С. А. Коровин, К. В. Кочеткова, В. А. Бородина, Д. И. Волобуев
Тюменский государственный университет, г. Тюмень

УДК 004.9

АВТОМАТИЗАЦИЯ ПРОВЕРКИ ДОСТИЖЕНИЙ СТУДЕНТОВ ПРИ ПОДАЧЕ ЗАЯВЛЕНИЙ НА ПОВЫШЕННУЮ ГОСУДАРСТВЕННУЮ АКАДЕМИЧЕСКУЮ СТИПЕНДИЮ

Аннотация. В статье рассмотрен подход к автоматизации проверки достижений студентов, при подаче заявлений на конкурс ПГАС. Сформирован набор данных из изображений различных форм поощрительных документов. Разработан алгоритм для распознавания ключевых признаков на них. Проведено тестирование используемой модели с целью выявления оптимальных параметров.

Ключевые слова: OCR, распознавание символов, сегментация текста, Tesseract, анализ естественного языка, изображения.

Введение. Согласно приказу Министерства образования и науки Российской Федерации от 27.12.2016 №1663 [1] в подчиненных министерству вузах выплачиваются государственный академические стипендии в повышенном размере (ПГАС) за достижения студентов в следующих областях деятельности: учебная, научно-исследовательская, общественная, культурно-творческая и спортивная. В Тюменском государственном университете по итогам каждого семестра проводится конкурс на получения ПГАС, для участия в котором студентам необходимо заполнить заявление, прикрепить подтверждающие деятельность документы и отправить на специальный адрес почты. В данной статье акцент сделан на заявления только одного из видов ПГАС — за общественную деятельность.

После окончания этапа приема заявлений каждая заявка и прикрепленные к ней файлы с достижениями (в случае с общественной деятельностью это благодарности, благодарственные письма и сертификаты) просматриваются вручную с целью проверки. Такой процесс занимает достаточно продолжительное время, затраты которого можно было бы уменьшить путем создания какого-либо цифрового решения, например, единой информационной системы учета мероприятий, описанной в [2], или системы учета достижений, проектируемой в работе [3], или сервиса для подачи заявлений на

ПГАС и добавления в подобные сервисы возможности автоматической проверки и подтверждения достижений, в том числе реализованной с помощью применения алгоритмов компьютерного зрения.

Существуют различные способы упрощения проверки достижений, например, поиск логотипа вуза, что позволит моментально отсеять достижения, выданые за пределами университета для верификации их вручную, или возможность подтверждать достижения за конкретные мероприятия единоразово путем группировки похожих изображений, или распознавание на достижении ключевой ин-формации (ФИО получателя, основания и даты выдачи, подписанта, а также регистрационного номера при наличии) для дальнейшей проверки подлинности или сверки с единым реестром таких документов.

В рамках данной работы был рассмотрен последний из способов, который сводится к задаче распознавания текста на изображении с последующей его сегментацией. Отсюда цель работы: собрать датасет прикрепляемых к заявлению на ПГАС документов, подтверждающих общественную деятельность студентов, и разработать алгоритм для распознавания ключевых элементов на таких изображениях.

Для перевода изображений в текстовые данные используются системы оптического распознавания символов (OCR). В результате сравнительного анализа [4] некоторых систем OCR, работающих с кириллическими символами, было установлено, что лучшей по точности распознавания кириллических символов и скорости работы среди не проприетарных систем OCR является Tesseract.

Одна из прикладных задач, решаемых с помощью технологий распознавания текста на изображениях — распознавание текстовых печатных полей сканированного документа. В статье [5] описана разработка сервиса для загрузки изображений документов и распознавание текста на вырезанных интересующих областях с использованием библиотеки Tesseract-OCR, для работы с которой на языке программирования Python предусмотрена библиотека pytesseract, применяемая в работе [6], в которой рассматривается разработка программного модуля для автоматического распознавания текста в конкретно заданных областях изображения, имеющих цвет, отличный от остальных областей изображения.

Задача распознавания текста на изображении с последующей его сегментацией решалась в работе [7], в которой описан алгоритм проверки полноты информации на этикетке путем распознавания текста на изображении и извлечения необходимых данных с помощью заданных шаблонов. Для решения задачи распознавания текста также применялась библиотека Tesseract-OCR.

Эти и другие подобные работы показывают эффективность применения библиотеки Tesseract для распознавания текста на изображениях при решении различных прикладных задач.

Материалы и методы. В рамках работы был собран датасет прикрепляемых студентами в заявлениях на ПГАС за общественную деятельность достижений (рис. 1), включающий в себя 400 изображений благодарностей, благодарственных писем и сертификатов, выданных по разным основаниям, причем некоторые присутствуют только в единственном экземпляре, а другие — во множественном числе. Каждое достижение — это файл в формате jpeg или png различной ориентации.

В рамках данной работы на изображениях достижений выделяются такие ключевые элементы, как: ФИО получателя, основание и дата выдачи, регистрационный номер при наличии (рис. 2).

Рис. 1. Фрагмент собранного датасета

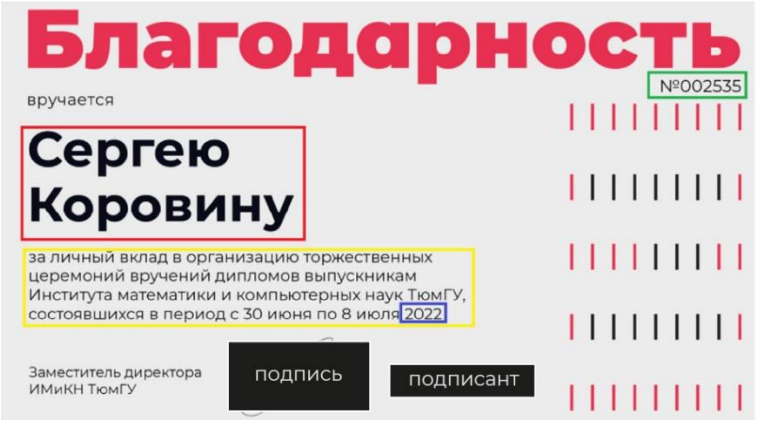

Рис. 2. Ключевые элементы достижения на примере

Для работы с изображениями используется библиотека Pillow, а для распознавания текста PyTesseract [8]. Для выявления оптимальных параметров модели было проведено тестирование, для которого вручную были размечены 38 изображений достижений с различными стилями, шрифтовой композицией и текстами.

Из библиотеки PyTesseract [8] были рассмотрены два метода получения распознанного текста: image_to_data — возвращает таблицу, строки которой содержат распознанное слово и дополнительную информацию о нем (ширина, высота, расположение, номер текстового блока и др.) и image_to_string — возвращает весь распознанный текст в виде строки. Для распознавания основания выдачи достижения был выбран первый метод, а для всей остальной информации — второй.

Результаты. В результате тестирования режимов сегментации, подходящих для рассматриваемых данных, с целью распознавания получателя достижения лучшим оказался режим сегментации № 11 (табл. 1), при котором текст воспринимается как разреженный, и поиск информации осуществляется в произвольном порядке. Критериями оценивания выступили: процент корректно найденных получателей достижения, процент совпадения найденного имени с ФИО отправителя документа, время работы.

Таблица 1

Сравнение режимов сегментации получателя документа

	Найдено	*Соответствие*	*Время*
Psm1	79%	89%	51 сек.
Psm3	79%	89%	38 сек.
Psm4	81%	89%	38 сек.
Psm6	55%	89%	34 сек.
Psm11	**84%**	**90%**	**39 сек.**
Psm12	84%	90%	53 сек.

Для поиска ФИО в полученном текстовом наборе сначала выделяются все персоны с помощью библиотеки Natasha [9], далее каждый найденный человек проверяется на схожесть с ФИО того, кто подал документ (рис. 3). Схожесть определяется путем нечеткого сравнения строк с помощью библиотеки FuzzyWuzzy [10], реализующей алгоритм Дамерау-Левенштейна [11].

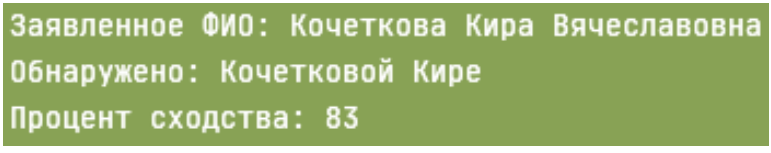

Рис. 3. Пример проверки схожести ФИО

Схожесть распознанного ФИО и ФИО отправителя достижения вычисляется как средняя схожесть по трем словам: фамилия, имя и отчество при наличии. Такой подход позволяет учесть тот факт, что ФИО может быть в разных падежах. Достижение считается принадлежащим отправителю, если на нем было найдено ФИО с процентом схожести выше 70% (значение было подобрано опытным путем).

Тестирование режимов сегментации для распознавания регистрационного номера достижения показало преимущества режима сегментации № 6 (табл. 2), при котором текст воспринимается как единый унифицированный блок, так как процент корректно распознанных номеров более значим, по сравнению с процентом найденных.

Критериями качества распознавания выступили: процент распознанных регистрационных номеров, число ошибок при распознании и время работы.

Регистрационный номер обязательно начинается с символа «#» или «№» и состоит из 6 цифр, что было выявлено в ходе анализа собранного датасета достижений, поэтому сначала отбираются слова, удовлетворяющие этим критериям. В случае наличия 7 цифр — обрезается первая, и номер считается распознанным, так как в результате тестирования было выявлено, что модель ошибочно принимает часть символа «№» за цифры.

Таблица 2

Сравнение режимов сегментации регистрационного номера

	Найдено	*Верно*	*Ошибка = 1*	*Ошибка > 1*	*Время*
Psm1	45%	100%	0	0	49 сек.
Psm3	45%	100%	0	0	37 сек.
Psm4	42%	100%	0	0	37 сек.
Psm6	**55%**	**95%**	**5%**	**0**	**33 сек.**
Psm11	60%	87%	4%	9%	38 сек.
Psm12	60%	87%	4%	9%	52 сек.

По результатам тестирования с целью распознавания основания выдачи достижения модель показала положительные результаты для решения данной задачи, наиболее предпочтительный вариант — режим № 3 (табл. 3). Кроме того, для распознавания был добавлен английский язык, так как некоторые мероприятия имеют иностранные названия. Качество распознавания оценивалось с помощью вычисления процента найденных описаний и соответствия оригиналу.

Для получения результата выполняется группировка слов, отнесенных моделью к одному текстовому блоку, в одно предложение, и среди всех блоков находится тот, который начинается со слова «за».

Для решения задачи распознавания даты выдачи достижения лучшим режимом сегментации оказался № 6 (табл. 4) по следующим критериям: процент найденных дат и проценты дат, определенных полностью, либо частично.

Таблица 3

Сравнение режимов сегментации основания выдачи

	Найдено	*Соответствие*	*Время*
Psm1	95%	99,5%	49 сек.
Psm3	95%	99,5%	37 сек.
Psm4	95%	99,3%	37 сек.
Psm11	95%	90%	38 сек.
Psm12	95%	90%	52 сек.
Psm3 + eng	**95%**	**99,7%**	**45 сек.**

Таблица 4

Сравнение режимов сегментации даты выдачи

	Найдено	*Верно*	*Частично*	*Неверно*	*Время*
Psm1	87%	97%	3%	0	49 сек.
Psm3	87%	97%	3%	0	37 сек.
Psm4	87%	97%	3%	0	36 сек.
Psm6	**95%**	**100%**	**0**	**0**	**34 сек.**
Psm11	87%	100%	0	0	38 сек.
Psm12	87%	100%	0	0	51 сек.

Распознанный текст разбивается на слова, а годом считается слово, соответствующее регулярному выражению «20\d{2}». Затем из полученного списка годов исключаются дубликаты и те, которые отличаются от текущего на 6 лет, так как это некорректное распознавание. Полученные результаты приводятся к виду, в котором они встречаются в распознаваемом документе.

Заключение. В рамках данной работы был реализован алгоритм распознавания и сегментации текста на подтверждающих общественную деятельность студентов документах, отправляемых в заявлениях на ПГАС в виде изображений, с использованием готовой модели от Google [12] и библиотеки PyTesseract [8] для работы с ней, а также проведено исследование с целью выявления оптимальных параметров выбранной модели для нахождения ключевых элементов достижений, таких как: ФИО получателя, основание и дата выдачи, регистрационный номер при наличии.

В дальнейших работах будут рассмотрены способы улучшения разработанного алгоритма, путем изменения параметров модели или применения более сложных моделей и технологий для решения данной задачи, реализованы и протестированы иные алгоритмы не только для проверки достижений за общественную деятельность, но и за иные виды деятельности [1], в том числе с целью создания единой платформы подачи заявлений на ПГАС и внедрения таких алгоритмов для упрощения проверки заявлений и достижений, отправляемых во время конкурса на ПГАС.

СПИСОК ЛИТЕРАТУРЫ

1. Приказ Министерства образования и науки Российской Федерации от 27.12.2016 №1663 «Об утверждении Порядка назначения государственной академической стипендии и (или) государственной социальной стипендии студентам, обучающимся по очной форме обучения за счет бюджетных ассигнований федерального бюджета, государственной стипендии аспирантам, ординаторам, ассистентам-стажерам, обучающимся по очной форме обучения за счет бюджетных ассигнований федерального бюджета, выплаты стипендий слушателям подготовительных отделений федеральных государственных образовательных организаций высшего образования, обучающимся за счет бюджетных ассигнований федерального бюджета» : принят Министром 27 декабря 2016 года. — Текст: непосредственный.
2. Покидько Е. В. Информационная система учета мероприятий: выпускная квалификационная работа (бакалаврская работа) студента 4 курса очной формы обучения по направлению подготовки 09.03.03 Прикладная информатика, профиль "Разработка информационных систем бизнеса" / Е. В. Покидько; науч. рук. Ю. Е. Карякин. — Тюмень, 2022. — Текст: электронный. URL:https://library.utmn.ru/dl/VKR_Tyumen/VKR_2022/IMiKN/PokidkoEV_2022.pdf.
3. Польская П. С. Анализ систем учета достижений студентов / П. С. Польская, К. Е. Огнегин, Е. М. Маркушин [и др.]. — Текст: непосредственный // Modern Science. — 2020. — № 4-1. — С. 378-382.
4. Качалин В. С. Сравнительный анализ различных систем оптического распознавания символов при работе с текстом, написанным с помощью кириллического алфавита / В. С. Качалин, Ю. Н. Панов, Н. Л. Э. Попов. — Текст: непосредственный // Современные наукоемкие технологии. — 2022. — № 8. — С. 65-70.

5. Минязев Р. Ш. Разработка сервиса для идентификации полей сканированного документа с использованием библиотеки машинного распознавания Tesseract-OCR / Р. Ш. Минязев, С. А. Дыганов, И. Р. Гумеров, М. Ю. Перухин. — Текст: непосредственный // 2018. — Т. 21, № 9. — С. 132-135.
6. Бабаев А. М. Автоматизация извлечения текста из изображения посредством оптического распознавания символов / А. М. Бабаев, Ю. В. Алексеенко, А. М. Бабаев, Т. Г. Авдеева. — Текст: непосредственный // Особенности современного этапа развития естественных и технических наук : Сборник научных трудов по материалам Международной научно-практической конференции. в 2-х частях, Белгород, 28 декабря 2017 года / Под общей редакцией Е.П. Ткачевой. Том Часть II. — Белгород: Общество с ограниченной ответственностью «Агентство перспективных научных исследований», 2018. — С. 31–36.
7. Перунова, А. В. Разработка сервиса для проверки полноты информации на этикетках пищевых продуктов / А. В. Перунова, А. В. Глазкова — Текст: непосредственный // Математическое и информационное моделирование : Материалы Всероссийской конференции молодых ученых, Тюмень, 01-08 июня 2020 года. Том Выпуск 18. — Тюмень: Тюменский государственный университет, 2020. — С. 277-287.
8. Официальный репозиторий библиотеки PyTesseract [Электронный ресурс]: GitHub, Inc., 2023. URL:https://github.com/madmaze/pytesseract (дата обращения: 10.04.2023).
9. Официальный репозиторий проекта Natasha — набор Python-библиотек для обработки текстов на естественном русском языке [Электронный ресурс]: GitHub, Inc., 2023. URL:https://github.com/natasha/natasha (дата обращения: 17.04.2023).
10. Официальный репозиторий проекта FuzzyWuzzy — библиотека на Python для нечеткого сравнения строк [Электронный ресурс]: GitHub, Inc., 2023. URL:https://github.com/seatgeek/fuzzywuzzy (дата обращения: 23.04.2023).
11. Damerau F. A. Technique for Computer Detection and Correction of Spelling Errors // Communications of the ACM. 1964. Vol. 7. No. 3. P. 171-176. — Direct text.
12. Главный репозиторий проекта Google Tesseract-OCR Engine [Электронный ресурс]: GitHub, Inc., 2023. URL:https://github.com/tesseract-ocr (дата обращения: 30.03.2023).