

Разработка программы для распознавания текста на изображении

Андрienко Иван Сергеевич

Приамурский государственный университет имени Шолом-Алейхема

Студент

Аннотация

В данной статье представлен процесс разработки десктопного приложения для распознавания текста на изображении на языке программирования Python. В качестве инструментов использовались библиотеки PyAutoGUI для взаимодействия с графическим интерфейсом пользователя и pytesseract для работы с Tesseract OCR, обеспечивающего функциональность оптического распознавания символов. Описаны этапы создания приложения, начиная от установки необходимых библиотек и заканчивая реализацией основных функций.

Ключевые слова: Python, десктопное приложение, распознавание текста, PyAutoGUI, pytesseract, Tesseract OCR,

Development of a program for recognizing text in an image

Andrienko Ivan Sergeevich

Sholom-Aleichem Priamursky State University

Student

Abstract

This article presents the process of developing a desktop application for recognizing text on an image in the Python programming language. The tools used were the PyAutoGUI libraries for interacting with the graphical user interface and pytesseract for working with Tesseract OCR, which provides optical character recognition functionality. The steps of creating an application are described, starting from installing the necessary libraries and ending with the implementation of basic functions.

Keywords: Python, desktop application, text recognition, PyAutoGUI, pytesseract, Tesseract OCR,

1 Введение

1.1 Актуальность

В современном мире с развитием технологий распознавание текста на изображениях становится все более востребованным и универсальным инструментом. Разработка десктопного приложения для данной цели на языке программирования Python открывает новые возможности для эффективной обработки информации, извлечения текста и автоматизации рутинных задач. Способность приложения работать с изображениями на разных языках делает

его универсальным инструментом для пользователей в различных областях, включая образование, бизнес и научные исследования. Разработка подобного приложения предоставляет программистам и IT-специалистам возможность расширить свои навыки в области обработки изображений и текстов, а также создать удобный инструмент для повседневного использования.

1.2 Обзор исследований

В своей работе Р.Ш. Минязев, С.А. Дыганов, И.Р. Гумеров и др. рассмотрели процесс автоматизации процесса связанных с документооборотом, использованием технологии OCR, изначально предложенную компанией ABBYY и широко использующуюся в их линейке продуктов. Среди имеющихся программ, реализующих подход OCR, можно выделить бесплатную программную библиотеку Tesseract-OCR с открытым исходным кодом, доступным под лицензией Apache. С ее помощью можно разрабатывать свои системы для распознавания [1]. О.А.Нейфельд, А.С. Рудникович исследовали возможность применения известной библиотеки Tesseract для распознавания текста (тайм-кода) по видеоизображениям. Выполненные с использованием DiViLab и разработанного программного модуля эксперименты показали, что для эффективного применения библиотеки к видеоданным требуется специальная предобработка изображений. Обработка необходима для подавления фона и обеспечения достаточной степени контраста распознаваемого текста по отношению к окружающим компонентам изображений [2]. В своей работе Р.М. Алиев, Д.Р. Пантич привели сравнительную характеристику методов распознавания текста на изображении, в ходе анализа выявлен наиболее оптимальный метод - нейронные сети, благодаря высокой точности распознавания и возможности работы с различными шрифтами. Представлена архитектура библиотеки tesseract OCR и результат работы библиотеки [3]. В своей работе Т.М. Куроченко рассмотрел вопросы обнаружения символов и распознавания текста с изображения при помощи Tesseract-OCR. Определение, что такое Tesseract. История: когда был создан Tesseract, как работает и почему эта программа актуальна до сих пор. А также, каким образом Tesseract-OCR взаимодействует с языком программирования Python [4]. Ю.Г. Стёпин, Н.И. Игнатенко, В.М. Локтевич Рассмотрели использование оптического распознавания подписей в технических документах на примере распознавания штрих кодов и информационных полей круговых диаграмм, используемых в газодобывающей промышленности. Описаны методы, средства и подходы к распознаванию информации различного рода: штрих-кодов, печатного текста, рукописного текста, текста определенного формата. Представленная работа позволит автоматизировать процесс переноса информации в базы данных [5].

1.3 Цель исследования

Цель исследования – создать десктопное приложение на языке программирования Python, используя библиотеки PyAutoGUI и pytesseract, с целью обеспечения возможности распознавания текста на изображении.

2 Материалы и методы

Разработка программы для распознавания текста на изображении осуществлялась с использованием языка программирования Python в среде PyCharm. Для снятия скриншотов и взаимодействия с графическим интерфейсом пользователя была использована библиотека PyAutoGUI. Для оптического распознавания символов использовалась библиотека pytesseract, интегрированная с Tesseract OCR.

3 Результаты и обсуждения

Для успешного старта разработки приложения по распознаванию текста на изображении необходимо выполнить первоначальную установку Tesseract OCR, мощного инструмента оптического распознавания символов [6]. Этот процесс представляет собой ключевой этап, обеспечивающий функциональность распознавания текста в приложении. После успешной установки Tesseract OCR, следующим этапом является установка дополнительных библиотек, необходимых для взаимодействия с Tesseract из языка программирования Python. Эти библиотеки обеспечат эффективное использование Tesseract в приложении. Установи библиотеку PyAutoGUI, pytesseract и Pillow командами «pip install pyautogui», «pip install pytesseract» и «pip install pillow» соответственно.

Пропишем основной код. Импортируем библиотеку pytesseract и модуль Image из библиотеки PIL (Python Imaging Library). Также он импортирует библиотеку tkinter для создания графического интерфейса пользователя (GUI) и модуль filedialog для работы с диалоговыми окнами выбора файлов. Также устанавливаем путь к исполняемому файлу Tesseract OCR. Это необходимо, если Tesseract не находится в системном PATH (рис. 1).



```
main.py
1 import pytesseract
2 from PIL import Image
3 import tkinter as tk
4 from tkinter import filedialog
5
6 pytesseract.pytesseract.tesseract_cmd = r'C:\Program Files\Tesseract-OCR\tesseract.exe'
7
```

Рисунок 1 – Импорт библиотек и установка пути к Tesseract OCR

Создадим функцию, предназначенную для извлечения текста из изображения. Внутри функции происходит открытие изображения с использованием библиотеки PIL с помощью Image.open. Затем библиотека pytesseract вызывается для распознавания текста на открытом изображении с помощью метода image_to_string. Результат распознавания сохраняется в переменную text и возвращается из функции. В случае возникновения ошибки при обработке изображения (например, если изображение не может быть открыто или текст не может быть распознан), функция обрабатывает

исключение, возвращая строку с сообщением об ошибке вместо текста (рис. 2).

```
9
10 def extract_text_from_image(image_path):
11     try:
12         image = Image.open(image_path)
13         text = pytesseract.image_to_string(image)
14         return text
15     except Exception as e:
16         return f"Ошибка при обработке изображения: {str(e)}"
17
```

Рисунок 2 – Создание функции «extract_text_from_image»

По задумке при запуске программы перед пользователем появляется окно с кнопкой загрузки скриншота. Функция `open_file_dialog` реализует открытие диалогового окна выбора файла с изображением. Пользователю предлагается выбрать изображение. Если пользователь выбирает файл, то полученный путь к выбранному изображению передается в функцию `extract_text_from_image`, которая затем извлекает текст из этого изображения. Результат распознавания текста затем отображается в виджете графического интерфейса пользователя. Этот код обеспечивает взаимодействие пользователя с приложением для выбора и обработки изображений с последующим отображением распознанного текста. (рис. 3).

```
def open_file_dialog():
    file_path = filedialog.askopenfilename(title="Выбор изображения", filetypes=[("Изображения", "*.png;*.jpg;*.jpeg")])
    if file_path:
        text = extract_text_from_image(file_path)
        result_label.config(text=text)
```

Рисунок 3 – Создание метода «onCreate»

Создадим графический интерфейс пользователя для приложения. Сначала инициализируется основное окно с помощью `tk.Tk()`, устанавливается его заголовок на "OCR изображения". Затем создается кнопка "Выбрать изображение" с использованием `tk.Button`, которая вызывает функцию `open_file_dialog` при нажатии. Далее создается метка `result_label` для отображения результата распознавания текста. Завершается код вызовом `app.mainloop()`, что запускает основной цикл обработки событий GUI, обеспечивая взаимодействие пользователя с приложением (рис. 4).

```
29 |
30 app = tk.Tk()
31 app.title("OCR изображения")
32
33
34 open_file_button = tk.Button(app, text="Выбрать изображение", command=open_file_dialog)
35 open_file_button.pack(pady=20)
36
37
38 result_label = tk.Label(app, text="")
39 result_label.pack(pady=20)
40
41 app.mainloop()
42
```

Рисунок 4 – Создание графического интерфейса

Тестируем приложение. В качестве скриншота, с которого нужно распознать текст, была выбрана часть кода (рис. 5).

```
import pytesseract
from PIL import Image
import tkinter as tk
from tkinter import filedialog

pytesseract.pytesseract.tesseract_cmd = r"C:\Program Files\Tesseract-OCR\tesseract.exe"
```

Рисунок 6 – Скриншот для распознавания текста

При запуске кода перед пользователем появляется окно программы. После выбора скриншота программа успешно распознала текст и отобразила в окне (рис. 7, 8).

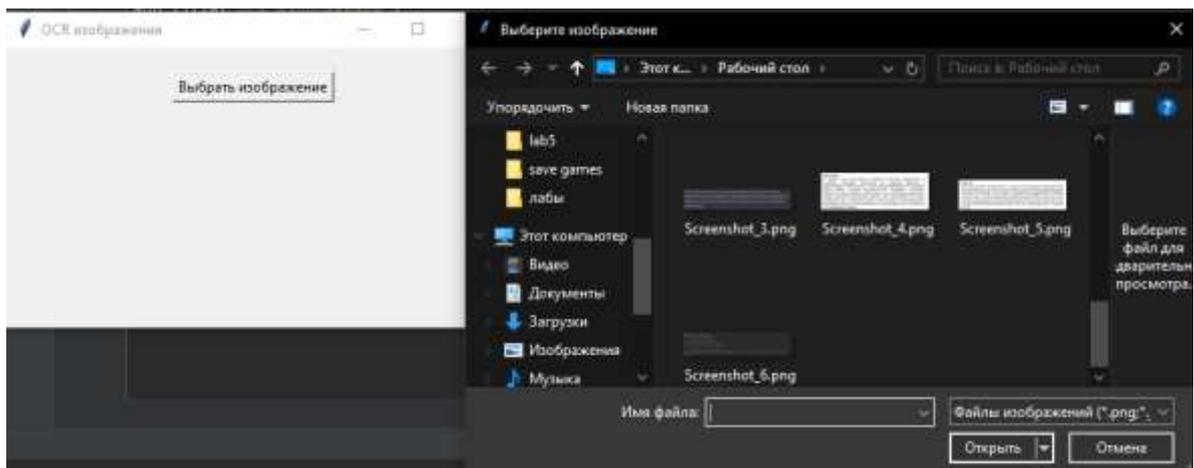


Рисунок 7 – Выбор скриншота

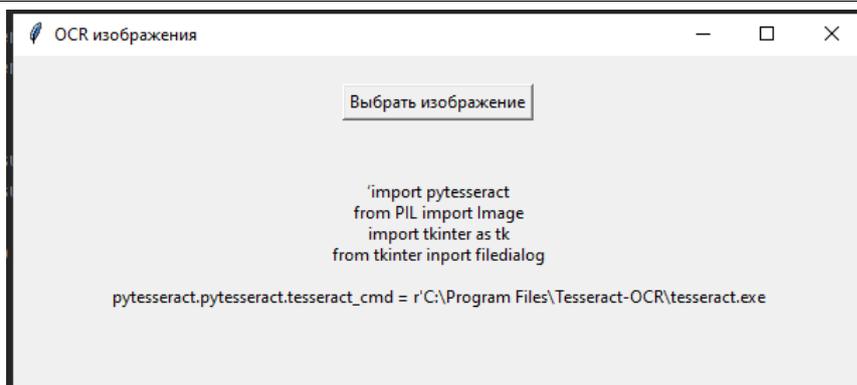


Рисунок 8 – Результат работы программы

Выводы

В данной работе был представлен пример разработки десктопного приложения на языке программирования Python для распознавания текста на изображении. Были рассмотрели важные шаги, начиная с установки Tesseract OCR и необходимых библиотек, таких как PyAutoGUI и pytesseract. Приложение было организовано с использованием библиотеки tkinter для создания графического интерфейса пользователя, который позволяет выбирать изображения и выводить результат распознавания текста. Работоспособность приложения была успешно продемонстрирована на примере обработки изображений и извлечения текста с использованием Tesseract OCR.

Библиографический список

1. Минязев Р.Ш., Дыганов С.А., Гумеров И.Р., Перухин М.Ю. Разработка сервиса для идентификации полей сканированного документа с использованием библиотеки машинного распознавания Tesseract-OCR. // Вестник Технологического университета. 2018. Т. 21. № 9. С. 132-135.
2. Нейфельд О.А., Рудникович А.С. Возможности ocr-библиотеки tesseract при распознавании текста по видеоизображениям. // Сборник избранных статей научной сессии ТУСУР. 2019. № 1-2. С. 16-19.
3. Алиев Р.М., Пантич Д.Р. Распознавание текста на изображении с помощью библиотеки Tesseract OCR. // Научные исследования молодых учёных. Сборник статей XII Международной научно-практической конференции. Пенза, 2021. С. 56-58.
4. Куроченко Т.М. Распознавание текста с изображения с помощью Tesseract-OCR. Аллея науки. 2022. Т. 1. № 12 (75). С. 741-744.
5. Стёпин Ю.Г., Игнатенко Н.И., Локтевич В.М. Оптическое распознавание подписей в технической документации. // Проблемы современной экономики: глобальный, национальный и региональный контекст. Сборник научных статей. Редколлегия: М.Е. Карпицкая (гл. ред.), С.Е. Витун (зам. гл. ред.) [и др.]. Гродно, 2021. С. 357-365.
6. Tesseract Open Source OCR Engine. URL: <https://github.com/tesseract-ocr/tesseract> (дата обращения 29.01.2024).