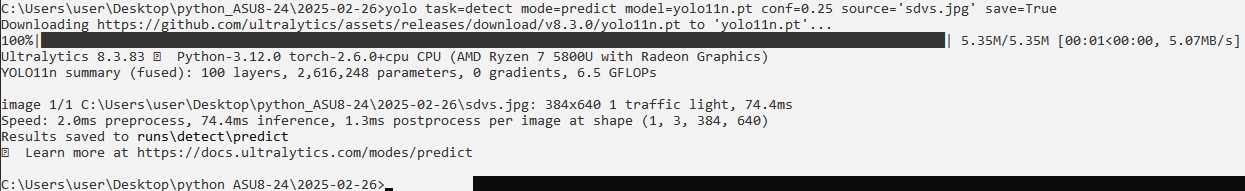
**ОТЧЕТ по проделанной работе (не в официальном виде!)**

Скачаны и проверены работоспособность OCR Tesseract (сырая вещь для таких сложных фото(башмаки), зато хорошо видит обычные цифры на хороших картинках(sdvs.jpg)).

Скачана YoloV11:



Проверил её на различных объектах

|  |  |
| --- | --- |
|  |  |
|  |  |
|  | |

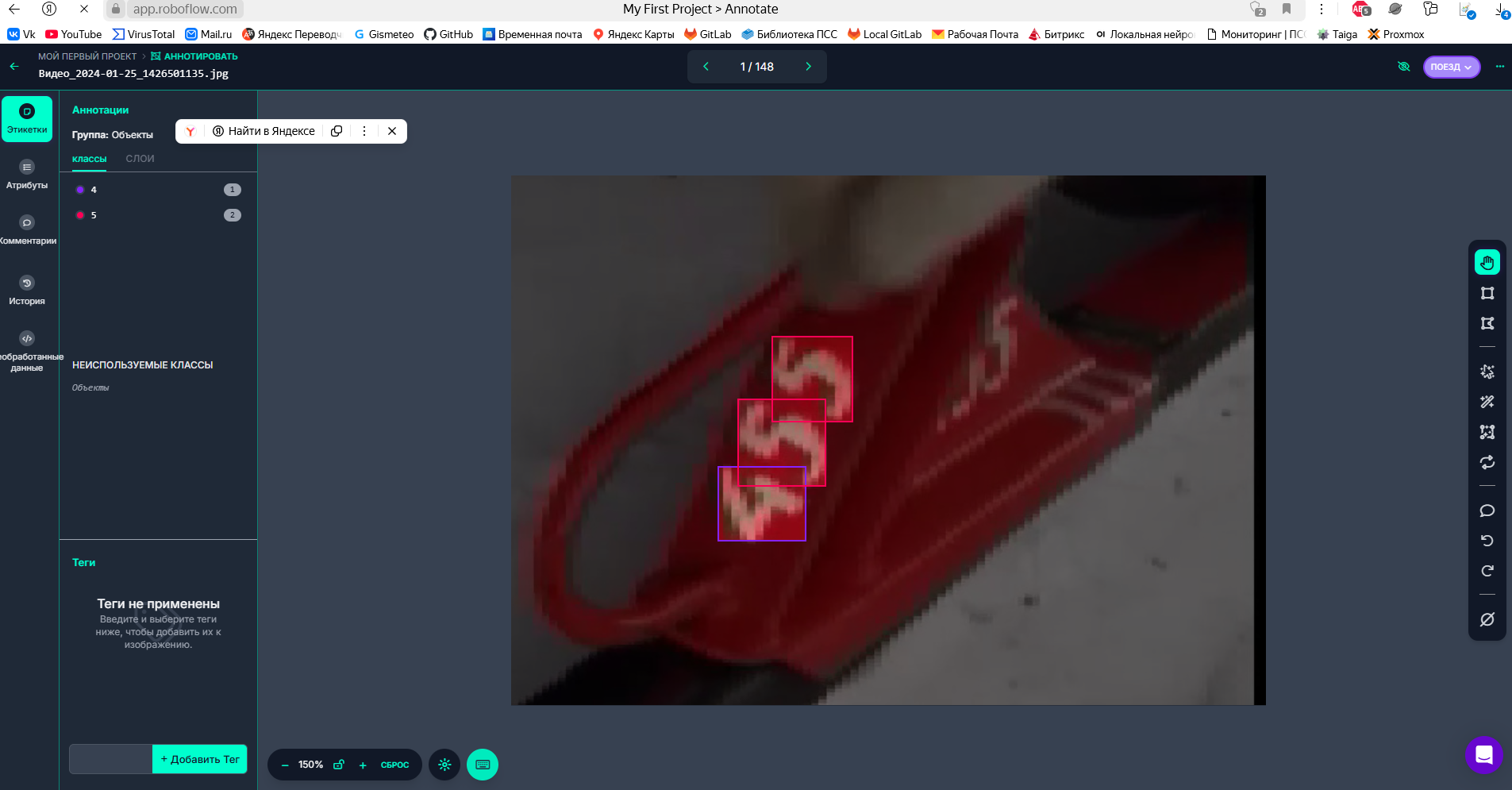
Проверка происходила командой:

yolo task=detect mode=predict model=yolo11n.pt conf=0.25 source='scale\_1200.png' save=True

где «scale\_1200.png» - Название фото

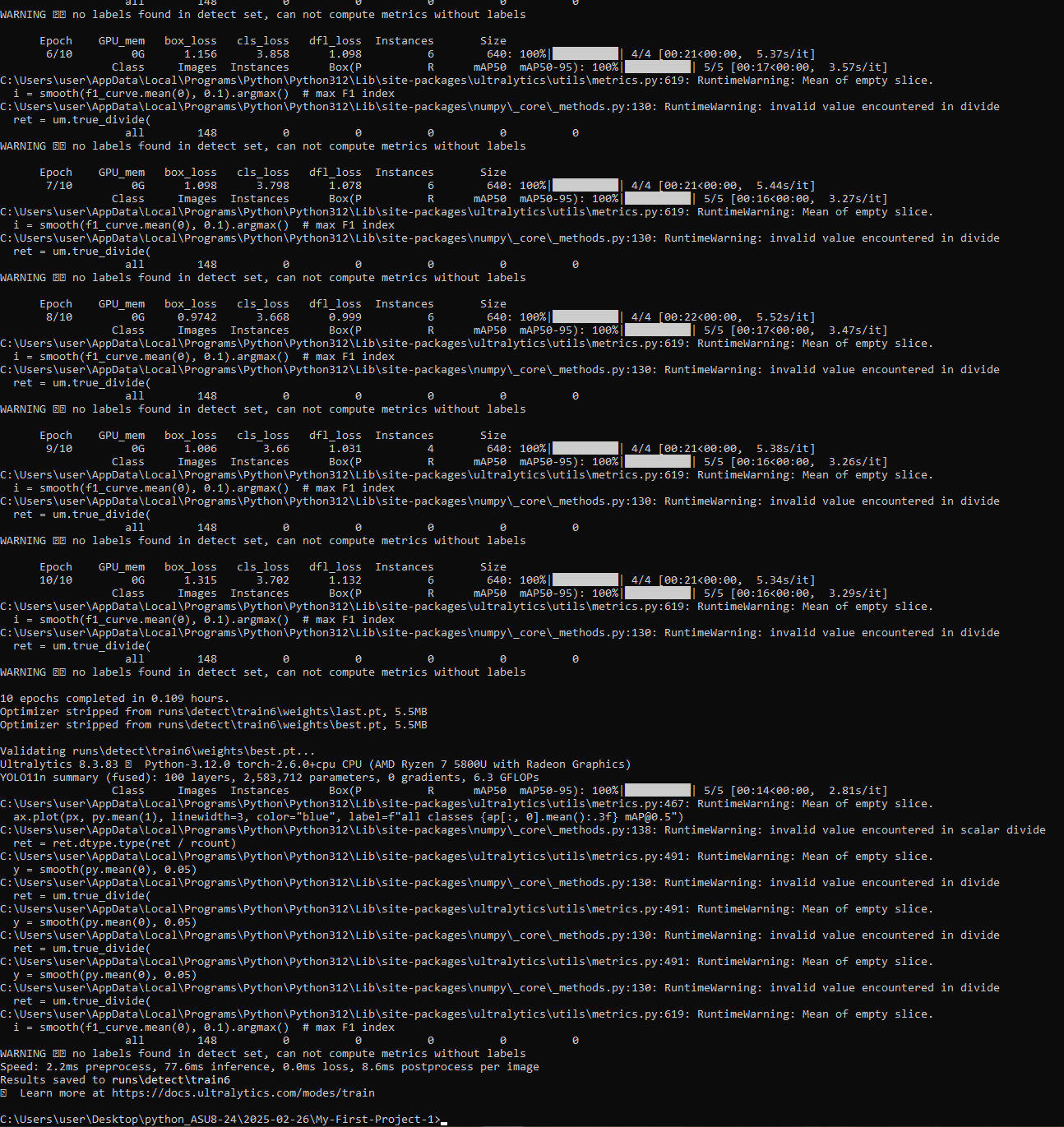
Работает очень хорошо, удивило распознавание Змей-Горыныча, ведь перевод слова kite – воздушный змей. Хотя Бурундука из Чип и Дейла определила неправильно. Цифры и числа на башмаке также не определила.

Создал свой датасет через roboflow. За пол час сделал только 50 изображений и сразу перевел их в раздел train. Думал, этого будет достаточно для проверки на хорошей фотографии.

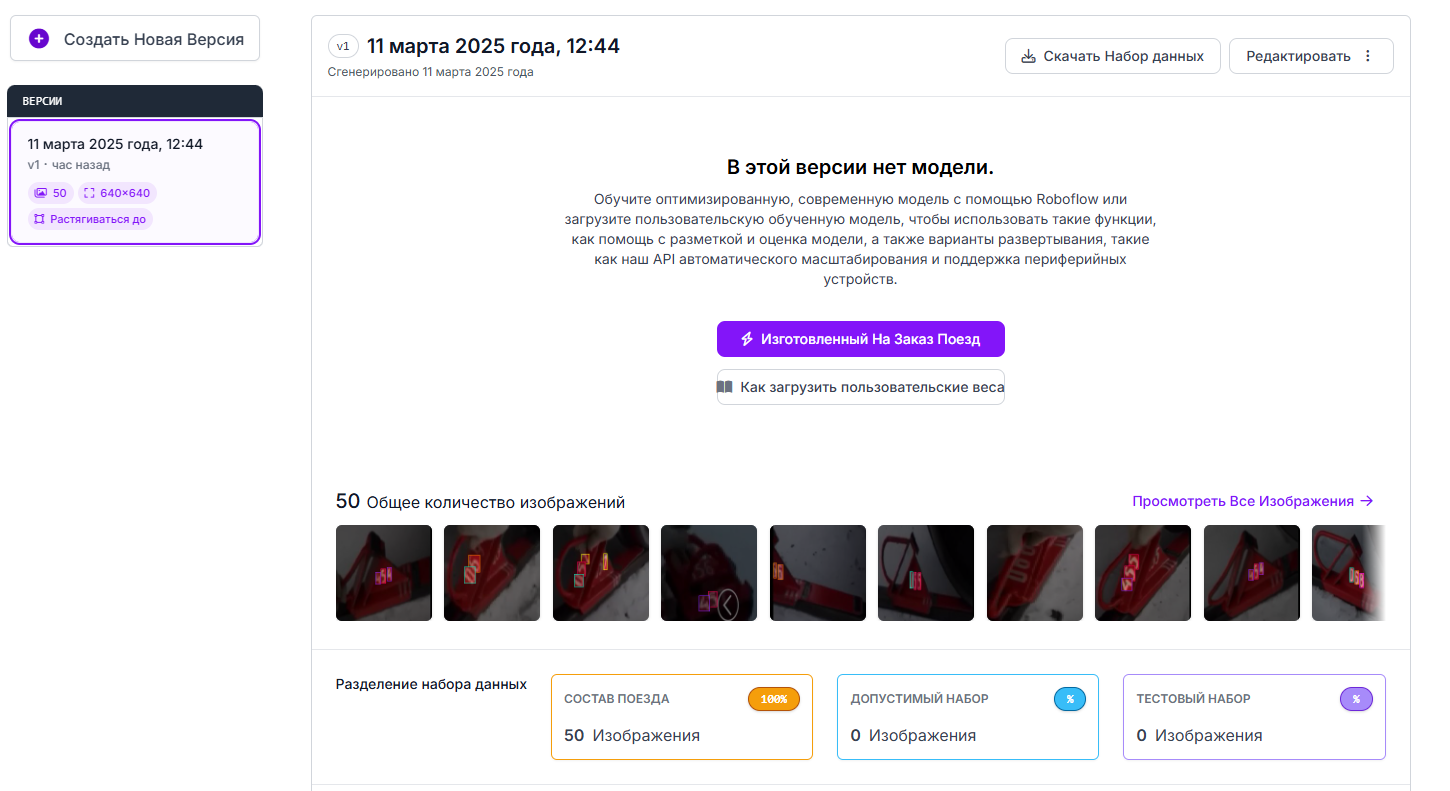


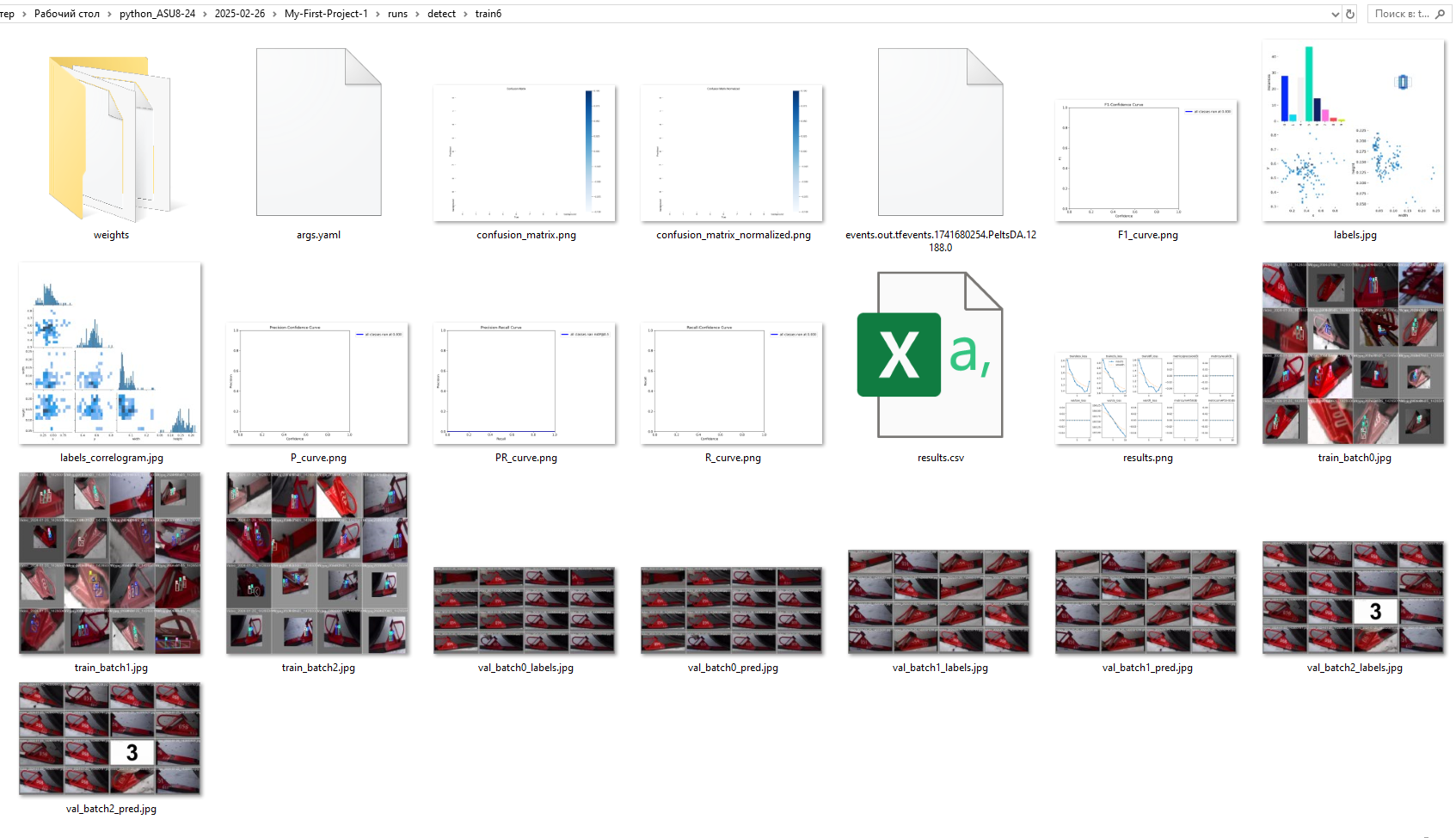
Создал модель. Скачал проект локально. В папке My-First-Project-1 теперь папка train для тренинга и data.yaml для структуры папки. Запустил обучение командой:

yolo task=detect mode=train model=yolo11n.pt data=data.yaml epochs=10 imgsz=640 plots=true



Теперь есть модель, натренированная на YoloV11 (брал 50 изображений, в тапку эти «башмаки»). Она находится на ./2025-02-26/My-First-Project-1/runs/detect/train6/weights/best.pt. Также можно посмотреть результаты тренировки модели в папке ./2025-02-26/My-First-Project-1/runs/detect/train6





**ВЫВОД:** В итоге, потратив огромное количество времени, понял, что сильно облажался с разметкой датасета. Делать всё заново в надежде, что всё заработает – можно, и может быть, даже нужно, но результат +- предсказуем.

Смысл ясен и понятен. В итоге, также хочется добавить, что сравнивать Tesseract и YOLOv8 в лоб – такое себе занятие. Да, оба могут работать с текстом, но у них вообще разный подход.

Tesseract – это OCR, он берет картинку и пытается превратить её в текст, но ему нужен четкий, читаемый шрифт. А YOLOv8 – это детектор объектов, он видит текст как просто области с символами, но читать их сам не умеет.

Лучший вариант – использовать их вместе: YOLOv8 находит нужные цифры на изображении, а Tesseract уже их распознает. Тогда будет и быстро, и точно.

**Сравнение Tesseract и YOLOv11**

|  |  |  |
| --- | --- | --- |
| **Критерий** | **Tesseract (OCR)** | **YOLOv11 (Обнаружение объектов)** |
| **Основное назначение** | Распознавание текста | Обнаружение объектов (включая цифры) |
| **Точность** | Высокая на четких изображениях, но падает на размытых или рукописных текстах | Отлично находит области текста, но не распознает его |
| **Скорость** | Довольно медленный на больших изображениях | Быстрее Tesseract, особенно на GPU |
| **Устойчивость к помехам** | Плохо справляется с низким качеством, наклоненным и мелким текстом | Лучше видит текст в сложных условиях |
| **Где лучше использовать** | Для преобразования текста в цифры и буквы | Для поиска текста в изображении перед OCR |
| **Оптимальный вариант** | Сам по себе не находит текстовые области | Использовать перед Tesseract для выделения текста |