



Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет
имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ

КАФЕДРА СИСТЕМЫ ОБРАБОТКИ ИНФОРМАЦИИ И УПРАВЛЕНИЯ

РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА

К НАУЧНО-ИССЛЕДОВАТЕЛЬСКОЙ РАБОТЕ

НА ТЕМУ:

Исследование возможности применения нейронных сетей для задачи zero-shot сегментации

Студент ИУ5-32М
(Группа)

(Подпись, дата) (И.О.Фамилия)

Руководитель

Ю.Е. Гапанюк

(Подпись, дата)

(И.О.Фамилия)

2023 г.

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет имени Н.Э. Баумана
(национальный исследовательский университет)
(МГТУ им. Н.Э. Баумана)

УТВЕРЖДАЮ
Заведующий кафедрой ИУ5
(Индекс)
_____ В.И. Терехов
(И.О.Фамилия)
« 04 » сентября 2023 г.

З А Д А Н И Е на выполнение научно-исследовательской работы

по теме Исследование возможности применения нейронных сетей для задачи zero-shot сегментации

Студент группы ИУ5-32М

(Фамилия, имя, отчество)

Направленность НИР (учебная, исследовательская, практическая, производственная, др.)
ИССЛЕДОВАТЕЛЬСКАЯ

Источник тематики (кафедра, предприятие, НИР) КАФЕДРА

График выполнения НИР: 25% к ____ нед., 50% к ____ нед., 75% к ____ нед., 100% к ____ нед.

Техническое задание исследовать возможности нейронных сетей, объединяющих в себе обработку изображений и текста на естественном языке, с целью решения задачи zero-shot сегментации

Оформление научно-исследовательской работы:

Расчетно-пояснительная записка на 22 листах формата А4.

Перечень графического (иллюстративного) материала (чертежи, плакаты, слайды и т.п.)

Дата выдачи задания « 04 » сентября 2023 г.

Руководитель НИР

_____ Ю.Е. Гапанюк
(Подпись, дата) _____ (И.О.Фамилия)

Студент

_____ (Подпись, дата) _____ (И.О.Фамилия)

Примечание: Задание оформляется в двух экземплярах: один выдается студенту, второй хранится на кафедре.

Оглавление

1. Введение.....	2
2. CLIP.....	2
3. CLIPSeg.....	4
4. SAM.....	7
5. DINO.....	9
6. Grounding DINO.....	9
7. Ансамбль моделей Grounding DINO и SAM.....	11
8. Заключение.....	21
9. Список литературы.....	22

1. Введение

Zero-shot задачи — это задачи, в которых модель может получить на вход классы, которые не встречала на стадии обучения. Например, задача zero-shot классификации может получить на вход изображение и набор описаний на естественном языке, из которого выбрать наиболее подходящее. Эти описания не ограничены классами, которые входили в обучающую выборку.

Задача zero-shot сегментации подразумевает генерацию маски для объектов, соответствующих текстовому описанию на естественном языке. В zero-shot сегментации модель сначала обрабатывает текстовое описание, чтобы получить его векторное представление, которое затем используется для предсказания маски объектов этого класса на изображении. Отдельные текстовые описания могут содержать информацию о форме, размере, цвете или других атрибутах объектов.

2. CLIP

Из репозитория OpenAI: «CLIP[1] (Contrastive Language-Image Pre-Training) - это нейронная сеть, обученная на различных парах (изображение, текст). Ей можно дать указание выбрать наиболее релевантный фрагмент текста на естественном языке по изображению без непосредственной оптимизации для этой задачи, аналогично zero-shot возможностям GPT-2 и GPT-3»

Решаемые задачи: zero-shot классификация.

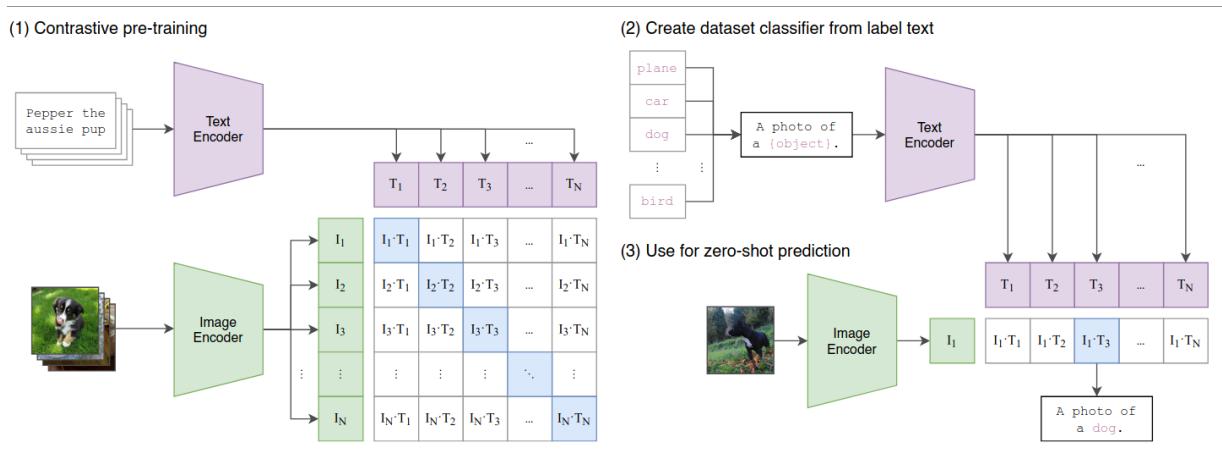


Рис 1. Архитектура CLIP

CLIP в своей работе не использует предопределенный набор классов, вместо этого принимая на вход произвольный текст на естественном языке, который обрабатывается энкодером текста. Это дает возможность использовать CLIP для zero-shot задач.

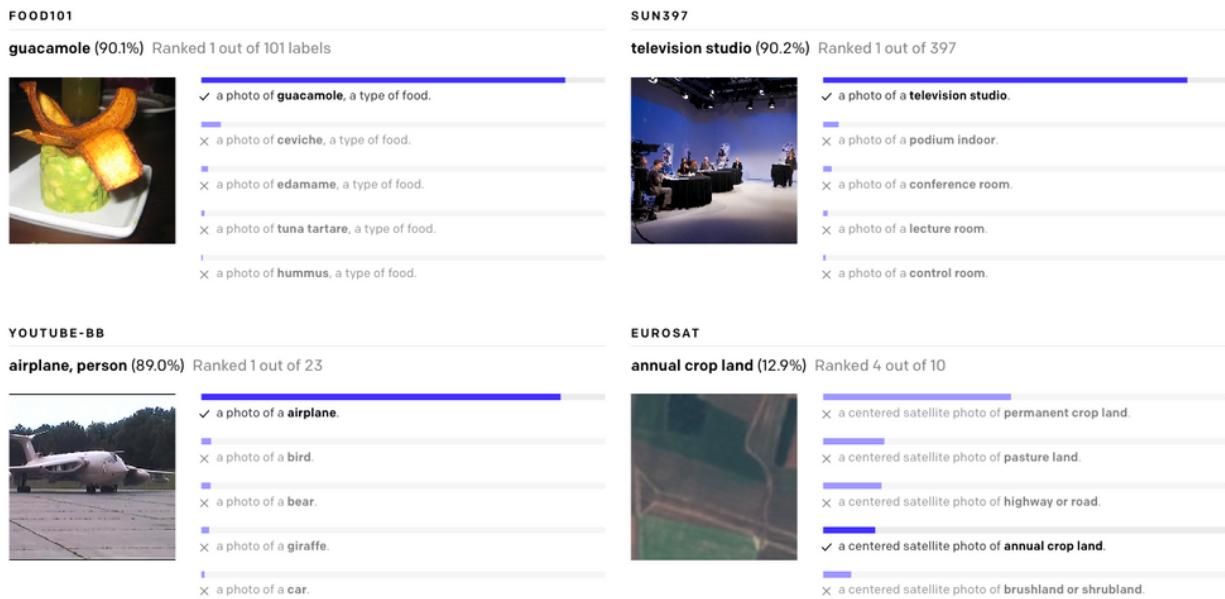


Рис 2. Использование CLIP для задач zero-shot классификации

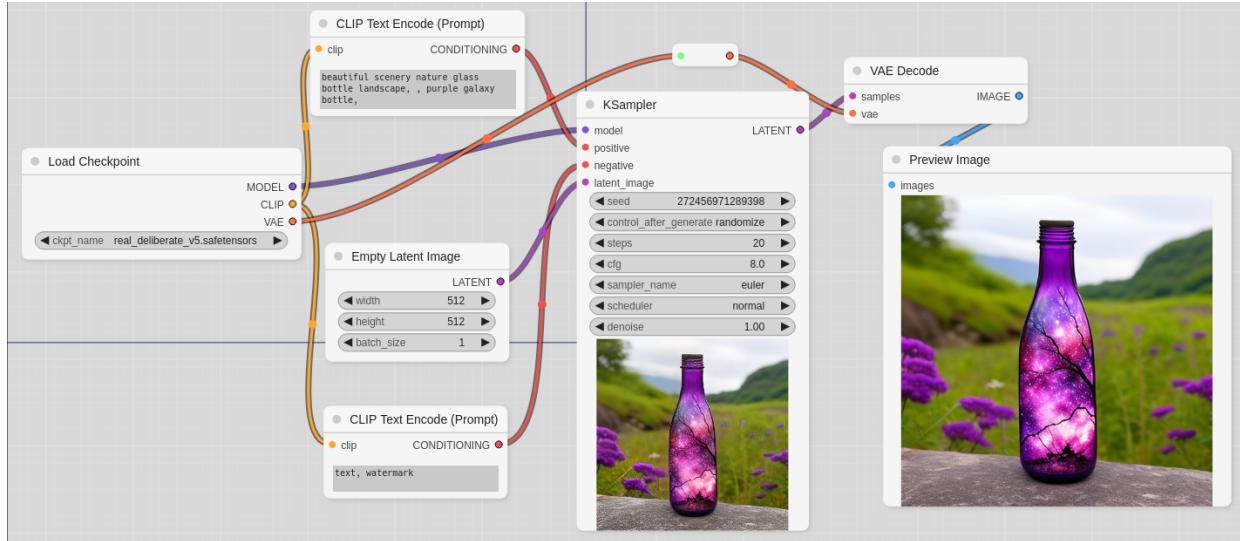


Рис 3. Использование только энкодера текста CLIP с другими моделями для генерации изображений (Stable Diffusion)

3. CLIPSeg

CLIPSeg[2] объединяет энкодеры из CLIP и свой декодер для решения задачи сегментации. На вход модели подается изначальное изображение, а запросом служит либо другое изображение, либо текстовое описание. Изображения и текст кодируются оригинальными энкодерами из CLIP и передаются в декодер.

Решаемые задачи: zero-shot сегментация.

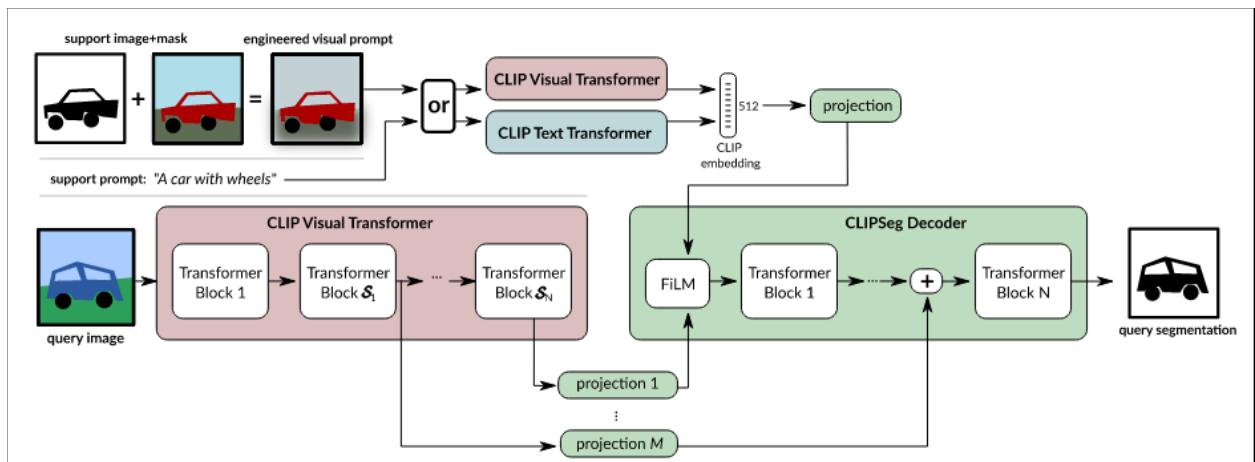


Рис 4. Архитектура CLIPSeg. Красные и синие блоки — предобученные части CLIP.

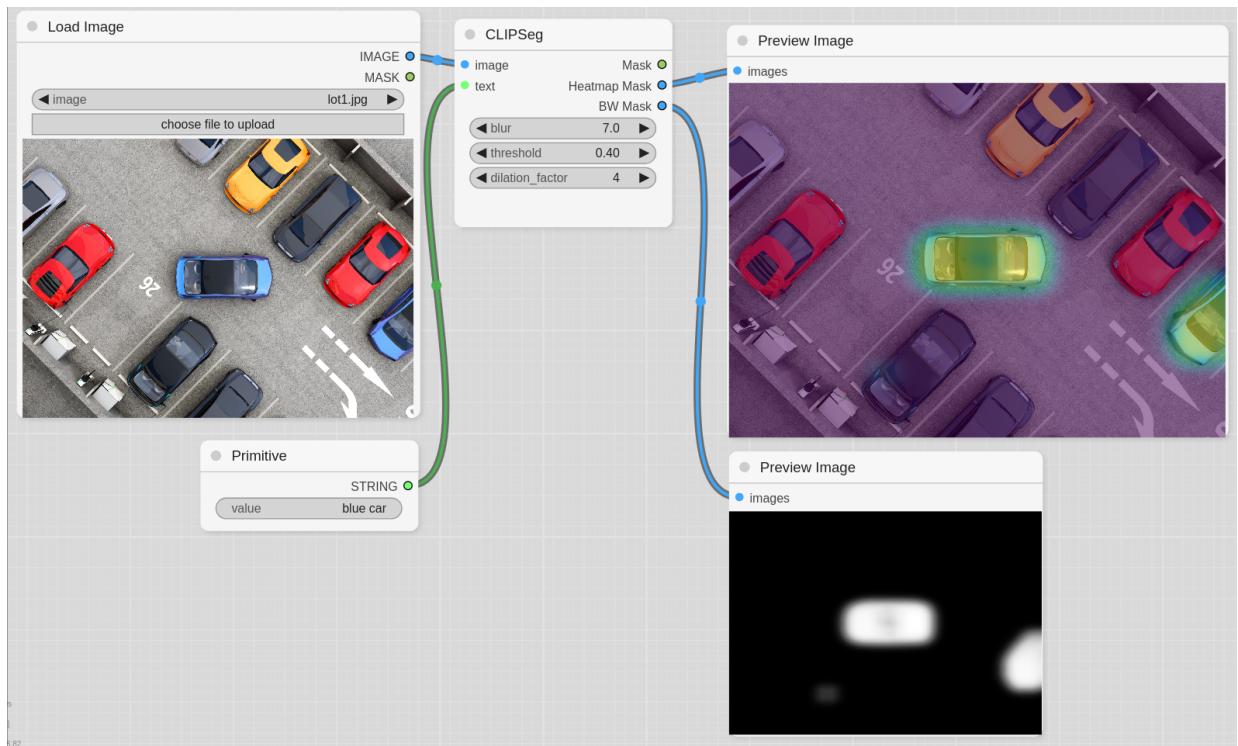


Рис 5. Zero-shot сегментация с использованием CLIPSeg, пример №1

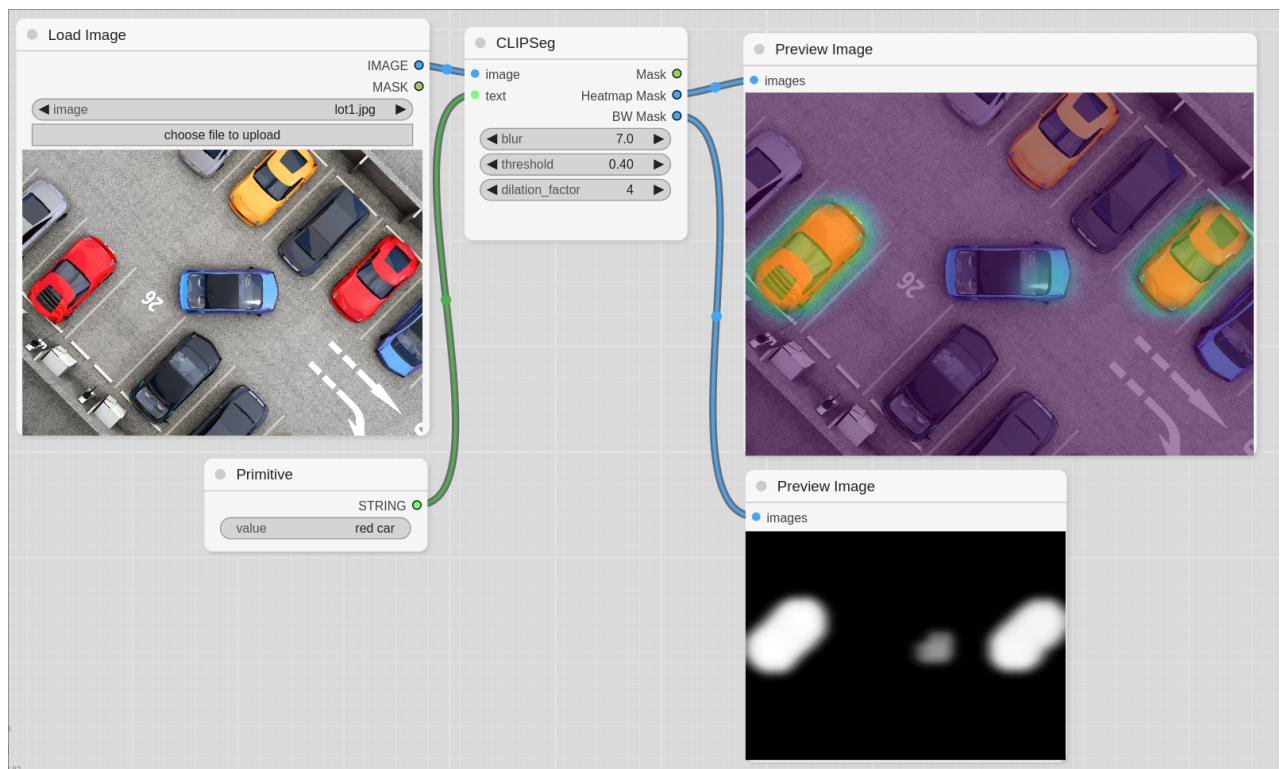


Рис 6. Zero-shot сегментация с использованием CLIPSeg, пример №2

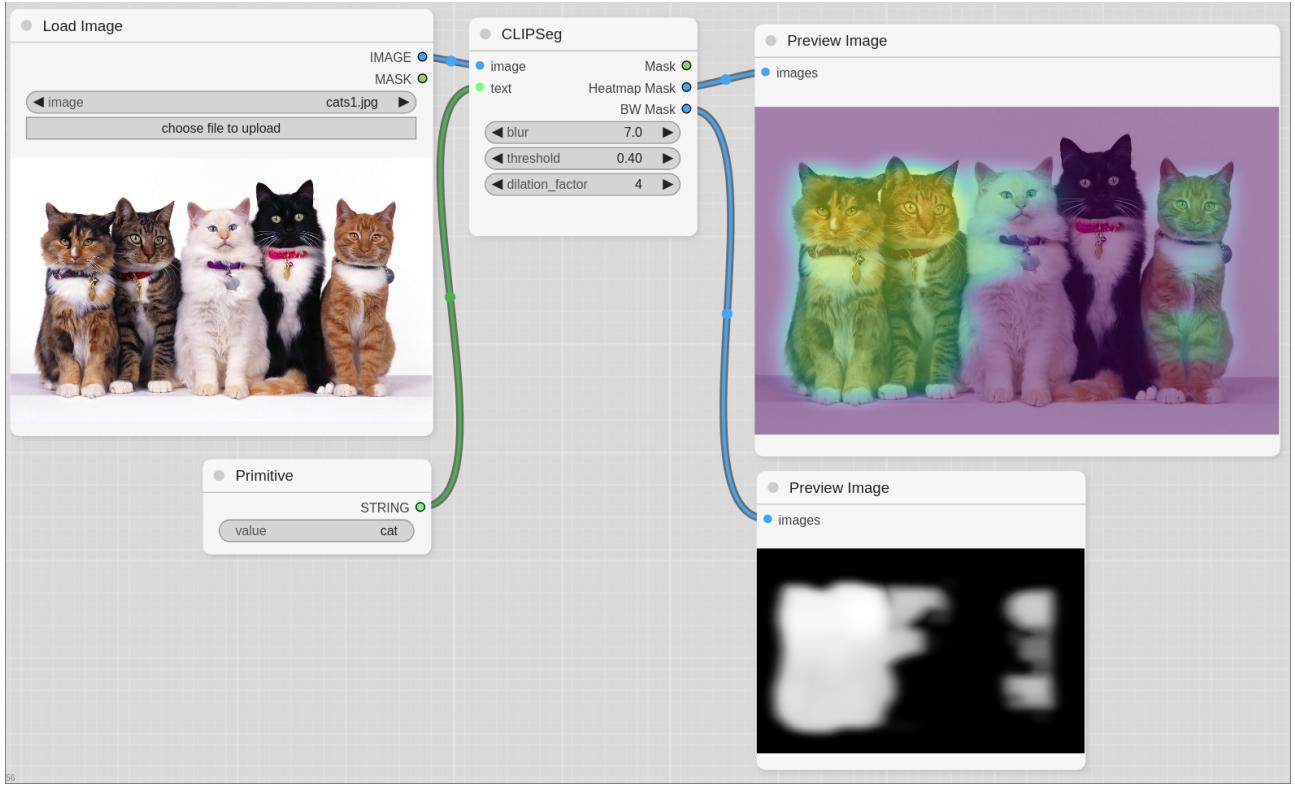


Рис 7. Zero-shot сегментация с использованием CLIPSeg, пример №3

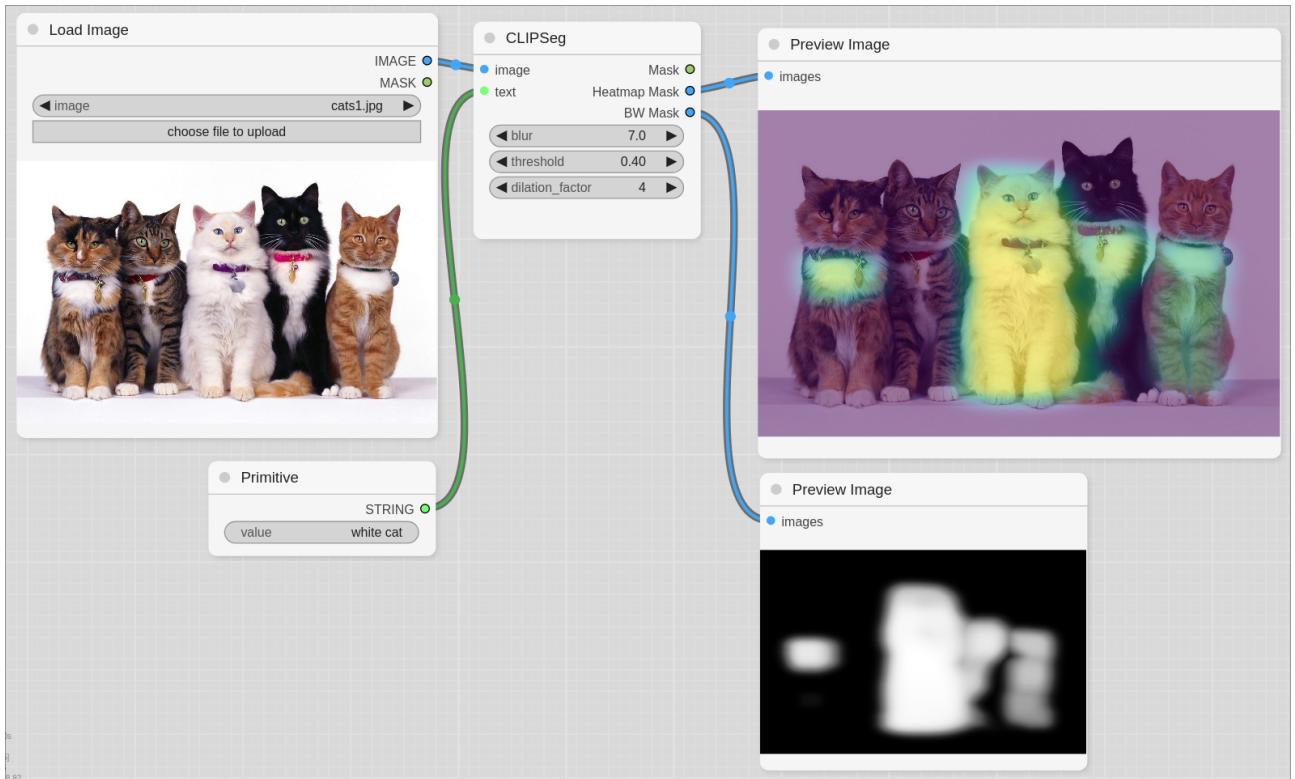


Рис 8. Zero-shot сегментация с использованием CLIPSeg, пример №4

Из примеров видно, что на запросе «white cat» были выделены все белые участки, а не вся белая кошка целиком. Также, белая кошка не попала в запрос «cat».

4. SAM

SAM[3] (Segment Anything Model) — это модель для сегментации, которая была выпущена Meta весной 2023 года. SAM сегментирует объекты на картинке в соответствии с запросом: им может быть точка на изображении или приблизительный прямоугольник. Модель достаточно быстрая, чтобы работать в реальном времени.

Решаемые задачи: сегментация по точке или прямоугольнику, (в теории) по текстовому запросу.

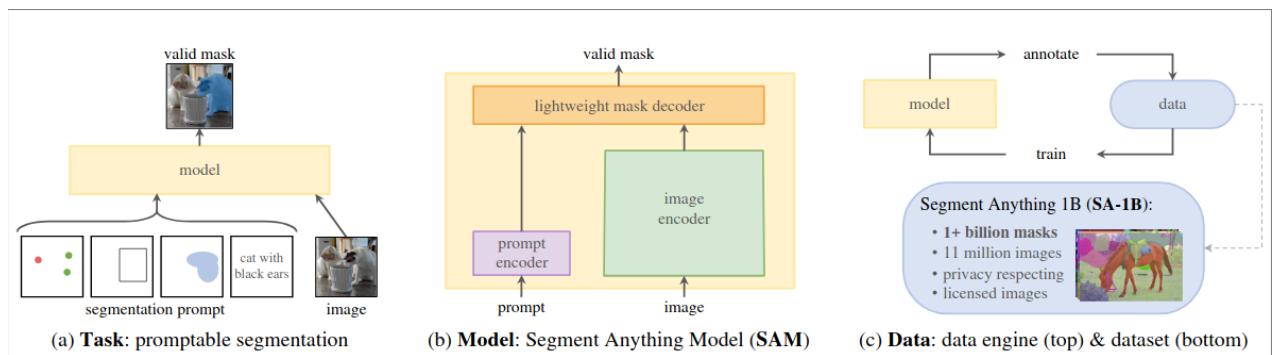


Рис 9. Архитектура SAM



Рис 10. Пример работы SAM

5. DINO

Модель DINO[5] (DETR with Improved deNoising anchOr boxes) — улучшенный вариант модели DETR[4]. Обе модели решают задачу обнаружения объектов на изображении.

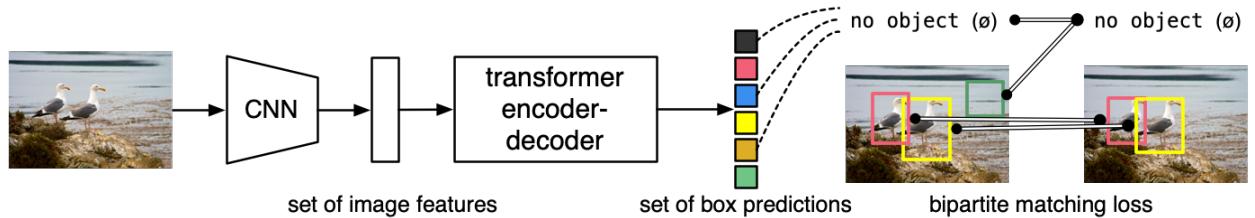


Рис 11. Архитектура DETR

6. Grounding DINO

Grounding DINO[6] — модифицированная DINO с применением наработок CLIP.

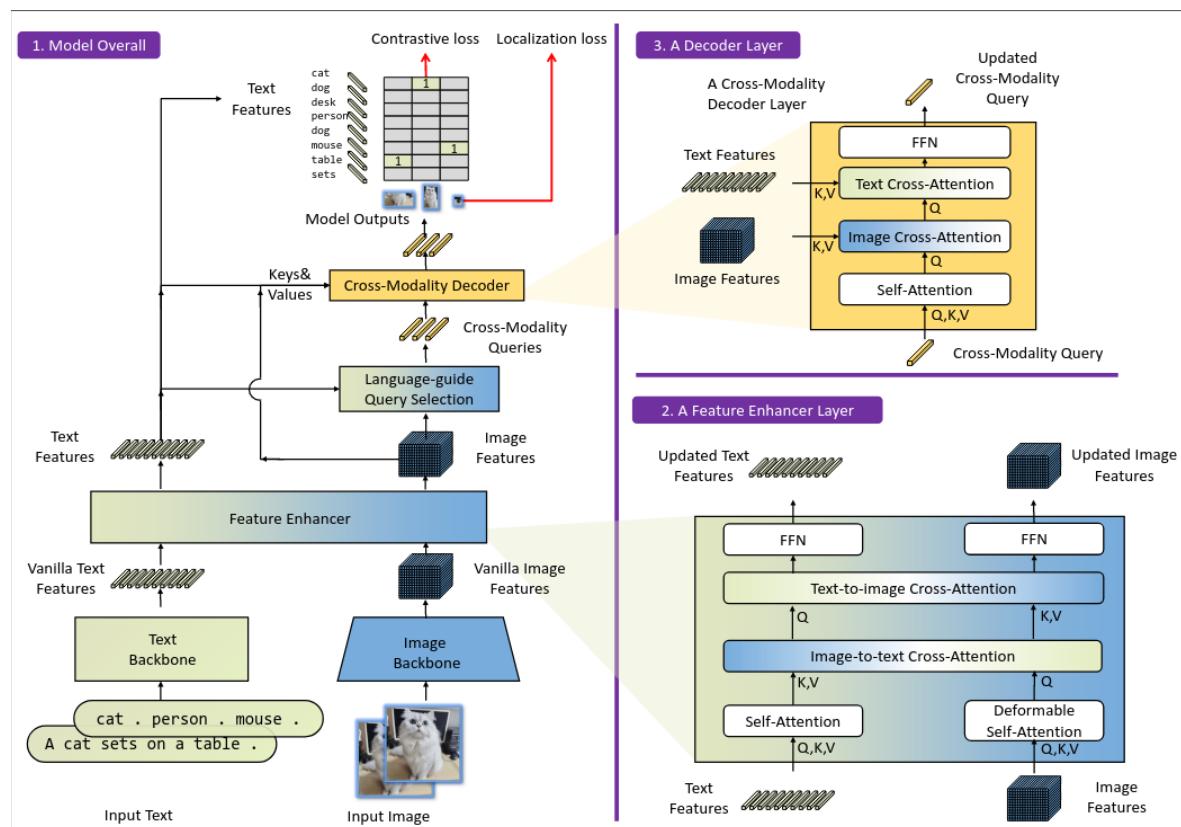


Рис 12. Архитектура Grounding DINO

Модель принимает на вход изображение и текстовые описания объектов, которые необходимо найти, на естественном языке, т.е. выполняет задачу zero-shot обнаружения объектов. На данный момент (конец 2023 года) модель Grounding-DINO-T показывает лучший результат среди прочих для задачи zero-shot обнаружения.

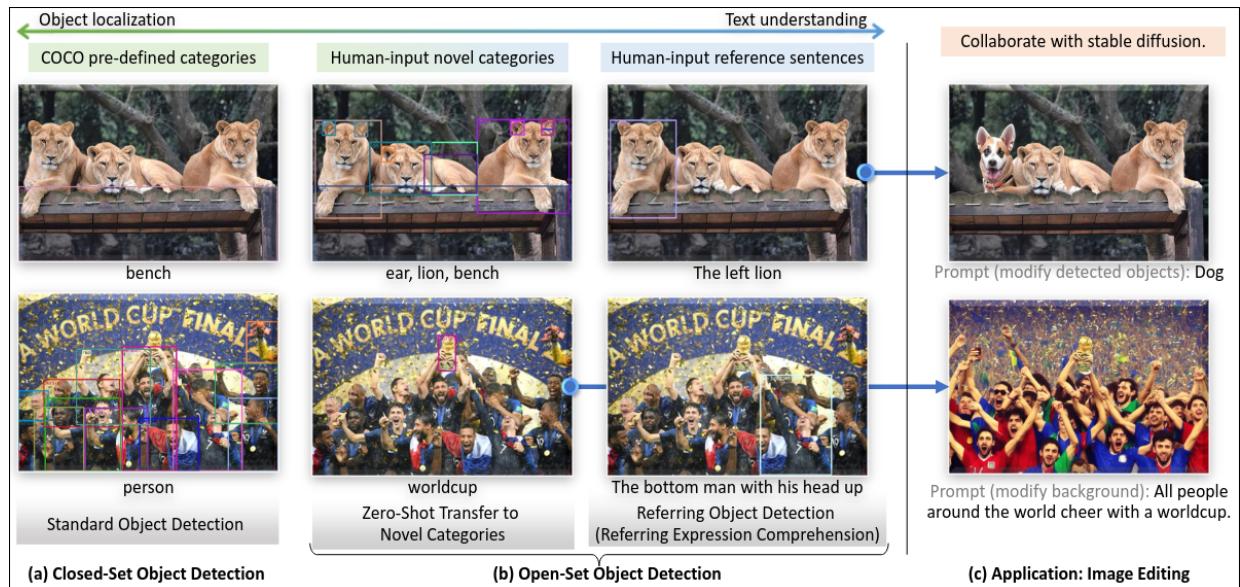


Рис 13. Применения Grounding DINO

Из рис. 13 видно, что модель способна понимать цвета, расположения и прочие атрибуты в текстовом описании.

7. Ансамбль моделей Grounding DINO и SAM

Для решения задачи zero-shot сегментации можно объединить две сети: сеть для zero-shot обнаружения объектов и сеть для сегментации, например Grounding Dino и SAM. Для этого напишем простое приложение на Python с использованием фреймворка Django.

Создадим проект:

```
$ django-admin startproject seg
```

Файл seg/seg/templates/index.html:

```
<!DOCTYPE html>
{% load static %}
<html>
  <head>
    <title>Segmentation</title>
    <style>
      .limit {
        max-width: 1024px;
        max-height: 768px;
      }
    </style>
  </head>
  <body>
    {% csrf_token %}
    <script type="module" src="{% static "main.js" %}"></script>
    <input type="file" id="upload" accept="image/*"></input>
    <br/>
    <input type="text" id="prompt" value="">
    <input type="button" value="Segment" id="segment">
    <br/>
    <img id="inpimage" class="limit">
    <img id="outimage" class="limit">
  </body>
</html>
```

Файл seg/seg/urls.py:

```
from django.urls import path
from . import views

urlpatterns = [
    path("", views.index, name="index"),
    path("segment", views.segment, name="segment"),
    path("result/", views.result, name="result"),
]
```

Изменения в seg/seg/settings.py:

```
import os
TEMPLATES = [
{
    "BACKEND": "django.template.backends.django.DjangoTemplates",
    "DIRS": [
        os.path.join(BASE_DIR / "seg/templates"),
    ],
    "APP_DIRS": True,
    "OPTIONS": {
        "context_processors": [
            "django.template.context_processors.debug",
            "django.template.context_processors.request",
            "django.contrib.auth.context_processors.auth",
            "django.contrib.messages.context_processors.messages",
        ],
    },
},
]
STATICFILES_DIRS = [
    BASE_DIR / "seg/static",
]
```

Файл seg/seg/views.py:

```
import cv2
import numpy as np

from django.shortcuts import render
from django.core.exceptions import BadRequest
from django.http import HttpResponse
from django.http import HttpResponseBadRequest

from .segment import Segment

seg = Segment()
img = None

def index(request):
    return render(request, "index.html")

def segment(request):
    global img
    global seg

    if request.method == "POST":
        f = request.FILES["file"]
        p = request.POST["prompt"]
        img = cv2.imencode(
            ".jpg",
            seg.annotate(
                cv2.imdecode(
                    np.frombuffer(f.read(), dtype = np.uint8),
                    cv2.IMREAD_COLOR,
                ),
                p,
            ),
            [ int(cv2.IMWRITE_JPEG_QUALITY), 100 ],
        )[1].tostring()
        return HttpResponse()

    return HttpResponseBadRequest()
```

```

def result(request):
    global img

    if request.method == "GET" and img is not None:
        return HttpResponse(
            img,
            content_type = "image/jpeg",
        )
    return HttpResponseBadRequest()

```

Файл seg/seg/static/main.js:

```

const upload = document.getElementById("upload");
const promptel = document.getElementById("prompt");
const segment = document.getElementById("segment");
const inpimage = document.getElementById("inpimage");
const outimage = document.getElementById("outimage");
const token = document.getElementsByName("csrfmiddlewaretoken")[0];

let f = null;
let n = 0;

segment.onclick = async () => {
    if (f !== null) {
        const form = new FormData();
        form.append("file", f);
        form.append("prompt", promptel.value);
        form.append("csrfmiddlewaretoken", token.value);
        await fetch("/segment", {
            method: "POST",
            body: form,
        });
        outimage.src = `/result/?t=${new Date().getTime()}`;
        n++;
    }
};

upload.onchange = async (_) => {
    f = upload.files[0];
    inpimage.src = URL.createObjectURL(f);
};

```

Файл seg/seg/segment.py:

```
import os
import sys
from .settings import BASE_DIR

os.environ["HF_HOME"] = str(BASE_DIR / "models" / "hf")

import cv2
import torch
import numpy as np
import torchvision
import supervision as sv

from groundingdino.util.inference import Model
from segment_anything import SamPredictor
from MobileSAM.setup_mobile_sam import setup_model

DEVICE = os.getenv("DEVICE", "cpu")

class Segment:
    def __init__(self):
        self.gdino = Model(
            model_config_path = str(BASE_DIR / "models" /
"GroundingDINO_SwinB_cfg.py"),
            model_checkpoint_path = str(BASE_DIR / "models" /
"groundingdino_swinb_cogcoor.pth"),
            device = DEVICE,
        )

        self.sam = setup_model()
        self.sam.load_state_dict(
            torch.load(str(BASE_DIR / "models" / "mobile_sam.pt")),
            strict = True
        )
        self.sam.to(device = DEVICE)
        self.predictor = SamPredictor(self.sam)

        self.box_annotator = sv.BoxAnnotator()
        self.mask_annotator = sv.MaskAnnotator()
```

```

@staticmethod
def segment(sam_predictor: SamPredictor, image: np.ndarray, xyxy: np.ndarray)
-> np.ndarray:
    sam_predictor.set_image(image)
    result_masks = []
    for box in xyxy:
        masks, scores, logits = sam_predictor.predict(box = box, multimask_output
= True)
        index = np.argmax(scores)
        result_masks.append(masks[index])
    return np.array(result_masks)

def annotate(self, image, caption):
    detections, labels = self.gdino.predict_with_caption(
        image = image,
        caption = caption,
        box_threshold = 0.3,
        text_threshold = 0.3,
    )

    l2id = {
        x: i
        for i, x in enumerate(set(labels))
    }

    id2l = {
        x: i
        for i, x in l2id.items()
    }

    print(l2id)

    detections.class_id = np.array(list(map(lambda x: l2id[x], labels)))

    nms_idx = torchvision.ops.nms(
        torch.from_numpy(detections.xyxy),
        torch.from_numpy(detections.confidence),
        0.8,
    ).numpy().tolist()

```

```

detections.xyxy = detections.xyxy[nms_idx]
detections.confidence = detections.confidence[nms_idx]
detections.class_id = detections.class_id[nms_idx]

detections.mask = self.segment(
    sam_predictor = self.predictor,
    image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB),
    xyxy = detections.xyxy,
)

labels = [
    f"{{id2l[class_id]}} {confidence:0.2f}"
    for i, (_, _, confidence, class_id, _) in enumerate(detections)
]

```

annotated_image = image.copy()

```

annotated_image = self.mask_annotator.annotate(
    scene = annotated_image,
    detections = detections,
)
annotated_image = self.box_annotator.annotate(
    scene = annotated_image,
    detections = detections,
    labels = labels,
)
return annotated_image

```

Из seg:

```

$ virtualenv --system-site-packages venv
$ . ./venv/bin/activate
$ git clone https://github.com/IDEA-Research/GroundingDINO --depth=1 && cd GroundingDINO && pip install -e .
$ git clone https://github.com/IDEA-Research/Grounded-Segment-Anything/ --depth=1 && mv Grounded-Segment-Anything/EfficientSAM/MobileSAM . && mv Grounded-Segment-Anything/segment_anything . && rm -rf Grounded-Segment-Anything/
$ cd segment Anything && pip install -e .

```

Файлы в seg/models:

GroundingDINO_SwinB_cfg.py:

https://raw.githubusercontent.com/IDEA-Research/GroundingDINO/main/groundingdino/config/GroundingDINO_SwinB_cfg.py

groundingdino_swinb_cogcoor.pth:

https://github.com/IDEA-Research/GroundingDINO/releases/download/v0.1.0-alpha2/groundingdino_swinb_cogcoor.pth

mobile_sam.pt:

https://github.com/ChaoningZhang/MobileSAM/raw/master/weights/mobile_sam.pt



Рис 14. «blue car» по мнению ансамбля Grounding DINO и SAM

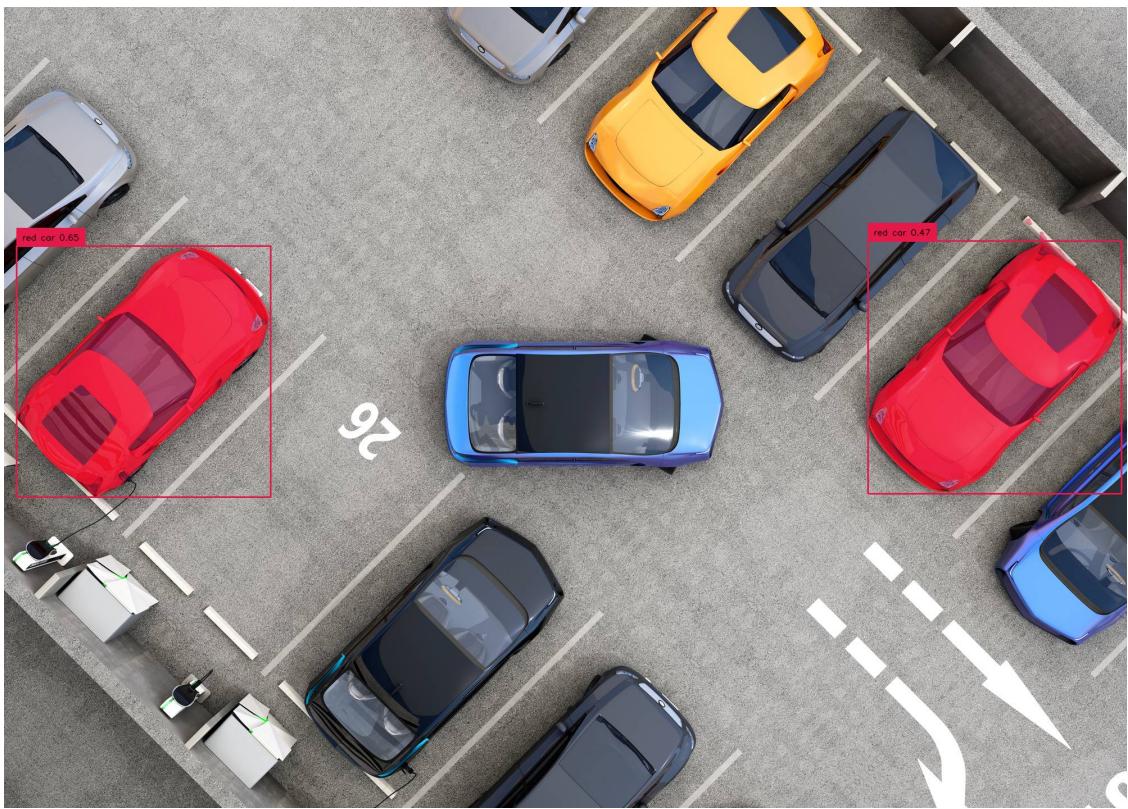


Рис 15. «red car» по мнению ансамбля Grounding DINO и SAM

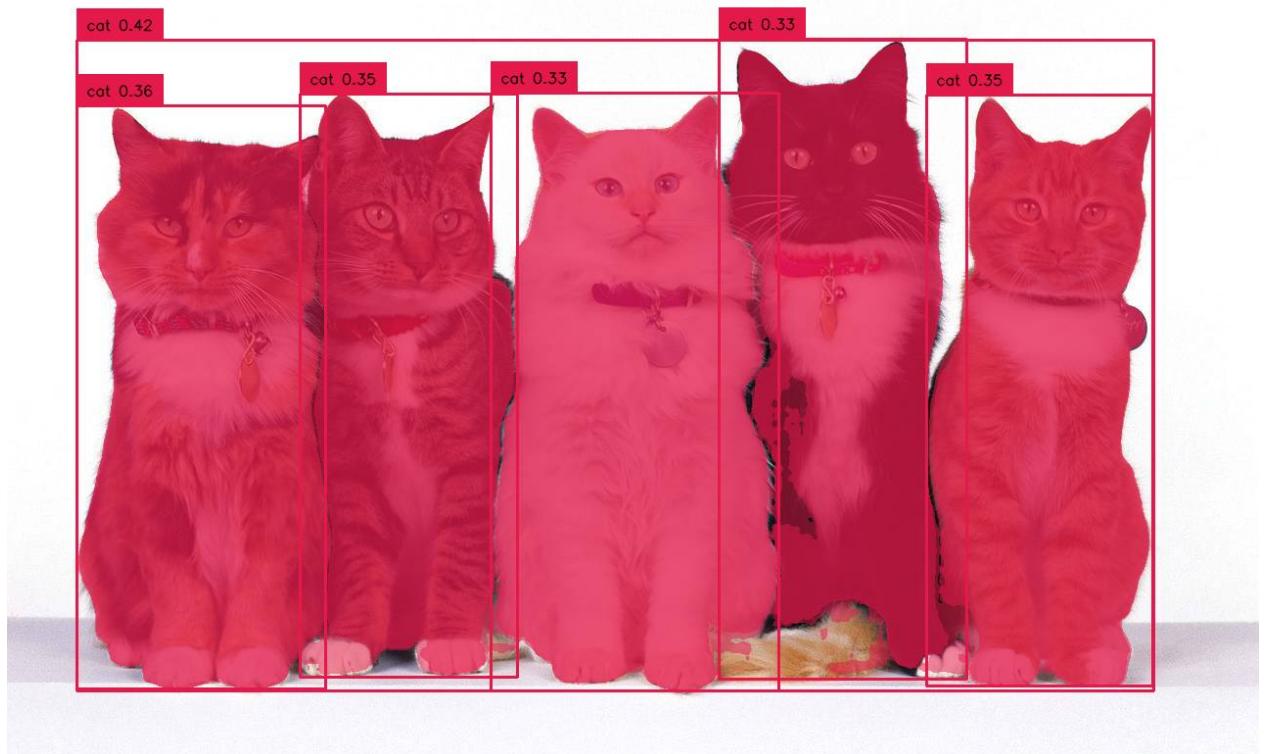


Рис 16. «cat» по мнению ансамбля Grounding DINO и SAM

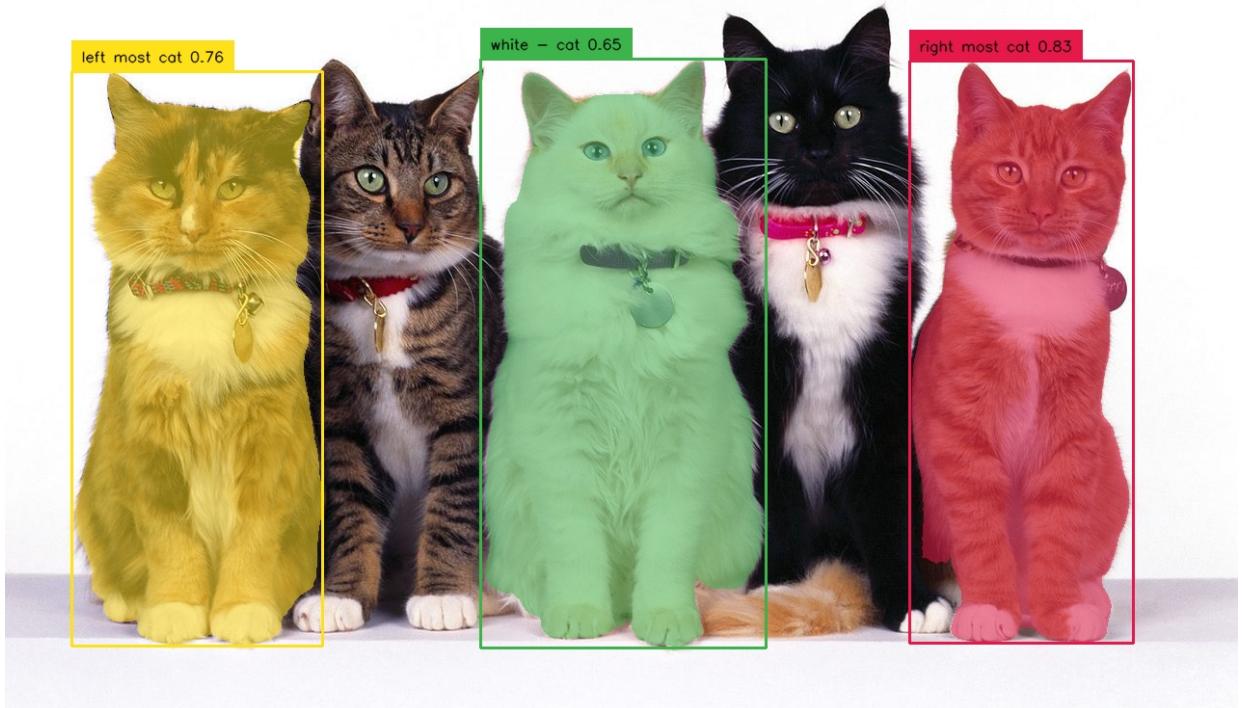


Рис 17. «left most cat, white cat, right most cat» по мнению ансамбля Grounding DINO и SAM

8. Заключение

Текущие решения zero-shot задач очень далеки от идеала. Как и со Stable Diffusion[7] и LLaMA[8], зачастую необходимо редактировать запрос, прибегая к типичным приемам «prompt engineering», чтобы модель выдавала адекватный результат, а некоторых вещей модель вовсе не знает. Grounding DINO, например, не воспринимает слово «collar», и вместо ошейника обводит всю кошку целиком.

Если необходима скорость, то стоит рассматривать модели на основе архитектуры CLIP: CLIPSeg, ZegCLIP[9] и т.д. Если важен результат, то имеет смысл рассматривать ансамбль из zero-shot детектора и отдельной модели для сегментации. Ансамбль Grounding DINO и SAM показали себя лучше, чем CLIPSeg, но скорость работы Grounding DINO очень низка, что делает ансамбль непригодным для realtime приложений, таких как приложения дополненной реальности.

Также стоит учитывать, что zero-shot модели не обладают такой же надежностью, которой обладают модели, специально обученные для решения задачи.

9. Список литературы

1. Radford A. et al. Learning transferable visual models from natural language supervision //International conference on machine learning. – PMLR, 2021. – C. 8748-8763.
2. Lüddecke T., Ecker A. Image segmentation using text and image prompts. In 2022 IEEE //CVF Conference on Computer Vision and Pattern Recognition (CVPR). – 2022. – C. 7076-7086.
3. Kirillov A. et al. Segment anything //arXiv preprint arXiv:2304.02643. – 2023.
4. Carion N. et al. End-to-end object detection with transformers //European conference on computer vision. – Cham : Springer International Publishing, 2020. – C. 213-229.
5. Zhang H. et al. Dino: Detr with improved denoising anchor boxes for end-to-end object detection //arXiv preprint arXiv:2203.03605. – 2022.
6. Liu S. et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection //arXiv preprint arXiv:2303.05499. – 2023.
7. Rombach R. et al. High-resolution image synthesis with latent diffusion models //Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. – 2022. – C. 10684-10695.
8. Touvron H. et al. Llama: Open and efficient foundation language models //arXiv preprint arXiv:2302.13971. – 2023.
9. Zhou Z. et al. Zegclip: Towards adapting clip for zero-shot semantic segmentation //Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. – 2023. – C. 11175-11185.