

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

**«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО»**

Кафедра математических основ информатики и олимпиадного
программирования

**ПРОЕКТИРОВАНИЕ И РАЗРАБОТКА МОБИЛЬНОГО
ПРИЛОЖЕНИЯ ДЛЯ СИМВОЛЬНОГО РАСПОЗНАВАНИЯ
РЕЛЬЕФНО-ТОЧЕЧНОГО ТАКТИЛЬНОГО ШРИФТА
БРАЙЛЯ**

АВТОРЕФЕРАТ МАГИСТЕРСКОЙ РАБОТЫ

студента 2 курса 273 группы
направления 02.04.03 — Математическое обеспечение и администрирование
информационных систем
факультета КНиИТ
Фишера Евгения Алексеевича

Научный руководитель

к. ф.-м. н., доцент

Ю. Н. Кондратова

Заведующий кафедрой

к. ф.-м. н., доцент

Ю. Н. Кондратова

Саратов 2022

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	3
1 Рельефно-точечный тактильный шрифт Брайля	4
1.1 Особенности реализаций систем шрифта Брайля	4
1.2 Компьютерная обработка и хранение текста	4
2 Разработка метода распознавания шрифта Брайля	6
2.1 Сегментация текстовых объектов при помощи UNet	6
2.2 Использование модели RetinaNet для распознавания.....	7
3 Разработка мобильного приложения	11
3.1 Интерфейс приложения на React Native	11
3.2 Нативный модуль	11
ЗАКЛЮЧЕНИЕ	14
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ.....	15

ВВЕДЕНИЕ

Очень часто у преподавателя, работающего в школе для детей с нарушениями зрения, возникает потребность во время урока беглым взглядом проверить классную или домашнюю работу обучающихся.

В данной работе рассматриваются различные системы и стандарты рельефно-точечных шрифтов Брайля, их способы представления и возможности хранения в цифровом виде. Также рассматриваются наиболее актуальные наработки в области оптического символьного распознавания как шрифтов Брайля [1], так и плоскочечатных текстов [2]. На основе анализа таких доступных способов описывается новый подход к оптическому распознаванию и разработка мобильного приложения, представляющее собой аналог сканера, который распознает символы рельефно-точечного тактильного шрифта Брайля и предоставляет на экране транскрипцию написанного на русском языке.

Таким образом, целью данной работы является реализация подхода к оптическому распознаванию символов шрифта Брайля и дальнейшее использование разработанного подхода для создания мобильного приложения, основной функцией которого будет являться символьное распознавание и перевод текста, записанного рельефно-точечным тактильным шрифтом Брайля.

Для достижения данной цели были поставлены следующие задачи:

- имплементировать модель нейронной сети для сегментации текстовых объектов для последующей обработки моделью распознавания;
- реализовать обучение на нескольких готовых датасетах;
- на основе полученного обучения модели препроцессинга и локализации текстовых объектов подготовить датасет к обучению выбранной модели распознавания;
- реализовать обучение на подготовленном датасете;
- оценить результаты обучения моделей по известным метрикам;
- написать нативный модуль, использующий обученные модели сетей, для перевода текста, набранного шрифтом Брайля, на фотографии;
- реализовать интерфейс в виде мобильного приложения.

1 Рельефно-точечный тактильный шрифт Брайля

В данной главе содержится информация об основных системах и стандартах шрифтов Брайля в России. Все спецификации, упоминаемые в данной главе, взяты из ГОСТ Р 56832-2015 «Шрифт Брайля. Требования и размеры».

Шрифт Брайля является рельефно-точечным тактильным шрифтом, что означает, что его можно читать посредством осязания. Знаки (буквы, цифры, знаки препинания и служебные знаки) образуются шестью рельефными точками, значение которых получается с одной стороны из количества и позиции точек в основной форме и, с другой стороны, из позиции знака в системе шрифта Брайля. Брайль в основном предназначен для письма и чтения незрячими и плохо видящими людьми.

1.1 Особенности реализаций систем шрифта Брайля

Каждый символ в шрифте Брайля для русского, английского и некоторых других языков изображается с использованием шести точек. Также перевод в систему Брайля является симметричной операцией, то есть каждому символу исходного алфавита будет соответствовать один символ шрифта Брайля. Данный факт существенно ограничивает количество используемых в тексте символов, и не каждый язык способен уместить весь свой алфавит с цифрами и знаками препинания в 63 символа. Однако существуют и применяются также и другие более сложные системы, имеющие восьмиточечную запись или контекстно-зависимые шеститочечные системы. Все эти усложнения добавляются для расширения количества возможных символов алфавита. Важно упомянуть, что при разработке приложения такие усложненные системы не были приняты во внимание ввиду их редкого применения в России, перевод полученного текста осуществляется относительно стандартной шеститочечной системы Брайля. Точки в такой системе расположены в два столбца и три строки.

1.2 Компьютерная обработка и хранение текста

Для простоты запоминания и компьютерного перевода обычно используют так называемую координатную сетку Брайля, где каждая точка символа имеет свой порядковый номер как закрепленную позицию от 1 до 6.

Такая сетка позволяет записывать Брайль не только имеющимися символами Unicode, но и последовательностями цифр, каждая из которых — это своя позиция следующей точки при прокалывании. Или же двоичными шестибитовыми последовательностями, где 0 будет означать отсутствие выпуклой точки в позиции с номером текущего разряда, начиная со старших, а 1 — ее наличие.

Данный способ записи при хранении символов шрифта Брайля используется в разработке приложения. Так хранятся словари соответствия брайлевских знаков буквам алфавита используемого языка.

2 Разработка метода распознавания шрифта Брайля

Данная работа предлагает потенциально новый подход к OCR рельефно-точечных шрифтов, основываясь на методе, предложенном в [1], которая в свою очередь опирается на архитектуру сверточной сети для обнаружения объектов, описанную в [3]. Таким образом, в этой работе рассмотрено применение модели RetinaNet, сверточной нейронной сети, включающей в себя как слои классификатора найденных объектов, так и слои, ответственные за регрессию и поиск областей интереса (RoI) для OCR [3]. Проводится сравнение различных моделей для извлечения признаков из изображений, который чаще всего используются в задачах классификации, нахождения объектов и сегментации [4, 5].

Однако существенным отличием от методов, предложенных в [3,6], а затем в [1] будет выступать этап препроцессинга фотографий. Работа описывает и реализует процесс предварительной обработки изображений через CNN-архитектуру UNet. Работа концентрирует внимание именно на этом подходе к сегментации текстовых объектов, так как описанный подход контурного анализа показал себя не таким универсальным в условиях, когда текст изображен на сложном фоне или запечатлен при сложном освещении, как было упомянуто ранее и в предыдущих работах. Этап препроцессинга должен сильно упростить задачу самой модели OCR и увеличить точность распознавания на датасете, который прошел препроцессинг.

2.1 Сегментация текстовых объектов при помощи UNet

Для более успешного обнаружения и классификации символов нейронной сетью стоит предварительно описать такой способ предварительной обработки изображения, который выдавал бы приблизительные области интереса, в которых находятся текстовые объекты, распознавание которых необходимо провести. Текстовыми объектами в данном случае могут являться, как и текст целиком, так и обособленно стоящие предложения или даже слова. Это позволит не только отфильтровать лишние области изображения, что упростит работу самой модели распознавания, но и позволит группировать отдельно распознанные символы в свои текстовые объекты для сохранения семантической целостности.

В качестве архитектуры для сегментации была выбрана модель UNet.

Архитектура сверточной нейронной сети UNet была разработана как раз для решения подобных задач [7]. UNet по сути своей является классической для сверточной FCN и состоит из двух отличимых ветвей ветвей: сжимающего пути (contracting path) и расширяющего пути (expanding path).

Также стоит упомянуть про используемую функцию потерь для обучения UNet. Выбор функции потерь очень важен для любой архитектуры нейронных сетей, чтобы как можно точнее рассчитывать ошибку вывода модели для ее обучения, и поэтому с 2012 года различные исследователи начали разрабатывать более специфичные функции потерь для каждой конкретной предметной области, чтобы получить лучшие результаты для своих наборов данных. В этой работе было рассмотрено несколько таких функций потерь для задач сегментации изображений, которые, как было доказано в [8], обеспечивают хорошие результаты в различных предметных областях. Итогом анализа доступных функций потерь стал выбор отрицательной логарифмической вероятности, что позволило достигнуть результатов обучения UNet, представленных в таблице 1 в метриках точности и Dice-коэффициента на датасетах DSBI и Angelina.

Таблица 1 – Результат обучения UNet на датасетах DSBI и Angelina

Датасет	DI	Точность
DSBI	0.4687	0.9649
Angelina	0.4590	0.9770

Однако в сыром выводе сети может присутствовать дополнительное зашумление, что может повлиять на качество обучение модели OCR. Для решения данной проблемы было принято решение использовать морфологические преобразования маски. Для реализации было решено применить инструментарий библиотеки OpenCV.

2.2 Использование модели RetinaNet для распознавания

Полученное в результате применения маски изображение можно использовать для дальнейшей обработки с целью обнаружения и сегментации отдельных символов. Для этого рассмотрим подход с использованием моделей сверточных нейронных сетей для обнаружения объектов.

Большинство подходов, использующих машинное обучение для символического распознавания, применяют к так называемой задаче распознава-

ния текста в «сцене». Это означает, что решение задачи распознавания текста будет основываться на предположении, что искомые символы находятся в контексте некоего текста. То есть, прежде чем перейти к распознаванию (классификации), модель находит и локализирует отдельные слова [2]. Таким образом, в качестве «объектов» выступают отдельные слова, части слов или словосочетания.

В более общих случаях применяют подход посимвольного распознавания, который позволяет отказаться от предположения о контексте и работать уже с отдельными символами [2]. Здесь уже искомыми объектами будут являться сами символы.

В некоторых задачах второй подход является даже более успешным из-за упрощенной, по очевидным причинам, задаче классификации. Из-за особенностей используемых датасетов, о которых будет описано чуть позже, и упомянутой слабой контекстной зависимости шрифта Брайля в данной работе было решено использовать именно этот подход.

В данной работе для целей обнаружения и сегментации символов шрифта Брайля предлагается использование модели RetinaNet, которая уже хорошо себя зарекомендовала в решении подобных задач обнаружения, как это видно из [3] и [1].

RetinaNet — это архитектура сверточной нейронной сети для решения задач нахождения объектов, состоящая из нескольких инкапсулированных частей: основной сети, более известной в машинном обучении как «backbone», и двух подсетей, каждая из которых используется при решении конкретных локализованных задач: нахождения координат объектов и их классификации.

Так как в дальнейшем использование обученной модели планируется в рамках мобильного приложения, которое будет запускаться на мобильных устройствах, у которых могут быть весьма ограниченные ресурсы хранилища и памяти, при обучении RetinaNet было решено рассмотреть архитектуры ResNet, DenseNet и MobileNetV в качестве backbone-сетей.

Для обучения RetinaNet использовалась функция потерь, основанная на IoU метрике. IoU или Intersection over Union — это термин, используемый для описания степени перекрытия двух областей. Чем больше область перекрытия, тем больше IoU. IoU довольно часто используется в качестве мет-

рики, снимаемой с моделей для обнаружения объектов, для оценки данных моделей [2]. В этом случае области, относительно которых вычисляют IoU, будут являться описывающими искомые объекты прямоугольниками, а сама метрика будет вычисляться по формуле:

$$IoU = \frac{\text{Площадь пересечения прямоугольников}}{\text{Площадь объединения прямоугольников}}$$

Для метрики распознаваний чаще всего используют так называемую AP (Average Precision) метрику, которая для каждого реального объекта считает IoU с объектами того же класса, если таковые есть, и берет среднее значение от всех высчитанных IoU [2]. Чтобы получить из этой метрики полноценную функцию потерь достаточно добавить сложение числителя и знаменателя с некоторой константой для вычислительной стабильности и вычесть получившееся значение AP из заданной константы, значение которой обычно больше или равно 1. Такую функцию потерь называют IoULoss. Обучение было проводилось на протяжении в 100 эпох, по окончании которых были выбраны веса с наибольшим значением метрик.

В таблице 2 представлены результаты обучения RetinaNet на датасете DSBI, используя модели ResNet50, MobileNetV3 и DenseNet121 в качестве backbone модуля. Результаты представлены в виде метрики AP, которая была описана ранее. Второй столбец последней строки таблицы представляет собой эталонное значение метрики AP, которое может выдавать RetinaNet, обученная на огромном, по сравнению с DSBI и Angelina, датасетом COCO, используя FPN-аугментированную версию ResNet50.

Таблица 2 – Результат обучения RetinaNet в метрике AP с разными моделями backbone-сети на датасетах DSBI и COCO

Модель backbone-сети	AP: DSBI & Angelina	AP: COCO
MobileNetV3 & RetinaNet	0.1733	-
DenseNet121 & RetinaNet	0.2011	-
ResNet50-FPN & RetinaNet [3]	0.3317	0.3640

Вывод сети на примере одиночного экземпляра валидационных данных представлен на рисунке 1.

У прямоугольника каждого объекта в выводе присутствует подпись, соответствующая присутствию отдельных точек символа.

Полученные показатели средних AP-метрик позволяют сделать вывод

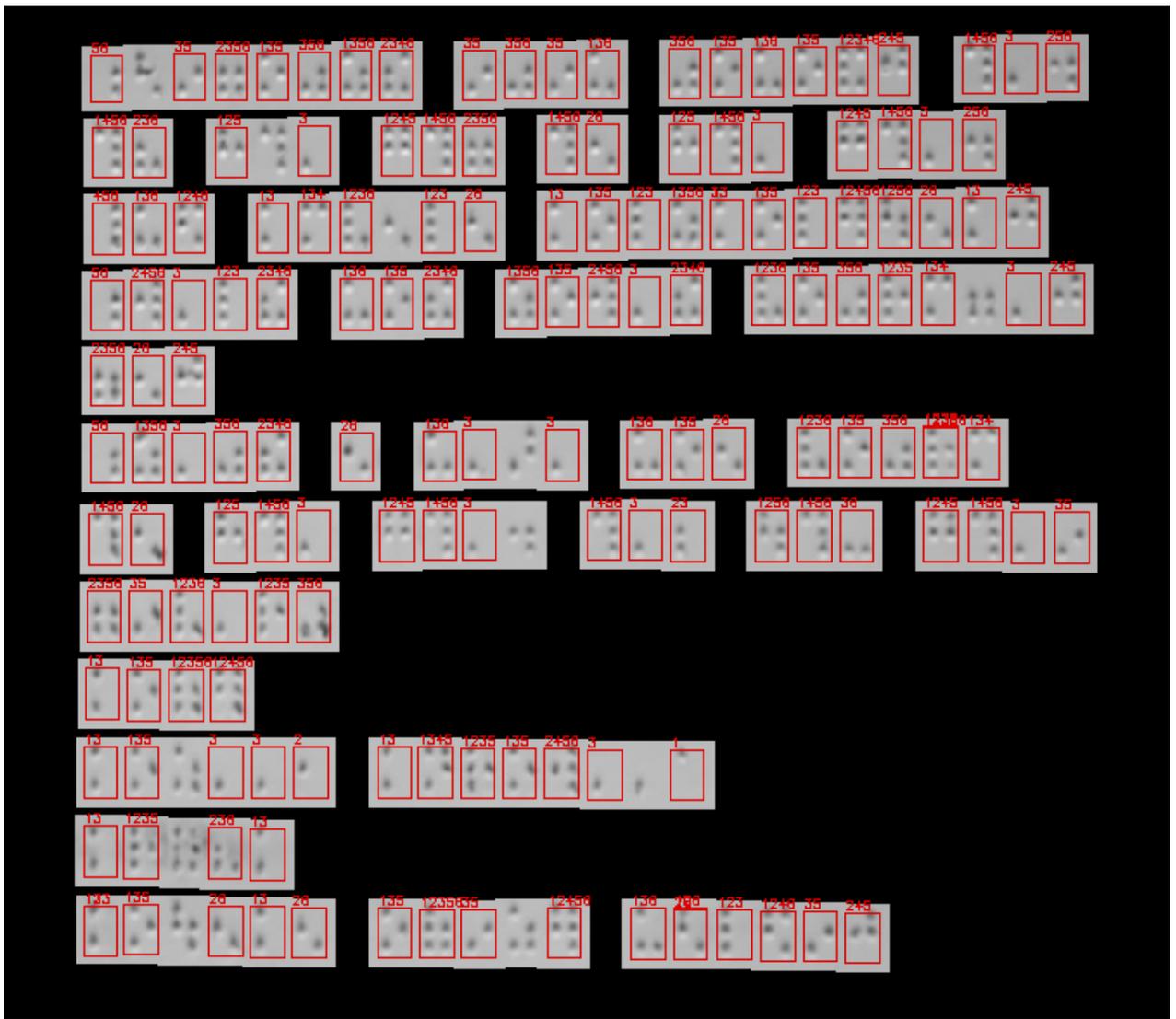


Рисунок 1 – Вывод обученной модели RetinaNet

о пригодности предлагаемого работой метода для оптического символьного распознавания рельефно-точечных шрифтов. По полученным метрикам также можно сделать вывод о том, что модель ResNet50 лучше всего показала себя в сравнении с другими архитектурами backbone-сетей.

3 Разработка мобильного приложения

Приложение, реализуемое в рамках данной работы, разрабатывается при помощи фреймворка React Native, что в теории позволит создать артефакты сборки, которые могут быть использованы как на ОС Android, так и на iOS. Пользовательский интерфейс приложения должен представлять собой удобную платформу, которая прежде всего имеет возможность отображения превью кадров, получаемых с камеры девайса, возможность делать снимки и в одно нажатие получать перевод сфотографированного текста.

3.1 Интерфейс приложения на React Native

Функционал самого пользовательского интерфейса не предполагался обширным. Также не шло речи о задействовании каких-либо эксклюзивных нативных функций устройств, что позволило отказаться от написания отдельных версий нативных приложений для каждой поддерживаемой ОС. Единственным исключением будет являться вышеупомянутый нативный модуль OCR.

Учитывая приведенные факты, было решено реализовывать интерфейс при помощи JavaScript-фреймворка React Native.

В рамках данного проекта было задумано реализовать две страницы внутри приложения: стартовую домашнюю страницу и страницу с функционалом камеры и возможностью использовать сделанные фотографии в качестве параметров при вызове программного интерфейса нативного модуля, отвечающего за перевод (см. рисунок 2).

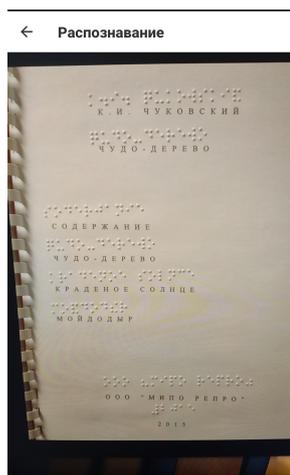
После того, как снимок был сделан, путь до его временного файла передается методу нативного модуля, который обернут в promise-объект для асинхронного вызова метода. При успешном выполнении нативного метода распознавания текста на изображении вызывается модальное окно. Содержимым модального окна будет являться значение строки, пришедшая в результате выполнения метода.

3.2 Нативный модуль

Потребность в нативном модуле возникла из-за необходимости использовать функционал библиотеки OpenCV для предварительной и промежуточной обработки изображений, о которых будет сказано подробнее позже в этом разделе, и для использования обученных весов моделей UNet и RetinaNet,

Доступный Брайль

Начать



Перевести

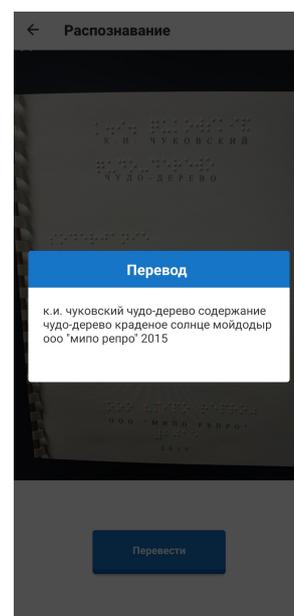


Рисунок 2 – Пользовательский интерфейс

которые непосредственно и осуществляют локализацию текстовых объектов, сегментацию и классификацию символов шрифта Брайля.

На текущий день не существует более элегантного программного способа подключения функционала OpenCV в мультиплатформенные приложения, описанные на таких гибридных фреймворках, как React Native и Flutter, кроме как создания отдельных нативных модулей для каждой поддерживаемой ОС, но и этот способ далек от элегантности.

Чтобы использовать библиотеки, написанные на C/C++, такие как OpenCV Android предоставляет разработчикам программный интерфейс инструментов NDK. Разработчик может использовать NDK для компиляции C и C++ кода в нативные библиотеки, которые упаковываются в APK с помощью Gradle на этапе сборки приложения, который был описан выше. Из кода самого приложения методы нативных библиотек доступны через интерфейс JNI.

Для того, что получить вывод обученных модель OpenCV предоставляет модуль DNN, способный читать веса форматов различных популярных движков для глубокого обучения. Однако при использовании форматов весов таких движков нужно указывать большое количество конфигурационных файлов, чтобы модуль смог адекватно прочитать бинарный формат весов. Вместо этого в рамках данной работы обученные веса приводятся к формату ONNX.

Логика работы модуля распознавания строится на поочередном прогоне сделанной фотографии через две обученные модели сверточных нейронных сетей: UNet и RetinaNet. Стоит отметить, что каждый шаг будет нуждаться в своем этапе предварительной обработки входных данных.

ЗАКЛЮЧЕНИЕ

В результате написания данной работы были представлены и описаны различные подходы к предварительной обработке изображений, включая подход разработки конвейера препроцессинга с использованием машинного обучения. Также была описана и реализована архитектура сверточной нейронной сети UNet для сегментации текстовых объектов на изображении и создания маски текста для последующего использования в символьном распознавании.

Был подробно изучен подход к решению задачи OCR при помощи моделей нейронных сетей для обнаружения объектов. Было проведено сравнение различных применяемых моделей, выбрано несколько моделей для практического сравнения. Было внесено несколько изменений в архитектуру RetinaNet с учетом контекста применения данной модели. Модель была обучена на датасетах DSBI и Angelina с применением трех различных backbone-сетей. Было произведено сравнение результатов обучений этих моделей по AP метрике. Результаты также сравнивались с эталонными результатами обучения RetinaNet на датасете COCO.

Результаты обучения были проанализированы, и был сделан вывод о жизнеспособности предлагаемого метода и пригодности его применения для целей оптического символьного распознавания шрифтов Брайля и использования в рамках создания мобильного приложения.

Также в результате написания работы было разработано мобильное приложение для символьного распознавания рельефно-точечного тактильного шрифта Брайля. В рамках разработки приложения был написан мультиплатформенный интерфейс с применением фреймворка ReactNative, а также — нативный модуль на Java для сборки приложения под операционную систему Android. Нативный модуль представляет программный интерфейс для обработки изображений и перевода обнаруженного текста с шрифта Брайля в плоскочечную версию текста.

Также по материалам работы была написана статья «Символьное распознавание рельефно-точечного тактильного шрифта Брайля при помощи сверточных нейронных сетей» и опубликована в сборнике материалов Всероссийской научно-практической конференции «Образование. Технологии. Качество» [9].

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- 1 *Зуев, В. А.* Распознавание символов Брайля на фотографиях // Сборник материалов XV Всероссийской молодежной научно-инновационной школы. — Москва, Россия: Интерконтакт Наука, 2021. — С. 145–146.
- 2 *Bourez, C.* Object detection deep learning networks for Optical Character Recognition / C. Bourez, A. Coquard // ICLR 2019 Conference. — New Orleans, USA: ICLR, 2019. — Pp. 1–15.
- 3 *Lin, T.* Focal Loss for Dense Object Detection / T. Lin, P. Goyal, R. B. Girshick, K. He, P. Dollár // *IEEE Transactions on Pattern Analysis and Machine Intelligence*. — 2020. — Vol. 42, no. 2. — Pp. 318–327.
- 4 *Sandler, M.* Inverted Residuals and Linear Bottlenecks: Mobile Networks for Classification, Detection and Segmentation // CVPR 2018. — Utah, USA: IEEE, 2018. — Pp. 4510–4520.
- 5 *He, K.* Deep Residual Learning for Image Recognition // 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). — Nevada, USA: IEEE, 2016. — Pp. 770–778.
- 6 *Tadesse, S.* Feature Extraction and Classification Schemes for Enhancing Amharic Braille Recognition System: Ph.D. thesis / Addis Ababa University. — Addis Ababa, Ethiopia, 2011.
- 7 *Ronneberger, O.* U-Net: Convolutional Networks for Biomedical Image Segmentation // Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015 / edited by N. Navab, J. Hornegger, W. M. Wells, A. F. Frangi. — Cham, Switzerland: Springer International Publishing, 2015. — Pp. 234–241.
- 8 Loss Odyssey in Medical Image Segmentation / J. Ma, J. Chen, M. Ng, R. Huang, Y. Li, C. Li, X. Yang, A. L. Martel // *Medical Image Analysis*. — 2021. — Vol. 71. — Pp. 10–35.
- 9 *Фишер, Е. А.* Символьное распознавание рельефно-точечного тактильного шрифта Брайля при помощи сверточных нейронных сетей // Образование. Технологии. Качество: Материалы Всероссийской научно-практической конференции. — Москва, Россия: Издательство «Перо», 2022. — С. 148–154.