УДК 004.8, 004.91

ИЗВЛЕЧЕНИЕ ДАННЫХ ИЗ СКАНИРОВАННЫХ ДОКУМЕНТОВ СО СХОДНОЙ СТРУКТУРОЙ

Р. Д. Саитгареев[1, [0000-0002-8184-6539]], Б. Р. Гиниятуллин[2, [0000-0001-8089-9893]], В. Ю. Топоров[3, [0000-0002-9809-5233]], А. А. Атнагулов[4, [0000-0001-9766-4804]], Ф. Р. Аглямов[5, [0000-0002-9939-7989]]

[1–5] *Институт информационных технологий и интеллектуальных систем, Казанский федеральный университет, г. Казань*

[1]srustem3@yandex.ru, [2]bulat.giniiatullin@gmail.com, [3]vladislavtoporov@gmail.com, [4]i@atnartur.ru, [5]aglyamov.fox@gmail.com;

Аннотация

На текущий момент времени значительная часть передаваемых и хранимых данных не структурирована. Количество неструктурированных данных растет большими темпами каждый год, несмотря на то, что по таким данным трудно производить поиск, к ним нельзя совершать запросы и в целом их обработка не автоматизирована. В то же время наблюдается развитие систем электронного документооборота.

Настоящая работа предлагает инструмент для извлечения данных из фотографий бумажных документов, принимая во внимание их структуру и разметку. Представлены результаты разных испытанных подходов, включая нейронные сети и алгоритмический метод, а также проведен анализ полученных результатов.

Ключевые слова*: нейронные сети, машинное обучение, извлечение структуры, извлечение структуры документов, OCR, неструктурированные данные, распознавание текста.*

1. ВВЕДЕНИЕ

Согласно аналитическим отчётам, рост систем электронного документооборота в России составляет последние несколько лет 5–7% в год [1, 2]. Эти системы должны сократить человеческие ресурсы, необходимые для обработки бумажных документов. Использование баз данных позволяет быстро находить нужную информацию в больших массивах данных и создавать различные отчёты.

Настоящая статья имеет следующую структуру: в разделе 2 сформулирована решаемая проблема, в разделе 3 сравнены существующие подходы, используемые при извлечении структуры документов и информации из таблиц. В разделе 4 описаны полученные результаты и предложена архитектура пайплайна (pipeline), также уделено внимание процессу подготовки и предобработки данных и сравнению разных подходов нахождения целевой информации в документах. В разделе 5 подведены итоги и предложены возможные улучшения предложенного подхода.

2. ПОСТАНОВКА ПРОБЛЕМЫ

90% всех хранимых и передаваемых данных в мире являются неструктурированными [3], объём этих данных растёт на 55–65% каждый год [4]. Неструктурированные данные включают в себя такие различные форматы, недоступные для поиска, как видео, аудио, изображения, текстовые документы [4]. Некоторые текстовые документы также являются неструктурированными. Счета-фактуры, платёжные документы, товарные накладные и подобные рассматриваются как полуструктурированные документы, а договоры, письма, статьи, служебные записки и ряд других могут рассматриваться как неструктурированные документы [5]. Поскольку данные в неструктурированных документах не являются доступными для поиска, их нельзя автоматически обрабатывать. Например, невозможно автоматически посчитать агрегированные данные (среднее, сумму) из платёжных документов в формате Microsoft Word.

Опрос 2011 года о неструктурированных документах (см., например, [6]) подтверждает, что компании не имеют эффективных решений по работе с неструктурированными данными. Большая часть опрошенных сообщает, что наиболее сложно работать с бизнес-документами (53%) и PDF-файлами (35%). Большинство респондентов отмечало, что количество корпоративных неструктурированных данных возрастёт на 33–47% в течение 3-х лет после опроса.

С другой стороны, структурированные данные, несмотря на их малую долю по сравнению с неструктурированными, имеют множество преимуществ: люди могут находить новую полезную информацию и получать знания из структурированных данных [7]. Преобразование неструктурированных данных в структурированный вид может предоставить эти возможности. Например, извлечение данных

из платёжных документов и их хранение в базах данных автоматизируют сбор различных метрик и агрегации (среднее, сумма).

Существует множество инструментов оптического распознавания символов (Optical Character Recognition, OCR), которые распознают машинописные и рукописные тексты в неструктурированных и полуструктурированных документах [8, 9]. Эти инструменты хорошо работают при обработке простого текста. Но в процессе конвертации неструктурированных данных в структурированный вид может потребоваться извлекать только специфичную информацию из документов, например, имя, фамилию, паспортные данные [10]. Часто это необходимо для форм, счетов-фактур, паспортов [11–13] и финансовых документов [14, 15]. Это приводит к промежуточному выводу, что необходимо выделять структуру документов для нахождения специфичной информации, которая будет использоваться при преобразовании неструктурированных данных в структурированные форматы.

3. ОБЗОР ЛИТЕРАТУРЫ

Фреймворк LayoutLM предназначен для извлечения текста и разметки и позволяет работать со смешанными входными данными, такими как эмбеддинг (числовые вектора, англ. embedding) текста, разметки, изображений [12].

В статье [13] представлен подход, который объединяет обнаружение текста и извлечение структурированных данных в едином решении, основанном на глубоком обучении. Предложенная модель распознает мультиязычный текст и надписи на изображениях.

Был разработан синтаксический анализатор SPADE (SPAtial DEpendency parser), который может обнаруживать информационные слои (information layers) и связи между ними в документах с форматом ключ-значение [11].

В статье [10] рассмотрены подходы для обнаружения таблиц и распознавания их структуры с помощью моделей TableNet, DeepDeSRT, GraphNN, CGAN, а также генетических алгоритмов. Решение обнаруживает разметку таблицы с помощью библиотеки OpenCV и извлекает текст с помощью библиотеки PDFMiner.

Главным вкладом статьи [16] является моделирование логической и извле-

чение семантической структуры более чем из миллиона электронных документов, научных статей из репозитория arXiv.org с помощью технологии глубокого обучения.

В статье [17] представлен метод извлечения данных из полуструктурированных документов с помощью анализа морфологического состава текста и обнаружения связей между названиями колонок с их текстовым содержимым. Кроме того, сопоставлены результаты анализа логической структуры с результатами анализа геометрического расположения.

Обнаружение текстовых областей с извлечением структуры документа, основанное на OCR и модели Faster R-CNN, использует технологию compounded layout segmentation. Это позволяет улучшить качество распознавания документов [14], так как объединяет извлечение признаков с анализом разметки и стилей внутри документа [12, 13].

4. РЕЗУЛЬТАТЫ

4.1. ПРЕДЛАГАЕМОЕ РЕШЕНИЕ

4.1.1. АРХИТЕКТУРА РЕШЕНИЯ

На рисунке 1 представлена общая архитектура пайплайна. На диаграмме представлены процессы обработки шаблонного (сверху) и целевого (снизу) изображений. Шаблонное изображение – это изображение документа, размеченное пользователем и используемое для извлечения из него метаинформации о взаиморасположении текстовых блоков в документе. Целевое изображение обрабатывается после завершения обработки шаблонного изображения и содержит настоящие данные, в извлечении которых пользователь заинтересован.

Шаблонные изображения обрабатываются следующим образом. Сначала происходит выравнивание изображения (деятельность 1), о котором сказано ниже в п. 4.1.3. Затем, используя размеченные пользователям целевые области, система производит поиск ближайших областей, не содержащих искомые данные (деятельность 2). В данной работе подобные области далее называются «якорными областями». Они используются для последующего определения расположения целевых областей в других документах с похожим форматом. Система вычисляет взаимное расположение выделенных и якорных областей (деятель-

ность 3). Затем она распознает текст в якорных областях (деятельность 4), используя инструменты оптического распознавания символов (OCR). В деятельности 5 система сохраняет метаданные: расположения целевых и якорных областей, их расположение относительно друг друга и распознанный текст в якорных областях.

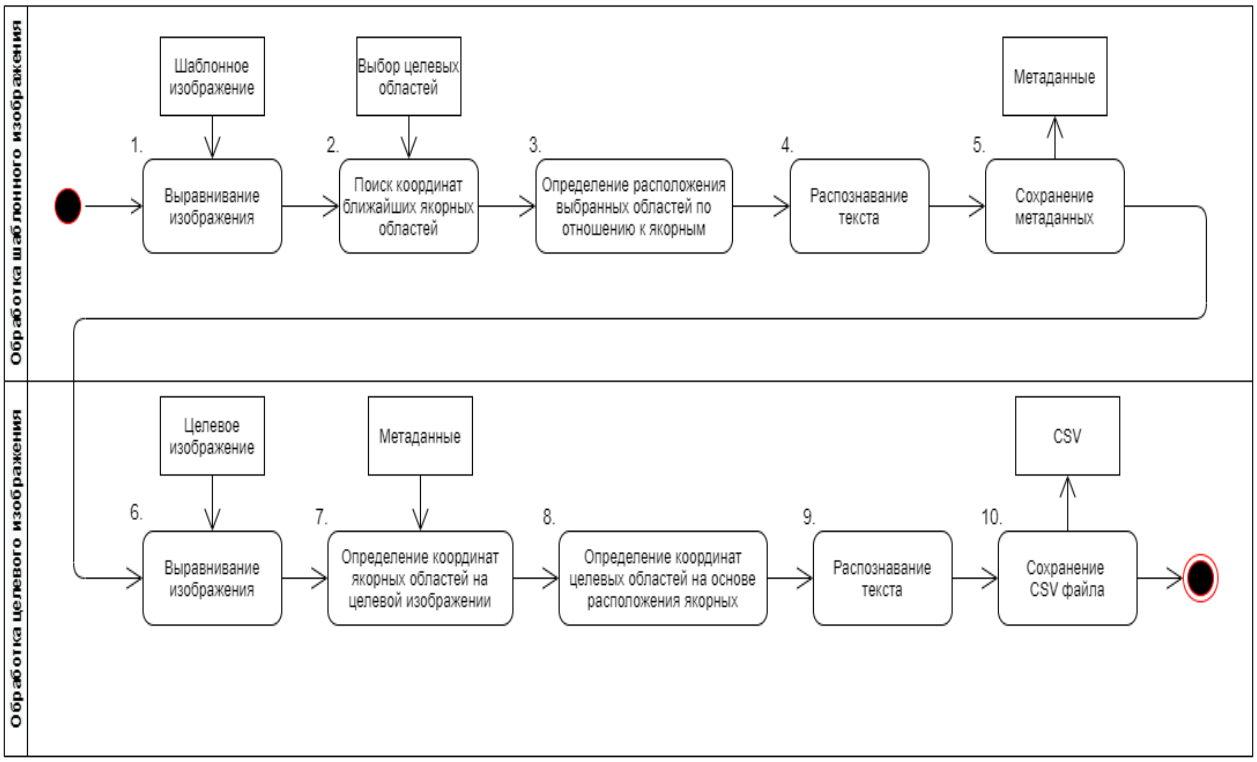


Рис. 1. Диаграмма деятельности (UML 2.5.1). Шаблонное изображение – это изображение, размечаемое пользователем и используемое для извлечения метаданных о расположении областей с текстом на документе. Целевое изображение – это изображение, передаваемое после завершения обработки шаблонного изображения. Из целевого изображения извлекаются реальные данные.

Обработка целевого изображения также начинается с его выравнивания (деятельность 6), идентичного выравниванию шаблонных изображений. Далее, используя сохраненную ранее метаинформацию, система находит якорные области на целевом изображении, соответствующие тем же областям на шаблонном изображении (деятельность 7). С использованием этой информации система ищет целевые области (деятельность 8), которые должны содержать искомые

текстовые данные. Затем происходит распознавание текста в этих областях (деятельность 9). В итоге система сохраняет полученный текст в файле в формате CSV (деятельность 10).

На рисунке 2 представлена диаграмма компонентов. UI – это интерфейс взаимодействия с системой, через который пользователь загружает шаблонное и целевое изображения и размечает необходимые области. Компонент “lifecycle” взаимодействует с пользователем и производит вычисления расположения областей на изображениях. Компонент “alignment” отвечает за выравнивание изображений, которое подробно описано в п. 4.1.3. Компонент “recognizer” – это сторонний инструмент на базе OCR, используемый системой для распознавания изображений. Компонент “recognizer” не является частью системы и не рассматривается в данной работе.

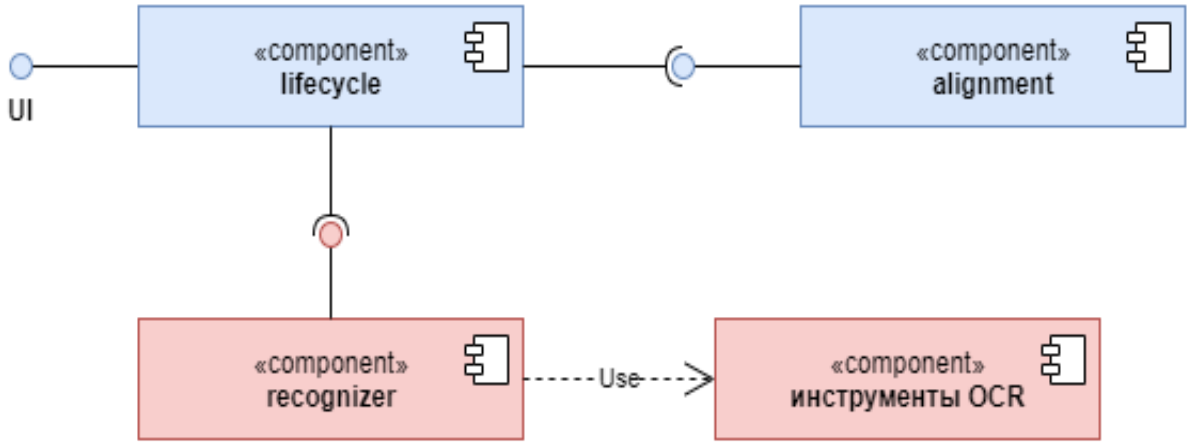


Рис. 2. Диаграмма компонентов (UML 2.5.1). Синим цветом помечены внутренние компоненты системы. Красным цветом помечены сторонние компоненты.

4.1.2. ПОДГОТОВКА ДАТАСЕТА

Для тестирования нашего решения был самостоятельно собран набор данных. В него вошли 9 видов бланков:

- бланк согласия для самостоятельного посещения ребенком детского сада,
- договор для оказания услуг дополнительного образования,
- заявление об увольнении,
- заявление для заселения в студенческое общежитие,

- заявление для освобождения от посещений учебных занятий,
- чек оплаты смены оздоровительного лагеря,
- заявление для материальной поддержки,
- ваучер для посещения детского лагеря.

Для каждого бланка было сгенерировано по 100 машинописных и по 100 рукописных документов. Все требуемые для заполнения поля были заполнены случайно сгенерированными данными. Для симуляции областей бланков, заполненных от руки, использовались текстовые вставки, использующие 7 рукописных шрифтов.

Далее якорные и целевые области были вручную размечены с помощью программы LabelMe[1]. В качестве выходных данных разметки был получен файл формата JSON, содержащий координаты областей и изображение документа, закодированное с помощью base64.

4.1.3. ПРЕДОБРАБОТКА ДАННЫХ

Была произведена аугментация (augmentation) изображений бланков, чтобы имитировать фотографии с различным углом съемки, кадрированием и поворотом для имитации входных данных в режиме реального использования. Для этих целей была использована библиотека imgaug [18]. Она позволяет настраивать угол поворота, кадрирование, добавление гауссовского шума, изменение разрешения (количество пикселей на единицу площади) и другие параметры.

Настоящие фотографии документов могут быть нечёткими и иногда содержат посторонние объекты на фоне. Было применено проективное преобразование проективной плоскости (homography) для выравнивания изображений, снятых под углом или с наклоном. Данный шаг предобработки называется выравниваем (alignment) на диаграмме компонентов (рис. 2).

Для выравнивания необходимо определить контуры бланка документа на изображении. Для этого была подготовлена матрица проекции 5x5 для черно-белого эквивалента исходного изображения (рис. 3, 2) с помощью метода *cv2.filter2D*. Затем был применен метод Оцу для автоматического определения

[1] Data markup from LabelMe // LabelMe – URL: https://labelme.ru

порогового значения изображения (threshold) с помощью метода *cv2.threshold(filtered, 250, 255, cv2.THRESH_OTSU).* Данный подход позволяет сделать контуры (буквенные очертания) более четкими и различимыми (рис. 3, 3). Затем выбирается наибольший контур с помощью библиотеки OpenCV (рис. 3, 4).

Затем были найдены контуры и углы изображения с помощью алгоритма Рамера–Дугласа–Пекера (рис. 3, 5). По угловым точкам была вычислена матрица гомографии и выполнено проективное преобразование (рис. 3, 6). В результате было получено кадрированное выровненное изображение (рис. 3, 7).

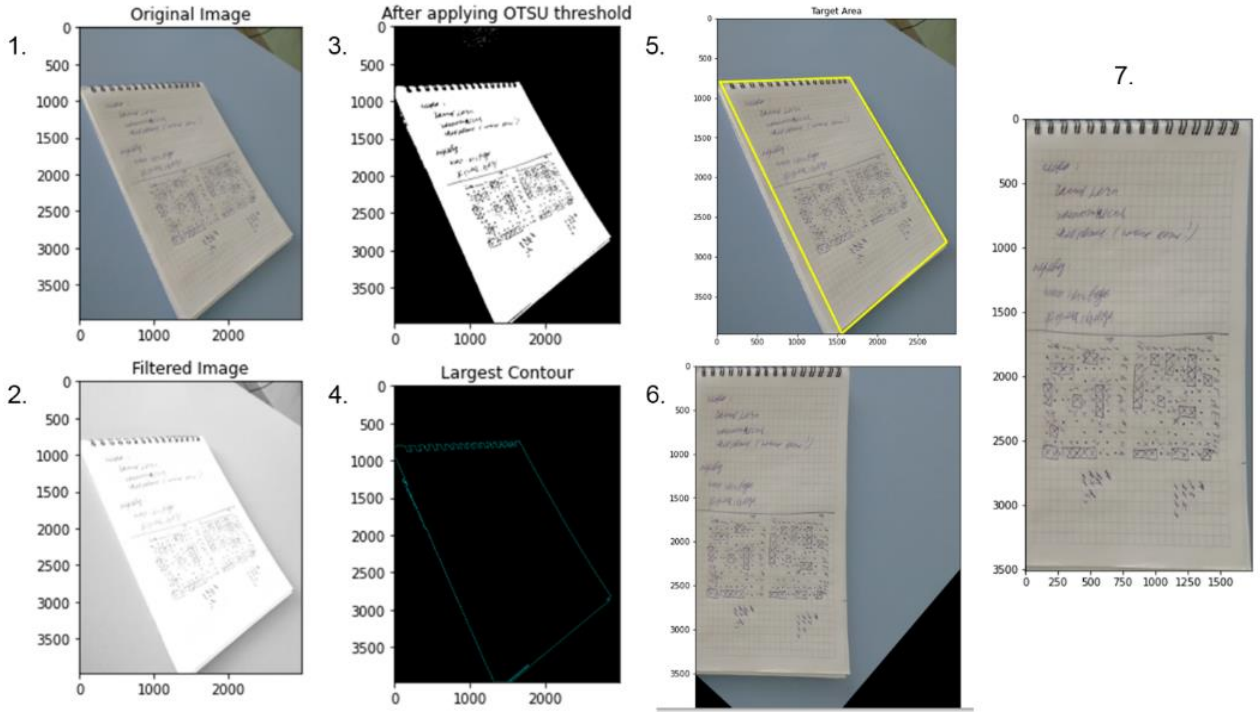


Рис. 3. Выравнивание изображения: 1 – оригинальное изображение документа; 2 – черно-белое изображение бланка; 3 – изображение после применения метода Оцу; 4 – изображение самого крупного контура; 5 – выявление углов с помощью алгоритма Рамера–Дугласа–Пекера; 6 – изображение с выровненной проекцией; 7 – финальное выровненное изображение

4.2. РЕШЕНИЕ, ОСНОВАННОЕ НА НЕЙРОННЫХ СЕТЯХ

4.2.1. ИССЛЕДОВАНИЕ СУЩЕСТВУЮЩИХ ПОДХОДОВ

Был проведен ряд экспериментов с использованием нескольких архитектурных моделей нейронных сетей.

4.2.1.1. CNN

Сначала была использована каноническая модель сверточной нейронной сети (рис. 4, 1). На вход модели был подан следующий набор данных:

- предобработанное методом аугментации изображение из обучающей выборки;
- целочисленный вектор с координатами границ якорной области изображения из обучающей выборки в формате [левый верхний угол, нижний правый угол];
- целочисленный вектор с координатами границ целевой области изображения из обучающей выборки в формате [левый верхний угол, нижний правый угол];
- предобработанное методом аугментации изображение из тестовой выборки;
- целочисленный вектор с координатами границ якорной области изображения из тестовой выборки в формате [левый верхний угол, нижний правый угол].

Выходными данными модели являлись целочисленный вектор с координатами границ целевой области изображения из тестовой выборки в формате [левый верхний угол, нижний правый угол].

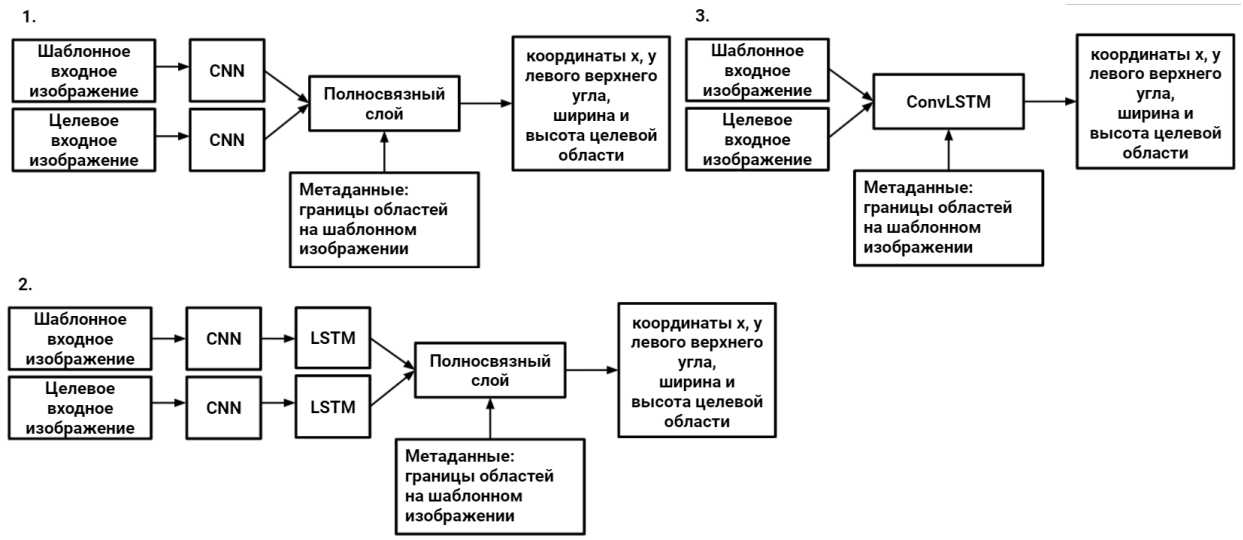


Рис. 4. Варианты архитектур нейронных сетей: 1 – CNN; 2 – CNN + LSTM; 3 – ConvLSTM

Использовалась гибкая настройка нескольких гиперпараметров для обучения нейронных сетей. Optimizer изменяет веса модели (weights) для уменьшения значения функции потерь (loss function value). Коэффициент обучения — это гиперпараметр, который определяет величину коррекции весов на каждой итерации алгоритма. Learning rate schedule – это механизм динамического изменения коэффициента обучения с ростом числа итераций в процессе обучения.

Был проверен ряд гипотез для уменьшения среднеквадратической ошибки (MSE):

- Использование ADAM optimizer. Результат — MSE уменьшилась и стабилизировалась в диапазоне [150000; 200000] без тенденции к дальнейшему уменьшению. Гипотеза принимается.
- Уменьшения коэффициента обучения. Результат — после уменьшения коэффициента до 1e-8 MSE уменьшилась и стабилизировалась в диапазоне [125000; 200000] без тенденции к дальнейшему уменьшению. Гипотеза принимается.
- Увеличение коэффициента обучения. Результат — MSE увеличилась. Гипотеза отклоняется.
- Использование OneCycleLR в качестве learning rate schedule. Результат — MSE уменьшилась и стабилизировалась в диапазоне [125000; 200000]. Гипотеза принимается.
- Обучение одной и той же обученной модели для получения свертки изображений из обучающей и тестовой подвыборок. Результат — MSE не изменилась, но время работы уменьшилось. Гипотеза принимается.
- Смена сверточного слоя на предобученный resnet18 без последнего слоя. Результат — MSE уменьшилась и стабилизировалась в диапазоне [98,000; 180,000] без тенденции к дальнейшему уменьшению. Mini-batch size уменьшен до 8, чтобы избежать превышения порога использования оперативной памяти. Гипотеза принимается.
- Смена сверточного слоя на непредобученный resnet18 без последнего слоя. Результат — MSE не изменилась. Гипотеза отклоняется.

Приемлемого уровня качества (MSE<50000) с использованием модели CNN достичь не удалось. Сверточные нейронные сети позволяют получить высокоуровневое представление признаков изображения, но в процессе теряют низкоуровневые признаки, которые могли бы использоваться в итоговом решении.

4.2.1.2. CNN + LSTM

Затем была реализована более сложная модель CNN + LSTM. Предыдущая модель была улучшена, к ней добавлен рекуррентный слой LSTM (рис. 4, 2). LSTM — это улучшенная модель RNN, которая использует долгую краткосрочную память. Ключевым преимуществом модели является специальная ячейка памяти, в которой сохраняется динамическое состояние системы при выполнении некоторых условий.

В итоге добиться приемлемого уровня качества с вышеупомянутой моделью не удалось. Аккумулирующей емкости модели всё ещё недостаточно, потому что верхнеуровневые представления сверточного слоя слабо различимы и не содержат достаточного количества низкоуровневых признаков, что не позволяет эффективно обучить слой LSTM.

4.2.1.3. ConvLSTM

Далее была реализована модель ConvLSTM. По сравнению с предыдущими моделями вместо двух отдельных слоев CNN и LSTM используется один слой ConvLSTM (рис. 4, 3). Модель ConvLSTM, реализованная в текущей работе, выполняет линейные преобразования над константами из метаинформации и входными эмбеддингами изображений для повышения точности регрессии координат.

В итоге добиться приемлемого уровня качества модели не удалось. Более того, последний подход требует значительно больше памяти по сравнению с предыдущими, что вынуждает использовать ручные вызовы сборщика мусора для очистки ресурсов после каждой итерации. Полученная среднеквадратическая ошибка для регрессии координат имеет большую дисперсию (рис. 5).

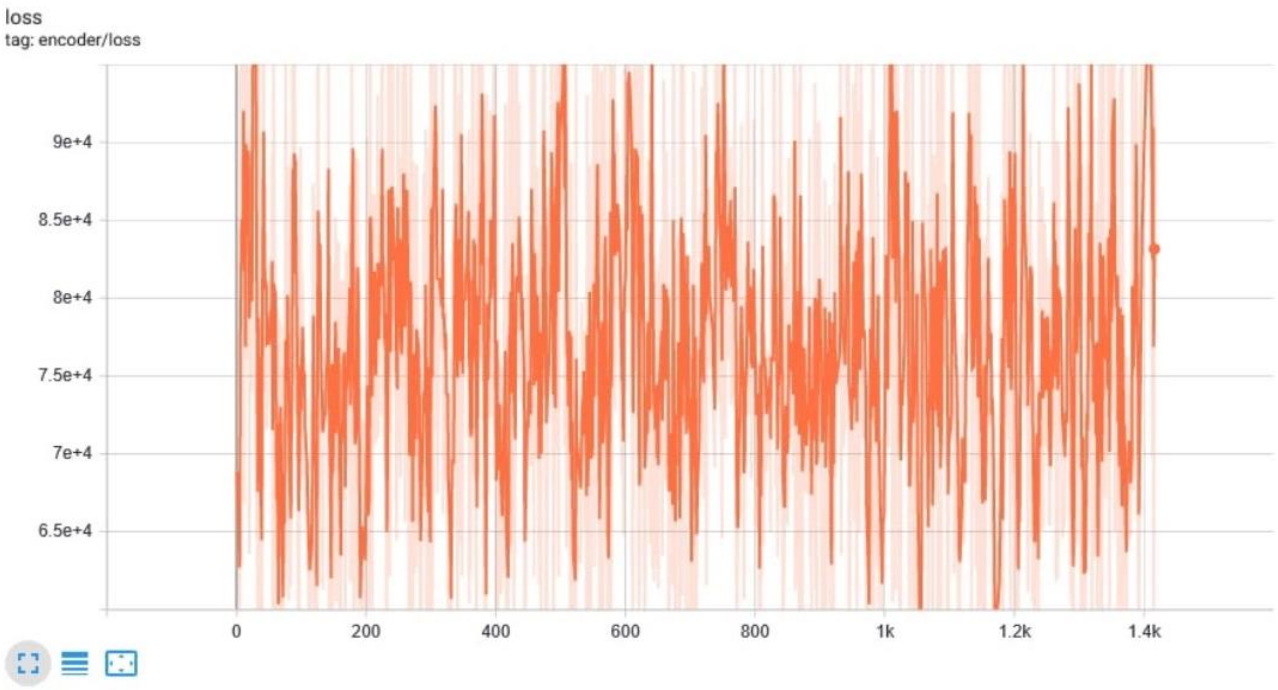


Рис. 5. Показатели работы ConvLSTM: значение функции потерь для каждой итерации; по оси абсцисс отложены итерации, по оси ординат — значения функции потерь.

4.2.2. ПРОМЕЖУТОЧНЫЕ ВЫВОДЫ И ОБСУЖДЕНИЕ

В ходе экспериментов все подходы, основанные на сверточных нейронных сетях, не продемонстрировали сходимость в процессе обучения. Было высказано предположение, что одной из причин неудовлетворительных результатов является основной принцип работы моделей, основанных на CNN. Были предприняты попытки обучить CNN для получения высокоуровневого признакового пространства, чтобы обнаружить похожие области в разных изображениях. Тем не менее, целью нашей работы было обнаружение одинаковых областей текста на разных изображениях. Конкретные текстовые включения — это часть низкоуровневого представления, поэтому они плохо поддаются анализу сверточными нейронными сетями, в результате высокоуровневые представления различных изображений слабо отличаются друг от друга.

4.3. АЛГОРИТМИЧЕСКОЕ РЕШЕНИЕ

В дополнение к подходам, основанным на нейронных сетях, был предложен алгоритмический подход. В данном разделе описаны технические детали поиска необходимых областей на изображении и ряд улучшений для распознавания текста. Эти шаги показаны в действиях 3 и 5 рисунка 1.

Tesseract OCR v4.1.1[2] с помощью метода image_to_data распознает слова с указанием области этого слова на изображении. Затем пользователь выбирает несколько областей со словами для извлечения информации с возможностью группировки нескольких областей в одну. Например, «1 января 2021» может быть представлено как отдельными областями («1», «января», «2021»), так и одной областью («1 января 2021»), но это по-прежнему остается одной смысловой единицей.

Как было обнаружено ранее, LSTM-сети не подошли для решения данной задачи, однако Tesseract основан на LSTM и успешно решает задачи по распознаванию текста с поддержкой 37 языков. Tesseract не решает задачу распознавания выделенных пользователем блоков информации в нескольких документах со схожей структурой, которая является основной в данной статье.

После того, как пользователь выделил целевые области, алгоритм находит якорные области для каждой целевой области по следующему правилу: якорная область – ближайшая область, не пересекающаяся с целевой. Близость определяется на основе евклидового расстояния между центральными точками областей. Пользователь может исправить автоматически определенные якорные области по необходимости. Целевые и якорные области – это метаинформация, которая в дальнейшем сохраняется и используется для обработки похожих изображений. После этого пользователь загружает другие (целевые) изображения похожего формата.

Начинается процесс извлечения информации из целевого изображения. Алгоритм находит сдвиг между целевой и якорной областями на основе метаинформации. Затем с использованием этих данных производится поиск якорных областей на целевом изображении. После этого близость текстов в найденных областях оценивается с помощью расстояния Левенштейна, чтобы компенсировать

[2] Репозиторий tesseract-ocr/tesseract. – URL: https://github.com/tesseract-ocr/tesseract

возможные ошибки в поиске областей. Текстовая область, ближайшая к целевой в шаблонном документе, выбирается в качестве якорной области также и в целевом документе.

После этого алгоритм определяет целевую область на основе якорной области и сдвига, заданных ранее. Алгоритм исключает области без текста, например, с высотой менее 4 пикселей и соотношением более 10 к 1 пикселей. Для улучшения распознавания текста инструментом OCR добавляются дополнительные 5 пикселей вокруг каждой области.

Затем была выполнена оценка результатов алгоритмического решения с использованием метрики Intersection over Union (IoU) [20]. Основная цель алгоритма – определение областей с текстовой информацией на целевом документе.

Для оценки были выбраны 20 образцов одного типа документа (заявление на заселение в студенческое общежитие). В каждом образце выделено 6 групп областей (год обучения, адрес, 2 телефонных номера – рис. 6, дата и полное имя – рис. 7). В группе находятся целевая область и соответствующая ей якорная область.

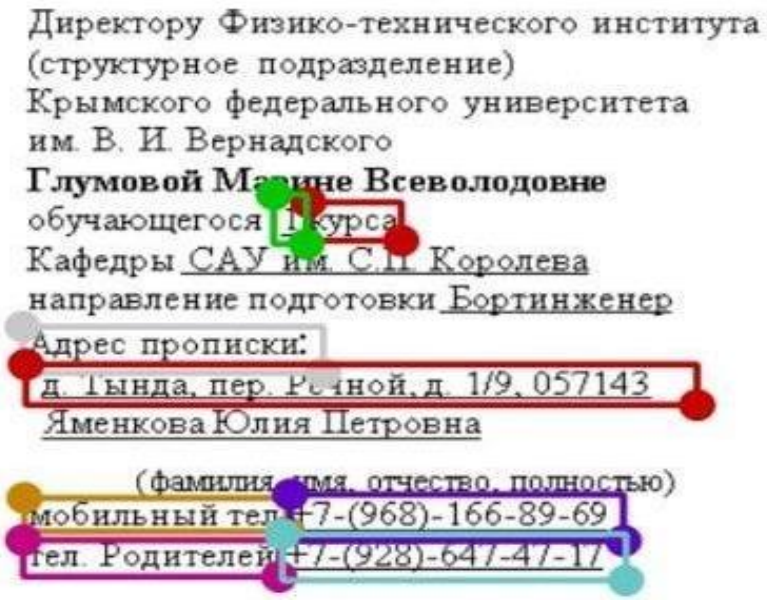


Рис. 6. Пример документа для оценки результатов работы алгоритма. Группы областей: год обучения, адрес, 2 телефонных номера.

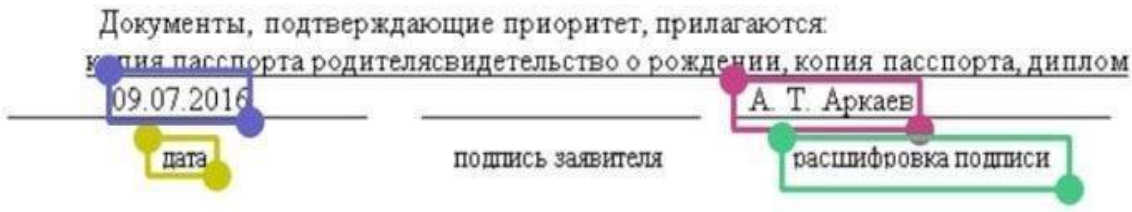


Рис. 7. Пример документа для оценки результатов работы алгоритма. Группы областей: дата и полное имя.

Метрика IoU помогает определить близость области, определенной алгоритмом, к области, которая была задана изначально. Достижение максимального значения IoU не требуется, так как перед распознаванием текста в области ее границы расширяются, как было сказано ранее.

Границы первой области очень малы, поэтому алгоритм обрабатывает ее с точностью ниже 30%. Границы других областей больше, и их результаты выше 30%. На основе анализа гистограммы 30% выбрано как пороговое значение для всех областей. Оно показано красной линией на рис. 8. С учетом вышеупомянутого расширения областей данное значение является удовлетворительным для данного решения. В итоге средняя точность работы алгоритма равна 62.82%.

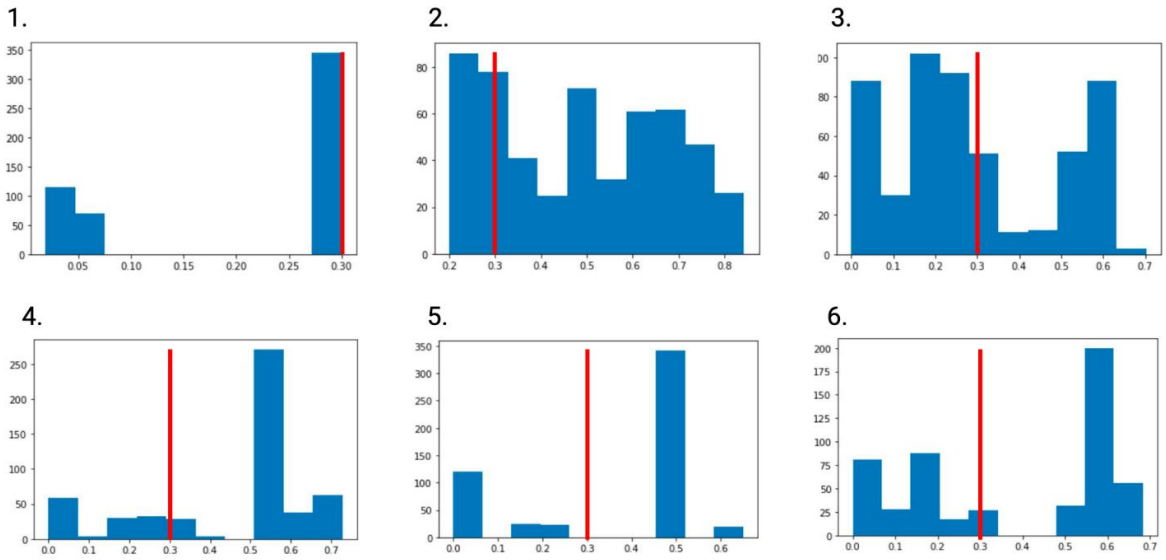


Рис. 8. Распределение метрики IoU для следующих областей: 1. год обучения, 2. адрес, 3. номер телефона 1, 4. номер телефона 2, 5. дата, 6. полное имя. Красной линией показано выбранное пороговое значение для оценки алгоритма.

5. ЗАКЛЮЧЕНИЕ

Настоящая статья представляет различные подходы к извлечению определённой информации из неструктурированных документов. Был создан пайплайн, показывающий шаги представленного решения и потоки данных. Сгенерирован набор данных из 100 машинописных и 100 рукописных заявлений, который был вручную размечен с использованием инструмента LabelMe. Была реализована предобработка данных, которая включает в себя выравнивание с использованием проективного преобразования и аугментацию. Были реализованы различные подходы, базирующиеся на CNN, но все они были отклонены, поскольку модели, реализованные на основе этих подходов, не сходились во время обучения. В результате было предложено алгоритмическое решение. Для оценки эффективности были построены гистограммы распределения метрики IoU для различных предсказанных областей и была достигнута точность 62.82% при пороге (threshold) 30%.

Предложенный алгоритм выделяет данные из различных типов текстовых документов, включая формы, платёжные документы, счета-фактуры и т. д. Эти данные могут быть загружены в базы данных и хранилища данных, чтобы автоматически их обрабатывать в рамках бизнес-процессов или анализировать их для извлечения новой информации, которая может быть использована для улучшения бизнес-процессов.

Для демонстрации работы алгоритма реализовано веб-приложение, исходный код которого доступен на Gitlab: https://gitlab.com/fgast-ai/webapp.

Дальнейшие улучшения предложенного алгоритма могут включать обработку естественного языка, чтобы уменьшить количество неверно предсказанных областей, используя контекст ближайших областей. Предложенный алгоритм также может быть использован как часть алгоритмов бустинга (boosting) машинного обучения, которые могут собирать результаты многих алгоритмов для более точного нахождения целевой области.

СПИСОК ЛИТЕРАТУРЫ

1. Развитие электронного документооборота в России. Статистика, факты, перспективы // Taxcom. URL: https://taxcom.ru/baza-znaniy/ elektronnyy-dokumentooborot/stati/razvitie-elektronnogo-dokumentooborota-v-rossii-statistika-fakty-perspektivy/ (дата обращения 24.02.2021).

2. СЭД (рынок России) // TAdviser.
URL: https://www.tadviser.ru/index.php/Статья:СЭД_(рынок_России) (дата обращения 08.03.2021).

3. AI Unleashes the Power of Unstructured Data // CIO.
URL: https://www.cio.com/article/3406806/ai-unleashes-the-power-of-unstructured-data.html (дата обращения 23.03.2021).

4. Structured vs. Unstructured Data // Datamation. URL: https://www.datamation.com/big-data/structured-vs-unstructured-data/ (дата обращения 23.03.2021).

5. Structured and Unstructured Documents: What are the Differences? // Optiform
URL: https://www.optiform.com/news/structured-unstructured-documents/ (дата обращения 23.03.2021).

6. *McKendrick J.* The Post-Relational Reality Sets in: 2011 Survey on Unstructured Data // Unisphere Research. 2011.

7. *Rusu O. et al.* Converting unstructured and semi-structured data into knowledge // 2013 11th RoEduNet International Conference. IEEE, 2013. P. 1–4.

8. *Mori S., Suen C. Y., Yamamoto K.* Historical review of OCR research and development // Proceedings of the IEEE. 1992. V. 80, No. 7. P. 1029–1058.

9. *Memon J. et al.* Handwritten optical character recognition (OCR): A comprehensive systematic literature review (SLR) // IEEE Access. 2020. V. 8. P. 142642–142668.

10. *Vihar Kurama.* Table Detection, Information Extraction and Structuring using Deep Learning // Nanonets. URL: https://nanonets.com/blog/table-extraction-deep-learning/ (дата обращения 23.02.2021).

11. *Hwang W. et al.* Spatial Dependency Parsing for Semi-Structured Document Information Extraction // arXiv. 2020.

12. *Xu Y. et al.* Layoutlm: Pre-training of text and layout for document image understanding // Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. 2020. P. 1192–1200.

13. *Ye Y. et al.* A unified scheme of text localization and structured data extraction for joint OCR and data mining // 2018 IEEE International Conference on Big Data (Big Data). IEEE. 2018. P. 2373–2382.

14. *Luo S. et al.* Deep Structured Feature Networks for Table Detection and Tabular Data Extraction from Scanned Financial Document Images // arXiv. 2021.

15. *Haase F., Kirchhoff S.* Taxy. io@ FinTOC-2020: Multilingual Document Structure Extraction using Transfer Learning // Proceedings of the 1st Joint Workshop on Financial Narrative Processing and MultiLing Financial Summarisation. 2020. P. 163–168.

16. *Rahman M.M., Finin T.* Unfolding the Structure of a Document using Deep Learning // arXiv. 2019.

17. *Dos Santos J.E.B.* Automatic content extraction on semi-structured documents //2011 International Conference on Document Analysis and Recognition. IEEE. 2011. P. 1235–1239.

18. *Alexander Jung.* Imgaug Documentation Release 0.4.0 // Readthedocs. URL: https://imgaug.readthedocs.io/en/latest/ (дата обращения 02.27.2021).

19. *Visvalingam M., Whyatt J. D.* The Douglas-Peucker algorithm for line simplification: re-evaluation through visualization // Computer Graphics Forum. Oxford, UK: Blackwell Publishing Ltd, 1990. V. 9, No. 3. P. 213–225.

20. Intersection over Union (IoU) for object detection // PyImageSearch. URL: https://www.pyimagesearch.com/2016/11/07/intersection-over-union-iou-for-object-detection/ (дата обращения 27.02.2021).

DATA EXTRACTION FROM SIMILARLY STRUCTURED SCANNED DOCUMENTS

R. D. Saitgareev[1, [0000-0002-8184-6539]], B. R. Giniyatullin[2, [0000-0001-8089-9893]],
V. Y. Toporov[3, [0000-0002-9809-5233]], A. A. Atnagulov[4, [0000-0001-9766-4804]],
F. R. Aglyamov[5, [0000-0002-9939-7989]]

[1-5] *Institute of Information Technology and Intelligent Systems, Kazan Federal University, Kazan;*

[1]srustem3@yandex.ru, [2]bulat.giniiatullin@gmail.com, [3]vladislavtoporov@gmail.com, [4]i@atnartur.ru, [5]aglyamov.fox@gmail.com;

Abstract

Currently, the major part of transmitted and stored data is unstructured, and the amount of unstructured data is growing rapidly each year, although it is hardly searchable, unqueryable, and its processing is not automated. At the same time, there is a growth of electronic document management systems. This paper proposes a solution for extracting data from paper documents considering their structure and layout based on document photos. By examining different approaches, including neural networks and plain algorithmic methods, we present their results and discuss them.

Keywords*: neural networks, machine learning, structure extraction, document structure extraction, OCR, unstructured data, text recognition*

REFERENCES

1. Document flow growth in Russia. Statistics, facts and perspectives // Taxcom URL: https://taxcom.ru/baza-znaniy/elektronnyy-dokumentooborot/stati/razvitie-elektronnogo-dokumentooborota-v-rossii-statistika-fakty-perspektivy/.

2. Electronic document management systems in Russia – market assessments and largest market players // TAdviser. URL: https://www.tadviser.ru/index.php/%D0%A1%D1%82%D0%B0%D1%82%D1%8C%D1%8F:%D0%A1%D0%AD%D0%94_(%D1%80%D1%8B%D0%BD%D0%BE%D0%BA_%D0%A0%D0%BE%D1%81%D1%81%D0%B8%D0%B8).

3. AI Unleashes the Power of Unstructured Data // CIO. URL: https://www.cio.com/article/3406806/ai-unleashes-the-power-of-unstructured-data.html.

4. Structured vs. Unstructured Data // Datamation. URL: https://www.datamation.com/big-data/structured-vs-unstructured-data/

5. Structured and Unstructured Documents: What are the Differences? // Optiform. URL: https://www.optiform.com/news/structured-unstructured-documents/

6. *McKendrick J.* The Post-Relational Reality Sets in: 2011 Survey on Unstructured Data // Unisphere Research. 2011.

7. *Rusu O. et al.* Converting unstructured and semi-structured data into knowledge // 2013 11th RoEduNet International Conference. IEEE. 2013. P. 1–4.

8. *Mori S., Suen C.Y., Yamamoto K.* Historical review of OCR research and development // Proceedings of the IEEE. 1992. V. 80, No. 7. P. 1029–1058.

9. *Memon J. et al.* Handwritten optical character recognition (OCR): A comprehensive systematic literature review (SLR) // IEEE Access. 2020. V. 8. P. 142642–142668.

10. *Vihar Kurama.* Table Detection, Information Extraction and Structuring using Deep Learning // Nanonets. URL: https://nanonets.com/blog/table-extraction-deep-learning/

11. *Hwang W. et al.* Spatial Dependency Parsing for Semi-Structured Document Information Extraction // arXiv. 2020.

12. *Xu Y. et al.* Layoutlm: Pre-training of text and layout for document image understanding // Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. 2020. P. 1192–1200.

13. *Ye Y. et al.* A unified scheme of text localization and structured data extraction for joint OCR and data mining // 2018 IEEE International Conference on Big Data (Big Data). IEEE. 2018. P. 2373–2382.

14. *Luo S. et al.* Deep Structured Feature Networks for Table Detection and Tabular Data Extraction from Scanned Financial Document Images // arXiv. 2021.

15. *Haase F., Kirchhoff S.* Taxy. io@ FinTOC-2020: Multilingual Document Structure Extraction using Transfer Learning // Proceedings of the 1st Joint Workshop

on Financial Narrative Processing and MultiLing Financial Summarisation. 2020. P. 163–168.

16. *Rahman M.M., Finin T.* Unfolding the Structure of a Document using Deep Learning // arXiv. 2019.

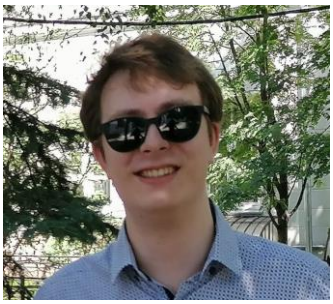
17. *Dos Santos J.E.B.* Automatic content extraction on semi-structured documents //2011 International Conference on Document Analysis and Recognition. IEEE. 2011. P. 1235–1239.

18. *Alexander Jung.* Imgaug Documentation Release 0.4.0 // Readthedocs. URL: https://imgaug.readthedocs.io/en/latest/.

19. *Visvalingam M., Whyatt J.D.* The Douglas-Peucker algorithm for line simplification: re-evaluation through visualization // Computer Graphics Forum. Oxford, UK: Blackwell Publishing Ltd, 1990. V. 9, No. 3. P. 213–225.

20. Intersection over Union (IoU) for object detection // PyImageSearch. URL: https://www.pyimagesearch.com/2016/11/07/intersection-over-union-iou-for-object-detection/.

СВЕДЕНИЯ ОБ АВТОРАХ



САИТГАРЕЕВ Рустем Дамирович – студент магистратуры кафедры программной инженерии Института информационных технологий и интеллектуальных систем, Казанский федеральный университет.

Rustem Damirovich SAITGAREEV – Master's student at the Department of Software Engineering, Institute of Information Technology and Intelligent Systems, Kazan Federal University.

email: srustem3@yandex.ru

ГИНИЯТУЛЛИН Булат Рифатович – студент магистратуры кафедры программной инженерии Института информационных технологий и интеллектуальных систем, Казанский федеральный университет.

***Bulat Rifatovich* GINIYATULLIN** – Master's student at the Department of Software Engineering, Institute of Information Technology and Intelligent Systems, Kazan Federal University.

email: bulat.giniiatullin@gmail.com

ТОПОРОВ Владислав Юрьевич – студент магистратуры кафедры программной инженерии Института информационных технологий и интеллектуальных систем, Казанский федеральный университет.

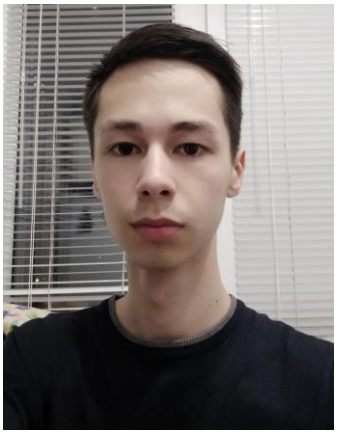
Vladislav Yurievich TOPOROV – Master's student at the Department of Software Engineering, Institute of Information Technology and Intelligent Systems, Kazan Federal University.

email: vladislavtoporov@gmail.com

АТНАГУЛОВ Артур Александрович – студент магистратуры кафедры программной инженерии Института информационных технологий и интеллектуальных систем, Казанский федеральный университет.

***Artur Aleksandrovich* ATNAGULOV** – Master's student at the Department of Software Engineering, Institute of Information Technology and Intelligent Systems, Kazan Federal University.

email: i@atnartur.ru



АГЛЯМОВ *Фарид Радикович* – студент магистратуры кафедры программной инженерии Института информационных технологий и интеллектуальных систем, Казанский федеральный университет.

***Farid Radikovich* AGLYAMOV** – Master's student at the Department of Software Engineering, Institute of Information Technology and Intelligent Systems, Kazan Federal University.

email: aglyamov.fox@gmail.com

Материал поступил в редакцию 20 июня 2021 года