SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA OSIJEK

Sveučilišni studij

Aplikacija za segmentaciju slike Segment Anything modelom

Projekt

Marko Budak

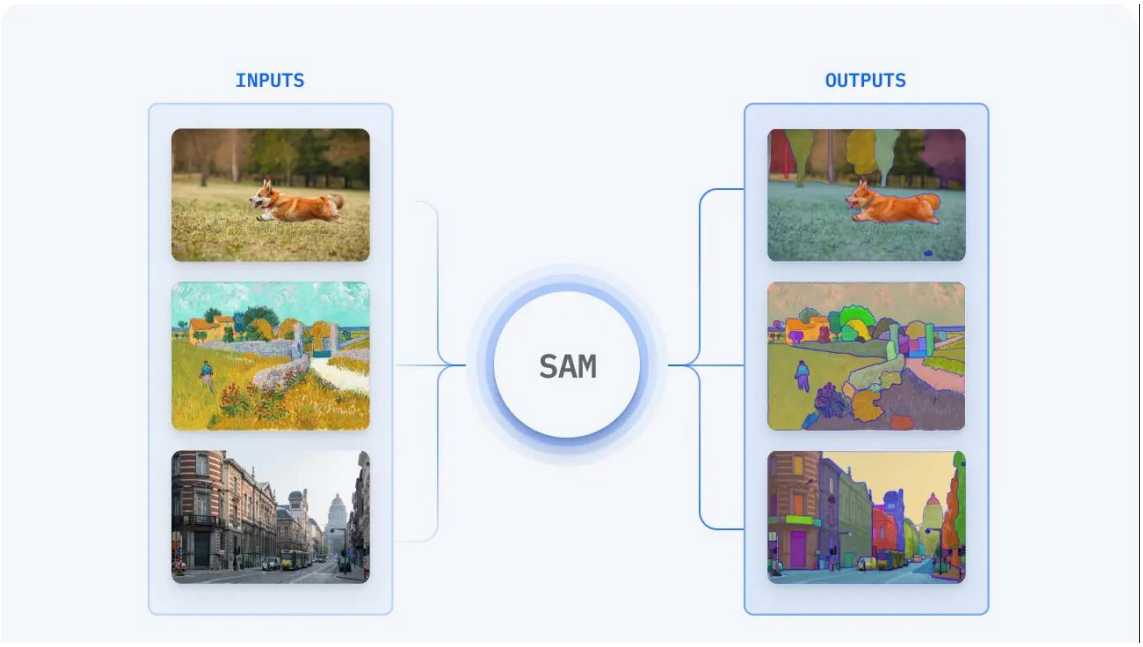
Osijek, 2024.

SADRŽAJ

SADRŽAJ

1. UVOD

Segmentacija slike predstavlja ključni aspekt u računalnom vidu, omogućujući analizu i razumijevanje strukture slike kroz identifikaciju i razdvajanje različitih regija ili objekata. U svijetu dubokog učenja, Segment Anything model predstavlja inovativan pristup ovom zadatku. Ovaj model, zasnovan na dubokim konvolucijskim neuronskim mrežama (CNN), ističe se svojom sposobnošću da precizno segmentira raznovrsne objekte unutar slika, bez obzira na njihovu formu, boju ili veličinu. Što čini Segment Anything model jedinstvenim je njegova sposobnost da segmentira gotovo bilo što unutar slike, uključujući objekte različitih oblika, tekstura i konteksta. Ovaj model nije ograničen specifičnim kategorijama objekata ili okolišima te demonstrira izuzetnu robusnost i generalizaciju u segmentaciji različitih scena. U kontekstu razvoja aplikacija za segmentaciju slika, razumijevanje i primjena Segment Anything modela otvara nove mogućnosti u različitim područjima. Primjene u medicinskoj dijagnostici, analizi satelitskih slika, industriji, sigurnosti i mnogim drugim sektorima postaju izvedive s ovim naprednim pristupom segmentaciji slika.



Slika 1: Primjer rada SAM-a

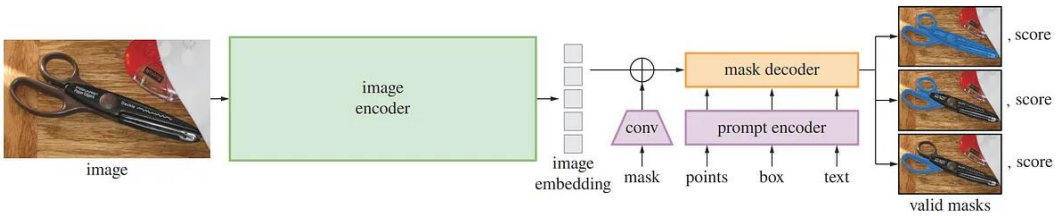
1.1 Zadatak rada

Razviti aplikaciju gdje korisnik može učitati sliku i na njoj nacrtati regiju objekta kojeg želi segmentirati. Sa strane prikazati rezultat segmentacije kao obrub oko segmentiranog objekta na slici. Za segmentaciju je potrebno koristiti Segment Anything model (link) tako što se predtrenirani model pokrene na vlastitom računalu i regija koju korisnik nacrta se modelu daje kao prompt. Detaljno opisati način rada Segment Anything modela.

2. ARHITEKTURA I KARAKTERISTIKE

Segment Anything Model, za razliku od konvencionalnih modela segmentacije slike koji zahtijevaju značajnu stručnost u modeliranju specifičnih zadataka, uklanja potrebu za takvim znanjem. Njegov je temeljni cilj pojednostaviti proces segmentacije djelujući kao model sposoban za rukovanje različitim unosima poput klikova, okvira ili teksta[6]. Ova karakteristika proširuje njegovu dostupnost, opskrbljujući širi niz korisnika i aplikacija..

Model se sastoji od: Koder slike (*Image encoder*), Prompt koder *(Prompt encoder)* koji prima različite vrste *prompta* i Dekoder maske (*Mask decoder*) za izradu maske [3].



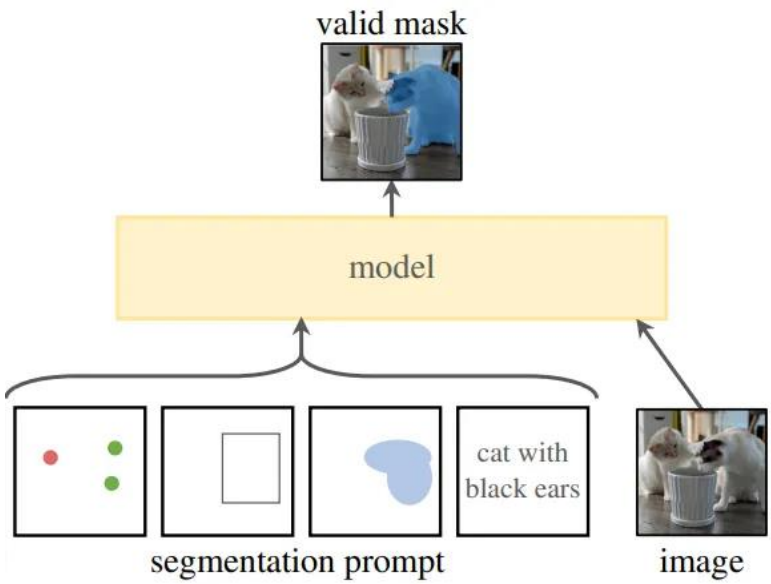
Slika 2: Arhitektura SAM-a

2.1 Koder slike

Središnja komponenta modela sastoji se od maskiranog auto enkodera uparenog s transformatorom vida za postizanje vrhunske skalabilnosti. Koristi se prethodno trenirani *Vision Transofmer* (ViT) [8] kako bi procesuirali slike velikih rezolucija.. Krajnji rezultat ovog kodera je ugrađivanje značajki, koje predstavlja smanjenu verziju izvorne slike faktorom 16x. Ova operacija smanjivanja je ključna za pojednostavljenu obradu uz zadržavanje ključnih karakteristika slike. Koder slike se koristi samo jednom po slici te se može pozvati prije slanja ikakvog *prompt-a* modelu [6].

2.2 Prompt koder

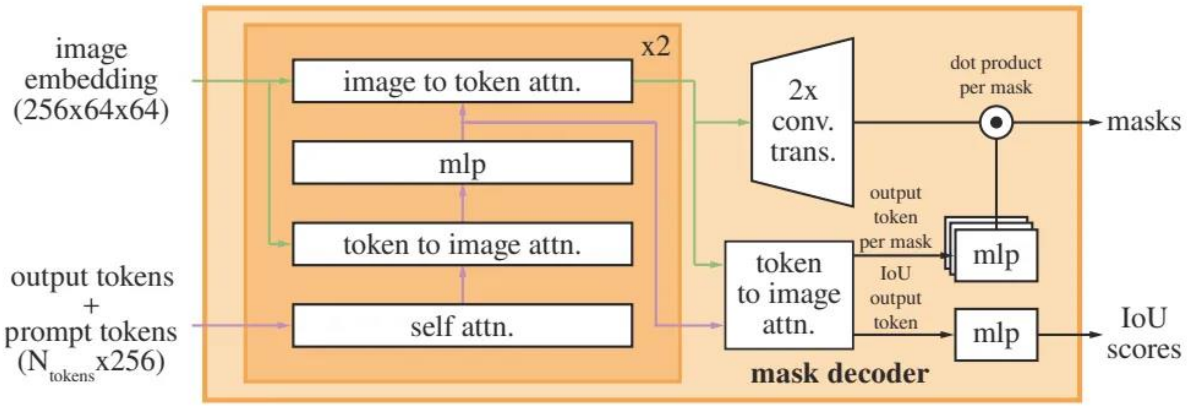
U modelu se koriste različite vrste upita, uključujući točke, okvire, maske i tekst. Točke i okviri su predstavljeni kombinacijom položajnog kodiranja [9] i naučenih *embeddinga,* dok tekstualni upiti koriste CLIP [10], dopuštajući korištenje bilo kojeg kodera teksta. *Promptovi* poput maski se ugrađuju pomoću konvolucija te imaju prostornu povezanost s slikom.



Slika 3: Vizualizacija Prompt kodera

2.3 Dekoder maske

Ovdje se ugrađivanja slika i brza ugrađivanja preslikavaju na konačnu masku. Da bi to izradili, koristi se modificirani Transformerov dekoder [11]. Ključni aspekt ovog procesa uključuje uvođenje ugradnje naučenog izlaznog tokena u promptno ugrađivanje prije nego što ga obradi dekoder. Ovo ugrađivanje izlaznog tokena igra ključnu ulogu u funkciji dekodera, sadržavajući bitne informacije potrebne za cjelokupni zadatak segmentacije slike.



Slika 4: Vizualizacija Dekoder maske

2.4 UČINKOVITOST

Jedna od najvažnijih karakteristika Segmenty Anything Modela je njegova učinkovitost [6]. S obzirom na prethodno izračunatu ugniježđenu sliku, prompt enkoder i dekoder maske izvršavaju se u web pregledniku, na CPU-u, u 50 milisekundi. Ova brzina izvršavanja u stvarnom vremenu omogućuje neprekidno, interaktivno korištenje upita modela.

3. CNN i CLIP

3.1 Konvolucijske neruonske mreže (CNN)

Konvolucijske neuronske mreže [12] su sastavni dio kodera slike arhitekture Segment Anything Model. Izvrsni su u prepoznavanju uzoraka na slikama učeći prostornu hijerarhiju značajki, od jednostavnih rubova do složenijih oblika. U SAM-u, CNN-ovi analiziraju i interpretiraju vizualne podatke, učinkovito obrađujući piksele kako bi otkrili i razumjeli različite značajke i objekte unutar slike. Ova sposobnost je ključna za početnu fazu analize slike SAM-ove arhitekture.

3.4 CLIP

CLIP [10], koji je razvio OpenAI, model je koji smanjuje različitost između teksta i slika. Sposobnost CLIP-a da razumije i tumači tekstualne upute u odnosu na slike od neprocjenjive je važnosti za rad SAM-a. Omogućuje SAM-u da obradi i odgovori na tekstualne unose, kao što su opisi ili oznake, i da ih točno poveže s vizualnim podacima. Ova integracija poboljšava SAM-ovu svestranost, omogućujući mu segmentiranje slika na temelju vizualnih znakova i praćenje tekstualnih uputa

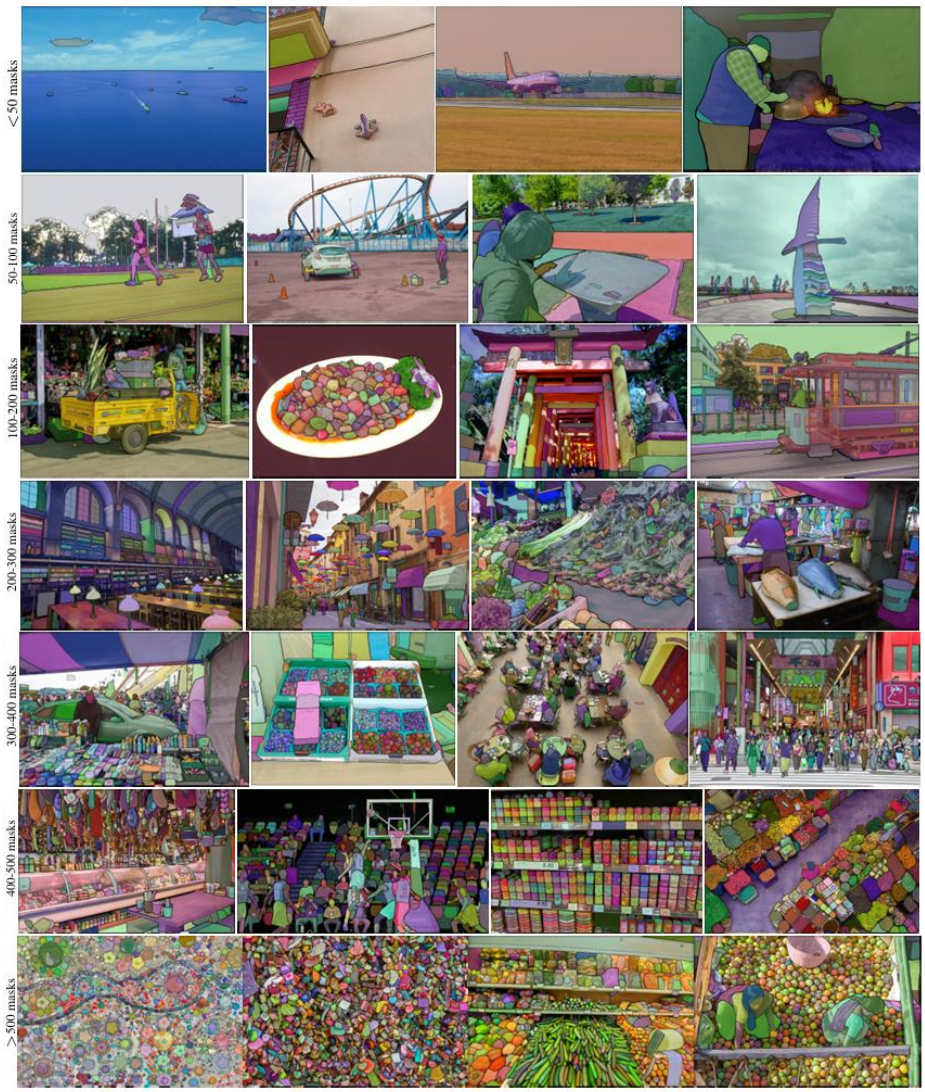
4. TRANSFERNI MODELI UČENJA I PRETHODNO TRENIRANI MODELI

Transferno učenje [13] uključuje korištenje modela uvježbanog na jednom zadatku kao temelj za drugi srodni zadatak. Unaprijed trenirani modeli kao što su ResNet i VGG [14], koji su opsežno trenirani na velikim skupovima podataka, najbolji su primjeri toga. ResNet je poznat po svojoj dubokoj mrežnoj arhitekturi, koja rješava problem nestajanja gradijenta, dopuštajući mu da uči iz ogromne količine vizualnih podataka. VGG je cijenjen zbog svoje jednostavnosti i učinkovitosti u zadacima prepoznavanja slika. Koristeći prijenos učenja i unaprijed obučene modele, SAM dobiva prednost u razumijevanju složenih značajki slike. To je bitno za njegovu visoku točnost i učinkovitost u segmentaciji slike.

Slika 5: Segment Anything Model demo, detekcija se izvodi bez potrebe za označavanjem slika

5. SA-1B DATASET

SAM-ova izvanredna prednost leži u njegovom ogromnom treniranom skupu podataka nazvanom SA-1B Dataset [7]. Skraćeno od Segment Anything 1 Billion, to je najopsežniji i najraznovrsniji skup podataka o segmentaciji slika koji je dostupan. Obuhvaća više od 1 milijarde visokokvalitetnih segmentacijskih maski izvedenih iz golemog niza od 11 milijuna slika i pokriva široki spektar scenarija, objekata i okruženja. Stvaranje ovog skupa podataka zahtijevalo je više faza proizvodnje: faza ručnog označavanja uz pomoć modela, poluautomatska faza s kombinacijom automatski predviđenih maski i označavanja uz pomoć modela, te potpuno automatska faza u kojoj model generira maske bez ulaza. [1].



Slika 6: Primjeri slika s preklapanjem maski iz našeg novo uvedenog skupa podataka, SA-1B.

5.1. Faza ručne anotacije uz pomoć modela

U prvoj fazi [6], ljudi su označavali maske koristeći alat u pregledniku koji je pokretan od strane SAM-a. Ovo modelom potpomognuta označavanje omogućilo je interakciju u stvarnom vremenu, dopuštajući označavanje objekata bez nametanja semantičkih ograničenja. Ukoliko je označavanje trajalo duže od 30 sekundi, prelazilo se na sljedeću sliku. SAM je prvotno treniran na javnim skupovima podataka i ponovno treniran s novo označenim maskama. Kako se model poboljšavao, vrijeme označavanja smanjivalo se sa 34 na 14 sekundi po maski. Prosječni broj maski po slici povećao se sa 20 na 44. Ukupno je prikupljeno 4,3 milijuna maski iz 120.000 slika.

5.2 Poluautomatska faza

U ovoj fazi [6], cilj je bio povećati raznolikost maski kako bi se poboljšala mogućnost modela da segmentira bilo što. Prvi korak je bio da se automatski prepoznaju maske koje je model sa visokom razinom sigurnosti točno prepoznao. Za uspješnost takvog automatskog prepoznavanja, treniran je *bounding box detector* [15] na svim maskama iz prve faze. Nakon toga smo zaposlenicima dali slike prepune takvih maski te ih tražili da označe objekte koje model sam nije uspio označiti. U ovoj fazi se prikupilo još 5.9 milijuna maski iz 180 tisuća slika te se postigao ukupni broj maski od 10.2 milijuna.

5.3 Potpuno automatska faza

U posljednjoj fazi [6], označavanje se odrađivalo automatski. Ovo su omogućila dva velika poboljšanja modela. Prikupljeno je dovoljno maski kako bi se poboljšala učinkovitost modela, uzimajući maske iz prošlih faza. Također, model se razvio da sam može predvidjeti maske. Ako se točka nalazi na manjem dijelu nekog objekta, model će nam vratiti i manji dio tog objekta i cijeli objekt. Pomoću *Non-maximum Suppression-a (NMS)*[16], filtriraju se svi duplikati maski. Potpuno automatsko generiranje maski se primijenilo na svih 11 milijuna slika podatkovnog skupa stvarajući 1.1 milijardu maski visoke rezolucije.

6. PRIMJERI KORIŠTENJA SEGMENT ANYTHING MODELA

SAM se može koristiti u gotovo svakoj mogućoj primjeni. Evo nekih od aplikacija u kojima se SAM pokazao kao najbolje rješenje za problem:

- Označavanje potpomognuto umjetnom inteligencijom: SAM značajno pojednostavljuje proces označavanja slika. Može automatski identificirati i segmentirati objekte unutar slika, drastično smanjujući vrijeme i trud potreban za ručno označavanje.
- Dostava lijekova: U zdravstvu, SAM-ove mogućnosti precizne segmentacije omogućuju identifikaciju specifičnih regija za isporuku lijekova. Time se osigurava preciznost u liječenju i minimiziraju nuspojave.
- Mapiranje zemljišnog pokrova: SAM se može koristiti za klasificiranje i kartiranje različitih tipova zemljišnog pokrova, što omogućuje primjenu u urbanom planiranju, poljoprivredi i praćenju okoliša.

7. PROGRAMSKO RJEŠENJE

7.1. Postavljanje okruženja za rad

Prije početka pisanja programskog rješenja, moramo u terminalu pomoću *pip install 'git+https://github.com/facebookresearch/segment-anything.git'* instalirati Segment Anything model (SAM) putem META-inog Git repozitorija [2]. Sljedeći korak je uvođenje svih potrebnih biblioteka te provjera da sve putanje potrebne za pristup SAM-u postoje. Također je i definirana funkcija *upload_image()* koja korisniku omogućuje uvoz slike koju želi segmentirati.

```python
# BITNO!! INSTALIRATI NAVEDENO U TERMINALU: pip install 'git+https://github.com/facebookresearch/segment-anything.git'
import os
import cv2
import tkinter as tk
from tkinter import filedialog
import numpy as np
import torch
from segment_anything import sam_model_registry, SamAutomaticMaskGenerator, SamPredictor
import matplotlib.pyplot as plt
import supervision as sv

DEVICE = torch.device('cuda:0' if torch.cuda.is_available() else 'cpu')
MODEL_TYPE = "vit_h"
CHECKPOINT_PATH = "sam_vit_h_4b8939.pth"
print(CHECKPOINT_PATH, "; exist:", os.path.isfile(CHECKPOINT_PATH))

sam = sam_model_registry[MODEL_TYPE](checkpoint=CHECKPOINT_PATH).to(device=DEVICE)

predictor = SamPredictor(sam)
mask_generator = SamAutomaticMaskGenerator(sam)

def upload_image():
    root = tk.Tk()
    root.withdraw()
    file_path = filedialog.askopenfilename(title="Select Image File")
    return file_path

print("Upload an image:")
IMAGE_PATH = upload_image()
if IMAGE_PATH:
    img = cv2.imread(IMAGE_PATH)

else:
    print("No image uploaded.")

points = []
segmented_masks = []
```

Slika 7: Postavljanje okruženja za rad

7.2. Funkcija *mouse_click()*

Cilj ovog projekta da korisnik može interaktivno označavati dijelove slike za segmentaciju, stoga se pomoću *OpenCV* definira *funkcija mouse_click()* koja korisniku omogućuje korištenje desnog i lijevog klika miša na sliku. Lijevim klikom miša se označava koji dio slike se želi segmentirati, dok se desnim klikom označava gdje korisnik želi da prva segmentacija završi. Na jednoj slici se može označiti proizvoljno segmenata. Lokacija svakog klika miša se sprema te se oko njih kreira segmentacijska maska.

```
36  points = []
37  segmented_masks = []
38
39  def mouse_click(event, x, y, flags, param):
40      global points, segmented_masks
41
42      if event == cv2.EVENT_LBUTTONDOWN:
43          points.append((x, y))
44          cv2.circle(image_rgb, (x, y), 10, (255, 0, 0), -1)
45          cv2.imshow('image', image_rgb)
46      elif event == cv2.EVENT_RBUTTONDOWN:
47          if len(points) >= 2:
48              segmented_masks.append(np.array(points))  #korisnicki oznacene tocke se spremaju kao maska
49              points = []
50              cv2.circle(image_rgb, (x, y), 10, (0, 0, 255), -1)
51              cv2.imshow('image', image_rgb)
52          else:
53              print("At least two points are needed to define a bounding box.")
54              points = []
55
56  image_bgr = cv2.imread(IMAGE_PATH)
57  image_rgb = cv2.cvtColor(image_bgr, cv2.COLOR_BGR2RGB)
58
59  cv2.imshow('image', image_rgb)
60  cv2.setMouseCallback('image', mouse_click)
61  cv2.waitKey(0)
62  cv2.destroyAllWindows()
63
```

Slika 8: Funkcija mouse_click()

7.3. Segmentiranje i način prikazivanja rezultata

Pomoću for petlje prolazimo kroz sve segmentirane maske, gdje svaka maska predstavlja jednu regiju objekta na slici. Unutar petlje, računaju se minimalni i maksimalni koordinatni položaji (x_min, y_min, x_max, y_max) za svaku masku. Ovi podaci se koriste za stvaranje okvira oko svake detektirane regije. Pomoću *predictor.predict()* funkcije, pozivamo Segment Anything model za predikciju maski za određeni okvir koji smo definirali s našim minimalnim i maksimalnim vrijednostima. Instance MaskAnnotator regije maske označava u ovom slučaju sa crvenom bojom. Kako korisnik ne bi imao jednu sliku za svaku segmentaciju, dodana je OpenCV funkcija *addWeighted*() koja spaja sve slike u jednu.

```
for mask_points in segmented_masks:
    x_min = min([point[0] for point in mask_points])
    y_min = min([point[1] for point in mask_points])
    x_max = max([point[0] for point in mask_points])
    y_max = max([point[1] for point in mask_points])
    box = np.array([x_min, y_min, x_max, y_max])

    predictor.set_image(image_rgb)

    masks, _, _ = predictor.predict(box=box, multimask_output=True)

    mask_annotator = sv.MaskAnnotator(color=sv.Color.RED, color_lookup=sv.ColorLookup.INDEX)

    detections = sv.Detections(
        xyxy=sv.mask_to_xyxy(masks=masks),
        mask=masks
    )
    detections = detections[detections.area == np.max(detections.area)]
    segmented_image = mask_annotator.annotate(scene=image_bgr.copy(), detections=detections)

    # Da ne bude vise segmentiranih slika, dodano da sve slike segmentiranja spoje u jednu
    alpha = 0.5
    cv2.addWeighted(segmented_image, alpha, image_bgr, 1 - alpha, 0, image_bgr)

cv2.imshow('Segmented Image', image_bgr)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

Slika 9: Segmentiranje i način prikazivanja rezultata

7.4. Prikaz rezultata

Slika 10: Primjer korisničko označene slike

Slika 11: Primjer rezultata

8. ZAKLJUČAK

Segment Anything Model (SAM) predstavlja revolucionaran pristup segmentaciji slika, koristeći napredne tehnike poput maskiranih autoenkodera i vizualnih transformatora. Kroz svoju inovativnu arhitekturu i opsežan skup podataka, SAM pokazuje izvanrednu skalabilnost i prilagodljivost različitim zadacima segmentacije. Njegove široko primjenjive mogućnosti, od kreiranja sadržaja do znanstvenih istraživanja, naglašavaju njegovu svestranost i učinkovitost u rješavanju složenih izazova segmentacije slika. Njegova jednostavnost korištenja, brzina i točnost rezultata otvara nove mogućnosti za učinkovitu i preciznu segmentaciju visoko razlučivih slika. SAM predstavlja značajan napredak u području računalnog vida, nudeći pristupačno rješenje za širok spektar stvarnih primjena.

9. LITERATURA

[1] Segment Anything Model (SAM) – The Complete 2024 Guide Read [online], dostpuno na: https://viso.ai/deep-learning/segment-anything-model-sam-explained/

[2] How to Use the Segment Anything Model (SAM) [online], dostpuno na: https://blog.roboflow.com/how-to-use-segment-anything-model-sam/

[3] Segment Anything Model (SAM): Intro, Use Cases, V7 Tutorial [online], dostpuno na: https://www.v7labs.com/blog/segment-anything-model-sam

[4] Segment Anything Model (SAM): Explained [online], dostupno na: https://medium.com/@utkarsh135/segment-anything-model-sam-explained-2900743cb61e

[5] Segment Anything Research by META AI [online], dostupno na: https://segment-anything.com/

[6] Segment Anything, Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, Ross Girshick: https://doi.org/10.48550/arXiv.2304.02643

[7] Segment Anything model, SA-1B Dataset Explorer, dostupno na: https://segment-anything.com/dataset/index.html

[8] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa De hghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. ICLR, 2021.

[9] Matthew Tancik, Pratul Srinivasan, Ben Mildenhall, Sara Fridovich-Keil, Nithin Raghavan, Utkarsh Singhal, Ravi Ra mamoorthi, Jonathan Barron, and Ren Ng. Fourier features let net works learn high frequency functions in low dimensional domains. NeurIPS, 2020.

[10] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. ICML, 2021.

[11] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. NeurIPS, 2017.

[12] ZhaoweiCaiandNunoVasconcelos. CascadeR-CNN:Delvinginto high quality object detection. CVPR, 2018.

[13] A Comprehensive Survey on Transfer Learning Fuzhen Zhuang, Zhiyuan Qi, Keyu Duan, Dongbo Xi, Yongchun Zhu, Hengshu Zhu, Hui Xiong, Qing He:

https://doi.org/10.48550/arXiv.1911.02685

[14] Deep Residual Learning for Image Recognition Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun: https://doi.org/10.48550/arXiv.1512.03385

[15] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. NeurIPS, 2015.

[16] Learning non-maximum suppression Jan Hosang, Rodrigo Benenson, Bernt Schiele: https://doi.org/10.48550/arXiv.1705.02950