#### 0. Обзор проекта и структуры

Занятие начинается с экскурсии по репозиторию: преподаватель открывает дерево каталогов и показывает, что в папке core находятся все базовые классы — датасет, функции потерь и трейнеры, в директории runs автоматически появляются логи TensorBoard, а скрипт segmentation\_exp\_seria.py в корне запускает всю серию экспериментов. Объясняется, что такая раздельная, «чистая» архитектура облегчает масштабирование, делает проект читаемым и повторяемым, и рекомендуется студентам придерживаться схожей структуры в собственных работах.

#### 1. Знакомство с данными и работа с COCO датасетом

Откройте ноутбук Json\_using.ipynb, выполните все ячейки и покажите исходный json файл разметки, в котором десятки категорий. Продемонстрируйте словарь, сворачивающий все эти категории в два выходных класса — «Норма» и «Патология». Затем перейдите в core/datasets.py, объясните, что такое формат формат COCO (images, annotations, categories). Покажите, как в классе датасета из json извлекается маска и накладывается на изображение

Продемонстрируйте примеры изображений с нанесенными контурами из скрипта Json\_using.ipynb и игрушечный подсчет весов прямо в ноутбуке

#### 2. Обучение классификатора

Откройте ноутбук train\_classifier.ipynb, покажите замену финального слоя у EfficientNet b0 на выход с нужным количеством классов. Покажите и опишите устройство трейнера для сетей классификаторов из файла core/trainers. Покажите работу сети в течении нескольких эпох

#### 3. Обучение сегментатора

Откройте train\_segmentator.ipynb. Используя библиотеку SMP инициализируйте SOTA архитектуру FPN c энкодером. EfficientNet. Откройте код SimpleSegmentationTrainer, объясните, его устройство, сравните с трейнером для классификации.

#### 4. Обучение регрессора

Напомните, что регрессия в медицине — это предсказание непрерывной величины: площади поражения, объёма опухоли или, скажем, возраста пациента по изображению. В ноутбуке train\_regressor.ipynb демонстрируется, как выход сегментатора агрегируется в одно число (сумма пикселей маски) и как тренер оптимизирует среднеквадратичную ошибку. Расскажите студентам, что можно было пойти другим путем и используя сеть классификатор не применять сигмоиду и использовать логиты или при больших значения как в нашем примере использовать слой с экспоненциальным усилением логитов

#### 5. Множественные эксперименты с подбором функций потерь

Финальный блок посвящён лосс функциям. Откройте core/losses.py и по очереди объясняет, чем отличается обычная BCE от взвешенной BCE, зачем нужен Focal loss, как считается IoU loss и почему полезно складывать IoU и Focal в Combo loss.

После этого запустите скрипт segmentation\_exp\_seria.py, который последовательно обучает шесть моделей: сначала BCE без весов, затем BCE с пиксельными весами, потом Focal loss с весами, чистый IoU, IoU с классовыми весами и, наконец, Combo loss в двух вариантах.

У каждого эксперимента собственное имя, и все логи отправляются в runs/. Откройте TensorBoard с готовыми рез ультами экспериментов покажите трейн и вал IoU. Акцентируйте внимание на их различиях напомните студентам что такое переобучение. Покажите им что худшая ошибка на валидационных данных может быть лучшей на тренировочных данных. Покажите, как радикально меняется результат использования BCE, при использовании всего простых пиксельных весов.

Также акцентируйте внимание на том том, как использование весов или иных лосс функций может влиять на скорость достижения метрик.

Проведите финальное сравнение результатов всех функций потерь предложите студентам проголосовать за функцию которую они считают лучшей.

Вероятно, они выберут IOU\_loss с классовыми весами, что кажется очевидным сравниваем мы нейронные сети по IOU\_метрике на val. Отметьте, что в реальных задачах чаще всего сегментатор будет использоваться как база для классификации и сегментации тоже, а там IOU loss может показывать значительно худшие результаты.

**Демонстрация работы модели (опционально)**

Запустите скрипт Test\_model (как обычно смените путь к вашему расположению семинара). Он нарисует маски предсказаний модели поверх изображения и настоящий контур патологии. Сделайте предсказания для моделей обученных по разному спросите студентов видят ли они какую то систематическую разницу в предсказаниях. Если ваша машина слабая можете использовать картинки уже сохраненные в юпитере, для минималистичной демонстрации.

**Домашнее задание**

Обучить несколько сетей классификаторов и сегментаторов для поиска только “пневмонии”, “гидроторакса”, “кальцинатов” в легких посмотреть, как метрики качества по каждому классу повлияет функция потерь, классовые и пиксельные веса.