SVEUČILIŠTE U ZAGREBU
**FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA**

DIPLOMSKI RAD br. 199

# UČENJE SEMANTIČKE SEGMENTACIJE NA NEPOTPUNIM OZNAKAMA

Petar Borko

Zagreb, lipanj 2023.

SVEUČILIŠTE U ZAGREBU
**FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA**

DIPLOMSKI RAD br. 199

# UČENJE SEMANTIČKE SEGMENTACIJE NA NEPOTPUNIM OZNAKAMA

Petar Borko

Zagreb, lipanj 2023.

**SVEUČILIŠTE U ZAGREBU**
**FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA**

Zagreb, 10. ožujka 2023.

# DIPLOMSKI ZADATAK br. 199

| | |
|---|---|
| Pristupnik: | **Petar Borko (0036514806)** |
| Studij: | Računarstvo |
| Profil: | Računarska znanost |
| Mentor: | prof. dr. sc. Siniša Šegvić |

Zadatak: **Učenje semantičke segmentacije na nepotpunim oznakama**

Opis zadatka:

Semantička segmentacija važan je zadatak računalnog vida s mnogim zanimljivim primjenama. U posljednje vrijeme vrlo zanimljive rezultate postižu duboki modeli koji znatan dio zaključivanja provode na poduzorkovanoj reprezentaciji. Međutim, standardno nadzirano učenje gustih modela zahtijeva ogromne količine označenih podataka koje nije ni jednostavno ni jeftino pripremiti. Zbog toga razmatramo oportunističke metode označavanja kod kojih semantičke oznake nisu dostupne u većini piksela. U okviru rada, potrebno je odabrati okvir za automatsku diferencijaciju te upoznati biblioteke za rukovanje matricama i slikama. Proučiti i ukratko opisati postojeće segmentacijske arhitekture utemeljene na konvolucijama i pažnji. Osmisliti praktični postupak za učenje segmentacijskog modela na slikama s nepotpunim oznakama. Uhodati postupke učenja modela te validiranje hiperparametara. Vrednovati naučene modele te prikazati i ocijeniti postignutu točnost. Radu priložiti izvorni i izvršni kod razvijenih postupaka, ispitne slijedove i rezultate, uz potrebna objašnjenja i dokumentaciju. Citirati korištenu literaturu i navesti dobivenu pomoć.

Rok za predaju rada: 23. lipnja 2023.

*Zahvaljujem se obitelji i prijateljima na podršci prilikom studiranja, te mentoru, prof. Siniši Šegviću, na strpljenju, znanju i pomoći u pisanju ovog rada.*

## Sadržaj

Uvod ..... 1

1. Teme dubokog učenja ..... 2

1.1. Konvolucijske mreže ..... 2

1.2. Semantička segmentacija ..... 4

1.3. Polunadzirano učenje ..... 7

1.4. Učenje nad pseudooznakama ..... 8

1.5. OpenCLIP ..... 10

1.6. Segment Anything (SAM) ..... 12

2. Korišteni model i skup podataka ..... 15

2.1. OpenSAM ..... 15

2.2. Skup podataka „Gušteri“ ..... 17

3. Eksperimenti ..... 20

3.1. Učinak SAM-a na OpenCLIP ..... 21

3.2. Zagrijavanje modela ..... 22

3.3. Treniranje modela nad pseudooznakama ..... 24

Zaključak ..... 30

Literatura ..... 31

Sažetak ..... 34

Summary ..... 35

# Uvod

Semantička segmentacija je jedan od glavnih zadataka računalnog vida, te je postao iznimno značajan u posljednjih desetak godina. Stalna potreba za poboljšanjem preciznosti i efikasnosti rezultirala je usponom dubokih modela za ovaj zadatak. Takvi modeli se uglavnom oslanjaju na veliku količinu označenih podataka za učenje, što često nije održivo s obzirom na vremenske i financijske troškove potrebne za njihovu pripremu, posebice u semantičkoj segmentaciji koja zahtjeva gustu predikciju. Mnogi dostupni skupovi podataka sadrže red veličine nekoliko desetaka tisuća označenih slika što je znatno manje od skupova iz drugih domena strojnog učenja. Iz tog razloga nam je cilj usmjeriti se na razmatranje alternativnih metoda za generalizaciju nad slabo označenim skupom podataka.

Cilj ovog diplomskog rada je osmisliti praktičan postupak za učenje semantičkog segmentacijskog modela na manjem skupu podataka gdje većina podataka nije označeno koristeći moderne dostupne modele. Da bi se to postiglo, bit će potrebno koristiti okvir za automatsku diferencijaciju i biblioteke za rukovanje matricama i slikama. Razmotriti će se kombinacija dva nova modela visokog kapaciteta, OpenCLIP i SAM, te se pokušati njihovim prijenosom domene naučiti distribuciju oznaka neviđenih klasa. Analizirati će se njihova *zero-shot* performansa na neviđenom skupu te će se ona nastojati poboljšati tehnikama polunadziranog dubokog učenja temeljene na pseudooznakama i učitelj-student ansamblima.

# 1. Teme dubokog učenja

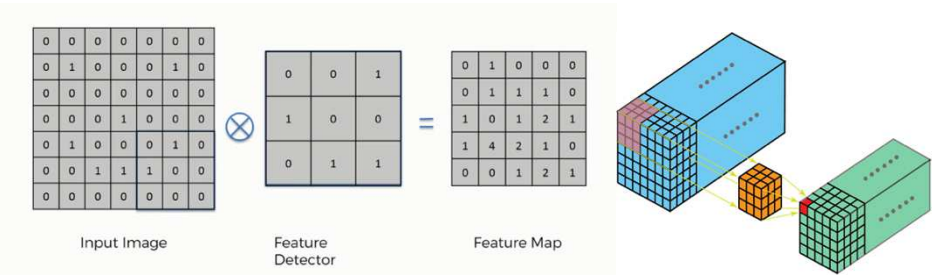
## 1.1. Konvolucijske mreže

**Duboko učenje** je grana strojnog učenja koja razmatra modele izražene slijedom naučenih nelinearnih transformacija. Umjetne neuronske mreže sastoje se od međusobno povezanih slojeva čvorova ili **neurona**, od kojih svaki prima ulaz od prethodnih slojeva, obrađuje ga i prosljeđuje izlaz sljedećim slojevima. Učenje se u tom kontekstu odnosi na prilagođavanje težina između tih neurona dok mreža uči iz gradijenata izračunatih tijekom treniranja.

**Konvolucijski duboki modeli** jedan su od najuspješnijih pristupa dubokog učenja za vizualne i auditivne domene. Konvolucijski modeli zasnivaju se na konvolucijama s naučenim parametrima:

$$s(t) = (x * w)(t)$$

Operatorom konvolucije (slika 1.1) vrši se proizvoljno-dimenzionalni skalarni produkt pomicanjem filtra $\boldsymbol{w}$ po ulazu $\boldsymbol{x}$ [14]. Takvom strukturom je operator specijaliziran za domene mrežne topologije koje imaju lokalne karakteristike (primjerice slika ili zvuk), te omogućava sažimanje njihovih informacija u latentni prostor. Jedna operacija konvolucije sa $n$ filtara predstavlja konvolucijski sloj. Ulančavanjem konvolucijskih slojeva te slojeva sažimanja, normalizacije i dr., dobivamo konvolucijsku neuronsku mrežu. Filtri $\boldsymbol{w}$ svih konvolucijskih slojeva predstavljaju parametre konvolucije koje je cilj naučiti da minimizira gubitak za dani zadatak. Ulančavanjem konvolucijskih slojeva postiže se detekcija hijerarhijske strukture ulaza [15]. Time je omogućena invarijantnost i ekvivarijantnost na translaciju i rotaciju između ulazne i latentne domene, što omogućava da modeli usvoje induktivnu pristranost ovisno o zadatku koji obavljaju.




Slika 1.1 Operator konvolucije [32, 33]

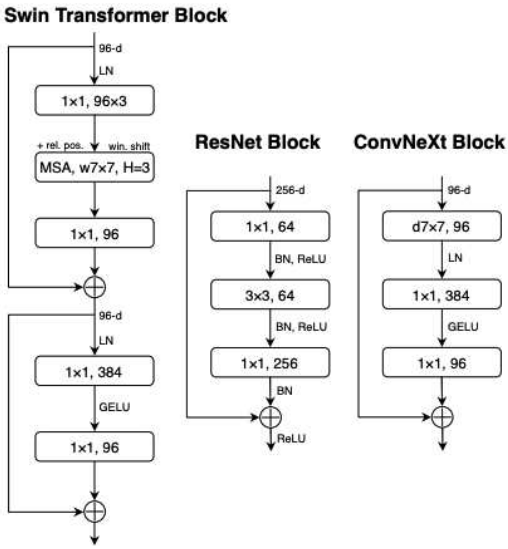
Konvolucijske neuronske mreže znatno su napredovale u zadnjih desetak godina. Razvojem modela poput AlexNet [14] pokazana je visoka moć kapaciteta dubokih modela. Danji napredak postigao je VGG [16] model koji je uz pomoć dublje mreže i manjih konvolucijskih filtera poboljšao performanse AlexNeta. 2015. godine predstavljen je ResNet [17] koji je uvodom preskočnih veza riješio problem degradacije mreže koji se pojavljivao pri velikim dubinama. DenseNet [18] nastavlja tu ideju tako da spaja n-ti sloj sa svakim prethodnim.

Još jednu prekretnicu postigao **vizualni transformer** (ViT) [19] korištenjem arhitekture transformera, familije modela koja se do tada koristila primarno u obradi jezika. Za razliku od CNN-ova gdje je induktivna pristranost lokalnosti, dvodimenzionalne strukture susjedstva i translacijske ekvivarijantnosti prisutna u svim slojevima, vizualnim transformerima su lokalnost i translacijska ekvivarijantnost prisutni isključivo u slojevima višeslojnih perceptrona, dok su mehanizmi pažnje globalni te je sva informacija dvodimenzionalne strukture susjedstva sadržana u pozicijskim ugrađivanjima. Unatoč tom nedostatku, uz dovoljno veliku količinu podataka i vrijeme treniranja ViT pokazuje rezultate konkurentne CNN-u. Danji napredak postignut je sa Swin Transformerom [20] koji je sposoban uhvatiti lokalne i globalne međuodnose u slikama uz pomoć razdvajanja pažnje između lokalnih regija.

Konačno, **ConvNeXt** [21] je moderni konvolucijski model koji koristi i neke tehnike koje su uvedene za modele utemeljene na transformerima. Njegov razvoj je motiviran prividnim prevladavanjem transformera u području računalnog vida te pretpostavlja da konvolucijske mreže mogu postići jednake ili bolje rezultate moderniziranjem klasičnog ResNeta uz tehnike modernijih modela poput Swin Transformera. Neke od korištenih unaprijeđenja uključuju:

- mijenjanje omjera računalne složenosti među stadijima sa ResNet-ovih (3, 4, 6, 3) na (3, 3, 9, 3), gdje se stadiji definiraju kao cjelina građena od više slojeva istog oblika.
- promjena ulazne konvolucije s jezgrom 7x7 i korakom 2 u konvoluciju s jezgrom 4x4 i korakom 4, uklanjajući preklapanja što je specifičnost ViT arhitektura
- upotreba grupne konvolucije, nalik ResNeXt-u [22], ali na razini jednog kanala (engl. *depthwise convolution*). Razdvajanje dubinske i prostorne domene je također naslijeđeno svojstvo transformera. Iako grupna konvolucija smanjuje preciznost, ona je nadoknađena povećavanjem dubine.

- upotreba invertiranog uskog grla, odnosno provođenje konvolucije s grupom 1 nad tenzorom s povećanim brojem kanala, popularizirao ga je MobileNetV2 [23]
- povećanje veličine jezgre na 7x7, pomaže pri raspoznavanju globalnih uzoraka, što je srodno načinu kako to rade transformatori




Slika 1.2 Usporedba organizacije konvolucijskih jedinica ResNeta, Swin Transformera i ConvNeXt-a. Primjećujemo upotrebu invertiranog uskog grla i dubinske konvolucije na ConvNeXt bloku. [21]

Očuvanjem konvolucijske paradigme i primjenom elemenata vizualnih transformera, ConvNext uspijeva konkurirati performansama vizualnih transformera, te će se on koristiti u ovom radu.

## 1.2. Semantička segmentacija

**Semantička segmentacija** je zadatak računalnog vida sa ciljem pružanja semantičke predikcije svakom pikselu na slici. Ovaj zadatak ne razmatra samo pitanje određivanja lokacije objekta unutar slike, već i prepoznavanja različitih dijelova istog objekta. U kontekstu autonomnih vozila, semantička segmentacija bi omogućila prepoznavanje ceste, automobila, pješaka, semafora i drugih relevantnih elemenata unutar jedne slike.

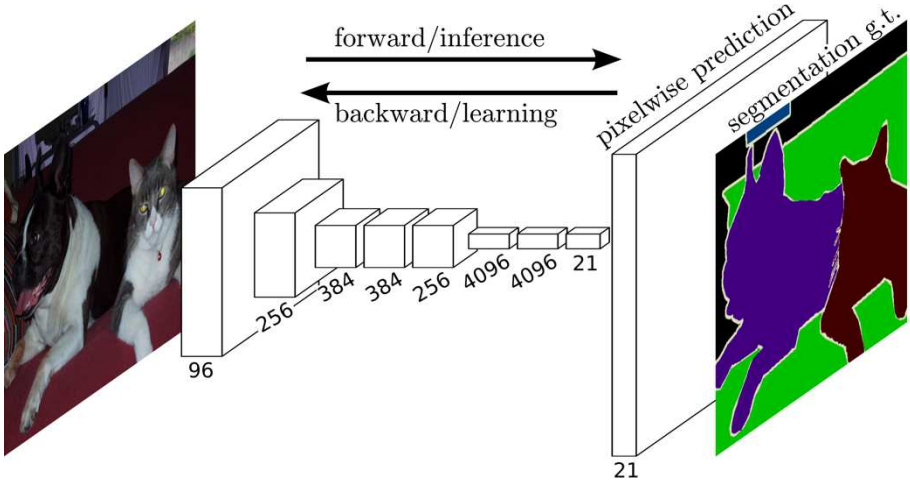
Mnoge implementacije semantičke segmentacije temelji se na konvolucijskim neuronskim mrežama. U početku, mreže su primarno bile strukturirane tako da su provodile konvoluciju

slike sa ograničenim kapacitetom i receptivnim poljem. Međutim, ovo je dovelo do gubitka prostornih informacija, što je u nekim slučajevima rezultiralo nepreciznim granicama segmentacije.


Slika 1.3 Primjer semantičke segmentacije na domeni prometnih slika [34]

Kao odgovor, razvijene su mreže temeljene na kombinaciji kodera i dekodera te se danas smatraju standardom u polju semantičke segmentacije. Takvi modeli, poput **Potpuno povezanih mreža** (FCN, *Fully Convolutional Networks*) [35] i **U-Net** [36], koriste se za mapiranje prostornih informacija kroz slojeve mreže, čime se postiže bolje očuvanje granica.

U posljednje vrijeme, počeli smo primjećivati razvoj arhitektura koje koriste mehanizme pažnje za dodatno poboljšanje performansi. Arhitekture poput **DANet** [37] koriste pažnju da bi bolje razumjele kontekstualne informacije u slici, a zatim te informacije koriste za poboljšanje segmentacijske točnosti. Unatoč značajnom napretku, semantička segmentacija još uvijek se suočava izazovima poput malih varijacija u izgledu istog objekta, različitih perspektiva, ili nedostatak dovoljno označenih podataka za treniranje modela. U ovom radu, usmjerit ćemo se na ove izazove, posebno istražujući kako polunadzirane metode mogu poboljšati semantičku segmentaciju s slabo označenim skupovima podataka.

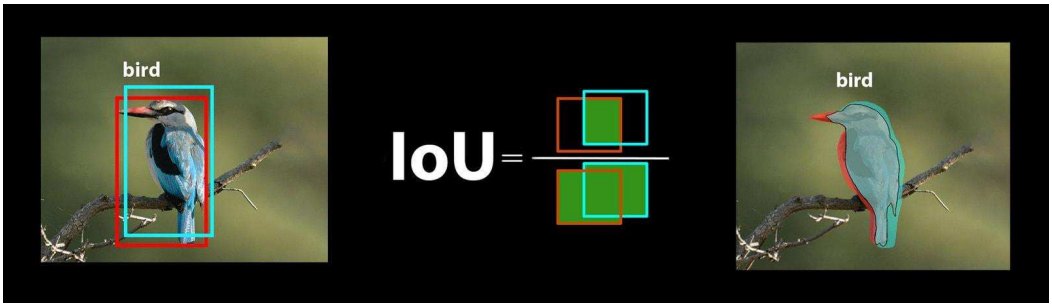



Slika 1.4 Primjer punopovezane konvolucijske mreže [35]

U semantičkoj segmentaciji najčešća mjera je **Jaccardov indeks**, ili **IoU** (engl. *Intersect over Union*) koji se najčešće računa kao prosjek svake klase (**mIoU**), a definira se kao:

$$IoU = \frac{TP}{TP + FN + FP} \tag{1.1}$$

gdje se TP definira kao točno označen pozitivni primjer, FN netočno označen negativni primjer, a FP netočno označen pozitivni primjer. Vizualizacija IoU dana je na slici 1.5




Slika 1.5 Vizualizacija korištenja IoU metrike [41]

## 1.3. Polunadzirano učenje

**Polunadzirano učenje** pristup je koji kombinira korištenje označenih i neoznačenih podataka tijekom procesa učenja. Ovo je posebno relevantno u kontekstu semantičke segmentacije gdje su, usprkos brzom napretku tehnologija za duboko učenje, označeni podaci često skupi i vremenski zahtjevni za pripremu. U polunadziranom okruženju, modeli se treniraju na kombinaciji označenih i neoznačenih slika, što može drastično proširiti dostupni skup podataka za učenje i poboljšati generalizaciju modela.

Polunadzirane metode možemo temeljiti na različitim strategijama. Jedna od njih je **samoučenje** [7] (engl. *self-training*), gdje model prvo trenira na označenim podacima, a zatim se koristi za generiranje predikcija na neoznačenim podacima. Tako dobivene pseudo-oznake se zatim koriste za daljnje treniranje modela. Druga se temelji na **regularizaciji konzistencijom** [11, 46], koja potiče model da daje konzistentne predikcije pod različitim uvjetima, poput malih izmjena na ulaznim slikama ili varijacija unutar modela, kao što su različiti skupovi težina. Treća se temelji na **augmentaciji podataka**, gdje se ulazni podaci augmentiraju (npr. afinim transformacijama) te se nastoji naučiti mreža uz pomoć generativnih modela [28]. Četvrta metoda je **minimizacija entropije** [29] čime se induktivno pretpostavlja da je distribucija podataka grupirana u klastere; pretpostavka koju postavlja i označavanje pseudo-oznakama. Takve metode pomogle su modelima da postignu bolje rezultate na standardnim skupovima podataka, poput **PASCAL VOC** [39] (za detekciju objekata) ili **Cityscapes** [40] (za semantičku segmentaciju), čak i kada su dostupne samo male količine označenih podataka.

Međutim, unatoč obećavajućim rezultatima, polunadzirana semantička segmentacija još uvijek se nalazi pred brojnim izazovima. Na primjer, metode samo-učenja mogu biti podložne kumulaciji pogreške, gdje netočne oznake generirane u početnim fazama mogu negativno utjecati na učenje modela u kasnijim fazama. Osim toga, tehnike regularizacije konzistencijom mogu biti teške za primjenu u praksi jer zahtijevaju precizno podešavanje hiperparametara [46]. Unatoč tim izazovima, polunadzirana semantička segmentacija ostaje zanimljivo i obećavajuće područje istraživanja. U ovom radu nastojimo dalje istražiti ovaj pristup, usredotočujući se na kako se može efikasno primijeniti u kontekstu segmentacije slika s nepotpunim oznakama.

## 1.4. Učenje nad pseudooznakama

**Pseudooznake** predstavljaju tehniku za povećanje dostupnosti označenih podataka za treniranje modela, a posebno su korisne u kontekstu semantičke segmentacije gdje je priprema detaljno označenih podataka vrlo zahtjevna. Ovaj pristup koristi već trenirani model da generira oznake za neoznačene podatke. Tako generirane oznake nazivamo pseudooznake.

Proces **samoučenja** (engl. *self-training*) modela nad pseudooznakama radi sljedeće [8]:

1. **Treniranje početnog modela – zagrijavanje učitelja**: Model se prvo trenira na dostupnim označenim podacima. Početni model ne mora biti savršen - njegova svrha je samo da pruži dovoljno točan temelj za daljnje učenje, ali se ipak pokazalo učinkovitije ako i učitelj daje bolje predikcije. Formalno, ako definiramo označene ulaze $\{(x^1, y^1), (x^2, y^2), \ldots, (x^n, y^n)\}$, tada učitelja treniramo nad označenim podacima funkcijom unakrsne entropije:

$$\frac{1}{n}\sum_{i=1}^{n} l\left(y_i, f^{šum}(x_i, \Theta^t)\right) \tag{1.2}$$

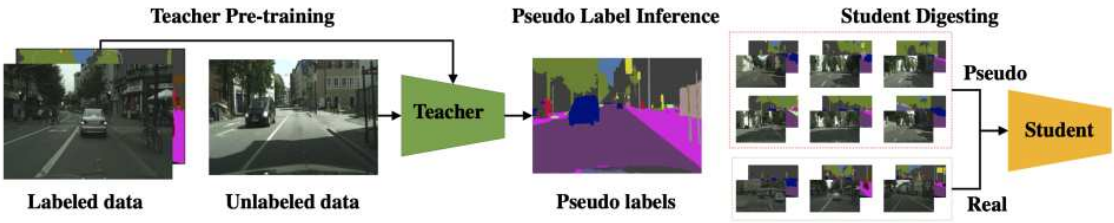
2. **Generiranje pseudooznaka**: Početni model se zatim koristi za generiranje pseudooznaka za neoznačene podatke. Ove oznake nisu savršene, ali se nadamo da će pružiti korisnu informaciju za daljnje učenje. Formalno:

$$y^i = f\left(x^i, \Theta^t\right), \forall i = 1, \cdots, m \tag{1.3}$$

3. **Treniranje nad pseudooznakama**: Model se zatim ponovno trenira, ovaj put koristeći i izvorne označene podatke i neoznačene podatke sa pseudooznakama. Ovaj korak može se ponavljati više puta, pri čemu se model svaki put koristi za ažuriranje pseudooznaka. Važno je napomenuti da se označeni i pseudooznačeni skupovi podataka mogu miješati prilikom treniranja ili trenirati zasebno. Eksperimenti [8] su pokazali da je bolje rješenje miješanje skupova podataka. Student može biti kopija učitelja, ali može biti i model jednakog ili većeg kapaciteta. Funkcija gubitka tada je:

$$\frac{1}{n}\sum_{i=1}^{n} l\left(y_i, f^{student}(x_i, \Theta^s)\right) + \frac{1}{m}\sum_{i=1}^{m} l\left(f^{učitelj}(x_i, \Theta^t), f^{student}(x_i, \Theta^s)\right), \quad m \gg n \tag{1.4}$$

gdje $f^{šum}$ označava funkciju na koju je primijenjen izvor šuma, bilo kroz metode propadanja težina (engl. *dropout*) ili augmentiranja ulaza.

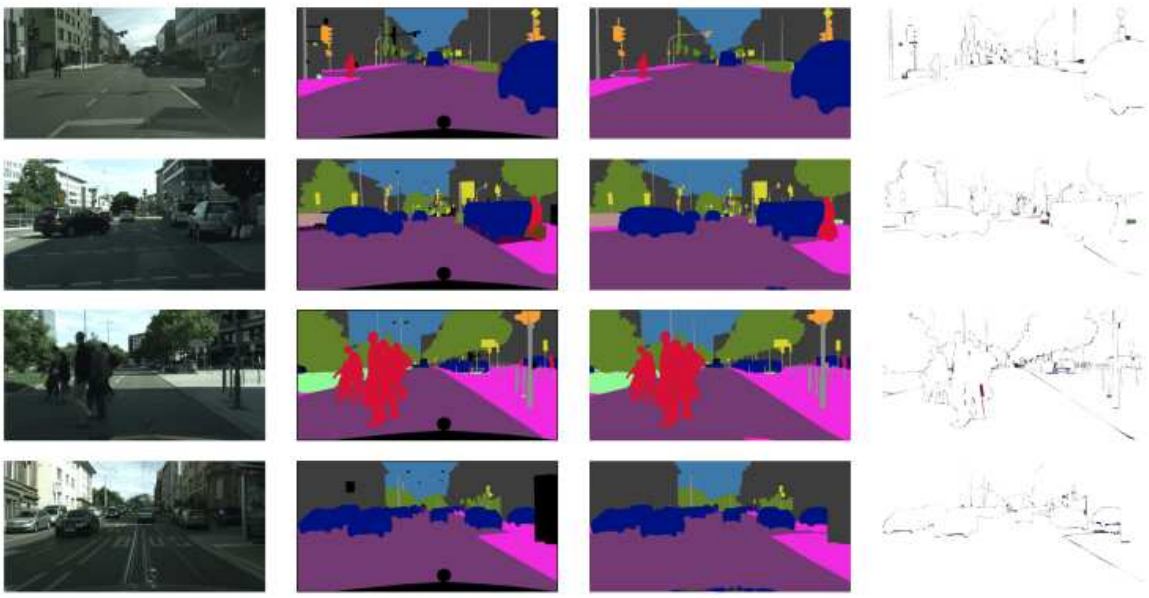



Slika 1.5 Arhitektura samoučenja pomoću ansambla studenta i učitelja [8].

Koristeći navedeni pristup, model može učiti od mnogo veće količine podataka nego što bi inače bilo moguće. Osim toga, model kontinuirano ažurira pseudooznake tijekom učenja zbog čega ima potencijal kontinuirano poboljšavati svoje performanse čak i kada su dostupni samo neoznačeni podaci.

Učenje modela nad pseudooznakama također nosi određene izazove. Jedan od glavnih je izvor šuma - ako početni model ima pogrešne predikcije, te će pogreške možda biti prenesene na pseudooznake što može negativno utjecati na učenje. Da bi se ovo ublažilo, često se koriste različite tehnike, poput pouzdanosti oznaka, gdje se za učenje koriste samo pseudooznake za koje model ima visoko povjerenje.

Unatoč tim izazovima, učenje modela nad pseudooznakama ostaje obećavajući pristup za semantičku segmentaciju i druge zadatke dubokog učenja, omogućavajući modelima da uče od mnogo veće količine podataka nego što bi inače bilo moguće.



Slika 1.6 Izvorna slika, ručno označeni podaci te pseudooznake i njihova razlika [8]. Vidljivo je da postoji lagana razlika između predikcije u rubovima koje algoritam samo-učenja nastoji korigirati.

## 1.5. OpenCLIP

Algoritam **Clip** [3] temelji se na kombinaciji modela vizualnog i tekstualnog kodera te korištenje kontrastnog gubitka. U kontekstu Clip-a, vizualni model izvlači značajke iz slika koristeći duboke konvolucijske mreže, dok jezični model koristi slojeve transformera za procesiranje teksta. Zbog svoje sposobnosti učenja zajedničkih značajki između slika i teksta postao je koristan alat u različitim domenama računalnog vida i obrade prirodnog jezika.

Okvirna struktura modela Clip opisana je algoritmom 1.1.

```
# enkoder_slike - ResNet ili ViT
# enkoder_teksta - CBOW ili tekstualni transformer
# I[n, h, w, c] - minibatch poravnatih slika
# T[n, l] - minibatch poravnatog teksta
# W_i[d_i, d_e] - naučena projekcija slike za ugrađivanje
# W_t[d_t, d_e] - naučena projekcija teksta za ugrađivanje
# t - naučena temperatura

# izvlačenje vizualnih i tekstualnih reprezentacija
I_f = koder_slike(I) #[n, d_i]
T_f = koder_teksta(T) #[n, d_t]
# udružena višemodalna ugrađivanja [n, d_e]
I_e = l2_normalize(np.dot(I_f, W_i), axis=1)
```

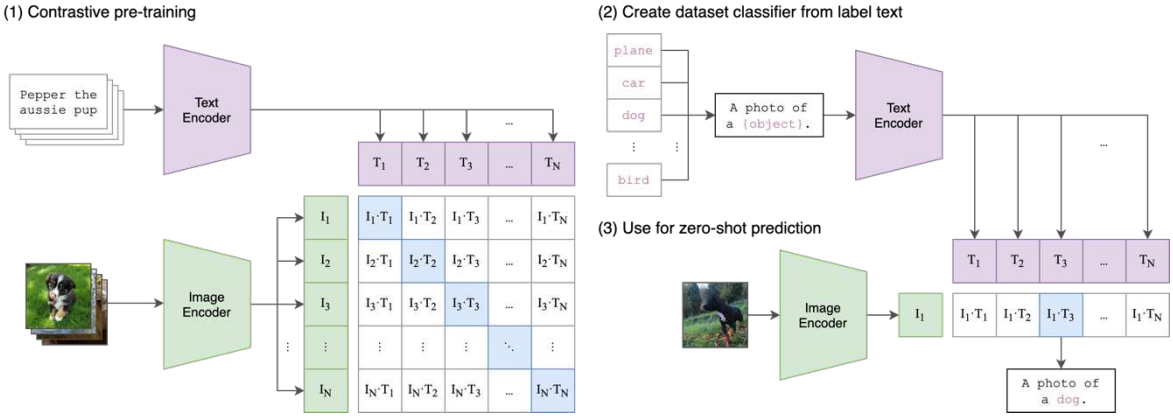
```
T_e = l2_normalize(np.dot(T_f, W_t), axis=1)

# skalirani parovi sličnosti kosinusa [n, n]
logiti = np.dot(I_e, T_e.T) * np.exp(t)

# simetrična funkcija gubitka
oznake = np.arange(n)
loss_img = gubitak_unakrsne_entropije(logiti, oznake, axis=0)
loss_text = gubitak_unakrsne_entropije(logiti, oznake, axis=1)
loss = (loss_img + loss_text)/2
```

Algoritam 1.1 Pseudokod Clip algoritma [3].

Cilj funkcije kontrastnog gubitka jest maksimizirati sličnosti između pozitivnih parova slika i teksta, dok minimizira sličnost između negativnih parova. Ovaj pristup omogućuje modelu da efikasno uči značajke koje su zajedničke slikama i za tekstu, što rezultira snažnim semantičkim vezama. To se pokazalo izuzetno korisnim u različitim zadacima gdje je modelu korisno razumjeti tekstualne opise objekata ili scena koji su povezani sa slikama.




Slika 1.7 Shema arhitekture predtreniranja i predikcije modela OpenCLIP [2].

Clip ima mnogo primjena i unaprijeđenja kao što je dodjeljivanje tekstualnog semantičkog značenja slikama (engl. image captioning), sinteza slike putem teksta [30] i odgovaranje na tekstualne upite [31]. Iako Clip postiže vrlo visoku zero-shot točnost nad novim domenama, trenutna namjena orijentirana mu je na zadatak klasifikacije i detekcije. Kako bi se ostvario zadatak segmentacije potrebno je preformulirati zadani model, što će biti tema ovog rada. Koristit će se verzija Clip-a otvorenog koda OpenClip [2] koji se pokazao kao dobar konkurent originalnom Clip-u [46].

## 1.6. Segment Anything (SAM)

**Segment Anything** (SAM) [1] model je tvrtke Facebook razvijen za rješavanje zadatka agnostičke segmentacije.

Slika 1.8 Primjeri predikcije modela Segment Anything [1].

**Zadatak modela** je pronaći zadovoljavajuću segmentacijsku masku za dani upit. Upit generalno može biti bilo koja informacija koja određuje što se želi segmentirati. U izvornom radu [1], upiti su ograničeni na točke, okvire i tekst. Iako, zbog proizvoljno nepredvidivih upita, predviđena maska može biti i višeznačna (primjerice maska kotača na automobilu i maska čitavog automobila, slika 1.8), zahtjeva se da barem jedna predviđena maska bude ispravna. Ovako definiran zadatak omogućava SAM-u *zero-shot* prijenos i integraciju u veće arhitekture kao komponenta koja daje predikcije na upite drugih modela bez dodatnog treniranja.

**Arhitektura modela** za pronalaženje maske sastoji se od **vizualnog kodera** za dobivanje ugrađivanja slike te **kodera za upite** za dobivanje ugrađivanja upita. Ta ugrađivanja koriste se na **dekoderu za maske** čiji je zadatak kombinirati modalnosti ugrađivanja slike i upita kako bi predvidio kvalitetne maske u zadovoljavajućem vremenu. (slika 1.9)

Detalji pojedinih komponenata dani su u nastavku:

- **Koder slike** - Koristi se Vision Transformer (ViT) predtrenirani na maskiranom autoenkoderu [10] te je prilagođen za obradu ulaza visoke rezolucije. Iteracija se provodi jednom po slici i može se primijeniti prije postavljanja upita modelu, što dozvoljava korištenje računalno zahtjevnijeg kodera.
- **Koder upita** – razlikuje rijetke (točke, okviri, tekst) i guste (maske) upite. Svi upiti su preslikani u 256-dimenzionalna ugrađivanja. Točke su preslikane sumom

pozicijskih ugrađivanja koordinata točke i naučenih ugrađivanja koja određuju je li točka na objektu ili pozadini. Upiti okvira su preslikani sumom pozicijskih ugrađivanja gornje lijeve i donje desne točke okvira te naučenih ugrađivanja za pripadajuće točke. Ugrađivanja za tekstualne upite preuzete su iz Clip-a. Gusti upiti se ugrađuju pomoću konvolucija i zbrajaju po elementima sa ugrađivanjima slike.

- **Dekoder maska** - efikasno preslikava ugrađivanja slike, upita i izlazni token na masku koristeći dvosmjerni mehanizam pažnje. Nakon dva bloka transformera slika se naduzorkuje te prolazi kroz višeslojni perceptron i dinamički linearni klasifikator koji daje vjerojatnosti klasa na razini piksela.

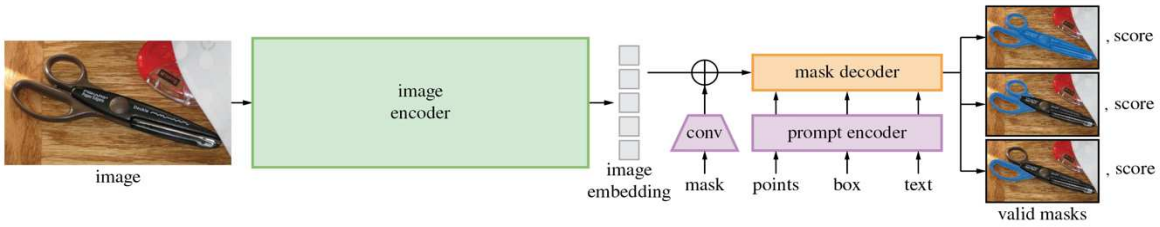
Model se trenira kroz više iteracija za svaku sliku.

1. U prvoj iteraciji se jednakom vjerojatnošću odabire ili točka ili okvir oko *ground-truth* oznake. Ukoliko se bira točka, ona se uniformnom vjerojatnošću uzorkuje sa ground-truth oznake, dok se okviri uzorkuju sa nadodanim šumom po svakoj točki.
2. Nakon inicijalne predikcije se uniformno uzorkuje točka u području pogreške prve iteracije, gdje netočno označene negativne regije (FN) predstavljaju objekt a netočno označene pozitivne regije (FP) predstavljaju pozadinu. Također se na upit dodaje i predikcija prijašnje iteracije kao logiti kako bi se sačuvala sva informacija. Ukoliko prijašnja iteracija predvidi više maski, iduća iteracija uzima masku najveće predviđene IoU metrike.
3. Zadnje dvije iteracije se provode bez uzorkovanja točke kako bi model u potpunosti učio iz predikcije prethodne iteracije.

Utvrđeno je da je optimalno provoditi 2. korak po 8 iteracija što ukupno iznosi 11 iteracija po slici. Za funkciju gubitka koristi se kombinacija fokalnog gubitka [46] i *"dice"* gubitka [47] u omjeru 20:1.

Osim označavanja prema unaprijed definiranim upitima, SAM dozvoljava automatsko generiranje maski na razini čitave slike. Taj proces uključuje veći broj parametara koje utječu na generiranje. Proces funkcionira tako da uzorkuje rešetkasto polje točaka čija gustoća se može konfigurirati parametrom **pps** (engl. points per side) koji iznosi broju točaka po svakoj dimenziji slike, što ukupno rezultira sa 2*pps točaka. Još jedan parametar koji utječe na generiranje maski jest postavljanje minimalne dozvoljene površine za uklananje

premalih maski (MMRA, min-max region area). Mjerenje utjecaja tih parametra dano je u poglavlju 3.1.




Slika 1.9 Arhitektura modela Segment Anything [1].

Treniranje SAM-a zahtjeva izuzetno velik skup označenih podataka koji nije dostupan u segmentacijskoj domeni. Iz tog razloga su autori razvili sustav označavanja **podatkovnim strojem** (engl. *data engine*) koji nastoji automatizirati označavanje predikcijom samog modela kroz tri faze. Model se na kraju svake tri faze ponovno učio nad novim, uvećanim skupom podataka.

- **Ručno označavanje potpomognuto modelom** (engl. *assisted-manual)*: označavanje je u potpunosti ručno uz pomoć interaktivnih alata koji integriraju predložene maske SAM-a. Pri označavanju se ne pamte imena oznaka jer je cilj izbjeći semantičke ograde i fokusirati se na pronalaženje općenitih objekata (engl. *"stuff" and "things"*)
- **Polu-automatsko označavanje** (engl. *semi-automatic*): SAM predlaže oznake visoke pouzdanosti (uz pomoć detektora okvira nad maskama prve faze) čime se automatizira označavanje podskupa maski visoke kvalitete, dok se preostale maske ručno označavaju
- **Potpuno automatsko označavanje** (engl. *fully automatic*)*:* označavanje je u potpunosti automatizirano SAM-ovim predikcijama tako da se modelu daje na upit rešetkasto polje točaka, što na izlazu daje u prosjeku 100 maska visoke kvalitete

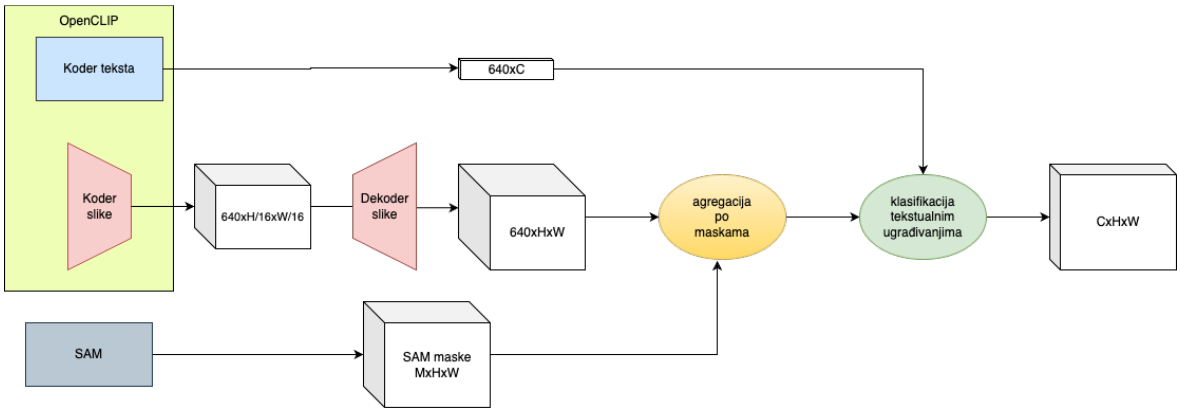
Ovako razvijen sustav omogućio je razvoj skupa podataka SA-1B sa 11 milijuna slika i 1.1 milijardi maska visoke kvalitete, čije generiranje je 99.1% potpuno automatizirano.

Iako je sam model visokog kapaciteta i mogućnosti, trenutno nije javno omogućeno davanje tekstualnih upita za maskiranje. Međutim, njegove predložene predikcije bez upita su dovoljno visoke zero-shot točnosti da se mogu iskoristiti u kombinaciji sa modelom koji ima koder teksta, kao što je primjerice OpenCLIP.

# 2. Korišteni model i skup podataka

## 2.1. OpenSAM

OpenCLIP stvara zaključke skalarnim umnoškom vizualnog i tekstulanog kodera, međutim korišten je isključivo za klasifikaciju predmeta na slici te nije opremljen za zadatak semantičke segmentacije. Postavlja se pitanje: je li moguće naučiti preslikavanje OpenClip-ovog kodera kako bi se ostvarila gusta predikcija? Dodatno, želimo izmjeriti utjecaj maskiranja ugrađivanja slike kako bi izdvojili objekt od interesa i klasificirati piksele izdvojenog ugrađivanja. U tu svrhu možemo koristiti SAM-ovu sposobnost zero-shot predikcije i koristiti dobivene maske za gustu predikciju. U svrhu rješavanja ovog problema predlažemo model **OpenClip-SAM**. Shema modela OpenClip-SAM nalazi se na slici 2.1.




Slika 2.1 **Model OpenClip-SAM.** Trodimenzionalnim oblicima označeni su tenzori u međukoracima predikcije modela. Za zajedničku dimenziju ***D*** stavljena je vrijednost 640. ***H*** i ***W*** su veličine izvorne slike, a ***M*** je broj maski.

Model integrira OpenCLIP i SAM na sljedeći način:

1. Na izlazu slikovnog kodera dobivena su ugrađivanja dimenzija $DxHxW$, gdje je $D$ zajednička dimenzija OpenCLIP-ovih slikovnih i tekstualnih ugrađivanja koja će se koristiti za predviđanje skalarnim umnoškom. U primjeru na slici 2.1 ta dimenzija iznosi 640.

2. Slika prolazi kroz dekoder, primjerice naduzorkovanjem do dimenzija originalne slike, međutim moguće je zamijeniti proizvoljnim dekoderom. Na izlazu se dobiva vektor $DxHxW$.

3. Potom slijedi operacija agregacije maska:

    a. SAM predloži **M** binarnih maski. Za svaku masku se vrši se MAP (engl. *Masked Average Pooling*) tako da se vrijednosti ugrađivanja u području maske uprosječuju po dimenzijama prostora HxW, a ostale vrijednosti zanemaruju (postavljaju na 0). Na izlazu ovog koraka se dobiva tenzor **w** dimenzije 1x640.

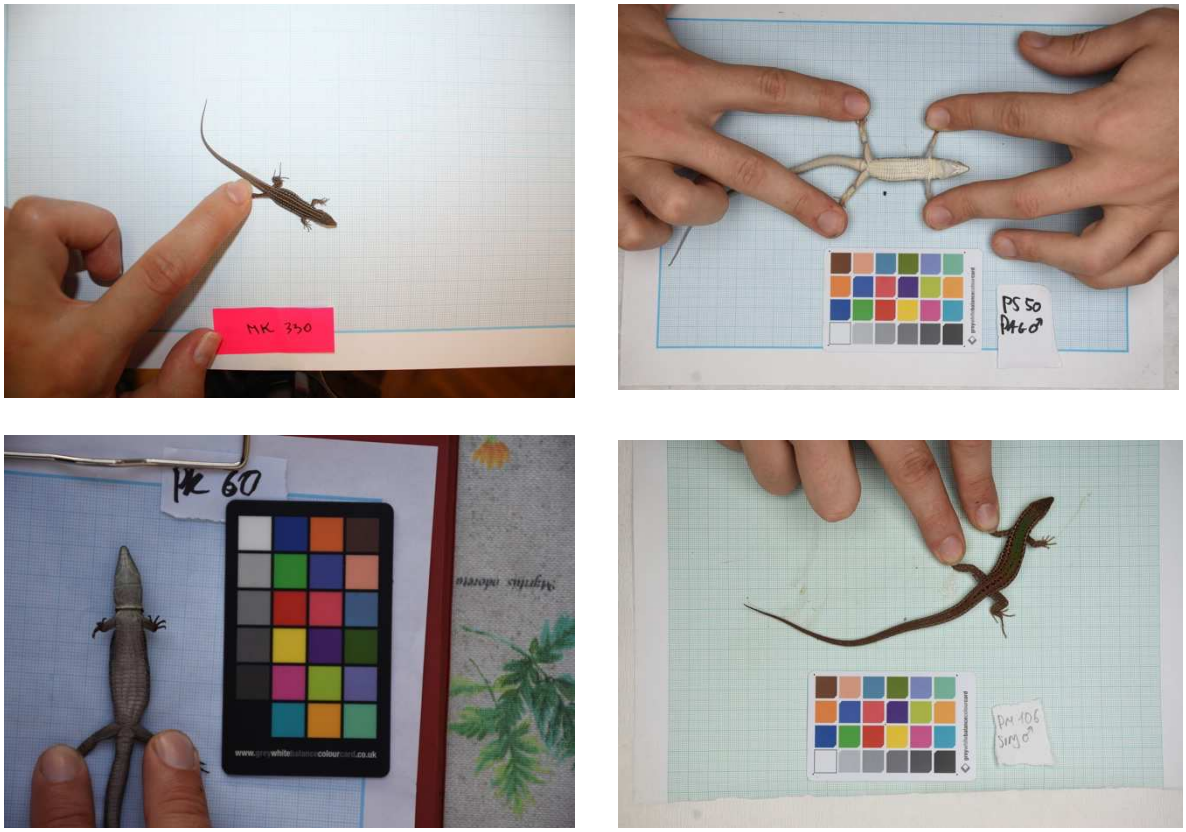
    b. OpenCLIP-ova tekstualna ugrađivanja za svaku klasu sadrže ugrađivanje dimenzije 640. Množenjem tih ugrađivanja tenzorom **w** dobivaju se logiti. Rezultat je tenzor dimenzije 1xC.

    c. Provlačenjem logita kroz *softmax* dobivamo vjerojatnosti predikcije. Klasa najveće vjerojatnosti dodjeljuje se pikselima na području originalne maske **M**.

4. Budući da SAM može predložiti više maski koje se mogu preklapati, one se sortiraju po površini od najveće prema najmanjoj te se operatorom unije agregiraju. Na izlaz konačno dobivamo tenzor $CxHxW$ gdje za svaki piksel imamo vjerojatnosnu distribuciju klasa

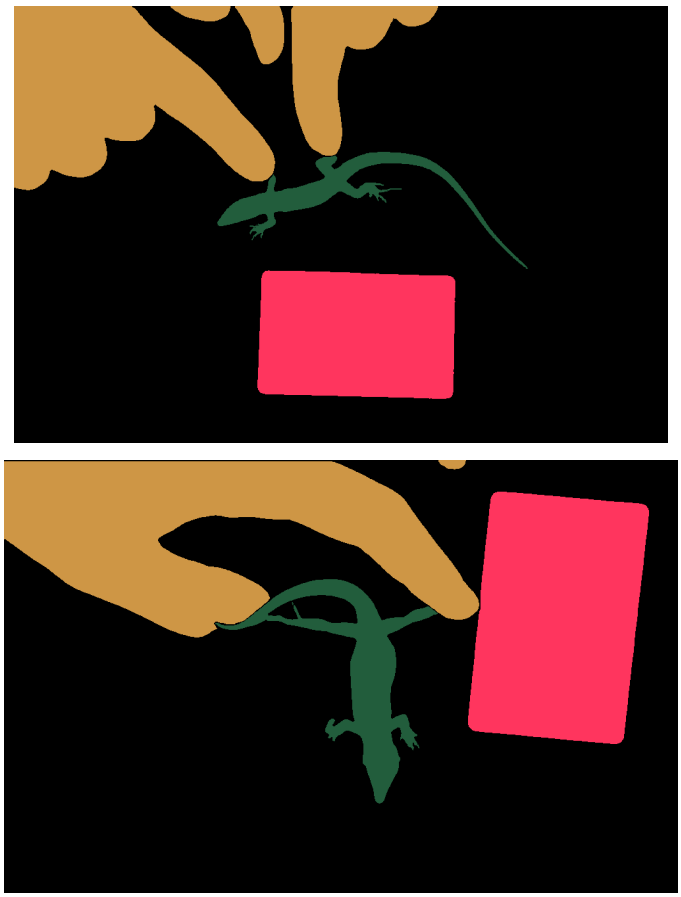
## 2.2. Skup podataka „Gušteri“

**Skup podataka „Gušteri“** (slika 2.2) razvijen je u suradnji sa doktorandom sa sveučilišta Prirodoslovnog Matematičkog Fakulteta u Zagrebu. Sačinjen je od 6150 slika guštera koje su podijeljene na 85 označenih slika te 6065 neoznačenih. Slike su slikane u sličnim okolnostima, sadržavajući objekte guštera ljudske ruke, legende boja te pozadine milimetarskog papira. Skup podataka sastoji se od 4 odabrane klase: „gušter“ ( *„a lizard“* ), „ljudska ruka“ ( *„a human hand“* ), „legenda boja“ ( *„a color bord“* ) i „pozadina“ ( *„a background“* ). Gušter se nalazi na svakoj slici, sa potpuno ili djelomično otkrivenim tijelom, nad različitim uvjetima svjetline i položaja. U svakoj slici je također prisutna ljudska ruka i legenda boja (za identifikaciju guštera). U pojedinim slikama se pojavljuju i anomalije poput papirića te one nemaju dodijeljenu klasu. Označeni skup podataka podijeljen je na skup za treniranje (60 primjeraka), validaciju (10 primjeraka) i testiranje (8 primjeraka). Slike su ručno označene uz pomoć alata CVAT koristeći Segmentation Mask 1.1 format. Primjer oznaka prikazan je na slici 8.

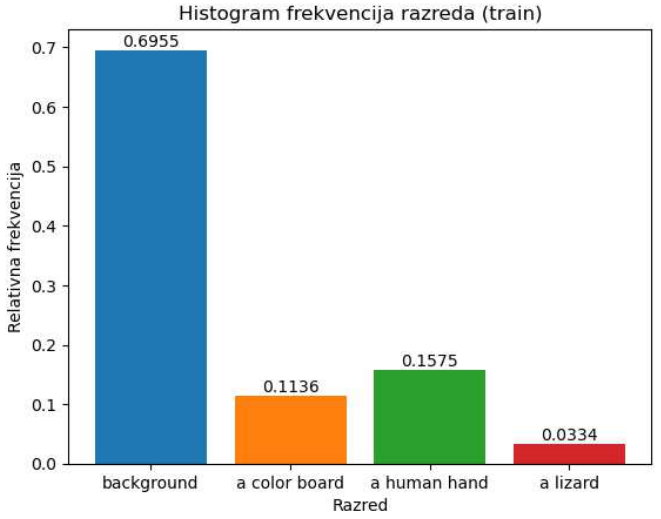


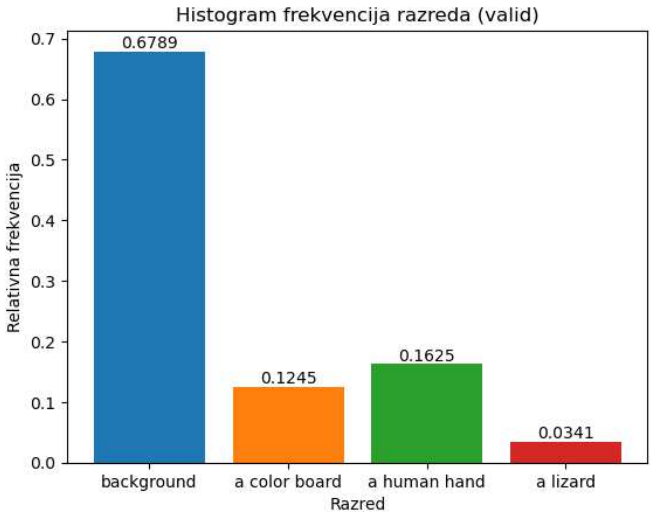

Slika 2.2 Primjeri slika iz skupa podataka "Gušteri".

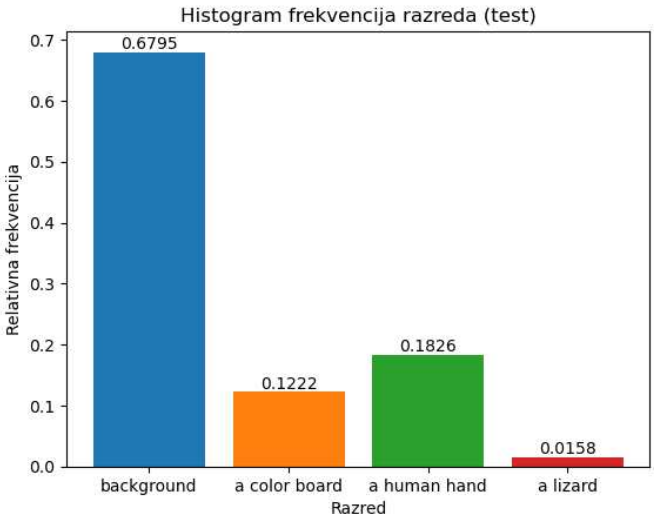
Radi vremenskih i resursnih ograničenja, mjerenja su odrađena pomoću skupova podataka označenih i neoznačenih slika u iznosima 60O:60N, 60O:120N, 60O:240N te 60O:480N, gdje O i N označavaju označene i neoznačene primjere. Slike su RGB rezolucije 1024x683 piksela, te se prije treniranja poduzorkuju na dimenzije 255x255 kako bi ubrzalo proces učenja. Budući da se zbog poduzorkovanja izgubi puno detalja sa slike, posebice sa teksture guštera i pozadine, očekujemo da će rezultati biti lošiji. Histogram na grafu 2.1 pokazuje distribuciju klasa svakog podskupa, koji će poslužiti za procjenu težina prilikom računanja funkcije gubitka.



Slika 2.3 Primjeri označenih slika skupa za treniranje.



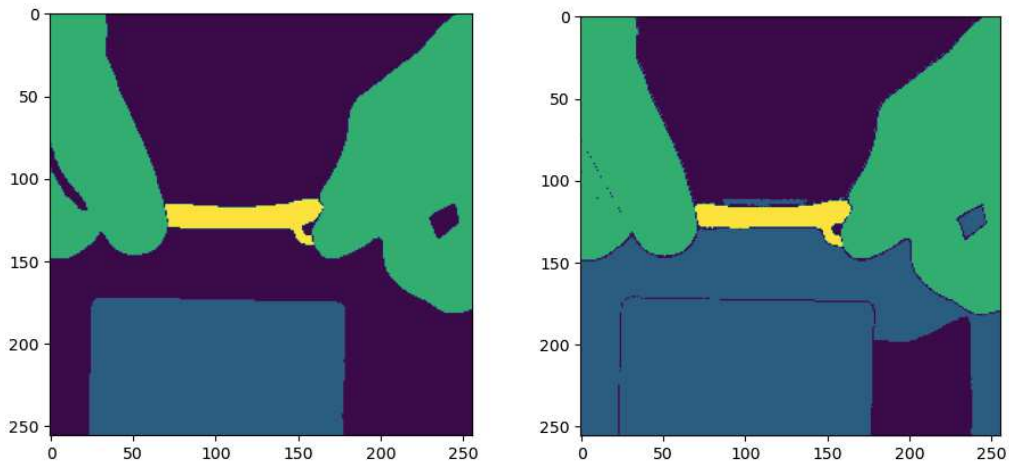




Graf 2.1 Histogram frekvencija po podskupovima skupa podataka „Gušteri“.

# 3. Eksperimenti

Model je u potpunosti oblikovan koristeći programski jezik Python 3.9. Za bilježenje podataka o treniranju korištena je platforma *Weights and Biases*. Treniranje je provedeno u Jupyter bilježnicama na dvije platforme, Google Collab i Kaggle, ovisno o dostupnosti grafičkih resursa. Obje platforme koriste Nvidia T4 grafičku karticu. U svrhu eksperimenata rađeni su specijalizirani DataLoader za učitavanje skupova podataka i rukovanje sa skupovima.

U eksperimentima je potrebno generirati SAM maske koje će se koristiti prilikom učenja. Budući da je njegova predikcija maski zahtjevna, maske se generiraju prije učenja te spremaju na tvrdi disk. Ovo ograničava fleksibilnost treniranja jer ne dozvoljava augmentacije položaja i rotacije slika prije treniranja.




Slika 3.1 Lijevo: *Ground-truth* oznaka slike guštera.
Desno: Primjer predikcije predloženim postupkom.
zelena – ljudska ruka, žuta – gušter, plava – legenda boja, ljubičasta – pozadina

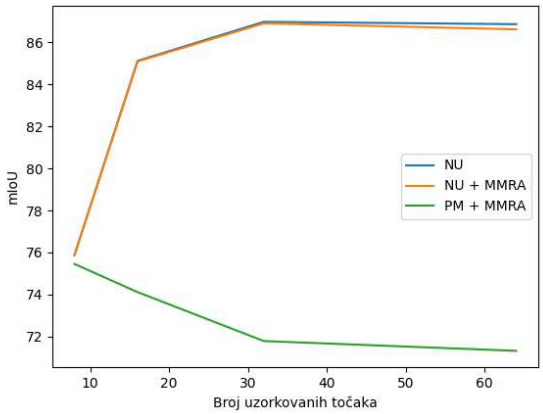
Kao kralježnicu, OpenSAM model koristi OpenCLIP-ov ConvNext bazni model predtreniran na skupu Laion2B [26]. Glava modela je skraćena da sačuva gustu predikciju ugrađivanja te kako bi se ona mogla dekodirati do veličine slike. SAM-ova konfiguracija definirana je ViT-H modelom.

## 3.1. Učinak SAM-a na OpenCLIP

Kako bismo utemeljili učinke čistog SAM modela prije treniranja radimo studiju ablacije nad skupom za treniranje. Uspoređujemo **isključivo predikciju** čistog OpenCLIP-a te agregiranog OpenCLIP-a preko SAM-ovih maski, bez prethodnog treniranja. Vidimo da SAM znatno poboljša predikcijsku moć OpenCLIP-a. Eksperimenti su provedeni nad različitim SAM konfiguracijama. Koristi se različito uzorkovanje točaka pps (engl. *point-per-side*), metoda naduzorkovanja ugrađivanja bilinearnom interpolacijom i poduzorkovanja maski metodom najbližeg susjeda, te korištenje MMRA – najmanja dozvoljena regija maski (engl. *min-mask region area*). Rezultati eksperimenata prikazani su na tablici 1. i slici 7. Naduzorkovanje i poduzorkovanje maski koristi se kao usporedba utjecaja različitih metoda dekodiranja.

| mIoU | baseline, OpenCLIP | OpenCLIP + SAM (pps=8) | OpenCLIP + SAM (pps=16) | OpenCLIP + SAM (pps=32) | OpenCLIP + SAM (pps=64) |
|---|---|---|---|---|---|
| 47 slika | 48.9240 | | | | |
| 47 slika, NU | | 75.8756 | 85.1199 | **86.9784** | 86.8669 |
| 47 slika, NU, MMRA | | 75.8669 | 85.0887 | 86.9052 | 86.6208 |
| 47 slika, PM, MMRA | | 75.4530 | 74.1124 | 71.7830 | 71.3158 |

Tablica 3.1 Utjecaj SAM maski na guste predikcije modelom OpenCLIP. [49]
NU - naduzorkovanje ugrađivanja, PM - poduzorkovanje maski.




Graf 3.1 Utjecaj različitih konfiguracija metode SAM na OpenCLIP predikcije. NU – naduzorkovanje ugrađivanja, PM – poduzorkovanje maski, MMRA – najmanje dozvoljena regija maski (postavljena na 100).

Vidljivo je da naduzorkovanje OpenCLIP ugrađivanja utječe bolje na predikciju u usporedbi s poduzorkovanjem maski. Pretpostavka je da se informacija na razini piksela kod poduzorkovanja gubi u grubim oznakama. Prilikom generiranja SAM maski, koristiti će se konfiguracija naduzorkovanja ugrađivanja sa 16 uzorkovanih točaka, kao kompromis između visoke točnosti i vremenske raspoloživosti.

## 3.2. Zagrijavanje modela

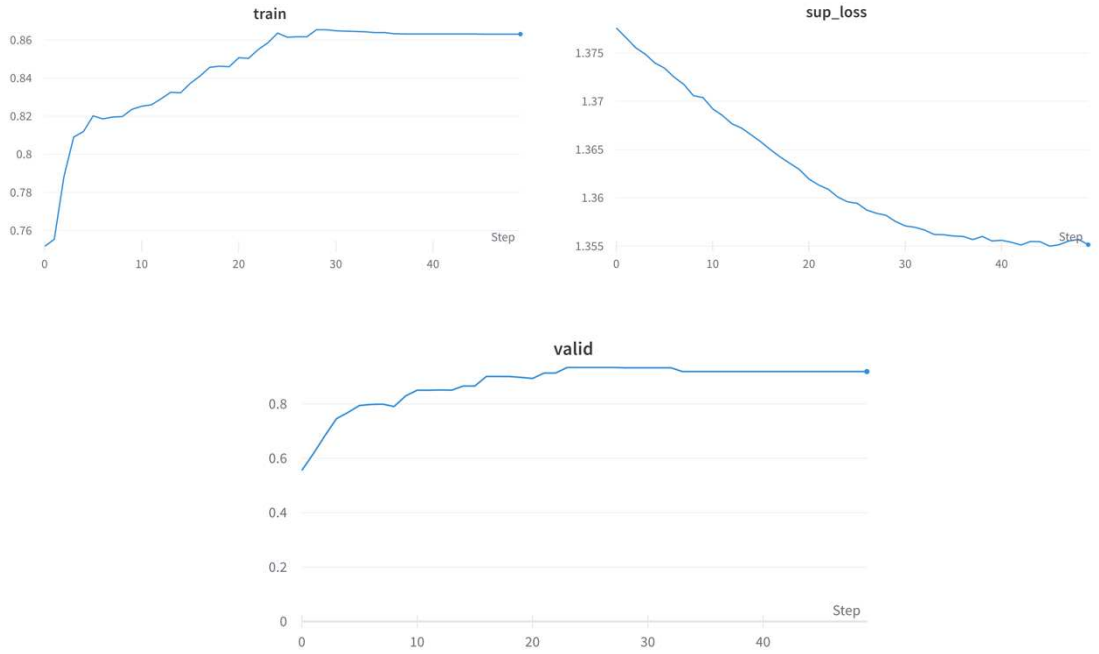
Zagrijavanje mreže učitelja potrebno je kako bi učitelj generirao bolje pseudooznake što je ključno kako nebi uveli preveliki šum. Zagrijavanje je isprobano na više hiperparametara i metoda optimizacije, te su odabrane vrijednosti dane u tablici 1.

Za metode optimizacije isproban je običan SGD optimizator te AdamW [25], te se pokazalo da AdamW postiže bolje rezultate, kao i u ranijim radovima [24]. Prilikom učenja model prolazi kroz fazu zagrijavanja kako bi se fino podesio na skup za učenje. Pri tome je važno pravilno podesiti stopu učenja kako se već u ovom koraku model ne bi zasitio. Regulator stope učenja je kosinusno kaljenje, opisano formulom 3.1.

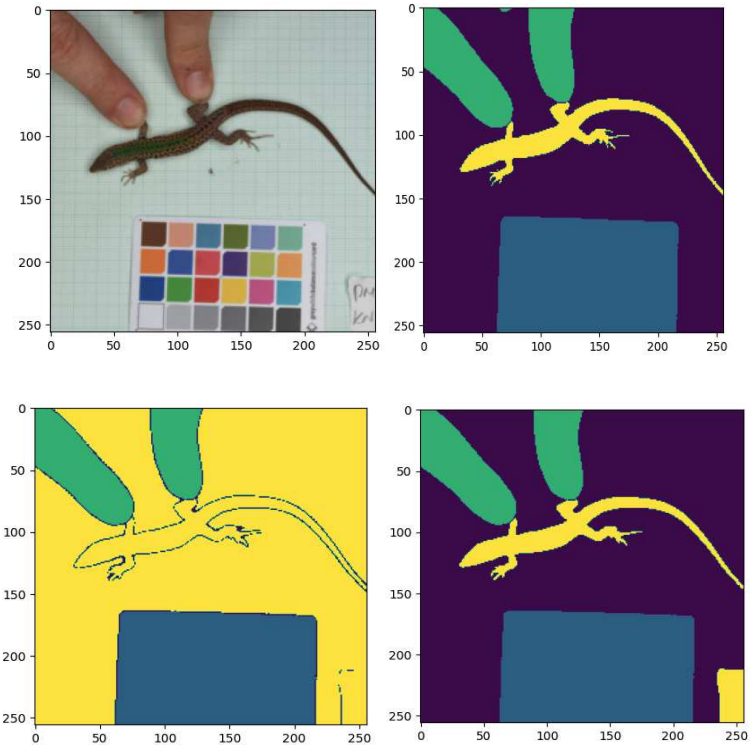
$$\eta_t = \eta_{min} + \frac{1}{2}(\eta_{max} - \eta_{min})(1 + cos(\frac{T_{cur}}{T_{max}}\pi)) \tag{3.1}$$

| **Hiperparametar** | **Vrijednost** |
|---|---|
| openCLIP slikovni enkoder | convnext_base_w (pretrained laion2b) |
| SAM slikovni enkoder | vit_H |
| broj epoha | 30 |
| stopa učenja | (eta_max, eta_min) = (3e-7, 1e-8) |
| optimizator | AdamW, β = (0.9, 0.999) |
| raspoređivač koraka učenja | Kosinusno kaljenje |
| propadanje težina | 1e-8 |
| težine razreda | (0.4, 0.4, 0.15, 0.05) |

Tablica 3.2 Hiperparametri i podaci o zagrijavanju modela.




Graf 3.2 Funkcija gubitka (gore lijevo), te točnost (mIoU) na skupu podataka za učenje i validaciju preko 50 epoha zagrijavanja učitelja.



Slika 3.2 Primjer originalne slike (gore lijevo) i oznake ispitnog skupa (gore desno), kao i predikcija na tom primjeru prije (dolje lijevo) i nakon (dolje desno) zagrijavanja učitelja. Vidimo pogrešno segmentiranu anomaliju na donjem desnom rubu slike.

Iz grafa 3.2 iščitavamo da model dođe do najbolje performanse nakon 30 epoha te nakon toga krene u zasićenje gdje točnost na skupu za validaciju padne. Iz tog razloga ćemo učitelja trenirati na 30 epoha sa prethodno navedenom konfiguracijom iz tablice 2. Također možemo vidjeti da model nakon zagrijavanja postiže dobre performanse, kao što je primjer na slici 3.2, međutim i dalje se muči sa anomalijama. Težine za balansiranje oznaka u funkciji gubitka računate su prema inverzu relativne frekvencije piksela kako bi se više kaznile manje zastupane klase.

## 3.3. Treniranje modela nad pseudooznakama

U pseudokodu 3.1 prikazan je skica algoritma polunadziranog učenja u nekoliko koraka. Objašnjenja za varijable su sljedeća:

```
# ucitava klasu Dataloader za podskupove
sup_loader, unsup_loader, valid_loader, test_loader =
inicijaliziraj_podskupove()

# predtreniranje modela ucitelja (prethodno poglavlje)
predtreniranje(ucitelj, sup_loader, warmup_epohe, warmup_lr)

# konkatenira skupove za grupno treniranje
concat_loader = ConcatDataset(sup_loader, unsup_loader)

od 1 do BR_ITER:
      # ucitelj postaje najbolji student
      ucitaj_ucitelja()

      # ucitelj generira pseudooznake
      generiraj_pseudooznake(ucitelj, unsup_loader)

      # student i ucitelj se istovremeno treniraju
      train_joint(student, concat, epohe, unsup_lr)

      # racuna se uspjeh studenta na skupu za validaciju
      # kako bi se ustanovilo je li student nadmasio ucitelja
      valid_eval = evaluate(student, valid_loader)
```

```
ako valid_best < valid_eval
      spremi_ucenika()
      valid_best = valid_eval
```

Pseudokod 3.1 Prikaz predloženog algoritma polunadziranog učenja [7, 38].

Prolazi se kroz BR_ITER broja iteracija te se u svakom koraku učitelj preuzme ulogu najboljeg studenta, a student se ponovno postavlja na početno stanje – stanje učitelja nakon faze zagrijavanja. Nakon toga se student trenira na pseudooznačenom i označenom skupu podataka. Ti skupovi mogu međusobno biti u više omjera, čiji ćemo utjecaj testirati u eksperimentima.

| **Hiperparametar** | **Vrijednost** |
|---|---|
| openCLIP slikovni enkoder | convnext_base_w (pretrained laion2b) |
| SAM slikovni enkoder | vit_H |
| broj epoha | 50 |
| stopa učenja | (eta_max, eta_min) = (3e-7, 1e-8) |
| optimizator | AdamW (0.9, 0.999) |
| raspoređivač koraka učenja | kosinusno kaljenje |
| propadanje težina | 1e-8 |
| težine razreda | (0.1, 0.15, 0.15, 0.6) |

Tablica 3.3 Hiperparametri i podaci o treniranju modela studenta sa pseudooznakama.

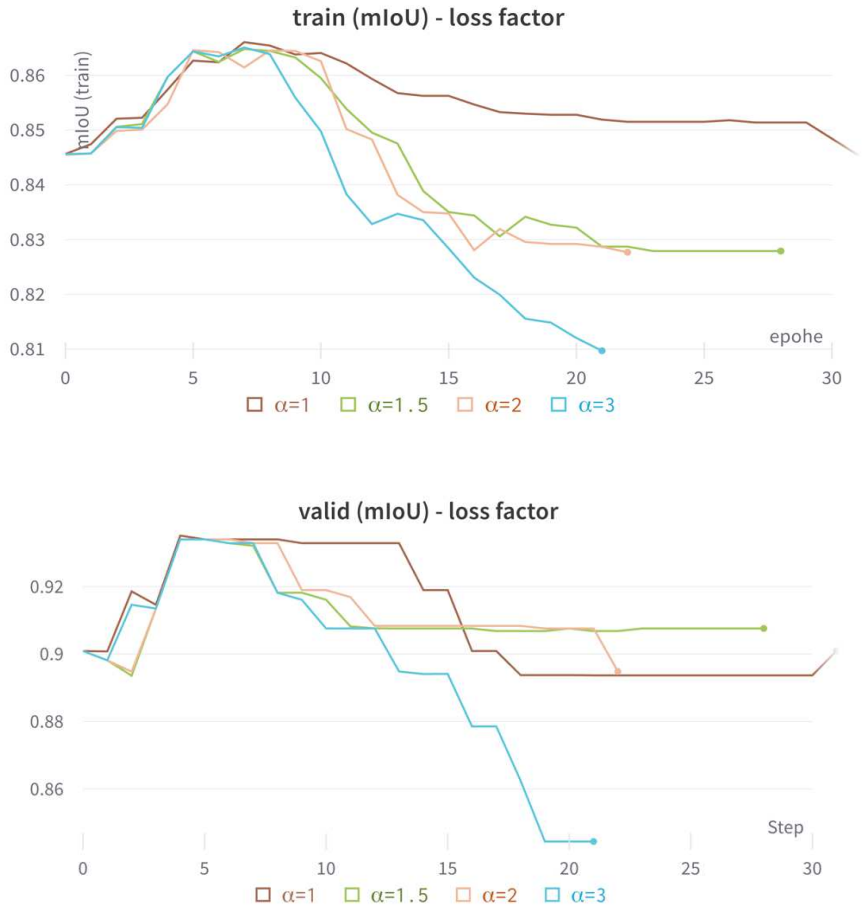
Kao funkciju gubitka koristimo zbroj unakrsne entropije na označenom i neoznačenom skupu podataka:

$$\frac{1}{n}\sum_{i=1}^{n} l\left(y_i, f^{šum}(x_i, \Theta^s)\right) + \alpha(t)\,\frac{1}{n}\sum_{i=1}^{m} l\left(f^{učitelj}(x_i, \Theta^t), f^{šum}(x_i, \Theta^s)\right) \tag{3.2}$$

gdje se za $\alpha(t)$ koristi sljedeća formula [38]:

$$\alpha(t) = \begin{cases} 0, & t < T_1 \\ \dfrac{t - T_1}{T_2 - T_1} * \alpha_f, & T_1 \leq t < T_2 \\ \alpha_f, & T_2 \leq t \end{cases} \tag{3.3}$$

Hiperparametri $T_1$, $T_2$ i $\alpha_f$ moraju biti pažljivo odabrani da šum pseudooznaka ne prevlada te točnost krene padati, kao što je vidljivo na grafu 3.3. Ako postavimo da gubitak pseudooznaka daje jaku predikciju, model će brzo pokupiti šum u pseudooznakama i divergirati. Vrijednosti $T_1$, $T_2$ se heuristički postavljaju na 20% i 80% broja ukupnih iteracija.

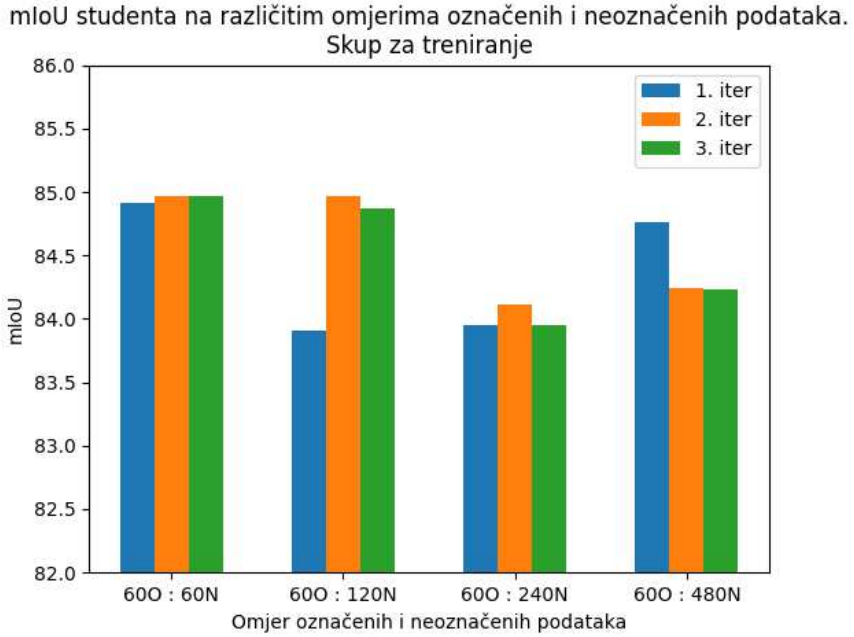


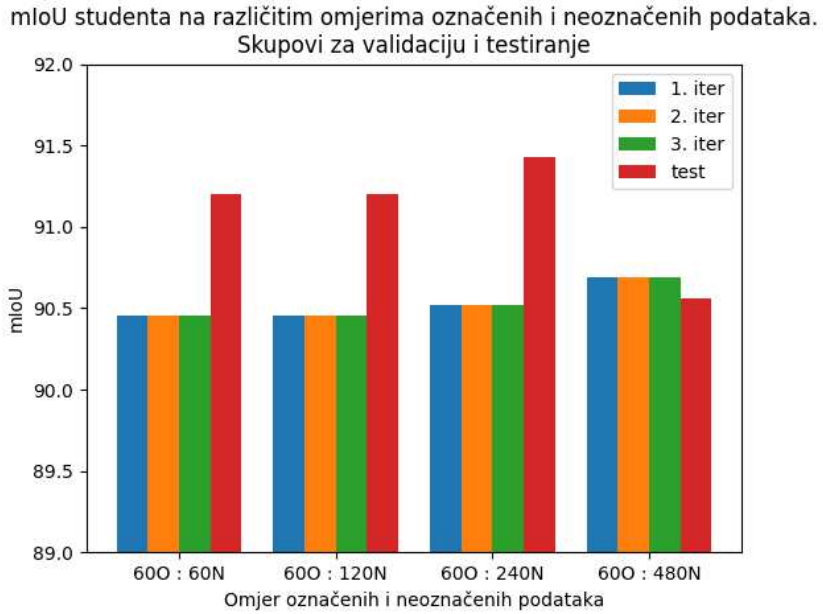

Graf 3.3 Učinak različitih hiperparametara $\alpha_f$ na performanse skupova podataka za učenje i validaciju.

Metrike nakon treniranja s različitim udjelima pseudooznaka dane su u tablici 3.4. Možemo primijetiti da je više iteracija pomoglo modelu da pronađe bolje rješenje. Dodatna poboljšanja bi uključivala dodavanje više šuma u model studenta, bilo kroz podatke ili model. Kroz podatke bi šum bilo moguće dodati augmentacijom ulaza, primjerice korištenjem RandAugment [42]. Nažalost, većina augmentacija iz RandAugment nisu bile moguće jer su SAM-maske generirane prije treniranja što ograničava augmentacije koje uključuju pomak ili rotaciju. Šum je također moguće dodavati putem modela metodama poput propadanja težina [43] ili stohastične dubine [44].

| | mIoU (train) | mIoU (valid) | mIoU (test) |
|---|---|---|---|
| inicijalno | 64.4105 | 47.9763 | 70.9361 |
| zagrijavanje | 83.6940 | 88.8560 | 70.9361 |
| | Omjer neoznačenih podataka 60O:60N | | |
| 1. iteracija | 84.9090 | **90.4554** | |
| 2. iteracija | **84.9665** | 90.4554 | |
| 3. iteracija | 84.9631 | 90.4554 | 91.1997 |
| | Omjer neoznačenih podataka 60O:120N | | |
| 1. iteracija | 83.9081 | **90.4554** | |
| 2. iteracija | **84.9622** | 90.4554 | |
| 3. iteracija | 84.8652 | 90.4554 | 91.1997 |
| | Omjer neoznačenih podataka 60O:240N | | |
| 1. iteracija | 83.9472 | **90.5196** | |
| 2. iteracija | **84.1165** | 90.5196 | |
| 3. iteracija | 83.9547 | 90.5196 | **91.4249** |
| | Omjer neoznačenih podataka 60O:480N | | |
| 1. iteracija | **84.7598** | **90.6886** | |
| 2. iteracija | 84.2446 | 90.6886 | |
| 3. iteracija | 84.2280 | 90.6886 | 90.5609 |

Tablica 3.4 Ukupno treniranje OpenClip-SAM-a sa različitim omjerima neoznačenih podataka




Graf 3.4. Treniranje pseudooznakama na studentu sa uključenim SAM-om na svim podskupovima. Primjećujemo da performansa na skupu za treniranje degradira dok validacija i testiranje generalno raste uz prisutnu anomaliju na skupu za testiranje pri omjeru 40O : 480N.

Na grafu 3.4 primjećujemo da performansa skupa za treniranje degradira dok na skupu za validaciju i testiranje raste. Rezultate možemo objasniti utjecajem pseudooznakama koje će svojom greškom uvesti lošije predikcije na skupu za treniranje, dok će bolje generalizirati. Međutim, na skupu za validaciju primjećujemo da se iznos validacije ne mijenja. Pretpostavka je da SAM-ove maske stvaraju stroge granice za mijenjanje predviđene klase te je zbog toga teže postići fine promjene na manjim skupovima. Model je nanovo treniran bez korištenja SAM-a prilikom evaluacije na validacijskom i testnom skupu. SAM se također ne koristi u studentu koji trenira na pseudooznakama. Drugim riječima, SAM je isključivo korišten prilikom finog podešavanja učitelja te generiranju pseudooznaka. U rezultatima tablice 3.4 vidljivo je da je validacijski skup promjenjiv jer granice SAM-ovih maski ne utječu na njegovu predikciju. Također je, zbog uklanjanja SAM-a tijekom predikcije, predviđena točnost manja.

| | mIoU (train) | mIoU (valid) | mIoU (test) |
|---|---|---|---|
| inicijalno | 33.3800 | 30.9646 | 34.9756 |
| zagrijavanje | 55.5797 | 56.8596 | 58.5013 |
| | Omjer neoznačenih podataka 60O:60N | | |
| 1. iteracija | **59.2863** | 59.5609 | |
| 2. iteracija | 59.2465 | **59.6018** | |
| 3. iteracija | 59.1431 | 59.1468 | 58.8884 |
| | Omjer neoznačenih podataka 60O:120N | | |
| 1. iteracija | 59.4219 | 59.3025 | |
| 2. iteracija | 59.3238 | 59.6713 | |
| 3. iteracija | **59.3773** | **59.7672** | 59.2697 |
| | Omjer neoznačenih podataka 60O:240N | | |
| 1. iteracija | 59.5312 | 59.7163 | |
| 2. iteracija | 59.5755 | 59.5090 | |
| 3. iteracija | **59.5870** | **59.7660** | **59.5526** |

Tablica 3.4 Treniranje OpenClip-SAM-a bez uporabe SAM-a na treniranju studenta te evaluaciji validacije i testnog skupa

# Zaključak

Predstavili smo novi model za semantičku segmentaciju kombiniranjem dva moćna postojeća modela računalnog vida, SAM i OpenCLIP. Pokazali smo da agregiranje preko generičkih segmentacijskih maski modela SAM pomaže modelu za ugrađivanje cijelih slika u jezični latentni prostor da ostvari znatno bolju segmentacijsku izvedbu od osnovice koja donosi nezavisne odluke u svakom pojedinom pikselu. Pri tome smo koristili jednostavan dekoder tako da guste predikcije dobivamo naduzorkovanjem slikovnih ugrađivanja. Tako pripremljeni model smo iskoristili za predviđanje pseudooznaka kako bismo dodatno poboljšali performanse modela, pri tome koristeći specijalizirani skup podataka kako bismo demonstrirali fino podešavanje modela na neviđenoj domeni. Vidimo da, unatoč velike količine neoznačenih podataka i novih klasa, danas postoje adekvatni modeli koji uz dovoljnu prilagodbu mogu postići visoke rezultate u neviđenim domenama.

# Literatura


[1] Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L, Xiao, T., Whitehead, S., Berg, A. C., Lo, W., Dollar, P., Girshick, R., *Segment Anything*, Meta, (2023.), CoRR abs/2304.02463, https://ai.facebook.com/research/publications/segment-anything/; pristupljeno 20. svibnja 2023.

[2] Cherti, M., Beaumont, R., Wightman, R., Wortsman, M. Ilharco, G., Gordon, C., Schuhmann, C., Schmidt, L., Jitsev, J., *Reproducible scaling laws for contrastive language-image learning*, (2022.), CoRR abs/2212.07143

[3] Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., Sustskever, I., *Learning Transferable Visual Models From Natural Language Supervision*, (2021.), ICML 2021: 8748-8763

[4] Lee, D. H. *Pseudo-Label : The Simple and Efficient Semi-Supervised Learning Method for Deep Neural Networks,* In Workshop on Challenges in Representation Learning, ICML. (2013).

[5] Zou, Y., Yu, Z., Kumar, B. V., Wang, J. *Unsupervised domain adaptation for semantic segmentation via class-balanced self-training.* In Proceedings of the European Conference on Computer Vision (ECCV). (2018.)

[6] Tarvainen, A., Valpola, H. *Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results*. ICLR, (2017.)

[7] Xie, Q., Luong, M. T., Hovy, E., & Le, Q. V. *Self-training with Noisy Student improves ImageNet classification.* CVPR 2020: 10684-10695, (2020.),

[8] Zhu, Y., Zhang, Z., Wu, C., Zhang, Z., He, T., Zhang, H., Manmatha, R., Li, M., Smola, A., *Improving Semantic Segmentation via Self-Training*, CoRR abs/2004.14960, (2020.)

[9] Yalniz, I.Z., Jgou, H., Chen, K., Paluri, M., Mahajan, D. *Billion-Scale SemiSupervised Learning for Image Classification*. (2019.), CoRR abs/1905.00546

[10] He, K., Chen, X., Xie, S., Li, Y., Dollar, P., Girshick, R. *Masked Autoencoders Are Scalable Vision Learners,* CVPR 2022: 15979-15988*,* (2022.)

[11] Englesson E., Azizpour H., *Consistency Regularization Can Improve Robustness to Label Noise,* CoRR abs/2100.01242, (2021.),

[12] Goodfellow, I., Bengio, Y., Courville, A. (2016.), *Deep Learning*, MIT Press

[13] Krizhevsky, A., Sutskever, I., Hinton, G. E., (2012). *Imagenet classification with deep convolution neural networks*, NIPS 2012: 1106-1114

[14] Dumoulin, V., Visin, F., *A guide to convolution arithmetic for deep learning*, CoRR abs/1603.07285, (2018.)

[15] Rawat, W., Wang, Z., *Deep Convolutional Networks for Image Classification: A Comprehensive Review*, (2017.) Neural Computation, Vol.29:9 2352-2449

[16] Simonyan, K., Zisserman, A., *Very Deep Convolutional Networks for Large-Scale Image Recognition,* ICLR 2015., (2015)

[17] He, K., Zhang, X., Ren., S., Sun, J., *Deep Residual Learning for Image Recognition*; CVPR 2016: 770-778, (2016.)

[18] Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K. Q., *Densely connected convolutional networks*, CVPR 2017: 2261-2269, (2017.)

[19] Dosovitskiy., A., Beyer., L., Kolesnikov, A., Weissenborn., D., Zhai., X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N., *An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale*, ICLR, (2021.)

[20] Liu, Z., Lin, Y., Cao, Y., Hu, H., Wei, Y., Zhang, Z., Lin, Z., Guo, B., *Swin Transformer: Hierarchical Vision Transformer using Shifted Windows*, ICCV 2021: 2929-2938, (2021.)

[21] Liu, Z., Mao, H., Wu., Chao-Yuan, Y., Feichtenhofer, C., Darrell, T., Xie, S., *A ConvNet for the 2020s*, CVPR 2022: 11966-11976, (2022.)

[22] Xie, S., Girshick, R., Dollar, P., Tu, Z., He, K., *Aggregated Residual Transformations for Deep Neural Networks*, CVPR 2017: 5987-5995, (2017.)

[23] Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.-C., *MobileNetV2: Inverted Residuals and Linear Bottlenecks*, CVPR 2018: 4510-4520, (2018.)

[24] Kumar, A., Shen, R., Bubeck, S., Gunasekar, S., *How to Fine Tune Vision Models with SGD*, (2022.), https://arxiv.org/pdf/2211.09359.pdf , pristupano 01.06.2023.

[25] Loshchilov, I., Hutter, F., *Decoupled Weight Decay Regularization*, University of Freiburg, ICLR, (2019.)

[26] Schuhmann, C., Beaumont, R., Vencu, R., Gordon, C., Coombes, T., Katta, A., Mullis, C., Wortmann, M., Schramowski, P., Kundurthy, S., Crowson, K., Schmidt, L., Kaczmarczyk, R., Jitsev, J., *LAION-5B: An open large-scale dataset for training next generation image-text models*, NeurIPS, (2022.)

[27] Hu, R., Rohrbach, M., Darrell, T., *Segmentation from Natural Language Expressions*, ECCV 2016: 108-124, (2016.)

[28] Chaitanya, K., Karani, N., Baumgartner, Becker, A., Donati, O., Konukoglu, E., *Semi-Supervised and Data Driven Augmentation*, IPMI 2019: 29-41, (2019.)

[29] Grandvalet, Y., Bengio, Y., *Semi-Supervised Learning by Entropy Minimization*, Advances in Neural Information Processing Systems, CAP 2005: 281-296, (2005.)

[30] Shen, S., Harold Li, L., Tan, H., Bansal, M., Rohrbach, A., Chang, K., Yao, Z., Keutzer, K., *How Much Can CLIP Benefit Vision-and-Language Tasks?*, ICLR 2022, (2022.)

[31] Tao, M., Bao B.-K., Tang, H., Chu, H., *GALIP: Generative Adversarial CLIPs for Text-to-Image Synthesis*, CoRR abs/2301.01217, (2023.)

[32] Chakrabarty, N., Chatterjee, S., *A Novel Approach to Age Classification from Hand Dorsal Images using Computer Vision*, ICCMC.2019.8819432, (2019.),

[33] Kadam, R., *Difference Between Channels and Kernels in Deep Learning*, Medium, (2021.), https://medium.com/analytics-vidhya/difference-between-channels-and-kernels-in-deep-learning-6db818038a11, pristupano 15.05.2023.

[34] Park, J.B., *Towards a Meaningful 3D Map Using a 3D Lidar and a Camera*, Sensors 18(8):2571, (2018.)

[35] Long, J., Shelhamer, E., Darrel, T., *Fully Convolutional Neural Networks for Semantic Segmentation*, CVPR 2015: 3431-3440, (2015.),

[36] Ronneberger, O., Fischer, P., Brox, T., *U-Net: Convolutional Networks for Biomedical Image Segmentation*, MICCAI 2015:234-241, (2015.)

[37] Fu, J. Liu, J., Tian, H., Li, Y., Bao, Y. Fang, Z., Lu, H., *Dual Attention Network for Scene Segmentation*, CVPR 2019: 3146-3154, (2019.)

[38] Petrač, T., *Polunadzirana semantička segmentacija temeljena na pseudooznačavanju*, FER, (2021.), http://www.zemris.fer.hr/~ssegvic/project/pubs/petrac21ms.pdf, pristupano 15.05.2023.

[39] Everingham, M., Gool, L. V., Williams, C., K. I., Winn, J., Zisserman, A., *The PASCAL Visual Object Classes (VOC) Challenge*, (2012.), https://homepages.inf.ed.ac.uk/ckiw/postscript/ijcv_voc09.pdf, pristupano 16.06.2023.

[40] Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B., *The Cityscapes Dataset for Semantic Urban Scene Understanding,* CVPR 2016: 3213-3223, (2016.)

[41] Kukil, Intersection over Union (IoU) in Object Detection & Segmentation, (2022.), https://learnopencv.com/intersection-over-union-iou-in-object-detection-and-segmentation/, pristupano 10.06.2023.

[42] Cubuk, E., Zoph, B., Shlens, J., Le, Q., *RandAugment: Practical automated data augmentation with a reduced search space*, NeurIPS 2020, (2020.)

[43] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R., *Dropout: A Simple Way to Prevent Neural Networks from Overfitting*, J. Mach. Learn. Res. 15(1): 1929-1958, (2014.)

[44] Huang, G., Sun, Y., Liu, Z., Sedra, D., Weinberger, K., *Deep Networks with Stochastic Depth*, ECCV 2016:646, (2016.)

[45] Grubišić, I., Oršić, M., Šegvić, S., *Revisiting Consistency for Semi-Supervised Semantic Segmentation*, Sensors, (2023.), 23, 940., https://doi.org/10.3390/s23020940, pristupano 20.06.2023.

[46] Ilharco G., Wortsman, M., Wightman, R., G., Cade, Carlini, N., Taori, Dave, A., Shankar, V., Namkoong, H., Miller, J., Hajishirzi, H., Farhadi, A., Schmidt, L., *Openclip*, (2021.)

[47] Lin, T.., Goyal, P., Girshick, R., He, K., Dollar, P., *Focal loss for dense object detection*. ICCV 2017: 2999-3007, (2017.)

[48] Milletari, F., Navab, N., Ahmadi, S.-A., *V-Net: Fully convolutional neural networks for volumetric medical image segmentation*, 3DV 2016: 565-571, (2016.)

[49] Kirillov, A., Mintun, E., Ravi, N., Mao, H., *Segment Anything*, FAIR, https://github.com/facebookresearch/segment-anything, pristupano 20.06.2023.

# Sažetak

**Učenje semantičke segmentacije na nepotpunim oznakama**

Semantička segmentacija važan je zadatak računalnog vida koji zahtjeva veliku količinu označenih podataka sa kojima često ne raspolaže. Iz tog razloga su smišljene tehnike polunadziranog strojnog učenja koje bi iskoristile moć neoznačenih ili slabo označenih podataka kako bi stvorili induktivne pretpostavke o čitavoj distribuciji. Ovaj rad bavi se uporabom kombinacije postojećih modela kako bi se ostvarili bolji rezultati na osobnom skupu podataka. Demonstriraju se moderni alati i tehnike učenja kako bi se ostvarila zadovoljavajuća točnost uz pomoć snage neoznačenih podataka.

**Ključne riječi:** duboko učenje, semantička segmentacija, polunadzirano učenje, prijenos domene, učenje nad nepotpunim oznakama, pseudooznačavanje, OpenCLIP, Segment Anything

# Summary

**Learning weakly supervised semantic segmentation**

Semantic segmentation is an important computer vision task that requires a great deal of annotated data, which is seldom the case. For this reason, weakly supervised techniques have been developed for leveraging the power of unlabeled and weakly labeled data to make inductive reasoning about the underlying data distribution. This work is based on using a combination of powerful models to achieve better results on a custom dataset. Modern pseudolabeling techniques are used to leverage the power of unlabeled and weakly annotated data to achieve sufficient accuracy.

**Keywords:** deep learning, semantic segmentation, semi-supervised learning, domain transfer, weakly supervised learning, pseudolabeling, OpenCLIP, Segment Anything