SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 604

SEGMENTACIJA PRIMJERAKA NAD NEURAVNOTEŽENIM TAKSONOMIJAMA

Rej Šafranko

Zagreb, rujan, 2024.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

Zagreb, 4. ožujka 2024.

DIPLOMSKI ZADATAK br. 604

Pristupnik:	**Rej Šafranko (0036525383)**
Studij:	Računarstvo
Profil:	Znanost o podacima
Mentor:	prof. dr. sc. Siniša Šegvić

Zadatak: **Segmentacija primjeraka nad neuravnoteženim taksonomijama**

Opis zadatka:

Segmentacija primjeraka važan je problem računalnog vida s mnogim zanimljivim primjenama. U posljednje vrijeme stanje tehnike postižu duboki modeli zasnovani na konvolucijama i slojevima pažnje. Ipak, standardni posutpci ne uspijevaju postići zadovoljavajuću generalizaciju u prisustvu malenih primjeraka i neuravnoteženih taksonomija. U okviru rada, potrebno je odabrati okvir za automatsku diferencijaciju te upoznati biblioteke za rukovanje tenzorima i slikama. Proučiti i ukratko opisati postojeće duboke arhitekture za gustu predikciju. Odabrati slobodno dostupne skupove slika te oblikovati podskupove za učenje, validaciju i testiranje. Primijeniti naučene modele, prikazati eksperimente na nekom javnom skupu podataka te usporediti generalizacijsku izvedbu sa stanjem tehnike. Komentirati učinkovitost učenja i zaključivanja. Preporučiti smjernice za ugradnju modela u praktičan tehnički sustav. Predložiti pravce za budući rad. Radu priložiti izvorni i izvršni kod razvijenih postupaka, ispitne slijedove i rezultate, uz potrebna objašnjenja i dokumentaciju. Citirati korištenu literaturu i navesti dobivenu pomoć.

Rok za predaju rada: 28. lipnja 2024.

Hvala prof. dr. sc. Siniši Šegviću na mentorstvu i pomoći pri izradi ovog rada

Sadržaj

1. Uvod

Računalni vid, kao jedno od najbrže rastućih područja u znanosti o računalima, ima široku primjenu u zadacima poput autonomnih vozila, medicinske dijagnostike i prepoznavanja lica. Unutar ovog područja, segmentacija primjeraka predstavlja ključni zadatak koji ne samo da klasificira svaki piksel slike, već i razlikuje različite primjerke unutar istog razreda objekata. Na primjer, u slici s više automobila, segmentacija primjeraka prepoznaje svaki automobil kao zaseban objekt.

Tradicionalni modeli, poput Mask R-CNN-a [1], postali su standard u segmentaciji primjeraka zbog svoje preciznosti i fleksibilnosti. Međutim, suočavaju se s izazovima kada rade s malim i deformabilnim objektima, kao što su otpaci u prirodnim okruženjima [2]. U novije vrijeme, modeli koji koriste jezične reprezentacije kao što su CLIP [3] i FC-CLIP [4] kombiniraju vizualne i tekstne informacije kako bi omogućili klasifikaciju i segmentaciju bez dodatnog učenja, što otvara nove mogućnosti za rad s nepoznatim razredima [3].

U ovom radu istraženi su različiti pristupi segmentaciji primjeraka, uključujući konvencionalne modele poput Mask R-CNN-a i Mask2Former-a [5], kao i modele s jezičnim ugrađivanjima CLIP i FC-CLIP. U eksperimentima s modelima Mask R-CNN i Mask2Former implementirao sam dinamičko uravnotežavanje težina segmentacijskog gubitka ovisno o validacijskom odzivu modela za različite razrede. Cilj ovog pristupa je istražiti mogućnost bolje segmentacijske izvedbe modela pri neuravnoteženom broju primjera po razredu. Ovaj rad pruža uvid u prednosti i nedostatke pristupa segmentaciji primjeraka te ističe važnost prilagodbe modela specifičnim zadacima, poput segmentacije otpada, kako bi se postigli optimalni rezultati.

2. Segmentacija

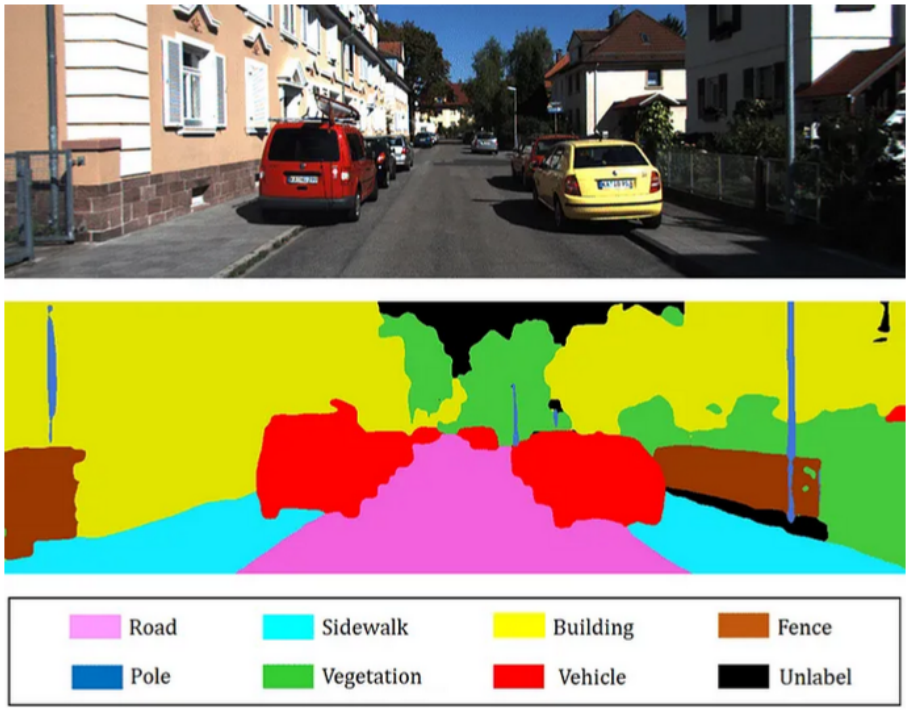
Segmentacija je ključan zadatak u obradi slika i računalnom vidu koji omogućuje detaljno razumijevanje sadržaja slike dijeleći je na različite semantičke ili fizičke cjeline. Za razliku od klasičnih metoda prepoznavanja objekata, gdje se identificiraju samo opći razredi objekata, segmentacija pruža dublji uvid određivanjem točnih granica i pozicija pojedinih objekata ili regija unutar slike. Ovaj postupak je od izuzetne važnosti u brojnim primjenama, uključujući autonomna vozila, medicinsku dijagnostiku, robotiku i analizu slika u stvarnom vremenu.

U okviru segmentacije razlikuju se tri glavne metode: semantička segmentacija, segmentacija primjeraka i panoptička segmentacija. Svaka od ovih metoda ima svoje specifične primjene i tehnike koje omogućuju različite razine detalja u analizi slike. Semantička segmentacija omogućuje klasifikaciju svakog piksela slike, dok segmentacija primjeraka ide korak dalje i razlikuje pojedinačne objekte unutar istog razreda. Panoptička segmentacija predstavlja sveobuhvatan pristup koji kombinira prednosti obje metode, omogućujući detaljno razumijevanje kompleksnih scena.

2.1. Semantička segmentacija

Semantička segmentacija je zadatak guste predikcije jer svaki piksel ulazne slike klasificira u odgovarajući razred. Semantička segmentacija ne razlikuje instance određenog razreda, već sve piksele tog razreda grupira u jednu cjelinu.

Tijekom predviđanja segmentacijske mape, iz ulazne slike nastaje više kanala jednakih dimenzija kao i ulazna slika. Svaki kanal odgovara točno jednom razredu od skupa svih mogućih razreda. Pojedinačan piksel nekog kanala ima vrijednost veću od 0 ako i samo ako pripada razredu kojeg taj kanal opisuje. Također, vrijednost piksela odgovara in-



Slika 2.1. Primjer semantičke segmentacije scene iz prometa, preuzeto iz [6]

deksu razreda. Izlazna segmentacijska mapa nastaje preslikavanjem pobjedničkog indeksa u kod (npr. boju) odgovarajućeg razreda.

2.2. Segmentacija primjeraka

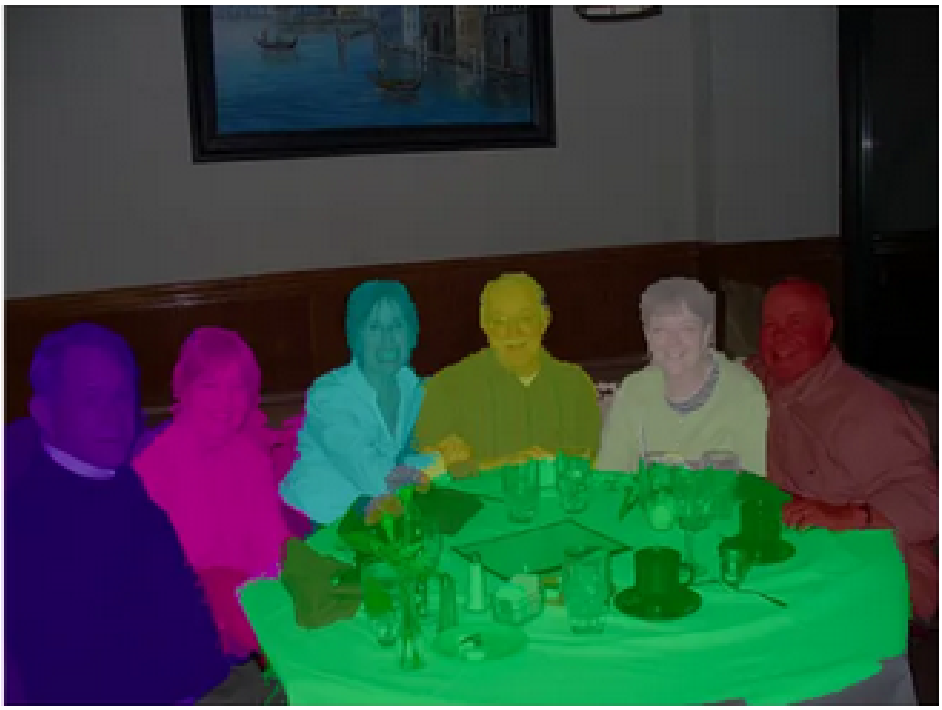
Segmentacija primjeraka nadogradnja je na zadatak semantičke segmentacije, a njen cilj je precizno locirati i klasificirati svaki pojedinačni primjerak objekta na slici [1]. Za razliku od semantičke segmentacije [7], koja označava svaki piksel na temelju pripadnosti određenom razredu, segmentacija primjeraka ide korak dalje pružajući dodatnu razinu specifičnosti razlikovanjem različitih primjeraka unutar istog razreda. Na primjer, u slici s više automobila, semantička segmentacija bi sve automobile označila istom oznakom, dok bi segmentacija primjeraka svakom automobilu dala jedinstvenu oznaku, što omogućuje precizniju analizu scena i bolje razumijevanje strukture slike.

Segmentacija primjeraka sastoji se od tri ključna zadatka: detekcija objekata, klasifikacija objekata i semantička segmentacija. Svaki od ovih zadataka igra ulogu u postizanju konačnog cilja – prepoznavanja i razlikovanja pojedinačnih objekata na slici te određivanja njihove precizne lokacije.

- **Detekcija objekata**: Ovaj zadatak obuhvaća pronalaženje objekata pomoću pravokutnika koji uokviruju svaki objekt na slici. Algoritmi poput Region Proposal Networks (RPN) [8] generiraju prijedloge za moguće regije unutar slike gdje se objekti mogu nalaziti. U okviru segmentacije primjeraka, detekcija objekata je prvi korak koji omogućuje lociranje svakog pojedinačnog objekta, ali bez informacija o preciznim granicama ili pikselima koje taj objekt zauzima.

- **Klasifikacija objekata**: Nakon što su objekti detektirani, zadatak klasifikacije se koristi za određivanje kojem razredu pripada svaki detektirani objekt. Klasifikacija dodjeljuje oznaku svakom objektu na temelju prepoznatog razreda, primjerice, je li objekt automobil, bicikl, pješak ili neki drugi razred. U ovoj fazi algoritam koristi klasifikacijske modele, obično temeljene na dubokim neuronskim mrežama, kako bi odredio pripadnost detektiranih objekata. Detekcija i klasifikacija najčešće dijele značajke, što znači da se iste značajke koje se koriste za detekciju objekata mogu koristiti i za njihovu klasifikaciju [1]. Klasifikacija je ključna u razlikovanju različitih razrede objekata na slici.

- **Segmentacija objekata**: Nakon što su objekti detektirani i klasificirani, zadatak segmentacije primjeraka odnosi se na precizno određivanje koji pikseli na slici ili uokvirenoj regiji slike pripadaju kojem pojedinačnom objektu. Za razliku od detekcije objekata koja daje samo okvir oko objekta, segmentacija primjeraka određuje točne granice svakog pojedinačnog objekta. Ovo omogućava razlučivanje složenih scena gdje se objekti preklapaju ili dodiruju. U segmentaciji primjeraka, maske koje predstavljaju različite objekte mogu se preklapati ako je takav skup podataka korišten za učenje modela. S druge strane, u panoptičkoj segmentaciji, maske moraju biti disjunktne, što znači da svaki piksel pripada točno jednom primjerku i jednom razredu.



Slika 2.2. Primjer segmentacije primjeraka nad slikom više osoba, preuzeto iz [9]

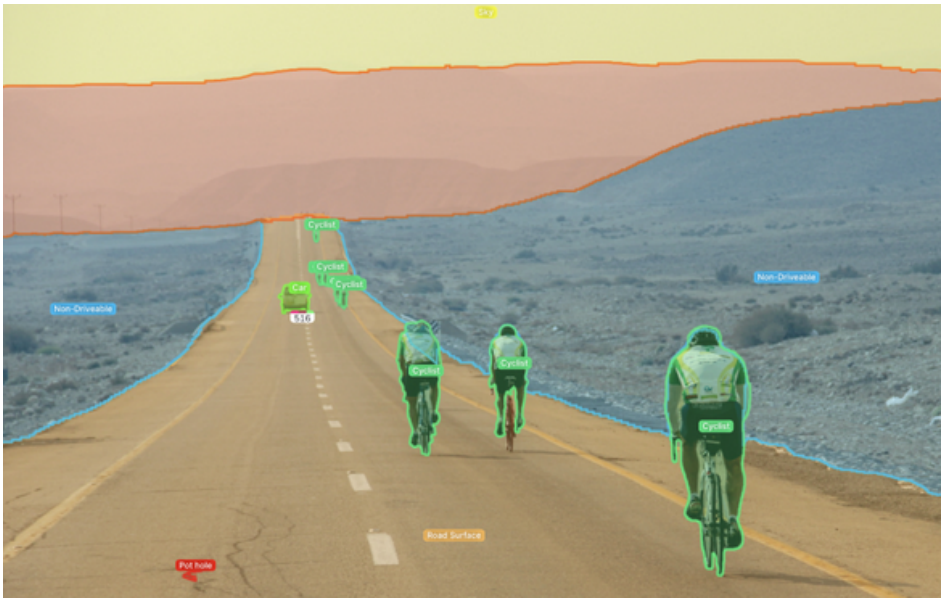
2.3. Panoptička segmentacija

Panoptička segmentacija predstavlja najnoviji i sveobuhvatni pristup u segmentaciji slika, koji kombinira prednosti semantičke segmentacije i segmentacije primjeraka. Cilj panoptičke segmentacije je ostvariti cjelovitu analizu scene tako da se svi pikseli klasificiraju u odgovarajuće razrede, dok se istovremeno razlikuju pojedinačni primjerci unutar tih razreda.

Jedan od pristupa panoptičkoj segmentaciji uključuje korištenje semantičke komponente koja osigurava klasifikaciju piksela u općenite razrede (npr. cesta, zgrada, vegetacija), dok komponenta segmentacije primjeraka omogućuje razlikovanje različitih primjeraka unutar razreda poput ljudi, vozila ili drugih objekata. Na taj način, panoptička segmentacija ne samo da pruža informacije o općim razredima objekata, nego i precizno identificira i razdvaja pojedinačne primjerke, čak i kada se oni preklapaju ili dodiruju u složenim scenama.

Postoje i integrirani panoptički pristupi, poput Mask2Former [5], koji koristi zajednički mehanizam za učinkovito i precizno izvođenje panoptičke segmentacije bez potrebe za odvojenim komponentama.

Panoptička segmentacija omogućuje sveobuhvatno razumijevanje slike, što ga čini posebno korisnim u primjenama poput autonomnih vozila, nadzora i robotike, gdje je važno ne samo prepoznati objekte, već i razlikovati njihove pojedinačne primjerke unutar scene.



Slika 2.3. Primjer panoptičke segmentacije scene iz prometa, preuzeto iz [10]

3. Pristupi segmentaciji primjeraka

Pojava dubokog učenja donijela je revolucionarne promjene u području segmentacije primjeraka, otvarajući nove mogućnosti za preciznija i učinkovitija rješenja ovog problema [1]. Konvolucijske neuronske mreže (CNN) postale su ključan alat u računalnom vidu jer omogućuju učenje složenih hijerarhijskih značajki izravno iz podataka, čime se izbjegava potreba za ručno definiranim značajkama [11]. Time su postignuta znatna poboljšanja u točnosti i sposobnosti generalizacije modela. Poseban napredak prema segmentaciji primjeraka postignut je razvojem modela zasnovanih na regijama, poput mreža R-CNN [12] te njihovih poboljšanih inačica, kao što su Fast R-CNN [8] i Faster R-CNN [13]. Ovi modeli integriraju regionalne prijedloge s konvolucijskim mrežama, omogućujući precizno određivanje položaja i segmentaciju objekata unutar slike uz povećanje brzine obrade. Osim toga, metode poput Mask R-CNN dodaju sloj za segmentaciju koji omogućuje preciznu klasifikaciju po pikselima unutar predloženih regija, što dovodi do ohrabrujućih rezultata u segmentaciji primjeraka u različitim primjenama, od autonomnih vozila do medicinske dijagnostike.

Moderni pristupi segmentaciji primjeraka se mogu kategorizirati u pristupe temeljene na regionalnim prijedlozima i metode izravne predikcije.

3.1. Pristupi utemeljeni na regionalnim prijedlozima

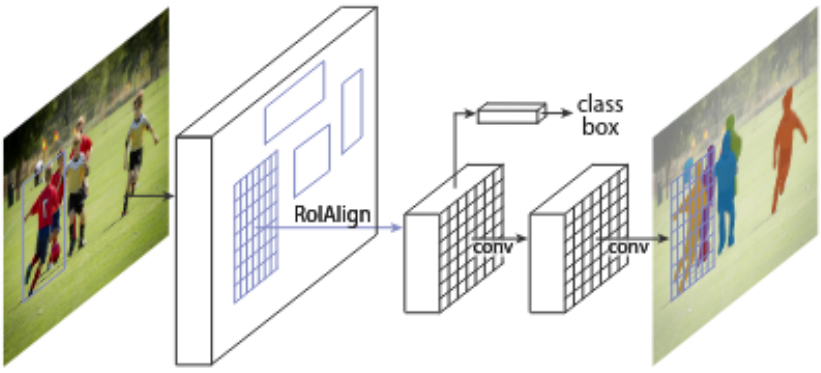
Metode temeljene na regionalnim prijedlozima za segmentaciju primjeraka obično slijede dvostupanjski postupak: graničnim pravokutnicima se uokviruju prijedlozi regija koji potencijalno sadrže objekte, a zatim se predložene regije dodatno obrađuju kako bi se proizvela segmentacijska mapa za svaki predloženi objekt [1]. Ovaj pristup, koji kombinira lokalizaciju objekata i njihovu segmentaciju, značajno je doprinio poboljšanju toč-

nosti i robusnosti modela za segmentaciju primjeraka. Ključni modeli u ovoj kategoriji uključuju Mask R-CNN, MaskLab [14] i Mask Scoring R-CNN [15], koji su postavili standarde u ovom području. Mask R-CNN, kao jedan od najistaknutijih modela, nadograđuje se na osnovu Faster R-CNN modela dodavanjem grane za segmentaciju koja omogućuje točnu piksel-po-piksel klasifikaciju unutar predloženih regija, čime se postiže visoka preciznost u razlikovanju različitih primjeraka [1]. U ovom radu fokusiram se na primjenu Mask R-CNN modela zbog njegove široke primjene i učinkovitosti u različitim zadacima segmentacije primjeraka.

3.1.1. Mask R-CNN

Mask R-CNN proširuje Faster R-CNN arhitekturu dodavanjem grane za predviđanje segmentacijskih mapi paralelno s postojećim granama za klasifikaciju i regresiju graničnog okvira. Ova arhitektura predstavlja ključni korak u unapređenju segmentacije primjeraka jer omogućuje simultano otkrivanje objekata i precizno segmentiranje svakog primjerka unutar slike.

Mask R-CNN slijedi dvostupanjsku arhitekturu: prvo, koristi mrežu za prijedloge regija (RPN) koja generira potencijalne regije s objektima. Zatim, na temelju tih regija koristi sloj ROIAlign, koji omogućuje precizno prostorno poravnanje značajki, čime se izbjegavaju pogreške uzrokovane kvantizacijom u ranijim pristupima poput RoIPool [12]. Ovo prostorno poravnanje ključno je za točnu piksel-po-piksel segmentaciju objekata.



Slika 3.1. Arhitektura Mask R-CNN modela, preuzeto iz [1]

Mask R-CNN ima tri paralelne grane: prvu za klasifikaciju objekata, drugu za regresiju graničnog okvira i treću za predviđanje binarne segmentacijske mape. Segmentacijska

mapa se za svaki objekt unutar svake regije interesa (ROI) generira korištenjem potpune konvolucijske mreže (FCN) [7], čime se zadržava prostorna struktura. Mapa se predviđa neovisno za svaki razred, što eliminira međusobno natjecanje između različitih razreda, za razliku od uobičajenih pristupa u semantičkoj segmentaciji. Formulacija uključuje funkciju gubitka koja uzima u obzir gubitak pri klasifikaciji, gubitak pri regresiji graničnog okvira i binarni unakrsni entropijski gubitak pri predviđanju mape. Ovakva formulacija omogućuje mreži da neovisno generira segmentacijske mape za svaki razred objekta.

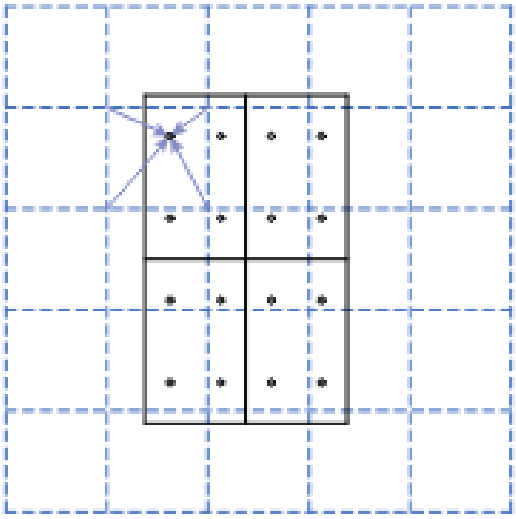
Jedna od ključnih inovacija Mask R-CNN-a je sloj ROIAlign, koji poboljšava preciznost segmentacije u usporedbi s ranijim metodama. ROIAlign izbjegava kvantizaciju i koristi bilinearnu interpolaciju za točno očitavanje značajki, što omogućuje visoku preciznost čak i kod vrlo sitnih objekata ili pri gusto zbijenim scenama [1].

```python
def _roi_align(input, rois, spatial_scale, pooled_height, pooled_width, sampling_ratio, aligned):
    orig_dtype = input.dtype
    input = maybe_cast(input)
    rois = maybe_cast(rois)
    _, _, height, width = input.size()
    ph = torch.arange(pooled_height, device=input.device)
    pw = torch.arange(pooled_width, device=input.device)
    roi_batch_ind = rois[:, 0].int()
    offset = 0.5 if aligned else 0.0
    roi_start_w = rois[:, 1] * spatial_scale - offset
    roi_start_h = rois[:, 2] * spatial_scale - offset
    roi_end_w = rois[:, 3] * spatial_scale - offset
    roi_end_h = rois[:, 4] * spatial_scale - offset
    roi_width = roi_end_w - roi_start_w
    roi_height = roi_end_h - roi_start_h
    if not aligned:
        roi_width = torch.clamp(roi_width, min=1.0)
        roi_height = torch.clamp(roi_height, min=1.0)

    bin_size_h = roi_height / pooled_height
    bin_size_w = roi_width / pooled_width
    exact_sampling = sampling_ratio > 0
    roi_bin_grid_h = sampling_ratio if exact_sampling else torch.ceil(roi_height / pooled_height)
    roi_bin_grid_w = sampling_ratio if exact_sampling else torch.ceil(roi_width / pooled_width)

    if exact_sampling:
        count = max(roi_bin_grid_h * roi_bin_grid_w, 1)
        iy = torch.arange(roi_bin_grid_h, device=input.device)
        ix = torch.arange(roi_bin_grid_w, device=input.device)
        ymask = None
        xmask = None
    else:
        count = torch.clamp(roi_bin_grid_h * roi_bin_grid_w, min=1)
        iy = torch.arange(height, device=input.device)
        ix = torch.arange(width, device=input.device)
        ymask = iy[None, :] < roi_bin_grid_h[:, None]
        xmask = ix[None, :] < roi_bin_grid_w[:, None]

    def from_K(t):
        return t[:, None, None]

    y = (
        from_K(roi_start_h)
        + ph[None, :, None] * from_K(bin_size_h)
        + (iy[None, None, :] + 0.5).to(input.dtype) * from_K(bin_size_h / roi_bin_grid_h)
    )
    x = (
        from_K(roi_start_w)
        + pw[None, :, None] * from_K(bin_size_w)
        + (ix[None, None, :] + 0.5).to(input.dtype) * from_K(bin_size_w / roi_bin_grid_w)
    )
    val = _bilinear_interpolate(input, roi_batch_ind, y, x, ymask, xmask)
    if not exact_sampling:
        val = torch.where(ymask[:, None, None, None, :, None], val, 0)
        val = torch.where(xmask[:, None, None, None, None, :], val, 0)
    output = val.sum((-1, -2))

    if isinstance(count, torch.Tensor):
        output /= count[:, None, None, None]
    else:
        output /= count

    output = output.to(orig_dtype)
    return output
```

Slika 3.2. Izvorni kod ROIAlign operacije u torchvision biblioteci. Metoda _roi_align koristi se za izdvajanje mapa značajki fiksne veličine iz mapa značajki regija interesa. Metoda izračunava vrijednosti značajki unutar regije interesa koristeći bilinearnu interpolaciju, osiguravajući prostornu preciznost. Metoda prilagođava koordinate regija interesa skaliranjem prema parametru spatial_scale, dijeli ih u ćelije te interpolira koristeći mrežu točaka uzorkovanja, kontroliranu parametrom sampling_ratio.

Sloj ROIAlign

RoIAlign (Region of Interest Align) [1] važna je inovacija uvedena u Mask R-CNN-u kako bi se riješila ograničenja tradicionalne operacije RoIPool (Region of Interest Pooling) [8]. RoIPool, standardna operacija u mrežama za otkrivanje objekata, pati od problema s kvantizacijom i neusklađenošću poravnanja prilikom izvlačenja značajki iz regija interesa (RoI). Ovi problemi su posebno štetni kad ciljamo na maske s točnošću na razini piksela u zadacima segmentacije primjeraka [1]. RoIPool kvantizira realne vrijednosti u diskretnu granulaciju mape značajki. Te vrijednosti se u konačnici sažimaju maksimalnom vrijednošću [16] što uzrokuje neporavnanje između izlučenih značajki i regija interesa. Kako bi izbjegli diskretizaciju, RoIAlign radi s realnim vrijednostima i koristi bilinearnu interpolaciju pri izračunu značajki za predodređene točke unutar regije interesa. Regije interesa se dijele u ćelije, a ROIAlign uzorkuje značajke ili na sredini svake ćelije ili na uniformno raspoređenim mjestima. Time postiže poravnanje između izlučenih značajki i regije interesa u razini piksela.



Slika 3.3. Regija interesa (RoI) je podijeljena na 4 ćelije i unutar svake se odabiru 4 uniformno raspoređene točke. One će biti bilinearno interpolirane RoIAlign metodom. Povećanje broja točaka za uzorkovanje može uhvatiti više prostornih detalja pod cijenu veće računalne složenosti. Preuzeto iz [1]

Mask R-CNN se pokazao izuzetno fleksibilnim, ne samo u segmentaciji primjeraka već i u drugim zadacima poput procjene ljudske poze. Model postiže vrhunske rezultate na COCO [17] mjerilu u segmentaciji primjeraka, detekciji objekata i procjeni ključnih točaka, a zbog svoje jednostavnosti i učinkovitosti postao je standard u području raču-

nalnog vida.

3.2. Pristupi s izravnom predikcijom primjeraka

Ovi pristupi izravno regresiraju skup primjeraka bez potrebe za prethodnim generiranjem regionalnih prijedloga. Ovi pristupi izravno predviđaju segmentacijske mape za svaki primjerak scene. Za razliku od pristupa koji se oslanjaju na dvostupanjski proces (kao što je generiranje regionalnih prijedloga i njihovu kasniju obradu), izravni pristupi nastoje objediniti cijeli proces segmentacije unutar jedinstvene arhitekture koja simultano detektira i segmentira objekte. Korištenje slojeva pažnje važan je preduvjet za ostvarenje pristupa utemeljenih na izravnoj predikciji. Zbog svoje jednostavnosti i visoke točnosti, pristupi temeljeni na izravnoj predikciji postaju sve popularniji, a modeli poput Mask2Former postavljaju nove standarde u segmentaciji primjeraka, panoptičkoj segmentaciji i semantičkoj segmentaciji. [5].

3.2.1. Mask2Former

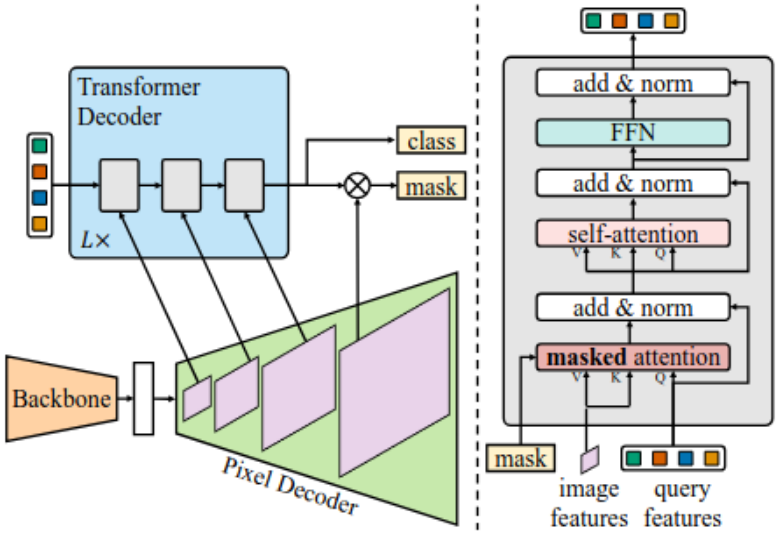
Mask2Former donosi novi pristup univerzalnoj segmentaciji slika koristeći arhitekturu modela zasnovanog na maskiranoj pažnji [18]. Ovaj model proširuje pristup koji je prvi put uveden u DETR [18], primjenjujući ga na segmentaciju primjeraka, semantičku i panoptičku segmentaciju [19].

Mask2Former se sastoji od tri glavne komponente: osnovnog prednaučenog modela (okosnica), dekodera piksela i dekodera maski (slika 3.4). Ključno poboljšanje s obzirom na Mask-RCNN leži u primjeni maskirane pažnje unutar dekodera maski, što omogućuje da se pažnja usmjeri na lokalizirane značajke unutar predviđenih segmentacijskih maski. Ovaj pristup izbjegava globalno usmjerenu pažnju, koja je često problematična zbog sporije konvergencije i manje preciznosti. Umjesto toga, Mask2Former koristi maskirane pažnje koje su lokalizirane unutar segmentacijskih maski, čime se značajno poboljšava brzina učenja i cjelokupna izvedba modela [5].

Mehanizam maskirane pažnje je različit u odnosu na klasični mehanizam pažnje u dekoder arhitekturama po tome što usmjerava pažnju samo na prednji plan segmentacijske mape umjesto na sve značajke ulazne slike. Ovim pristupom pažnja je ograničena na po-

područja gdje se objekti vjerojatno nalaze, što pomaže modelu da učinkovitije uči značajke i granice objekata. Binarna mapa generirana u prethodnom sloju identificira relevantna područja (prednji plan), a mehanizam pažnje prilagođava se tako da djeluje samo unutar tih područja. Utjecaj pažnje za pozadinske dijelove je potisnut, što omogućuje ignoriranje nevažnih dijelova slike [5].

Model koristi višerazinske značajke visoke razlučivosti kako bi poboljšao segmentaciju malih objekata, dok optimizacije poput izmjene redoslijeda slojeva pažnje i uklanjanja nepotrebnih slojeva dodatno poboljšavaju generalizacijsko svojstvo modela. Zahvaljujući ovim inovacijama, Mask2Former postiže vrhunske rezultate na popularnim skupovima podataka poput COCO, ADE20K [20] i Cityscapes [21] u sve tri glavne zadaće segmentacije: panoptičkoj segmentaciji, semantičkoj segmentaciji i segmentaciji primjeraka.



Slika 3.4. Opisana arhitektura Mask2Former modela, preuzeto iz [5]

U usporedbi sa starijim univerzalnim modelima, Mask2Former nadmašuje specijalizirane arhitekture i omogućuje preciznu segmentaciju čak i u zahtjevnim scenarijima.

Time predstavlja suvremen primjer pristupa utemeljenih na izravnoj predikciji koristeći napredne tehnike poput modela zasnovanog na maskiranim pažnjama, čineći ga izuzetno učinkovitim u svim segmentacijskim zadacima.

4. Primjena jezičnih reprezentacija za segmentaciju primjeraka

Tradicionalni pristupi segmentaciji primjeraka, poput konvolucijskih neuronskih mreža u kombinaciji s modelima za detekciju objekata, postigli su značajan napredak u preciznosti i učinkovitosti. Ipak, najnovija istraživanja u području multimodalnih arhitektura [3], koje primaju više vrsta ulaznih podataka, poput slike i teksta, pokazuju potencijal za dodatno poboljšanje performansi segmentacije s otvorenim rječnikom. Takvi modeli, koji koriste višedimenzionalne reprezentacije informacija, mogu obogatiti proces učenja i omogućiti bolju generalizaciju u različitim kontekstima.

4.1. CLIP

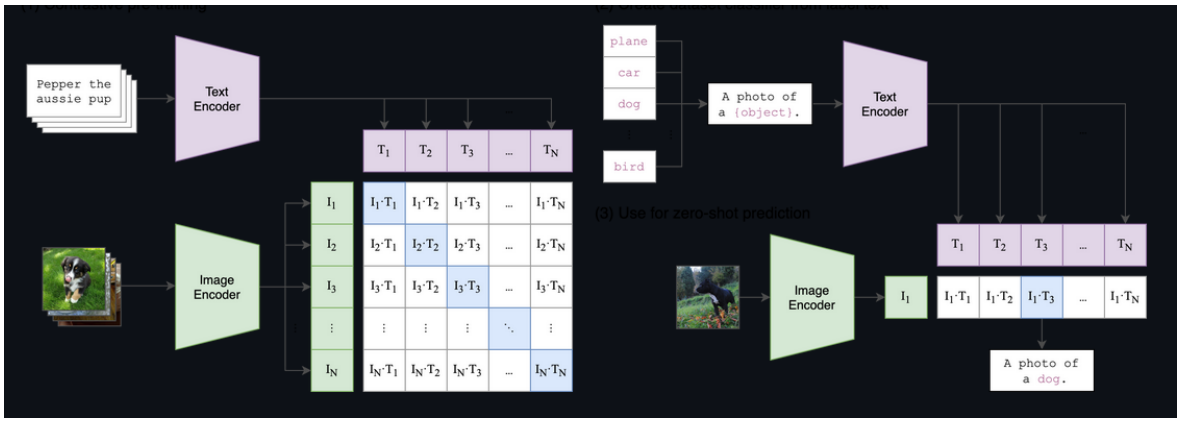
CLIP (Contrastive Language-Image Pre-training) multimodalni je model koji povezuje računalni vid i obradu prirodnog jezika. Razvijen od strane OpenAI-a 2021., CLIP je dizajniran kako bi omogućio učinkovit prijenos znanja između slika i njihovih odgovarajućih tekstnih opisa. Temelji se na kontrastivnom učenju [22], gdje se model uči na velikom skupu podataka od 400 milijuna parova slika i pripadajućih tekstnih opisa. Glavna ideja CLIP-a je stvaranje zajedničkog vektorskog prostora u kojem se semantički povezane slike i tekstovi nalaze blizu jedno drugome, omogućujući tako širok raspon zadataka poput klasifikacije slika bez učenja (zero-shot learning), pretraživanja slika putem tekstnih upita i drugih primjena [3]. Arhitektura CLIP-a sastoji se od dvije glavne komponente:

- **Enkoder slike**: Enkoder slike, koji u izvedenim radovima često igra ulogu okosnice, može biti ili model zasnovan na pažnji za obradu slika (Vision Transformer) [23] ili konvolucijska neuronska mreža (CNN), poput ResNet-a [24]. Enkoder slike obrađuje ulazne slike i generira značajke koje se smještaju u zajednički ugrađeni

prostor.

- **Enkoder teksta** Enkoder teksta temelji se na modelu zasnovanom na pažnji za obradu teksta, koji se pokazao revolucionarnim u obradi prirodnog jezika [25]. Enkoder teksta pretvara ulazne tekstove (poput opisa ili razreda objekata) u vektorske reprezentacije u istom ugrađenom prostoru kao i enkoder slike.

CLIP se uči kontrastivnim učenjem koje nastoji maksimizirati sličnost između stvarnih parova slike i teksta, dok se istovremeno minimizira sličnost između pogrešnih parova slike i teksta. U procesu učenja, za svaki par slike i teksta, CLIP uspoređuje sve moguće parove unutar jedne grupe ulaznih primjera i koristi oblik kontrastnog gubitka (gubitak N parova) [26]. Ovakav pristup omogućuje CLIP-u da nauči bogate reprezentacije koje dobro generaliziraju čak i na zadatke na kojima nije bio izravno učen [3].



Slika 4.1. Arhitektura CLIP modela. Primjer matričnog množenja tekstnih značajki i značajki izlučenih iz slike, preuzeto iz [3]

4.1.1. Uvod u klasifikaciju slika

Klasifikacija slika temeljni je zadatak u računalnom vidu, gdje je cilj pridružiti ulaznoj slici odgovarajuću oznaku razreda na temelju njenog sadržaja. Tradicionalni modeli za klasifikaciju slika oslanjaju se na vizualne značajke izlučene pomoću konvolucijskih neuronskih mreža (CNN) te predviđaju razrede iz unaprijed definiranog, zatvorenog skupa oznaka (taksonomije). Međutim, ovakav pristup ograničava modele na prepoznavanje samo onih razreda koji su prisutni u skupu za učenje, što smanjuje njihovu sposobnost generalizacije na nepoznate razrede.

Pojava modela poput CLIP-a uvela je novi pristup razumijevanju i obrade slika, koji ko-

risti ne samo vizualne informacije nego i bogat semantički sadržaj ugrađen u tekstne opise. CLIP je model koji zajednički uči reprezentacije slika i tekstova u zajedničkom latentnom semantičkom prostoru. Umjesto da izravno predviđa kategoričke oznake, CLIP omogućuje usporedbu semantičke sličnosti između slike i teksta, što omogućuje generalizaciju na zadatke s otvorenim vokabularom bez potrebe za opsežnim skupovima označenih podataka. U ovom potpoglavlju prikazujem načela koja stoje iza CLIP-a i kako se on učinkovito može primijeniti na klasifikaciju slika.

4.1.2. Klasifikacija slika CLIP-om

CLIP uvodi novu paradigmu za klasifikaciju slika omogućujući modelu da klasificira slike na temelju opisa u prirodnom jeziku, a ne unaprijed definiranih oznaka razreda. Ključna ideja je iskoristiti zajednički ugrađeni prostor za slike i tekstove kako bi se odredila sličnost između slike i potencijalnih tekstnih opisa njenog razreda [3].

- **Dizajn tekstnih upita**: Prije klasifikacije slike, prvo se stvara skup tekstnih upita koji predstavljaju svaki mogući razred. Na primjer, ako želimo klasificirati sliku kao "pas", "mačka" ili "ptica", možemo koristiti upite poput "fotografija psa", "fotografija mačke" i "fotografija ptice". Ova fleksibilnost omogućuje uključivanje detaljnijih ili specifičnih izraza ovisno o domeni primjene.

- **Ugrađivanje teksta**: Tekstni upiti prolaze kroz enkoder teksta kako bi generirali vektore značajki za svaki razred unutar zajedničkog ugrađenog prostora.

- **Ugrađivanje slike**: Ulazna slika se obrađuje pomoću enkodera slike kako bi proizveli odgovarajući vektor značajki.

- **Usporedba sličnosti**: Klasifikacija se izvodi izračunavanjem kosinusne sličnosti između vektora značajki slike i svakog od vektora značajki za razrede u tekstnom obliku. Razred s najvišim rezultatom sličnosti odabire se kao predviđen razred te slike.

Ovakav pristup omogućuje CLIP-u generalizaciju na nove još neviđene razrede bez potrebe za eksplicitnim ponovnim učenjem. Jednostavnom promjenom ili proširivanjem skupa tekstnih upita, model može klasificirati slike na fleksibilan način u otvorenom vokabularu [3]. Na primjer, ako je standardni model za klasifikaciju slika učen za pre-

poznavanje životinja, ali ne i specifičnih pasmina, taj model ne bi mogao razlikovati različite pasmine pasa. CLIP, s druge strane, može razlikovati pasmine ako mu se daju upiti poput "fotografija labradora" ili "fotografija ovčara", čak i bez eksplicitnog učenja na tim razredima.

Osim zero-shot klasifikacije, CLIP se može dodatno učiti što zovemo few-shot učenje. Samo s nekoliko označenih primjera, CLIP može prilagoditi svoj ugrađeni prostor kako bi poboljšao izvedbu na specifičnim razredima, blago mijenjajući tekstne upite ili koristeći dodatne podatke specifične za domenu.

4.2. FC-CLIP

FC-CLIP (Frozen Convolutional CLIP) je model za segmentaciju s otvorenim vokabularom i izravnom predikcijom primjeraka. Ovaj model je proširenje Mask2Former-a s CLIP okosnicom s ciljem segmentacije s otvorenim vokabularom. Model je osmišljen kako bi prevladao ograničenja postojećih dvostupanjskih pristupa segmentaciji, koji su računalno zahtjevni zbog odvojenih procesa generiranja segmentacijskih mapi i klasifikacije. Izvorni CLIP model uči zajednički ugrađeni prostor za slike i tekst putem kontrastivnog učenja na velikim skupovima podataka koji sadrže parove slika i tekstnih opisa. U ovom prostoru, semantički slične slike i tekstovi nalaze se blizu jedni drugih, omogućujući zero-shot klasifikaciju i pretraživanje u različitim domenama. FC-CLIP se temelji na arhitekturi Mask2Former, koja integrira generiranje segmentacijskih maski i klasifikaciju u jedan korak dijeleći zamrznutu CLIP okosnicu između oba zadatka. Ovakav pristup omogućuje modelu da zadrži unaprijed naučenu usklađenost slika i tekstova u CLIP-u, čime se izbjegavaju problemi s razmještanjem značajki koji se mogu pojaviti prilikom dodatnog učenja CLIP-a [4].

FFC-CLIP nadograđuje temelje postavljene u Mask2Former-u tako što se specifično fokusira na fino strukturirane vizualne značajke, koje su ključne za zadatke poput segmentacije primjeraka. Dok Mask2Former integrira generiranje segmentacijskih maski i klasifikaciju dijeleći zamrznutu CLIP okosnicu između oba zadatka, FC-CLIP dodatno unapređuje ovaj pristup uvodeći mehanizme za usklađivanje višerazinskih značajki s jezičnim konceptima. Umjesto da se oslanja samo na globalne značajke, kao što to čini izvorni CLIP model, FC-CLIP koristi detaljnije vizualne informacije iz različitih razina

značajki, čineći ga učinkovitijim za zadatke na razini piksela, poput segmentacije primjeraka [4].

Snaga FC-CLIP-a leži u njegovoj sposobnosti da zadrži CLIP-ove sposobnosti klasifikacije u zero-shot načinu, dok istovremeno generira precizne mape za segmentaciju. Konvolucijski CLIP kao okosnica općenito pokazuje bolju generalizaciju na većim rezolucijama u usporedbi s CLIP modelima temeljenima na modelima zasnovanima na pažnji, što ga čini izuzetno učinkovitim za guste predikcijske zadatke poput panoptičke, semantičke i segmentacije primjeraka.

4.2.1. Razredno-agnostički generator segmentacijskih mapa

Razredno-agnostički generator segmentacijskih mapa je odgovoran za stvaranje segmentacijskih mapa bez dodjeljivanja specifičnih razreda pikselima. Ova komponenta se zasniva na arhitekturi Mask2Former koja ima dekoder piksela i dekoder mape. Dekoder piksela izlučuje značajke ulazne slike na razini piksela. Te značajke zajedno sa ugrađenim reprezentacijama za detekciju objekata prolaze kroz niz dekodera mape koji se sastoje od mehanizma pažnje, mehanizma maskirane pažnje i potpuno povezanih mreža. Prilikom prolaska kroz dekoder mape, ugrađene reprezentacije za detekciju objekata se ažuriraju, a na kraju procesa matričnim množenjem tih ugrađenih reprezentacija i značajkama izlučenih na razini piksela dobiva se konačna segmentacijska mapa. Te mape su razredno-agnostičke jer pikselima nisu dodijeljeni specifični indeksi razreda, nego predstavljaju skup mogućih mapi objekata koje se kasnije uparuju s istinitim segmentacijskim mapama metodom "Mađarskog uparivanja" [27].

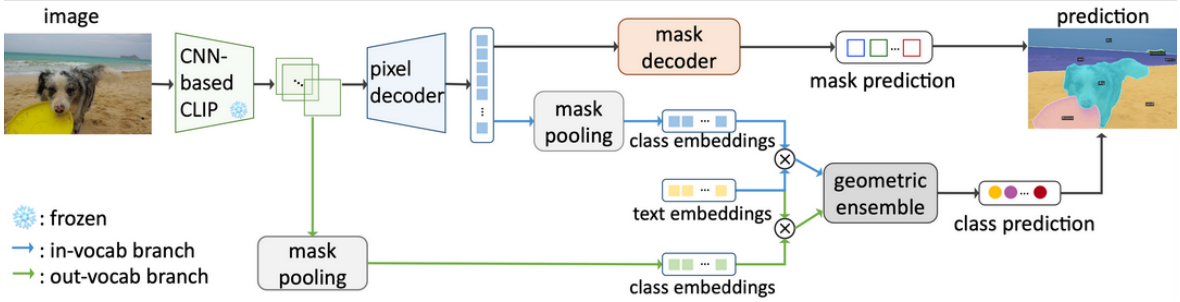
4.2.2. Klasifikator za poznati vokabular

Kada su jednom generirane razredno-agnostičke mape, klasifikator za poznati vokabular dodjeljuje tim mapama razrede na temelju skupa razreda na kojima je model učen. Ovaj klasifikator koristi kontrastivno učenje kako bi mapirao značajke mapi u isti semantički prostor kao i tekstne ugradnje koje proizvodi enkoder teksta CLIP modela. To znači da se za svaku predviđenu mapu njezine značajke sažimaju (pooling) i projiciraju u prostor ugradnji. Klasifikator zatim uspoređuje ugradnje mape s ugradnjama teksta (koje predstavljaju nazive razreda) računajući kosinusnu sličnost. Konačna klasifikacija za svaku

mapu dobiva se primjenom softmax funkcije za normalizaciju predikcija. Važno je napomenuti da su ugradnje teksta unaprijed izračunate koristeći CLIP-ov enkoder teksta i pohranjene u memoriji, što ovaj proces čini vrlo učinkovitijim jer nije potrebno dodatno računanje ugradnji. Ovaj klasifikator je odgovoran za rukovanje klasifikacijom objekata iz poznatog vokabulara, što znači objekata koji pripadaju skupu razreda viđenih tijekom učenja.

4.2.3. Klasifikator za nepoznati vokabular

Klasifikator za nepoznati vokabular uveden je kako bi se riješio izazov prepoznavanja razreda objekata koji nisu bili dio skupa za učenja. Tijekom zaključivanja modela, klasifikator za poznati vokabular neće moći generalizirati na još neviđene razrede, pa stupa na snagu klasifikator za nepoznati vokabular. Ovaj klasifikator ponovno koristi zamrznutu CLIP okosnicu i primjenjuje tehniku sažimanja mapi (mask-poolinga) na značajke izlučene iz te okosnice. Za razliku od klasifikatora za poznati vokabular, koji se oslanja na značajke dekodera piksela, klasifikator za nepoznati vokabular sažima značajke izravno iz zamrznute CLIP okosnice. Korištenjem snažnih sposobnosti CLIP-a za prepoznavanje s otvorenim vokabularom, model može prepoznati razrede objekata iz šireg skupa razreda bez potrebe za dodatnim prilagođavanjem. Kako bi iskoristio predikcije oba klasifikatora (poznatog i nepoznatog vokabulara), FC-CLIP koristi tehniku geometrijskog ansambla. Tijekom ovog procesa, klasifikacijski rezultati oba klasifikatora ansambliraju se na temelju ponderiranog geometrijskog prosjeka, što pomaže uravnotežiti izvedbu modela u prepoznavanju kako poznatih (in-vocabulary), tako i nepoznatih (out-of-vocabulary) razreda. Klasifikator za nepoznati vokabular uveden je kako bi se riješio izazov prepoznavanja objekata koji nisu bili dio skupa za učenje (novi objekti). Tijekom izvođenja modela, klasifikator za poznati vokabular može imati poteškoća s generalizacijom na ove neviđene razrede, pa stupa na snagu klasifikator za nepoznati vokabular kako bi riješio ovaj zadatak.



Slika 4.2. Arhitektura FC-CLIP modela. Zelena grana označava klasifikator za nepoznati vokabular, plava grana klasifikator za poznati vokabular, dok dekoder piksela (eng. pixel decoder, označen plavim okvirom) i dekoder maski (eng. mask decoder, označen narančastim okvirom) čine razredno-agnostički generator segmentacijskih mapa. Zajednička zamrznuta CLIP okosnica (eng. CNN-based CLIP) je označena zelenim okvirom. Preuzeto iz [4]

5. Problem neuravnoteženih razreda

Jedan od značajnih izazova u strojnom učenju je problem neuravnoteženih razreda, koji se javlja kada skup podataka sadrži nejednak broj primjeraka po razredima. Ova neuravnoteženost može ozbiljno utjecati na izvedbu modela, osobito u zadacima klasifikacije, segmentacije i detekcije objekata [28].

5.1. Utjecaj neuravnoteženih razreda na modele strojnog učenja

Kada su razredi neuravnoteženi, model se može nagnuti prema zastupljenijem razredu tijekom učenja. To znači da model može postati pristran prema razredu s više primjera i ignorirati primjere rijetkih razreda. Na primjer, u zadatku klasifikacije slika s neuravnoteženim skupom podataka gdje većina slika pripada jednom razredu, model će vjerojatno imati visoku točnost u predviđanju tog razreda, ali će loše generalizirati nad rijetkim razredima. Ovaj problem se često manifestira kao visoka ukupna točnost, dok su metrike poput preciznosti i odziva za rijetke razrede znatno smanjeni.

5.2. Pristupi problemu neuravnoteženih razreda

Postoji nekoliko strategija koje se mogu primijeniti za ublažavanje problema neuravnoteženih razreda:

- **Otežavanje gubitka**: Jedan od pristupa je dodjeljivanje većih težina rijetkim razredima u funkciji gubitka tijekom učenja. Ova metoda podiže važnost rijetkih razreda, smanjujući pristranost modela prema dominantnim razredima.
- **Neravnomjerno uzorkovanje**: Postoje dvije glavne metode uzorkovanja: nadu-

zorkovanje manjinskih razreda (over sampling) i poduzorkovanje većinskih razreda (under sampling). Preuzorkovanje podrazumijeva repliciranje primjeraka iz rijetkih razreda kako bi se povećala njihova zastupljenost, dok poduzorkovanje smanjuje broj primjeraka iz dominantnih razreda kako bi se postigla ravnoteža.

- **Generativni modeli**: Generativni modeli, poput GAN-ova (Generative Adversarial Networks) [29], mogu se koristiti za generiranje novih umjetnih primjeraka za rijetke razrede, čime se povećava broj primjeraka u tim razredima.

- **Fokalni gubitak**: Fokalni gubitak [30] je modifikacija standardne funkcije gubitka koja je posebno korisna za neuravnotežene razrede. Umanjuje važnost dobro klasificiranih primjeraka i povećava važnost onih koji su teži za klasificirati, što pomaže u boljem učenju rijetkih razreda.

- **Detekcija anomalija**: U ekstremnim slučajevima, kada je jedan razred znatno više zastupljen, zadatak klasifikacije se može preoblikovati u zadatak detekcije anomalija, gdje se rijetki razredi tretiraju kao anomalije koje treba identificirati.

Problem neuravnoteženih razreda često se pojavljuje u aplikacijama poput medicinske dijagnostike, gdje je broj primjeraka zdrave populacije znatno veći od broja primjeraka s određenom bolešću. Također, u autonomnim vozilima, rijetki događaji poput nesreća ili iznenadnih prepreka moraju se precizno detektirati, iako su ti događaji rijetko zastupljeni u podacima za učenje.

Primjenom navedenih strategija, modeli mogu postići bolju generalizacijsku sposobnost, posebno u zadacima gdje je izvedba modela za rijetke razreda kritična.

5.3. Dinamičko određivanje težina gubitka za učenje više zadataka

Učenje više zadataka postavlja izazove u ravnoteži između različitih zadataka, osobito kada su podaci za pojedine zadatke neuravnoteženi. Naime, u kontekstu dubokog učenja, zadaci koji uključuju rijetke razrede mogu dominirati ukupnim gubitkom, što može rezultirati pogoršanjem izvedbi na drugim zadacima [31].

5.3.1. Problem neuravnoteženih razreda u učenju s više zadataka

Kod klasičnog unakrsnog entropijskog gubitka, ukupni gubitak optimizira se kao prosjek po razredima, pri čemu težine ovise o relativnoj učestalosti svakog razreda. U scenarijima s neuravnoteženim razredima, ovaj pristup može dovesti do toga da modeli favoriziraju češće razrede, ignorirajući rijetke razrede koji su ipak ključni za specifične zadatke. Kada se povećava težina rijetkih razreda kako bi se kompenzirala ova neuravnoteženost, može doći do povećanja lažno pozitivnih predikcija, što smanjuje preciznost.

5.3.2. Dinamičko određivanje težine gubitka

Kako bi se adresirali ovi izazovi, dinamičko određivanje težina gubitka uvodi se kao mehanizam koji prilagođava težine gubitka na temelju izvedbe modela tijekom učenja. Konkretno, težine gubitka za svaki razred dinamički se prilagođavaju na temelju odziva tog razreda na validacijskom skupu nakon svake epohe učenja. Na taj način, težina razreda koji već postiže visok odziv smanjuje se, dok se težina razreda s niskim odzivom povećava, čime se osigurava uravnoteženiji doprinos svakog razreda ukupnom gubitku [31]. Matematički, težina gubitka za pojedini razred se definira kao:

$$wR_{c,t} = \frac{N}{N_c}(1 - R_{c,t}) \tag{5.1}$$

gdje je N_c broj primjera u razredu c, a $R_{c,t}$ odziv razreda c nakon epohe $t - 1$. Ovaj pristup omogućuje dinamičko prilagođavanje težina gubitka, ovisno o trenutnim izvedbama modela.

6. Eksperimenti

U ovom poglavlju ću opisati korišteni skup podataka, korištene programske biblioteke, izvedene eksperimente i konačne rezultate. Usredotočio sam se na implementaciju i integraciju modula za dinamičko određivanje težina segmentacijskog gubitka kako bih usporedio segmentacijsku izvedbu osnovnih modela i modela koji koriste razvijeni modul. Također, evaluirao sam model s jezičnim ugrađivanjima na istom skupu podataka kako bih istražio segmentacijsku izvedbu takvog modela bez dodatnog učenja.

6.1. Skup podataka TACO

TACO (Trash Annotations in Context) [2] je javni skup podataka osmišljen za detekciju i segmentaciju otpada u različitim okruženjima. Skup podataka raste zbog stalnih doprinosa korisnika koji pridonose novim slikama i oznakama. TACO je jedinstven u svojoj namjeni jer se fokusira na detekciju otpada u prirodnim okruženjima, poput plaža, gradova i drugih javnih prostora, za razliku od konvencionalnih skupova podataka koji se fokusiraju na čisto vizualno prepoznavanje objekata bez konteksta. TACO sadrži slike visoke rezolucije, uglavnom snimljene mobilnim uređajima. Slike su pohranjene na platformi Flickr, dok server TACO-a upravlja oznakama i periodično prikuplja nove potencijalne slike otpada pretraživanjem interneta. Otpad je označen i segmentiran koristeći hijerarhijsku taksonomiju koja sadrži 60 razreda otpada, grupiranih u 28 nadrazreda. To uključuje i poseban razred "Nejednoznačan otpad" ili "Other" za objekte koji su nejasni ili ne spadaju u ostale definirane razrede. Za razliku od drugih skupova podataka gdje je ključna distinkcija između razreda, TACO se fokusira na detekciju otpada u kontekstu, uzimajući u obzir različite pozadinske elemente i specifične scenarije u kojima se otpad pojavljuje. TACO je vrijedan resurs za istraživače i inženjere koji rade na problemu automatske detekcije otpada, s ciljem stvaranja autonomnih sustava koji mogu pomoći

u smanjenju otpada u okolišu [2].

U ovom radu koristim isti skup podataka TACO, no ograničavam se na 10 razreda. Takav podskup autori rada [2] nazivaju TACO 10. Koristeći programski jezik Python i njegove biblioteke json i Pandas mapiram skup od 60 razreda na skup od 10 razreda: "Can", "Other", "Bottle", "Bottle cap", "Cup", "Lid", "Plastic bag + wrapper", "Pop tab", "Straw" i "Cigarette". Skup za učenje sadrži 1200 slika te 3711 označenih primjeraka. Skup za validaciju sadrži 150 slika i 504 označenih primjeraka. Dok skup za ispitivanje sadrži 150 slika i 569 označenih primjeraka. Razred "Other" služi kao sveobuhvatni razred za sve razrede koji se nisu mogli logički mapirati u ostalih 9 razreda.

6.2. Korištene programske biblioteke

U eksperimentalnom dijelu rada koristio sam niz suvremenih programskih biblioteka i alata koji su omogućili implementaciju i evaluaciju modela za segmentaciju primjeraka. Radno okruženje temeljilo se na radnom okviru Detectron2, izgrađenom na PyTorch-u, koji je služio kao glavna platforma za korištenje modela poput Mask R-CNN, Mask2Former i FC-CLIP. Eksperimente sam provodio na Google Colab platformi, koristeći NVidia A100 grafičku karticu, koja je omogućila visoke performanse potrebne za učenje dubokih neuronskih mreža. Koristio sam radni okvir PyTorch i COCOAPI biblioteku za modifikaciju izvršnog koda navedenih modela i razvoj modula za dinamičko određivanje težina gubitka. Biblioteke json i Pandas poslužile su za manipulacije nad razredima skupa podataka. Korištenje ovih tehnologija omogućilo je uspješno provođenje eksperimentalnog dijela zadatka, od učenja modela do evaluacije njihovih izvedbi u zadatku segmentacije primjeraka.

6.3. Korištena metrika

U zadacima segmentacije primjeraka koristim niz metričkih pokazatelja kako bih procijenio učinkovitost modela. Ove metrike omogućuju detaljnu evaluaciju izvedbe modela na različitim tipovima objekata i razredima, uzimajući u obzir točnost prepoznavanja i segmentacije objekata različitih veličina.

IoU mjera

U kontekstu segmentacije primjeraka, primjer se označava istinito pozitivnom kada predviđena maska objekta ima dovoljno preklapanja s anotiranom maskom objekta prema IoU (Intersection over Union) pragu [17]. IoU je mjera koliko se predviđena maska objekta i anotirana maska objekta preklapaju, a računa se kao:

$$\text{IoU} = \frac{\text{područje preklapanja predviđene i anotirane maske objekta}}{\text{područje unije predviđene i anotirane maske objekta}} \tag{6.1}$$

Da bi se predikcija smatrala točnom (istinito pozitivna), IoU između predviđene i anotirane maske objekta mora premašiti unaprijed definirani prag (npr. 50% ili 75%).

Prosječan odziv

Odziv je mjera koja pokazuje koliko je model uspješan u pronalaženju svih relevantnih objekata u skupu podataka. Odziv kvantificira udio stvarnih primjera (tj. objekata koji su prisutni u anotaciji) koje model uspješno identificira i pravilno segmentira. Izraz za odziv definira se kao:

$$\text{Odziv} = \frac{\text{istinito\_pozitivni}}{\text{istinito\_pozitivni} + \text{lažno\_negativni}} \tag{6.2}$$

gdje istinito_pozitivni označava primjere koje model ispravno identificira i segmentira, dok lažno_negativni označava primjere gdje model ne uspijeva ispravno detektirati ili segmentirati objekt koji je prisutan u anotaciji.

U segmentaciji primjeraka, visok odziv znači da model učinkovito pronalazi većinu ili sve objekte prisutne na slici. Međutim, visok odziv ne znači nužno da su predikcije modela precizne ili točne. Model može imati visok odziv, ali također može proizvesti mnogo lažno pozitivnih predikcija. Iz tog razloga, odziv se tumači zajedno s preciznošću (udio točno predviđenih objekata). Model za segmentaciju s niskim odzivom predstavlja problem jer propušta objekte, što dovodi do nepotpunih detekcija. U primjenama poput medicinske dijagnostike slika (gdje propuštanje tumora može biti opasno po život) ili autonomne vožnje (gdje propuštanje pješaka može uzrokovati nesreće), visoki odziv je ključan.

Dok odziv pruža uvid u sposobnost modela da pronađe objekte pri određenom pragu pouzdanosti i IoU pragu, prosječni odziv (AR) proširuje ovaj koncept mjerenjem odziva modela kroz više pragova pouzdanosti i uprosječivanjem preko raspona IoU pragova. Ovakav pristup omogućuje procjenu izvedbe modela pri različitim razinama sigurnosti predikcija i različitim stupnjevima preklapanja između predikcija i stvarnih objekata. Prosječan odziv je uprosječena vrijednost odziva kroz raspon IoU pragova, obično od 0.50 do 0.95 u koracima od 0.05 (npr. IoU pragovi 0.50, 0.55, 0.60, ..., 0.95). Ovaj pristup omogućuje procjenu izvedbe modela pri blagim (npr. IoU = 0.50) i strogim (npr. IoU = 0.95) pragovima preklapanja. Formulacija prosječnog odziva je sljedeća:

$$\text{AR} = \frac{1}{N} \sum_{i=1}^{N} \text{odziv\_pri\_IoU}_i \tag{6.3}$$

gdje je N broj IoU pragova, a odziv_pri_IoU_i vrijednost odziva izračunata na svakom specifičnom IoU pragu. Model koji se dobro ponaša pri IoU = 0.50, ali loše pri IoU = 0.75 i višim, vjerojatno daje grube segmentacije koje nisu precizne. Model koji održava visok odziv čak i pri strožim IoU pragovima (npr. IoU = 0.90) smatra se da proizvodi preciznije segmentacije.

Prosječna preciznost

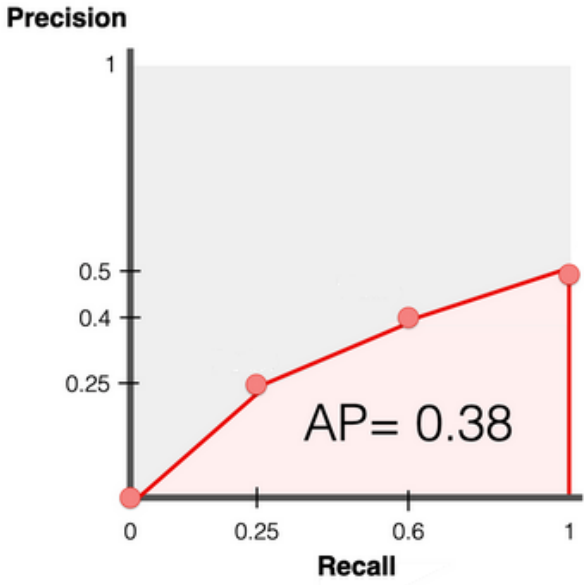
U kontekstu segmentacije primjeraka, preciznost mjeri udio ispravno predviđenih maski objekata (istinito pozitivni) u odnosu na sve pozitivno predviđene maske objekata (uključujući istinito pozitivne i lažno pozitivne predikcije). Izraz za preciznost definira se kao:

$$\text{Preciznost} = \frac{\text{istinito\_pozitivni}}{\text{istinito\_pozitivni} + \text{lažno\_pozitivni}} \tag{6.4}$$

gdje istinito_pozitivni označava primjere koje model ispravno identificira i segmentira, dok lažno_pozitivni označava pogrešno predviđene maske objekata, gdje model predviđa masku koja ili ne odgovara nijednom stvarnom objektu ili se preklapa s anotiranom maskom, ali ne zadovoljava prag.

U segmentaciji primjeraka, procjena izvedbe modela često se oslanja na skup mjernih vrijednosti nazvanih prosječna preciznost (AP). Ove mjere kvantificiraju ravnotežu između preciznosti i odziva za sve pragove pouzdanosti detekcije. AP mjere osmišljene su

kako bi sažele ukupnu kvalitetu segmentacijskog modela uzimajući u obzir koliko objekata propušta (odziv) i koliko su te detekcije točne (preciznost). AP se računa kao površina ispod krivulje preciznosti i odziva dobivene variranjem praga pouzdanosti detekcije od najnižeg do najvišeg mogućeg. U COCO evaluacijskom protokolu, AP se dodatno uprosječava preko više IoU pragova. Konkretno, AP se računa na IoU pragovima od 0.50 do 0.95 u koracima od 0.05, a zatim se rezultati uprosječuju kako bi se dobio konačni AP.



Slika 6.1. Interpretacija prosječne preciznosti kao površina ispod krivulje preciznosti i odziva, preuzeto iz [32]

AP50 je prosječna preciznost izračunata pri IoU pragu od 0.50. To znači da se predviđena segmentacijska maska smatra točnom ako se preklapa sa anotiranom maskom najmanje 50% (IoU >= 0,50). AP50 je relativno blag kriterij jer dopušta djelomično preklapanje između predviđenih i anotiranih maski. Često se koristi kao osnovni pokazatelj može li model približno lokalizirati objekte.

AP75 je prosječna preciznost pri strožem IoU pragu od 0,75. Predviđena maska mora se preklapati sa anotiranom maskom najmanje 75% (IoU >= 0,75) da bi se smatrala ispravnom. AP75 je stroža mjera koja zahtijeva veću preciznost u segmentaciji objekata. Modeli s visokim AP75 rezultatom bolje proizvode precizne i dobro lokalizirane segmentacije koje blisko odgovaraju obliku i veličini stvarnih objekata.

APs procjenjuje prosječnu preciznost za male objekte, gdje se mali objekti definiraju kao oni s površinom manjom od 32^2 piksela. Mali objekti često predstavljaju izazov za

modele segmentacije primjeraka zbog ograničene rezolucije i manje prepoznatljivih značajki. APs odražava koliko dobro model upravlja ovim teškim slučajevima.

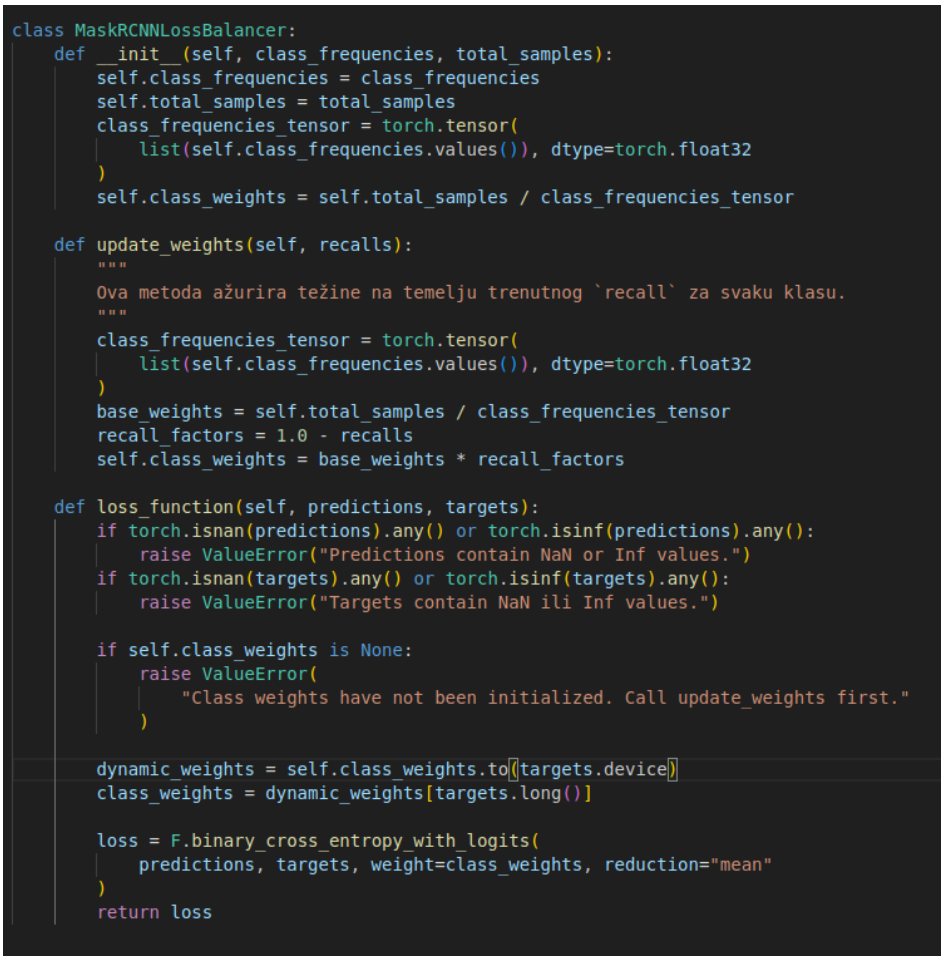
APm je prosječna preciznost za objekte srednje veličine, gdje su srednje veliki objekti definirani kao oni s površinama između 32^2 i 96^2 piksela. APm procjenjuje sposobnost modela da detektira i segmentira objekte srednje veličine, koji su obično lakši za rukovanje od malih objekata, ali mogu i dalje predstavljati izazov u određenim slučajevima.

APl je prosječna preciznost za velike objekte, gdje su veliki objekti definirani kao oni s površinama većim od 96^2 piksela. APl mjeri performanse modela na velikim objektima, koji su obično lakši za detekciju i segmentaciju zbog njihove veličine i jasnijih značajki. Visoki APl rezultati očekuju se jer veći objekti pružaju više piksela za točnu segmentaciju.

6.4. Modul za dinamičko određivanje težina gubitka

Kako bih odredio težine segmentacijskog gubitka ovisno o odzivu za razrede, prvo sam modificirao validacijski proces Detectron2 radnog okvira i implementirao vlastitu petlju za učenje. Nadjačao sam metodu _derive_coco_results razreda COCOEvaluator tako da osim prosječne preciznosti po razredu vraća i prosječni odziv po razredu. Tako nakon svake validacije, metoda inference_on_dataset vraća obje metrike za svaki razred. Nakon dohvaćenih odziva ažuriram težine gubitka za svaki razred i te težine se koriste pri izračunu gubitka do sljedeće validacije kada se one ponovno ažuriraju. Važno je istaknuti kako se modul razlikuje i za Mask R-CNN model i za Mask2Former model jer Mask2Former model dodaje jedan dodatan razred kao token za kraj sekvence. U mojem slučaju to je bio 11. razred kod Mask2Former modela.

Nadjačao sam razred koji definira segmentacijsku glavu Mask R-CNN modela kako bih inicijalizirao modul i podesio izračun gubitka da koristi metodu modula loss_function. Nadjačani razred sam registrirao u Detectron2 radni okvir i podesio konfiguracijske postavke da Detectron2 zaista izgradi model s modificiranom segmentacijskom glavom.



```python
class MaskRCNNLossBalancer:
    def __init__(self, class_frequencies, total_samples):
        self.class_frequencies = class_frequencies
        self.total_samples = total_samples
        class_frequencies_tensor = torch.tensor(
            list(self.class_frequencies.values()), dtype=torch.float32
        )
        self.class_weights = self.total_samples / class_frequencies_tensor

    def update_weights(self, recalls):
        """
        Ova metoda ažurira težine na temelju trenutnog `recall` za svaku klasu.
        """
        class_frequencies_tensor = torch.tensor(
            list(self.class_frequencies.values()), dtype=torch.float32
        )
        base_weights = self.total_samples / class_frequencies_tensor
        recall_factors = 1.0 - recalls
        self.class_weights = base_weights * recall_factors

    def loss_function(self, predictions, targets):
        if torch.isnan(predictions).any() or torch.isinf(predictions).any():
            raise ValueError("Predictions contain NaN or Inf values.")
        if torch.isnan(targets).any() or torch.isinf(targets).any():
            raise ValueError("Targets contain NaN ili Inf values.")

        if self.class_weights is None:
            raise ValueError(
                "Class weights have not been initialized. Call update_weights first."
            )

        dynamic_weights = self.class_weights.to(targets.device)
        class_weights = dynamic_weights[targets.long()]

        loss = F.binary_cross_entropy_with_logits(
            predictions, targets, weight=class_weights, reduction="mean"
        )
        return loss
```

Slika 6.2. Mask R-CNN modul za dinamičko određivanje težina gubitka

Razerd koji definira Mask2Former model koristi SetCriterion razred za izračun gubitaka. Moj pristup je bio nadjačavanje i registriranje modificiranog Mask2Former razreda u Detectron2 radni okvir i konfiguracijsku datoteku kako bih koristio moju verziju SetCriterion razreda. U toj verziji SetCriterion razreda inicijaliziram modul za dinamičko određivanje težina gubitaka i prilagođavam izračun gubitka kako bi koristio metodu modula loss_function.

```python
class Mask2FormerLossBalancer:
    def __init__(self, class_frequencies, total_samples, eos_file=0.1):
        self.class_frequencies = class_frequencies
        self.total_samples = total_samples
        class_frequencies_tensor = torch.tensor(
            list(self.class_frequencies.values()), dtype=torch.float32
        )
        self.class_weights = self.total_samples / class_frequencies_tensor
        self.eos_file = eos_file
        self.class_weights = torch.cat(
            [self.class_weights, torch.tensor([self.eos_file])]
        )

    def update_weights(self, recalls):
        """
        Ova metoda ažurira težine na temelju trenutnog `recall` za svaku klasu.
        """
        recall_factors = 1.0 - recalls
        class_weights = (
            self.class_weights.clone()
        )  # Clone to avoid modifying the original tensor.
        class_weights[
            :-1
        ] *= recall_factors  # Apply the recall factor only to object classes.
        self.class_weights = class_weights

    def loss_function(self, predictions, targets):
        if torch.isnan(predictions).any() or torch.isinf(predictions).any():
            raise ValueError("Predictions contain NaN or Inf values.")
        if torch.isnan(targets).any() or torch.isinf(targets).any():
            raise ValueError("Targets contain NaN or Inf values.")

        dynamic_weights = self.class_weights
        dynamic_weights = dynamic_weights.to(targets.device)
        class_weights = dynamic_weights[targets.long()]

        loss = F.binary_cross_entropy_with_logits(
            predictions, targets, weight=class_weights, reduction="none"
        )
        return loss
```

Slika 6.3. Mask2Former modul za dinamičko određivanje težina gubitka

6.5. Eksperiment s osnovnim modelima

Kako bih postavio bazne rezultate na temelju kojih mogu usporediti rezultate modificiranih modela, naučio sam Mask R-CNN i Mask2Former modele na skupu podataka TACO 10. Oba modela su prednaučena na skupu podataka COCO [17], što im je omogućilo dobru početnu točku za daljnje učenje specifičnih zadataka segmentacije primjeraka otpada na skupu podataka TACO 10. U eksperimentima sam odlučio ne provoditi transformaciju slika, kao što je to učinjeno u originalnom TACO radu [2], kako bih osigurao da se modeli uspoređuju isključivo na temelju utjecaja modifikacije segmentacijskog gubitka, a ne na temelju dodatnih faktora kao što su promjene u predobradi podataka. Na ovaj način, postavljeni bazni rezultati omogućuju jasno razumijevanje učinka modifikacije primijenjene na modele tijekom daljnjih eksperimenata.

Eksperiment s Mask R-CNN modelom

Veličina grupe je postavljena na 2, u skladu s originalnim Mask R-CNN radom [1], što znači da se dvije slike koriste u svakoj grupi tijekom učenja. Stopa učenja određena je na 0.001, što definira korak kojim model uči tijekom optimizacije. Prateći izvedbu na skupu podataka za validaciju odredio sam da učenje na 60 epoha daje najbolje rezultate na skupu podataka za ispitivanje. Detectron2 ne pruža mogućnost direktno odrediti broj epoha, već koristi broj iteracija. Kako bih dobio 60 epoha postavio sam broj iteracija na 36000. Broj iteracija dobijemo tako da pomnožimo broj željenih epoha sa brojem slika u skupu podataka za učenje te podijelimo sa veličinom grupe. Broj uzoraka koje ROI glava [1] obrađuje po slici postavljen je na 128, dok je broj razreda koje model treba prepoznati postavljen na 10, što odgovara broju razreda u skupu podataka.

Metrika	**AP**	**AP50**	**AP75**	**APs**	**APm**	**APl**
Rezultat	22.092	33.583	22.897	1.821	19.187	32.699

Tablica 6.1. Evaluacija Mask R-CNN modela.

Razred	**AP**
Can	41.577
Other	25.966
Bottle	48.423
Bottle cap	24.805
Cup	18.575
Lid	14.752
Plastic bag + wrapper	28.222
Pop tab	8.482
Straw	3.775
Cigarette	6.346

Tablica 6.2. AP po razredu za Mask R-CNN model.

Eksperiment s Mask2Former modelom

Broj razreda za segmentacijsku glavu modela postavljen je na 10, što odgovara broju razreda u skupu podataka. Korišteno je rezanje gradijenata tipa "value" kako bi se kontrolirala veličina gradijenata tijekom učenja, s maksimalnom vrijednošću postavljenom na 1. Veličinu grupe sam postavio na 2, što znači da se dvije slike obrađuju u svakoj grupi, kako bih održao konzistentnost eksperimentalnih postavka. Stopa učenja postavljena je na 0.0001, što definira sporiji korak učenja modela tijekom optimizacije no korišten je u originalnom radu [5]. Praćenjem izvedbe na skupu za validaciju, metrika je počela opadati nakon 16. epohe pa sam odredio da broj epoha bude 16. U ovom eksperimentu opravdavam učenje modela kroz manji broj epoha jer u originalnom radu [5] autori naglašavaju kako Mask2Former konvergira 8 puta brže od Mask R-CNN modela na skupu podataka COCO.

Metrika	**AP**	**AP50**	**AP75**	**APs**	**APm**	**APl**
Rezultat	3.193	5.703	2.939	0.779	1.585	3.943

Tablica 6.3. Evaluacija Mask2Former modela.

Razred	**AP**
Can	6.081
Other	2.678
Bottle	6.434
Bottle cap	4.466
Cup	0.483
Lid	6.960
Plastic bag + wrapper	3.927
Pop tab	0.000
Straw	0.047
Cigarette	0.854

Tablica 6.4. AP po razredu za Mask2Former model.

6.6. Eksperimenti s modificiranim modelima

Pri inicijalizaciji modula za dinamičko određivanje težina gubitaka, u konstruktor modula sam predao rječnik razreda i njihovih ponavljanja. Odnosno, koliko primjeraka nekog razreda se nalazi u skupu za učenje. Pošto se težina gubitka razreda s niskim odzivom povećava, dok se težina gubitka razreda s visokim odzivom smanjuje, utežavanje svakog razreda ovisno o frekvenciji pojavljivanje osigurava uravnoteženiji doprinos ukupnom gubitku. Ovaj pristup nastoji poboljšati preciznost i ravnotežu modela u zadacima segmentacije. Parametre i hiperparametre oba modela postavio sam iste kao i kod eksperimenata s osnovnim modelima kako bih očuvao konzistentnost eksperimenata. Evaluacija i ažuriranje težina gubitaka su vršeni nakon svake epohe.

Eksperiment s modificiranim Mask R-CNN modelom

Metrika	**AP**	**AP50**	**AP75**	**APs**	**APm**	**APl**
Rezultat	19.974	31.243	20.612	1.036	11.109	26.342

Tablica 6.5. Evaluacija modificiranog Mask R-CNN modela.

Razred	**AP**
Can	40.572
Other	25.082
Bottle	45.125
Bottle cap	23.801
Cup	16.935
Lid	9.020
Plastic bag + wrapper	27.695
Pop tab	1.617
Straw	4.366
Cigarette	5.528

Tablica 6.6. AP po razredu za modificirani Mask R-CNN model.

Eksperiment s modificiranim Mask2Former modelom

Metrika	AP	AP50	AP75	APs	APm	APl
Rezultat	3.961	6.574	3.680	0.343	2.480	9.299

Tablica 6.7. Evaluacija modificiranog Mask2Former modela.

Razred	AP
Can	12.113
Other	3.622
Bottle	4.924
Bottle cap	2.707
Cup	1.046
Lid	1.772
Plastic bag + wrapper	4.752
Pop tab	8.130
Straw	0.227
Cigarette	0.317

Tablica 6.8. AP po razredu za modificirani Mask2Former model.

6.7. Eksperimenti s modelom s jezičnim ugrađivanjima

U ovom poglavlju prikazujem rezultate eksperimenata s FC-CLIP modelom, koji koristi kombinaciju vizualnih i tekstnih informacija za zadatak panoptičke. Proveo sam eksperimente zaključivanja u zadatku segmentacije primjeraka bez dodatnog učenja FC-CLIP modela prenaučenog na LAION [33] skupu podataka. FC-CLIP koristi CLIP model zasnovan na ConvNext modelu [34] kao okosnicu. Koristio sam Detectron2 radni okvir za evaluaciju FC-CLIP-a na TACO 10 skupu podataka za ispitivanje.

Pošto sam eksperimentirao s evaluacijom na zadatku segmentacije primjeraka bez dodatnog učenja, morao sam promijeniti konfiguracijske postavke modela i prilagoditi tekstne upite modela. FC-CLIP je dizajniran za zadatak panoptičke segmentacije, no panoptička segmentacija je kombinacija semantičke segmentacije i segmentacije primjeraka pa sam u konfiguracijskoj datoteci modela postavio zastavicu za segmentaciju primjeraka na True, a zastavice za semantičku i panoptičku segmentaciju na False. Ovo je

omogućilo da model vraća samo rezultate evaluacije za segmentaciju primjeraka.

Kako radim sa skupom podataka za detekciju otpada u kontekstu prirodnog okruženja, promijenio sam tekstne upite u skladu sa zahtjevima skupa podataka da čim više iskoristim prednaučenu semantičku moć CLIP okosnice modela.

```
PROMPTS = [
    "A discarded {} found on the ground.",
    "A {} lying among scattered trash.",
    "A {} left as litter in the street.",
    "A photo of a {} thrown away in a public space.",
    "A crumpled {} lying near other garbage.",
    "A {} mixed with other debris on the ground.",
    "A broken {} discarded in the environment.",
    "This is a {} among other littered objects.",
    "A small {} found discarded in a cluttered area.",
    "A {} partially covered by other trash in the scene.",
    "A large {} lying near a pile of garbage.",
    "A weathered {} discarded on the side of the road.",
    "A crushed {} among other waste on the street.",
    "A {} that has been thrown away and left as litter.",
    "A {} carelessly discarded in a public park.",
    "A {} found among a variety of litter in this scene.",
    "A {} thrown into the corner of a littered area.",
    "This is a {} abandoned as trash in the environment.",
    "A {} lying in a heap of litter, partially hidden.",
    "A photo of a {} found on the sidewalk among other debris."
]
```

Slika 6.4. Promijenjeni tekstni upiti

Metrika	**AP**	**AP50**	**AP75**	**APs**	**APm**	**APl**
Rezultat	5.715	9.617	5.103	0.000	10.538	8.537

Tablica 6.9. Evaluacija FC-CLIP modela.

Razred	**AP**
Can	3.731
Other	1.152
Bottle	26.329
Bottle cap	1.381
Cup	17.556
Lid	0.936
Plastic bag + wrapper	4.904
Pop tab	0.000
Straw	0.073
Cigarette	1.089

Tablica 6.10. AP po razredu za FC-CLIP model.

7. Zaključak

U ovom radu istražio sam različite pristupe segmentaciji primjeraka s posebnim naglaskom na upotrebu modernih dubokih neuronskih mreža i modela s jezičnim ugrađivanjem. Kroz eksperimentalnu evaluaciju, usporedio sam izvedbu klasičnih modela segmentacije, poput Mask R-CNN, s novijim pristupima kao što su Mask2Former i modeli s jezičnim ugrađivanjima poput CLIP-a i FC-CLIP-a. Također, istražio sam utjecaj dinamičkog određivanja težina segmentacijskog gubitka na segmentacijsku izvedbu Mask R-CNN i Mask2Former modela.

Mask R-CNN, sa svojom mrežom za prijedlog regija (RPN), pokazao se kao najpouzdaniji model, postigavši najvišu ukupnu prosječnu preciznost i demonstrirajući snažnu izvedbu za sve veličine objekata i sve razrede. Međutim, uvođenje dinamičkog određivanja težina segmentacijskog gubitka kod Mask R-CNN-a dovelo je do pada u izvedbi, osobito za srednje i velike objekte, jer je model previše davao prednost razredima s nižim odzivom na štetu ostalih razreda. Mask2Former, koji koristi arhitekturu temeljenu na modelu zasnovanom na pažnji, u osnovnom obliku je postigao slabije rezultate od Mask R-CNN-a, ali je pokazao poboljšanje s dinamičkim određivanjem težina segmentacijskog gubitka. Modificirani Mask2Former postigao je višu prosječnu preciznost, osobito za velike objekte, ali i dalje postiže nezadovoljavajuće rezultate u segmentaciji malih objekata. Mask2Former koristi modele zasnovane na pažnji koji mogu iskoristiti globalni kontekst cijele slike. Dinamičko određivanje težina gubitka omogućilo je modelu da usmjeri mehanizam pažnje na razrede s lošijim odzivom što je poboljšalo segmentaciju velikih objekata.

FC-CLIP, model s jezičnim ugrađivanjima za panoptičku segmentaciju koji koristi prednaučeni CLIP model kao okosnicu, pokazao je iznenađujuću snagu za određene razrede, poput boce (Bottle) i šalice (Cup), te je nadmašio Mask2Former na srednjim i velikim

objektima. Međutim, njegova nesposobnost segmentiranja malih objekata i oslanjanje na opće semantičke značajke ograničili su njegovu izvedbu u zadacima precizne segmentacije primjeraka.

Rezultati ovog rada otvaraju nekoliko zanimljivih smjerova za buduća istraživanja u području segmentacije primjeraka. Prvi potencijalni smjer je daljnje istraživanje i optimizacija dinamičkog određivanja težina gubitka. Budući radovi mogli bi se usredotočiti na prilagođavanje mehanizama određivanja težina tijekom učenja kako bi se spriječio pad izvedbe kod dobro uočenih razreda, što je zabilježeno kod Mask R-CNN-a. Daljnji pravac istraživanja odnosi se na kombiniranje vizualnih i tekstnih informacija kao što je to implementirano u FC-CLIP modelu. Iako je FC-CLIP pokazao potencijal u segmentaciji primjeraka bez učenja, njegov pristup mogao bi se poboljšati dodatnim učenjem i prilagodbom tekstnih ugradnji za specifične zadatke segmentacije.

Literatura

[1] P. D. R. G. Kaiming He, Georgia Gkioxari, “Mask r-cnn”, *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017.

[2] P. S. Pedro F Proença, “Taco: Trash annotations in context for litter detection”, *arXiv:2003.06975v2*, 2020.

[3] C. H. A. R. G. G. S. A. G. S. A. A. P. M. J. C. G. K. I. S. Alec Radford, Jong Wook Kim, “Learning transferable visual models from natural language supervision”, *2021 IEEE International Conference on Machine Learning (ICML)*, 2021.

[4] X. D. X. S. L.-C. C. Qihang Yu, Ju He, “Convolutions die hard: Open-vocabulary segmentation with single frozen convolutional clip”, *37th Conference on Neural Information Processing Systems (NeurIPS 2023)*, 2023.

[5] A. G. S. A. K.-R. G. Bowen Cheng, Ishan Misra, “Masked-attention mask transformer for universal image segmentation”, *2022 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[6] Https://pub.towardsai.net/machine-learning-23997460cbc4.

[7] T. D. Jonathan Long, Evan Shelhamer, “Fully convolutional networks for semantic segmentation”, *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[8] R. Girshick, “Fast r-cnn”, *2015 IEEE International Conference on Computer Vision (ICCV)*, 2015.

[9] Https://blog.roboflow.com/difference-semantic-segmentation-instance-segmentation/.

[10] Https://www.v7labs.com/blog/panoptic-segmentation-guide.

[11] Y. L. Y. Bengio, "Convolutional networks for images, speech, and time-series", *1998 Conference: The Handbook of Brain Theory and Neural Networks*, 1998.

[12] T. D. J. M. Ross Girshick, Jeff Donahue, "Rich feature hierarchies for accurate object detection and semantic segmentation", *2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[13] R. G. J. S. Shaoqing Ren, Kaiming He, "Faster r-cnn: Towards real-time object detection with region proposal networks", *2016 IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2016.

[14] G. P. F. S.-P. W. H. A. Liang-Chieh Chen, Alexander Hermans, "Masklab: Instance segmentation by refining object detection with semantic and direction features", *2018 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[15] Y. G. C. H. X. W. Zhaojin Huang, Lichao Huang, "Mask scoring r-cnn", *2019 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[16] S. Y. Min Lin, Qiang Chen, "Network in network", *International Conference on Learning Representations (ICLR 2014)*, 2014.

[17] S. B. L. B. R. G. J. H. P. P. D. R. C. L. Z. P. D. Tsung-Yi Lin, Michael Maire, "Microsoft coco: Common objects in context", *Computer Vision – ECCV 2014*, 2014.

[18] G. S. N. U. A. K. S. Z. Nicolas Carion, Francisco Massa, "End-to-end object detection with transformers", *Computer Vision – ECCV 2020*, 2020.

[19] R. G. C. R. P. D. Alexander Kirillov, Kaiming He, "Panoptic segmentation", *2019 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[20] X. P. T. X. S. F. A. B. A. T. Bolei Zhou, Hang Zhao, "Semantic understanding of scenes through the ade20k dataset", *International Journal of Computer Vision (ICJV 2016)*, 2016.

[21] S. R. e. a. Marius Cordts, Mohamed Omran, “The cityscapes dataset for semantic urban scene understanding”, *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[22] M. N. G. H. Ting Chen, Simon Kornblith, “A simple framework for contrastive learning of visual representations”, *2020 IEEE International Conference on Machine Learning (ICML)*, 2020.

[23] S. R. T. R. M. E. R. B. U. F. S. R. B. S. Marius Cordts, Mohamed Omran, “An image is worth 16x16 words: Transformers for image recognition at scale”, *International Conference on Learning Representations (ICLR 2020)*, 2020.

[24] S. R. J. S. Kaiming He, Xiangyu Zhang, “Deep residual learning for image recognition”, *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[25] N. P. J. U. L. J. A. N. G. L. K. I. P. Ashish Vaswani, Noam Shazeer, “Attention is all you need”, *Advances in Neural Information Processing Systems 30 (NIPS 2017)*, 2017.

[26] K. Sohn, “Improved deep metric learning with multi-class n-pair loss objective”, *Advances in Neural Information Processing Systems 29 (NIPS 2016)*, 2016.

[27] H. Kuhn, “The hungarian method for the assignment problem”, *Naval Research Logistics Quarterly 1955*, 1955.

[28] N. Z. Lian Yu, “Survey of imbalanced data methodologies”, *arXiv:2104.02240*, 2021.

[29] M. M. B. X. D. W.-F. S. O. A. C. Y. B. Ian J. Goodfellow, Jean Pouget-Abadie, “Generative adversarial networks”, *Advances in Neural Information Processing Systems 3(11) (NIPS 2014)*, 2014.

[30] R. G. K. H. P. D. Tsung-Yi Lin, Priya Goyal, “Focal loss for dense object detection”, *2018 IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2018.

[31] S. Marin Kačan, Marko Ševrović, “Dynamic loss balancing and sequential enhancement for road-safety assessment and traffic scene classification”, *arXiv:2211.04165*,

2022.

[32] Https://www.evidentlyai.com/ranking-metrics/mean-average-precision-map.

[33] R. V. C. G. R. W. M. C.-T. C. A. K. C. M. M. W. P. S. S. K. K. C. L. S. R. K. J. J. Christoph Schuhmann, Romain Beaumont, "Laion-5b: An open large-scale dataset for training next generation image-text models", *36th Conference on Neural Information Processing Systems (NeurIPS 2022)*, 2022.

[34] C.-Y. W. C. F. T. D. S. X. Zhuang Liu, Hanzi Mao, "A convnet for the 2020s", *2022 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022.

Sažetak

Segmentacija Primjeraka nad Neuravnoteženim Taksonomijama

Rej Šafranko

Ovaj diplomski rad bavi se istraživanjem različitih pristupa segmentaciji primjeraka u računalnom vidu, s posebnim naglaskom na konvencionalne modele dubokog učenja i suvremene modele s jezičnim ugrađivanjima. U prvom dijelu rada, provedena je detaljna evaluacija klasičnih modela kao što su Mask R-CNN i Mask2Former, koji su naučeni na skupu podataka TACO za segmentaciju otpada. Rezultati pokazuju da Mask R-CNN pruža solidne rezultate, dok Mask2Former nailazi na poteškoće s kontekstualnim razumijevanjem malih i vizualno sličnih objekata otpada. U drugom dijelu rada, istraženi su modeli s jezičnim ugrađivanjima CLIP i FC-CLIP, koji kombiniraju vizualne i tekstne informacije za zero-shot klasifikaciju i segmentaciju primjeraka. FC-CLIP koristi CLIP model kao okosnicu i pokazao je zanimljive rezultate u segmentaciji primjeraka, što sugerira potrebu za daljnjim prilagodbama i istraživanjem. Rad pruža uvid u prednosti i nedostatke različitih modela segmentacije, te ističe važnost prilagodbe modela specifičnim zadacima i skupovima podataka.

Ključne riječi: računalni vid; segmentacija primjeraka; duboko učenje; multimodalni modeli; Mask R-CNN; Mask2Former; TACO; dinamički gubitak; CLIP; FC-CLIP

Abstract

Unbalanced Taxonomy Instance Segmentation

Rej Šafranko

This thesis explores different approaches to instance segmentation in computer vision, with a special focus on conventional deep learning models and contemporary models which use language embeddings. In the first part of the thesis, a detailed evaluation of classical models such as Mask R-CNN and Mask2Former, trained on the TACO dataset for waste segmentation, was conducted. The results show that Mask R-CNN provides solid performance, while Mask2Former encounters challenges with contextual understanding of small and visually similar waste objects. In the second part of the thesis, models utilizing CLIP and FC-CLIP language embeddings were investigated, which combine visual and textual information for zero-shot classification and instance segmentation. FC-CLIP uses the CLIP model as its backbone and demonstrated interesting results in instance segmentation, suggesting the need for further adjustments and exploration. The thesis offers insights into the advantages and disadvantages of different segmentation models and highlights the importance of tailoring models to specific tasks and datasets.

Keywords: Computer Vision; Instance Segmentation; Deep Learning; Multimodal Models; Mask R-CNN; Mask2Former; TACO; Dynamic Loss; CLIP; FC-CLIP