Автореферат выступления

Слайд #1

Слайд #2, Архитектура проекта

Наш подход в решении задачи определения координат по фотографии местности заключается в использовании различных подходов и дальнейшем их ансамблировании.

Архитектурно наша работа является частью студенческого проекта «ИИ-мониторинг биоценоза Новозыбковского округа на основе данных открытых источников». Этот проект позволяет студентам реализовывать собственные решения, используя и уже собранные данные (фотографии, видео, тексты новостей и статей) и результаты работ своих товарищей.

Поэтому проект имеет микромодульную структуру. Каждый модуль может использовать общую базу данных, но может вносить изменения только в свою часть. Таким образом студенты получают определенную свободу действий в выборе своего подхода, не боясь повлиять на чужие данные.

Побочным, но приятным эффектом, получаемым от такой структуры, является возможность использовать отдельные модули самостоятельно. Достаточно согласовать структуру баз данных, и Вы можете использовать нужный отдельный модуль самостоятельно.

Слайд #3, Модуль 0100-GeoGuessrr (Жданов)

Этот модуль является реализацией классического алгоритма GeoGuessrr: мы разбили город на квадраты со стороной 100 м. Использую АПИ ВКонтакте для каждого квадрата собрали фотографии, относящиеся к этому квадрату. Затем обучили нейронную сети MobileNet v3 и EfficientNet b5 на этих данных. Полученные результаты Вы видите на слайде. Мы видим явное переобучение нейронной сети. Связано это со следующими причинами:

1. Очень мало исходных данных. Для большинства классов количество фотографий не превышает 10. Этого очень мало для полноценного обучения.
2. Много «грязных» данных. Это не только фотографии местности, но и селфи, фотографии в помещениях, гаражах. Много снимков домашних праздников, одежды, рассады, мебели и пр. Они явно не годятся для определения местности. Для решения этой проблемы мы вручную отобрали подходящие снимки (мы их назвали GeoGood) и неподходящие (соответственно, GeoBad) и обучили MobileNet v3 различать их. В процессе разработки, мы решили выделить и третий класс, «В помещении». Сейчас он не используется, но в последующем мы планируем применять и его для определения помещения и/или кабинета, в котором сделан снимок.
3. Много снимков определены Вконтакте с недостаточной точностью. Т.е. один снимок может относиться к нескольким классам.

Слайд #4, Модуль 0400-Cosin ViT (Жданов)

Это наш второй вариант решения, поиск похожих изображений. Для этого мы используем расчет минимального косинусного расстояния между эмбедингами изображений, полученных визуальным трансформером clip-ViT-B-16. Такой подход учитывает семантическую схожесть между изображениями. Он не требует обучения, относительно быстрый. И как показывает практика, пока дает результаты заметно лучше предыдущей модели. (до 54% на данный момент) Но мы считаем, что с увеличением количества фотографий, точность первого подхода должна превысить точность этого модуля.

Оба эти подхода уже сейчас помогают и существенно ускоряют ручную разметку, реализованную в админке.

Слайд #5, Модуль 0300-YandexMap (Макаркина)

Панорамы Ягдекс.Карт являются еще одним, очень «богатым» источником изображений с точной геолокацией. Однако АПИ Карт не позволяет получить снимки в виде отдельных изображений. Поэтому мне пришлось использовать обходные пути получения этих данных. *(Как – настоящие леди на такие вопросы не отвечают, потому-что настоящие джентельмены их не задают)*. Ничего противозаконного конечно, но парсинг был автоматический, что запрещено в пользовательском соглашении. Полученные фотографии были использованы при обучении GeoGuessrr для обогащения датасета.

Слайд #6, Модуль 0600-Gemma 3 (Макаркина)

В процессе работы на парсингом, я обратила внимание, что на фотографиях Я.Карт, кроме непосредственно снимков местности есть текстовые данные: название улицы, номера домов. И я решила, что их можно использовать для обучения image-to-text модели. И текстовой составляющей будут именно названия улицы и номера домов. В качестве модели я выбрала unsloth/gemma-3n-E4B-it. Это квантизированная до 4 бит версия культовой модели gemma-3n от компании Google.

Я использовала EasyOCR для распознавания названия улицы, благо, оно всегда находится на одном месте в верхнем левом углу и наивный алгоритм поиска названия домов по шаблону: вертикальная полоса шириной 1 пиксель с цветами, характерными для фона этих меток (с некоторым допуском, который является артефактом сжатия по алгоритму JPEG) И с ее обучением связан забавный момент: первая версия показала изумительные результаты на тестовой выборке из панорам (точность до 99,5%), при этом на тестовой выборке на фотографиях из ВК не распознано НИ ОДНОГО СНИМКА! В процессе исследования этого феномена, я пришла к выводу, что модель сама научилась использовать тексты названия улиц и номеров домов, которые я не стирала. Поэтому модель пришлось переобучать на отдельном датасете из панорам, но с закрашенными участками изображения с названиями улиц и номеров домов. Даже метку слева вверху «Открыть в Яндекс.Картах» тоже пришлось закрасить, иначе все снимки были с этим текстом.

Слайд #7, Модуль 0650-NER (Якушенко)

Кроме панорам Я.Карт, географические названия очень часто встречаются в новостях, статьях, постах. И если мы научимся сопоставлять их с географическим координатами, то мы получим огромный массив геоданных, которые можно использовать, даже если метаданные из изображений удалены. Для решения этой задачи я пробовал использовать готовые решения, например модели ru\_core\_news\_lg из пакета Spacy для выделения именованных сущностей. Но они не совсем подходят для наших целей, так как в тексте часто встречаются названия, которые не относятся именно к месту, где проведена сьемка. Поэтому в админке я сделал блок для ручного выделения нужных NER, и по мере накопления данных будет происходить дообучение моделей на нашем корпусе. Накопленные данные также можно использовать в «Геокодере» от Яндекс, моделях Text2Geo, больших языковых моделях.

Слайд #8, Модуль 0500 – K-Means (Якушенко)

И то самое решение, позволяющее объединить результаты, полученные предыдущими методами и выбрать из них самый точный.

Как мы видим, большинство изображений, с разной точностью конечно, но примерно правильно попадают в нужный нам класс. Наша задача, выбрать их, отбросить откровенно неподходящие результаты и, по возможности, улучшить итоговую точность. Для этого мы:

1. Кластеризуем по координатам все полученные геоточки. Количество классов выбираем по минимуму метрики silhouette\_score;
2. Кластер, с минимальной метрикой, и в который входит максимальное число точек считаем верным;
3. Вычисляем среднее для всех точек верного кластера

Таким образом, даже для большого числа неверных фото мы можем получить верный результат. При этом средняя точность будет выше за счет компенсации выбросов, за счет усреднения данных

Слайд #9, Перспективы развития (Жданов)

1. Увеличение числа используемых источников – 2ГИС, соцсети, сайты, блоги, ТМ-каналы