



# Расчет БЖУ блюда по фото

Разработка телеграм-бота для анализа изображений блюд и расчета БЖУ.

# Постановка задачи

## Задача:

Создать бота, который анализирует изображения блюд и рассчитывает их БЖУ (белки, жиры, углеводы), а также калорийность.

## Функционал:

Пользователь отправляет боту фотографию блюда, бот анализирует ее и предоставляет информацию о его питательной ценности.

# Поиск датасета

## The Food-101 Data Set



Нашли датасеты для классификации изображений:

- Food-101 – датасет 2014 года, один из самых основных для классификации блюд: [ссылка](#)
- Датасет из статьи 2023 года «Chop & Learn: Recognizing and Generating Object-State Compositions»: [ссылка](#)



# Поиск датасета

## Проблема

В задаче классификации мы сможем относить изображение только к 1 классу!

Наша же задача – правильно выделять все классы еды, которые есть на фото

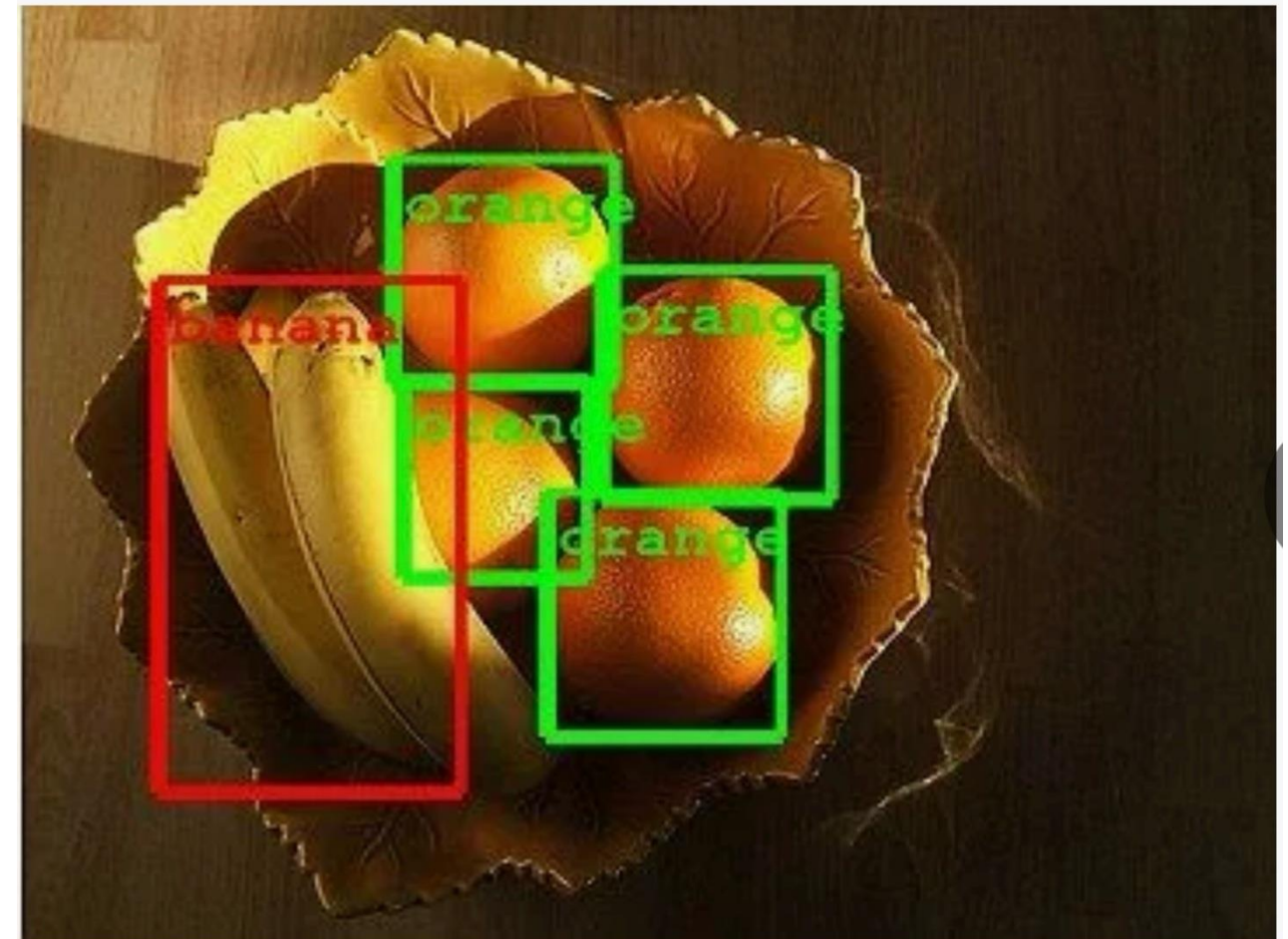
## Решение

Будем обучать и использовать модели для задачи **Детекции** на изображении

## Датасет

Нашел уже существующий датасет на [roboflow](https://roboflow.com/) и немного дополнил его своими изображениями.

Все изображения 640x640



# Подбор моделей

В качестве моделей для дообучения и сравнения выбрал **архитектуру YOLO (You Only Look Once)** – базовая архитектура в настоящее время для детекции на изображениях

Выбранные модели:



## YOLO 11 (2024)

Новейшая модель от ultralytics. YOLO11 вносит значительные улучшения в архитектуру и методы обучения, что делает его универсальным выбором для широкого спектра задач компьютерного зрения.

- + Небольшое количество параметров (хватает 32 гб MPS) при обучении
- + Быстрый инференс
- + Понятная документация
- Долгое обучение на MPS (порядка часа на одну эпоху)

## RT-DETR (Realtime Detection Transformer)

### RT-DETR

Модель обнаружения объектов в реальном времени, разработанная Baidu. Она построена на архитектуре Transformer и обеспечивает высокую точность при сохранении производительности в реальном времени.

- + Использование трансформеров
- + Хорошее качество
- Слишком большая модель для обучения (не влезит на 32 гб MPS)

Итог: не хватило памяти

## YOLOv8

### YOLO 8 (2023)

YOLOv8 был выпущен Ultralytics 10 января 2023 года, предлагая передовую производительность с точки зрения точности и скорости. Опираясь на достижения предыдущих версий YOLO, YOLOv8 представил новые функции и оптимизации, которые делают его идеальным выбором для различных задач обнаружения объектов в широком спектре приложений.

Плюсы и минусы схожи с YOLO 11, но это предыдущая версия

- + Более быстрое обучение

# Обучение моделей

# YOLO 11

Обучение на 8 эпохах заняло порядка 5-7 часов на нашем датасете на MPS

**Dfl\_loss** - метрика, которая показывает распределения смещений границ boxes, а не напрямую их значения.

**Cls\_loss** - метрика, которая показывает количество ошибок в обнаруженных классах объектов

**Box\_loss** - метрика, которая измеряет разницу между предсказанными bbox и фактическими bbox объектов в обучающих данных



# Обучение моделей

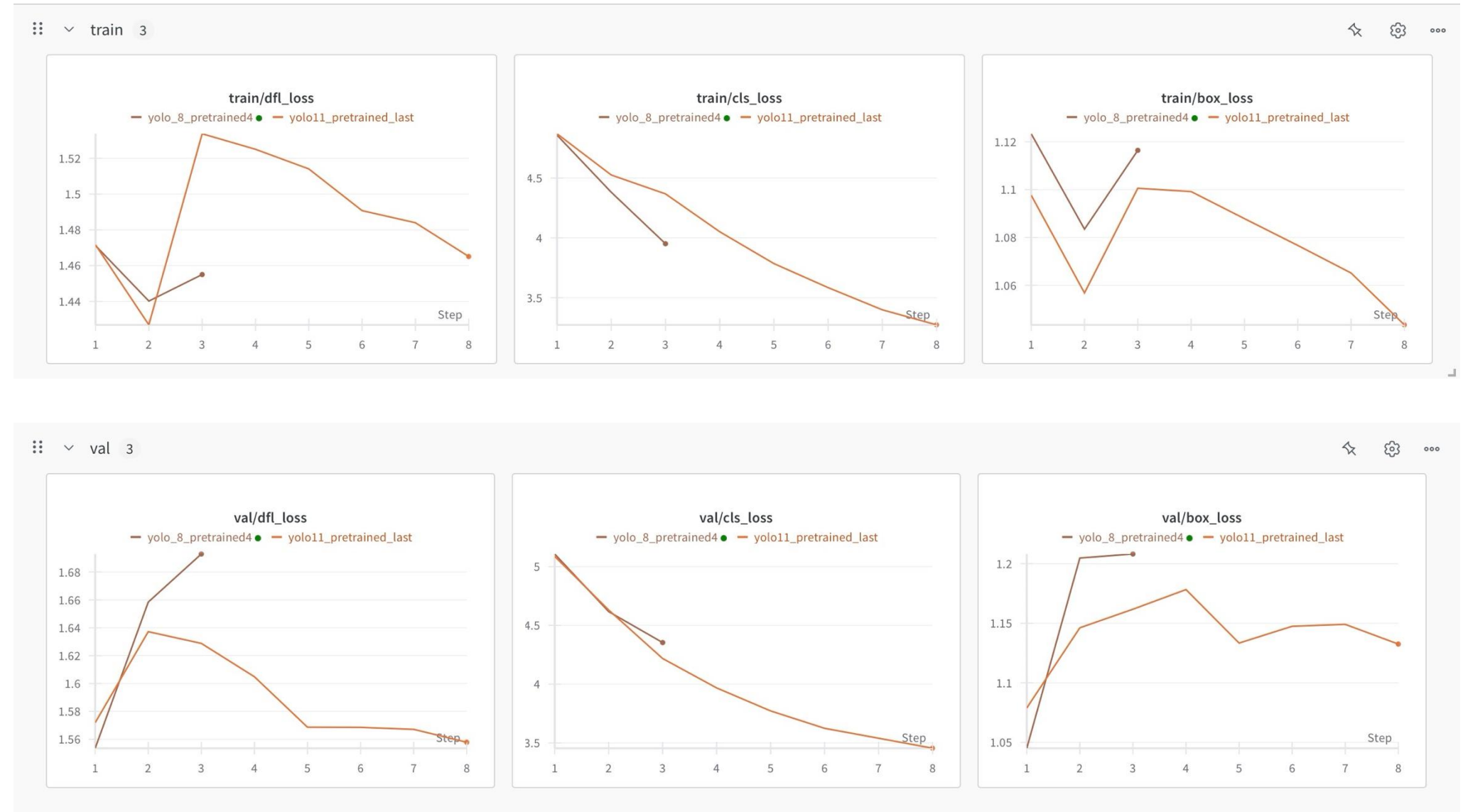
# YOLO 8

Обучение на **4 эпохах** заняло порядка **3 часов** на нашем датасете на **MPS**

**Dfl\_loss** - метрика, которая показывает распределения смещений границ boxes, а не напрямую их значения.

**Cls\_loss** - метрика, которая показывает количество ошибок в обнаруженных классах объектов

**Box\_loss** - метрика, которая измеряет разницу между предсказанными bbox и фактическими bbox объектов в обучающих данных

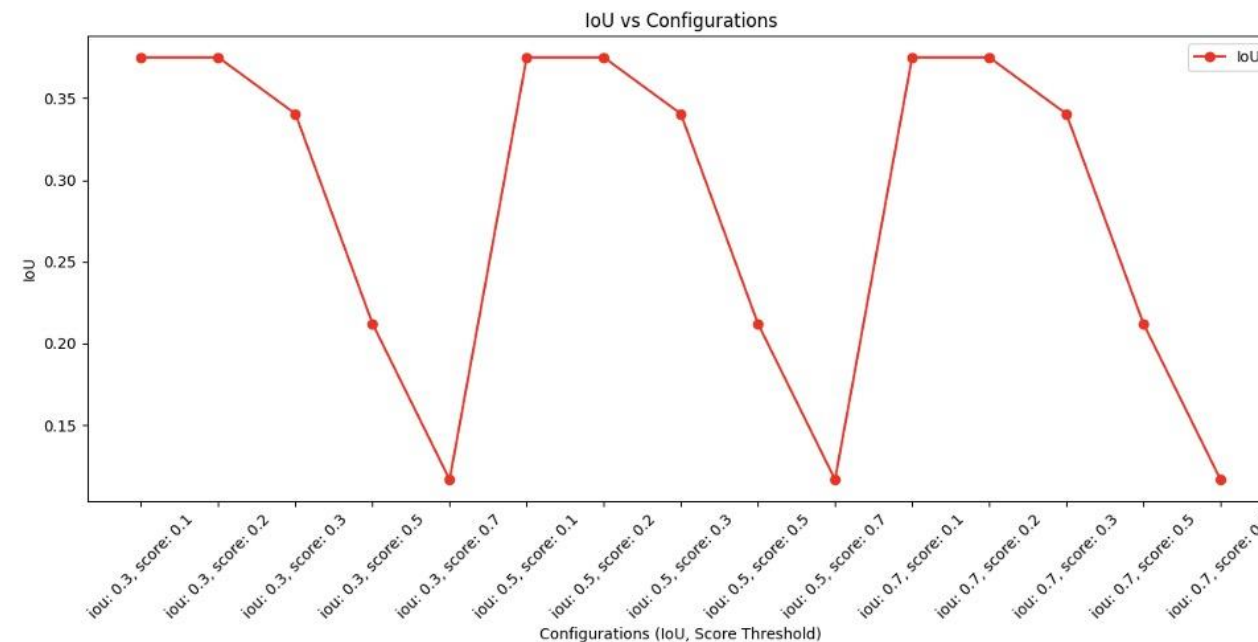
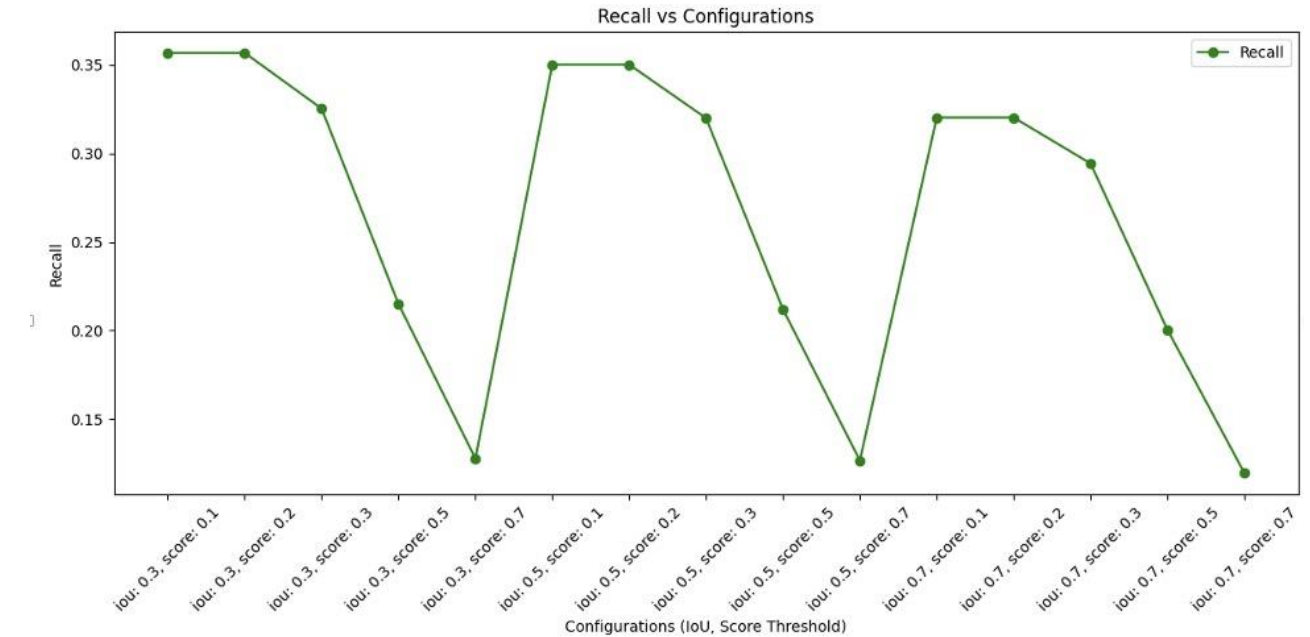
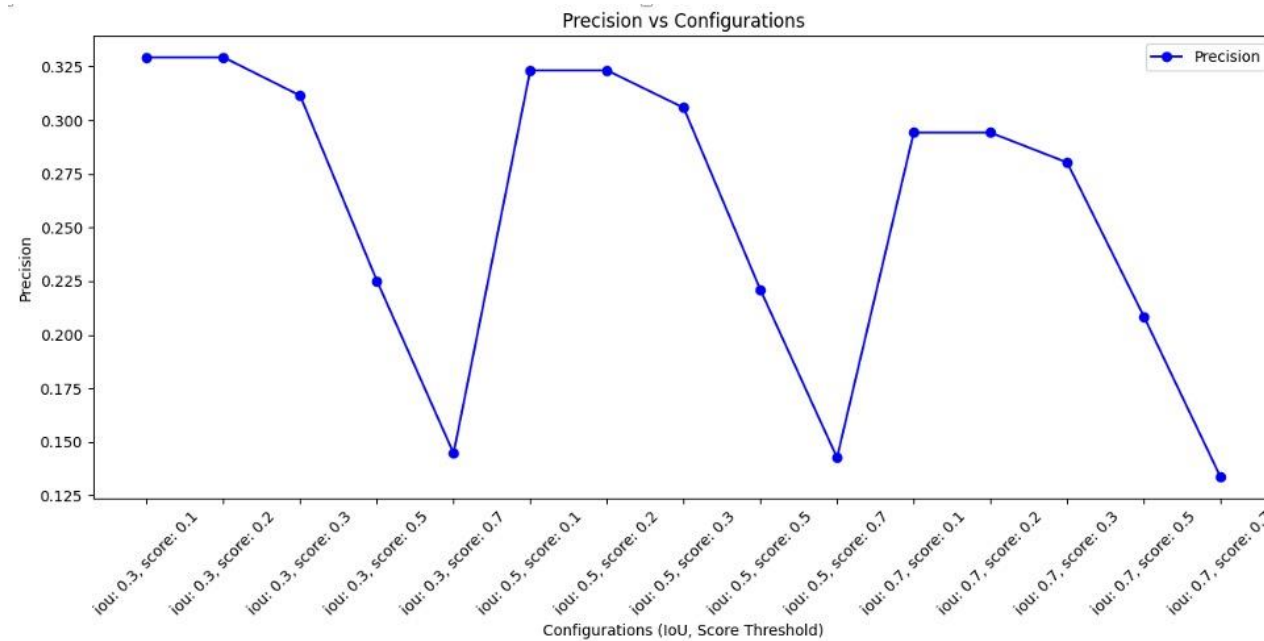




# Обучение моделей

# YOLO 11

Подбор гиперпараметров: **IoU** (intersection over union) threshold and **score** threshold на валидации



Подбирал параметры по метрикам:

precision, recall, IoU

Наилучшие показатели при:

- iou = 0.3, score = 0.1 / 0.3
- iou = 0.5, score = 0.1 / 0.3

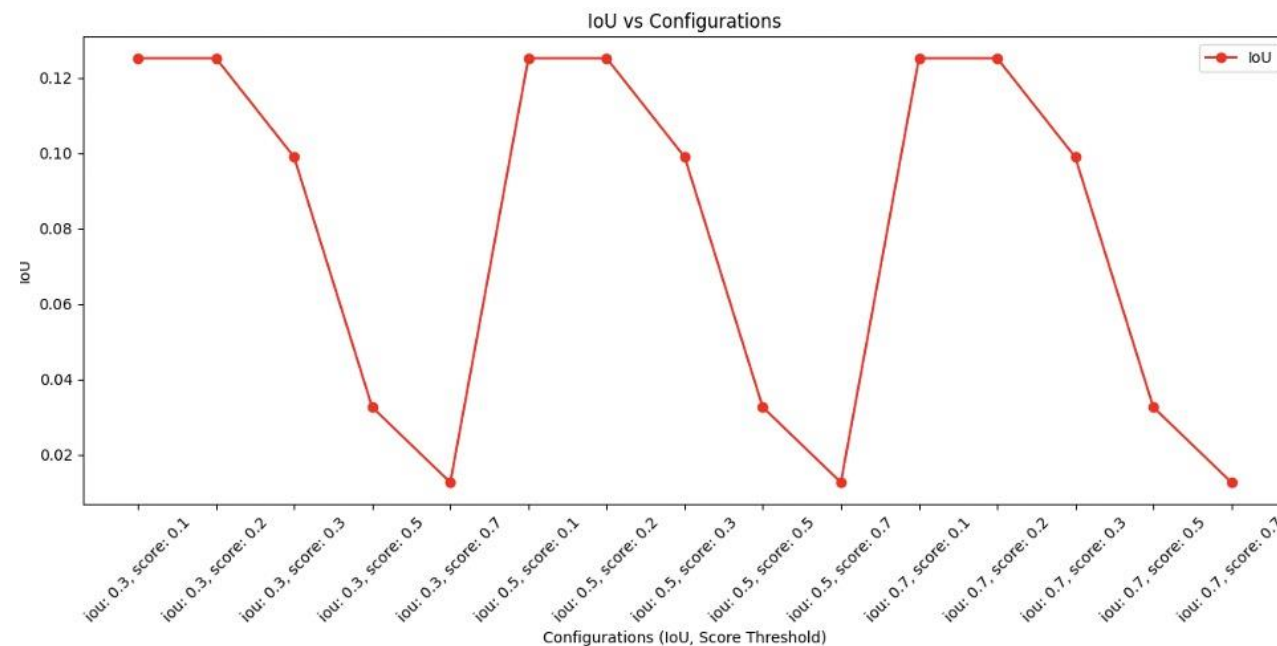
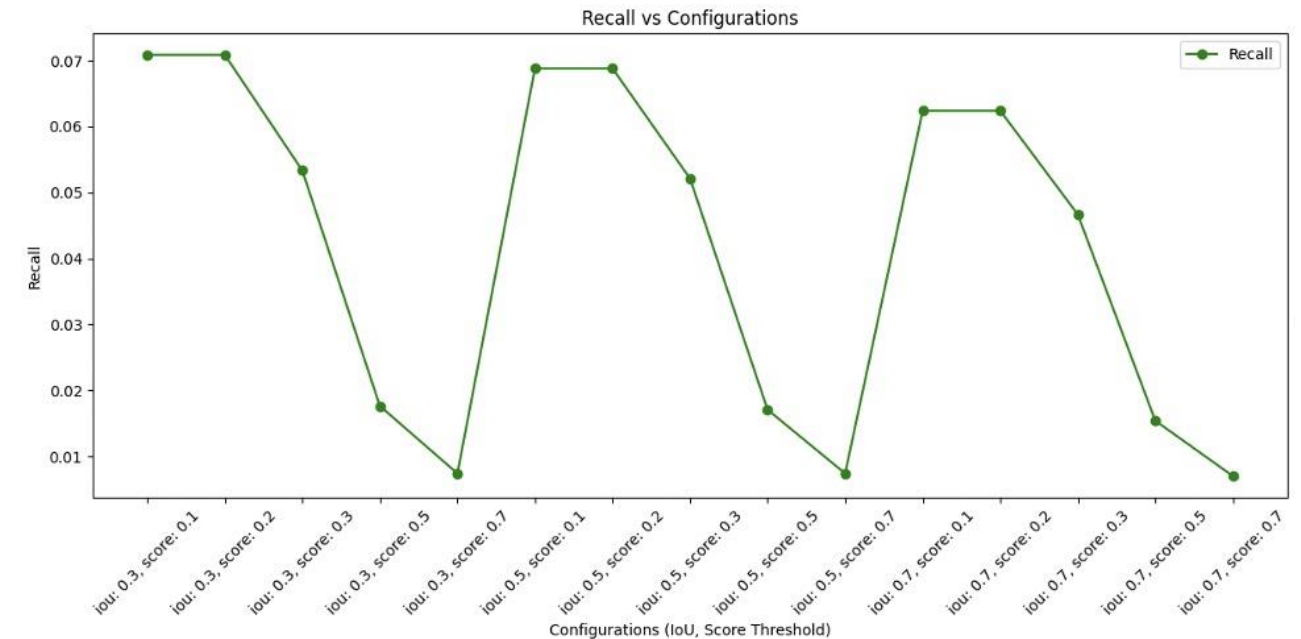
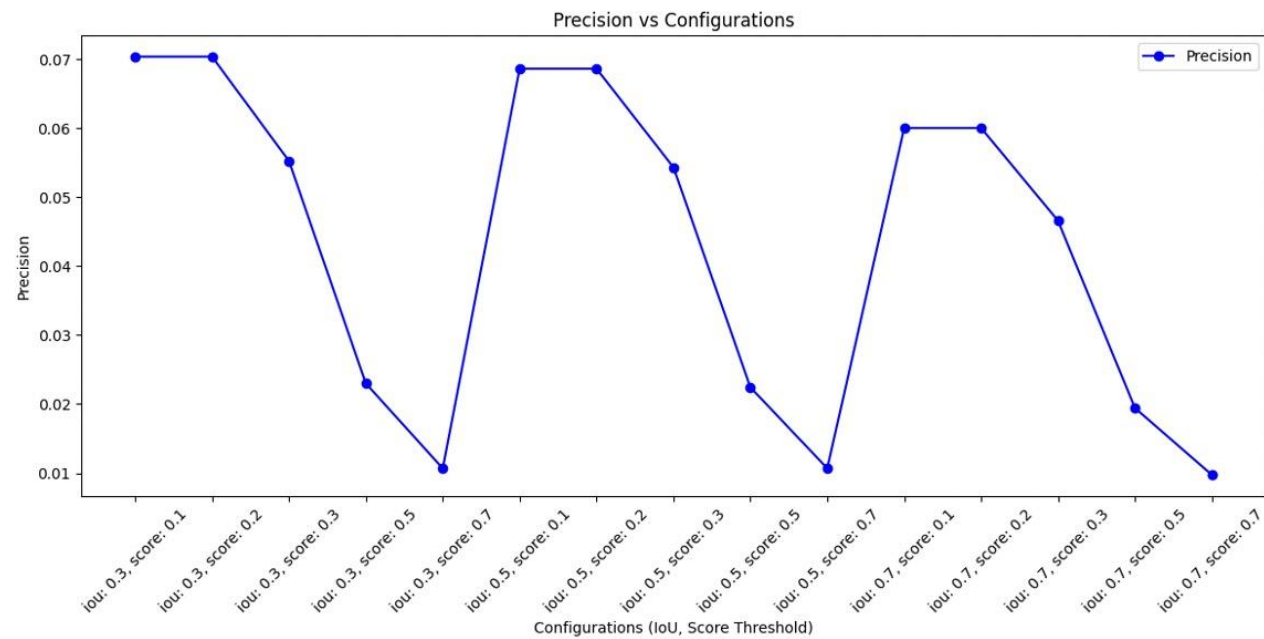
Так как метрики близки, то лучше взять более строгие параметры: iou = 0.5, score = 0.2



# Обучение моделей

# YOLO 8

Подбор гиперпараметров: **IoU** (intersection over union) threshold and **score** threshold на валидации



Подбирал параметры по метрикам:

precision, recall, IoU

Наилучшие показатели при:

- iou = 0.3, score = 0.1 / 0.3
- iou = 0.5, score = 0.1 / 0.3

Так как метрики близки, то лучше взять более строгие параметры: iou = 0.5, score = 0.2

# Итоговые результаты моделей

## YOLO 11

### Train metrics

'precision': 0.40

'recall': 0.42,

'iou': 0.41

### Test metrics

'precision': 0.31

'recall': 0.34

'iou': 0.39



```
iou = compute_iou(boxes, cur_targets[id]["boxes"])
matched_gt = torch.zeros(len(cur_targets[id]['boxes']), dtype=torch.bool)
tp = 0
fp = 0

for j in range(len(boxes)):
    max_iou, max_idx = iou[j].max(dim=0)
    label_now = labels[j].cpu().item()
    if max_iou > iou_threshold and not matched_gt[max_idx] and label_now == cur_targets[id]['labels'][max_idx]:
        tp += 1
        matched_gt[max_idx] = True
    else:
        fp += 1

fn = (matched_gt == 0).sum().item()
precision = tp / (tp + fp) if (tp + fp) > 0 else 0
recall = tp / (tp + fn) if (tp + fn) > 0 else 0
```



### Train metrics

'precision': 0.06,

'recall': 0.05,

'iou': 0.09

### Test metrics

'precision': 0.05,

'recall': 0.05,

'iou': 0.10

## YOLO 8



# Telegram bot

Был написан telegram bot, который использует нашу лучшую модель

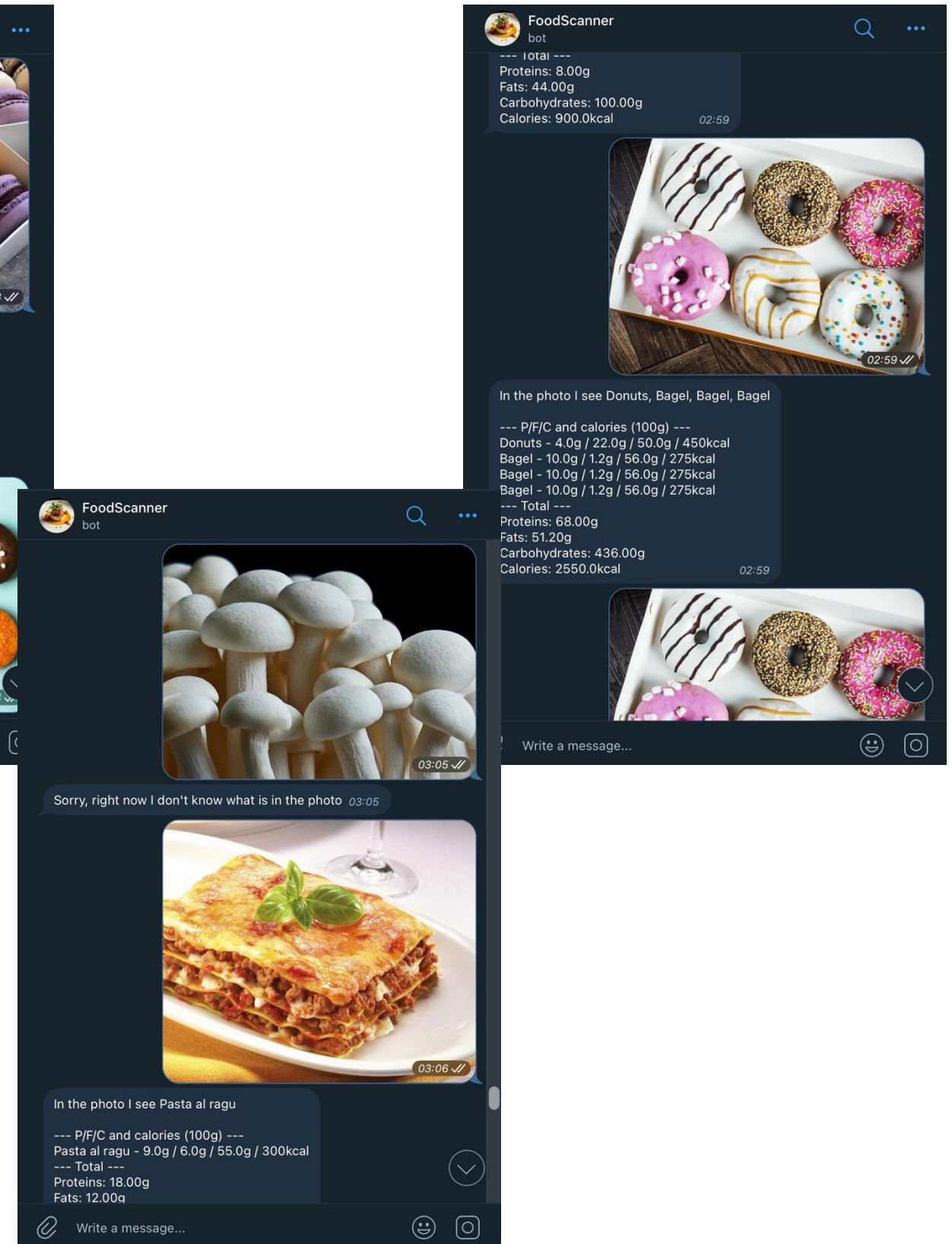
## Ввод:

Пользователь отправляет боту фото своего блюда / еды

## Вывод:

Пользователь получает:

- Название блюда, которое предсказала модель
- Среднее количество БЖУ и калорий на 100г в таком блюде
- Рассчитанное количество БЖУ и калорий в среднестатистической порции блюда



# Ссылки

TG bot: [@foodscanner\\_bot](#)

Github: <https://github.com/Kaparya/FoodScanner>