Report: Команда 25

Краткое описание проекта

`Название проекта`: Распознавание пневмонии на рентгенограмме органов грудной клетки.

Сейчас занимаемся задачей `классификации`.

Текущий прогресс

1. Собраны данные, проведён разведочный анализ данных. Выяснилось, что данных достаточно и они хорошего качества. Есть классы и bbox для задачи детекции.
2. Построили бейзлайн.
    a. Особенности: хотели чтобы датасет помещался в память, и чтобы модели не обучались дольше 5-10 минут.
    b. Попробовали обучение без сжатия изображений, со сжатием, классику cv (SIFT и HOG), снижение размерности при помощи SVD-разложения.
    c. Наилучшие результаты показало SVD-разложение.
    d. Попробовали линейные и нелинейные модели классификации.
    e. Нелинейные модели дают небольшой прирост в качестве.

Бейзлайн:

```
PCA via SVD(1024x1024 → 128) + SVM(rbf kernel)
```

В MVP выбрали облегченную версию модели бейзлайна. Т.к. приоритезировали быстродействие и занимаемую память. SVD(from 1024x1024 → 128) весит больше 1Гб.

```
Resize(1024x1024 → 128x128)
+
PCA via SVD(128x128 → 32)
+
Random Forest
```

Сделали MVP, состоящий из fastapi(бэкенд) и streamlit(фронт).

Дальнейшие планы

Классификация изображений с использованием deep learning моделей.	Ориентировочно в 3 модуле
Реализация задачи сегментации. Выявление лёгочных затемнений, характерных для пневмонии.	Ориентировочно в 4 модуле

Структура проекта

Текущая структура проекта

- FastAPI-приложение
  - fastapi_app.py
  - data_models.py
- Streamlit-приложение
  - streamlit_app.py

Планируем добавить

- Telegram-бот

Функционал MVP

Фронт Streamlit комплементарен FastAPI бэкенду. То есть MVP состоит из двух частей и суммарно реализует функциональность:

1. Получение предсказания на одном изображении (наша модель или дообученная)
2. Загрузить свой датасет
3. Проводить эксперименты: дообучать различные доступные модели на загруженных данных.
4. Оценка качества экспериментов: метрики и графики.
5. При инициализации сервиса подгружаются: предобученная модель, дополнительный датасет

Инструкции: развёртывание, дальнельная разработка и использование

Инструкции по развёртыванию docker контейнера

1. Клонируем репозиторий
    a. git clone https://github.com/Amlaith/medical_diseases_recognition.git
2. Переходим в директорию с четвертым чекпоинтом
    a. cd checkpoint_4_service
3. Запускаем контейнеры
    a. docker-compose up
4. Открываем streamlit-приложение в браузере
    a. http://localhost:8501/

fastapi_app.py

Идейно бэкенд берёт на себя максимальную долю вычислений, так как при деплое может использоваться бесплатная версия Streamlit Cloud.

Метанастройки streamlit_app.py

Настройка визуальных эффектов успеха

```
NEW_YEAR_EDITION = True
```

Если True, то визуальным эффектом будет `снег`, иначе `шарики`.

Указывать root бэкэнда, к которому надо обращаться. Та ссылка, по которой будет показано "App is healthy" или его аналог.

```
BASE_URL = "http://localhost:8000/"  # Root backend endpoint
```

Пользовательская инструкция

Как получить предсказание по картинке

1. Перейти на страницу «Получение предсказаний»
2. Загрузить изображение. Либо получить демо.
3. Нажать на кнопку
4. Увидеть результат

Как дообучить модель

1. Выбрать данные для дообучения.
    a. Загрузить свой датасет: архив с изображениями и labels.csv
    b. Скачать демо датасета, затем его загрузить.
    c. Использовать датасет по умолчанию
2. Дообучить модель.

    a. Можно выбрать некоторые гиперпараметры
3. Сравнить свою модели/свои модели с нашей дефолтной на вкладке «Сравнение моделей».
    a. Посмотреть на метрики
    b. Посмотреть на графики

Демонстрация работы

YouTube плейлист