



Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Московский государственный технический университет  
имени Н.Э. Баумана  
(национальный исследовательский университет)»  
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ ИНФОРМАТИКА И СИСТЕМЫ УПРАВЛЕНИЯ

КАФЕДРА СИСТЕМЫ ОБРАБОТКИ ИНФОРМАЦИИ И УПРАВЛЕНИЯ

# РАСЧЕТНО-ПОЯСНИТЕЛЬНАЯ ЗАПИСКА К НАУЧНО-ИССЛЕДОВАТЕЛЬСКОЙ РАБОТЕ

**НА ТЕМУ:**

**ОРС СИСТЕМА**

**ДЛЯ ИЗОБРАЖЕНИЯ ТЕКСТОВЫХ ЛОГОВ**

Студент ИУ5-31М  
(Группа)

(Подпись, дата)

М.Р. Камалов  
(И.О.Фамилия)

Руководитель

(Подпись, дата)

Ю.Е. Гапанюк  
(И.О.Фамилия)

2023 г.

Министерство науки и высшего образования Российской Федерации  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Московский государственный технический университет имени Н.Э. Баумана  
(национальный исследовательский университет)»  
(МГТУ им. Н.Э. Баумана)

УТВЕРЖДАЮ  
Заведующий кафедрой ИУ5  
(Индекс)  
В.И. Терехов  
(И.О.Фамилия)  
« 04 » сентября 2023 г.

**ЗАДАНИЕ**  
**на выполнение научно-исследовательской работы**

по теме ORC система для изображения текстовых логов

Студент группы ИУ5-31М

Камалов Марат Равилевич  
(Фамилия, имя, отчество)

Направленность НИР (учебная, исследовательская, практическая, производственная, др.)  
ИССЛЕДОВАТЕЛЬСКАЯ

Источник тематики (кафедра, предприятие, НИР) КАФЕДРА

График выполнения НИР: 25% к 4 нед., 50% к 8 нед., 75% к 12 нед., 100% к 16 нед.

**Техническое задание** Исследовать существующие подходы к оптическому распознаванию символов, выбрать наилучший подход с учетом специфики предметной области и особенностей распознаваемых символов и реализовать этот подход программными методами

**Оформление научно-исследовательской работы:**

Расчетно-пояснительная записка на 14 листах формата А4.

Перечень графического (иллюстративного) материала (чертежи, плакаты, слайды и т.п.)

Дата выдачи задания « 04 » сентября 2023 г.

Руководитель НИР

Ю.Е. Гапанюк  
(Подпись, дата) (И.О.Фамилия)

Студент

М.Р. Камалов  
(Подпись, дата) (И.О.Фамилия)

Примечание: Задание оформляется в двух экземплярах: один выдается студенту, второй хранится на кафедре.

## Оглавление

Введение .....	4
Основная часть: описание процесса разработки подсистемы распознавания текста .....	5
Заключение .....	14
Список использованных источников .....	14

## **Введение**

Для накопления статистики по прошедшим событиям и оповещения о наступлении критических событий из заданного перечня системе необходимо получать информацию из журнала событий. Данный журнал представлен на экране пользователя в виде изображения разноцветного текста на неоднородном, постоянно меняющемся фоне. Видеоигра ARK: Survival Evolved не предоставляет интерфейса для получения данных из журнала событий. Идея перехватывать пакеты, приходящие от сервера и содержащие информацию о событиях, является несостоятельной, поскольку такой подход значительно увеличит сложность клиентской части программного продукта. Извлечение данных из процессов игры обладает недостатком предыдущего подхода, а помимо этого добавляется риск блокировки со стороны анти-чита. Учитывая все вышесказанное, мною было выбрано оптическое распознавание символов с изображения экрана пользователя. Данный подход наилучшим образом «облегчает» клиентскую часть системы, а также минимизирует риски для пользователя.

Целью научно-исследовательской работы является создание подсистемы, преобразующей изображение текстовых логов в данные строкового типа.

В ходе выполнения работы должны быть выполнены следующие задачи:

- исследование существующих подходов по оптическому распознаванию символов;
- выбор наилучшего подхода с учетом специфики предметной области и особенностей распознаваемых символов;
- реализация наилучшего подхода программными методами.

## Основная часть: описание процесса разработки подсистемы распознавания текста.

Изображение текстовых логов, которое и необходимо преобразовать в строковые данные представлено на рисунке 1.



The image shows a screenshot of a game log with a dark blue background and white text. The log entries are timestamped and describe various in-game events, including taming a Triceratops, triggering events by survivors and dinosaurs, removing and adding Watchlogs, setting Rank Groups, and claiming/unclaiming a Parasaur. The text is color-coded for different types of events and entities.

```
Day 2, 20:06:12: Your Tribe Tamed a Triceratops - Lvl 3 (Triceratops)!
Day 2, 18:58:46: abcdefghijklmnop triggered by a friendly survivor.
Day 2, 18:58:13: ABCDEFGHIJKLMNOPQ triggered by a friendly survivor.
Day 2, 17:34:51: WATCHLOG was removed from the Tribe by Human!
Day 2, 17:33:40: WATCHLOG set to Rank Group Admin!
Day 2, 17:32:42: WATCHLOG was demoted from Tribe Admin by Human!
Day 2, 17:32:01: WATCHLOG set to Rank Group Admin!
Day 2, 17:26:02: Human set to Rank Group Admin!
Day 2, 17:24:50: WATCHLOG set to Rank Group Admin!
Day 2, 17:24:30: Human set to Rank Group Admin!
Day 2, 17:23:16: WATCHLOG was promoted to a Tribe Admin by Human!
Day 2, 17:22:04: WATCHLOG set to Rank Group Admin!
Day 2, 17:20:52: WATCHLOG was added to the Tribe by Human!
Day 2, 17:15:39: ABCDEFGHIJKLMNOPQ triggered by an enemy survivor.
Day 2, 16:33:13: abcdefghijklmnop triggered by an enemy survivor.
Day 2, 16:31:17: abcdefghijklmnop triggered by by an enemy dino.
Day 2, 16:31:15: Human unclaimed 'Parasaur - Lvl 13 (Parasaur)'!
Day 2, 16:27:39: abcdefghijklmnop triggered by by a friendly dino.
Day 2, 16:27:37: abcdefghijklmnop triggered by a friendly survivor.
Day 2, 16:26:24: abcdefghijklmnop triggered by a friendly survivor.
Day 2, 16:25:45: abcdefghijklmnop triggered by by a friendly dino.
Day 2, 16:25:45: ABCDEFGHIJKLMNOPQ triggered by by a friendly dino.
Day 2, 16:25:45: Human claimed 'Parasaur - Lvl 13 (Parasaur)'!
Day 2, 16:03:51: Tek Sensor triggered by by an enemy dino.
Day 2, 15:39:25: ABCDEFGHIJKLMNOPQ triggered by by an enemy dino.
Day 2, 15:39:25: Human unclaimed 'Parasaur - Lvl 13 (Parasaur)'!
Day 2, 15:35:37: ABCDEFGHIJKLMNOPQ triggered by
```

Рисунок 1 – Изображение текстовых логов

Первым делом необходимо преобразовать изображение из цветного в черно-белое. Текст при этом должен быть однородно черным, а фон однородно белым. Библиотека `opencv` не предоставляет готовых функций, для выполнения подобного преобразования, поэтому необходимо написать свой алгоритм.

Суть алгоритма заключается в нахождении для каждого цвета, в который окрашен текст, таких граничных координат цвета (Hue, Saturation, Value — тон, насыщенность, значение цвета — координаты цвета в цветовой модели HSV), чтобы получить маску данного конкретного цвета. После чего маска вычитается из белого фона, и как результат получается черный текст на белом фоне.

Для каждого из 14 цветов, в которые может быть окрашен текст, опытным путем были установлены свои значения тона, насыщенности, значения цвета. Пример нахождения соответствующих значений представлен на рисунке 2.

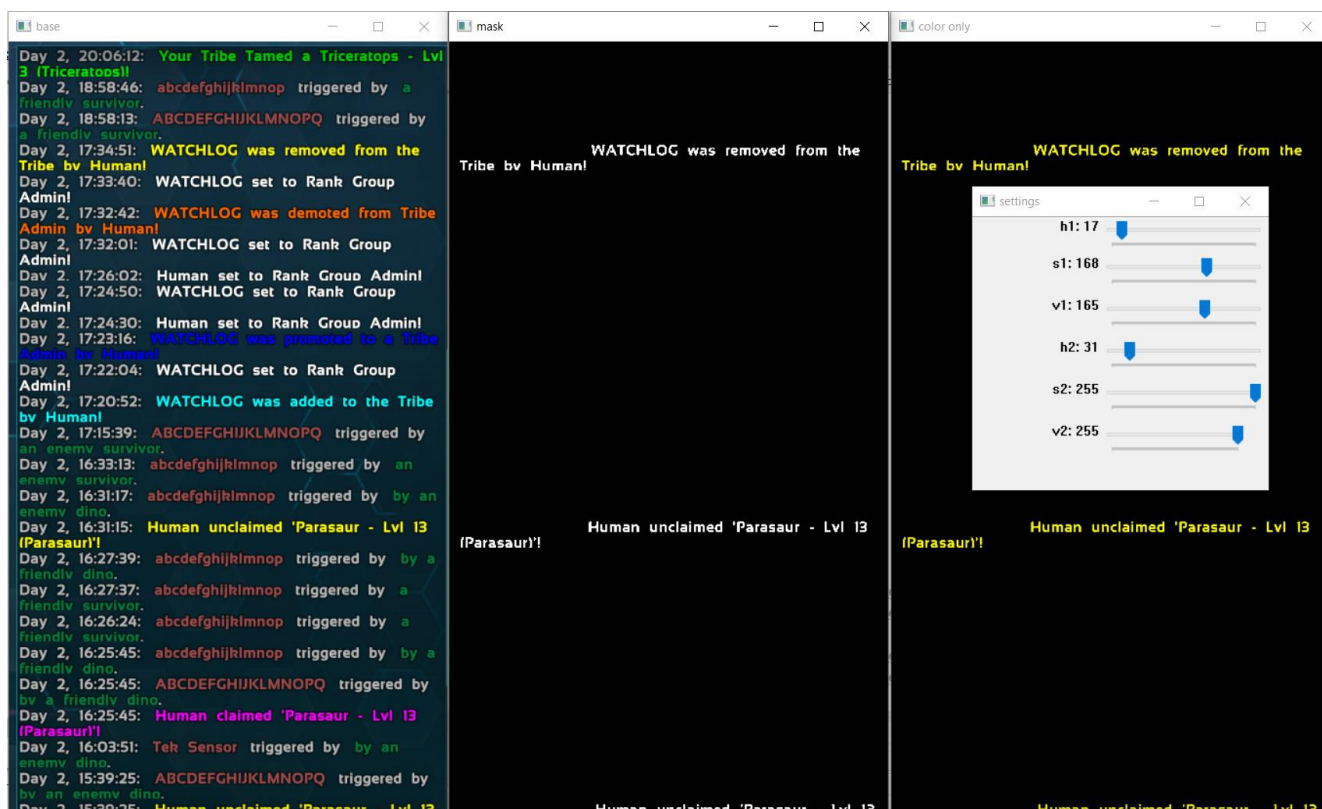


Рисунок 2 – Нахождение значений HSV для жёлтого цвета

Значения тона, насыщенности и значения цвета для всех 14 цветов, в которые может быть окрашен текст, представлены в таблице 1.

Таблица 1 – Значения тона, насыщенности и значения цвета для цветов, в которые может быть окрашен текст

<b>Цвет</b>	<b>Нижняя граница тона</b>	<b>Нижняя граница насыщенности</b>	<b>Нижняя граница значения цвета</b>	<b>Верхняя граница тона</b>	<b>Верхняя граница насыщенности</b>	<b>Верхняя граница значения цвета</b>
Чёрный	10	0	0	250	255	30
Красный	0	150	160	10	255	255
Зеленый	45	150	150	65	255	255
Серый	0	0	135	255	35	255
Розовый	145	250	170	155	255	255
Оранжевый	0	230	110	15	255	255
Желтый	17	168	165	31	255	255
Темно-зеленый	60	250	105	75	255	150
Кирпичный	0	124	104	65	151	238
Светло-зеленый	153	146	144	255	255	255
Светло-красный	66	108	168	69	255	255
Голубой	64	120	168	144	255	255
Синий	153	146	144	255	255	255
Салатовый	59	149	193	66	255	255

При наложении масок всех цветов из таблицы 1 получается изображение черного текста на белом фоне. Результат преобразования исходного изображения в черно-белое представлен на рисунке 3.

```

Day 2, 20:06:12: Your Tribe Tamed a Triceratops - Lvl
3 (Triceratops)!
Day 2, 18:58:46: abcdefghijklmnop triggered by a
friendly survivor.
Day 2, 18:58:13: ABCDEFGHIJKLMNOPQ triggered by
a friendly survivor.
Day 2, 17:34:51: WATCHLOG was removed from the
Tribe by Human!
Day 2, 17:33:40: WATCHLOG set to Rank Group
Admin!
Day 2, 17:32:42: WATCHLOG was demoted from Tribe
Admin by Human!
Day 2, 17:32:01: WATCHLOG set to Rank Group
Admin!
Day 2, 17:26:02: Human set to Rank Group Admin!
Day 2, 17:24:50: WATCHLOG set to Rank Group
Admin!
Day 2, 17:24:30: Human set to Rank Group Admin!
Day 2, 17:23:16: WATCHLOG was promoted to a Tribe
Admin by Human!
Day 2, 17:22:04: WATCHLOG set to Rank Group
Admin!
Day 2, 17:20:52: WATCHLOG was added to the Tribe
by Human!
Day 2, 17:15:39: ABCDEFGHIJKLMNOPQ triggered by
an enemy survivor.
Day 2, 16:33:13: abcdefghijklmnop triggered by an
enemy survivor.
Day 2, 16:31:17: abcdefghijklmnop triggered by by an
enemy dino.
Day 2, 16:31:15: Human unclaimed 'Parasaur - Lvl 13
(Parasaur)'!
Day 2, 16:27:39: abcdefghijklmnop triggered by by a
friendly dino.
Day 2, 16:27:37: abcdefghijklmnop triggered by a
friendly survivor.
Day 2, 16:26:24: abcdefghijklmnop triggered by a
friendly survivor.
Day 2, 16:25:45: abcdefghijklmnop triggered by by a
friendly dino.
Day 2, 16:25:45: ABCDEFGHIJKLMNOPQ triggered by
by a friendly dino.
Day 2, 16:25:45: Human claimed 'Parasaur - Lvl 13
(Parasaur)'!
Day 2, 16:03:51: Tek Sensor triggered by by an
enemy dino.
Day 2, 15:39:25: ABCDEFGHIJKLMNOPQ triggered by
by an enemy dino.
Day 2, 15:39:25: Human unclaimed 'Parasaur - Lvl 13
(Parasaur)'!
Day 2, 15:35:37: ABCDEFGHIJKLMNOPQ triggered by

```

Рисунок 3 – Преобразованное черно-белое изображение текстовых логов



После этого необходимо создать несколько преобразованных черно-белых изображений, на которых бы содержались все необходимые для дальнейшего обучения символы. Подобранные по вышеуказанному критерию изображения представлены на рисунке 4.



Рисунок 4 – Преобразованные черно-белые изображения

Подобранные изображения упаковываются в многостраничный tiff файл. Далее в этом файле необходимо разметить символы, то есть каждому изображению символа необходимо поставить в соответствие область изображения, в которой этот символ находится, а также непосредственно сам символ в кодировке utf-8. Данная разметка технически реализуется путем создания специального box-файла с использованием утилиты jTessBoxEditor. Фрагмент box-файла с размеченными символами представлен на рисунках 5 и 6.

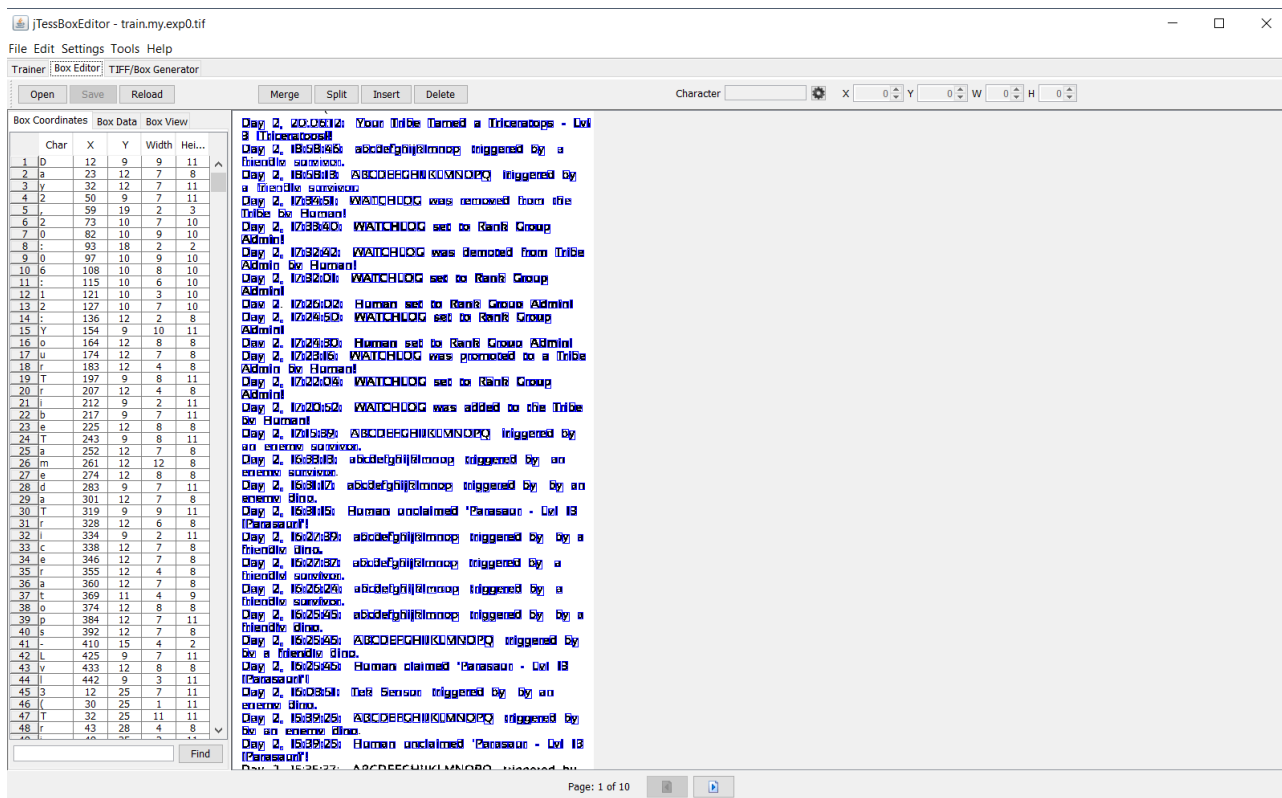


Рисунок 5 – Размеченный бок-файл

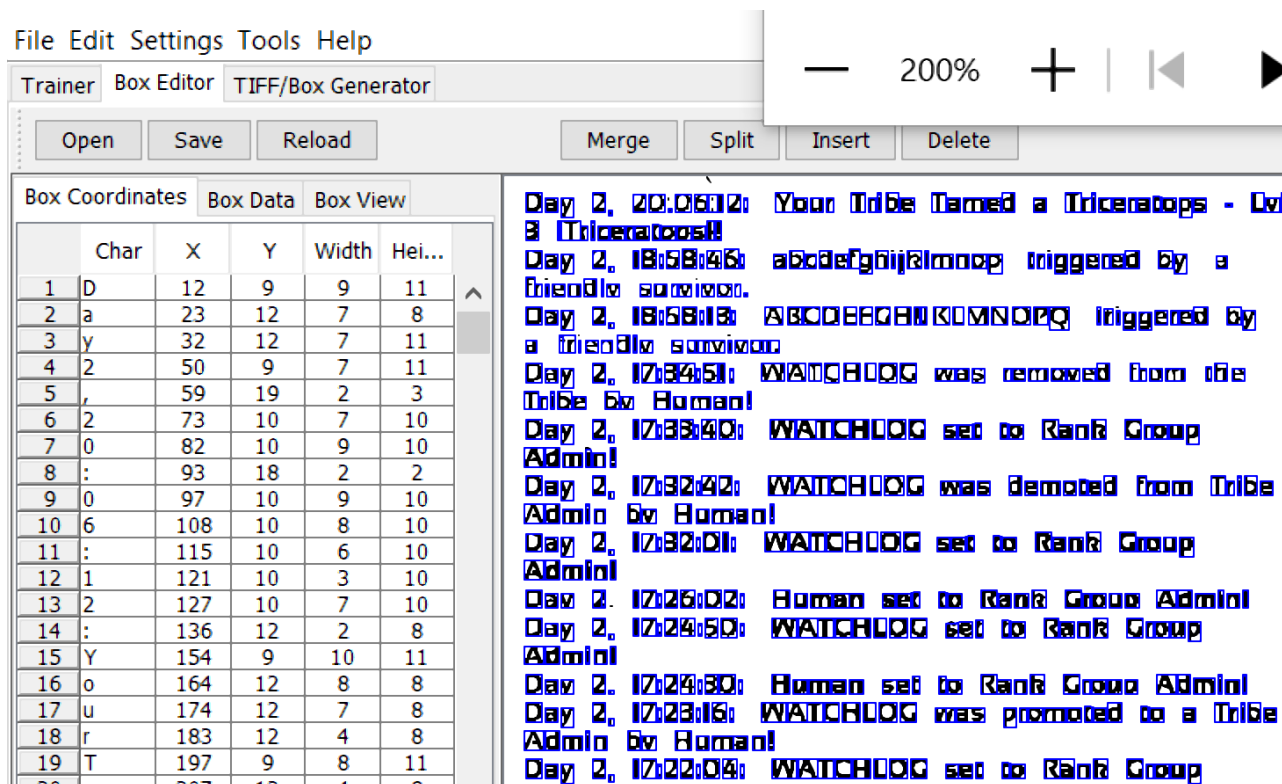


Рисунок 6 – Размеченный бок-файл с приближением в 200%

После разметки символов начинается непосредственно процесс обучения. Обучение проводилось на базе библиотеки Tesseract OCR по следующему алгоритму:

1. Создается .tr файл (данный файл представляет собой соединение файла изображения и box-файла);
2. Извлекается набор символов из box-файла (в результате получается unicharset файл);
3. Создается font\_properties файл (в данном файле описывается стиль шрифта; в нашем случае это echo "Sansation 0 0 1 0 0" > font\_properties);
4. Создаются файлы с кластерами фигур и прототипами;
5. Создаются словари наиболее часто встречающихся слов;
6. Создается файл .traineddata (данный файл представляет собой объединение всех ранее созданных файлов).

Полученный .traineddata файл необходимо поместить в папку tessdata, которая находится в директории Tesseract-OCR. После этого уже можно использовать функцию pytesseract.image\_to\_string библиотеки Tesseract-OCR для преобразования изображения в строковые данные. В качестве параметров данная функция принимает изображение (в нашем случае используется изображение, представленное на рисунке 1), язык (в нашем случае значение параметра «язык» будет равно названию .traineddata файла, который был создан ранее) и конфигурация (данный параметр состоит из двух подпараметров: режим работы механизма распознавания текста и режим сегментации страницы; возможные значения данных подпараметров представлены на рисунках 7 и 8 соответственно; в нашем случае значение подпараметра «режим работы механизма распознавания текста» равно 3, а значение подпараметра «режим сегментации страницы» равно 6).

```
OCR Engine modes:
0    Legacy engine only.
1    Neural nets LSTM engine only.
2    Legacy + LSTM engines.
3    Default, based on what is available.
```

Рисунок 7 – Возможные значения подпараметра «режимы работы механизма распознавания текста»

```
Page segmentation modes:
0    Orientation and script detection (OSD) only.
1    Automatic page segmentation with OSD.
2    Automatic page segmentation, but no OSD, or OCR.
3    Fully automatic page segmentation, but no OSD. (Default)
4    Assume a single column of text of variable sizes.
5    Assume a single uniform block of vertically aligned text.
6    Assume a single uniform block of text.
7    Treat the image as a single text line.
8    Treat the image as a single word.
9    Treat the image as a single word in a circle.
10   Treat the image as a single character.
11   Sparse text. Find as much text as possible in no particular order.
12   Sparse text with OSD.
13   Raw line. Treat the image as a single text line,
    bypassing hacks that are Tesseract-specific.
```

Рисунок 8 – Возможные значения подпараметра «режим сегментации страницы»

Результат выполнения функции `pytesseract.image_to_string` с описанными выше параметрами представлен на рисунке 9.

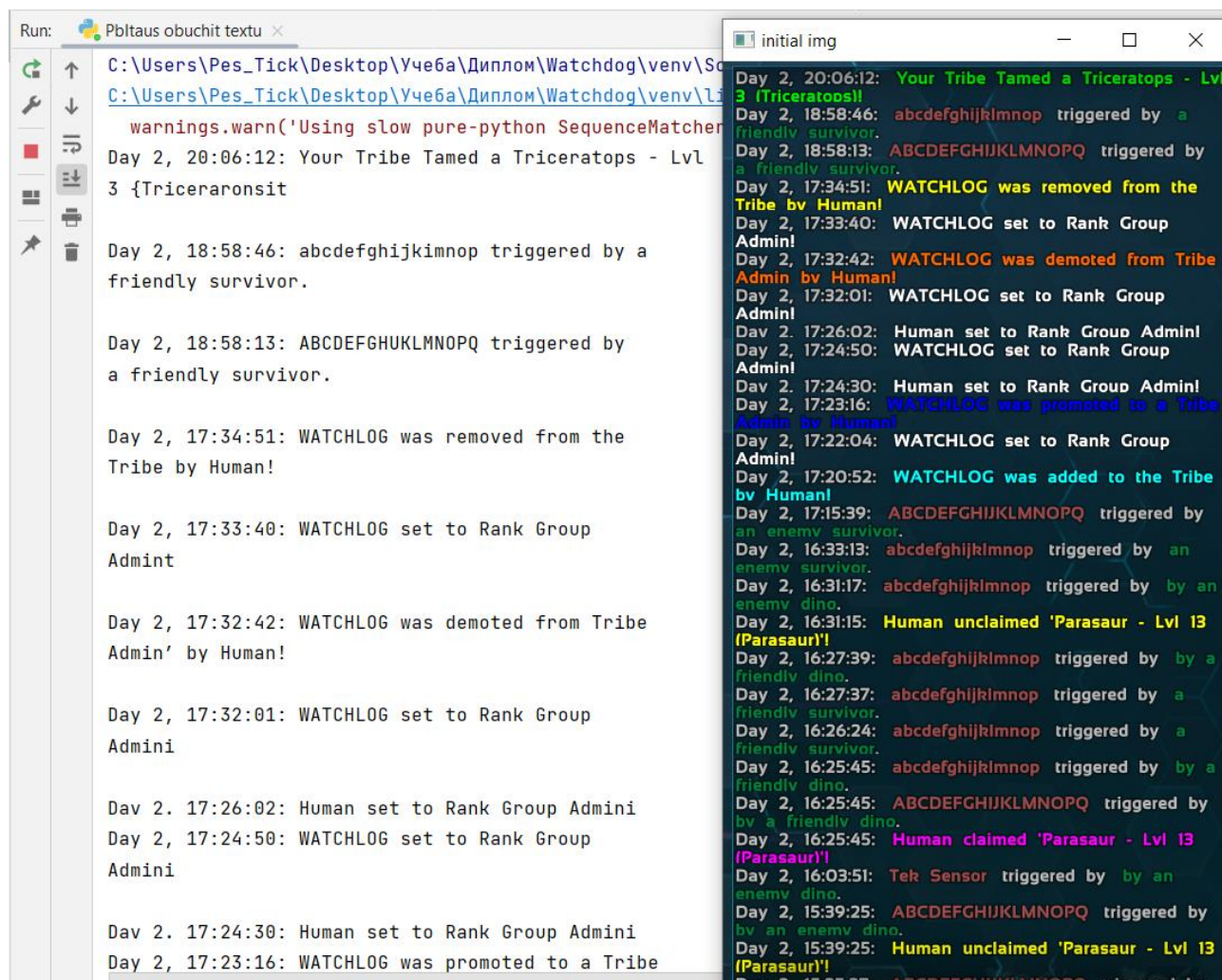


Рисунок 9 – Результат выполнения функции pyesseract.image\_to\_string

Как видно из рисунка 9, в результате выполнения функции pyesseract.image\_to\_string преобразование изображения в текст имеет приемлемое качество. Неточности в распознавании наблюдаются с символами восклицательного знака, скобок, запятой. Также неверно распознаются символы, которые на самом изображении представлены в обрезанном или усеченном виде. Но все эти недостатки достаточно легко могут быть устранены при постобработке полученного текста методами обработки естественного языка.



## **Заключение**

Таким образом, в ходе научно-исследовательской работы была разработана подсистема распознавания текста, которая преобразует изображение текстовых логов, отображаемых на экране пользователя, в данные строкового типа, которые в дальнейшем могут быть использованы для решения прикладных задач.

## **Список использованных источников**

1. Sansation - URL: <https://fonts-online.ru/fonts/sansation> (дата обращения 12.12.2023)
2. Tesseract ORC. - URL: <https://github.com/tesseract-ocr/tesseract> (дата обращения 12.12.2023)
3. Цветовые пространства HSV - URL: <https://learn.microsoft.com/ru-ru/windows/win32/wcs/hsv-color-spaces> (дата обращения 12.12.2023)
4. Обучаем вместе сTesseract ORC - URL: <https://favicon.tech/obuchaem-vmeste-s-tesseract-ocr/> (дата обращения 12.12.2023)
5. OpenCV на python: цветовой фильтр - URL: <https://robotclass.ru/tutorials/opencv-color-range-filter/> (дата обращения 12.12.2023)
6. jTessBoxEditor - URL: <https://vietocr.sourceforge.net/training.html> (дата обращения 12.12.2023)