Отчет о проделанной работе по разработке блока распознавания

Настройка Tensorflow

Для разработки блока распознавания был использован язык python и платформа для работы в области машинного обучения TensorFlow. Первое, что было сделано — был склонирован репозиторий TensorFlow/Object\_detection. Далее были установлены все недостающие библиотеки и запущен скрипт инициализации для проверки правильности установки (рисунок 1).

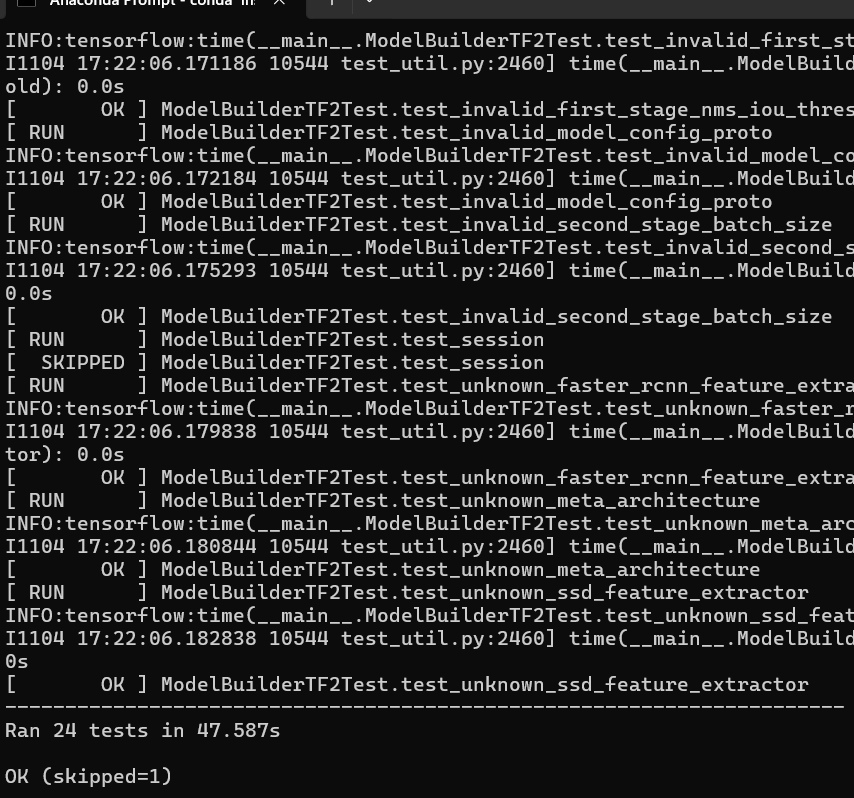


Рисунок 1 — Валидация установки

Создание скриптов для запуска распознавания  
Далее были внедрены скрипты для запуска распознавания объектов на статичном изображении.



Рисунок 2 — Часть скрипта распознавания на изображении

Выбор модели

TensorFlow предоставляет множество pre-trained моделей для распознавания изображений, но в условиях ограниченных вычислительных ресурсов была выбрана модель ssd\_efficientdet\_d0\_512x512\_coco17\_tpu-8. Модель основана на иерархии Single-Shot Detector и не является оптимальной. Использована для создания рабочего прототипа и будет заменена в будущем на более мощную.

Сбор датасета

Так как возможности собрать собственный датасет повреждений ветрогенераторов (достаточного уровня качества) нет, то был использован датасет, размещенный в открытом источнике, содержащий около 5000 изображений и только три типа повреждений: erosion, crack, damage (эрозия, трещины, урон). Этого оказалось достаточно для создания рабочего прототипа системы распознавания, но в будущем датасет будет заменен на более качественный и полный. Данные были представлены в формате tfrecord.

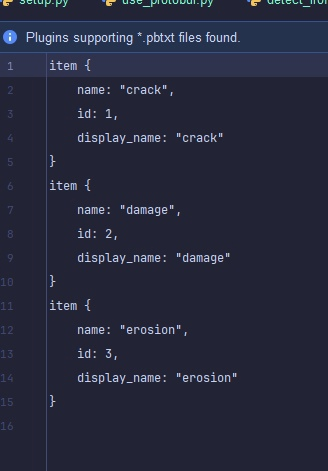


Рисунок 3 — Классы объектов для распознавания

Тренировка модели  
Тренировка модели заключается в повторной тренировке верхних слоев pre-trained модели. Схема на рисунке 4. Ниже представлены программы для запуска тренировки (рисунок 5) и параметры модели для тренировки (6-8).

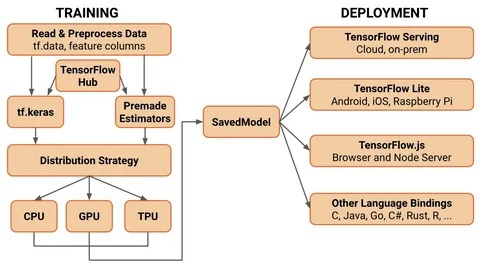


Рисунок 4 — Общая схема тренировки



Рисунок 5 — Часть программы для запуска тренировки

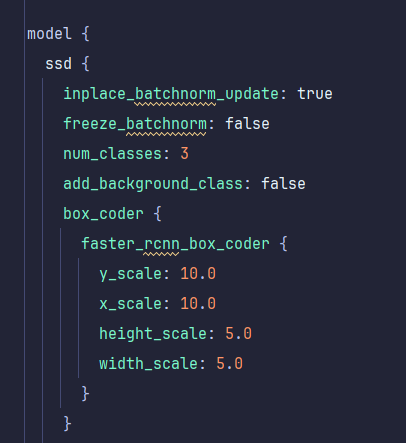


Рисунок 6 — Часть конфигурации модели (установка количества классов)

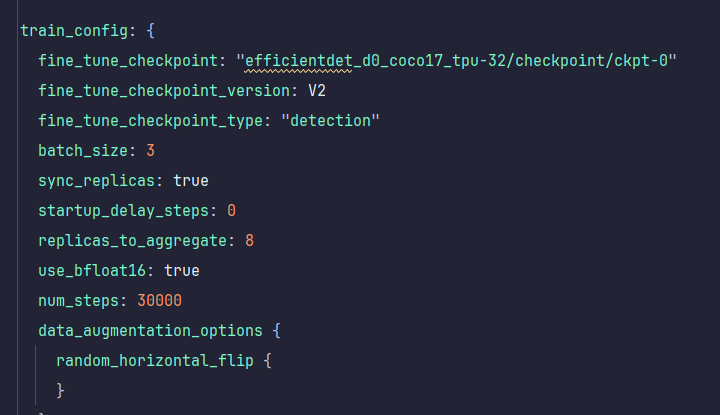


Рисунок 7 — Часть конфигурации модели (установка количества шагов и batch\_size)

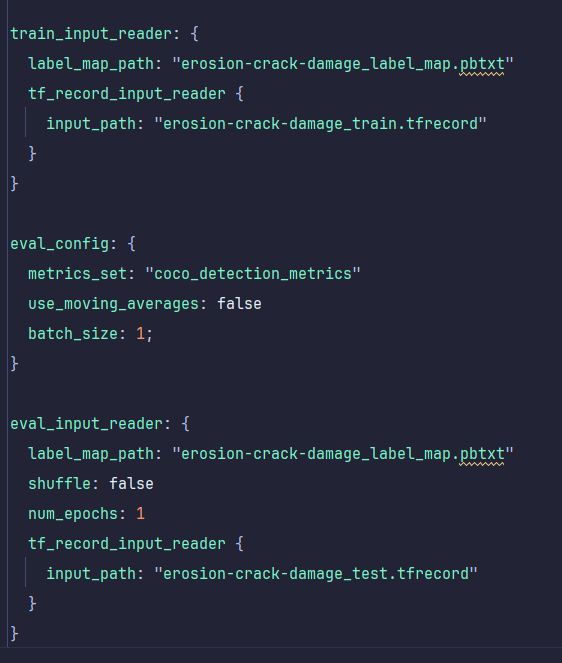


Рисунок 8 — Часть конфигурации модели (установка директорий расположения лейблов)

Процесс длился несколько часов, и был нестабилен. Это произошло из-за недостаточного качества датасета и слабой использованной модели. Создать статистику и графики по тренировке так же не удалось в силу слишком маленькой выборки изображений (датасета). Однако после тренировки модель смогла распознать изображения с точностью до 70% (10-11). Процесс тренировки показан на рисунке 9.

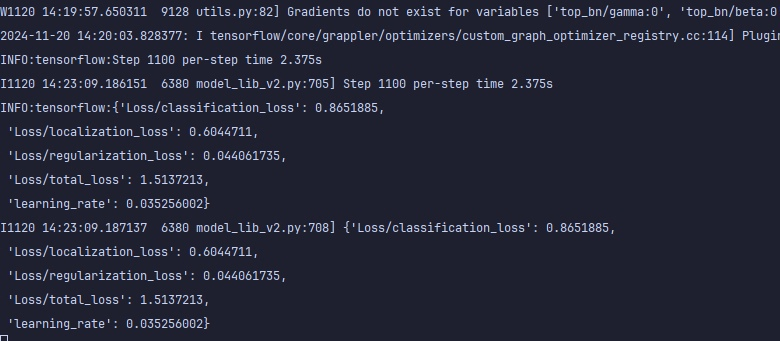


Рисунок 9 — Процесс тренировки модели

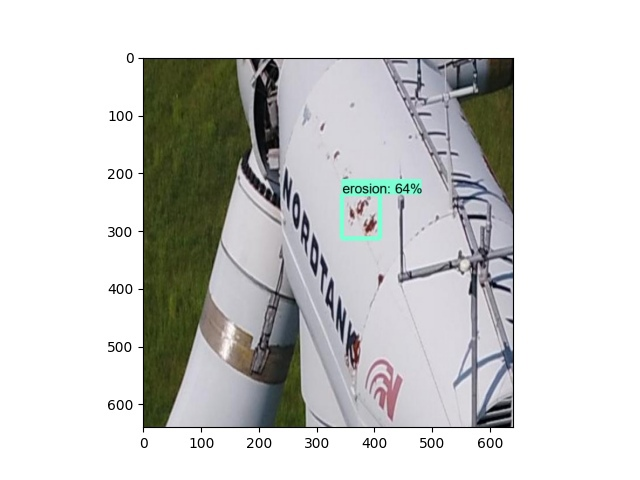


Рисунок 10 — Пример успешного распознавания

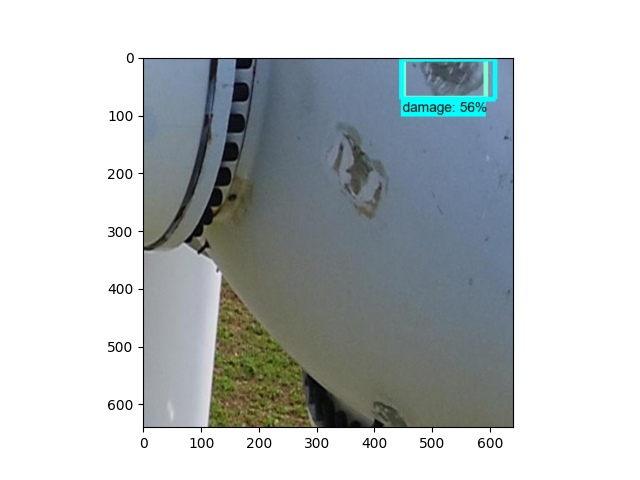


Рисунок 11 — Пример успешного распознавания

Следующий этап

В будущем модель от tensorflow будет заменена на одну из модели архитектуры YOLO (You Only Look Once). Это необходимо, посколько YOLOv5 или YOLOv8 значительно более удобны для решения прикладных задач в области распознавания объектов, чем модели tensorflow. К тому же, подготовить датасет под этот тип модели проще.