

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«МОСКОВСКИЙ АВИАЦИОННЫЙ ИНСТИТУТ
(НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ)»

Журнал практики

Институт № 8 «Компьютерные науки и прикладная математика»

Кафедра 806 Учебная группа M8O-212Б-23

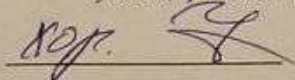
ФИО обучающегося Андрюшин Лев Дмитриевич

Направление подготовки/
специальность 01.03.02 «Прикладная математика и информатика»

шифр, наименование направления подготовки/специальности

Вид практики производственная

учебная, производственная, преддипломная или другой вид практики



Ухов П. А.

ФИО руководителя практики от МАИ

Москва

2024

5. Отзыв руководителя практики от организации/предприятия:

Обучающийся группы М8О-212Б-22 Андриюшин Л.Д. проходил практику на кафедре 806. В ходе работы были выполнены следующие задачи: подготовка датасета для обучения сети YOLOv8 и обучение сети YOLOv8 на детекцию подсолнухов с камеры БПЛА. Подготовка датасета включала сбор и аннотирование изображений подсолнухов, полученных с камеры беспилотного летательного аппарата (БПЛА), обработку и аугментацию данных для улучшения качества и разнообразия обучающей выборки, а также организацию данных в требуемую структуру для эффективного использования в процессе обучения модели. Обучение сети YOLOv8 включало настройку параметров обучения, таких как размер батча, количество эпох, скорость обучения и другие гиперпараметры, обучение модели YOLOv8n на подготовленном датасете с использованием GPU для повышения скорости и эффективности процесса, а также оценку результатов обучения и корректировку параметров для достижения необходимого уровня точности и качества детекции. Рекомендую оценку _____. Материалы, изложенные в отчёте обучающегося, полностью соответствуют индивидуальному заданию. Отчёт подробно описывает все этапы выполненной работы, включая подготовку данных, процесс обучения модели и анализ полученных результатов.

подпись руководителя от
организации/предприятия

/ Ухов П.А. /
расшифровка подписи

11 июля 2024 г.
дата

6. Отчет обучающего по практике:

Я просмотрел множество разных датасетов, пробовал запускать обучение нейросети с некоторыми из них, пробовал объединять их или модернизировать с помощью roboflow, но по итогу решил остановиться на одном конкретном. В нем 779 изображений: 76 test, 157 val и 546 train. Ссылку на этот датасет на QR коде на рисунке 1



Рисунок 1 — Ссылка на датасет (QR-код)

Вот так выглядит мой код на google colab

```
### 1. Mount Google Drive ###
```

```
from google.colab import drive  
drive.mount('/content/gdrive')
```

```
### 2. Define root directory ###
```

```
ROOT_DIR = '/content/gdrive/My Drive/ml'
```

```
### 3. Install Ultralytics ###
```

```
!pip install ultralytics
```

```
### 4. Train model ###
```

```

import os

from ultralytics import YOLO

# Load a model
model = YOLO("yolov8n.pt") # load pre-trained model

training_params = {
    "epochs": 20,
    "batch": 16,
    "imgsz": 640,
    "optimizer": "Adam",
    "lr0": 0.001,
    "momentum": 0.9,
    "weight_decay": 0.0005,
    "warmup_epochs": 2,
    "warmup_momentum": 0.8,
    "warmup_bias_lr": 0.1,
    "cos_lr": True,
    "close_mosaic": 10,
}

# Use the model

results = model.train(data=os.path.join(ROOT_DIR, "google_colab_config.yaml"),
**training_params) # train the model

# Start training
results = model.train(**training_params)

```

```
### 5. Preparation for saving the model ###
```

```
import os
```

```
destination_dir = '/content/gdrive/My Drive/ml'
```

```
if not os.path.exists(destination_dir):
```

```
os.makedirs(destination_dir)
```

```
### 6. Copying training results ###
```

```
import shutil
```

```
shutil.copytree('/content/runs', destination_dir, dirs_exist_ok=True)
```

Что касается изменения различных параметров, то я пробовал немного их менять, но какой-то прибавки в качестве обнаружения подсолнухов я получить не смог, поэтому я решил оставить стандартные параметры. А т.к. время работы с сильными мощностями в Google Colab ограничено, у меня было не так много попыток. По итогу, единственно, что я изменил – это количество эпох. Я сделал это число равным 90. Но я мог сделать и меньше, ибо после определенной эпохи качество обнаружения перестает развиваться, что можно увидеть на рисунке 2.

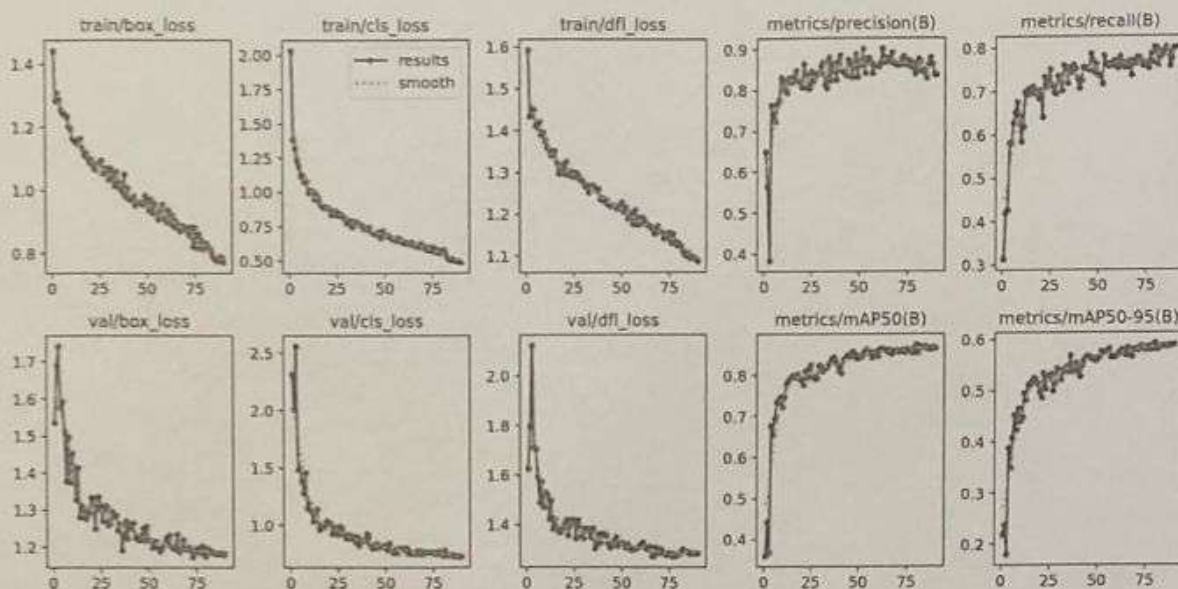


Рисунок 2 — Графики

Как можно заметить параметр, на который в нашем случае нужно ориентироваться при обучении (AP) перестает расти примерно в районе 60 эпохи. В целом мы могли бы остановиться на 60 эпохах. Precision после 60 эпохи вообще начинает расти вниз. Но у меня получалось добиться такого же результата в mAP50 и при 25 эпохах и точно таких же параметрах, что можно увидеть на рисунке 3

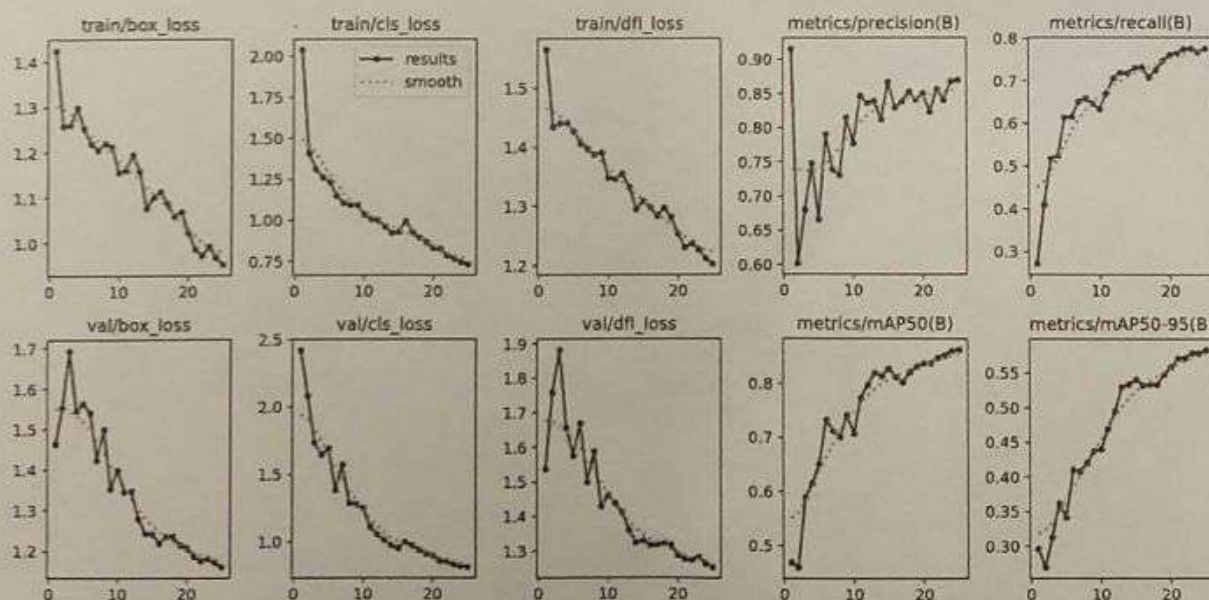


Рисунок 3 — Графики №2

Precision можно интерпретировать как долю объектов, названных классификатором положительными и при этом действительно являющимися положительными, а recall

показывает, какую долю объектов положительного класса из всех объектов положительного класса нашел алгоритм (по этой причине в нашем случае опираться на параметр `recall` не имеет смысла. `Mean Average Precision (mAP)` — это средняя точность предсказания для всех классов при различных порогах уверенности.

Краткое описание всех параметров обучения:

1. `epochs`: количество эпох, в течение которых будет проходить обучение. Определяет, сколько раз модель будет проходить через весь тренировочный набор данных. Большее количество эпох может улучшить обучение, но может также привести к переобучению.
2. `batch`: размер батча, используемый при обучении. Определяет количество примеров, которые будут обрабатываться одновременно. Большой размер батча может ускорить обучение, но требует больше памяти.
3. `imgsz`: размер изображений. Определяет размер входных изображений для модели. Увеличение размера изображений может улучшить точность, но увеличит время обработки и потребление памяти.
4. `patience`: количество эпох без улучшений до остановки обучения. Используется для раннего прекращения обучения, если модель не показывает улучшений. Это помогает избежать лишнего времени на обучение, если модель достигла своей максимальной производительности.
5. `optimizer`: оптимизатор для обучения (например, Adam, SGD). Определяет метод, используемый для обновления весов модели. Разные оптимизаторы могут привести к различной скорости и эффективности обучения.
6. `lr0`: начальная скорость обучения. Определяет размер шага при обновлении весов модели. Слишком высокая скорость обучения может привести к неустойчивому обучению, а слишком низкая — к медленному обучению.
7. `momentum`: момент для оптимизатора. Помогает ускорить обучение и преодолеть локальные минимумы, используя информацию о предыдущих градиентах.
8. `weight_decay`: коэффициент L2-регуляризации. Используется для предотвращения переобучения путем добавления штрафа за большие веса в функции потерь.
9. `warmup_epochs`: количество эпох для разогрева. Медленное увеличение скорости обучения в начальных эпохах, чтобы предотвратить резкие изменения весов в начале обучения.

10. `warmup_momentum`: момент для разогрева. Определяет, насколько сильно будут учитываться предыдущие градиенты в начальный период обучения.
11. `warmup_bias_lr`: скорость обучения для разогрева. Определяет начальную скорость обучения для разогрева.
12. `cos_lr`: использование косинусной функции для уменьшения скорости обучения. Позволяет плавно уменьшать скорость обучения по косинусной функции, что может помочь модели лучше сходиться.
13. `close_mosaic`: закрытие мозаичной аугментации на последних эпохах. Отключает мозаичное аугментирование на последних эпохах обучения, что может улучшить точность модели на валидирующем наборе данных.

Далее на рисунке 4 пример результата обучения нейросети.



Рисунок 4 — Пример результата обучения

Далее представлены ссылка на весы на рисунке 5 и на Google Colab на рисунке 6



Рисунок 5 — Ссылка на весы (QR-код)



Рисунок 6 — Ссылка на Google Colab (QR-код)

A handwritten signature in black ink, consisting of stylized letters.

подпись обучающегося

/ Андрюшин Л.Д. /

расшифровка подписи

11 июля 2024 г.

дата

7. Источники:

1. Датасет [электронный ресурс]: <https://universe.roboflow.com/mugabi-siro-mvlzs/sunflowers-object-detection/browse?queryText=&pageSize=50&startIndex=0&browseQuery=true> (дата обращения: 11.07.2024)
2. Весы [электронный ресурс]: <https://github.com/Vingaell/practice/tree/main/train/weights> (дата обращения: 11.07.2024)
3. Google Colab [электронный ресурс]: https://drive.google.com/file/d/1aTLR_p5Yeu4Amb6j_1XxwP6LbYVhSbKO/view?usp=sharing (дата обращения: 11.07.2024)