl. *panoptic segmentation*) koja pokušava dati odgovor na ovo pitanje. Panoptička segmentacija ujedinjuje semantičku segmentaciju i segmentaciju primjeraka te nastoji svakom pikselu pridijeliti oznaku razreda i indeks primjerka. Zbog potrebe za evaluacijom rezultata panoptičke segmentacije došlo je do razvoja nove mjere kvalitete nazvane panoptička kvaliteta [13] (engl. *panoptic qualtily*, PQ) jer nijedna od postojećih mjera nije bila dovoljna za evaluaciju segmentacije prebrojivih i neprebrojivih razreda.

Cilj ovog rada je opisati pristupe za panoptičku segmentaciju, objasniti i vrednovati modele Panoptic SwiftNet [24], Panoptic DeepLab [2] i MaskFormer [4] te usporediti njihove rezultate. Ovaj rad prikazuje rezultate treniranja i evaluacije modela Panoptic SwiftNet na skupu podataka Cityscapes u sniženoj rezoluciji.

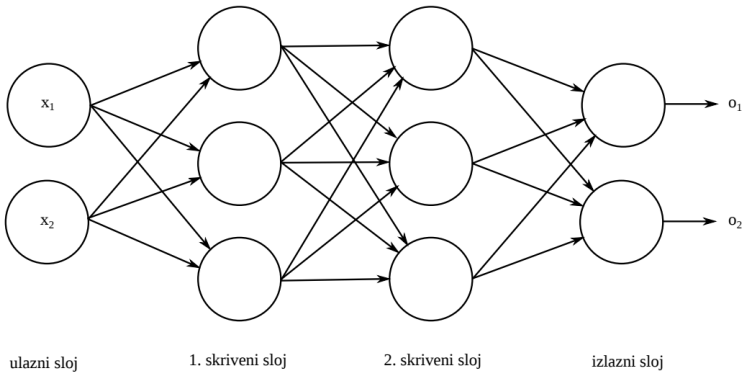
2. Umjetne neuronske mreže

Umjetne neuronske mreže algoritam su strojnog učenja nadahnut biološkim neuronima, koji ima široku primjenu u klasifikaciji i funkcijskoj regresiji. Građene su od umjetnih neurona, jedinica koje na ulaz primaju vektor značajki na temelju kojih računaju izlaznu vrijednost. Neuron za svaku vrijednost ulaznog vektora ima odgovarajuću težinu te računa izlaznu vrijednost kao sumu umnožaka ulaznih vrijednosti i odgovarajućih težina kojoj dodaje vrijednost pomaka (engl. *bias*). Na dobivenu sumu primjenjuje se aktivacijska funkcija te se dobivena vrijednost prosljeđuje na ulaz neurona sljedećeg sloja.

Kao i svaki algoritam strojnog učenja, umjetne neuronske mreže definirane su modelom, gubitkom i metodom optimizacije .

2.1. Arhitektura neuronske mreže

Neuronske mreže građene su od više povezanih slojeva umjetnih neurona. Sadrže najmanje dva sloja, ulazni i izlazni sloj, te skrivene slojeve između njih. Ako mreža sadrži skrivene slojeve naziva se duboka neuronska mreža. U potpuno povezanoj mreži, izlaz svakog neurona jednog sloja povezan je s ulazom svakog neurona u sljedećem sloju. Primjer potpuno povezane mreže prikazan je na slici 2.1.



Slika 2.1: Potpuno povezana mreža s 4 sloja. Preuzeto iz [22].

2.1.1. Aktivacijske funkcije

Aktivacijske funkcije omogućavaju stvaranje nelinearnih odnosa i utječu na točnost predikcije modela. Bez nelinearnih odnosa cijela bi se mreža mogla svesti na jedan sloj ili jedan neuron.

Najjednostavnija aktivacijska funkcija je funkcija skoka (engl. *step*) definirana sljedećim izrazom:

$$step(net) = \begin{cases} 0, & \text{net} < 0 \\ 1, & \text{net} \geq 1 \end{cases} \tag{2.1}$$

Ponekad se koristi i druga definicija koja umjesto izlaza 0 ili 1 daje izlaz -1 ili 1:

$$step(net) = \begin{cases} -1, & \text{net} < 0 \\ 1, & \text{net} \geq 1 \end{cases} \tag{2.2}$$

Ovu funkciju nije moguće koristiti ako se postupak učenja temelji na derivacijama jer je derivacija funkcije jednaka 0 na cijeloj domeni, a u točki $net = 0$ je nedefinirana.

Sigmoidalna funkcija poprima vrijednosti od 0 do 1 kao i funkcija skoka definirana s 2.1, ali se njena vrijednost postupno mijenja te je derivabilna. Ovo svojstvo omogućava učenje modela postupcima temeljenim na gradijentnom spustu [22]. Definirana je izrazom:

$$\sigma(net) = \frac{1}{1 + e^{-net}} \tag{2.3}$$

Njena je derivacija dana izrazom:

$$\frac{d\,\sigma(net)}{d\,net} = \sigma(net) \cdot (1 - \sigma(net)) \tag{2.4}$$

Funkcija tangens hiperbolni poopćuje funkciju skoka definiranu izrazom 2.2. Njena vrijednost postupno se mijenja od -1 do 1 te je derivabilna kao i sigmoidalna funkcija. Ovu funkciju moguće je izvesti pomoću sigmoidalne funkcije izrazom:

$$tanh(net) = 2 \cdot \sigma(2 \cdot net) - 1 \tag{2.5}$$

Derivaciju funkcije tangens hiperbolni možemo izraziti preko nje same:

$$\frac{d\,tanh(net)}{d\,net} = 1 - tanh^2(net) \tag{2.6}$$

U dubokim neuronskim mrežama najčešće se koristi zglobnica (engl. *Rectified Linear Unit*, ReLU). Ova funkcija pozitivne vrijednosti propušta nepromijenjene, a negativne vrijednosti preslikava u 0. Definirana je izrazom:

$$ReLU(net) = max(0, net) \tag{2.7}$$

Derivacija funkcije dana je izrazom:

$$\frac{d\,ReLU(net)}{d\,net} = \begin{cases} 1, & \text{net} > 0 \\ 0, & \text{net} \leq 0 \end{cases} \tag{2.8}$$

Ova funkcija u neaktivnom stanju ne propušta signal unaprijed niti gradijent unazad pa je zato uvedeno poopćenje, propusna zglobnica (engl. *Leaky ReLU*, LReLU), definirana izrazom:

$$LReLU = \begin{cases} net, & \text{net} > 0 \\ \alpha \cdot net, & \text{net} \leq 0 \end{cases} \tag{2.9}$$

Ako je parametar α zadan kao mali pozitivan broj, funkcija vrijednosti veće od 0 propušta nepromijenjene, dok one manje 0 prigušuje. Njena je derivacija:

$$\frac{d\,LReLU(net)}{d\,net} = \begin{cases} 1, & \text{net} > 0 \\ \alpha, & \text{net} \leq 0 \end{cases} \tag{2.10}$$

2.2. Postupak učenja

Učenje neuronske mreže postupak je prilagodbe težina i pomaka neurona u mreži kako bi mreža što bolje aproksimirala izlaznu funkciju. Kod nadziranog učenja, na ulaz mreže dovode se parovi ulaza i željenog izlaza iz skupa za učenje. Cilj učenja je postići što veću sposobnost generalizacije, što znači da mreža daje što bolje rezultate na do tada neviđenim podacima. Ako se mreža, u nekom trenutku učenja, počne previše prilagođavati skupu za učenje i gubiti sposobnost generalizacije, dolazi do prenaučenosti. Kako bismo spriječili prenaučenost mreže, skup podataka dijelimo na skup za učenje, skup za provjeru i skup za testiranje. Tijekom učenja mreže na skupu za učenje, povremeno kontroliramo mrežu podacima iz skupa za provjeru te pritom ne mijenjamo težine i pomake. Kada gubitak na skupu za provjeru počne rasti, znači da se mreža počela prilagođavati šumu u podacima za učenje i gubi sposobnost generalizacije pa učenje treba prekinuti da se spriječi prenaučenost. Skup za testiranje služi za vrednovanje mreže nakon učenja.

2.2.1. Funkcija gubitka

Funkcija gubitka važan je dio učenja neuronskih mreža koji odražava razliku između predikcije modela i željenog izlaza. Cilj učenja je minimizirati funkciju gubitka kako bi dobili što veću učinkovitost modela.

Najjednostavnija funkcija gubitka je 0-1 gubitak definiran izrazom:

$$\mathcal{L}_{01} = \begin{cases} 0, & y = \hat{y} \\ 1, & y \neq \hat{y} \end{cases} \tag{2.11}$$

Ova funkcija nije derivabilna što otežava minimizaciju.

Za zadatke regresije često se koriste se funkcije srednje apsolutne pogreške (engl. *Mean Absolute Error*) $\mathcal{L}_1$ ili $\mathcal{L}_{MAE}$ i srednje kvadratne pogreške (engl. *Mean Squared Error*) $\mathcal{L}_2$ ili $\mathcal{L}_{MSE}$ definirane izrazima:

$$\mathcal{L}_{MAE} = \frac{1}{N} \sum_{i=0}^{N} |y_i - \hat{y}_i| \tag{2.12}$$

$$\mathcal{L}_{MSE} = \frac{1}{N} \sum_{i=0}^{N} (y_i - \hat{y}_i)^2 \tag{2.13}$$

Funkcija unakrsne entropije (engl. *cross-entropy loss*) $\mathcal{L}_{CE}$ koristi se za probleme klasifikacije i definirana je izrazom:

$$\mathcal{L}_{CE} = -\sum_{i=0}^{N} y_i \cdot log(\hat{y}_i) \tag{2.14}$$

2.2.2. Optimizacijski algoritmi

Optimizacijski algoritmi su postupci minimizacije funkcije pogreške u strojnom učenju. Neki od korištenih algoritama su gradijentni spust, stohastički gradijentni spust i ADAM.

Algoritam gradijentnog spusta računa parcijalne derivacije funkcije gubitka te određuje gradijent funkcije. Gradijent funkcije gubitka dan je izrazom:

$$\nabla \mathcal{L} = \left[\frac{\partial \mathcal{L}}{\partial w_1}, ..., \frac{\partial \mathcal{L}}{\partial w_N} \right]^T \tag{2.15}$$

Promjenu vrijednosti gubitka aproksimiramo umnoškom vektora pomaka i gradijenta funkcije:

$$\Delta \mathcal{L} \approx \Delta w \cdot \nabla \mathcal{L} \tag{2.16}$$

Ako je vektor pomaka $\Delta w = -\eta \nabla \mathcal{L}$, onda vrijedi:

$$\nabla \mathcal{L} \approx -\eta ||\nabla \mathcal{L}||^2 \tag{2.17}$$

S obzirom da gradijent pokazuje u smjeru najbržeg rasta funkcije, a cilj je pronalazak minimuma, želimo se kretati u smjeru suprotnom od gradijenta te zato uzimamo negativnu vrijednost. Stopa učenja η određuje veličinu koraka koji radimo kod podešavanja težina i pomaka u mreži. Stopa učenja je mali pozitivan broj.

Težine u mreži mijenjamo prema formuli:

$$w' = w - \eta \nabla \mathcal{L} \tag{2.18}$$

Ako u navedene izraze umjesto težina uvrstimo pomake b (engl. *bias*), dobivamo izraze za prilagodbu pomaka.

Stohastički gradijentni spust (engl. *Stohastic Gradient Descent*, SGD) vrsta je gradijentnog spusta kod kojeg se podešavanje vrijednosti težina i pomaka događa nakon obrade jednog podatka ili minigrupe. Učinkovit je na velikim skupovima podataka, relativno brz i može pomoći u bijegu iz lokalnog minimuma i traženju globalnog minimuma.

Metoda učenja sa zaletom postupak je koji nastoji ublažiti oscilacije gradijenata u stohastičkom gradijentnom spustu te postići bržu konvergenciju. Definira se vektor koji predstavlja eksponencijalno umanjujući prosjek prethodnih gradijenata [23], a definiran je izrazom:

$$v' = \alpha \cdot v - \eta \cdot \nabla \mathcal{L} \tag{2.19}$$

Parametar α određuje utjecaj prethodnih gradijenata. Parametri se ažuriraju korištenjem izračunatog prosjeka:

$$w' = w + v \tag{2.20}$$

Još jedan često korišten optimizator je ADAM [11] (engl. *Adaptive Moment Estimation*). Ova metoda izračunava adaptivne stope učenja za različite parametre prema prosječnom kretanju gradijenata i kvadrata gradijenata. Osnovna inačica ADAM-a omogućava bržu konvergenciju od stohastičkog gradijentnog spusta, ali ponekad daje lošiju generalizaciju. Postoje različite modifikacije ovog optimizatora, primjerice ADAMAX, NADAM i AMSGrad [23].

2.2.3. Algoritam propagacije pogreške unazad

Algoritam propagacije pogreške unazad metoda je izračuna gradijenata kompozicije funkcija, koje optimizacijski algoritmi koriste za učenje parametara mreže. Algoritam gradijente računa rekurzivno koristeći pravila ulančavanja. Pri učenju mreže gradijentnim spustom računamo gradijente funkcije gubitka s obzirom na težine i pomake neurona.

Za izračun gradijenata potrebno je odrediti parcijalne derivacije funkcije gubitka $\frac{\partial \mathcal{L}}{\partial w_{jk}^l}$ i $\frac{\partial \mathcal{L}}{\partial b_j^l}$. Težina w_{jk}^l povezuje k-ti neuron iz sloja $l-1$ s j-tim neuronom iz sloja l, a b_j^l je pomak j-tog neurona iz sloja l. Pogreška j-tog neurona u izlaznom sloju računa

se na sljedeći način:

$$\delta_j = \frac{\partial \mathcal{L}}{\partial z_j} = \frac{\partial \mathcal{L}}{\partial y_j} \cdot \frac{\partial y_j}{\partial z_j} = \frac{\partial \mathcal{L}}{\partial y_j} \cdot f'(z_j) \tag{2.21}$$

gdje je z_j izlaz j-tog neurona bez primjene aktivacijske funkcije, a y_j izlaz nakon primjene aktivacijske funkcije. Aktivacijska funkcija označena je s $f(z)$.

Kako bismo izračunali pogrešku j-tog neurona u nekom od prijašnjih slojeva l, množimo derivaciju aktivacijske funkcije i sumu umnožaka pogrešaka neurona sljedećeg sloja i odgovarajućih težina.

$$\delta_j^l = (w^{l+1})^T \delta^{l+1} \odot f'(z_j^l) \tag{2.22}$$

Budući da je $\frac{\partial \mathcal{L}}{\partial w_{j,k}^l} = y_k^{l-1} \delta_j^l$ i $\frac{\partial \mathcal{L}}{\partial b_j^l} = \delta_j^l$, nove vrijednosti parametara možemo izračunati pomoću izraza:

$$w_{i,j}^{l\ \prime} = w_{i,j}^l - \eta \cdot y_i^l \cdot \delta_j^{l+1} \tag{2.23}$$

$$b_i^{l\prime} = b_i^l - \eta \delta_j^{l+1} \tag{2.24}$$

3. Konvolucijske neuronske mreže

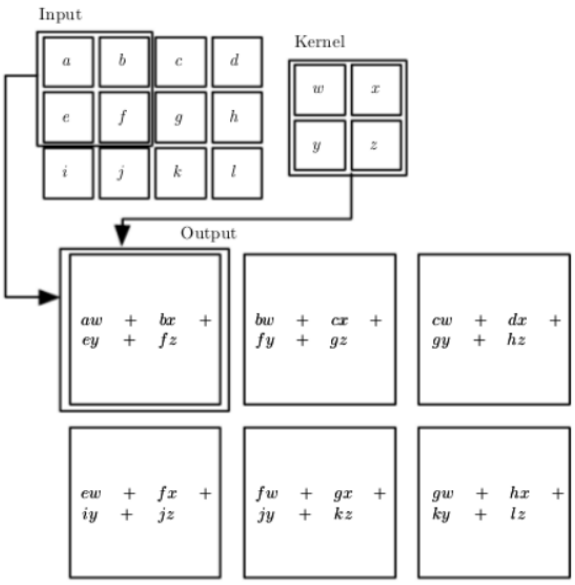
Konvolucijske neuronske mreže vrsta su neuronskih mreža prilagođena za podatke s topologijom rešetke kao što su slike [14]. Konvolucijski modeli moraju imati najmanje jedan konvolucijski sloj, a najčešće koriste i slojeve sažimanja, potpuno povezane slojeve i aktivacijske funkcije. U radu sa slikama, konvolucijske mreže mijenjaju potpuno povezane mreže jer potpuno povezane teško prepoznaju translatirane slike osim ako ih učimo za svaku translaciju. Također, kod potpuno povezanih modela, izlaz svakog neurona povezan je sa svim neuronima sljedećeg sloja pa pri radu sa slikama postoji previše parametara što otežava učenje i zaključivanje.

3.1. Konvolucijski slojevi

Konvolucijski slojevi imaju tri dimenzije, dvije prostorne i jednu semantičku. Za razliku od potpuno povezanih modela, neuroni u konvolucijskim modelima su povezani samo s malim brojem susjednih neurona prethodnog sloja pa se tako smanjuje broj parametara i olakšava učenje i zaključivanje. To znači da pri radu sa slikama izlazni pikseli ovise samo o lokalnom susjedstvu ulaznih piksela [14].

Parametri konvolucijskih slojeva koje prilagođavamo pri učenju nazivaju se jezgre. Jezgre su četverodimenzionalni vektori obično malih prostornih dimenzija. Kada ulazni podatak dođe na konvolucijski sloj, obavlja se dvodimenzionalna konvolucija ulaza s jezgrom i kao rezultat dobivamo dvodimenzionalnu mapu značajki. Pomičemo jezgru po dimenzijama ulaza i računamo skalarni produkt vrijednosti jezgre i vrijednosti ulaza na odgovarajućim pozicijama. Pošto konvolucijski slojevi imaju više jezgara, izračunava se više dvodimenzionalnih mapa značajki te se one slažu u dubinu i formiraju izlaz sloja.

Broj koji određuje za koliko se mjesta pomiče jezgra po ulaznom podatku zove se pomak (engl. *stride*). Površina ulaza koji utječe na jedan neuron sljedećeg sloja naziva se receptivno polje te odgovara veličini jezgre.



Slika 3.1: Prikaz dvodimenzionalne konvolucije ulaza s jezgrom. Preuzeto iz [14]

Popunjavanje nula (engl. *zero-padding*) je postupak proširivanja ulaznog podatka nulama kako bi se moglo kontrolirati veličinu izlaznog podatka te kako ne bi došlo do gubitka informacija [17].

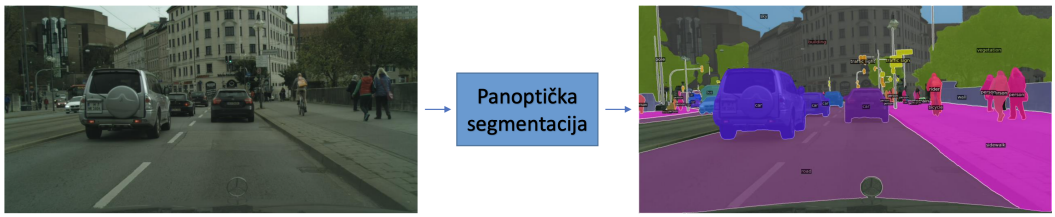
3.2. Slojevi sažimanja

Slojevi sažimanja mapiraju prostorno bliske značajke ulaznih podataka u jednu značajku na izlazu. Tako smanjuju dimenzionalnost podataka čime se smanjuje broj parametara i računalna zahtjevnost modela. Sloju sažimanja zadani su veličina regije sažimanja i pomak.

Primjeri sloja sažimanja su sažimanje maksimalnom vrijednosti, sažimanje srednjom vrijednosti, sažimanje L2 normom i sažimanje težinskim usrednjavanjem.

4. Panoptička segmentacija

Cilj je panoptičke segmentacije prepoznati prebrojive i neprebrojive razrede na slici tako da svakom pikselu pridruži oznaku razreda i identifikator primjerka. Kako bi to ostvarili potrebno je koristiti semantičku segmentaciju i segmentaciju primjeraka. Zbog različitih načina povezivanja njihovog povezivanja većinu suvremenih metoda panoptičke segmentacije možemo podijeliti u dvije kategorije [2]: metode od vrha prema dolje (engl. *top-down, box-based*) i metode od dna prema gore (engl. *bottom-up, box-free*).



Slika 4.1: Primjer panoptičke segmentacije slike prometa iz skupa podataka Cityscapes. Korišten je model Mask2Former.

4.1. Metode od vrha prema dolje

Metode od vrha prema dolje najčešće imaju dvije paralelne grane modela od kojih jedna nastoji predvidjeti okvir primjerka i pripadnu segmentacijsku masku, a druga obavlja semantičku segmentaciju. Nakon toga ove grane se spajaju i daju panoptička predviđanja.

Jedan od prvih primjera metoda od vrha prema dolje je model Panoptic Feature Pyramid Network [12] koji koristi mrežu Mask R-CNN [10] s granom za semantičku segmentaciju te pri formiranju panoptičke predikcije, za piksele kojima su dvije grane pridijelile različite razrede, preferira rezultat grane za segmentaciju primjeraka.

UPSNet [20] radi na sličnom principu ali koristi samo jednu Mask R-CNN okosnicu (engl. *backbone*) te ima dvije glave za semantičku segmentaciju i segmentaciju

primjeraka. Na kraju ima panoptičku glavu bez parametara (engl. *parameter-free panoptic head*) koja razrješava sukobe fuzije prethodne dvije glave uvodeći novi razred nepoznato (engl. *unknown*).

EfficientPS [15] temeljen na okosnici EfficientNet [18] ima veću učinkovitost i brzinu zaključivanja od prethodnih modela, ali ni za ovaj model ne možemo reći da model zaključuje u stvarnom vremenu. Metode od vrha prema dolje uglavnom su spore zbog više uzastopnih procesa u modelu.

4.2. Metode od dna prema gore

Metode od dna prema gore najčešće prvo obavljaju semantičku segmentaciju, a zatim prepoznavanje primjeraka grupiranjem piksela koji pripadaju prebrojivim razredima u grozdove.

Model DeeperLab [21] prvi je ovakav model koji uz granu za semantičku segmentaciju koristi i granu za segmentaciju primjeraka bez prepoznavanja kojem razredu primjerak pripada (engl. *class-agnostic*) koristeći kutove okvira primjerka (engl. *bounding-box corners*) i središta primjeraka.

Panoptic DeepLab [2] radi na sličnom principu, ali implementira nešto jednostavniju granu za segmentaciju primjeraka s dvije predikcijske glave od kojih jedna na izlaz daje toplinsku mapu središta objekata, a druga gustu mapu vektora pomaka. U naknadnoj obradi korištenjem ovih izlaza svakom se pikselu pridružuje odgovarajuće središte te se tako formiraju primjerci.

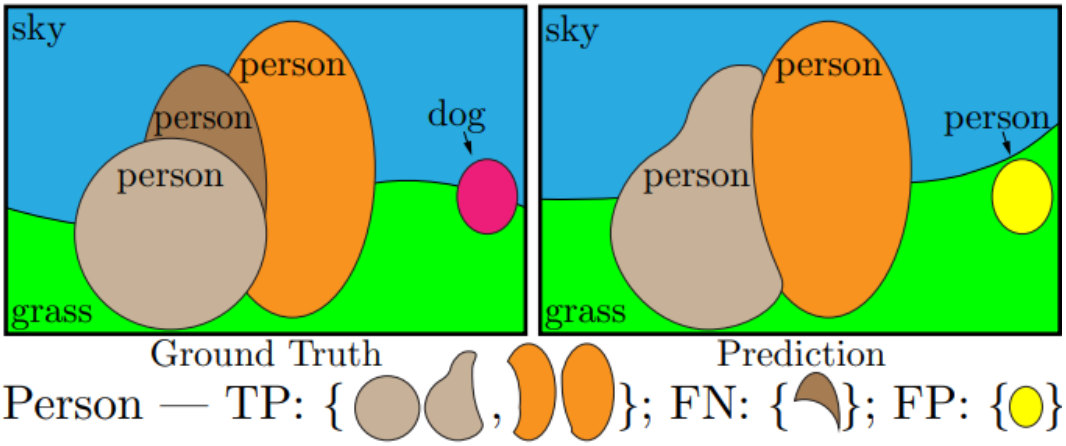
Metode od dna prema gore manje su zastupljene u panoptičkoj segmentaciji jer, iako postižu veće brzine detekcije, daju lošije rezultate od metoda od vrha prema dolje korištenjem istih mjerila.

4.3. Mjere kvalitete

Postojeće mjere kvalitete prilagođene su za semantičku segmentaciju i segmentaciju primjeraka te nisu dovoljno dobre za zadatke panoptičke segmentacije koji ujedinjuju oba postupka. Korištenje postojećih mjera uzrokovalo bi probleme u usporedbi rješenja i u komunikaciji te je zato bilo potrebno uvesti novu mjeru kvalitete. Zahtjevi mjere kvalitete su: potpunost (mjera treba tretirati prebrojive i neprebrojive razrede na jednak način obuhvaćajući sve aspekte zadatka), interpretabilnost (mjera treba imati jasno značenje i omogućavati usporedbu i komunikaciju) i jednostavnost (mjera mora

imati jednostavnu definiciju i implementaciju). Jednostavnost mjere važna je zbog brzine evaluacije modela. [13] predlaže mjeru panoptičke kvalitete PQ (engl. *panoptic quality*) vođenu ovim trima principima.

Ova mjera navodi da se segment koji je model predvidio i stvarni očekivani izlaz podudaraju ako je njihov omjer presjeka i unije (engl. *Intersection over Union*, IoU) strogo veći od 0.5. Zbog ovog zahtjeva i svojstva panoptičke segmentacije da se objekti ne preklapaju može postojati samo jedan predviđeni segment koji se podudara s očekivanim izlazom.



Slika 4.2: Ilustracija očekivanog izlaza (engl. *ground truth*) i predikcije panoptičke segmentacije na slici. Parovi segmenata iste boje imaju IoU veći od 0.5 i podudaraju se. Prikazano je kako su segmenti razreda osoba (engl. *person*) zastupljeni u ispravno pozitivnim TP, lažno negativnim FN i lažno pozitivnim FP rezultatima. Preuzeto iz [13].

Zahtjev da podudaranja moraju imati IoU veći od 0.5 osigurava svojstva jednostavnosti i interpretabilnosti zbog jednostavnosti izračunavanja i lako razumljivog objašnjenja.

PQ se računa pojedinačno za svaki razred te se zatim računa njihov prosjek. Za svaki razred, podudaranja se dijele na ispravno pozitivna TP, lažno pozitivna FP i lažno negativna podudaranja FN. Ispravno pozitivna podudaranja su ona u kojima su i predviđeni segment i segment očekivanog izlaza jednako klasificirani. Lažno pozitivna podudaranja su ona u kojima je model prepoznao razred koji se promatra, a odgovarajući segment na očekivanom izlazu ima pridijeljen različit razred. Lažno negativno podudaranje znači da je model prepoznao pogrešan razred, a očekivan je promatrani razred. Formula za PQ je:

$$PQ = \frac{\sum_{(p,q)\in TP} IoU(p,q)}{|TP| + \frac{1}{2}|FP| + \frac{1}{2}|FN|} \tag{4.1}$$

Izraz $\frac{1}{|TP|}\sum_{(p,q)\in TP} IoU(p,q)$ predstavlja aritmetičku sredinu IoU podudarnih segmenata, a $\frac{1}{2}|FP| + \frac{1}{2}|FN|$ se dodaje nazivniku kako bi se kaznili segmenti bez podudaranja.

PQ možemo prikazati kao umnožak segmentacijske kvalitete SQ i kvalitete prepoznavanja RQ:

$$PQ = \frac{\sum_{(p,q)\in TP} IoU(p,q)}{|TP|} \cdot \frac{|TP|}{|TP| + \frac{1}{2}|FP| + \frac{1}{2}|FN|} \tag{4.2}$$

gdje prvi faktor predstavlja SQ, a drugi RQ [13]. Ove dvije vrijednosti nisu nezavisne, ali je dekompozicija korisna u analizi rezultata. Ova definicija panoptičke kvalitete mjeri rezultate na svim razredima na jednak način koristeći jednostavnu i interpretabilnu formulu te tako zadovoljava postavljene zahtjeve.

5. Implementacije panoptičke segmentacije

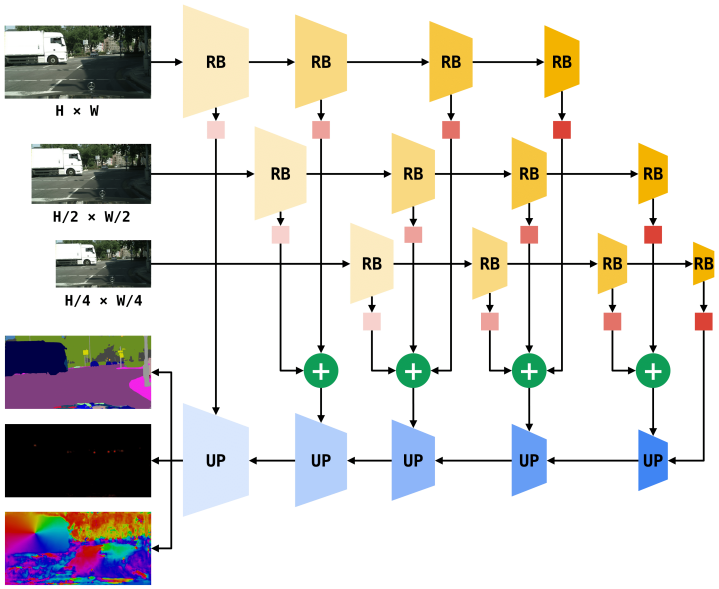
5.1. Panoptic SwiftNet

Panoptic SwiftNet [24] model je koji zadatke panoptičke segmentacije rješava primjenom piramidalne arhitekture SwiftNet [16]. Većina trenutnih rješenja koristi vrlo složene okosnice za postizanje velikog receptivnog polja što je potrebno za precizno prepoznavanje na slikama visoke rezolucije. Za razliku od ovih rješenja, Panoptic SwiftNet povećava receptivno polje i rasterećuje okosnicu primjenom slikovne piramide s različitim rezolucijama slike. Zbog ovakvog pristupa model pokazuje bolju generalizaciju od mnogih postojećih modela efikasne panoptičke segmentacije i pritom ostvaruje 60% brže zaključivanje.

Model sadrži tri glave za semantičku segmentaciju, regresiju središta i regresiju pomaka te na svaku od njih dovodi izlaz istog dekodera. Segmentacija primjeraka obavlja se kombiniranjem izlaza glava za regresiju središta i pomaka pri čemu se određuje razred prepoznatog primjerka. Konačno, fuzijom sve tri predikcijske glave formira se panoptička predikcija.

Panoptic SwiftNet za povećanje receptivnog polja koristi piramidalnu fuziju. To je postupak u kojem se na ulaz okosnice dovodi piramida s različitim rezolucijama iste slike što značajno unaprjeđuje prepoznavanje velikih objekata na slikama velike rezolucije uz neznatno povećanje računalnog opterećenja. Na taj način omogućeno je korištenje jednostavnijih okosnica bez žrtvovanja receptivnog polja čime se postiže brže zaključivanje. Ovaj model koristi okosnicu ResNet-18 [9] i slikovnu piramidu s punom rezolucijom te polovinom i četvrtinom rezolucije.

Arhitektura modela prikazana je na slici 5.1. Rezidualni blokovi okosnice ResNet-18 rade sa slikama na 1/4, 1/8, 1/16 i 1/32 izvorne rezolucije, a prikazani su žutim trapezima pri čemu trapezi iste boje dijele iste parametre. Preskočne veze rezidualnih blokova izvedene su pomoću 1×1 konvolucije i označene crvenim kvadratima na slici.



Slika 5.1: Na slici je prikazan model Panoptic SwiftNet s piramidalnom fuzijom. Žuti trapezi predstavljaju Rezidualni blokovi prikazani su žutim trapezima, 1×1 konvolucijske projekcije crvenim kvadratima, a moduli za naduzorkovanje plavim trapezima. Moduli iste boje dijele parametre. Preuzeto iz [24]

Značajke iz različitih razina piramide kombiniraju se zbrajanjem po elementima koje je na slici označeno zelenim krugovima.

Moduli za naduzorkovanje prikazani plavim trapezima zbrajanjem po elementima i jednom 3×3 konvolucijom kombiniraju značajke dobivene iz preskočnih veza iz okosnice i značajke koje daje prethodni modul za naduzorkovanje te ih naduzorkuje bilinearnom interpolacijom. Izlaz posljednjeg modula je tenzor značajki poduzorkovan na četvrtinu izvorne rezolucije. Ovaj je tenzor ulaz u sve tri predikcijske glave koje se sastoje od 1×1 konvolucije i naduzorkovanja bilinearnom interpolacijom na 4 puta veću rezoluciju od ulazne, odnosno na izvornu rezoluciju. Autori modela [24], BN-ReLU-CONV blok nazivaju konvolucijom zbog kratkoće zapisa.

Model primjenjuje Boundary-Aware Offset Loss za računanje gubitka pri učenju. Ovaj način računanja gubitka pikselima postavlja prioritet tako što svaki primjerak dijeli na četiri područja te pikselima u svakom području pridjeljuje težine. Najveću težinu imaju pikseli u području najbliže središtu primjerka, a najmanju oni uz rub primjerka. Gubitak se računa formulom:

$$\mathcal{L}_{BAOL} = \frac{1}{H \cdot W} \sum_{i,j}^{H,W} w_{i,j} \cdot |O_{i,j} - O_{i,j}^{GT}|, w_{i,j} \in 1, 2, 4, 8 \tag{5.1}$$

Gubitak koji se koristi za učenje modela sastoji se od gubitka semantičke segmentacije

$\mathcal{L}_{SEM}$, gubitka regresije središta $\mathcal{L}_{CEN}$ i gubitka pomaka $\mathcal{L}_{BAOL}$ koji se zbrajaju uz pripadne faktore λ:

$$\mathcal{L} = \lambda_{SEM} \cdot \mathcal{L}_{SEM} + \lambda_{CEN} \cdot \mathcal{L}_{CEN} + \lambda_{BAOL} \cdot \mathcal{L}_{BAOL} \tag{5.2}$$

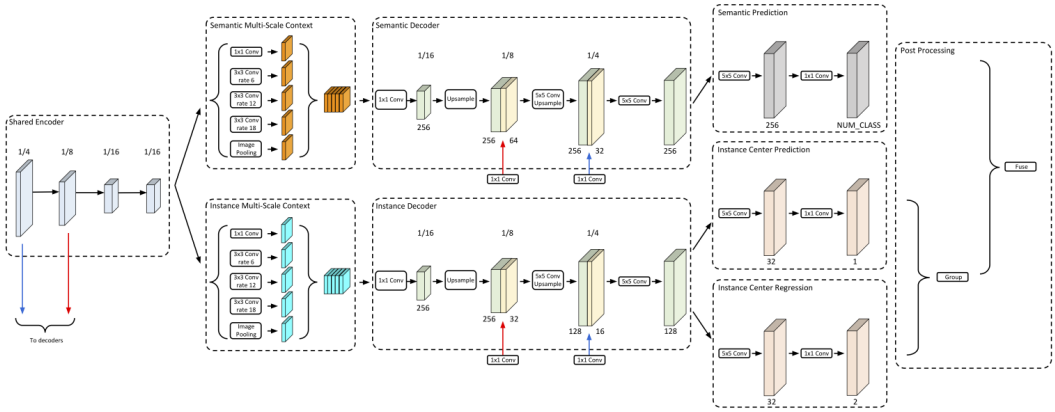
Panoptičku predikciju dobivamo naknadnom obradom (eng. *postprocessing*) izlaza glava za semantičku segmentaciju, regresiju središta i regresiju pomaka. Primjenjujući potiskivanje ne-maksimalnih odaziva (engl. *non-maximal suppresion*, NMS) na toplinske mape središta s izlaza glave za regresiju središta dobivamo središta primjeraka te svakom pikselu pridružujemo najbliže središte primjerka s obzirom na odgovarajuće pomake iz mape pomaka s izlaza glave za regresiju pomaka. Za svaki prepoznati primjerak određuje se semantički razred traženjem najzastupljenijeg prepoznatog razreda na pikselima tog primjerka.

5.2. Panoptic DeepLab

Panoptic DeepLab [2] metoda je panoptičke segmentacije od dna prema gore koja pruža rezultate usporedive s metodama od vrha prema dolje uz veliku brzinu zaključivanja. Panoptic DeepLab sastoji se od zajedničkog enkodera, dva odvojena modula za prostorno piramidalno sažimanje dilatiranim konvolucijama (engl. *Atrous Spatial Pyramid Pooling*, ASPP), dva dekodera specifična za semantičku segmentaciju i segmentaciju primjeraka i predikcijskih glava za semantičku segmentaciju, regresiju središta i regresiju pomaka. Grana za semantičku segmentaciju ima tipičnu strukturu modela za semantičku segmentaciju, primjerice DeepLab [1], a grana za segmentaciju primjeraka primjenjuje jednostavnu regresiju središta bez prepoznavanja razreda primjerka (engl. *class-agnostic*).

Okosnica enkodera preuzeta je od unaprijed naučene ImageNet neuronske mreže [6] uparene s dilatiranom konvolucijom (engl. *atrous convolution*) za dobivanje gušće mape značajki u zadnjem bloku. Zbog pretpostavke da grane za semantičku segmentaciju i segmentaciju primjeraka koriste različit kontekst i informacije za dekodiranje, model koristi odvojene ASPP i dekoder module. Glava za semantičku segmentaciju koristi težinski bootstrap gubitak unakrsne entropije (engl. *weighted bootstrapped cross entropy loss*) i predviđa prebrojive i neprebrojive razrede na slici. Gubitak se prilagođava korištenjem različitih težina za svaki piksel. Glava za segmentaciju primjeraka određuje središta primjeraka i pikselima prebrojivih razreda pridjeljuje pomake u odnosu na odgovarajuće središte bez prepoznavanja razreda kojem primjerak pripada.

Koristi se srednja kvadratna pogreška (engl. *Mean Squared Error*, MSE) za minimizaciju razlike predviđenih i očekivanih toplinskih mapa. Za predviđanje pomaka koristi se $\mathcal{L}_1$ gubitak koji se računa samo za piksele primjeraka prebrojivih razreda.



Slika 5.2: Panoptic-DeepLab sadrži enkoder, dva kontekstna modula i dva modula za dekodiranje za semantičku segmentaciju i segmentaciju primjeraka te glave za semantičku segmentaciju, regresiju središta i regresiju pomaka. U zadnjem sloju okosnice mreže koristi dilatiranu konvoluciju radi dobivanja gušće mape značajki. Modul za prostorno piramidalno sažimanje dilatiranim konvolucijama (ASPP) koristi se u kontekstnim modulima i modulima za dekodiranje koji se sastoje od jedne konvolucije u svakoj fazi naduzorkovanja. Segmentacija primjeraka obavlja se pridruživanjem središta pikselima s obzirom na mapu pomaka. Ovi rezultati i izlaz glave za semantičku segmentaciju spajaju se u panoptičku predikciju korištenjem principa "većine glasova". Preuzeto iz [2]

Formiranje panoptičke predikcije obavlja se na isti način kao i kod modela Panoptic SwiftNet. Prvo se na izlazima glava za regresiju središta i pomaka obavlja segmentacija primjeraka, a zatim se svakom primjerku dodjeljuje semantički razred po principu "većine glasova" [21]

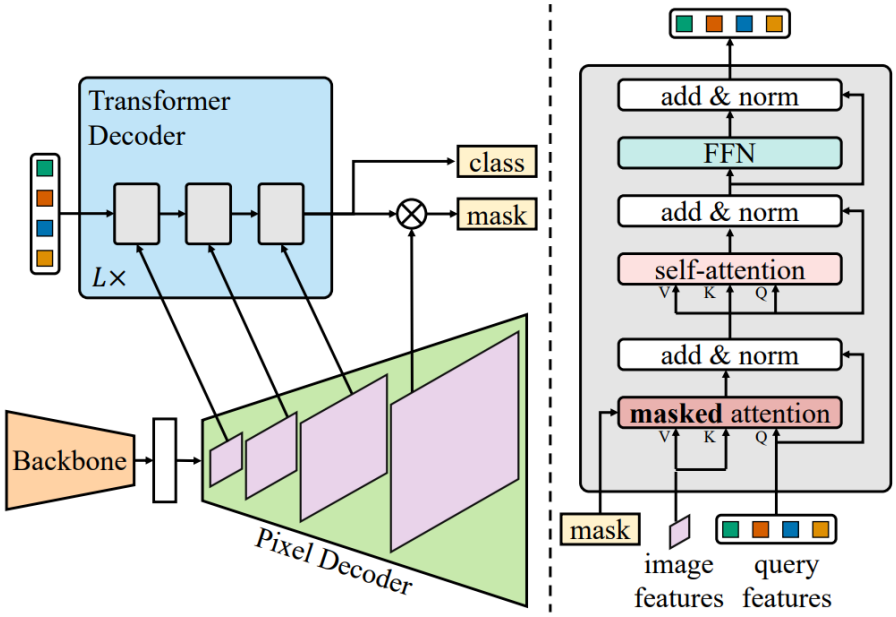
5.3. Mask2Former

Arhitekture za klasifikaciju maski grupiraju piksele u segmente predviđajući binarne maske za svaki segment i odgovarajuće oznake za svaku masku. Klasifikacijom maski moguće je obaviti svaki zadatak segmentacije pridjeljujući grupama piksela oznaku razreda za semantičku segmentaciju ili identifikator primjerka za segmentaciju primjeraka. Jednostavna arhitektura klasifikacije maski sastoji se od okosnice za izvlačenje značajki niske rezolucije iz slike, dekodera piksela koji postepeno naduzorkuje značajke niske rezolucije s izlaza okosnice te transformer dekoder koji radi sa zna-

čajkama slike. Konačna predviđanja binarne maske dekodiraju se iz izlaza dekodera piksela i dekodera transformer. MaskFormer [3] implementira ovakvu arhitekturu, a Mask2Former [4] prilagođuje arhitekturu mijenjanjem transformera.

Mask2Former koristi transformer dekoder s operatorom maskirane pažnje (engl. *masked attention operator*) koji izlučuje lokalizirane značajke ograničavajući unakrsnu pažnju na prednji plan predviđene maske. Studije [7] i [8] su pokazale da globalni kontekst u slojevima unakrsne pažnje uzrokuje sporu konvergenciju kod modela zasnovanih na transformeru, zato je u ovom modelu zamijenjen sloj unakrsne pažnje. Maskirana pažnja temelji se na pretpostavci da su lokalizirane značajke dovoljne za ažuriranje značajki upita te da se kontekstne informacije mogu izvući pomoću pažnje na samoga sebe (engl. *self-attention*).

Iako značajke visoke rezolucije unaprjeđuju učinkovitost modela, njihova je obrada računalno zahtjevna. Umjesto konstantnog korištenja mape značajki visoke rezolucije, dekoder piksela modela Mask2Former stvara piramidu značajki koja se sastoji od značajki niske i visoke rezolucije te daje jednu po jednu rezoluciju jednom sloju dekodera transformer u svakom trenutku.



Slika 5.3: Mask2Former koristi istu arhitekturu kao MaskFormer, s različitim dekoderom transformer koji koristi maskiranu pažnju umjesto unakrsne pažnje. Za rad s malim objektima iskorištavaju se značajke visoke rezolucije iz dekodera piksela dovodeći jednu rezoluciju značajke na jedan sloj transformer dekodera u jednom trenutku. Također je zamijenjen redoslijed sloja maskirane pažnje i sloja pozornosti na samog sebe. (engl. *self-attention*). Preuzeto iz [4].

6. Korišteni skup podataka

6.1. Cityscapes

Ovaj rad koristi skup podataka Cityscapes [5] za učenje i evaluaciju modela. Skup podataka sastoji se od slika prometa rezolucije 1024x2048 uslikanih u 50 gradova tijekom nekoliko mjeseci u proljeće, ljeto i jesen. Sve slike su uslikane danju u dobrim ili umjerenim vremenskim uvjetima. Skup sadrži 5000 slika s preciznim oznakama te 20000 slika s grubim oznakama na kojima je moguće prepoznati 30 razreda. U ovom radu koristim slike s preciznim oznakama i 19 od 30 razreda. Skup slika podijeljen je na skup za učenje koji sadrži 2975 slika, skup za provjeru koji sadrži 1525 slika i skup za testiranje koji sadrži 500 slika.

Za potrebe ovog rada, skup je pripremljen prema uputama projekta Detectron2 o pripremi skupa Cityscapes za panoptičku segmentaciju.

Odabrao sam skup podataka Cityscapes zato što sve korištene implementacije imaju dostupne prethodno trenirane modele na ovom skupu koje sam mogao evaluirati i usporediti. Sve implementacije također imaju dostupne prethodno trenirane modele na skupu COCO, ali zbog ograničenog prostora na usluzi Google Drive i veličine skupa COCO nisam mogao trenirati i evaluirati modele na usluzi Google Colab na ovom skupu.

7. Eksperimentalni rezultati

Eksperimenti u ovom radu izvedeni su na izvornom kodu opisanom u radovima Panoptic SwiftNet, Panoptic DeepLab i Mask2Former s manjim promjenama u kodu. Eksperimenti su provedeni na servisu Google Colab te na osobnom računalu s grafičkom karticom Nvidia GTX970 s 4 GB VRAM-a. Cilj ovih eksperimenata bio je usporediti rezultate različitih modela panoptičke segmentacije te pokazati utjecaj smanjenja rezolucije ulaznih slika na panoptičku kvalitetu i brzinu zaključivanja modela Panoptic SwiftNet.

7.1. Usporedba modela na punoj rezoluciji

Za provođenje eksperimenata bilo je potrebno instalirati biblioteku Detectron2. Evaluacija svih modela provedena je na servisu Google Colab koristeći grafičku karticu Nvidia Tesla K80 s 12 GB VRAM-a, koristeći dostupne težine prethodno treniranih modela. Za usporedbu učinkovitosti modela korištene su mjere PQ, SQ i RQ te vrijeme zaključivanja.

Model Panoptic SwiftNet naučena je na 2975 slika iz skupa za učenje. Učen je na 90000 iteracija s optimizatorom ADAM na rezoluciji 1024x2048 s 8 slika po grupi za učenje. Stopa učenja smanjuje se polinomijalno s $5 \cdot 10^{-5}$ na 10^{-7}. Rezultati evaluacije prikazani su u tablici 7.1. Ukupno vrijeme zaključivanja pri evaluaciji na 500 slika skupa za testiranje iznosi 3 minute i 32 sekunde, a ukupno vrijeme zaključivanja po slici 0.4282 sekunde. Ukupno vrijeme samog izračuna predikcije (engl. *inference pure compute time*) bez naknadne obrade izlaza (engl. *post-processing*) iznosi 57 sekundi, odnosno 0.1166 sekunde po slici.

Model Panoptic DeepLab koji sam evaluirao koristi okosnicu HRNet-48 [19] jer s tom okosnicom daje najbolje rezultate. Učen je na 90000 iteracija s optimizatorom ADAM na rezoluciji 1024x2048 s 32 slike po grupi za učenje. Početna stopa učenja je 10^{-3} te se polinomijalno smanjuje. Rezultati su prikazani u tablici 7.2 Ukupno vrijeme zaključivanja pri evaluaciji na skupu za testiranje iznosi 10 minuta i 7 sekundi,

Tablica 7.1: Rezultati evaluacije modela Panoptic SwiftNet [24]

	PQ	SQ	RQ	mIoU	broj razreda
Svi razredi	55.943	79.477	68.895	77.213	19
Prebrojivi razredi	44.956	77.592	57.428	77.913	8
Neprebrojivi razredi	63.918	80.847	77.234	76.700	11

a ukupno vrijeme zaključivanja po slici 1.2274 sekunde. Ukupno vrijeme samog izračuna predikcije bez naknadne obrade izlaza iznosi 4 minute i 37 sekundi, odnosno 0.5616 sekundi po slici.

Tablica 7.2: Rezultati evaluacije modela Panoptic DeepLab [2]

	PQ	SQ	RQ	mIoU	broj razreda
Svi razredi	63.406	81.879	76.369	80.579	19
Prebrojivi razredi	56.307	80.389	69.835	81.150	8
Neprebrojivi razredi	68.570	82.962	81.121	80.182	11

Evaluirao sam model Mask2Former s okosnicom Swin-L i R101. Modeli su naučeni na 90000 iteracija s optimizatorom ADAMW na rezoluciji 1024x2048 sa 16 slika po grupi za učenje. Početna stopa učenja je 10^{-4}. Potrebno je napomenuti da Swin-L koristi unaprijed naučene težine na ImageNet-u za inicijalizaciju.

Rezultati modela s okosnicom R101 prikazani su u tablici 7.3. Ukupno vrijeme zaključivanja pri evaluaciji na 500 slika skupa za testiranje iznosi 19 minuta i 17 sekundi, a ukupno vrijeme samog zaključivanja po slici 2.3370 sekundi. Ukupno vrijeme samog izračuna predikcije bez naknadne obrade izlaza iznosi 5 minuta i 20 sekundi, odnosno 0.6476 sekundi po slici.

Tablica 7.3: Rezultati evaluacije modela Mask2Former s okosnicom R101 [4]

	PQ	SQ	RQ	mIoU	broj razreda
Svi razredi	62.360	81.676	75.437	78.570	19
Prebrojivi razredi	54.751	81.031	67.359	77.650	8
Neprebrojivi razredi	67.894	82.146	81.312	79.245	11

Rezultati modela s okosnicom Swin-L prikazani su u tablici 7.4. Ukupno vrijeme zaključivanja pri evaluaciji na 500 slika skupa za testiranje iznosi 28 minuta i 31 sekundu, a ukupno vrijeme zaključivanja po slici 3.4561 sekunde. Ukupno vrijeme sa-

mog izračuna predikcije bez naknadne obrade izlaza iznosi 15 minuta i 10 sekundi, odnosno 1.8390 sekundi po slici.

Tablica 7.4: Rezultati evaluacije modela Mask2Former s okosnicom Swin-L [4]

	PQ	SQ	RQ	mIoU	broj razreda
Svi razredi	66.648	83.053	79.499	82.861	19
Prebrojivi razredi	60.400	82.310	73.103	83.788	8
Neprebrojivi razredi	71.192	83.593	84.151	82.191	11

Prikazani rezultati pokazuju da modeli Panoptic DeepLab i Mask2Former nude bolju točnost (PQ), ali je vrijeme zaključivanja znatno veće nego kod modela Panoptic SwifNet. Panoptic SwiftNet na skupu podataka Cityscapes nudi točnost približnu najboljim rješenjima uz znatno višu brzinu zaključivanja te teži zaključivanju u stvarnom vremenu. Treba napomenuti da se modeli Panoptic DeepLab i Mask2Former mogu koristiti i s nešto lakšim okosnicama čime se ubrzava zaključivanje, ali i znatno smanjuje panoptička kvaliteta.

Arhitektura Panoptic SwiftNeta omogućava učenje na znatno manjim računalnim resursima nego druga dva uspoređena modela, što mi je omogućilo učenje modela na platformi Google Colab i osobnom računalu.

7.2. Usporedba rezultata modela Panoptic SwiftNet na različitim rezolucijama

Kako bih analizirao utjecaj rezolucije ulaznih slika, model sam učio na rezoluciji 256×512, 512×1024 i punoj rezoluciji 1024×2048. Sva tri učenja izvedena su na 15000 iteracija s grupom za učenje od 1 slike zbog ograničenja platforme Google Colab.

Za usporedbu učinkovitosti modela navedene su mjere PQ, SQ i RQ te vrijeme zaključivanja i vrijeme učenja.

Rezultati učenja na slikama rezolucije 256×512 prikazani su u tablici 7.5. Ukupno vrijeme zaključivanja na 500 slika skupa za testiranje iznosi 3 minute i 8 sekundi, odnosno 0.3804 sekunde po slici. Ukupno vrijeme samog izračuna predikcije (engl. *inference pure compute time*) bez naknadne obrade izlaza (engl. *post-processing*) iznosi 16 sekundi, odnosno 0.0333 sekunde po slici. Ukupno vrijeme učenja na 15000 iteracija iznosi 51 minutu i 14 sekundi.

Rezultati učenja na slikama rezolucije 512×1024 prikazani su u tablici 7.6. Ukupno

Tablica 7.5: Rezultati učenja modela Panoptic SwiftNet na slikama rezolucije 256×512.

	PQ	SQ	RQ	mIoU	broj razreda
Svi razredi	26.058	69.109	33.143	45.844	19
Prebrojivi razredi	10.331	66.352	14.647	36.788	8
Neprebrojivi razredi	37.497	71.114	46.594	52.427	11

vrijeme zaključivanja na skupu za testiranje iznosi 3 minute i 21 sekundu, odnosno 0.4068 sekundi po slici. Ukupno vrijeme samog izračuna predikcije bez naknadne obrade izlaza iznosi 28 sekundi, odnosno 0.0579 sekundi po slici. Ukupno vrijeme učenja na 15000 iteracija iznosi 1 sat 9 minuta i 23 sekunde.

Tablica 7.6: Rezultati učenja modela Panoptic SwiftNet na slikama rezolucije 512×1024.

	PQ	SQ	RQ	mIoU	broj razreda
Svi razredi	33.350	72.617	42.198	57.099	19
Prebrojivi razredi	15.708	69.331	22.196	49.150	8
Neprebrojivi razredi	46.180	75.008	56.745	62.891	11

Rezultati učenja na slikama rezolucije 1024×2048 prikazani su u tablici 7.7. Ukupno vrijeme zaključivanja na skupu za testiranje iznosi 3 minute i 47 sekundi, odnosno 0.4587 sekundi po slici. Ukupno vrijeme samog izračuna predikcije bez naknadne obrade izlaza iznosi 57 sekundi, odnosno 0.1171 sekundi po slici. Ukupno vrijeme učenja na 15000 iteracija iznosi 2 sata i 16 sekundi.

Tablica 7.7: Rezultati učenja modela Panoptic SwiftNet na slikama rezolucije 1024×2048.

	PQ	SQ	RQ	mIoU	broj razreda
Svi razredi	36.946	73.712	46.849	60.5802	19
Prebrojivi razredi	15.284	69.869	21.788	49.038	8
Neprebrojivi razredi	52.700	76.507	65.076	68.955	11

Iz prikazanih rezultata vidljivo je da se smanjenjem rezolucije ulaznih slika smanjuje i točnost, ali povećava brzina zaključivanja. Smanjenje rezolucije na pola smanjuje vrijeme izračuna predikcije bez naknadne obrade izlaza za 50% uz smanjenje panoptičke kvalitete za 10%. Daljnje smanjenje rezolucije, na četvrtinu izvorne, smanjuje vrijeme izračuna predikcije bez naknadne obrade za 43% ali pri tome smanjuje panoptičku kvalitetu za 22%.

Za bolju usporedbu koliko rezolucija ulaznih slika utječe na efikasnost, učenje sam ponovio na 30000 iteracija za izvornu rezoluciju i rezoluciju 512×1024 kako bih usporedio rezultate s višim vrijednostima panoptičke kvalitete.

Rezultati učenja na slikama rezolucije 512×1024 prikazani su u tablici 7.8. Ukupno vrijeme zaključivanja na skupu za testiranje iznosi 3 minute i 11 sekundi, odnosno 0.3872 sekundi po slici. Ukupno vrijeme samog izračuna predikcije bez naknadne obrade izlaza iznosi 29 sekundi, odnosno 0.0592 sekundi po slici.

Tablica 7.8: Rezultati učenja modela Panoptic SwiftNet na slikama rezolucije 512×1024 na 30000 iteracija.

	PQ	SQ	RQ	mIoU	broj razreda
Svi razredi	38.479	74.327	48.567	62.500	19
Prebrojivi razredi	24.718	72.332	33.663	58.525	8
Neprebrojivi razredi	48.486	75.777	59.406	65.400	11

Rezultati učenja na slikama rezolucije 1024×2048 prikazani su u tablici 7.9. Ukupno vrijeme zaključivanja na skupu za testiranje iznosi 3 minute i 46 sekundi, odnosno 0.4581 sekundi po slici. Ukupno vrijeme samog izračuna predikcije bez naknadne obrade izlaza iznosi 59 sekundi, odnosno 0.1195 sekundi po slici.

Tablica 7.9: Rezultati učenja modela Panoptic SwiftNet na slikama rezolucije 1024×2048 na 30000 iteracija.

	PQ	SQ	RQ	mIoU	broj razreda
Svi razredi	45.039	76.235	56.707	68.565	19
Prebrojivi razredi	29.797	73.985	40.129	64.725	8
Neprebrojivi razredi	56.125	77.872	68.763	71.336	11

Iz navedenih rezultata vidljivo je da je razlika PQ nešto veća nego kod rezultata učenja na 15000 iteracija (15%), ali vrijeme zaključivanja ostaje isto i razlika u vremenu zaključivanja ostaje ista.

Proveo sam učenje modela na slikama rezolucije 512×1024 sa 6 slika po grupi za učenje na 90000 iteracija. Model je pokazao najveću vrijednost panoptičke kvalitete na 60000 iteracija. Ovi rezultati prikazani su u tablici 7.10. Ukupno vrijeme zaključivanja na skupu za testiranje iznosi 3 minute i 4 sekunde, odnosno 0.3727 sekundi po slici. Ukupno vrijeme samog izračuna predikcije bez naknadne obrade izlaza iznosi 28 sekundi, odnosno 0.0575 sekundi po slici.

Tablica 7.10: Rezultati učenja modela Panoptic SwiftNet na slikama rezolucije 512×1024 na 60000 iteracija sa 6 slika po grupi za učenje.

	PQ	SQ	RQ	mIoU	broj razreda
Svi razredi	43.870	75.844	55.088	67.218	19
Prebrojivi razredi	31.641	74.519	41.537	64.825	8
Neprebrojivi razredi	52.763	76.808	64.942	68.973	11

Vrijednost dobivene panoptičke kvalitete manja je za 21.6% od panoptičke kvalitete modela učenog na slikama rezolucije 1024×2048 prikazane u radu [24], ali treba napomenuti da je model u navedenom radu učen s većom grupom za učenje. U provedenom eksperimentu veličina grupe za učenje bila je ograničena računalnim resursima.

8. Zaključak

U ovom radu objašnjene su osnove umjetnih neuronskih mreža. Opisana je njihova arhitektura i postupak učenja. Posebno su pojašnjene konvolucijske mreže zbog njihove primjene u panoptičkoj segmentaciji.

Detaljnije je pojašnjen problem panoptičke segmentacije i podjela metoda rješavanja problema na metode od vrha prema dolje (engl. *top-down, box-based*) i metode od dna prema gore (engl. *bottom-up, box-free*). Opisana je mjera kvalitete nazvana panoptička kvaliteta predložena u radu [13] koja je korištena za vrednovanje opisanih modela te je prikazan način izračunavanja njene vrijednosti.

Cilj rada bio je opisati i vrednovati implementacije panoptičke segmentacije te usporediti njihove rezultate na skupu podataka Cityscapes. Uspoređene su panoptičke kvalitete PQ i vremena zaključivanja modela Panoptic SwiftNet, Panoptic DeepLab i Mask2Former, što je pokazalo da Panoptic DeepLab i Mask2Former imaju nešto više vrijednosti PQ, ali i značajno veće vrijeme zaključivanja od modela Panoptic SwiftNet. Provedeno je učenje modela Panoptic SwiftNet na ulaznim slikama različite rezolucije te su uspoređene vrijednosti PQ i vremena zaključivanja. Pokazalo se da smanjenje rezolucije za pola smanjuje vrijeme zaključivanja za 50% uz smanjenje panoptičke kvalitete za 10 ili 15% ovisno o postavkama učenja modela.

U budućem radu bilo bi korisno vrednovati i usporediti neke od metoda od vrha prema dolje (engl. *top-bottom*) s opisanim metodama. Također, bilo bi zanimljivo naučiti i vrednovati korištene modele s različitim okosnicama ili na drugim skupovima podataka. Uz veće računalne resurse, moglo bi se izvesti učenje modela Panoptic SwiftNet s različitim rezolucijama ulaznih slika na većem broju iteracija i s većom grupom za učenje radi bolje usporedbe rezultata. Bilo bi korisno odrediti optimalne hiperparametre pri učenju modela Panoptic SwiftNet na slikama rezolucije 512×1024 i usporediti rezultate s rezultatima iz izvornog članka.

LITERATURA

[1] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, i Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. U *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.

[2] Bowen Cheng, Maxwell D Collins, Yukun Zhu, Ting Liu, Thomas S Huang, Hartwig Adam, i Liang-Chieh Chen. Panoptic-deeplab: A simple, strong, and fast baseline for bottom-up panoptic segmentation. U *CVPR*, 2020.

[3] Bowen Cheng, Alex Schwing, i Alexander Kirillov. Per-pixel classification is not all you need for semantic segmentation. U M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, i J. Wortman Vaughan, urednici, *Advances in Neural Information Processing Systems*, svezak 34, stranice 17864–17875. Curran Associates, Inc., 2021. URL `https://proceedings.neurips.cc/paper/2021/file/950a4152c2b4aa3ad78bdd6b366cc179-Paper.pdf`.

[4] Bowen Cheng, Ishan Misra, Alexander G Schwing, Alexander Kirillov, i Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. U *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, stranice 1290–1299, 2022.

[5] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, i Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. U *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[6] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, i Li Fei-Fei. Imagenet: A large-scale hierarchical image database. U *2009 IEEE Conference on Computer Vision and Pattern Recognition*, stranice 248–255, 2009. doi: 10.1109/CVPR.2009.5206848.

[7] Peng Gao, Minghang Zheng, Xiaogang Wang, Jifeng Dai, i Hongsheng Li. Fast convergence of detr with spatially modulated co-attention. U *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, stranice 3621–3630, 2021.

[8] Golnaz Ghiasi, Yin Cui, Aravind Srinivas, Rui Qian, Tsung-Yi Lin, Ekin D. Cubuk, Quoc V. Le, i Barret Zoph. Simple copy-paste is a strong data augmentation method for instance segmentation. U *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, stranice 2918–2928, 2021.

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, i Jian Sun. Deep residual learning for image recognition. U *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[10] Kaiming He, Georgia Gkioxari, Piotr Dollar, i Ross Girshick. Mask r-cnn. U *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017.

[11] Diederik P. Kingma i Jimmy Ba. Adam: A method for stochastic optimization, 2014. URL `https://arxiv.org/abs/1412.6980`.

[12] Alexander Kirillov, Ross Girshick, Kaiming He, i Piotr Dollar. Panoptic feature pyramid networks. U *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[13] Alexander Kirillov, Kaiming He, Ross Girshick, Carsten Rother, i Piotr Dollar. Panoptic segmentation. U *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[14] Josip Krapac i Siniša Šegvić. Konvolucijski modeli. URL `http://www.zemris.fer.hr/~ssegvic/du/du2convnet.pdf`. Pristupljeno: 25. lipnja 2022.

[15] Rohit Mohan i Abhinav Valada. Efficientps: Efficient panoptic segmentation. *International Journal of Computer Vision*, 129(5):1551–1579, 2021.

[16] Marin Oršić i Siniša Šegvić. Efficient semantic segmentation with pyramidal fusion. *Pattern Recognition*, 110:107611, 2021. ISSN 0031-3203. doi: https://doi.org/10.1016/j.patcog.2020.107611. URL `https://www.sciencedirect.com/science/article/pii/S0031320320304143`.

[17] Keiron O'Shea i Ryan Nash. An introduction to convolutional neural networks, 2015. URL https://arxiv.org/abs/1511.08458.

[18] Mingxing Tan i Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. U *International conference on machine learning*, stranice 6105–6114. PMLR, 2019.

[19] Jingdong Wang, Ke Sun, Tianheng Cheng, Borui Jiang, Chaorui Deng, Yang Zhao, Dong Liu, Yadong Mu, Mingkui Tan, Xinggang Wang, Wenyu Liu, i Bin Xiao. Deep high-resolution representation learning for visual recognition. *TPAMI*, 2019.

[20] Yuwen Xiong, Renjie Liao, Hengshuang Zhao, Rui Hu, Min Bai, Ersin Yumer, i Raquel Urtasun. Upsnet: A unified panoptic segmentation network. U *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[21] Tien-Ju Yang, Maxwell D. Collins, Yukun Zhu, Jyh-Jing Hwang, Ting Liu, Xiao Zhang, Vivienne Sze, George Papandreou, i Liang-Chieh Chen. Deeperlab: Single-shot image parser, 2019. URL https://arxiv.org/abs/1902.05093.

[22] Marko Čupić. Umjetna inteligencija: Umjetne neuronske mreže, 2016. URL http://java.zemris.fer.hr/nastava/ui/ann/ann-20180604.pdf.

[23] Marko Čupić. Duboko učenje: Optimizacija parametara modela, 2019. URL http://www.zemris.fer.hr/~ssegvic/du/du3optimization.pdf.

[24] Josip Šarić, Marin Oršić, i Siniša Šegvić. Panoptic swiftnet: Pyramidal fusion for real-time panoptic segmentation, 2022. URL https://arxiv.org/abs/2203.07908.

Konvolucijski modeli za panoptičku segmentaciju

Sažetak

Panoptička segmentacija problem je računalnog vida koji svakom pikselu slike pridružuje oznaku razreda i identifikator primjerka. Ujedinjuje semantičku segmentaciju i segmentaciju primjeraka. U ovom su radu opisani modeli panoptičke segmentacije, njihova podjela i način vrednovanja. Također je dan kratki opis osnova dubokih neuronskih mreže, posebno konvolucijskih mreža, i postupka učenja. Opisani modeli Panoptic SwiftNet, Panoptic DeepLab i Mask2Former naučeni su i vrednovani na skupu podataka Cityscapes. Modeli su uspoređeni na temelju vrijednosti panoptičke kvalitete i vremena zaključivanja. Model Panoptic SwiftNet naučen je na slikama rezolucije 256×512, 512×1024 i 1024×2048. Na temelju tih rezultata, napravljena je analiza utjecaja rezolucije ulaznih slika na vrijednost panoptičke kvalitete i vrijeme zaključivanja modela.

Ključne riječi: panoptička segmentacija, neuronske mreže, konvolucijske neuronske mreže, Panoptic SwiftNet, Panoptic DeepLab, Mask2Former

Convolutional models for panoptic segmentation

Abstract

Panoptic segmentation is a computer vision task that assigns a class label and instance index to every pixel on an image. It unifies semantic and instance segmentation. This paper describes panoptic segmentation models, their classification, and the method of evaluation. It also provides a short description of deep neural networks basics, especially convolutional networks, and the learning process. Panoptic SwiftNet, Panoptic DeepLab and Mask2Former models are described and evaluated on the Cityscapes dataset. Models are compared based on panoptic quality value and inference time. Panoptic SwiftNet model is learned on images with resolution of 256×512, 512×1024 and 1024×2048. An analysis of the influence that input image resolution has on panoptic quality value and inference time was made based on these results.

Keywords: panoptic segmentation, neural networks, convolutional neural networks, Panoptic SwiftNet, Panoptic DeepLab, Mask2Former