ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»

Факультет компьютерных наук
Образовательная программа «Программная инженерия»

УДК 004.93

СОГЛАСОВАНО
Руководитель проекта
Руководитель команды в совместной лаборатории РАУ и ИСП РАН
(Армения, г. Ереван)

_____ Ц. Г. Гукасян
«___» _____ 2020_ г.

УТВЕРЖДАЮ
Академический руководитель образовательной программы «Программная инженерия», профессор департамента программной инженерии, канд. техн. наук

_____ В. В. Шилов
«___» _____ 2020_ г.

Отчёт по курсовой работе

Исследование и разработка методов автоматического распознавания и перевода текста англоязычных комиксов на русский язык

по направлению подготовки бакалавров 09.03.04 «Программная инженерия»

Выполнил
студент группы БПИ181
образовательной программы
09.03.04 «Программная инженерия»

А. А. Матевосян

_____
Подпись, Дата

Москва 2020

Реферат

Отчёт 25 с., 19 рис., 4. табл., 20 источн.

Ключевые слова: *машинное обучение; обработка изображений; компьютерное зрение; OCR; машинный перевод; digital comics;*

В отчете представлены результаты курсового проекта на тему “ Исследование и разработка методов автоматического распознавания и перевода текста англоязычных комиксов на русский язык“, выполненной на основе учебного плана подготовки бакалавров по направлению 09.03.04 "Программная инженерия" и утвержденная академическим руководителем тема курсового проекта.

Объект исследования – цифровые американские комиксы.

Предмет исследования - технологии автоматического распознавания фрагментов текста с фрагментов выносок из цифровых американских комиксов и их автоматический перевод.

Цель исследования – исследование и разработка методов автоматического распознавания и перевода текста англоязычных комиксов на русский язык.

Задачи исследования:

1. Исследование существующих методов автоматического распознания текста из комиксов:
    a. Исследование существующих методов обнаружения участков текста в комиксе;
    b. Создание набора тестовых данных комиксов и их текстов;
    c. Исследование существующих OCR [9] методов;
    d. Проведение экспериментов по определению качества методов распознавания.

2. Исследование и оценка качества существующих инструментов машинного перевода:
    a. Создание набора тестовых данных - текст с его переведенным вариантом;
    b. Проведение экспериментов по определению качества перевода.

3. Исследование методов вставки переведенного текста вместо оригинального в комиксах.

Методы исследования:

- изучение монографий, публикаций и статей [2] [3] [5] [7] [4];
- сравнительный анализ;
- машинное обучение.

Научная новизна работы может состоять в следующем:

- анализ решений, которые включают в себе комбинацию как распознавание текста с комиксов, так и автоматический перевод текста с английского на русский

Достоверность научных результатов может быть подтверждена результатами экспериментальных исследований с использованием существующих программных решений и материалов, опубликованных в рамках этой работы.

Практическая значимость. Результаты работы могут быть использованы для интеграции существующих решений в ряде вспомогательных программ. Например, распознавание фрагментов текста поможет переводчикам быстрее и легче переводить комиксы, так же, пользователи смогут без ожидания перевода читать комиксы. Так же исследуемые алгоритмы смогут помочь в преобразовании страницы комикса из цифрового изображения в более удобный формат данных (разметка фрагментов с выносками, аннотации с текстом и другие метаданные).

Ожидаемые результаты работы

- Исследование и оценка качества существующих алгоритмов цифрового зрения в задаче распознавания выносок из комиксов
- Исследование и оценка качества существующих алгоритмов оптического распознавания символов в задаче распознавания текстов из комиксов
- Исследование и оценка качества инструментов машинного перевода для текстов из комиксов
- Определение подходящего-продуктивного набора алгоритмов для распознавания и перевода комиксов

Содержание

Введение

Комиксы являются одними из самых популярных и знакомых форм графического контента по всему миру и играют большую роль в распространении культуры страны. В настоящее время, массовая оцифровка и распространение материалов в цифровом формате позволяют читать страницы на мобильных телефонах и устройствах.

Обычно комиксы не имеют официального перевода от самих издателей. Дистрибуторы покупают права на распространение для определенного числа комиксов и после перевода или иногда и без продают их. Такой механизм распространения сильно ограничивает читателей от чтения огромного числа выпуска комиксов. В основном для чтения таких комиксов, приходится искать любительские переводы в сети или читать на оригинальном языке. Важно понимать, что по сравнению с книгами, процесс перевода комиксов является более комплексной задачей. Так как комиксы изначально распространяются как набор изображении, основная работа происходит с графическим контентом, а не с текстом. Если в случаи книг, переводчик уже имеет оригинальный текст и прямо работает с ним, а редактор может легко отредактировать переведенные фрагменты текста и легко создать переведённый вариант книги, не отличаемый от оригинала, то при работе с комиксами, приходится сталкиваться со многими трудностями, которые сильно мешают создать переведенный вариант комиксов.

Необходимость извлечения текста из соответствующих фрагментов является одной из главных проблем, которые возникают при переводе комиксов. Часто авторы могут использовать нестандартные шрифты или рисовать текст как иллюстрационный рисунок, что значительно осложняет работу. Кроме этого, нужно учесть также особенности размещения уже переведенного фрагмента вместо оригинального текста. Часто переводчики бросают дело и перестают выпускать переводы. Из-за вышеописанных трудностей появляется необходимость средства для автоматического перевода комиксов. Кроме того, в ходе автоматического перевода, появляется возможность индексации комикса, так выполнения перевода включает в себя процесс анализа содержания комикса.

Из-за сложной структуры и отсутствия определённых стандартов, очень сложно проводить исследования. Сама структура страниц, полисы и панели на нем могут сильно отличаться, так же может отличаться направление фрагментов с текстом (слева направо, с верху вниз).

Несмотря на уже принятые соглашения по разработке комиксов, сами авторы не ограничены ими и могут действовать своевольно. Часто панели могут перекрыть друг друга или иметь нестандартный общий вид.

Целью данной работы является, исследование и разработка методов автоматического распознавания и перевода текста англоязычных комиксов на русский язык.

В качестве задач будут рассматриваться следующие пункты:

1. Исследование существующих методов автоматического распознавания текста из комиксов
    a. Исследование существующих методов обнаружения участков текста в комиксе
    b. Создание базы тестовых данных комиксов и их текстов
    c. Исследование существующих OCR методов
    d. Проведение экспериментов по определению качества методов распознавания
2. Исследование существующих методов автоматического перевода текста
    a. Создание базы тестовых данных текст с его переведенным вариантом
    b. Проведение экспериментов по определению качества перевода
3. Исследование методов вставки переведенного текста вместо оригинального в комиксах

В данной работе будут рассматриваться некоторые алгоритмы и подходы, которые могут быть использованы для обнаружения выносок и извлечения из них фрагментов текста, а также автоматический перевод и вставка переведенных фрагментов на страницу.

Обзор комиксов

Структура комиксов

Каждый комикс состоит из страниц (пластина), которая разбивается на фрагменты полосы (см. рис. 1). Пластины в свою очередь разбиваются на панели. Панель показывает зафиксированный момент истории. Они обычно разделены между собой решетками. Иногда иллюстрации одной панели могут выйти из границ и перекрыть другую панель.

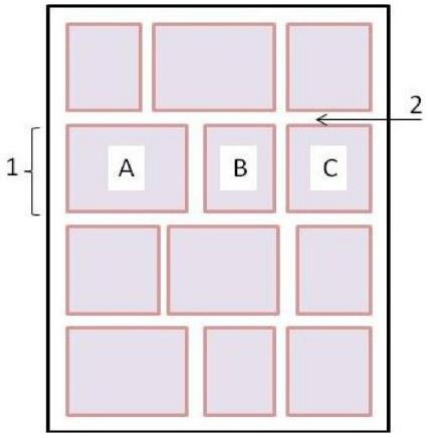

Рис. 1: Пример пластинки: (1) полоска; (A) , (B) , (C) панель ; (2) решетка

Существует целый ряд комиксов с разнообразным оформлением. Автоматический анализ всех этих комиксов - настоящая проблема. Также по сравнению со старыми классическими комиксами, сегодняшние комиксы распространяется только в цифровом формате. Такие комиксы создаются с помощью разных графических программ. Другие же более старые комиксы создавались вручную и не имели изначально цифрового варианта. Конечно, такие комиксы можно сохранить в цифровом виде, но для этого нужно провести сканирование самих комиксов.

В основном комиксы можно разделить на три категории: американские, японские и корейские: Корейские в отличие от американских не имеют панелей, текст читается с верху вниз, а полосы отсутствуют. Японские же имеют другое направление чтения по сравнению с американскими и обычно бывают в черно белом варианте. В добавок к этому в японских комиксах текст пишется с верху вниз.

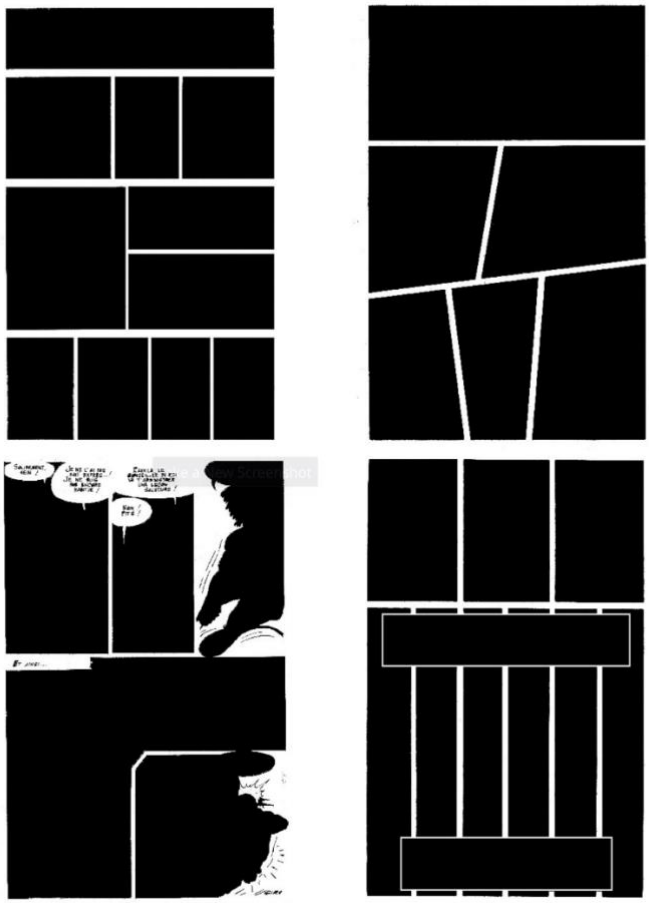

Рис. 2: Примеры оформления комиксов

В этой работе мы рассматриваем только традиционные комиксы с однотонными цветными пластинами, которые изначально распространялись в цифровом формате.

Построение датасета

Для проведения экспериментов и дальнейшего анализа полученных результатов нужно построить датасет. В него должны входить цифровые страницы комиксов и соответствующая аннотация. Аннотация включает в себе расположение выносок и текст, который находиться в них. Можно сделать подобный датасет с помощью ручной обработки страниц, также можно взять существующий датасет. Была попытка создания датасета вручную, но из-за некоторых обстоятельств, было решено использовать уже существующие датасеты. Был проведен поиск существующих датасетов, в результате которого, были обнаружены следующие варианты:

1. Graphic Narrative Corpus (подборка из 240 западных комиксов, опубликованных с 1970-х по 2010-е годы) [3]
2. Manga109 (подборка из 109 наименований манги, опубликованных с 1970-х по 2010-е годы (всего 21142 изображения) [10]
3. eBDtheque (подборка из ста страниц комиксов из Америки, Японии (манга) и Европы) [11]

После анализа датасетов в сети, и исключения Manga109, из-за несовместимого типа комиксов(“манга”), были отправлены запросы для получения копии оставшихся датасетов. Пришло одобрение только от авторов eBDtheque.

Структура датсета eBDtheque

Содержание датасета eBDtheque:

1. 100 страниц
2. 850 панелей
3. 1081 выносок
4. 1620 персонажей комиксов
5. 4693 строк текста

Семантические и визуальные аннотации на данной странице собраны в одном файле SVG (Scalable Vector Graphics - масштабируемая векторная графика). Каждый файл представляет собой одну страницу комикса.

Файловый документ состоит из корневого узла с метаданными о странице. Оно включает дочерние элементы, которые описывают разные классы единиц комикса (панель, линия, выноска, страница).

```
<?xml version="1.0" encoding="UTF-8"?>
<svg>
    <svg>
        <image xlink:href="cosmozone.jpg" xlink:show="embed"/>
        <metadata collectionTitle="Cosmozone" editorName="Cyborga" pageNumber="16"/>
    </svg>
    <style type="text/css" xml:space="preserve"><![CDATA[
        .Panel { fill: red; stroke: none; opacity: 0.2; visibility: visible; }
        ...
]]></style>
    <svg class="Panel">
        <polygon points="10,86 277,86 277,356 10,356 10,86">
            <metadata rank="2"/>
        </polygon>
        ...
    </svg>
    <svg class="Line">
        <polygon points="210,195 240,195 240,206 210,206 210,195">
            <metadata>Cool.</metadata>
        </polygon>
       ...
    </svg>
    <svg class="Balloon">
        <polygon points="156,112 291,112 291,222 156,222 156,112">
            <metadata shape="rectangle" rank="1" queueDirection="SW"/>
        </polygon>
    </svg>
</svg>
```

Рис. 3: Пример файла из датасета в формате SVG

Некоторые файлы из датасета не соответствовали критериям объявленными ранее, поэтому после ручной обработки не соответствующие страницы были исключены из датасета.

Автоматическое распознавание выносок

Большая часть текста в комиксах находится в выносках, таким образом, обнаружение этих элементов является необходимым условием для OCR. Вследствие этого задача обнаружения участков текста в комиксе, превращается в задачу автоматическое распознавание выносок.

Для обнаружения выносок, в основном используются два типа алгоритмов:

1. Алгоритмы, которые работают с помощью машинного обучения. Строиться модель, и оно обучается с помощью тестовых данных.
2. Используются стандартные алгоритмы работы с изображениями.

Обнаружение и сегментация выносок на основе CNN

Сегментация — это разбиение изображения или сканирование на несколько сегментов или наборов пикселей - задача, с которой прекрасно справляется искусственный интеллект (ИИ).

Исследователи родительской компании Google Alphabets DeepMind недавно опубликовали в академической статье, что они разработали систему, способную сегментировать КТ-сканирование с "показателями, близкими к человеческим". Теперь ученые из Университета Потсдама в Германии разработали инструмент сегментации ИИ для чуть более карикатурной среды: комиксов. В качестве сегментов рассматриваются набор пикселей, которые входят в выноски.

В своей статье "Deep CNN-based Speech Balloon Detection and Segmentation for Comic Books" [2], описывается нейронная сеть (т. е. слои математических функций, смоделированные по образцу биологических нейронов), способная обнаруживать и изолировать выноски в графических романах и комиксах. Оно использует полностью конволюционный подход, предсказывающий сегментацию изображения на основе пикселей. Сеть основана на архитектуре U-Net и использует конволюционную часть модели VGG-16. Исследователи протестировали подготовленную систему ИИ на подборке изображений, полученных из Graphic Narrative Corpus.

Показательно, что ей удалось приблизить иллюзорные контуры - границы выносок, очерченные не физическими линиями, а "воображаемыми" продолжениями линий, определяющих пространство между панелями. Сам датасет содержит около 750 аннотированных страниц Graphic Narrative Corpus [3], представляющих англоязычные графические романы с разнообразными стилями.

Бинарные маски были сгенерированы из аннотаций Graphic Narrative Corpus, которые представлены списками вершин многоугольников в формате XML. Для задачи сегментации шаров был снижен масштаб изображений тела. с фиксированным размером 768 × 512 пикселей в RGB.

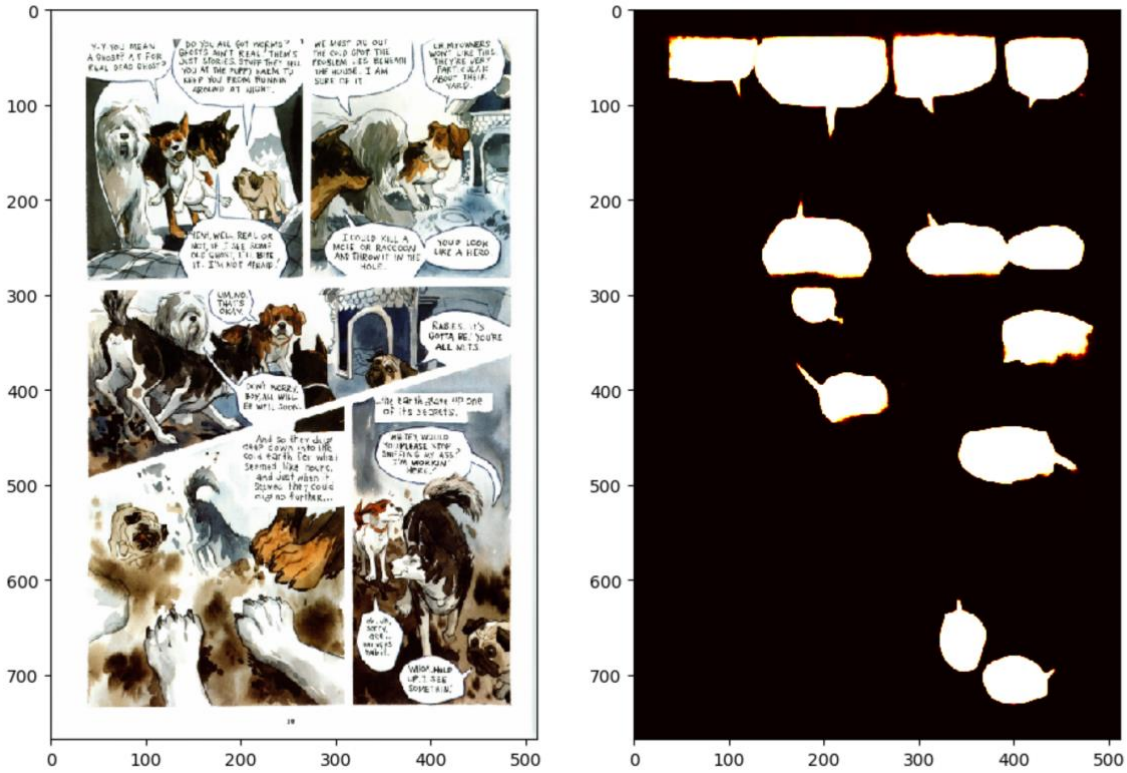

Рис. 4: Пример работы модели

Алгоритмы и методы обнаружения выносок на основе стандартных функции работы с изображениями

Дальше будут описываться две другие методы обнаружения выносок, которые используют библиотеку OpenCV [12].

Метод 1: OpenCV Basic

Прежде всего, нам необходимо найти все (или как можно больше) выносок на странице комикса. К счастью, выноски обычно имеют относительно четко определенные грани и в основном прямоугольную форму. Чтобы использовать эти свойства при обнаружении выносок, мы используем функцию findContours() из cv2(модуль opencv на Python) для распознавания краев в странице комиксов и привязываем их прямоугольниками (boundingRect()), которые затем становятся кандидатами в выноски. Как показано ниже, функция findContours() улавливает большое количество шумов, которые не являются выносками.

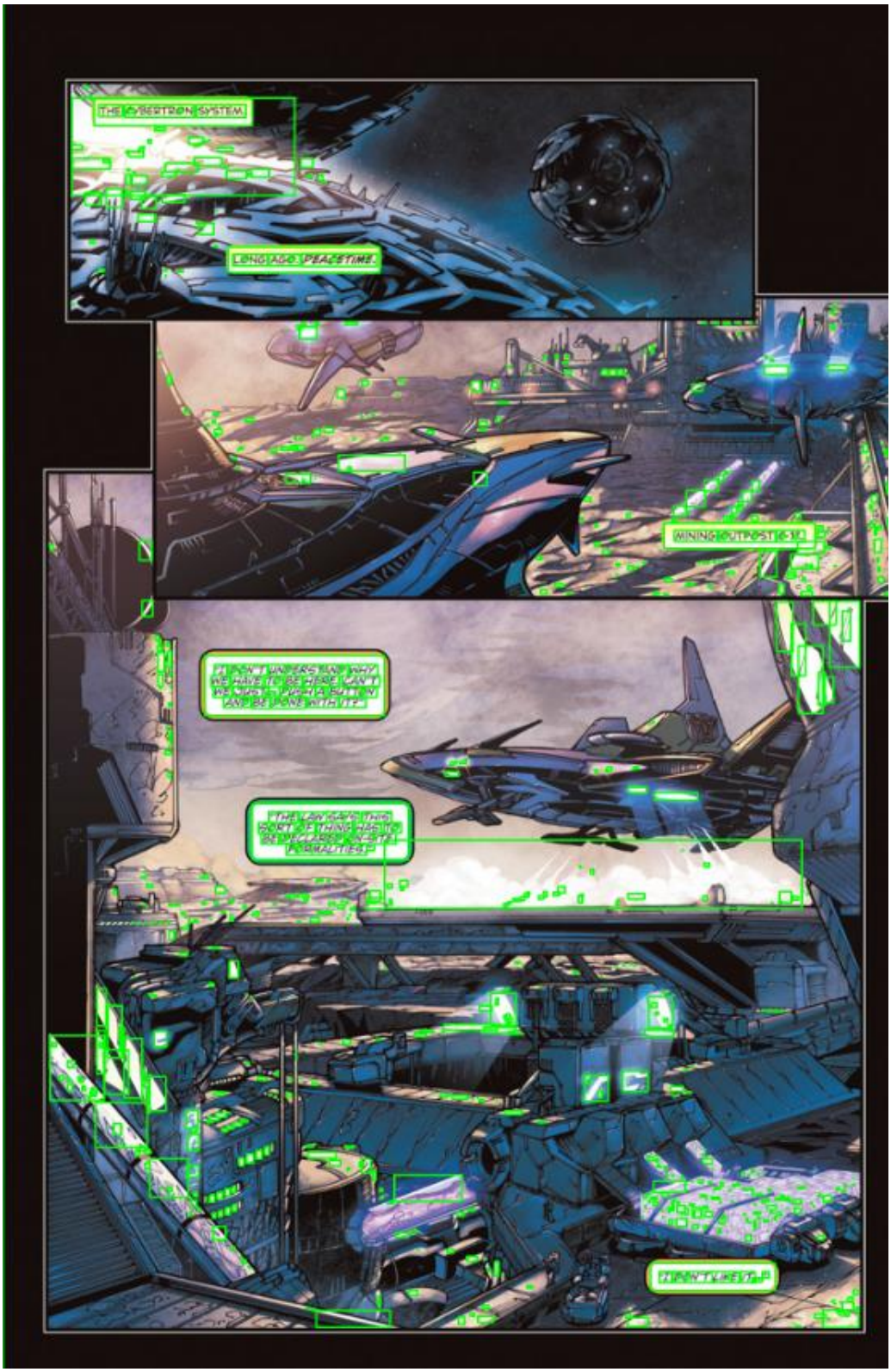

Рис. 5: Стр. 1 - Transformers: Megatron Origin #1 с кандидатами выносок (зеленым цветом) перед фильтрацией.

К счастью, выноски имеют небольшой диапазон размеров. Таким образом, мы можем отфильтровать кандидатов, которые вряд ли будут выносками, потому что они либо слишком большие, либо слишком маленькие, как показано ниже.

Рис. 6: Стр. 1 - Transformers: Megatron Origin #1 с кандидатами выносок (зеленым цветом) после фильтрации

Чтобы улучшить определение контура, перед выполнением описанных выше шагов, мы преобразовываем изображение в черно-белую используя стандартные функции конвертации из библиотеки, отфильтровываем шумы и добавляем некоторые фильтры, чтобы сделать края более острыми.

Метод 2: OpenCV Advanced

С начала изображение проходит предобработку с помощью алгоритма адаптивного порога. В отличии от метода обычного порога, где пороговое значение статическое для всего изображения, при алгоритме адаптивного порога, пороговое значение вычисляется для небольших областей изображения отдельно. После чего пороговое значение для изображения вычисляется с помощью, взвешенной сумму значений окрестностей. Данный подход хорошо подходит, когда фрагменты изображения имеют разные уровни освещенности.

Дальше изображение подвергается эрозии. Основная идея эрозии похожа на эрозию почвы, она размывает границы объекта переднего плана. В зависимости от размера ядра, все пиксели, находящиеся вблизи границы, отбрасываются. Таким образом, толщина или размер объекта переднего плана уменьшается или просто уменьшается белая область на изображении. Это полезно для удаления небольших белых шумов, отсоединения двух соединенных объектов и т. д.

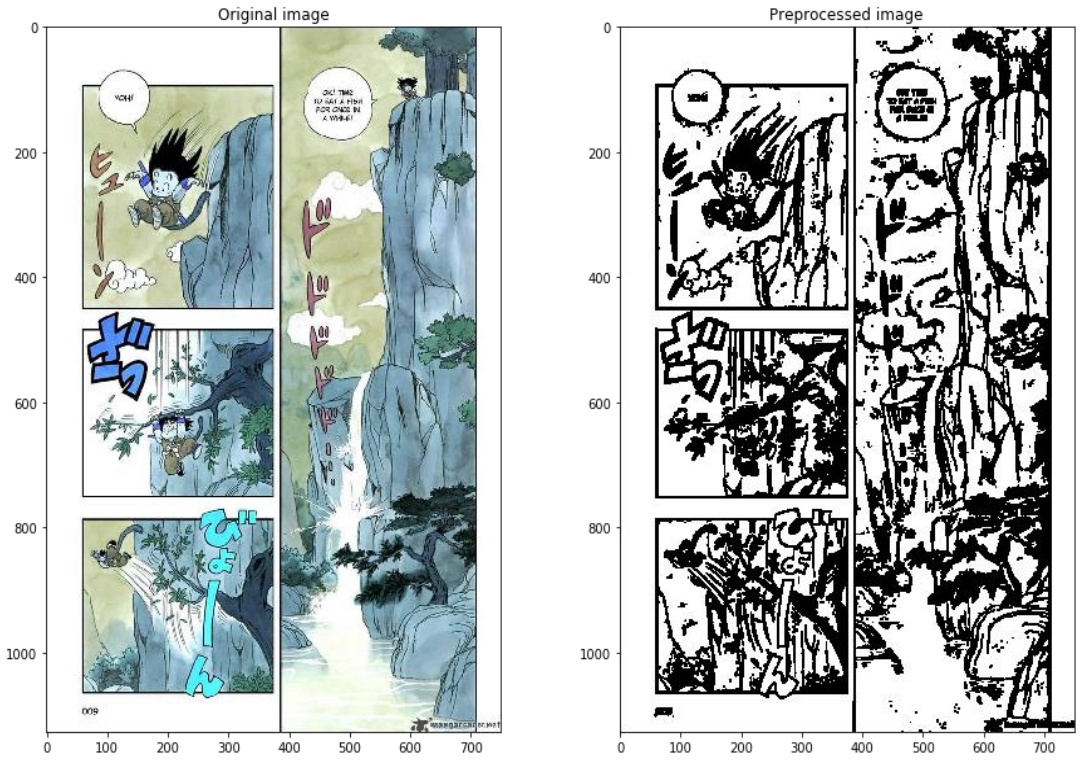

Рис. 7: Пример предобработки изображения

Грубая оценка пограничных боксов с помощью метода связанных компонентов содержит зашумленную информацию, однако всегда включает во все выноски/текстовые боксы, поэтому мы используем ее для получения кандидатов для выносок.

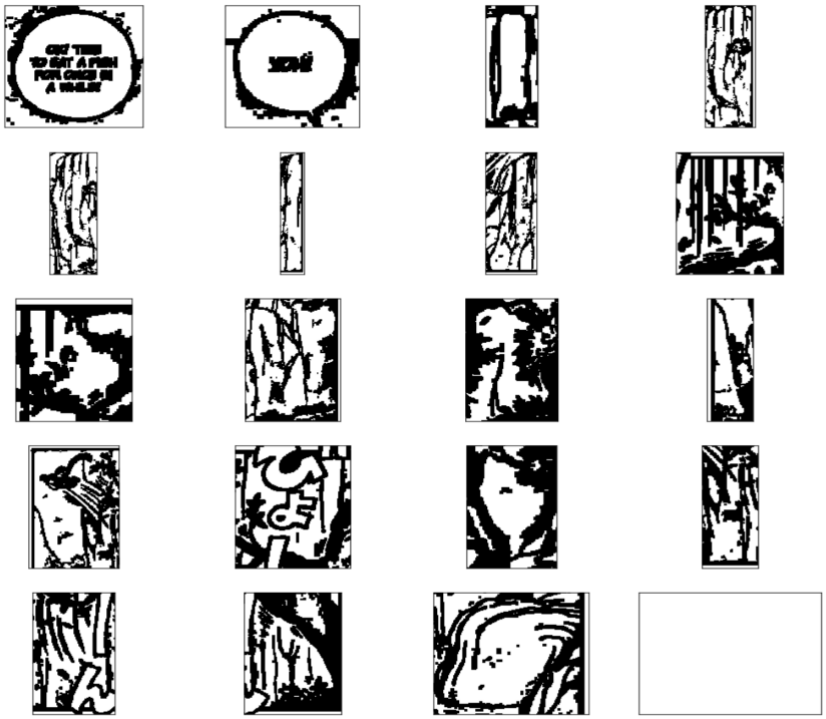

Рис. 8: Пример кандидатов для выносок

Дальше для каждого из кандидатов применяться метод findContours() (см. выше). Каждый из контуров фильтруется, проводиться уточнение оценок границ и отказ от избыточных/неподходящих контуров.

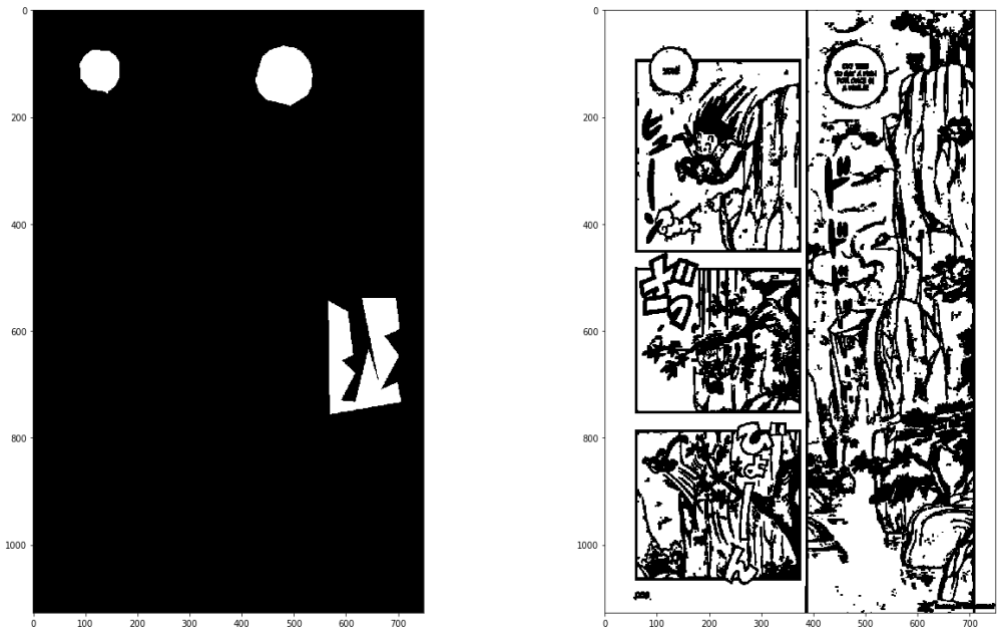

Рис. 9: Пример отфильтрованных контуров

Проведение экспериментов и сбор результатов

В ходе эксперимента были протестированы вышеперечисленные методы на базе выбранного датасета. Были построены графики “Precision x Recall curve” и вычислены “Average Precision” [6].

Кривая “Precision x Recall” является хорошим способом оценивания работы детектора объектов, так как confidence изменяется путем построения кривой для каждого класса объектов. Объектный детектор определенного класса считается хорошим, если его точность остается высокой по мере увеличения recall, что означает, что, если вы изменяете порог confidence, the precision and recall все равно будут высокими. Другой способ определить хороший детектор объектов - искать детектор, который может идентифицировать только релевантные объекты (0 False Positives = high precision), обнаруживая все истинные объекты (0 False Negatives = high recall).

Плохой детектор объектов должен увеличить количество обнаруженных объектов (increasing False Positives = lower precision), чтобы найти все истинные объекты (high recall). Вот почему кривая "Precision x Recall" обычно начинается с высоких значений точности, уменьшаясь с увеличением вызова.

Другой способ сравнить производительность детекторов объектов — это вычислить площадь под кривой (AUC) кривой "Precision x Recall ". Так как AP кривые часто являются зигзагообразными кривыми, идущими вверх и вниз, сравнение различных кривых (разных детекторов) на одном и том же участке обычно не является легкой задачей - потому что кривые имеют тенденцию пересекаться друг с другом очень часто. Вот почему Average Precision (AP), числовая метрика, также может помочь нам сравнить различные детекторы. На практике AP — это точность, усредненная по всем значениям вызова в диапазоне от 0 до 1.

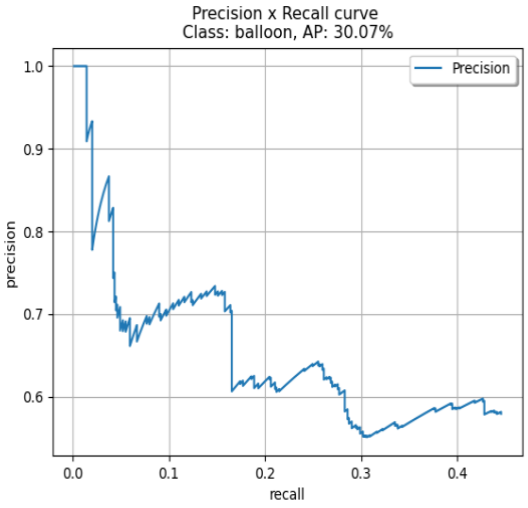

Рис. 10: Кривая “Precision x Recall” метода на основе CNN

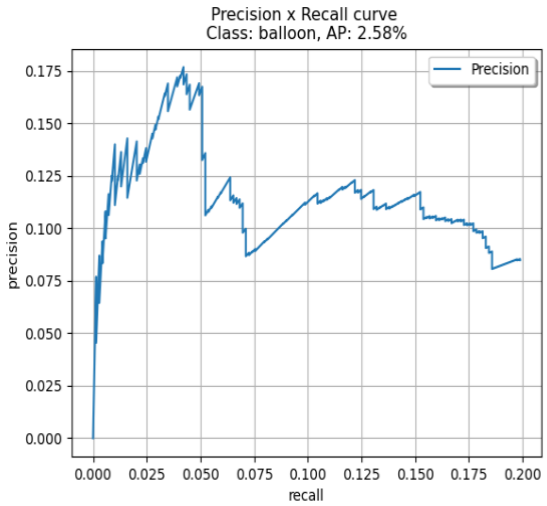

Рис. 11: Кривая “Precision x Recall” метода OpenCV Basic

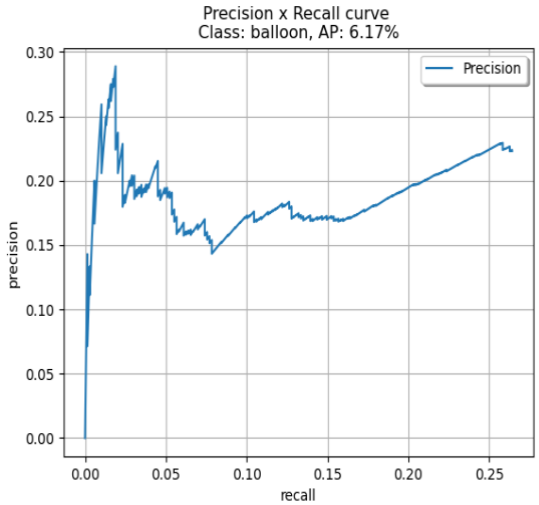

Рис. 12: Кривая “Precision x Recall” метода OpenCV Advanced

Название метода	Average Precision (AP)
Метод на основе CNN	30.07%
Метод OpenCV Basic	2.58%
Метод OpenCV Advanced	6.17%

Табл. 1: Таблица с результатами эксперимента по автоматическому обнаружению выносок

Подведение итогов

Как можно увидеть наихудшим из методов является OpenCV Basic, который очень плохо справляется с работой. Наиболее эффективным является метод на основе CNN. Метод OpenCV Advanced, выполняет работу намного лучше, чем OpenCV Basic, но намного уступает методу на основе CNN.

Извлечения тестов из выносок с помощью OCR

Технология оптического распознавания символов (OCR) является решением для автоматизации извлечения данных из распечатанного или написанного текста с отсканированного документа или файла изображения с последующим преобразованием текста в машиночитаемую форму.

В качестве входных данных для выполнения OCR выступают фрагменты изображений содержащие выноски. Сами выноски хранятся в датасете как координаты точек многоугольника (см. рис. 14). Для каждой выноски вычисляется минимальный прямоугольник, который включает в себе выноску.

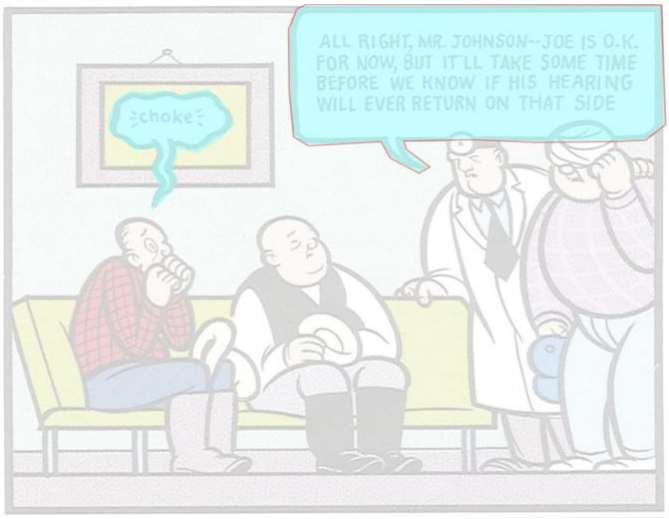

Рис. 13: Фрагмент страницы комикса с выноской (точки многоугольника соединены между собой линиями красного цвета, а сам многоугольник синим)

Рис. 14: Сгенерированное изображение из выноски

Из фрагмента изображения пиксели, которые находятся в заданном прямоугольнике, формулируют новое изображение (см. рис. 14). Далее данное изображение отдается в качестве входного параметра для OCR решения, которое возвращает найденный текст в качестве выходных данных.

*"**ALL RIGHT, MR. JOHNSON-- JOE IS O.K. FOR NOW, BuT ITLL TAKC SOME TIMC DEFORE WE INOW IF HIS HEARING IILL CVCR RETURN ON THAT SIDE**"*

Как можно увидеть, сам текст не совсем правильный для входного изображения. Легко заметить, что программа перепутала букву E с C или B с D. Сами OCR решения не идеальны и у каждого фреймворка, существуют свои ограничения и рекомендованные стандарты изображении.

В данной работе были использованы следующие библиотеки и фреймворки:

1. Cuneiform [13]
2. Tesseract [14]
3. EasyOCR [15]

Для каждого из сгенерированных изображении были вычислены соответствующие выходные ответы при применении OCR решения. В качестве метрики была использована метрика fWER. Для получения более достоверных значений тексты подвергаются обработке (нет заглавных букв, удаляются пустые строки, пробелы и другие специальные символы).

WER [16] - это расстояние Левенштейна на словесном уровне, которое представляет собой минимальное количество замен (S), удалений (D) и вставок (I) необходимых для превращения одной последовательности символов в другую.

$$WER = \frac{I + S + D}{N}$$

Есть два способа интерпретации WER над набором данных.

Один из способов - взять среднее значение WER, обозначаемое как gWER для каждой точки данных:

$$g_{WER} = \mathbb{E}\left[\frac{I + S + D}{N}\right]$$

которая сильно искажена из-за ошибки для коротких выражений.

Другой способ, обозначаемый как fWER, состоит в том, чтобы просуммировать ошибки всех точек, а затем нормализовать их по сумме длин последовательностей.

$$f_{WER} = \frac{\sum_i (I + S + D)_i}{\sum_j N_j} = \frac{\mathbb{E}\left[I + S + D\right]}{\mathbb{E}\left[N\right]}$$

Это устраняет сильную зависимость ошибки от коротких выражений. Кроме того, она может быть интерпретирована как WER одной очень большой последовательности.

	Tesseract	EasyOCR	Cuneiform
WER	15.12	13.52	40.72

Табл. 2: Таблица с результатами эксперимента по извлечению текста из выносок с помощью OCR

Подведение итогов

Как можно увидеть на таблице выше, Cuneiform резко отличается от двух других фреймворков. Сам Cuneiform довольно старый и часто не используется, в то время как Tesseract и EasyOCR являются лидерами в сфере OCR. Можно заметить, что самым оптимальным из решений с точки зрения точности с точки зрения точности является EasyOCR, который побеждает Tesseract с очень малым отрывом.

Методы автоматического перевода текста

Следующим шагом после получения фрагментов текста является перевод с английского языка на русский язык. В качестве инструментов перевода были выбраны следующие онлайн сервисы:

1. Google [17]
2. Yandex [18]
3. Bing [19]
4. Deepl [20]

Были выбрани 100 предложений с разной длиной и сложностью. Для каждого предложения и сервиса перевода были получены переведенные предложения. В качестве метрики для оценивания была использована следующая шкала (см. рис. 15) [1]:

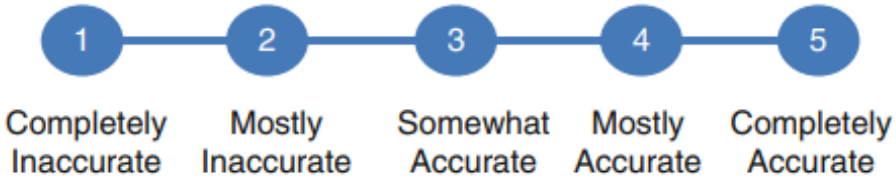

Рис. 15: Шкала оценки переведённого текста

Каждый из переведнных франментов будет классифицирован к одному из классов шкалы (Completely Accurate, …). После этого каждый класс получить соответствующий балл из шкалы.

После проведения эксперимента были получены следующие результаты:

	Google	Yandex	Bing	Deepl
Сред. значение	3.89	4.81	4.31	4.47
Сред. отклонение	0.7506	0.4191	0.7745	0.7447

Табл. 3: Таблица с результатами эксперимента по автоматическому переводу текста

Рассмотрим на примере один из предложений.

Оригинальное предложение: “THE DOOR OPENED AND THEY WERE GREETED BY NONE OTHER THAN DOUGAL...”

Сервис	Перевод
Google	ДВЕРЬ ОТКРЫЛАСЬ, И НИКОГДА ИХ НЕ ПРИВЕТСТВОВАЛ, КРОМЕ ДУГАЛА ...
Yandex	ДВЕРЬ ОТКРЫЛАСЬ, И ИХ ВСТРЕТИЛ НЕ КТО ИНОЙ, КАК ДУГАЛ...
Bing	ДВЕРЬ ОТКРЫЛАСЬ, И ОНИ БЫЛИ ВСТРЕЧЕНЫ НИКТО ИНОЙ, КАК ДУГАЛ ...
Deepl	ДВЕРЬ ОТКРЫЛАСЬ, И ИХ ПРИВЕТСТВОВАЛ НЕ КТО ИНОЙ, КАК ДУГАЛ...

Табл. 4: Таблица с переводами примерного предложения

Можно увидеть, что только у Yandex и Deepl получились предложения, которые вполне соответствуют оригиналу и могут легко конкурировать с аналогичными ручными вариантами переводов. Можно увидеть, что у Bing получилось слегка хуже и если поменять “НИКТО ИНОЙ”, на “НИКЕМ ИНЫМ”, то предложение станет более грамотным. Перевод от Google является самым неточным из них и полностью меняет смысл предложения.

.

Подведение итогов

Как можно увидеть на таблице выше, лучшим из сервисов является Yandex, у которого, среднее значение лишь слегка отстает от наивысшего (5) значения. По сравнению с ним Google показывает наихудший результат. Bing и Deepl находятся почти на одном уровне и последний лишь слегка опережает другого.

Вставка переведенного текста вместо оригинального

Последним шагом для получения готового переведенного комикса является вставка переведенного фрагмента вместо оригинала. Для начала нужно найти участок выноски (см. рис. 14), где нужно вставить текст. Один из возможных решений является поиск наибольшего прямоугольника, которого охватывает контур выноски (см. рис. 16). После чего нужно узнать цвет фона. Самым простым вариантом будет, поиск всевозможных цветов и выбор наиболее часто встречаемого. Также можно использовать методы OCR решений для поиска расположении символов и выбрать средний цвет, который не охватывает символы.

Рис. 16: Сгенерированное изображение из контура выноски

После выбора фона участок с найденным прямоугольником окрашивается в соответствующий цвет (см. рис. 17). Перед покраской можно использовать отступы.

Рис. 17: Участок выноски после окраски

Как можно увидеть, сам оригинальный текст пропал, а модифицированное изображение имеет чистый участок для вставки. Дальше нужно определить механизмы вставки переведенного текста.

Рассмотрим на примере один из предложений.

Переведенное предложение: **“ХОРОШО, МИСТЕР ДЖОНСОН, ПОКА ДЖО В ПОРЯДКЕ, НО ПРОЙДЕТ НЕКОТОРОЕ ВРЕМЯ, ПРЕЖДЕ ЧЕМ МЫ УЗНАЕМ, ВЕРНЕТСЯ ЛИ КОГДА-НИБУДЬ ЕГО СЛУХ С ЭТОЙ СТОРОНЫ”**

С начала нужно разделить предложение на несколько строк, так как при вставке в одну строку, шрифт будет слишком маленьким, если кончено его правильно посчитать, а при

отсутствии размера шрифта, может быть использовано значение по умолчанию и текст может выйти за границы изображение.

Чтобы избежать таких проблем, нужно посчитать размер подтекстов при выбранном шрифте и размере и выровнять их так, чтобы полученный итоговый размер предложения помешался в прямоугольник. Также можно использовать отступы или добавить коэффициент для размера шрифта.

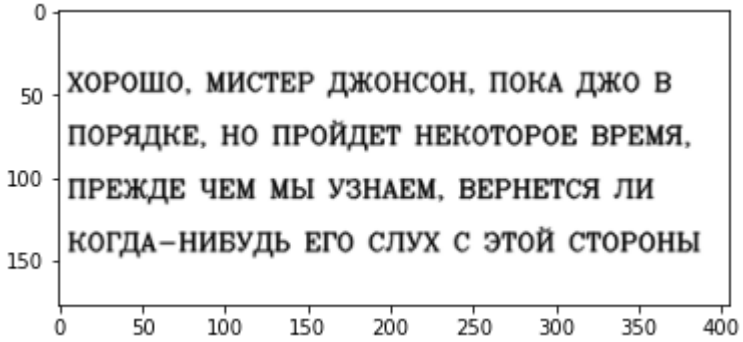

Рис. 18: Выровненный текст для выбранного прямоугольника

После предобработки с помощью уже выбранных коэффициентов нужно ставить в модифицированное изображение обработанное переведенное изображение.

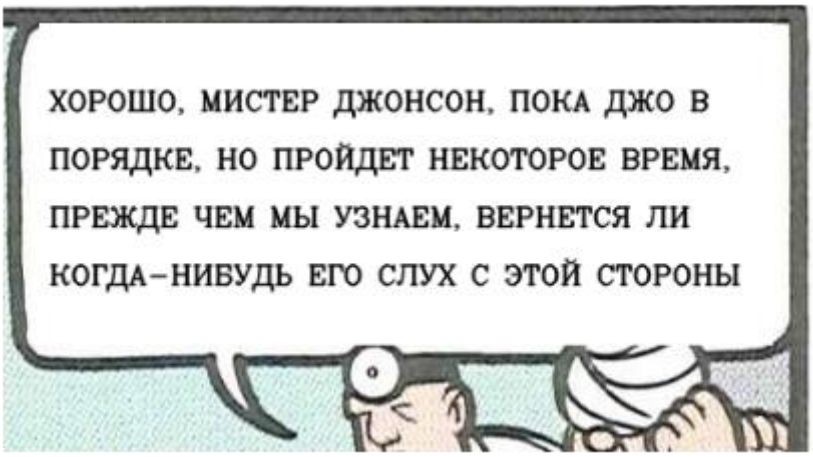

Рис. 19: Изображение после вставки переведенного текста

Подведение итогов

Как можно увидеть в результате мы получили переведенный фрагмент из комикса, которого сложно отличить от оригинала. На самом деле можно заметить, что часть повязки пациента пропала. При автоматическом поиске участка для вставки могут быть недочеты, также может быть неправильно выбран цвет фона или участок прямоугольника.

Заключение

В итоге были исследованы методы и алгоритмы для автоматического поиска выносок, извлечение текста из выносок их автоматических перевод и вставка вместо оригинала. Были выявлены проблемы каждого из этапов. Хотя сама процедура автоматического перевода англоязычных комиксов сложная и требует много доработок, уже можно увидеть некоторый результат.

Приведенные методы могут помочь переводчикам для более быстрого перевода комиксов, также они могут быть использованы для составления нового формата хранения комиксов, где вместо изображения, комиксы можно хранить в формате SVG или в другом древовидном формате. Также можно индексировать комиксы, иметь возможность полноценного поиска. Также данные методы могут быть использованы для макетов, постеров или других графических контентов при некоторой модификации.

В качестве будущих доработок может понадобиться сделать следующие шаги:

1. Создать собственные модели для распознавания выносок;
2. Исследовать методы повышения качества обработки комиксов путем предварительной обработки изображения и текстов;
3. Использовать другие метрики и методы проведения экспериментов

Список источников

1. Castilho S. [и др.]. Approaches to Human and Machine Translation Quality Assessment под ред. J. Moorkens [и др.]., Cham: Springer International Publishing, 2018.С. 9–38.
2. Dubray D., Laubrock J. Deep CNN-based Speech Balloon Detection and Segmentation for Comic Books.
3. Dunst A., Hartel R., Laubrock J. The Graphic Narrative Corpus (GNC): Design, Annotation, and Analysis for the Digital Humanities 2017.С. 15–20.
4. Ho A. K. N., Burie J., Ogier J. Panel and Speech Balloon Extraction from Comic Books 2012.С. 424–428.
5. Kang S., Choo J., Chang J. Consistent Comic Colorization with Pixel-wise Background Classification.
6. Padilla R. [и др.]. A Comparative Analysis of Object Detection Metrics with a Companion Open-Source Toolkit // Electronics. 2021. № 3 (10).
7. Rigaud C. [и др.]. An Active Contour Model for Speech Balloon Detection in Comics 2013.С. 1240–1244.
8. Digital comic - Wikipedia [Электронный ресурс]. URL: https://en.wikipedia.org/wiki/Digital_comic (дата обращения: 14.11.2020).
9. Optical character recognition - Wikipedia [Электронный ресурс]. URL: https://en.wikipedia.org/wiki/Optical_character_recognition (дата обращения: 14.11.2020).
10. Manga109 [Электронный ресурс]. URL: http://www.manga109.org/en/ (дата обращения: 17.04.2021).
11. Database - eBDtheque [Электронный ресурс]. URL: http://ebdtheque.univ-lr.fr/database/ (дата обращения: 17.04.2021).
12. OpenCV [Электронный ресурс]. URL: https://opencv.org/ (дата обращения: 12.11.2020).
13. Cuneiform for Linux in Launchpad [Электронный ресурс]. URL: https://launchpad.net/cuneiform-linux (дата обращения: 17.04.2021).
14. tesseract-ocr/tesseract: Tesseract Open Source OCR Engine (main repository) [Электронный ресурс]. URL: https://github.com/tesseract-ocr/tesseract (дата обращения: 12.11.2020).
15. JaidedAI/EasyOCR: Ready-to-use OCR with 80+ supported languages and all popular writing scripts including Latin, Chinese, Arabic, Devanagari, Cyrillic and etc. [Электронный ресурс]. URL: https://github.com/JaidedAI/EasyOCR (дата обращения: 17.04.2021).
16. Word Error Rate Mechanism, ASR Transcription and Challenges in Accuracy Measurement [Электронный ресурс]. URL: https://www.gmrtranscription.com/blog/word-error-rate-mechanism-asr-transcription-and-challenges-in-accuracy-measurement (дата обращения: 17.04.2021).
17. Google Translate [Электронный ресурс]. URL: https://translate.google.com/ (дата обращения: 17.04.2021).
18. Yandex.Translate – dictionary and online translation between English and over 90 other languages. [Электронный ресурс]. URL: https://translate.yandex.com/ (дата обращения: 17.04.2021).
19. Bing Microsoft Translator [Электронный ресурс]. URL: https://www.bing.com/translator (дата обращения: 17.04.2021).
20. DeepL Translate [Электронный ресурс]. URL: https://www.deepl.com/translator (дата обращения: 17.04.2021).