Олимпиада PSRS Вычислительные технологии | Лист 1

Задание (V2)

Для решения задач сегментации текстовых и графических областей в документах, без использования нейронных сетей, можно использовать алгоритм, основанный на комбинации морфологического анализа, статистики связанных компонентов и текстурных признаков. Основная идея заключается в использовании различий между структурой текста и фотографий, так как текстовые блоки состоят из мн-ва контрастных мелких эл-тов с регулярным расположением, а изображения – крупные области со сложной текстурой и плавными цветовыми переходами, к тому же, на изображениях используется большая палитра цветов, в отличие от текста.

На первом этапе алгоритма будем выполнять предобработку документа. Конвертируем его градации серого с помощью функции "CV2.cvtColor" из библиотеки OpenCV, после чего применим адаптивную бинаризацию, используя CV2.adaptiveThreshold для компенсации неравномерной освещенности. Далее, морфологическая операция закрытия CV2.morphologyEx с ядром 3×3 будет объединять отдельные символы в блоки – то есть в связанные текстовые строки, сохраняя при этом ~~границы~~ границы крупных графических эл-тов.

Анализ связанных компонентов будет осуществляться через функцию cv2.connectedComponentsWithStats, она выделяет все связные области бинарного

изображения. Для фильтрации текстовых блоков будем использовать при ключевых параметра:

1) площадь области (от 100 до 10^5 пикселей)
2) соотношение сторон ($0{,}1 < width/height < 10$)
3) плотность краев, которое будем рассчитывать через применение оператора Собеля.

Текстовые регионы показывают более высокую плотность границ, по сравнению с фотографиями.

Классификация регионов выполняется через анализ распределения компонентов. Текстовые блоки группируются в строки с помощью вероятностного преобразования Хафа, для этого можно использовать: skimage.transform.probabilistic_hough_line.

вер-ое преобразование Хафа обнаруживает линии базовых уровней текста.

Для окончательного разделения типов областей будем использовать комбинацию из двух критериев; это контрастность, определяемая как средняя яркость между объектом и фоном, и цветовая энтропия. контрастность у текста выше, а в контексте цветовой энтропии, у фотографий наблюдается более равномерное распределение цветов.

На этапе постобработки будем устранять пересечение блоков через анализ иерархии вложенности областей. Для графических объектов можно дополнительно проверить наличие главных градиентов с помощью преобразования Фурье, к примеру ~~резкие~~ резкие переходы характерны для текста, а фото содержат широкий спектр ~~[illegible]~~ компонентов.

Оценка алгоритма:

Главные преимущества алгоритма заключаются в простоте реализации, благодаря использованию библиотекам, а так же достаточно быстрой и точной сегментации областей документа, благодаря адаптивной бинаризации, морфологическому анализу и стат анализу связанных ~~[illegible]~~ компонентов.

Однако, этот алгоритм имеет некоторые недостатки, а именно он может столкнуться с несовсем точной сегментацией особых документов, ~~[illegible]~~ с шумами, некачественным изображением информации. Так же при наличи хаотично распределенных данных в документе (расположение изображений и текста) в большом количестве, алгоритм может потребовать более точного подбора пороговых значений, что усложняет использование алгоритма, как универсального инструмента для всех документов.

Задание (№3) Умные системы. | лист 4

Разработаем систему умного полива растений. Главное преимущество нашей системы – в адаптивности к внешней среде. ~~Так как~~ Так как растения нуждаются в поливе не только в определенные временные промежутки, ~~а в зависимости~~ а еще и в зависимости от окружающей среды и внешних условий.

На грядке поставим следующие датчики:

1) Датчик ветра

2) Датчик солнца

3) Датчик влажности почвы

Так же, на грядке, будут расположены системы орошения (устройства для полива) естественно

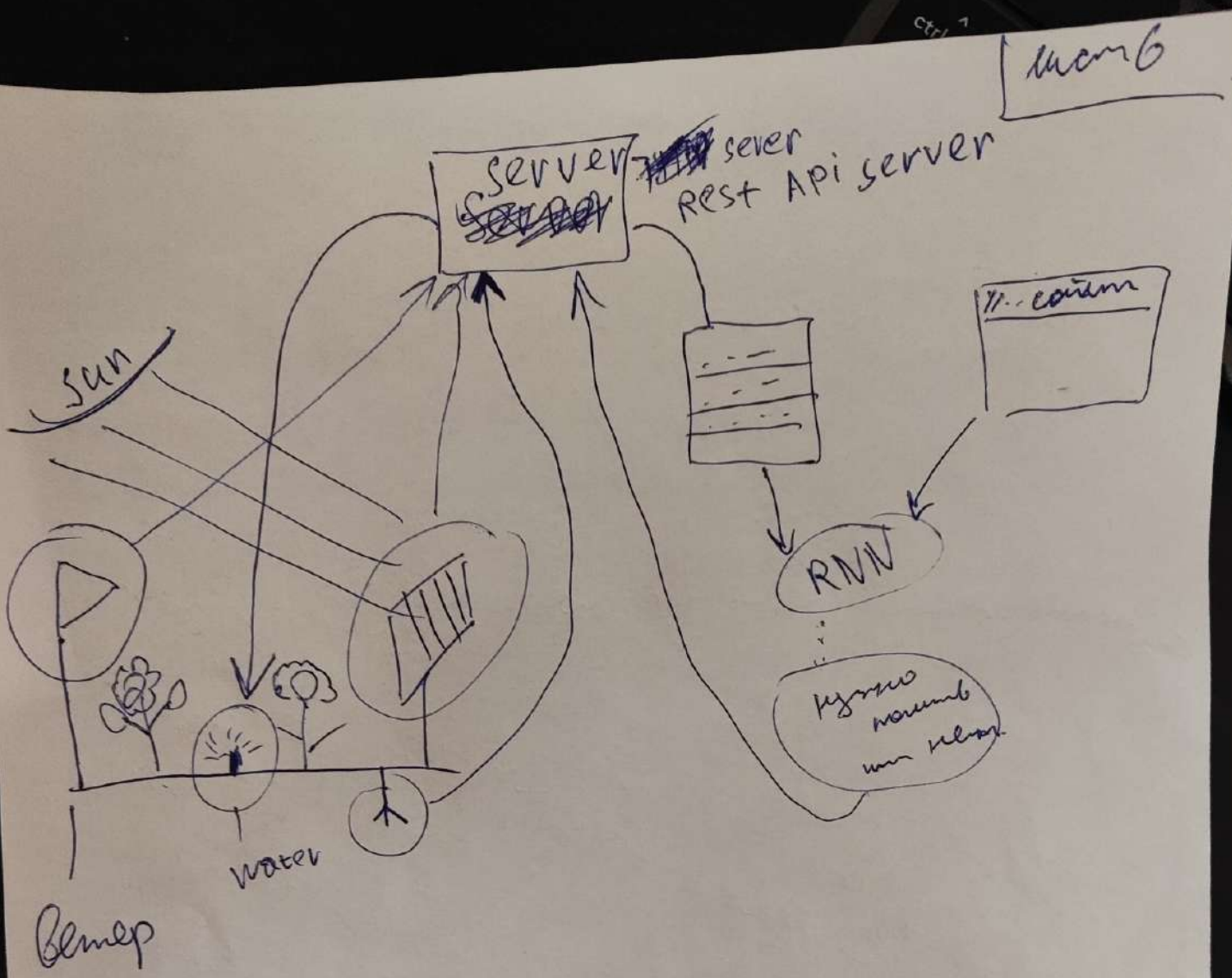
Основой умного полива будет искусственный интеллект, а именно рекуррентная нейронная сеть – RNN, которая способна работать с временными последовательностями данных. Архитектуру сети будем использовать из библиотеки tensorflow, предварительно обучив модель на синтетических данных.

Данные с датчиков на грядке будут поступать через сервер RestAPI, что гарантирует безопасность обмена.

Помимо датчиков, наша система

будет ~~подключаться~~ парсить данные с сайта/источника с информацией о погоде и ее прогнозах. Далее, поступившие данные передаются в RNN модель и она выдает ответ, нужен ли полив сейчас или нет. В работу сети на данном этапе ~~я~~ не будем углубляться, так как это уже отдельный большой и сложный этап разработки. Предствим, что наша нейросеть уже хорошо обучена и дает корректные указания для полива, учитывая ~~к примеру, прогноз~~ все данные с датчиков и прогноз погоды. К примеру, если по показаниям влажности почвы и ветру и солнца у нас долго сухая высыхали ~~в~~ почвы, но по прогнозу в течение скорого времени пойдет дождь, то ~~почвы~~ модель скажет, что полив не нужен, так как если полить и пойдет дождь, то влажность будет слишком высокая.

И так, к примеру, дождя не ожидается, а показания почвы ~~и~~ всех датчиков ~~по~~ показывают, что требуется полив, тогда модель говорит полив и через Rest API сервер отправляет указание включить полив до тех пор, пока показания с датчиков не покажут отправить информацию о ~~при~~ нужном содержании влаги в почве – тогда прекращаем


server
sever
Rest Api server
sun
RNN
water