

РАСПОЗНАВАНИЕ РУКОПИСНОГО ТЕКСТА ДЛЯ АВТОМАТИЗАЦИИ СЕРВИСА ВАКЦИНАЦИИ

Кондратьева Надежда Геннадьевна¹, Булякова Ирина Александровна²

¹Студент;

Государственный университет «Дубна»;

141980, Россия, Московская обл., г. Дубна, ул. Университетская, 19;

e-mail: kng.sdo20@uni-dubna.ru.

²Старший преподаватель;

Государственный университет «Дубна»;

141980, Россия, Московская обл., г. Дубна, ул. Университетская, 19;

e-mail: buljakova@mail.ru.

В рамках статьи рассмотрены текущие проблемы в области обработки медицинской документации, проведен обзор существующих технологий распознавания рукописного текста и предложено собственное техническое решение, способное эффективно интегрироваться в практику.

Ключевые слова: вакцинация, анализ данных, компьютерное зрение, распознавание рукописного текста.

Для цитирования:

Кондратьева Н. Г., Булякова И. А. Распознавание рукописного текста для автоматизации сервиса вакцинации // Системный анализ в науке и образовании. 2026. № 1. С. 1–11. EDN: TZKNTS. URL: <https://sanse.ru/index.php/sanse/article/view/699>.

HANDWRITTEN TEXT RECOGNITION FOR AUTOMATING VACCINATION SERVICE

Kondrateva Nadezhda G.¹, Bulyakova Irina A.²

¹Student;

Dubna State University;

19 Universitetskaya Str., Dubna, Moscow region, 141980, Russia;

e-mail: kng.sdo20@uni-dubna.ru.

²Senior teacher;

Dubna State University;

19 Universitetskaya Str., Dubna, Moscow region, 141980, Russia;

e-mail: buljakova@mail.ru.

This article examines current issues in the field of medical documentation processing, provides an overview of existing handwriting recognition technologies, and proposes a proprietary technical solution that can be effectively integrated into practice.

Keywords: vaccination, data analysis, computer vision, handwritten text recognition.

For citation:

Kondrateva N. G., Bulyakova I. A. Handwritten text recognition for automating vaccination service. *System analysis in science and education*, 2026;(1):1-11(in Russ). EDN: TZKNTS. Available from: <https://sanse.ru/index.php/sanse/article/view/699>.



Статья находится в открытом доступе и распространяется в соответствии с лицензией Creative Commons «Attribution» («Атрибуция») 4.0 Всемирная (CC BY 4.0) <https://creativecommons.org/licenses/by/4.0/deed.ru>

Введение

В настоящее время, в условиях стремительного развития информационных технологий, вопросы автоматизации и оптимизации процессов в различных областях жизни становятся все более актуальными. Одной из перспективных областей, где внедрение современных технологий может привести к значительному улучшению качества обслуживания и повышению эффективности, является медицинская сфера. Регистрация и обработка большого объема медицинской документации по-прежнему требует значительных временных и человеческих ресурсов, особенно когда речь идет о странах с ограничениями к доступу к сети интернет. В связи с этим возникает необходимость внедрения инновационных решений, способных ускорить и упростить процессы обработки информации. Одним из перспективных направлений в этом контексте является разработка системы распознавания рукописного текста, специально адаптированной для потребностей медицинских учреждений.

Данная статья посвящена разработке модуля системы распознавания рукописного текста с целью автоматизации сервиса вакцинации.

1. Особенности автоматизации сервиса вакцинации

В свете пандемии *COVID-19* вопрос вакцинации приобрел невероятную актуальность. Опыт последних лет показал, что она играет ключевую роль в предотвращении распространения инфекций и создании коллективного иммунитета. Но помимо *COVID-19*, вакцинация продолжает быть критически важной и для предотвращения других заболеваний, таких как грипп, корь, дифтерия и т. д. Учитывая актуальность программ иммунизации населения, существует потребность в разработке информационных технологий, которые позволят автоматизировать процессы в данной сфере медицинских услуг [2].

Одной из ключевых проблем в развивающихся странах является ограниченный доступ к сети интернет. Поскольку эти регионы не сильно урбанизированы, большая часть населения проживает в деревнях. Реализация вакцинации осуществляется мобильными медицинскими бригадами, которые выезжают в отдаленные места, жители которых иначе могли бы остаться не привитыми из-за трудностей доступа к медицинским учреждениям. Важная информация о вакцинированных лицах, такая как личные данные и вид вакцины, регистрируется медицинскими работниками вручную на бумаге на специализированных бланках и требует последующего внесения в электронные базы данных. Таким образом, огромный объем информации и на сегодняшний день регистрируется на бумажных носителях без возможности перехода в электронный формат в ближайшее время. Это создает значительные трудности при внедрении цифровых решений для организации вакцинации и требует поиска альтернативных методов.

Другой важной особенностью программ вакцинации в развивающихся странах является их финансирование за счет доноров, таких как международные организации и благотворительные фонды. Это финансирование, направленное на приобретение вакцин, медицинского оборудования, автоматизацию процессов и организацию вакцинационных кампаний, часто предполагает строгий контроль за использованием средств и требует детального отчета о затратах. Необходимость максимальной прозрачности в использовании ресурсов при сохранении традиционного подхода (хранение большого объема информации на бумаге) требует разработки инновационных решений, способных эффективно интегрировать цифровые технологии для надежного учета данных, обеспечивая в то же время полноту и достоверность информации.

Целью данной работы является автоматизация процессов внесения, хранения и анализа информации о вакцинации клиентов: оптимизация обработки оцифрованных бланков.

2. Существующие информационные технологии обработки оцифрованных бланков

Задача распознавания текста на бланках и формах является актуальной не только в рамках автоматизации сервиса вакцинации, но и в рамках огромного количества других областей и

организаций: логистика, промышленность, банки, индивидуальное предпринимательство, налоговые и муниципальные органы, юридические службы и суды, образование, правоохранительные органы и, конечно же, здравоохранение. На сегодняшний день существует несколько крупных программных продуктов, которые нацелены на решение задачи получения текстовой информации из оцифрованных бланков. Рассмотрим три из них: библиотеку *Tesseract*, продукты *ABBYY FineReader* и *Google Cloud Vision* [1].

Google Cloud Vision представляет собой облачный сервис, специализирующийся на распознавании текста на изображениях [6]. Доступ к нему можно получить только на платной основе. Основным преимуществом данного продукта по сравнению с двумя другими является его более высокие показатели точности при распознавании рукописного текста.

ABBYY FineReader является разработкой российской компании *ABBYY*, которая уже долгие годы занимается задачей оптического распознавания текстов на бланках [3]. Основным преимуществом является хорошее извлечение информации из таблиц, при этом не только из отсканированных документов, но также и со снятых на камеру. Одним из главных недостатков является цена продукта. Кроме того, ПО не нацелено на работу с рукописным текстом, поэтому *ABBYY FineReader OCR* уступает по данному показателю *Google Cloud Vision*.

В отличие от описанных выше двух продуктов, библиотека *Tesseract* является программным обеспечением, распространяющимся на бесплатной основе [22]. Достаточно хорошо распознает отсканированный текст, но при задаче распознавания рукописного текста возникает заметное количество ошибок. Для качественной обработки табличной информации требуется использование сторонних библиотек.

Ни одно из существующих решений не подходило для решения поставленной задачи, в связи с чем возникла необходимость анализа литературы и разработки собственного модуля информационной системы, распознающего рукописный текст с оцифрованных бланков.

3. Алгоритмы оптического распознавания текста и сверточные нейронные сети

Задача распознавания текста на бланках относится к группе задач оптического распознавания символов из области компьютерного зрения. Оптическое распознавание символов (*optical character recognition, OCR*) – одно из направлений распознавания образов, задача которого заключается в переводе изображений рукописного, машинного или печатного текста в текстовые данные, используемые для представления символов в компьютере, например, в текстовом редакторе [4].

За последнее десятилетие в области машинного обучения, специализирующаяся на задачах компьютерного зрения, был совершен грандиозный прорыв: были созданы трансформеры, генеративно-состязательные сети и другие технологии, которые позволяют решать задачи генерации, восстановления, повышения размерности, детекции, смены стилей и многие другие [11]. Обзор данных алгоритмов выходит за пределы нашей работы, так как для решения задачи оптического распознавания рукописного текста с бланков достаточно использования сверточных нейронных сетей.

Сверточные нейронные сети (*Convolutional Neural Networks, CNN*) представляют собой класс глубоких нейронных сетей, которые особенно эффективны в обработке визуальных данных благодаря их способности автоматически выделять важные признаки [10]. На сегодняшний день существует множество предобученных моделей для распознавания текста, основанных на сверточных нейронных сетях. Эти модели, обученные на больших наборах данных, могут быть легко адаптированы для решения специфических задач распознавания текста на бланках. Среди них можно выделить такие модели, как *EasyOCR* и *CRNN (Convolutional Recurrent Neural Network)*. Эти модели предоставляют высокую точность и эффективность, позволяя разработчикам интегрировать *OCR* функционал в свои приложения без необходимости обучения модели с нуля.

Главная цель данной работы заключается в доработке и повышении точности системы распознавания существующей ИС и для этого мы использовали технологию дообучения (*transfer learning*). Процедура дообучения позволяет взять готовую модель, обученную на большом объеме данных, и затем использовать накопленный ею “опыт” для дообучения на целевом и сравнительно

небольшом объеме данных. Этот подход, таким образом, позволяет не только значительно облегчить задачу подготовки датасета (обучение сверточной нейронной сети с нуля требует огромного количества данных, а дообучение – нет), но и является оптимальным для решения задачи оптического распознавания отсканированного текста с точки зрения оптимизации вычислительных, временных и финансовых затрат.

4. Описание задачи

Задача была реализована на базе международной технической компании, разрабатывающая ПО для систем здравоохранения развивающихся стран для решения различных задач, в том числе цифровизации сервиса вакцинации.

У заказчика имелся бланк вакцинации *Child Vaccination* версии V07 (рис. 1). Основная проблема при распознавании возникала с областью *Client SPT ID*, где располагаются 7–9 значные идентификационные номера клиентов. Важно добавить, что в рамках реализации ПО для решения задачи распознавания и извлечения текстовой информации с отсканированных бланков компания-заказчик несколько лет использовала продукт *ABBYY FineReader OCR*. Но недостаточно высокий уровень распознавания рукописного текста и очень высокая стоимость продукта в итоге привела компанию к решению самостоятельно разработать необходимый сервис. Данный сервис был разработан, но, к сожалению, процент ошибок распознавания увеличился и повысил затраты на исправление ошибок в извлеченной информации. Количество ошибочно распознанных номеров в данной системе превышало 20%.

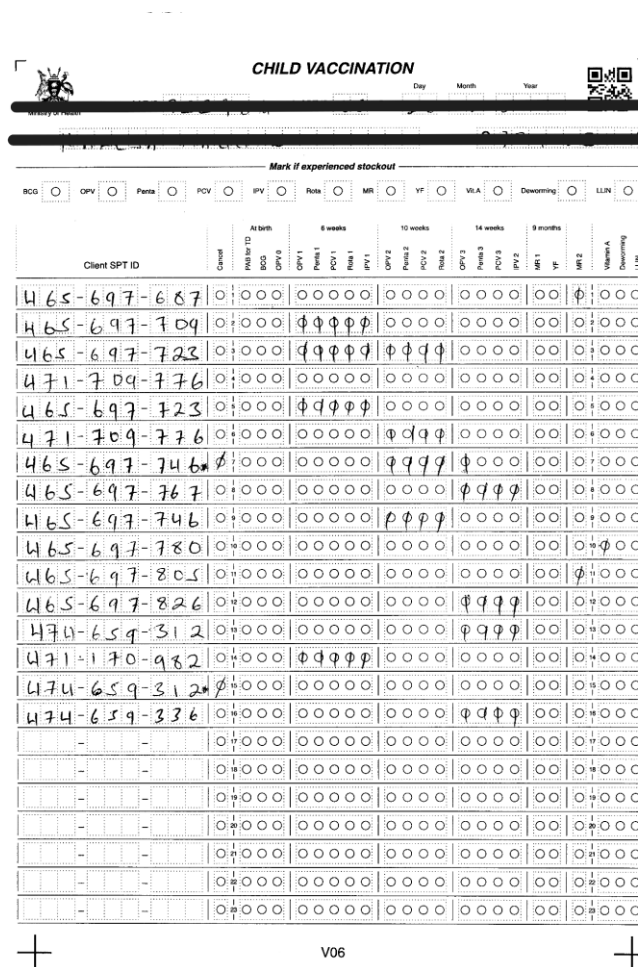


Рис. 1. Бланк *Child Vaccination* версии V07

Основной запрос со стороны заказчика – разработать такую систему распознавания *ID* из области *Client SPT ID* на бланке *Child Vaccination* версии V07, которая снизит количество ошибочно

распознанных идентификационных номеров клиентов до 5%. Таким образом, перед нами стояла задача повышение точности распознавания рукописных цифр на отсканированном бланке в формате *.pdf*, решение которой описано в данной статье.

5. Данные и разметка бланков

Для решения поставленной задачи у компании было получено более 200 заполненных бланков V07. На рисунке 1 изображен один из бланков. Также у компании был запрошен бланк-оригинал формы.

Для разметки данных использовался онлайн инструмент *Supervisely.com* [21], где была размечена вся форма и в дальнейшем результаты разметки были выгружены в формате *.json*, где совместно с *ID* элемента передается и информация о его координатах верхнего левого и нижнего правого угла в формате списка "exterior": $[[54,457],[84,492]]$.

Для разметки использовался оригинальный (не отсканированный) документ. На рис. 2 показан визуальный результат проведенной разметки.

The image shows a 'CHILD VACCINATION' form with the following sections:

- Facility ID:** A row of 10 colored boxes.
- Outreach ID:** A row of 2 colored boxes.
- Date:** Fields for Day, Month, and Year.
- HM place:** A row of 10 colored boxes.
- Mark if experienced afected:** A row of 10 colored boxes with corresponding disease names: BCG, OPV, Polio, PCV, IPV, Hib, MMR, Hib, Hib, Hib.
- Client ID:** A grid of 10 columns and 10 rows of colored boxes.
- Vaccination Status:** A grid of 10 columns and 10 rows of colored boxes, each containing a small icon representing a vaccination status.

The form is labeled 'V07' at the bottom.

Рис. 2. Размеченный бланк Child Vaccination версии V07

Каждый из прямоугольных цветных элементов обладает своим определенным идентификационным номером и другими необходимыми данными, которые можно выгрузить в том числе в формате *.json*. Мы заранее отметили все потенциально важные элементы бланка, а не только

область *Client SPT ID*, но в решении поставленной в рамках данной работы задачи использовали только элементы из области, определенной заказчиком.

6. Подготовка датасета

Параллельно с этим велась работа над подготовкой датасета для проведения *transfer learning*. Было принято решение сконцентрироваться на повышении точности распознавания идентификационных номеров, поэтому акцент был уделен более точной и аккуратной подготовке датасета. Мы решили подавать на модель каждую цифру отдельно, а не использовать модели для распознавания последовательного текста, так как последние обладают меньшей точностью в процессе распознавания. Так как нам заведомо известно в каком формате будет поступать бланк на распознавание, то для повышения результатов распознавания датасет для обучения должен быть подготовлен на тех же бланках компании. Таким образом, на данном этапе перед нами стояла задача получения изображений цифр с области *Client SPT ID* для проведения *transfer learning*.

Сложность подготовки датасета заключалась в том, что отсканированные формы могут иметь значительный перекося в силу неточности процесса сканирования, что с большой вероятностью приведет к неточному выделению элемента при наложении сетки с разметки. Не все отсканированные бланки обладали засечками, которые мы могли бы использовать в качестве ключевых точек для выравнивания, поэтому для решения данной проблемы нами была проанализирована литература и было выбрано использование функции аффинного преобразования для выравнивания отсканированного бланка. Для подготовки датасета был написан скрипт из двух функций с использованием библиотеки *OpenCV* для вырезки каждой из цифр из бланка [16]. Первая функция приводила отсканированное изображение в соответствие с той формой, по которой мы делали шаблон для учебного датасета, т. е. выравнивала ее при использовании функции аффинного преобразования. Вторая функция вырезала интересующие нас элементы из области *Client SPT ID*.

Данный скрипт позволил подготовить датасет из цифр, которые затем вручную были разложены по папкам для дальнейшего использования их в дообучении моделей. На рис. 3 отображен путь и содержание папки 2.

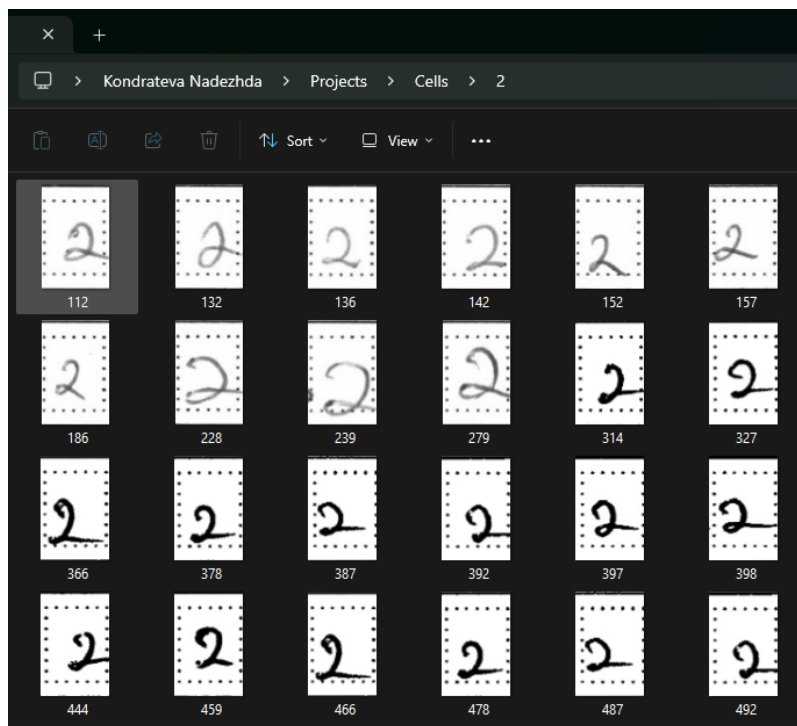


Рис. 3. Содержание папки 2

Всего было сформировано 11 папок с числами от 0 до 9 и пустыми ячейками, которым был подобран лейбл “E” от английского слова “empty”.

В итоге для проведения первичного анализа и выбора модели было подготовлено 1457 изображений для обучения и 333 изображения для проведения тестирования.

7. Выбор моделей

В рамках поставленной задачи мы решили выбрать несколько архитектур сверточных нейронных сетей: *MobileNetV3 Small* [13], *ShuffleNet V2 0.5x* [20] и *ResNet18* [19].

ShuffleNet V2 0.5x – это архитектура сверточной нейронной сети, разработанная для обеспечения высокой производительности на мобильных устройствах и в ограниченных вычислительных средах. *ShuffleNet V2* представляет собой улучшенную версию *ShuffleNet*, известную своей эффективностью благодаря использованию блоков слияния (*shuffle units*), которые позволяют эффективно уменьшить количество вычислений при сохранении качества обучения. Версия *0.5x* (*0.5x* означает уменьшенный размер модели) обеспечивает еще большую эффективность и скорость работы за счет уменьшения количества параметров и сложности архитектуры.

MobileNetV3 Small – это одна из архитектур глубоких сверточных нейронных сетей, разработанная *Google*, оптимизированная для мобильных устройств. Она является улучшенной версией предыдущих моделей *MobileNet*, нацеленных на обеспечение высокой производительности при низком потреблении ресурсов. *MobileNetV3 Small* отличается от своих предшественников улучшенной эффективностью, эффективными блоками операций, такими как *Mobile Inverted Bottleneck (MIB)* и использованием техник оптимизации, таких как *squeeze-and-excitation*, которые помогают улучшить точность при сохранении компактности и скорости работы модели.

ResNet18 – это одна из архитектур глубоких сверточных нейронных сетей из семейства *ResNet (Residual Networks)*, разработанных для решения задач компьютерного зрения. *ResNet18* является небольшой версией *ResNet* с 18 слоями, которая известна своей эффективностью и легкостью для обучения. Эта модель использует концепцию "соединений остатков" (*residual connections*), позволяющую устранить проблему затухающих градиентов при обучении глубоких нейронных сетей. *ResNet18* хорошо сбалансирована между производительностью и размером модели, что делает ее широко используемой в области компьютерного зрения и передачи обученных моделей для различных приложений.

Мы решили взять в эксперимент все три описанные выше модели – *MobileNetV3 Small*, *ShuffleNet V2 0.5x* и *ResNet18*. Каждая из них представляет собой уникальную архитектуру с определенными характеристиками, предназначенными для работы в ограниченных вычислительных средах. Использование всех трех моделей в эксперименте позволит нам оценить их производительность, точность и эффективность в конкретном контексте нашей задачи, а также определить, какая из них лучше всего подходит для наших целей при работе с ограниченными вычислительными ресурсами.

8. Этапы решения

В первую очередь требовалось развернуть систему логирования *ClearML* [5] и хранилище данных *MinIO* [14]. Используя документацию по установке *ClearML* и развертыванию локального сервера, был создан файл *clearml.conf* и в него были внесены все требуемые ключи доступа, полученные после прохождения регистрации в веб-интерфейсе. Вместе с тем было развернуто хранилище данных. Далее хранилище было связано с *ClearML* путем внесения изменений в файл *clearml.conf*. Файл хранится локально на компьютере, таким образом никакая чувствительная информация не является доступной извне и не может быть случайно выложена в сеть.

Далее с использованием библиотек *PyCharm*, *PyCharmLightning* и др. были написаны классы и функции для загрузки датасетов и моделей, настройки параметров моделей, проведения обучения и сохранения процесса обучения и результатов в *ClearML* [7][9][12][15][17][18]. Отдельно отметим, что каждая модель была адаптирована к требуемой задаче классификации на 11 классов путем добавления последнего полносвязного слоя. Модели были дообучены на датасете, который включал в себя данные для обучения и валидации. Этот этап включал в себя циклы обучения с последующей проверкой на валидационных данных. В экспериментах проверялись различные гиперпараметры и наилучшие результаты были получены при использовании оптимизатора *RAdam* и функции потерь

cross-entropy. После каждого цикла обучения проводился анализ результатов, включая визуализацию графиков обучения, изучение кривых обучения и валидации, оценку показателей производительности моделей.

9. Анализ полученных результатов

Результаты, полученные при проведении экспериментов, представлены на рис. 4, 5, 6.

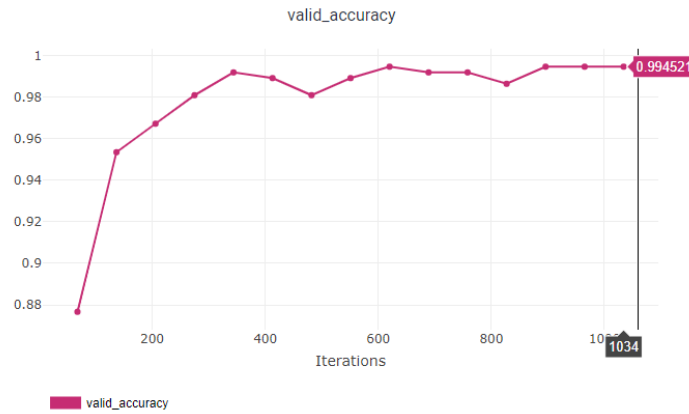


Рис. 4. ResNet18, train_accuracy=1, validation_accuracy=0.995

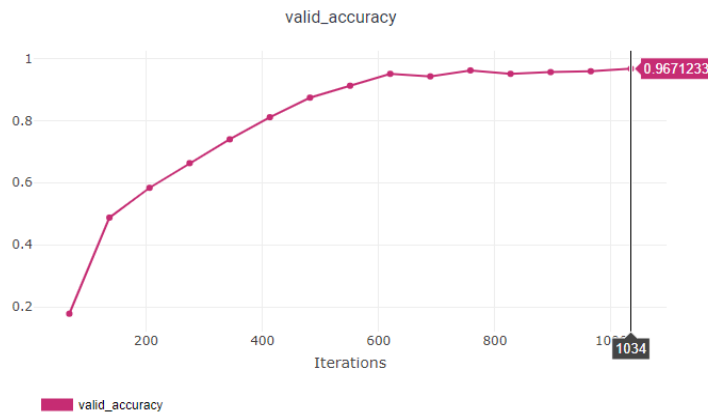


Рис. 5. MobileNet_v3_small, train_accuracy=1, validation_accuracy=0.967

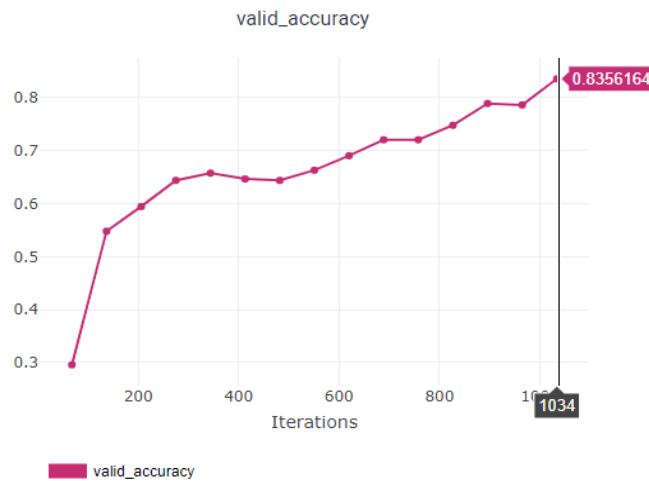


Рис. 6. Shafflenet_v2_005, train_accuracy=0.875, validation_accuracy=0.836

Исходя из полученных результатов оценки производительности трех моделей на предоставленных данных, *ResNet18* проявила наивысшую точность классификации как на обучающем, так и на валидационном наборе данных. Это позволяет сделать вывод о том, что при реализации программного обеспечения на основе моделей глубокого обучения для конкретной задачи классификации на 11 классов, *ResNet18* представляет наилучший выбор благодаря своей высокой точности и эффективности.

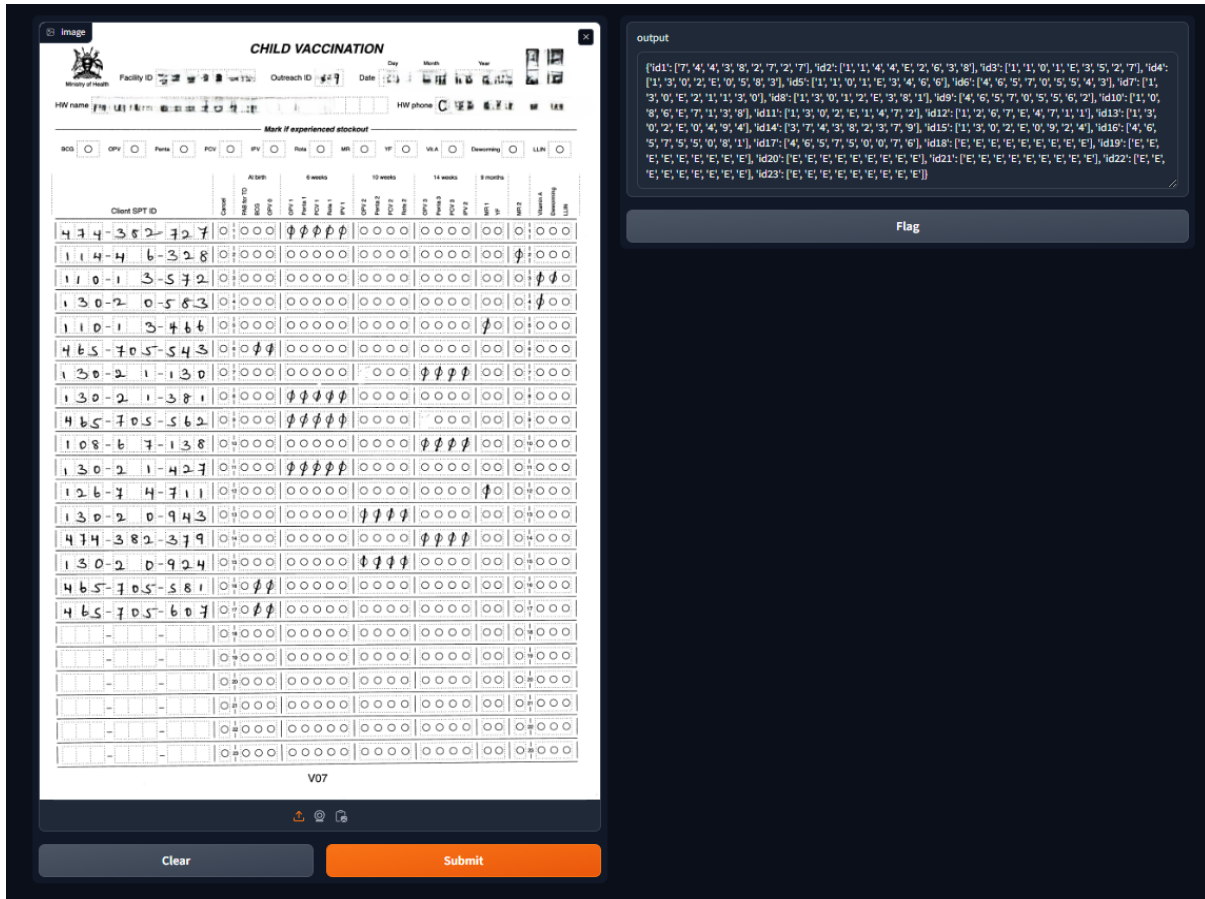


Рис. 7. Веб-интерфейс для демонстрации заказчику

Для демонстрации заказчику работы модуля ИС, разработанного на основе дообученной модели *ResNet18*, был реализован веб-интерфейс при использовании библиотеки *gradio* (рис 7) [8].

Были проанализированы результаты распознавания на нескольких случайно вытянутых бланках и получено 3 неверно распознанных символа из 621, что соответствует результату на валидационной выборке в 0,5% ошибок. Так как для заказчика важен процент верно распознанных идентификационных номеров, которые состоят из 8–9 символов, то при анализе количества верно распознанных ИД мы получили следующий результат: 96% ИД номеров клиентов были распознаны верно. Таким образом, основной запрос со стороны заказчика (разработать такую систему распознавания *ID* из области *Client SPT ID* на бланке *Child Vaccination* версии *V07*, которая снизит количество ошибочно распознанных идентификационных номеров клиентов до 5%) был полностью удовлетворен. Выбранный подход (предварительная выборка интересующих областей, а затем использование моделей распознавания) позволяет достичь поставленных целей, оставляя при этом требования к вычислительным ресурсам на низком уровне.

Заключение

В данной статье рассмотрен процесс разработки модуля системы распознавания рукописного текста для автоматизации сервиса вакцинации. Представленная система демонстрирует значительный потенциал в повышении эффективности работы медицинских учреждений, особенно в условиях ограниченного доступа к интернету. Внедрение такой технологии может существенно снизить

затраты времени и человеческих ресурсов на обработку медицинской документации, что, в свою очередь, способствует улучшению качества обслуживания пациентов.

Проведенный анализ показал, что разработка и внедрение системы распознавания рукописного текста способна не только оптимизировать процессы в медицинской сфере, но и повысить общую производительность и точность в сборе и обработке данных. В будущем данное направление может быть расширено и применено в других областях медицины, что позволит еще более эффективно использовать современные технологии для улучшения здравоохранения.

Таким образом, разработка таких систем открывает новые перспективы для медицинских учреждений, способствуя переходу к более цифровизированному и технологически оснащенному будущему.

Список источников

1. Бобров К.А., Шульман В. Д., Власов К. П. Анализ технологий распознавания текста из изображения // Международный журнал гуманитарных и естественных наук. – 2022. – № 2-3. – С. 124-128. – <https://doi.org/10.24412/2500-1000-2022-3-2-124-128>.
2. Вакцинация и иммунизация // Всемирная организация здравоохранения (ВОЗ). – ВОЗ, 2023. – URL: <https://www.who.int/ru/health-topics/vaccines-and-immunization> (дата обращения: 15.12.2023).
3. АБВУУ : [сайт компании]. – АБВУУ, 2023. – URL: <https://www.abbyu.com> (дата обращения: 23.12.2023).
4. Чабан Л. Н. Теория и алгоритмы распознавания образов : учебное пособие / Л. Н. Чабан; М-во образования Рос. Федерации, Моск. гос. ун-т геодезии и картографии [(МИИГАИК)]. – Москва : [Моск. гос. ун-т геодезии и картографии (МИИГАИК)], 2004. – 71 с.
5. ClearML. – URL: <https://clear.ml/> (дата обращения: 14.01.2024).
6. Google Cloud Vision API Documentation // Google Cloud Documentation. – URL: <https://cloud.google.com/vision/docs> (date of treatment: 15.11.2023).
7. Google Colaboratory. – URL: <https://colab.google/> (дата обращения: 15.01.2024).
8. Gradio App. – URL: <https://www.gradio.app/> (дата обращения: 15.01.2024).
9. Keras. – URL: <https://keras.io/> (дата обращения: 15.01.2024).
10. Krizhevsky A., Sutskever I., Hinton G. E. ImageNet Classification with Deep Convolutional Neural Networks // Proceedings of the 25th International Conference on Neural Information Processing Systems. – 2012. – P. 1097–1105.
11. Gradient-Based Learning Applied to Document Recognition / Y. LeCun, L. Bottou, Y. Bengio, P. Haffner // Proceedings of the IEEE. – 1998. – P. 2278-2324.
12. Matplotlib // ClearML Documentation. – URL: <https://matplotlib.org/> (дата обращения: 14.01.2024).
13. MobileNet V3 Small. – URL: https://pytorch.org/vision/main/models/generated/torchvision.models.mobilenet_v3_small.html (дата обращения: 15.11.2023).
14. MinIO Object Storage for Windows // MinIO Documentation. – Present, MinIO, Inc., 2024. – URL: <https://min.io/docs/minio/windows/index.html> (дата обращения: 15.01.2024).
15. NumPy : [package for scientific computing with Python]. – NumPy team, 2024. – URL: <https://numpy.org/> (дата обращения: 07.01.2024).
16. OpenCV Library. – URL: <https://opencv.org/> (дата обращения: 07.01.2024).
17. pandas - Python Data Analysis Library. – URL: <https://pandas.pydata.org/> (дата обращения: 15.11.2023).
18. Python 3.10.13 documentation. – Python Software Foundation, 2001-2023. – URL: <https://docs.python.org/3.10/> (дата обращения: 07.12.2023).

19. resnet18 — Torchvision main documentation / Linux Foundation. – URL: <https://pytorch.org/vision/main/models/generated/torchvision.models.resnet18.html> (дата обращения: 15.11.2023).
20. shufflenet_v2_x0_5 — Torchvision main documentation / Linux Foundation. – URL: https://pytorch.org/vision/main/models/generated/torchvision.models.shufflenet_v2_x0_5.html (дата обращения: 15.11.2023).
21. Supervisely Platform. – Supervisely OÜ, 2024. – URL: <https://supervisely.com/> (дата обращения: 07.01.2024).
22. Tesseract documentation. – URL: <https://tesseract-ocr.github.io/> (дата обращения: 14.11.2023).